



TESIS PROFESIONAL

**BASE DE DATOS NO
ESTRUCTURADA PARA EL
ALMACENAMIENTO DE
IMÁGENES Y VIDEOS DEL
TRANSITO URBANO.**



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



INSTITUTO
TECNOLÓGICO
NACIONAL DE MÉXICO



VERACRUZ
GOBIERNO
DEL ESTADO



SEV
Secretaría
de Educación

SEMSys
Subsecretaría de Educación
Media Superior y Superior



DET
Departamento de Educación
Tercer Nivel y Posgrado
de Veracruz



TECNOLÓGICO
NACIONAL DE MÉXICO®



BASE DE DATOS NO ESTRUCTURADA PARA EL ALMACENAMIENTO DE IMÁGENES Y VIDEOS DEL TRANSITO URBANO

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:

Seydi Sarai Molina Morales

Asesores:

**M.C.A Sofia Isabel Fernandez
Gregorio**

**Dr. Luis Alberto Morales
Rosales**

Martínez de la Torre, Veracruz. Agosto 2021

FICHA TÉCNICA

Estudiante

Apellido paterno	Apellido materno	Nombre(s)
------------------	------------------	-----------

Molina Morales Seydi Sarai

No. de control: 16010300

Carrera: Ingeniería en Sistemas Computacionales

Correo: 160i0300@tecmartinez.edu.mx

Asesor(es) y/o colaboradores ITSMT

M.C.A Sofia Isabel Fernandez Gregorio

Datos del lugar donde se desarrolló el proyecto

Universidad Michoacana de San Nicolás de Hidalgo

Av. General Francisco j. Múgica s/n Col. Felicitas del Rio, Morelia,
Michoacán

Asesor externo: Dr. Luis Alberto Morales Rosales

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

Cargo: Catedrático

Correo: lamorales@conacyt.mx

Teléfono y extensión: +52(443) 322 3500 ext. 4346

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:	Seydí Saraf Molina Morales		
Carrera:	Ingeniería en Sistemas Computacionales	No. De control:	160I0300
Nombre Proyecto:	Base de datos no estructurada para el almacenamiento de imágenes y videos del tránsito urbano.		
Opción Titulación:	Tesis		

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

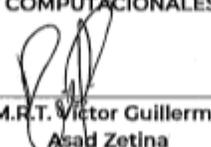
A T E N T A M E N T E

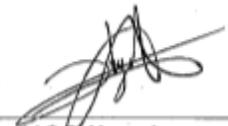

M.C. JULIO CÉSAR MACÍAS GÓMEZ
JEFE DE DIVISIÓN DE CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES



JEFATURA DE CARRERA
ING. SISTEMAS COMPUTACIONALES


M.C.A. Sofia Isabel Fernández Gregorio
ASESOR


M.R.T. Victor Guillermo Asad Zetina
REVISOR


M.S.C. Hugo Lucas Alvarado
REVISOR

C.c.p Estudiante.
 C.c.p. 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/051/2021

ASUNTO: Autorización de entrega

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

C. SEYDI SARAÍ MOLINA MORALES
No DE CONTROL 16010300
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:

Base de datos no estructurada para el almacenamiento de imágenes y videos de tránsito urbano.

Dándose 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. Keija Elena Ocaña Drouaillet
Presidenta de Academia de Ingeniería
en Sistemas Computacionales


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



JEFATURA DE CARRERA
ING. SISTEMAS COMPUTACIONALES

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

F-11-09
Rev. 1

DEDICATORIA

Dedico con todo mi corazón esta tesis a mis padres quienes dedicaron su vida a la mía para ayudarme a cumplir mis sueños, son los que me motivaron cuando pensaba que todo se terminaba, en especial a mi madre pues ella fue mi principal motor para llegar hasta aquí.

A aquellos amigos y familia que desde un principio creyeron en mi y nunca dudaron de que podía lograrlo.

AGRADECIMIENTOS

Agradezco primeramente y sobre todo a Dios por su inmenso amor y bondad, por ayudarme en los momentos difíciles, por llenarme de sabiduría y porque sin Él no hubiera podido llegar hasta aquí.

A mi familia por brindarme el apoyo y palabras de aliento cuando este proceso parecía duro y difícil, en especial a mi mamá Charito por su sacrificio y esfuerzo para poder apoyarme económicamente, quien estuvo conmigo en todo momento y me animó a no rendirme y esforzarme para alcanzar mis metas. Y así mismo un agradecimiento especial a mi hermano Jonathan pues siempre ha sido un ejemplo a seguir.

Al Instituto Tecnológico Superior de Martínez de la Torre por abrirme sus puertas y que en colaboración de sus docentes me brindaron los conocimientos necesarios con paciencia y dedicación, los cuales me han permitido avanzar hasta este punto de la carrera, gracias al personal administrativo y jefatura de esta institución que, por su trabajo y esfuerzo constante, mi estadía en este lugar y este proceso fueron más sencillos.

Así también un agradecimiento especial a mis asesores la M.C.A Sofía Isabel Fernández Gregorio y el Dr. Luis Alberto Morales Rosales quienes me orientaron tan paciente y acertadamente en el desarrollo de esta tesis, proporcionándome las guías necesarias para poder culminarla.

RESUMEN

Día a día en las vialidades urbanas transitan diferentes tipos de vehículos, y cada uno de ellos con un comportamiento propio. Es decir, cada conductor tiene diferentes hábitos de manejo que se refleja en el tránsito urbano.

En la vialidad podremos encontrar situaciones diversas como: congestión vial, accidentes de tránsito, conducción acelerada, faltas a señalamientos viales; por mencionar algunos. Por ello se considera importante obtener imágenes y videos que demuestren los comportamientos descritos. Actualmente es normal ver colocadas en estas zonas, cámaras de vigilancia vial que contribuyen al análisis de la conducción vial en zonas urbanas, sin embargo, es poco sabido que se disponga de un sistema que permita el almacenamiento de los videos e imágenes dentro de un servidor. Por lo tanto, en este trabajo se buscó generar un medio que permitiera el almacenamiento de imágenes y videos recolectados del tránsito urbano dentro de una base de datos no estructurada en la que se almacenen dichas imágenes y videos para que pudieran ser consultados y analizados posteriormente. Por lo que dentro de este trabajo se presentó el desarrollo de un sistema web que buscó asistir a este problema y hacer algunas otras funciones que facilitará el trabajo a los encargados en esta área.

Palabras Clave: *Base de datos no estructurada, paginas web, tránsito urbano, observación vial, almacenamiento de información.*

ABSTRACT

Day by day, different types of vehicles travel on urban roads, each with its own behavior. That is, each driver has different driving habits that are reflected in urban traffic. On the road we can find various situations such as: traffic congestion, traffic accidents, accelerated driving, faults in road signs; to name a few. For this reason, it is considered important to obtain images and videos that demonstrate the behaviors described. Currently it is normal to see road surveillance cameras placed in these areas that contribute to the analysis of road driving in urban areas, however, it is little known that there is a system that allows the storage of videos and images within a server. Therefore, in this work it was sought to generate a medium that would allow the storage of images and videos collected from urban traffic within an unstructured database in which said images and videos are stored so that they can be consulted and analyzed later. Therefore, within this work, the development of a web system was presented that sought to attend to this problem and perform some other functions that would facilitate the work of those in charge in this area.

Keywords: *Unstructured Database, web pages, urban traffic, road observation, information storage.*

Contenido

FICHA TÉCNICA.....	IV
FORMATO DE LIBERACIÓN DEL PROYECTO PARA TITULACIÓN INTEGRAL	V
CARTA DE AUTORIZACIÓN DE ENTREGA DE TESIS EN SOPORTE DIGITAL.....	V
DEDICATORIA.....	VII
AGRADECIMIENTOS.....	VIII
RESUMEN.....	IX
ABSTRACT	X
LISTA DE FIGURAS	XIII
LISTA DE TABLAS.....	XIV
1. INTRODUCCIÓN.....	1
2. PROBLEMA DE LA INVESTIGACIÓN.....	2
3. JUSTIFICACIÓN	3
4. OBJETIVOS DEL PROYECTO	4
4.1 Objetivo general.....	4
4.2 Objetivos específicos.....	4
5. HIPÓTESIS	5
6. MARCO TEÓRICO	6
6.1 Antecedentes o marco histórico.....	6
6.1.1 Seguridad vial.....	6
6.1.2 Sistema de video vigilancia vial.....	6
6.2 Marco conceptual.....	7
6.2.1 Bases de datos NoSQL.....	7
6.2.2 Aplicación.....	9
6.2.3 Tipos de sitios Web	12
6.2.4 Frameworks	14

6.3 Marco referencial.....	19
6.3.1 Axis Camera Station.....	19
6.3.2 Geovision VMS.	19
6.3.3 ISS SecurOS.....	19
6.3.4 Milestone XProtect.....	19
7. MATERIALES Y MÉTODOS.....	20
7.1 Recolección de información para el desarrollo de la base de datos no estructurada. ¡Error! Marcador no definido.	
7.2 Análisis de parámetros de imágenes y videos ¡Error! Marcador no definido.	
7.3 Diseño de la base de datos nosql..... ¡Error! Marcador no definido.	
7.4 Creación de la base de datos en cassandra..... ¡Error! Marcador no definido.	
7.4.1 Creación de las tablas que conforman a la base de datos. ¡Error! Marcador no definido.	
7.5 Desarrollo de un sistema web para el manejo de la base de datos no estructurada. ¡Error! Marcador no definido.	
8. RESULTADOS Y DISCUSIÓN	33
8.1 Base de datos no estructurada	33
8.2 Desarrollo de página web	34
9. CONCLUSIONES	37
10. RECOMENDACIONES	38
11. REFERENCIAS BIBLIOGRÁFICAS	39

LISTA DE FIGURAS

Figura 1. Esquema de ejecución de las diferentes partes de una aplicación web.....	11
Figura 2. Descripción gráfica de un sitio web estático.....	13
Figura 3. Descripción gráfica de un sitio web dinámico.....	14
Figura 4. Representación del patrón MVC (Modelo, Vista, Controlador).....	17
Figura 5. Creación de la base de datos “estudiovial”.....	25
Figura 6. Instrucción para usar y manipular base de datos “estudiovial”.....	26
Figura 7. Creación de las tablas usuario, imágenes y videos en la base de datos.....	26
Figura 8. Inserción de nuevo dato de tipo blob dentro de tablas imágenes y videos.....	27
Figura 9. Instalación global de Vue Cli 4.....	27
Figura 10. Selección del gestor de paquetes “npm” para el proyecto.....	28
Figura 11. Selección de preset "manual" para la elección libre de características.....	28
Figura 12. Selección de características que conformaran el proyecto.....	29
Figura 13. Estructura del proyecto del sistema web " estudio-vial".....	29
Figura 14. Codificación para la vista "Inicio" del sistema web.....	30
Figura 15. Código para el desarrollo del formulario "Añadir Imagen". Primer parte.....	31
Figura 16. Código para el desarrollo del formulario "Añadir Imagen". Segunda parte.....	31
Figura 17. Codificación del formulario "Añadir Videos". Primera parte.....	32
Figura 18. Codificación del formulario "Añadir videos". Segunda parte.....	32
Figura 19. Tabla "imágenes" con datos insertados.....	33
Figura 20. Tabla "videos" con datos insertados.....	34
Figura 21. Tabla "usuario" con datos insertados.....	34
Figura 22. Vista principal de la página web.....	35
Figura 23. Sitio de la página web para subir las imágenes.....	35
Figura 24. Sitio de la página web para subir los videos.....	36

LISTA DE TABLAS

Tabla 1. Tipos de datos en Cassandra y su equivalencia en JavaScript	23
Tabla 2. Representacion de la tabla imagenes que conforma la BD.	24
Tabla 3. Representacion de la tabla usuario que conforma la BD.....	24
Tabla 4. Representacion de tabla videos que conforma la BD.....	24

1. INTRODUCCIÓN

Un estudio de tránsito tiene como finalidad analizar la movilidad en una zona determinada, contemplando de manera coordinada los diferentes elementos que la componen y simulando la interacción de los diversos proyectos viales con la red proyectada o existente, realizando un diagnóstico que proporcione soluciones ajustadas a cada proyecto con el fin de obtener una movilidad eficiente, segura y además comprometida con el medio ambiente. Con el paso de los años y la tecnología avanzando, una amplia variedad de sistemas de gestión se ha desarrollado para el control del tránsito urbano. Pensando en facilitar la labor de estudio del tránsito vial, el propósito del siguiente proyecto es la elaboración de una base de datos no estructurada para el almacenamiento y consulta de videos e imágenes recolectados por cámaras colocadas en determinadas vialidades, haciendo uso de un sistema web para que el usuario pueda ingresar a la base desde cualquier dispositivo a través de la red.

Este documento se conforma de 3 capítulos, en el primer capítulo se abordan las generalidades del proyecto conformado primeramente por el problema principal a resolver, la justificación del problema donde se plantea la importancia y la necesidad que se tiene de implementar una base de datos no estructurada para el almacenamiento de imágenes y videos recolectados para el estudio vial, los objetivos y la hipótesis, de acuerdo a los prerequisites establecidos por el departamento.

El segundo capítulo se conforma del marco teórico en donde se colocaron antecedentes y conceptos teóricos relacionados con el desarrollo del proyecto, así como la metodología que se utilizó para el desarrollo del sistema web en general.

En el tercer capítulo se plasma el desarrollo del proyecto en donde se describe el proceso de las actividades realizadas, para esto se tomó en cuenta la elaboración de un cronograma para llevar a cabo cada actividad durante un periodo establecido.

2. PROBLEMA DE LA INVESTIGACIÓN

En la universidad Michoacana de san Nicolás de Hidalgo, específicamente en el área de Maestría en Infraestructura del transporte en la rama de las vías terrestres se carece de un sistema de almacenamiento para los videos e imágenes generados por cámaras de vigilancia vial que se estudian y que seria de gran utilidad especialmente para quienes participan en esta área ya que deben realizar el análisis de conducción vial en zonas urbanas, a través de cantidades grandes de videos e imágenes generados diariamente, por lo que actualmente este trabajo puede llegar a ser complicado o tedioso para ellos. Al analizar dicha situación se destacan particularmente los principales problemas a resolver de la empresa, los cuales son, la falta de una base de datos que permita el almacenamiento de grandes volúmenes de datos para el almacenamiento de las imágenes y videos extraídos de las cámaras de vigilancia vial, se puede pensar también en la falta de organización de la información (imágenes y videos) almacenada y así mismo la falta de seguridad de la información.

3. JUSTIFICACIÓN

Los principales impactos que analiza un estudio de tránsito son la congestión y la accidentalidad relacionada, por lo que los elementos que normalmente componen un estudio de tránsito son la observación y análisis de, Congestionamiento vial, Accidentes de tránsito, conducción acelerada, faltas a señalamientos viales y tipo de vialidad. El área de Maestría de infraestructura del transporte en la rama de las vías terrestres de la UMSNH que se encarga de realizar estos estudios hasta el momento lo hace directamente de un ordenador en el que se guardan los videos e imágenes generados por las cámaras.

Es por ello que surge la idea de crear una base de datos no estructurada que almacene imágenes y videos del tránsito urbano, dirigida a los encargados de estos estudios para hacer su función más fácil y menos tediosa mediante un sistema web que contenga:

- Modulo para cargar los videos e imágenes a la base de datos
- Modulo para la búsqueda de los archivos almacenados en la base
- Visualización de los videos e imágenes.

La propuesta de desarrollar un sistema web surge de analizar que son aplicaciones a las que se podrán acceder mediante cualquier dispositivo sin importar su sistema operativo o tipo de dispositivo, ya que se adaptan de la misma forma, tanto a un celular como a una computadora y aunque el inconveniente de estas aplicaciones es que se debe tener una conexión obligatoria a internet para su utilización, la gran mayoría de usuarios las prefieren por su bajo costo.

4. OBJETIVOS DEL PROYECTO

4.1 Objetivo general

Desarrollar una pagina web que permita el almacenamiento de imágenes y videos recolectados por cámaras instaladas sobre áreas de transito urbano que faciliten la realización del estudio vial.

4.2 Objetivos específicos

- Determinar los parámetros requeridos para el almacenamiento en la base de datos no estructurada que permita realizar un futuro análisis de videos.
- Analizar los requerimientos funcionales que se van a necesitar para obtener los módulos de información que deberá contener la base de datos a diseñar.
- Crear la base de datos no estructurada en el servidor para consulta y administración.
- Diseñar una pagina web que permita la manipulacion de la base de datos no estructurada.

5 HIPÓTESIS

La creación de un sistema web que almacene imágenes y videos del tránsito urbano dentro de una base de datos no estructurada, proporciona mayor escalabilidad en el resguardo de la información y además permitirá a los encargados del análisis del trafico realizar su tarea con más rapidez y facilidad al organizar los datos.

6 MARCO TEÓRICO

6.2 Antecedentes o marco histórico

Para sustentar la propuesta que se plantea en esta investigación, se presentan dos antecedentes sobre sistemas de seguridad vial que están relacionados con el objeto en estudio, el primer antecedente menciona la importancia de realizar análisis de los comportamientos viales y el segundo menciona el uso de un sistema web para la vigilancia vial.

6.1.1 Seguridad vial

Gallo, A. & Castillo, G (2018). En su trabajo de grado “análisis de las condiciones de seguridad vial ligadas a temas de infraestructura en las vías rápidas de Bogotá” mencionan que la movilidad es uno de los factores más importantes que determina el desarrollo de las comunidades y regiones en un país, dado que la infraestructura vial se comporta como uno de los factores que favorecen una ágil movilidad y que, marca una pauta importante en las características de medición del crecimiento económico y la comodidad de las ciudades. Es por lo anterior, que la identificación de características de valoración de la movilidad, el comportamiento de los conductores y con ello de la accidentalidad, determinan aspectos de gran importancia en los análisis que se realizan para determinar los riesgos que implica movilizarse por las calles y avenidas.

6.1.2 Sistema de video vigilancia vial

El sistema de video vigilancia es uno de los sistemas más empleados en los últimos años se habla de seguridad, que se fundamenta en la comunicación mediante un monitoreo, que puede ser vigilado desde un sitio web. Hoy en día es muy usado el internet ya que este ha alcanzado una gran demanda en diferentes procesos para el desarrollo de la mayoría de las actividades. Con ayuda del internet es posible obtener resultados a distancia de alta calidad con ayuda de las cámaras de vigilancia. El monitoreo a distancia se realiza con una estructura cliente- servidor para así compartir información. El sistema de video vigilancia puede ser manejado desde cualquier computador desde el explorador, cabe destacar que este computador debe ser

conectado a internet (Blanco, 2006).

Los sistemas de vigilancia actualmente son muy comunes en vialidades urbanas para contribuir al monitoreo de la seguridad vial y además a la realización de estudios que son necesarios para mejorar el transito urbano.

6.3 Marco conceptual

6.2.1 Bases de datos NoSQL

Una base de datos no relacional no requiere de estructuras de datos fijas (tablas), no da completas garantías de las características ACID y realiza escala horizontal de forma óptima. Se hace uso de ellas en entornos distribuidos que deban estar continuamente disponibles y operativos y que además gestionan un gran volumen de datos. En la actualidad debido a la gran demanda que están generando estas bases de datos, existen ya diversos gestores de este tipo, a continuación, se mencionan algunos de ellos.

6.2.1.1 Apache Cassandra.

Apache Cassandra es un sistema de gestión de bases de datos (DBMS) de código abierto para bases de datos muy grandes, pero estructuradas. Gracias a la buena escalabilidad, estas bases de datos se pueden distribuir a diferentes clústeres, por lo que Cassandra no se encuentra unida a un único servidor. Como base de datos NoSQL, Cassandra cuenta con un enfoque redundante, lo que reduce mucho la probabilidad de fallo. Sin embargo, las bases de datos relacionales suelen tener problemas al replicar los datos. Cassandra usa CQL (Cassandra Query Language). Es un dialecto de SQL para definir y manipular los datos, (IONOS, 2021). Entre las características más importantes se encuentra que Cassandra no soporta operaciones de tipo JOIN. Entre los usos más comunes de Apache Cassandra se encuentran las aplicaciones IoT, en estos sistemas, es frecuente que se tengan miles de sensores enviando información variable. En ocasiones, estos eventos deben ser sincronizados y correlacionados en tiempo real. La monitorización de aplicaciones web, cuando se es necesario almacenar la actividad de los usuarios en sitios web, (Fernández, 2021).

6.2.1.2 Redis.

Redis es un almacén de estructura de datos en memoria de código abierto (con licencia BSD), que se utiliza como base de datos, caché y agente de mensajes. Redis proporciona estructuras de datos como cadenas, hashes, listas, conjuntos, conjuntos ordenados con consultas de rango, mapas de bits, hiperloglogs, índices geoespaciales y flujos. Redis tiene replicación incorporada, secuencias de comandos Lua, desalojo de LRU, transacciones y diferentes niveles de persistencia en disco, y proporciona alta disponibilidad a través de Redis Sentinel y particionamiento automático con Redis Cluster, (Sanfilippo, 2015).

6.2.1.3 MongoDB.

MongoDB es una base de datos orientada a documentos. Esto quiere decir que, en lugar de guardar los datos en registros, guarda los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON. Una de las diferencias más importantes con respecto a las bases de datos relacionales, es que no es necesario seguir un esquema. MongoDB está escrito en C++, aunque las consultas se hacen pasando objetos JSON como parámetro. Es algo bastante lógico, dado que los propios documentos se almacenan en BSON, (Rubenfa, 2014).

6.2.1.4 CouchDB.

CouchDB es una base de datos NoSQL de código abierto basada en estándares comunes para facilitar la accesibilidad y compatibilidad web con una diversidad de dispositivos. Los datos en CouchDB se almacenan en el formato de notificación de objetos JavaScript (JSON), y están organizados en pares de valor clave. La clave es un identificador único de los datos, y el valor es el propio dato o un apuntador a la ubicación de los datos. Las funciones estándar de la base de datos son realizadas por JavaScript. Los estándares web agnósticos de sistema operativo, e independientes de dispositivos permiten a las bases de datos desempeñarse bien en la variedad más amplia de usuarios, (Margaret Rouse, 2017).

En este estudio de investigación se opta por el uso de Casandra debido a su gran

escalabilidad y disponibilidad sin comprometer su rendimiento, ya que se manejarán cantidades altas de archivos con grandes pesos.

6.2.2 Aplicación

Una aplicación es un programa informático diseñado como una herramienta para realizar operaciones o funciones específicas. Generalmente, son diseñadas para facilitar ciertas tareas complejas y hacer más sencilla la experiencia informática de las personas.

Cabe destacar que, aunque todas las aplicaciones son programas, no todos los programas son aplicaciones. Existe multitud de software en el mercado, pero sólo se denomina así a aquel que ha sido creado con un fin determinado, para realizar tareas concretas.

Las aplicaciones nacen de alguna necesidad concreta de los usuarios, y se usan para facilitar o permitir la ejecución de ciertas tareas en las que un analista o un programador ha detectado una cierta necesidad.

A la hora de desarrollar una aplicación, existen cuatro tipos: app nativa, app híbrida, web app y React Native.

6.2.2.1 Aplicaciones nativas.

Las Apps nativas son aquellas aplicaciones que están desarrolladas para un equipo o plataforma determinada. Es decir, funciona en el equipo sin necesidad de ningún programa externo ya que se ha desarrollado en el lenguaje de programación específico de cada equipo. El término de App Nativa está habitualmente asociado a los dispositivos móviles y por tanto hay Apps Nativas para cada sistema operativo como iOS o Android.

La principal ventaja de Native Apps es que se puede adaptar al 100% al dispositivo, y puedes utilizar todas sus funciones sin encontrarte con mayores problemas (rendimiento gráfico, GPS, cámara, acelerómetro).

La principal desventaja es que, si desea instalar la aplicación para diferentes sistemas, debe copiar el proyecto una vez para cada sistema (iOS, Android) donde desea instalar la aplicación y copiar la versión del proyecto varias veces. Esto aumenta enormemente el tiempo y el costo de desarrollo, (Cardenas, 2019).

6.2.2.2 Aplicación web

Son aplicaciones desarrolladas con tecnología Web y pueden utilizarse accediendo a un servidor Web a través de Internet (o Intranet) a través de un navegador. Para ser precisos, el hecho de que se pueda acceder a ellos desde un navegador web ofrece a las aplicaciones web las mayores ventajas (y desventajas).

Entre las ventajas de las Web apps están la sencillez de desarrollo y su menor coste. En este caso se desarrolla una única web app que es distribuida a todos los usuarios mediante el uso del navegador, sea desde un dispositivo móvil o desde un dispositivo de escritorio. Así utilizando el diseño responsive se puede hacer que una aplicación web se adapte a cualquier tipo de dispositivo.

El principal inconveniente de las aplicaciones web es que no es posible acceder a todas las funciones específicas del dispositivo (como rendimiento gráfico, GPS, cámara, acelerómetro, altímetro, contactos, etc.). También podría ofrecer un nivel algo más bajo de seguridad ya que la misma depende de la seguridad del propio navegador utilizado, (Pérez, 2017).

En una aplicación web, como un gestor de contenidos o una tienda en línea o cualquier otra web que haga uso de bases de datos, podemos encontrar hasta cuatro tipos de desarrollo:

1. Desarrollo de HTML + CSS
2. Desarrollo JavaScript
3. Desarrollo en PHP
4. Consultas SQL

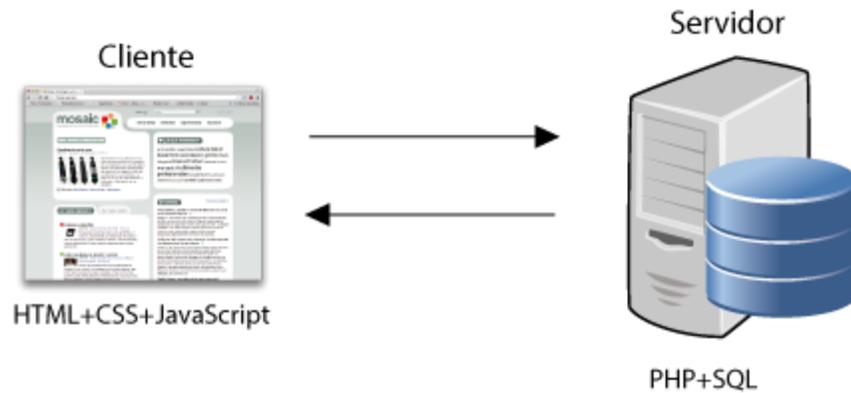


Figura 1. Esquema de ejecución de las diferentes partes de una aplicación web.

6.2.2.3 Aplicaciones híbridas.

Las aplicaciones híbridas son una mezcla de las dos anteriores. En ellas se aprovecha la versatilidad del desarrollo de una aplicación web con HTML5, CSS y Javascript, permitiendo el uso de algunas de las funciones hardware, disponibles en las aplicaciones nativas y que no se podían usar en las web apps. Para ello la app desarrollada, en vez de visualizarse en un navegador web, como en el caso de las web apps, se visualiza en un WebView que viene a ser un navegador integrado en una aplicación nativa y que por tanto puede utilizar esos recursos hardware específicos por medio de plugins.

La principal ventaja de las App Híbridas, además de la posibilidad de usar recursos hardware directamente, es que son independientes del sistema operativo (iOS y Android). Eso permite reducir la inversión inicial en el proyecto a medio plazo respecto a las App nativas, ya que no requiere mantener varias versiones diferentes dependiendo del sistema operativo.

Por contra, el principal inconveniente respecto a las Apps Nativas, es que la experiencia de usuario no es tan buena como en estas y el rendimiento y la velocidad de carga es mucho más lenta que una App Nativa, (Pérez, 2017).

6.2.2.4 React Native.

Se trata de la tipología de app más reciente, toda una evolución a la hora de desarrollar aplicaciones móviles. En este caso, se basa en la tecnología React Native, que permite crear apps

nativas simplificando los procesos de desarrollo y haciendo posible desarrollarlas bajo un único código tanto para iOS como para Android.

React Native utiliza un lenguaje de programación Javascript y el gestor de paquetes NPM, lo que es sinónimo de garantía y estabilidad a largo plazo. Este tipo de aplicaciones optimiza costes, ya que, al contrario de la nativa, no requiere programar para cada plataforma y además asegura que la experiencia de usuario será igual que si se tratase de una app nativa, (Pérez, 2017).

6.2.3 Tipos de sitios Web

6.2.3.1 Web estáticas

Una web estática es un archivo de texto HTML que despacha el servidor hacia el navegador, por lo que requiere menor procesamiento del lado servidor, ya que solo necesita localizar el archivo HTML, leerlo y despacharlo al navegador (Figura 2). Son ideales para las empresas que no quieren muchas pretensiones con su sitio web, sino que simplemente informan a sus clientes sobre sus productos y dan a conocer ampliamente la imagen de su empresa. Su principal ventaja es que son baratos. Su diseño y desarrollo son más rápidos que los sitios web dinámicos, pero los usuarios no pueden elegir, ordenar o modificar el contenido o diseño de la página, (BrandMedia, 2019).



Figura 2. Descripción gráfica de un sitio web estático.

6.2.3.2 Web dinámicas

Las páginas web dinámicas son generadas en el servidor por la ejecución del lenguaje Script sobre el cual se encuentran programadas y el resultado es una salida HTML despachada al navegador, además hacen uso de bases de datos para el almacenamiento y gestión de información (Figura 3). Se hicieron necesarias hasta el momento de que crecieron tantas que surgió la necesidad de separar el contenido en una estructura mejor adaptada para su administración y lo mejor fue usar una base de datos para gestionarlo. Estas permiten la creación de aplicaciones dentro de la propia web y brindan una mayor interactividad con los usuarios que acceden a ella. Su creación es más complicada porque requiere conocimientos específicos de lenguaje de programación y gestión de bases de datos. Con este tipo de página web, puedes hacer lo que quieras. La Web 2.0 es en realidad una Web dinámica en la que los usuarios pueden interactuar con la información contenida en la Web, ya que dicha información cambiará en tiempo real según la elección del usuario, (BrandMedia, 2019).



Figura 3. Descripción gráfica de un sitio web dinámico.

6.2.4 Frameworks

Cuando se habla de desarrollo web, es importante mencionar los marcos. Con el paso del tiempo y la complejidad de la tecnología, los estándares de las aplicaciones Web mejoran constantemente. El Frameworks trae bibliotecas con código pre-escrito y características adicionales, que permiten a los desarrolladores crear aplicaciones web ricas e interactivas, (Glajumedia, 2020). A continuación, se presentan algunos de los frameworks más populares entre los desarrolladores en la actualidad.

6.2.4.1 Django-phyton framework

Es un framework que funciona alrededor de la calidad de las aplicaciones web. Django se usa generalmente para el rápido desarrollo de API y aplicaciones web. Sin embargo, funciona mucho mejor para aplicaciones web con una sólida administración de back-end, y si se maneja mucho tráfico en el sitio o aplicación, Django es la mejor opción. Django se convierte en una excelente opción para la gestión de inventario, el sistema CRM y los sitios de redes sociales.

6.2.4.2 Spring

Es un framework de Modelo-Vista-Controlador para aplicaciones basadas en Java. Spring es realmente útil cuando administra transacciones confiables, almacenamiento en caché, seguridad o registro. Por ejemplo, empresas como Wix, TicketMaster y BillGuard usan este framework para arrancar su rendimiento y les permiten escalar mucho más rápido.

6.2.4.3 Laravel

Es el mejor framework de desarrollo web de PHP que sigue el patrón Modelo-vista-controlador y un marco de fondo completo. Los desarrolladores generalmente trabajan con Laravel para crear funcionalidades sólidas, como autenticación, integración de API, ejecución de vulnerabilidades habituales, habilitación de pruebas web de automatización y almacenamiento en caché. Laravel tiene la capacidad de retransmitir eventos en tiempo real y crear una sólida base y procesamiento de trabajos. Funciona esencialmente con aplicaciones web como comunidades en línea, sitios de comercio electrónico, aplicaciones de redes sociales, CRM y sistemas CMS.

6.2.4.4 React

React no es un framework, pero es una biblioteca frontend de JavaScript creada por desarrolladores de Facebook. Aun así, muchos desarrolladores consideran React un framework, porque tiene una arquitectura con componentes excelentes. Por ejemplo, compañías como Facebook, Instagram, Netflix y Apple lo usan. De hecho, React admite diferentes tecnologías e idiomas, lo que permite la combinación de nuevas características. Además, tiene una interpretación rápida del código basado en HTML en el diseño de página real.

6.2.4.5 Angular

Este framework de frontend fue desarrollado por Google, que actualmente posee con una alta demanda. Se ejecuta en JavaScript, que es realmente útil para crear sitios web de una sola página. Angular usa HTML para construir interfaces de usuario y rompe la funcionalidad en los módulos. Además, este framework es compatible con la última versión de Chrome, Firefox, Edge, Android y iOS.

6.2.4.6 Vue.js

Es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas Single-Page Applications cuando se utiliza en combinación con herramientas modernas y librerías de apoyo. Vue.js es el más reciente framework para el desarrollo web, y además se está convirtiendo muy popular entre los desarrolladores debido a sus increíbles características. Es un framework progresivo, esto significa que un proyecto ya desarrollado se puede adaptar a Vue sin ningún problema. De hecho, Vue tiene un generador de proyectos CLI con un asistente intuitivo que hace posible esta adaptación. Para comprender como trabajan los frameworks Web existentes es imprescindible conocer el patrón MVC.

6.2.4.7 Patrón MVC (Modelo vista controlador).

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema (Figura 4). La mayoría, por no decir todos, de los frameworks para Web implementan este patrón. La mayoría de frameworks Web se encargan de ofrecer una capa de controladores de acuerdo con el patrón MVC, ofreciendo mecanismos para facilitar la integración con otras herramientas para la implementación de las capas de negocio y presentación, (Gutiérrez, 2020).

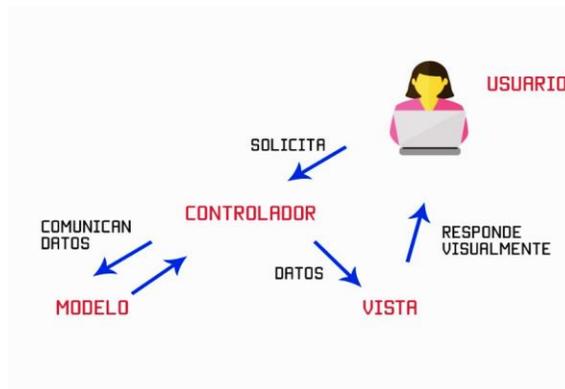


Figura 4. Representación del patrón MVC (Modelo, Vista, Controlador).

6.2.5 metodologías de desarrollo de software

En la actualidad se pueden diferenciar dos grandes grupos de metodologías de desarrollo de software: las ágiles y las tradicionales. A continuación, se explican las características de cada una de ellas.

6.2.5.1 Metodologías de desarrollo de software tradicionales

Se caracterizan por definir total y rígidamente los requisitos al inicio de los proyectos de ingeniería de software. Los ciclos de desarrollo son poco flexibles y no permiten realizar cambios, al contrario que las metodologías ágiles; lo que ha propiciado el incremento en el uso de las segundas.

La organización del trabajo de las metodologías tradicionales es lineal, es decir, las etapas se suceden una tras otra y no se puede empezar la siguiente sin terminar la anterior. Tampoco se puede volver hacia atrás una vez se ha cambiado de etapa. Estas metodologías, no se adaptan nada bien a los cambios, y el mundo actual cambia constantemente (Santander Universidades, 2020).

6.2.5.2 Metodologías de desarrollo de software ágiles

Las metodologías ágiles de desarrollo de software son las más utilizadas hoy en día debido a su alta flexibilidad y agilidad. Los equipos de trabajo que las utilizan son mucho más productivos y eficientes, ya que saben lo que tienen que hacer en cada momento. Además, la metodología permite adaptar el software a las necesidades que van surgiendo por el camino, lo que facilita construir aplicaciones más funcionales (Santander Universidades, 2020).

Las principales metodologías ágiles son:

Kanban: Consiste en dividir las tareas en porciones mínimas y organizarlas en un tablero de trabajo dividido en tareas pendientes, en curso y finalizadas. De esta forma, se crea un flujo de trabajo muy visual basado en tareas prioritarias e incrementando el valor del producto. Según un nuevo estudio de (EALDE Business School, 2020)

Scrum: Scrum es un marco de trabajo para los desarrollos ágiles. Sus principales características son:

- Adoptar una estrategia de desarrollo incremental, se planifican varios bloques temporales (de dos semanas o 1 mes, por ejemplo) llamados iteraciones. Cada iteración incluye: planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. De manera incremental, se irán completando los requisitos.
- Priorización, estos requisitos se irán cumpliendo dando prioridad primero a los que aporten más valor al cliente. Así conseguimos descargar presión a los desarrolladores del proyecto y el cliente recibirá los resultados más importantes en las primeras iteraciones.
- Equipos auto organizados
- Desarrollo de fases de forma paralela. Si las fases del proyecto son independientes, avanzar en más de una fase a la vez acortará los tiempos del proyecto. (Quonext, 2020)

Lean: La metodología Lean es una forma innovadora de gestionar los procesos de una empresa. Su objetivo es eliminar actividades que no aportan valor, para así poder obtener un producto o servicio de mayor calidad y que mejore la experiencia de los clientes.

Resumiendo, se trata de optimizar los procesos empresariales (productivos y de gestión), con el fin de utilizar menos recursos en los mismos. (Humanes, 2019)

Programación extrema (XP): Su principal objetivo es crear un buen ambiente de trabajo en equipo y que haya un feedback constante del cliente. El trabajo se basa en 12 conceptos: diseño sencillo, testing, refactorización y codificación con estándares, propiedad colectiva del código, programación en parejas, integración continua, entregas semanales e integridad con el cliente, cliente in situ, entregas frecuentes y planificación, (Santander Universidades, 2020).

6.3 Marco referencial

En la actualidad existen diferentes sistemas y aplicaciones desarrolladas con un mismo enfoque al almacenamiento de imágenes y videos o sistemas que se enfocan en el estudio vial, sin embargo, hasta el momento no se encontró algún sistema igual o con características similares al planteado en el objetivo de este proyecto, por mencionar que si hay sistemas web que permiten el almacenamiento de imágenes y videos tenemos a Google drive, Microsoft one drive, Amazon prime, Mega, Dropbox y MediaFire, por mencionar algunos, y aun que estos servicios son libres y gratuitos solo permiten el almacenamiento de archivos con cantidades limitadas (Martí, 2020)

Existen diversas plataformas de software para la gestión de videos de vigilancia como los que proporciona la empresa Critical Solution.

6.3.1 Axis Camera Station.

Es la solución ideal para garantizar una videovigilancia eficaz en instalaciones pequeñas y medianas, como por ejemplo establecimientos, hoteles, centros educativos y plantas de producción (Critical solutions, 2018).

6.3.2 Geovision VMS.

Es un sistema de manejo de video que permite grabar hasta 64 canales de video tanto de GeoVision como de terceros. Adicionalmente dispone de una variedad de análisis de video inteligente (Critical solutions, 2018).

6.3.3 ISS SecurOS.

Es el estándar global en gestión de video y analíticas, SecurOS es el núcleo de una solución completa de videovigilancia y seguridad. El sistema puede llegar a tener la capacidad de manejar y monitorear un número ilimitado de cámaras (análogas e IP) y dispositivos, integrarse con sistemas de analítica de video u otros sistemas, para crear una interfaz de control centralizada (Critical solutions, 2018).

6.3.4 Milestone XProtect.

Es un potente software de gestión de vídeo (VMS) líder global sobre IP, diseñado para implementación de soluciones a gran escala y de alta seguridad. Su interfaz de gestión única

permite la administración eficiente del sistema, incluidas todas las cámaras y dispositivos de seguridad, independientemente de su tamaño o de si está distribuido entre diversos sitios.

Aunque sean servicios dedicados a la gestión de videos de vigilancia, no son sistemas que cuenten con las características que suplan las necesidades que se plantean en el problema, ya que lo que se plantea es, el desarrollo de un sistema web que permita el almacenamiento, organización y visualización de videos e imágenes de vigilancia vial de acuerdo a características que se almacenen en la misma base de datos y que permitan la identificación de cada video o imagen para posteriores búsquedas (Critical solutions, 2018).

7 MATERIALES Y MÉTODOS

Actividad 1. Recoleccion de información para el desarrollo de la base de datos no estructurada: Para esta etapa de desarrollo, como inicio del proyecto, lo que se hizo fue la recopilación de la información sobre la base de datos que se escogió. Cómo el gestor de base de datos no estructurado que se eligió fue Cassandra, se investigó, sus características, su lenguaje, los tipos de datos que maneja y las instrucciones básicas para la creación de tablas y consultas.

Actividad 2. Recoleccion de información sobre los tipos de datos en el lenguaje CQL: Se recolectó la información de la equivalencia de los tipos de datos en CQL a los tipos de datos en JavaScript (Tabla 1) ya que se sabía que el objetivo del proyecto consistía en el desarrollo de un sistema web que permitiría el manejo de la base de datos.

Actividad 3. Analizar los parámetros de imágenes y videos: Mediante el análisis de 2 videos proporcionados por la empresa se obtuvieron sus datos, ambos videos tenían los parámetros similares por lo que se piensa que cualquier video de ese tamaño y calidad contara con los mismos parámetros.

Actividad 4. Diseñar la base de datos no estructurada: Para construir el diseño de la base de datos no estructurada de almacenamiento de imágenes y videos del tránsito urbano se analizó y recolectó la información necesaria, se identificaron los módulos de información que nos permitieron organizar el almacenamiento de los videos, se diseño y documento la base de datos no estructurada.

Actividad 5. Creación de la base de datos en Cassandra.

Actividad 6. Creacion de las tablas que conforman la base de datos: Posteriormente se crearon las tablas dentro de la base de datos, para comprobar que se construyó correctamente la base de datos, se accedió a ella mediante la instrucción USE estudiantil.

Actividad 7. Desarrollo de un sistema web para el manejo de la base de datos no estructurada: Para el desarrollo del sistema web se realizó la instalación de algunas herramientas que se usaron en la creación del sistema, las cuales fueron, el editor de código visual studio code, y node.js el cual es considerado un entorno JavaScript que nos permite ejecutar en el servidor, de manera asíncrona, con una arquitectura orientada a eventos y basado en el motor v8 de google, (Cabana, 2017).

Una vez instaladas principalmente estas dos herramientas, se utilizó el framework vue.js para la construcción del sistema web. Para la instalación de este framework se usó la terminal de visual studio code y a través de npm que es el gestor de paquetes de Node js se hizo la

instalación global de Vue Cli 4 (Figura 9).

Tipos de datos en Cassandra y JavaScript

CQL data type	JavaScript type
ascii	String
bigint	Long / BigInt
blob	Buffer
boolean	Boolean
counter	Long / BigInt
date	LocalDate
decimal	BigDecimal
double	Number
duration	Duration
float	Number
inet	InetAddress
int	Number
list	Array
map	Object / ECMAScript 6 Map
set	Array / ECMAScript 6 Set
smallint	Number
text	String
time	LocalTime

CQL data type	JavaScript type
timestamp	Date
timeuuid	TimeUuid
tinyint	Number
tuple	Tuple
uuid	Uuid
varchar	String
varint	Integer

Tabla 1. Tipos de datos en Cassandra y su equivalencia en JavaScript

Instrucciones básicas de Cassandra (Aggarwal, 2017).

- Crear una base de datos “KeySpace”

```
CREATE KEYSPACE IF NOT EXISTS k1 WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '1'} AND durable_writes = true;
```
- Usar una base de datos

```
cassandra@cqlsh> USE k1;
```
- Crear y alterar una tabla

```
cassandra@cqlsh:k1> CREATE TABLE person (id text, name text, surname text, PRIMARY KEY (id));  
cassandra@cqlsh:k1> ALTER TABLE person ADD email text;
```
- insertar y actualizar datos

```
cassandra@cqlsh:k1> INSERT INTO person (id, name, surname, email);  
cassandra@cqlsh:k1> UPDATE person SET email='shalabh.agrwal@gmail.com' WHERE id='001';
```

- consultar datos

```
cassandra@cqlsh:k1> SELECT * FROM person;
```

```
cassandra@cqlsh:k1> SELECT name FROM person WHERE id='001';
```

- Eliminar tablas de base de datos

```
cassandra@cqlsh:k1> DROP TABLE person;
```

Los datos obtenidos de ambos video son los siguientes:

Tamaño: 1.85 gb

Resolución: 1280 x 720 pixeles

Velocidad de datos: 14983 kbps

Velocidad de bits: 15111 kbtps

Duracion: 17:34

Diseño de la estructura de la base de datos.

IMAGENES
Congestionamientos
Comportamiento vial
Tipo de vialidad
Fecha
Hora
Latitud
Longitud

Tabla 3. Representacion de la tabla imagenes que conforma la BD.

USUARIO
Nombre
Correo electrónico
Contraseña

Tabla 2. Representacion de la tabla usuario que conforma la BD.

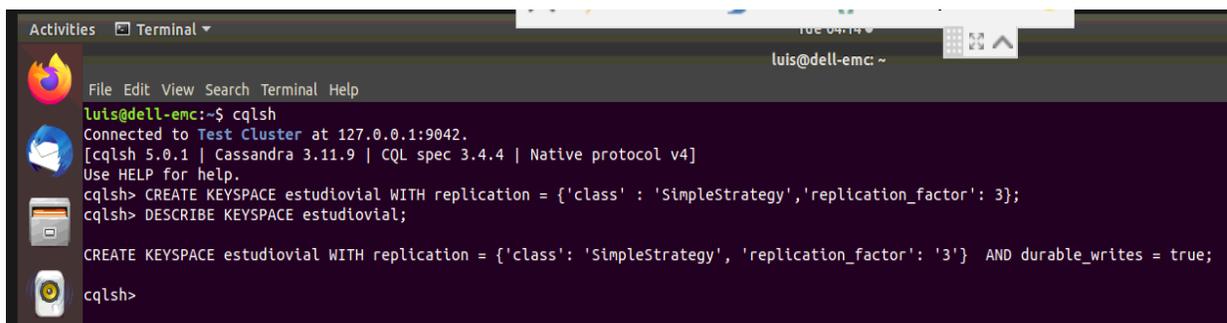
VIDEOS
Congestionamientos
Comportamiento vial
Tipo de vialidad
Fecha
Hora
Latitud
Longitud

Tabla 4. Representacion de tabla videos que conforma la BD.

La base de datos está compuesta de dos tablas, en donde se almacenan videos e imágenes respectivamente, tomando en cuenta los atributos para caracterizar y organizar los archivos de acuerdo a lo que se visualiza en ellos, ya sea que en el video se esté mostrando que existe congestión en determinada vía, accidentes o faltas a señalamientos y además que se puedan organizar de acuerdo al tipo de vía o intersección que se muestra. Sin embargo, para la realización del sistema se integrará una tercera tabla que contenga datos de los usuarios que deberán acceder, para crear un inicio de sesión al sistema.

La base de datos se creó directamente en el servidor, para ello se accedió a la terminal y se ejecutó el comando `cqlsh` para acceder al gestor Cassandra. Una vez dentro de Cassandra, para crear la base de datos se escribió la instrucción `CREATE KEYSPACE estudiovial WITH replication = {'class' : 'SimpleStrategy', 'replication_factor': '3'}`, (Figura 5).

El nombre de la estrategia que se usa es Simple Strategy (se usa para un único centro de datos, coloca la primera réplica en un nodo determinado por el particionador) ya que se tiene un único servidor, el factor de replicación hace referencia al número de réplicas que se van a realizar sobre cada servidor. Para asegurarse de que no haya fallos, se coloca 3 en el factor de replicación, esto significa que se configura el esquema de la base de datos para crear tres copias de los datos en sus tres nodos. Cada vez que la aplicación se conecta con un nodo e introduce datos en una tabla, éstos se replican automáticamente en los otros dos nodos. Esto le proporcionará más fiabilidad para almacenar datos de forma segura.



```
luis@dell-emc:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.9 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> CREATE KEYSPACE estudiovial WITH replication = {'class' : 'SimpleStrategy', 'replication_factor': 3};
cqlsh> DESCRIBE KEYSPACE estudiovial;

CREATE KEYSPACE estudiovial WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '3'} AND durable_writes = true;
cqlsh>
```

Figura 5. Creacion de la base de datos “estudiovial”.

```
luis@dell-enc:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.9 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> CREATE KEYSPACE estudiovial WITH replication = {'class' : 'SimpleStrategy','replication_factor': 3};
cqlsh> DESCRIBE KEYSPACE estudiovial;

CREATE KEYSPACE estudiovial WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '3'} AND durable_writes = true;
cqlsh> USE estudiovial;
```

Figura 6. Instrucción para usar y manipular base de datos “estudiovial”.

Una tabla contiene la definición de una familia de columnas. La instrucción CQL para la creación de una tabla en Cassandra es muy similar a sql. CREATE TABLE table_name (column1_name datatype PRIMARY KEY, column2_name datatype, column3_name datatype.);

Las tablas que conforman nuestra base de datos quedaron de la siguiente manera (Figura 7):

```
CREATE TABLE imagenes (id uuid, viabilidad varchar, longitud varchar, latitud varchar, fecha date, hora time, comportamiento text, congestionamiento text, PRIMARY KEY(id));
CREATE TABLE videos (id uuid, viabilidad varchar, longitud varchar, latitud varchar, fecha date, hora time, comportamiento text, congestionamiento text, PRIMARY KEY(id));
DESCRIBE estudiovial;

CREATE KEYSPACE estudiovial WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '3'} AND durable_writes = true;

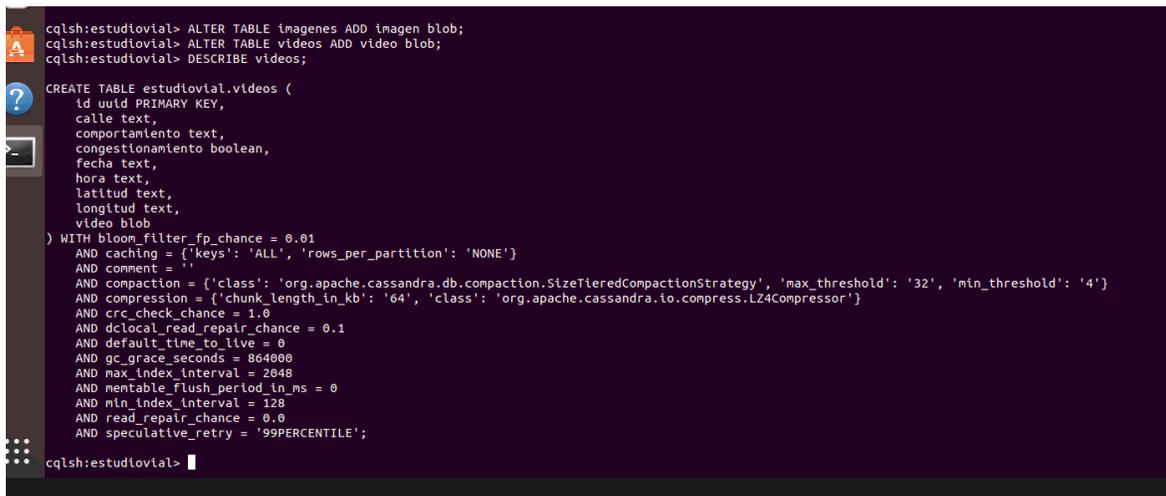
CREATE TABLE estudiovial.usuario (
  id uuid PRIMARY KEY,
  correo text,
  nombre text,
  password text
) WITH bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND crc_check_chance = 1.0
AND dlocal_read_repair_chance = 0.1
AND default_time_to_live = 0
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair_chance = 0.0
AND speculative_retry = '99PERCENTILE';

CREATE TABLE estudiovial.videos (
  id uuid PRIMARY KEY,
  comportamiento text,
  congestionamiento text,
  fecha date,
  hora time,
  latitud text,
  longitud text,
  viabilidad text
) WITH bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND crc_check_chance = 1.0
AND dlocal_read_repair_chance = 0.1
AND default_time_to_live = 0
```

Figura 7. Creación de las tablas usuario, imágenes y videos en la base de datos.

Una vez creadas todas las tablas que conforman la base de datos, se observó que hacía falta un elemento importante a utilizar dentro de las tablas imágenes y videos, para ello se

recurrió a la instrucción ALTER TABLE para hacer la inserción del dato faltante (Figura 8).



```
cqlsh:estudiovial> ALTER TABLE imagenes ADD imagen blob;
cqlsh:estudiovial> ALTER TABLE videos ADD video blob;
cqlsh:estudiovial> DESCRIBE videos;

CREATE TABLE estudiovial.videos (
  id uuid PRIMARY KEY,
  calle text,
  comportamiento text,
  congestionamiento boolean,
  fecha text,
  hora text,
  latitud text,
  longitud text,
  video blob
) WITH bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND crc_check_chance = 1.0
AND dclocal_read_repair_chance = 0.1
AND default_time_to_live = 0
AND gc_grace_seconds = 864000
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair_chance = 0.0
AND speculative_retry = '99PERCENTILE';

cqlsh:estudiovial>
```

Figura 8. Inserción de nuevo dato de tipo blob dentro de tablas imágenes y videos.

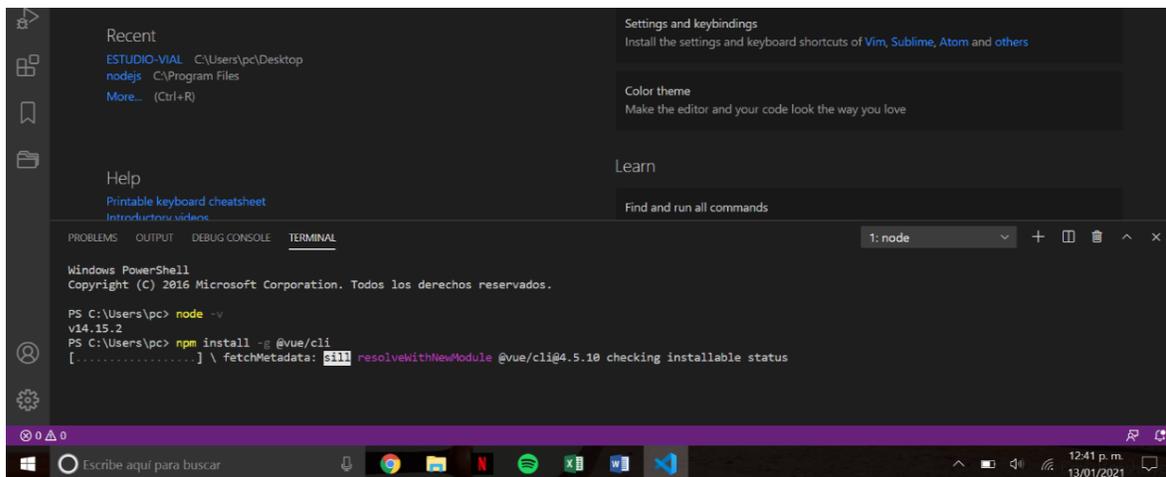


Figura 9. Instalación global de Vue Cli 4.

Para comenzar con la creación del proyecto, una vez teniendo instalado Vue, se creó una carpeta y dentro de la carpeta se ejecutó el comando “vue ui” para la instalación y configuración del proyecto de manera gráfica. En las Figuras 10, 11 y 12 se observan las configuraciones que se le dieron al proyecto.

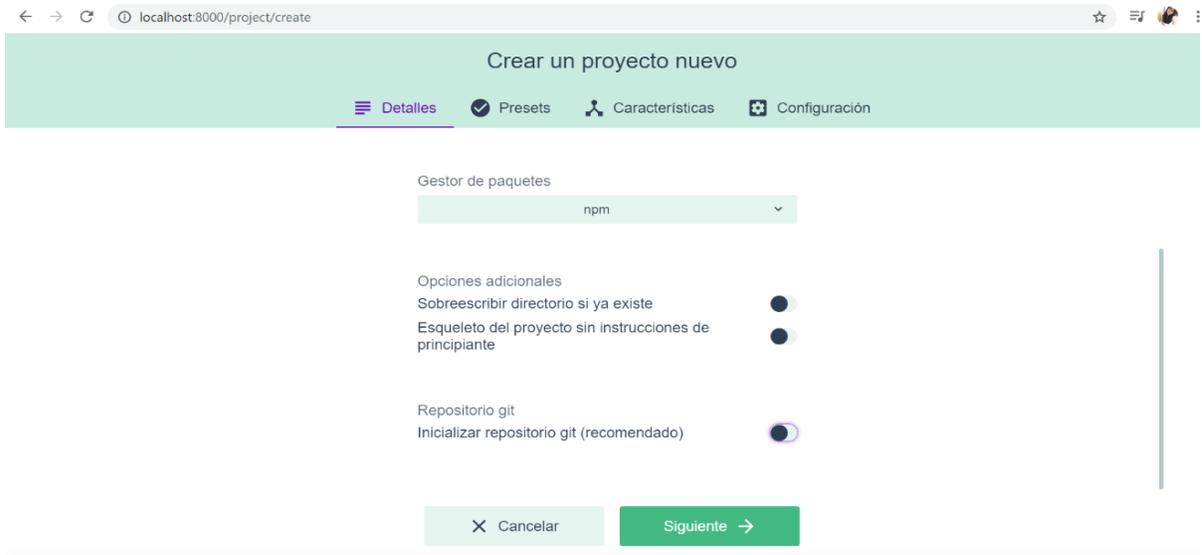


Figura 10. Selección del gestor de paquetes “npm” para el proyecto.

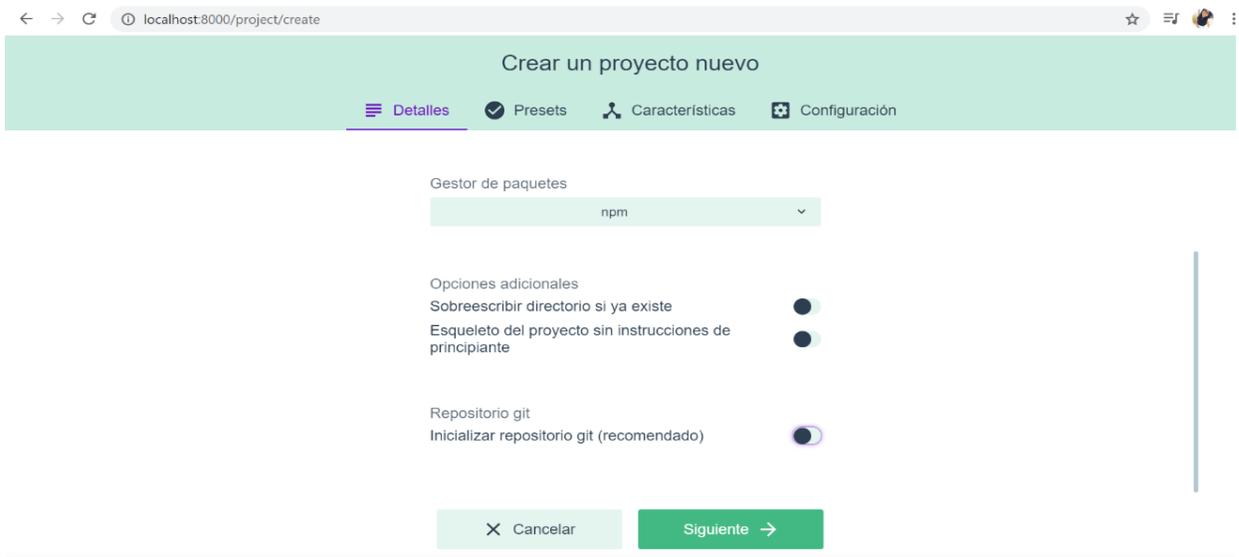


Figura 11. Selección de preset "manual" para la elección libre de características.

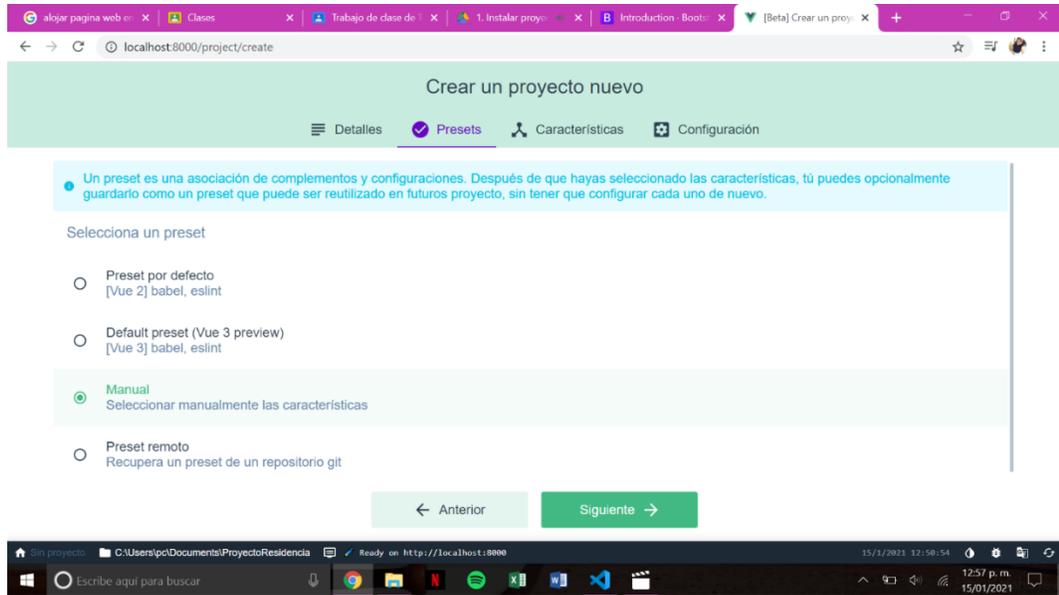


Figura 12. Selección de características que conformaran el proyecto.

Una vez que se terminó de configurar el proyecto, se procedió con la creación del sistema web. el proyecto quedo estructurado de la siguiente forma (Figura 13).

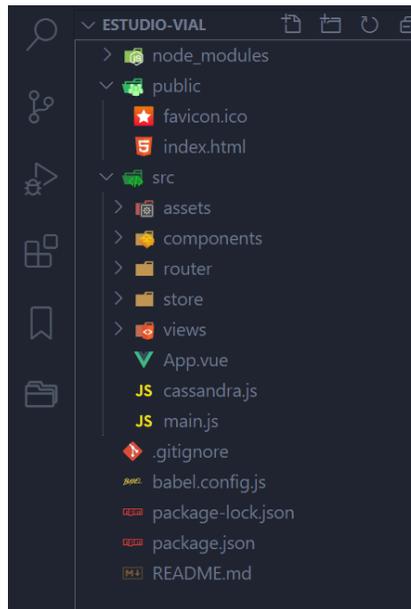
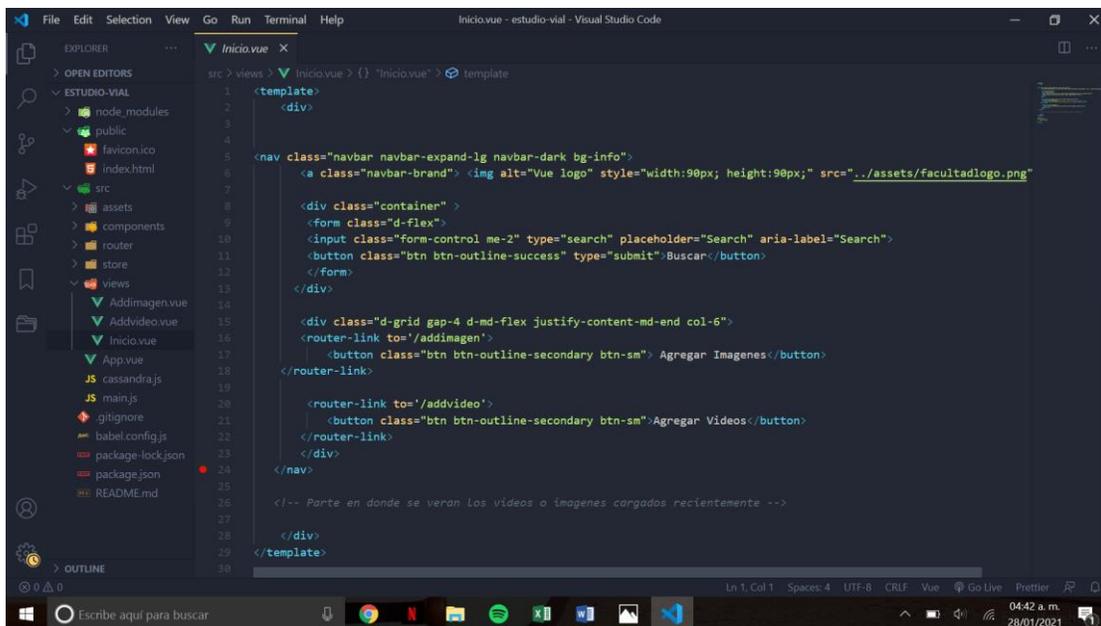


Figura 13. Estructura del proyecto del sistema web " estudio-vial".

Como se trabajó bajo el patrón MVC, lo primero que se realizaron fueron las vistas, para ello dentro de la carpeta views se crearon los archivos Inicio.vue, Addvideo.vue y Addimagen.vue.

Para la vista Inicio, dentro de un template se colocó un navbar, para colocar una barra en la parte superior de la página, dentro del navbar se colocó un buscador y dos botones como rutas que se redirigen a las vistas Addvideo y Addimagen, respectivamente (Figura 14).



```
1 <template>
2 <div>
3
4
5 <nav class="navbar navbar-expand-lg navbar-dark bg-info">
6   <a class="navbar-brand"> 
9     <form class="d-flex">
10      <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
11      <button class="btn btn-outline-success" type="submit">Buscar</button>
12    </form>
13  </div>
14
15  <div class="d-grid gap-4 d-md-flex justify-content-md-end col-6">
16    <router-link to="/addimagen">
17      <button class="btn btn-outline-secondary btn-sm"> Agregar Imagenes</button>
18    </router-link>
19
20    <router-link to="/addvideo">
21      <button class="btn btn-outline-secondary btn-sm"> Agregar Videos</button>
22    </router-link>
23  </div>
24 </nav>
25
26 <!-- Parte en donde se veran Los videos o imagenes cargados recientemente -->
27
28 </div>
29 </template>
```

Figura 14. Codificación para la vista "Inicio" del sistema web.

Para crear la vista “añadir imágenes”, se colocó un navbar para darle el mismo estilo que la vista de inicio. Bajo el navbar se creó un formulario para la inserción de los metadatos que serían insertados junto a la imagen, para la caracterización de cada archivo almacenados dentro de la base. Los datos que se manejaron fueron, vialidad, longitud, latitud, comportamiento, estos datos con formato de tipo texto, la fecha con formato de tipo date, la hora con formato de tipo time, congestionamiento con formato de tipo radio para elegir entre las opciones “sí o no” y por último el apartado subir imagen con formato de tipo archivo. Dentro de la Figura 15 y 16 se muestran más específicamente los códigos manejados para la creación de dichos formularios.

```

1 <template>
2 <div>
3
4 <nav class="navbar navbar-expand-lg navbar-dark bg-info">
5 <a class="navbar-brand"> 
7 </a>
8
9 </nav>
10
11 <form>
12 <div class="form-row">
13 <div class="form-group col-md-6">
14 <label for="inputApellidos">Vialidad</label>
15 <input type="text" class="form-control">
16
17 <label for="inputNombre">Longitud</label>
18 <input type="text" class="form-control width 280px">
19 </div>
20 <div class="form-group col-md-6">
21 <label for="inputApellidos">Latitud</label>
22 <input type="text" class="form-control">
23 </div>
24 <div class="form-group col-md-6">
25 <label for="inputApellidos">Fecha</label>
26 <input type="date" id="start" name="trip-start" class="form-control">
27 </div>
28 <div class="form-group col-md-6">
29 <label for="inputApellidos">Hora</label>
30 <input type="time" class="form-control">

```

Figura 15. Código para el desarrollo del formulario "Añadir Imagen". Primer parte.

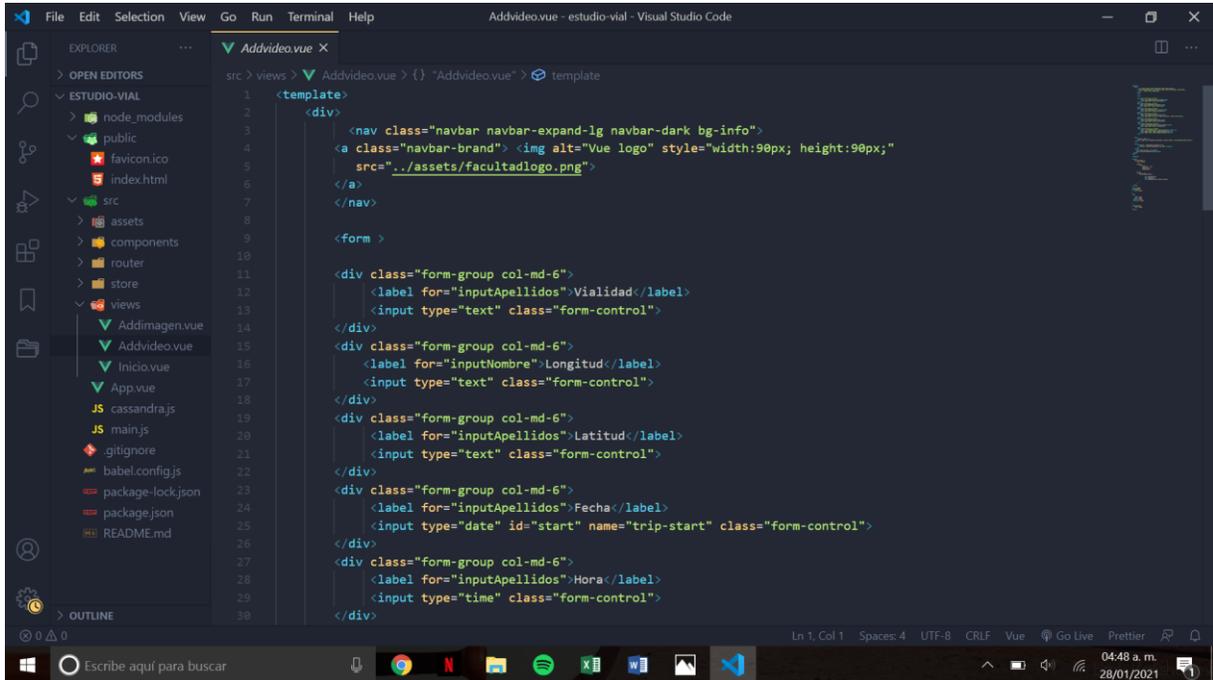
```

31 </div>
32 <div class="form-group col-md-6">
33 <label for="inputApellidos">Comportamiento</label>
34 <input type="text" class="form-control">
35 </div>
36 <div class="form-group col-md-6">
37 <label for="inputApellidos">Congestionamiento</label> <br>
38 <input type="radio" name="congestionamiento">Sí <br>
39 <input type="radio" name="congestionamiento">No <br>
40 </div>
41
42 <div>
43 <p>Subir imagen:</p>
44 <input type="file" @change="previewImage" accept="image/*">
45 </div>
46 <div>
47 <p>Progress: {{uploadValue.toFixed()+"%"}}</p>
48 <progress :value="uploadValue" max="100"></progress>
49 </div>
50 <div>
51 
52 <button @click="onUpload">Subir</button>
53 </div>
54 </form>
55 </div>
56
57 </template>
58
59

```

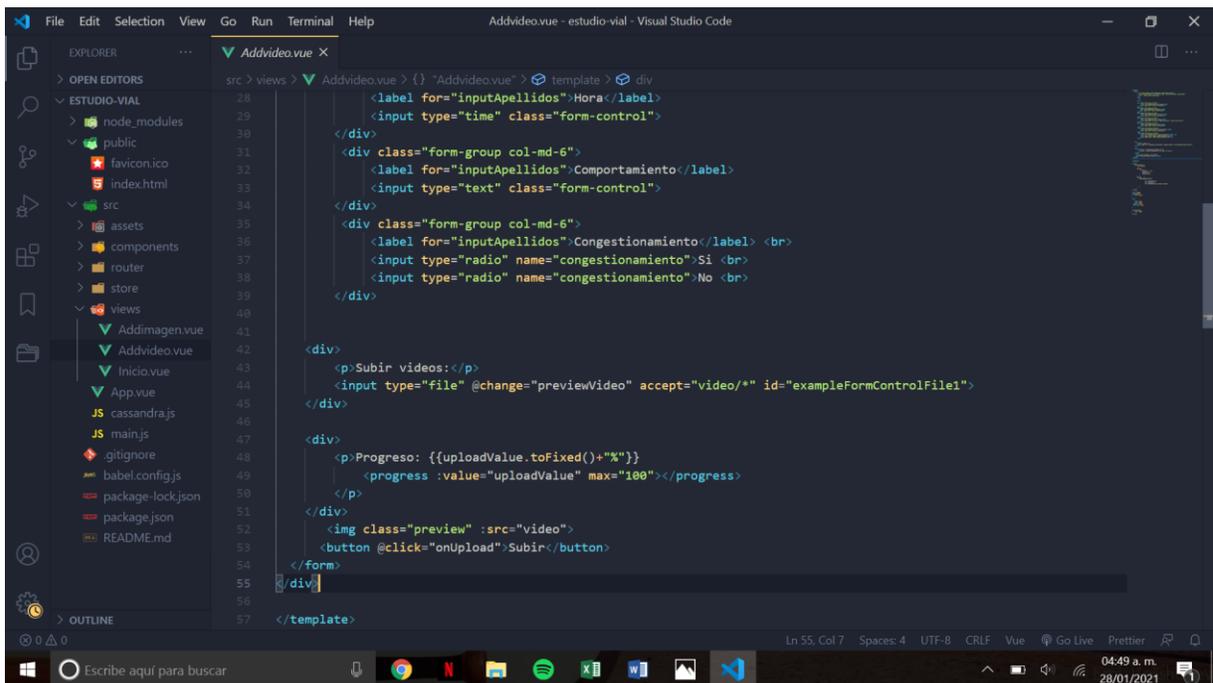
Figura 16. Código para el desarrollo del formulario "Añadir Imagen". Segunda parte.

Para la creación del formulario, para agregar videos a la base, se creó una vista aparte, sin embargo, como los datos a ingresar son los mismos, se reutilizó el código anterior, lo único que cambió fue el tipo de accept para que, a la hora de seleccionar el archivo a cargar, permita solo la carga de formatos de video (Figura 17 y 18).



```
1 <template>
2   <div>
3     <nav class="navbar navbar-expand-lg navbar-dark bg-info">
4       <a class="navbar-brand"> 
6     </a>
7   </nav>
8
9   <form >
10
11   <div class="form-group col-md-6">
12     <label for="inputApellidos">Vialidad</label>
13     <input type="text" class="form-control">
14   </div>
15   <div class="form-group col-md-6">
16     <label for="inputNombre">Longitud</label>
17     <input type="text" class="form-control">
18   </div>
19   <div class="form-group col-md-6">
20     <label for="inputApellidos">Latitud</label>
21     <input type="text" class="form-control">
22   </div>
23   <div class="form-group col-md-6">
24     <label for="inputApellidos">Fecha</label>
25     <input type="date" id="start" name="trip-start" class="form-control">
26   </div>
27   <div class="form-group col-md-6">
28     <label for="inputApellidos">Hora</label>
29     <input type="time" class="form-control">
30   </div>
31 </div>
```

Figura 17. Codificación del formulario "Añadir Videos". Primera parte.



```
28   <label for="inputApellidos">Hora</label>
29   <input type="time" class="form-control">
30 </div>
31
32   <div class="form-group col-md-6">
33     <label for="inputApellidos">Comportamiento</label>
34     <input type="text" class="form-control">
35   </div>
36   <div class="form-group col-md-6">
37     <label for="inputApellidos">Congestionamiento</label> <br>
38     <input type="radio" name="congestionamiento">Si <br>
39     <input type="radio" name="congestionamiento">No <br>
40   </div>
41
42   <div>
43     <p>Subir videos:</p>
44     <input type="file" @change="previewVideo" accept="video/*" id="exampleFormControlFile1">
45   </div>
46
47   <div>
48     <p>Progreso: {{uploadValue.toFixed()+"%"}}</p>
49     <progress :value="uploadValue" max="100"></progress>
50   </div>
51
52   
53   <button @click="onUpload">Subir</button>
54 </form>
55 </div>
56
57 </template>
```

Figura 18. Codificación del formulario "Añadir videos". Segunda parte.

8 RESULTADOS Y DISCUSIÓN

8.1 Base de datos no estructurada

Como resultado se obtuvo la base de datos no estructurada con 3 tablas, una para el registro de usuario para el inicio de sesión a la página, y las otras dos para el almacenamiento de los videos e imágenes, para comprobar el funcionamiento correcto de la base de datos, se hicieron algunas inserciones.

Para insertar datos a las tablas se utilizó la instrucción INSERT INTO de la siguiente manera: Cqlsh: estudiovia1> INSERT INTO imágenes (id, archivo, comportamiento, congestionamiento, fecha, hora, latitud, longitud, vialidad) VALUES (123e34a2-34eb-12d3-123a-426577721200, bigintAsBlob (3), 'accidente leve', 'si', '23-01-21', '12:30:23', '41°,24,12.2E', '12°,24,15N', 'publica'); y es así como estos datos fueron insertados en la tabla imágenes (Figura 19).

```
luis@dell-enc:~$ cqlsh
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.9 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use estudiovia1;
cqlsh:estudiovia1> SELECT *FROM imagenes;

id | archivo | comportamiento | congestionamiento | fecha | hora | latitud | longitud | vialidad
-----|-----|-----|-----|-----|-----|-----|-----|-----
123e34a2-34eb-12d3-123a-426577721200 | 0x0000000000000003 | accidente leve | si | 0023-01-21 | 12:30:23.000000000 | 41°,24,12.2E | 12°.24,15N | publica
(1 rows)
cqlsh:estudiovia1>
```

Figura 19. Tabla "imágenes" con datos insertados.

Como la tabla videos y la tabla imágenes tiene la misma estructura, el llenado de los datos fue el mismo, solo se cambiaron los valores (Figura 20).

```

cqlsh:estudiovial> INSERT INTO videos (id, archivo, comportamiento, congestionamiento, fecha, hora, latitud, longitud, vialidad) VALUES (123e34a3-45eb-13d3-123a-426577721200, bigintAsBlob (3), 'conduccion acelerada', 'si', '21-01-15', '10:20:43', '41°,24",12.2E', '12°,24",15N', 'primaria');
cqlsh:estudiovial> SELECT *FROM videos;

id | archivo | comportamiento | congestionamiento | fecha | hora | latitud | longitud | vialidad |
-----+-----+-----+-----+-----+-----+-----+-----+-----+
123e34a3-45eb-13d3-123a-426577721200 | 0x0000000000000003 | conduccion acelerada |  | st | 0021-01-15 | 10:20:43.000000000 | 41°,24",12.2E | 12°,24",15N | primaria
(1 rows)
cqlsh:estudiovial>

```

Figura 20. Tabla "videos" con datos insertados.

Para la tabla usuarios como también se realizó una inserción, se utilizó la misma instrucción INSERT INTO, pero esta tabla solo pide el registro de nombre, correo y contraseña (Figura 21).

```

cqlsh:estudiovial> INSERT INTO usuario (id, nombre, correo, password) VALUES (343e34a2-34eb-11d3-113a-234577721255, 'Antonio Sanchez', 'asanchez@unmsh.com', '12345678');
cqlsh:estudiovial> SELECT *FROM usuario;

id | correo | nombre | password |
-----+-----+-----+-----+
343e34a2-34eb-11d3-113a-234577721255 | asanchez@unmsh.com | Antonio Sanchez | 12345678
(1 rows)
cqlsh:estudiovial>

```

Figura 21. Tabla "usuario" con datos insertados.

8.2 Diseño de página web

El diseño de la pagina web para la manipulación de la base de datos, quedó de la siguiente manera:

A la hora de acceder a la página, se tiene la vista principal, que es donde se podrán visualizar los videos e imágenes recientemente añadidos. En la parte superior de ésta se tiene un buscador para posteriores búsquedas de los videos e imágenes que se encuentran dentro de la base de datos y a un lado del buscador se localizan dos botones, cada uno se dirige a una nueva vista que contienen ambos un formulario para subir y registrar los archivos (imágenes y videos) y su información en la base de datos (Figura 22).

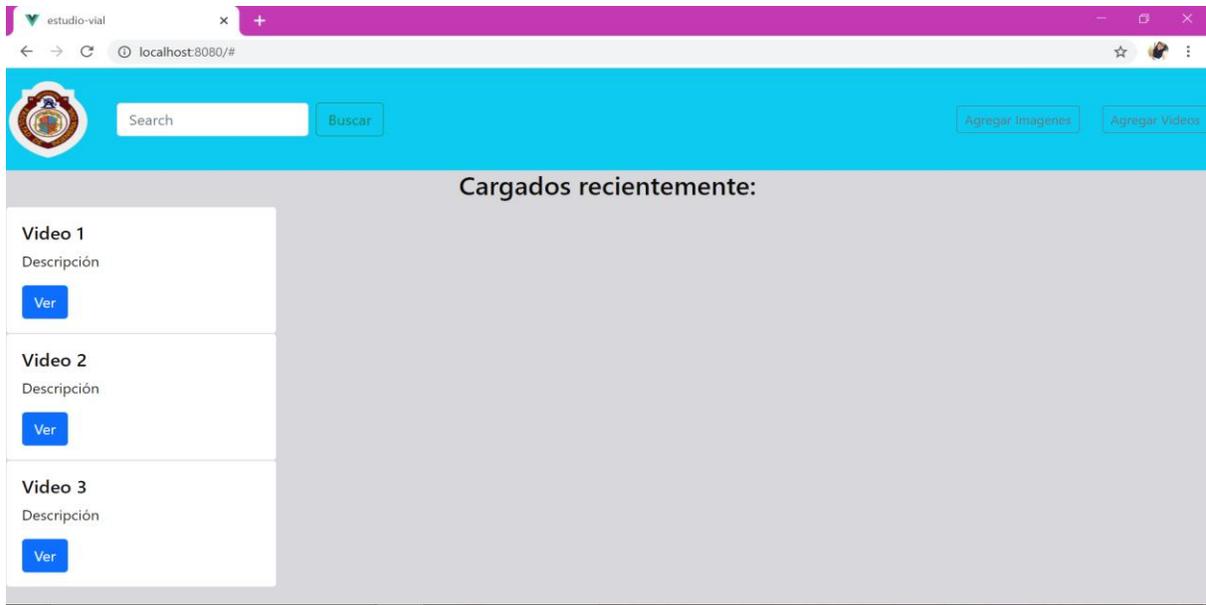
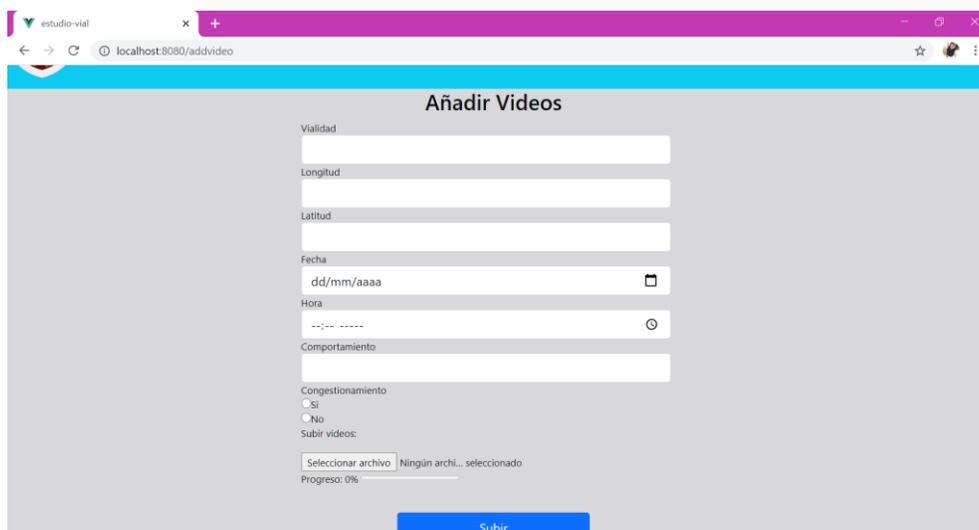


Figura 22. Vista principal de la página web

Quando desde la vista principal se da clic al botón “añadir imagen”, se redirige al sitio donde se encuentra el formulario para subir cada imagen, como se puede observar en la imagen (Figura 24) los datos que pide el formulario son el tipo de vialidad, la longitud y latitud, fecha, hora, comportamiento y congestionamiento, todos estos datos se llenan conforme lo que se muestra en la imagen a subir. Y por último pide subir la imagen.

Figura 23. Sitio de la página web para subir las imágenes.

Para subir los videos también se da clic al botón “Añadir videos” que se encuentra en la página de inicio, y como los datos que se le asignaran a cada video son los mismos para las imágenes, los datos que pide el formulario son los mismos (figura 24).



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/addvideo'. The page title is 'Añadir Videos'. The form contains the following fields and controls:

- Vialidad: Text input field.
- Longitud: Text input field.
- Latitud: Text input field.
- Fecha: Text input field with a date picker icon, containing the placeholder 'dd/mm/aaaa'.
- Hora: Text input field with a time picker icon, containing the placeholder '---:--:-----'.
- Comportamiento: Text input field.
- Congestionamiento: Radio buttons for 'Si' and 'No'.
- Subir videos: A file selection button labeled 'Seleccionar archivo' with the text 'Ningún archi... seleccionado' and a progress indicator 'Progreso: 0%'.
- Subir: A blue button at the bottom of the form.

Figura 24. Sitio de la página web para subir los videos.

Estos formularios permiten asignar los datos que se están pidiendo, a los videos o imágenes al momento de almacenarse en la base de datos, como metacaracteres para que por medio de estos puedan ser posteriormente buscados dentro de la base de datos por medio del buscador que se encuentra en la vista principal. Además estos mismos datos serán visibles como descripción al pie de cada video o imagen que se muestren en el área de “cargados recientemente”.

9 CONCLUSIONES

La realización de este proyecto para el área de maestría en infraestructura del transporte en la rama de las vías terrestres que conforma a la universidad Michoacana de San Nicolás de Hidalgo es de gran aporte para la plataforma de observación vial que ahí se maneja ya que además de ayudar a la realización de los estudios que se hacen para obtener un análisis más efectivo de las principales causas y factores que provocan un accidente vial, le darán a la información un amplio resguardo.

Se generó una base de datos no estructurada que permite el almacenamiento de imágenes y videos obtenidos de cámaras de vigilancia vial y el desarrollo de un sitio web como interfaz gráfica para la gestión de dicha base. Con ello se reforzaron los conocimientos de Ingeniería de Software y Desarrollo de Aplicaciones Web para la construcción de sitios de consulta de información para la toma de decisiones y búsqueda de datos referentes a accidentes viales.

10 RECOMENDACIONES

- Al utilizar el buscador, es recomendable ser específico e introducir claramente los conceptos de búsqueda para que pueda arrojar los resultados correctos.
- Es preciso recomendar la mejora de la interfaz gráfica para que esta sea responsiva y pueda ser usada a través de un dispositivo móvil como de una computadora.
- Añadir nuevas funciones al sitio web que contribuyan al análisis de los videos e imágenes.

11 REFERENCIAS BIBLIOGRÁFICAS

- Aggarwal, S. (2017). Comenzando Con Cassandra: Usando CQL API y CQLSH. <https://code.tutsplus.com/es/articles/getting-started-with-cassandra-using-cql-api-and-cqlsh--cms-28026>
- BrandMedia. (2019). *Diferencias entre página web estática y dinámica: ¿Cuál es mejor?* <https://brandmedia.es/diferencias-pagina-web-estatica-dinamica-mejor/>
- Cabana, J. (2017). ¿Qué es Node.js y para qué sirve? | Drauta. <https://www.drauta.com/que-es-nodejs-y-para-que-sirve>
- Cardenas, R. (2019). *¿Que necesito? ¿Web Apps, App Nativa o App Híbrida?* - GSoft. <https://www.gsoft.es/articulos/que-necesito-web-apps-app-nativa-o-app-hibrida/>
- Critical solutions. (2018, 12 noviembre). Software de Gestión de Video (VMS) - Critical Solutions - Videovigilancia. <http://criticalsolutions.mx/software-de-gestion-de-video-vms>
- Definición de Aplicación - Significado y definición de Aplicación.* (2016). <https://sistemas.com/aplicacion.php>
- EALDE Business School. (2020, 4 agosto). La metodología ágil Kanban: en qué consiste y para qué sirve. <https://www.ealde.es/metodologia-agil-kanban/>
- Fernández, Ó. (2021). *Apache Cassandra: Introducción | Aprender BIG DATA.* <https://aprenderbigdata.com/introduccion-apache-cassandra/>
- Foundation, T. A. S. (2016). *Apache Cassandra (4.0-beta4).* <https://cassandra.apache.org/>
- Glajumedia. (2020). *TOP FRAMEWORKS DE DESARROLLO WEB EN EL 2020.* <https://www.glajumedia.com/top-frameworks-de-desarrollo-web-en-el-2020/>
- Gutiérrez, J. J. (2020). ¿Qué es un framework web? http://www.lsi.us.es/~javierj/investigacion_ficheos/Framework.pdf
- Humanes, M. (2019, 23 julio). Metodología Lean: ¿qué es y cómo aplicarla en tu empresa? Ekon. <https://www.ekon.es/metodologia-lean-empresa/>
- IONOS. (2021). *Apache Cassandra: base de datos NoSQL para grandes volúmenes - IONOS.* <https://www.ionos.mx/digitalguide/hosting/cuestiones-tecnicas/apache-cassandra/>
- Margaret Rouse. (2017). *¿Qué es CouchDB?* <https://searchdatacenter.techtarget.com/es/definicion/CouchDB>
- Martí, S. (2020). Los mejores servicios de almacenamiento en la nube de 2020. Retrieved 13 January 2021, from <https://www.pcworld.es/mejores-productos/almacenamiento/servicios-almacenamiento-nube-3673539/>
- Martínez, C. C. (2010). *Fundamentos y evolución de la multimedia » Las aplicaciones web y las*

- bases de datos*. <http://multimedia.uoc.edu/blogs/fem/es/las-aplicaciones-web-y-las-bases-de-datos/>
- OpenWebinars. (2020). Qué es Apache Cassandra. <https://openwebinars.net/blog/que-es-apache-cassandra/>
- Pérez, A. (2017). *¿Cuáles son los tipos de aplicaciones móviles?* <https://cuatroochenta.com/cuales-son-los-tipos-de-aplicaciones/>
- Quonext. (2020). Metodologías ágiles | Scrum, Kanban y XP. Blog Quonext. <https://www.quonext.com/blog/metodologias-agiles-scrum-kanban-xp/>
- Rubenfa. (2014). *MongoDB*. <https://www.genbeta.com/desarrollo/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no>
- Sanfilippo, S. (2015). *Redis*. Redis Labs. <https://redis.io/>
- Santander Universidades. (2020, 21 diciembre). Metodologías de desarrollo de software: ¿qué son? [blog.becas-santander. https://blog.becas-santander.com/es/metodologias-desarrollo-software.html](https://blog.becas-santander.com/es/metodologias-desarrollo-software.html)