



“SISTEMA PARA EL MONITOREO, DETECCIÓN Y ALERTA DE
SOMNOLENCIA DEL CONDUCTOR MEDIANTE VISIÓN ARTIFICIAL,
COMUNICACIÓN INALÁMBRICA Y GEOLOCALIZACIÓN”

Tercer Reporte Parcial

Lista de actividades

- Búsqueda de manejadores de bases de datos
- Elección del manejador de bases de datos
- Modelado de la base de datos
- Análisis del Módulo de Telemetría
- Diseño del Módulo de Telemetría
- Investigación de Servidores de alojamiento
- Elección del Servidor de Alojamiento
- Análisis del Módulo de Estación Base
- Diseño del Módulo de Estación Base
- Análisis de la aplicación Web
- Búsqueda Lenguajes de programación web
- Elección del lenguaje de programación para la aplicación web
- Diseño de la aplicación web
- Cotización de materiales

Autores:

Alan Eduardo Gamboa Del
Ángel
Maite Paulette Díaz Martínez
Nilsen Alfaro Parra

Asesores:

M.en C. Niels Henrik Navarrete
Manzanilla
Dr. Rodolfo Vera Amaro

Índice

1. Búsqueda de manejadores de bases de datos	4
2. Elección del manejador de bases de datos	10
3. Modelado de la base de datos	11
4. Análisis del Módulo de Telemetría	13
5. Diseño del Módulo de Telemetría	14
6. Investigación de Servidores de alojamiento	15
7. Elección del Servidor de Alojamiento	16
8. Análisis del Módulo de Estación Base	17
9. Diseño del Módulo de Estación Base	22
10. Búsqueda de servidores de alojamiento	24
11. Análisis de la aplicación Web	28
12. Búsqueda de Lenguajes de Programación Web	34
13. Elección del lenguaje de programación para la aplicación web	36
14. Diseño de la aplicación web	37
15. Cotización de materiales	38
16. Conclusiones	39
17. Bibliografía	40

Índice de figuras

1.	Icono de MongoDB	10
2.	Estructura de un Documento en MongoDB	11
3.	Modelo propuesto para la base de datos en MongoDB	11
4.	Documento de Registro de Conductores	12
5.	Documento de registro de Incidencias	12
6.	Diagrama de Casos de Usos del Módulo de Estación Base	20
7.	Diagrama de Actividades del Módulo de Estación Base	21
8.	Diagrama de Comunicación para guardar video	22
9.	Diagrama de Comunicación para guardar incidencia	22
10.	Diagrama de Comunicación para conectar aplicación web	22
11.	Diagrama de Comunicación para desplegar aplicación web	22
12.	Diseño del Módulo de Estación Base	23
13.	Tipos de Planes de AWS Amplify	25
14.	Tiempos de Respuesta	26

Índice de tablas

1.	Sistemas de Gestión de Bases de Datos Relacionales	7
2.	Sistemas de Gestión de Bases de Datos No Relacionales	9
3.	RF01- Guardar Incidencia	17
4.	RF02- Guardar Video	18
5.	RF03- Conectar Aplicación Web	18
6.	RF04- Desplegar Aplicación Web	19
7.	RF05- Guardar Credenciales de Usuario Administrador	19
8.	Requerimientos No Funcionales del Módulo de Estación Base	20
9.	Tipos de Contenido Multimedia	24
10.	RF01- Iniciar Sesión	28
11.	RF02- Mostrar el Historial de Reportes de Incidencia	28
12.	RF02- Mostrar el Historial de Reportes de Incidencia	29
13.	RF02- Mostrar el Historial de Reportes de Incidencia	29
14.	RF02- Mostrar el Historial de Reportes de Incidencia	30
15.	RF02- Mostrar el Historial de Reportes de Incidencia	30
16.	RF02- Mostrar el Historial de Reportes de Incidencia	31
17.	RF02- Mostrar el Historial de Reportes de Incidencia	31
18.	RF02- Mostrar el Historial de Reportes de Incidencia	31
19.	RF02- Mostrar el Historial de Reportes de Incidencia	32
20.	RF02- Mostrar el Historial de Reportes de Incidencia	32
21.	RF02- Mostrar el Historial de Reportes de Incidencia	33

1. Búsqueda de manejadores de bases de datos

Un Sistema Gestor de Base de Datos (SGBD) o *DataBase Managment System* (DBMS) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de información del modo más eficiente posible. [1]

Los sistemas gestores de bases de datos son clasificados según la forma en que administran los datos en:

- Relacionales (SQL)
- No relacionales (NoSQL)

Sistemas Gestores de bases de datos Relacionales (SQL)

SQL es un lenguaje estándar de consultas o *queries* a bases de datos. La programación o consultas en SQL sirve para almacenar, manipular y recuperar datos de bases de datos relacionales. [2]

La información que almacena la BBDD (base de datos) está relacionada entre sí. Los datos relacionados (registros o filas) son almacenados en tablas que constan de varios campos (columnas).

Características:

- Integridad de los datos
Asegura que los datos sean válidos, correctos y completos. SQL y las bases de datos relacionales tienen la función de preservar esta integridad.
- Lenguaje estandarizado
SQL es un lenguaje estandarizado, esto quiere decir que se pueden desplegar implementaciones del mismo lenguaje SQL en diferentes sistemas. De esta manera se puede utilizar el mismo código para todos ellos.
- Sencillez y claridad
SQL es un lenguaje integral desde el punto de vista conceptual, esto quiere decir que SQL es un lenguaje unificado, claro y simple, de fácil comprensión.
- Flexibilidad
Una de las razones por las que SQL es un lenguaje tan utilizado es por su flexibilidad, dada su versatilidad a la hora de implantar soluciones y por permitir definir diferentes formas de ver los datos para satisfacer las especificaciones requeridas por parte del usuario. [2]

Ventajas:

- Opción segura
Las tablas y datos permanecen protegidos con los nombres de usuario y sus contraseñas. Por lo tanto, solo una persona autorizada puede acceder a esas tablas y datos para trabajar.

- Fácil de usar
No existe ningún proceso de procesamiento de consultas y, por lo tanto, las bases de datos relacionales son simples. El usuario no está obligado a pasar ninguna consulta compleja para acceder a los datos.
- Rendimiento rápido
Fáciles de usar y también son simples por su estructura. Permite al usuario obtener una experiencia rápida. El usuario obtiene muchos filtros de optimización u opciones para aumentar la velocidad de acceso a los datos.
- Presenta datos con precisión
Las bases de datos contienen tablas precisas e interrelacionadas. Usa las claves primaria y externa para establecer la relación entre el archivo *raw*. Por lo tanto, los usuarios no tienen ningún problema con la reputación de los datos anteriores que ya han recibido. Además, hay cero posibilidades de duplicación de datos en bases de datos relacionales.
- Accesible en cualquier momento
El usuario no necesita ninguna ruta predefinida. Por lo tanto, la accesibilidad en el RDBMS es más fácil en comparación con otras bases de datos. Además, los usuarios también pueden modificar los datos rápidamente.
- Puede tener varios usuarios
Muchos usuarios pueden acceder a la base de datos al mismo tiempo. Incluso si los datos ya están actualizados, también muchos usuarios pueden visitar sin ningún problema. Por lo tanto, se puede evitar el bloqueo causado principalmente al obtener un inicio de sesión de múltiples usuarios en la base de datos. [3]

Desventajas:

- Incrementa el gasto
Estas bases de datos son bastante caras de usar. Además, la empresa o el usuario deben comprar el software y las herramientas para obtener el resultado completo de RDBMS. Por lo tanto, si se tiene un presupuesto bajo, puede resultar difícil obtener una base de datos de este tipo.
- Se puede perder información
El RDBMS contiene el espacio limitado y estos dispositivos de almacenamiento no pueden almacenar más datos si no hay espacio. Como no hay más almacenamiento, estos datos se pueden perder y crear problemas en el futuro.
- Requerido para el uso de dispositivos de almacenamiento físico
Requiere importantes dispositivos de almacenamiento, especialmente dispositivos de memoria física.
- Problema de rendimiento
El nivel de rendimiento puede ser bajo en las bases de datos relacionales porque depende de filas y columnas. Por lo tanto, si hay muchas filas y tablas, la consulta tardará más en procesar el resultado. Además, si hay una cantidad considerable de datos en la máquina, puede ralentizar el proceso de trabajo.

- Puede ser complejo de entender

Cuando aumenta la cantidad de datos almacenados, se vuelve complicado comprender esos datos de forma simplificada. Sucede porque todos los datos han seguido utilizando la función particular que genera grandes conjuntos de datos para filtrar. [3]

Principales sistemas de gestión de bases de datos relacionales

A continuación, se muestra una tabla comparativa de los principales sistemas de gestión de bases de datos relacionales:

Sistemas Gestores de bases de datos No Relacionales (SQL)

Se utilizan en entornos distribuidos que han de estar siempre disponibles y operativos y que gestionan un importante volumen de datos.

Para la administración de este tipo de bases de datos, actualmente los principales sistemas gestores de bases de datos (SGBD NoSQL) son los siguientes:

Características:

- Los datos no tienen que almacenarse en tablas.
- Generalmente, su arquitectura es distribuida, la información se almacena en más de una máquina del sistema. Por lo tanto, los sistemas que las soportan tienen una mayor escalabilidad horizontal (a mayor número de nodos mayor rendimiento) y también mayor tolerancia ante fallos en los distintos nodos.
- Son más eficientes en el procesamiento de los datos que las BBDD relacionales, por tal motivo, son la elección para aplicaciones que hacen un uso intensivo de estos, como por ejemplo *streaming*.
- Utilizan lo que se conoce como consistencia eventual, que consiste en que los cambios realizados en los datos serán replicados a todos los nodos del sistema, lo cual aumenta el rendimiento de estos sistemas en contraposición a las propiedades ACID de las BBDD relacionales (Atomicidad, Consistencia/Integridad, Aislamiento y Durabilidad).

Ventajas:

- Aplicaciones de big data
Grandes volúmenes de información son manejados fácilmente por las bases de datos NoSQL.
- Administración de la base de datos
Requieren menos administración práctica, cuenta con capacidades de distribución de datos y reparación automática, modelos de datos simplificados y menos requisitos de ajuste y administración.
- Versatilidad
Las posibilidades de crecimiento en el volumen de datos o la posibilidad de incluir cambios sobre la forma en la que ingresan los datos sin necesidad de alterar la estructura, lo que permite adaptarse de forma rápida a un entorno de alto dinamismo.

SGBDR	Descripción	Ventajas	Desventajas
MySQL	Es un SGBD multihilo y multiusuario utilizado en aplicaciones creadas como software libre.	-Facilidad de uso y gran rendimiento -Facilidad para instalar y configurar -Soporte multiplataforma	Escalabilidad, no trabaja de manera eficiente con bases de datos muy grandes.
MariaDB	Este SGBD es una derivación de MySQL que cuenta con la mayoría de características de este e incluye varias extensiones.	-Aumento de motores de almacenamiento -Gran escalabilidad -Seguridad y rapidez en transacciones -Extensiones y nuevas características relacionadas con su aplicación para Bases de datos NoSQL. -Compatible con MySQL.	Incompatibilidades en la migración de MariaDB y MySQL. Atrasos en la liberación de versiones estables.
PostgreSQL	Es un SGBD orientado a objetos y es libre, publicado bajo la licencia BSD.	-Flexibilidad en cuanto a lenguajes de programación. -Multiplataforma. -Herramienta fácil e intuitiva para la administración de las bases de datos. -Robustez, Eficiencia y Estabilidad.	Lentitud para la administración de bases de datos pequeñas ya que está optimizado para gestionar grandes volúmenes de datos.
Microsoft SQL Server	Es un SGBD basado en el lenguaje <i>Transact-SQL</i> capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea.	-Escalabilidad, estabilidad y seguridad. -Posibilidad de cancelar consultas. -Potente entorno gráfico de administración. -Nativo para Windows, pero puede utilizarse en otras plataformas como Linux o Docker.	Coste del software, cuenta con un plan gratuito (Express) pero lo normal es la elección de alguno de los planes de pago disponibles.
Oracle	Oracle ha sido el SGBD por excelencia para el mundo empresarial, considerado siempre como el más completo y robusto.	-Soporte de transacciones. -Estabilidad. -Escalabilidad. -Multiplataforma.	Coste del software, aunque cuenta con una versión gratuita (Express Edition o XE), sus principales opciones son de pago.

Tabla 1: Sistemas de Gestión de Bases de Datos Relacionales

- **Crecimiento Horizontal**
Son altamente escalables, si se requiere instalar mayor cantidad de nodos para ampliar la capacidad, se puede hacer sin problemas. Esto no interrumpe la usabilidad o consultas dentro de la base de datos.
- **Economía**
No se necesitan servidores con gran cantidad de recursos para operar. La adaptabilidad y flexibilidad permiten empezar con bajos niveles de inversión en equipos e ir ampliando la capacidad a medida de las necesidades.

Desventajas:

- **Menos maduro**
La mayoría de las alternativas de bases de datos NoSQL apenas han salido de las etapas de preproducción, y hay muchas características importantes que aún no se han implementado.
- **Atomicidad**
Algunas de estas bases de datos no incorporan la atomicidad de información. Esto puede derivar en que la información no sea consistente entre nodos.
- **Software poco documentado**
Al ser tan relativamente nuevo, el NoSQL puede adolecer de que algunas operaciones sean limitadas por la falta de información sobre las herramientas y sus características.
- **Baja estandarización**
No se tiene un criterio plenamente definido entre los motores que se utilizan en este tipo de base de datos. El lenguaje tiende a variar según el tipo de base de datos que se vaya a utilizar.
- **Herramientas GUI**
La mayoría de las bases de datos NoSQL no contienen una interfaz gráfica. Requiere conocimiento especial para poder ejecutar algunas de ellas.

Principales sistemas de gestión de bases de datos no relacionales

Se utilizan en entornos distribuidos que han de estar siempre disponibles y operativos y que gestionan un importante volumen de datos.

Para la administración de este tipo de bases de datos, actualmente los principales sistemas gestores de bases de datos (SGBD NoSQL) son los siguientes:

SGBDR	Descripción	Ventajas	Desventajas
MongoDB	Es un SGBD NoSQL orientado a documentos que almacena la información en estructuras JSON con un esquema dinámico que permite su facilidad de integración.	<ul style="list-style-type: none"> -Indexación y replicación. -Balanceo de carga -Almacenamiento en documentos. -Consultas ad hoc. -Escalabilidad horizontal. -Código abierto. -Multiplataforma. 	<ul style="list-style-type: none"> -MongoDB no es un SGBD adecuado para realizar transacciones complejas. -Documentos limitados a 16MB.
Redis	Redis está basado en el almacenamiento clave-valor. El principal uso de este SGBD es para el almacenamiento en memoria caché y la administración de sesiones.	<ul style="list-style-type: none"> -Atomicidad y persistencia -Gran velocidad -Simplicidad -Multiplataforma 	<ul style="list-style-type: none"> - La implementación en la nube a gran escala puede ser difícil. -No ofrece ninguna solución de conjuntos en clústeres madura.
Cassandra	Cassandra utiliza almacenamiento clave-valor. Es un SGBD NoSQL distribuido y masivamente escalable. Dispone de un lenguaje propio para las consultas denominado CQL (<i>Cassandra Query Language</i>).	<ul style="list-style-type: none"> -Multiplataforma -Propio lenguaje de consultas -Escalado lineal y horizontal -Es un SGBD distribuido -Utiliza una arquitectura punto a punto 	<ul style="list-style-type: none"> -Duplicación de datos. -Ejecución de lecturas lenta. -Consultas limitadas debido a la recuperación de datos.
DynamoDB	Base de datos NoSQL de clave-valor sin servidor y completamente administrada que está diseñada para ejecutar aplicaciones de alto rendimiento a cualquier escala.	<ul style="list-style-type: none"> -Ventajas avanzadas en la nube de AWS. -Facilidad de uso. -Coste. -Integración -Administrable. 	<ul style="list-style-type: none"> -Capacidad de consultas deficiente -Compatible solo con AWS. - No puede utilizar scripts del lado del servidor con DynamoDB.

Tabla 2: Sistemas de Gestión de Bases de Datos No Relacionales

2. Elección del manejador de bases de datos

Debido a la escalabilidad, facilidad de implementación, bajo coste y flexibilidad que ofrece los gestores de bases de datos NoSQL para adaptarse a las necesidades del proyecto, se optó por la elección de un gestor de este tipo. Además, se contemplaron las posibles modificaciones dentro del modelado de la base de datos, las cuales tendrán como motivo, cumplir los requisitos del sistema y presentar la información necesaria en la aplicación web. Estas modificaciones (solo en caso de ser necesarias) podrán ser fácilmente aplicadas usando un SGBD NoSQL ya que ofrecen una mayor facilidad de realizar cambios dentro del esquema a diferencia de un SGBD SQL.

Dentro de los SGBD NoSQL se eligió MongoDB, ya que la aplicación web se realizará utilizando como base el lenguaje JavaScript Node.js para realizar el *backend* de la aplicación, por lo cual MongoDB complementa y facilita el desarrollo del proyecto en herramientas basados en JavaScript. Además, MongoDB es de código abierto, y es una herramienta gratuita, por lo cual no hay necesidad de pagar el uso de licencias. Otra característica importante es que es una base de datos multiplataforma, esto nos permitirá registrar información desde el sistema operativo Linux, que se ocupará en la Raspberry Pi y por otro lado se ocupará Windows en la aplicación web para la inserción, modificación, lectura y eliminación de los datos en la base de datos.



Figura 1: Icono de MongoDB

3. Modelado de la base de datos

Tomando en cuenta la elección de MongoDB como gestor de la base de datos, se consultó la documentación de MongoDB para conocer la estructura y el modelado de los datos.

MongoDB utiliza documentos para registrar los datos, estos documentos son almacenados dentro de las colecciones, las cuales, en una base de datos relacional se asemejaría a una tabla. Las bases de datos en MongoDB están conformadas por una o más colecciones de documentos.

MongoDB almacena registros de datos como documentos BSON. BSON es una representación binaria de documentos JSON, aunque contiene más tipos de datos que JSON.

Un registro en MongoDB es un documento, que es una estructura de datos compuesta por pares de campo y valor. Los documentos MongoDB son similares a los objetos JSON. Los valores de los campos pueden incluir otros documentos, matrices y matrices de documentos.

```

{
  name: "sue",
  age: 26,
  status: "A",
  groups: [ "news", "sports" ]
}

```

← field: value
 ← field: value
 ← field: value
 ← field: value

Figura 2: Estructura de un Documento en MongoDB

Basado en la estructura NOSQL de MongoDB, se propone el siguiente modelado para la base de datos, la cual contará con dos colecciones: Conductores e Incidencias. Cabe mencionar que en dicho modelado no se contempla a los administradores, debido a que se utilizará el sistema de Amazon Cognito, que ofrece herramientas de autenticación, registro, verificación e inicio de sesión para el usuario.

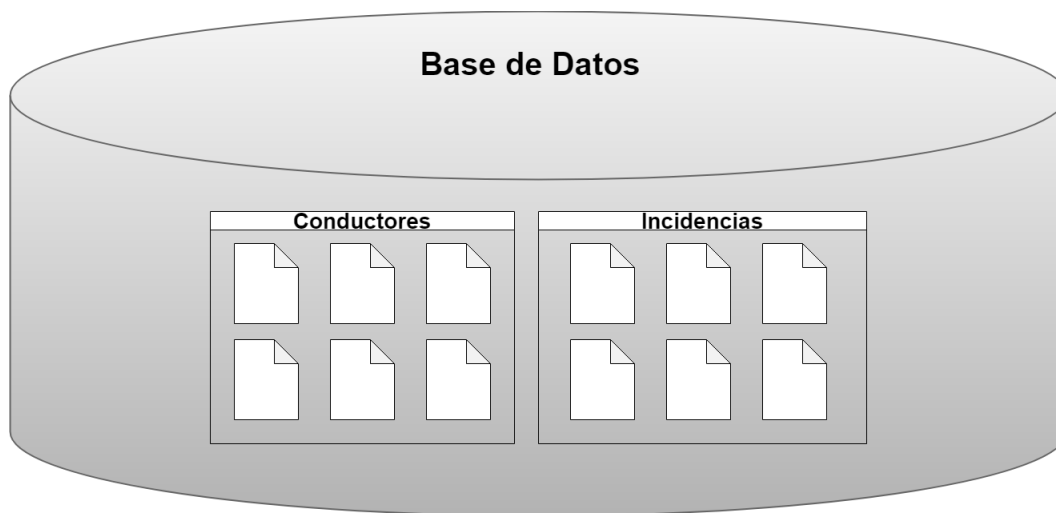


Figura 3: Modelo propuesto para la base de datos en MongoDB

El modelo de documento para la colección ?Conductores?, se usará para registrar el nombre y apellido de cada conductor, así mismo se contabilizará el número de incidencias que presenté cada

uno de ellos, el contador del conductor en marcha aumentará cada vez que el Módulo Central de Procesamiento envíe un reporte de incidencia a la base de datos. Si el Usuario Administrador del Módulo de la Estación Base revisa el video de la incidencia y lo cataloga como ?Descartar? entonces se restará la incidencia, si esta es catalogada como ?Confirmar? el contador permanecerá igual. Es importante mencionar que el registro de cada conductor se realizará desde la Aplicación Web. A continuación, se muestran los campos y un ejemplo de los valores que pueden ser ingresados dentro del documento de registro de conductores.

```
{  "_id" : 1,
  "NombreConductor" : "Nombre Conductor",
  "ApellidoConductor" : "ApellidoMaterno ApellidoPaterno",
  "NúmeroIncidencias" : 0
}
```

Figura 4: Documento de Registro de Conductores

El modelo de documento para la colección ?Incidencias?, se usará para registrar el Reporte de Incidencia que presente un conductor, el cual será enviado desde el Módulo Central de Procesamiento, por tanto, se realizará una consulta previa a la colección ?Conductores? para obtener el id, nombre y apellido del conductor en marcha. Posteriormente se realizará el reporte de la incidencia, el cual contendrá el Id del conductor, la fecha y hora de la incidencia, el nombre y apellido del conductor, y el estado de incidencia, este último muestra si la incidencia fue catalogada como descartada o confirmada por parte del Usuario Administrador después de revisar el video de la incidencia. Como se mencionó anteriormente, MongoDB permite anidar documentos, por lo que, dentro de este documento se encontrará anidado un segundo documento llamado ?Detalles?, el cual registrará la Ubicación donde se detectó la incidencia y la URL donde se encontrará almacenado el video.

```
{  "_id" : 1,
  "_idConductor" : "1",
  "NombreConductor" : "Alan",
  "ApellidoConductor" : "Gamboa del Angel",
  "FechaHora" : "2022-11-16T13:30:00.000Z",
  "Estado_Incidencia" : null,
  "Detalles": {
    "Ubicación" : [ -73.856077, 40.848447 ],
    "Video" : "URL"
  }
}
```

Figura 5: Documento de registro de Incidencias

Cabe señalar que los anteriores modelos muestran el campo de ?Id?, sin embargo, MongoDB asigna automáticamente un Id, si el id no es especificado, esto lo realiza cada vez que se crea un nuevo documento dentro de una colección, el cual es un valor hexadecimal de 12 bytes representado en 24 caracteres.

4. Análisis del Módulo de Telemetría

5. Diseño del Módulo de Telemetría

6. Investigación de Servidores de alojamiento

7. Elección del Servidor de Alojamiento

8. Análisis del Módulo de Estación Base

El módulo de la estación base tiene como objetivo que el usuario administrador, visualice y confirme el estado de los reportes de incidencia que se hayan presentado por parte del conductor, por tal motivo, se realizará una aplicación web, la cual se conectará a una base de datos NoSQL. En ella se guardarán los reportes de incidencias y se podrán visualizar por medio de la aplicación web. Cabe aclarar que el video de la incidencia se almacenará en la nube, ya que al ser contenido multimedia no se puede guardar en la base de datos, únicamente se guardará el URL para acceder al video. Las credenciales del usuario serán almacenadas en Amazon Cognito, con la cuales podrán iniciar sesión dentro de la Aplicación Web, la cual estará alojada en un servidor web. El sistema contará con los siguientes requerimientos:

Requerimientos Funcionales del Módulo de Estación Base

ID	Nombre corto del Requerimiento
RF01	<i>Guardar Incidencia</i>
Descripción: La base de datos almacenará la información de cada reporte de incidencia que se envíe desde el Módulo Central de Procesamiento.	
Elementos: <ul style="list-style-type: none">• Id del Conductor• Nombre de Conductor• Apellidos de Conductor• Número de Incidencias• Fecha• Hora• Estado de la Incidencia• Ubicación• URL del video	
Solución del Requerimiento: El Módulo de Procesamiento Central puede insertar datos en la base datos.	

Tabla 3: RF01- Guardar Incidencia

ID	Nombre corto del Requerimiento
RF02	<i>Guardar Video</i>
Descripción: La base de datos almacenará la información de cada reporte de incidencia que se envíe desde el Módulo Central de Procesamiento.	
Elementos: <ul style="list-style-type: none">• Video de incidencia	
Solución del Requerimiento: El Módulo de Procesamiento Central puede insertar el video en el almacenador de objetos en la nube, mientras que la base de datos obtiene la URL del video guardado.	

Tabla 4: RF02- Guardar Video

ID	Nombre corto del Requerimiento
RF03	<i>Conectar Aplicación Web</i>
Descripción: La base de datos almacenará la información de cada reporte de incidencia que se envíe desde el Módulo Central de Procesamiento.	
Elementos: <ul style="list-style-type: none">• Node.js• React• Express• MongoDB	
Solución del Requerimiento: El usuario administrador puede realizar la inserción, modificación, eliminación y consulta de datos desde la aplicación web.	

Tabla 5: RF03- Conectar Aplicación Web

ID	Nombre corto del Requerimiento
<i>RF04</i>	<i>Desplegar Aplicación Web</i>
Descripción: La aplicación web se alojará en una red de entrega de contenido (CDN), disponible con una URL.	
Elementos: <ul style="list-style-type: none">• AWS Amplify• GitHub• Amazon S3	
Solución del Requerimiento: El usuario administrador puede acceder a la interfaz de la aplicación web, haciendo uso de la URL en el navegador web.	

Tabla 6: RF04- Desplegar Aplicación Web

ID	Nombre corto del Requerimiento
<i>RF05</i>	<i>Guardar Credenciales de Usuario Administrador</i>
Descripción: Se guardarán únicamente las credenciales de los usuarios administradores que podrán acceder a la aplicación web.	
Elementos: <ul style="list-style-type: none">• Amazon Cognito	
Solución del Requerimiento: El usuario administrador puede iniciar sesión en la aplicación web.	

Tabla 7: RF05- Guardar Credenciales de Usuario Administrador

Requerimientos No Funcionales del Módulo de Estación Base

<i>ID</i>	<i>Nombre del requerimiento</i>	<i>Descripción</i>
<i>RNF01</i>	Disponibilidad	La disponibilidad del sistema será continua, el usuario podrá acceder a la información las 24 horas del día.
<i>RNF02</i>	Interoperabilidad	El sistema será capaz de intercambiar información con el Módulo Central de Procesamiento a través del Módulo de Telemetría.
<i>RNF03</i>	Seguridad	Las credenciales de acceso al sistema podrán ser cambiadas únicamente por el administrador.
<i>RNF04</i>	Usabilidad	El sistema estará enfocado a la visualización de reportes, por lo que el usuario le tomará menos de 1 hora en familiarizarse con el sistema. El sistema proporcionará mensajes de advertencia orientados al usuario, en caso de ocurrir un error en el Módulo Central de Procesamiento.
<i>RNF05</i>	Interfaz del sistema	La aplicación web presentará una interfaz de usuario sencilla que permitirá al usuario un fácil manejo del sistema.

Tabla 8: Requerimientos No Funcionales del Módulo de Estación Base

Con base en los requerimientos se realizó el siguiente diagrama de casos de uso, el cual muestra las actividades y la interacción con el Módulo de Estación base y el Módulo Central de Procesamiento.

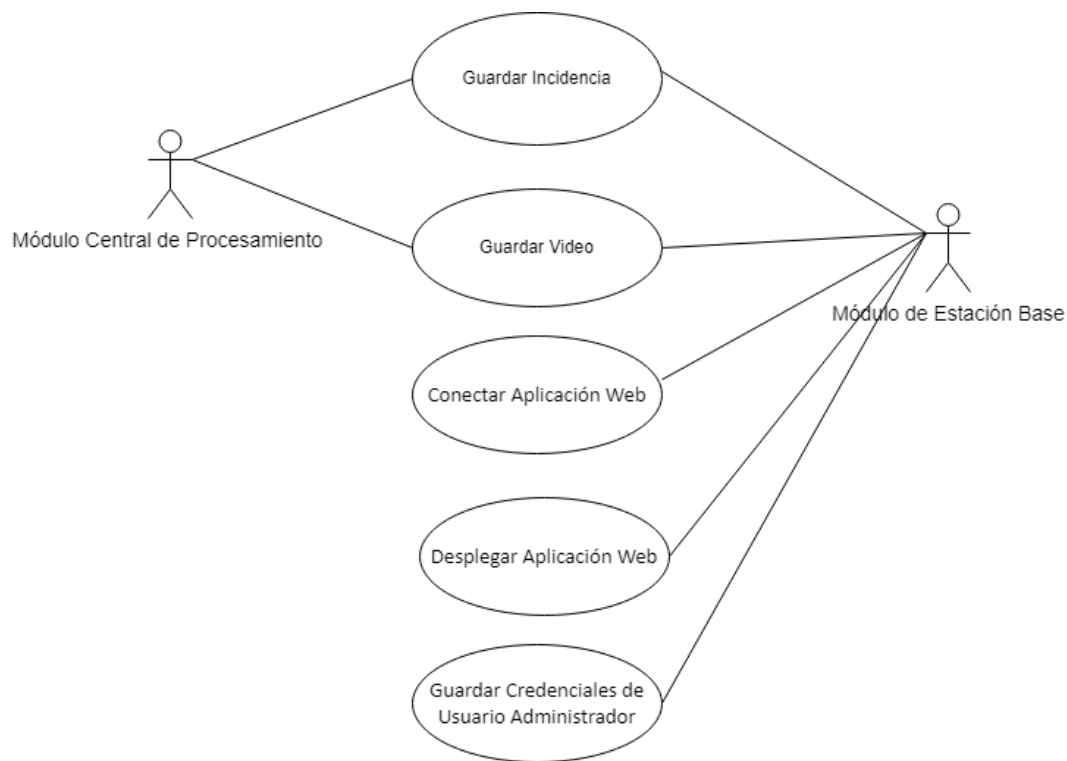


Figura 6: Diagrama de Casos de Usos del Módulo de Estación Base

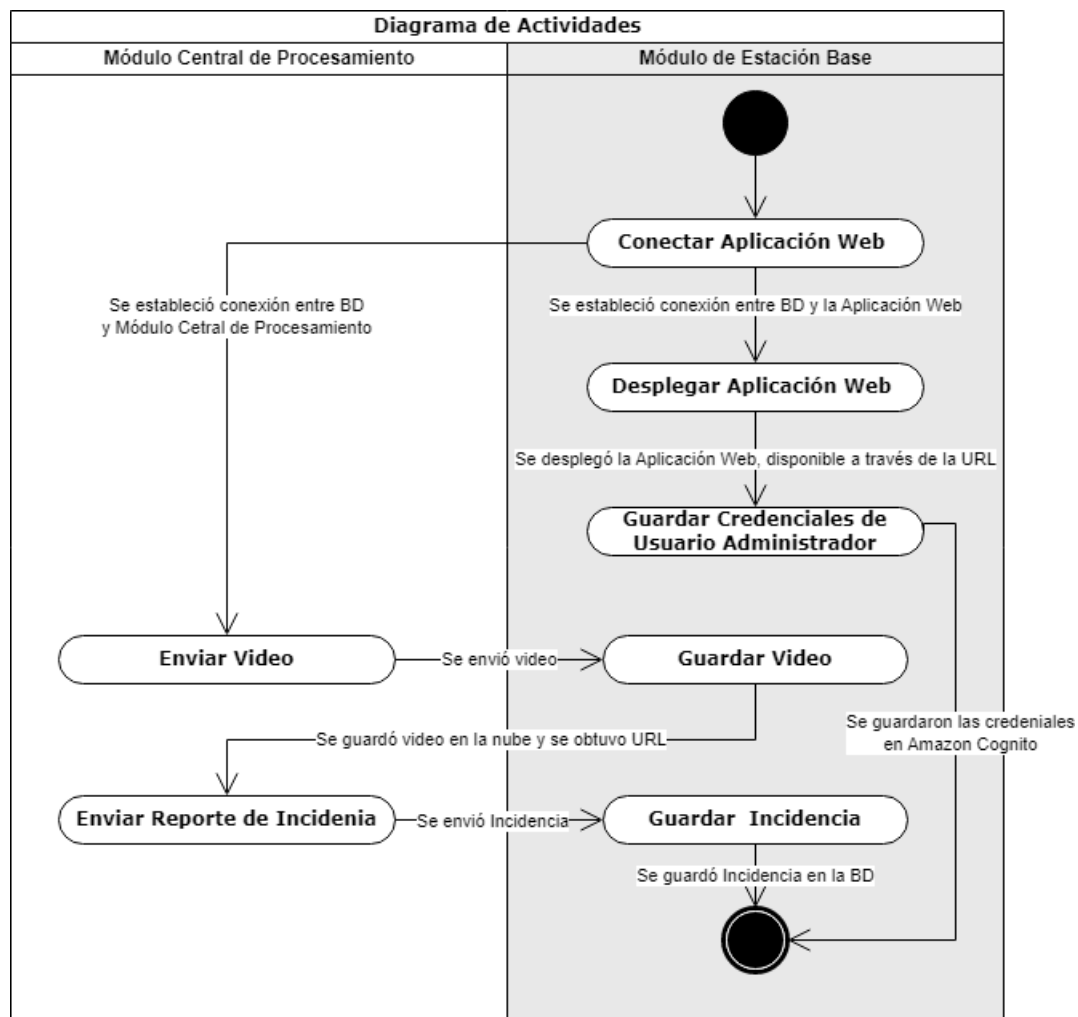


Figura 7: Diagrama de Actividades del Módulo de Estación Base

El diagrama de actividades del Módulo de Estación Base muestra la secuencia de las actividades que se realizarán en dicho módulo, la primera actividad conectará la aplicación web hacia la Base de Datos, lo que permitirá insertar, modificar, leer y eliminar los datos del conductor. En el caso de los Reportes de Incidencias, solo podrán ser visualizadas, modificadas o eliminadas desde la aplicación web, ya que la inserción de estas se deberá realizar desde el Módulo Central de Procesamiento, este módulo se encargará de realizar el reporte y posteriormente enviarlo a la Base de Datos. La segunda actividad consiste en el alojamiento y despliegue de la aplicación web, la cual estará disponible usando la URL dentro del navegador web.

Las credenciales del Usuario Administrador no se almacenarán en la Base de Datos, estas se guardarán en Amazon Cognito y serán utilizadas para iniciar sesión dentro de la aplicación. Como se mencionó anteriormente, el Módulo Central de Procesamiento realizará el Reporte de Incidencia, sin embargo, primero deberá guardar el video en la nube y posteriormente obtener la URL donde se almacenó el video, una vez recabados los datos necesarios para la incidencia el reporte será enviado y almacenado en la Base de Datos.

9. Diseño del Módulo de Estación Base



Figura 8: Diagrama de Comunicación para guardar video

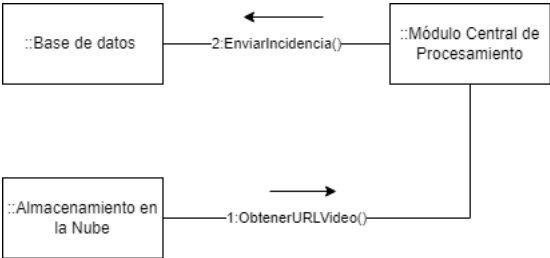


Figura 9: Diagrama de Comunicación para guardar incidencia



Figura 10: Diagrama de Comunicación para conectar aplicación web



Figura 11: Diagrama de Comunicación para desplegar aplicación web

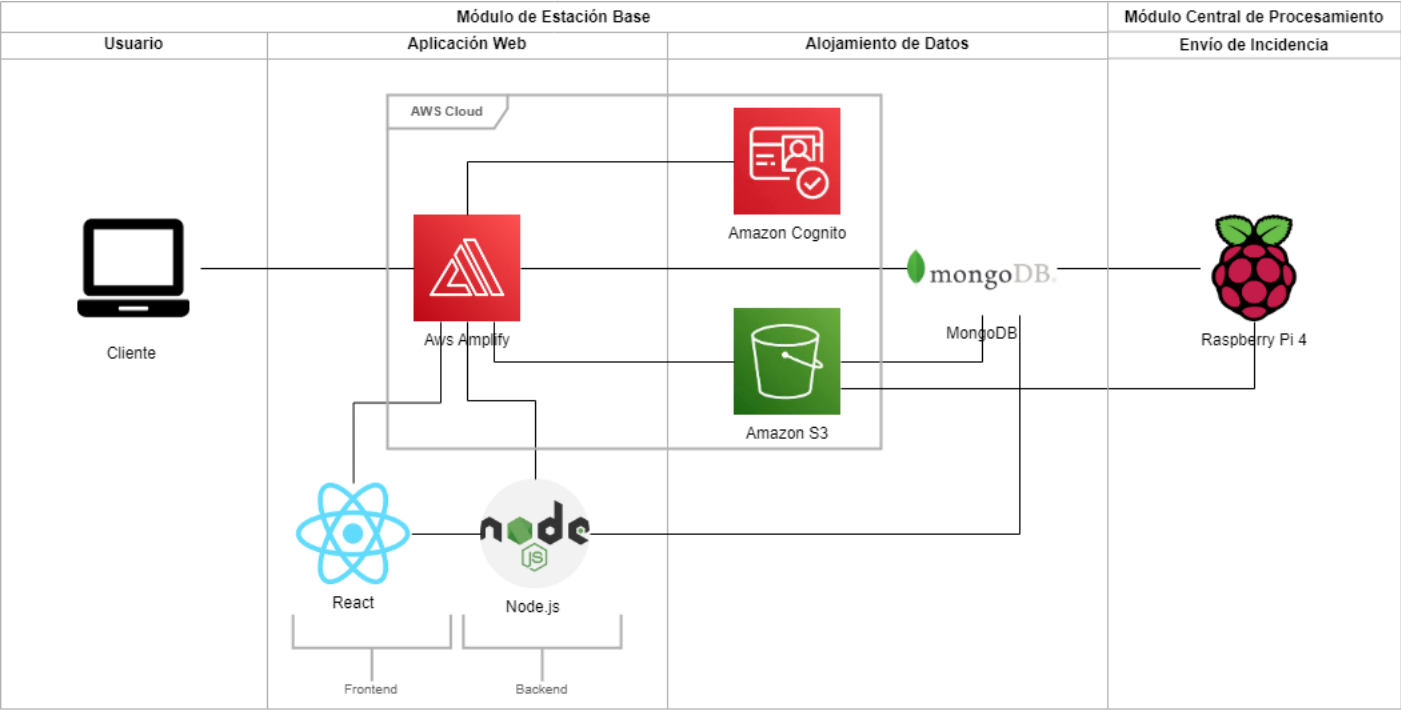


Figura 12: Diseño del Módulo de Estación Base

10. Búsqueda de servidores de alojamiento

Para el presente proyecto se requiere una capacidad de almacenamiento que nos permita el uso de archivos multimedia. Primero, veremos los diferentes tipos de medios que puede hospedar y qué funciones de hospedaje necesita para admitir esos medios.

Contenido Multimedia	Ejemplos
Imágenes	JPEG, GIF, archivos TIFF
Audio	MP3, WAV, AAC
Video	MP4, Youtube, QuickTime

Tabla 9: Tipos de Contenido Multimedia

El alojamiento de multimedia sugiere que el propietario de un sitio almacene y entregue una gran cantidad de imágenes, archivos de audio o videos, y los proporcione a como su contenido principal, a diferencia de los sitios que solo necesitan unos pocos recursos de medios para complementar principalmente los basados en texto. A continuación se describen los puntos mas importantes a considerar si se decide a tener un alojamiento basado en multimedia:

- **Ancho de banda**

Un plan de alojamiento en la nube le permite pagar por un uso menor durante las horas de menor actividad y explotar cuando recibe mucho tráfico. En pocas palabras, pagas por lo que utilizas. La flexibilidad de la computación en la nube ha llevado incluso a Netflix a abandonar la mayoría de sus centros de datos en favor del alojamiento en la nube de Amazon.

- **Almacenamiento**

Si se tiene una gran cantidad de archivos multimedia, entonces se requiere gran capacidad de almacenamiento. Lo ideal sería contar con almacenamiento ilimitado, pero esto no sería costeable. Se debe de analizar un aproximado de almacenamiento necesario y en base a eso se deberá elegir el plan de almacenamiento que mejor cubra esas necesidades.

- **CDN (*Content Delivery Network*)**

Lo último que debe considerar con el alojamiento de medios es una red de entrega de contenido o CDN. Una CDN intenta mejorar el rendimiento de la transmisión de medios mediante la ubicación de servidores que están geográficamente cerca del usuario mediante sofisticados algoritmos de ubicación.

Sabiendo lo anterior, se analizarán distintos proveedores de alojamiento con la intención de saber cual de ellos se adapta mejor a nuestras necesidades.

- **AWS Amplify Hosting**

AWS Amplify Hosting es un servicio de alojamiento y CI/CD *Continuous Integration/Continuous Delivery* completamente auto administrado para aplicaciones estáticas, rápidas, seguras, fiables, renderizadas del lado del servidor y que escalan con su empresa. Es compatible con marcos web modernos como React, Angular, Vue, Next.js, Gatsby, Hugo, Jekyll, entre otros. Algunas de sus principalmente ventajas son:

- **Implementación de contenido web con rapidez**

AWS permite la implementación continua de una aplicación web estática o renderizada del lado del servidor, una página de inicio de la aplicación móvil o una aplicación progresiva en cada confirmación de código.

- **Alojamiento confiable cerca de los usuarios**

AWS Amplify Hosting cuenta con la red de entrega de contenido (CDN) de Amazon CloudFront, con cientos de puntos de presencia en todo el mundo.

- **Mayor Velocidad de Producción**

Cuenta con una alta velocidad del ciclo de lanzamiento de su aplicación con flujos de trabajo de CI/CD integrados, versiones preliminares de solicitudes de extracción y pruebas.⁷

- **Monitoreo**

AWS cuenta con un sistema de monitoreo de tráfico en tiempo real. También permite crear alarmas personalizadas para enviar notificaciones cuando la métrica haya superado el límite establecido.

- **Precio**

AWS cuenta con dos planes, uno gratuito y otro de paga, a continuación se detallan las características de cada uno.

Aloje una aplicación Gratuito por 12 meses	Aloje una aplicación Pago por uso
Implemente fácilmente su aplicación web o sitio web en la red de entrega de contenido (CDN) de AWS rápida, segura y confiable mediante Amplify Hosting.	¿Necesita más capacidad que el nivel gratuito? Amplify cambia a los precios de pago por uso una vez que supera los límites del nivel gratuito, sin interrupciones en sus aplicaciones.
CREACIÓN E IMPLEMENTACIÓN	CREACIÓN E IMPLEMENTACIÓN
Sin costo hasta los 1000 minutos de creación por mes	0,01 USD por minuto
ALMACENAMIENTO DE DATOS	ALMACENAMIENTO DE DATOS
Sin costo hasta los 5 GB almacenados en la CDN por mes	0,023 USD por GB por mes (este cargo se repite hasta que se elimine la aplicación)
TRANSFERENCIA SALIENTE DE DATOS	TRANSFERENCIA SALIENTE DE DATOS
Sin costo hasta los 15 GB por mes	0,15 USD por GB servido

Figura 13: Tipos de Planes de AWS Amplify

■ A2Hosting

Además de los planes de alojamiento, en A2 Hosting también puede obtener nombres de dominio y sus precios son justos. Por ejemplo, un nombre de dominio .com le costará \$14.95 por año. Entre otras extensiones, puedes adquirir: dominios .com, .info, .net y .org. Los TLD específicos de países como .co.uk, .de, .fr, .es o .au no están disponibles. Algunas de sus características se listan a continuación:

- **Alojamiento Compartido**

A2Hosting es muy popular por sus planes de alojamiento compartido. Su costo de renovación va desde \$10.99 a \$25.99 por mes. Es el que necesitarán la mayoría de los proyectos, a menos que tenga un sitio con mucho tráfico o necesite una configuración de servidor especial.

- **VPS**

VPS significa servidor privado virtual. Es una forma de dividir un servidor en (sub)servidores individuales más pequeños. Esto significa que se puede configurar de acuerdo a las necesidades del cliente y no tener que compartir recursos con otros clientes. Un VPS en A2 Hosting cuesta desde tan solo \$7.65 al mes hasta alrededor de \$200.

- **CDN**

A2 Hosting no ofrece servicio de CDN, sin embargo sus planes ofrecen la posibilidad de contratar un servicio de CDN proveedores externos pagando un precio extra del precio base dependiendo del plan contratado.

- **HostGator**

- **Tiempo de Respuesta**

HostGator posee tiempos de respuesta mínimos para el territorio de EEUU. No tanto así para el resto del mundo. Esto debido a que solo poseen dos centros de datos, uno en el área este de EEUU y otro en el área oeste. A continuación se muestra una comparación de los tiempos de respuesta de distintos territorios del mundo. Esta información es obtenida de la página de Bitcatcha, que se encarga de comparar los tiempos de respuesta de distintas páginas web desde distintos lugares del mundo.

US (W)	US (E)	London	Singapore	Sao Paulo
32 ms	36 ms	221 ms	221 ms	139 ms
Bangalore	Sydney	Japan	Canada	Germany
717 ms	205 ms	153 ms	36 ms	140 ms

Figura 14: Tiempos de Respuesta

- **Disponibilidad**

HostGator ofrece una disponibilidad del 99.9 %, esto quiere decir que los sitios que usen su servicio de alojamiento, rara vez tendrán problemas de disponibilidad.

- **Precios**

HostGator cuenta con dos tipos de planes, uno gratuito y uno de paga. El plan de gratuito ofrece hasta 2 GB de almacenamiento gratis. Sin embargo, el tipo de hosting es uno compartido, lo cual puede ocasionar lentitud en el tiempo de respuesta del sitio web. El plan de paga ofrece hasta 100 GB de almacenamiento y hosting individual, el precio de este plan es de 11.59 dls mensuales

- **CDN**

HostGator no cuenta con ningún tipo de CDN, si se requiere una CDN se tendría que contratar de manera externa sólo con el plan de paga.

■ BlueHost

Bluehost se destaca con sus opciones de alojamiento compartido y VPS, que son ideales para individuos, pequeñas empresas, blogueros y otros que necesitan un sitio web confiable. El hecho de que Bluehost haga que sea tan fácil crear y administrar un sitio también hace que su servicio sea excelente para aquellos que son nuevos en la ejecución de sitios web.

- **Precios** Bluehost tiene alojamiento para todo tipo de sitios web, lo que significa que tiene opciones a muchos precios diferentes. Puede comenzar con una cuenta básica de alojamiento compartido, que comienza en \$2.95 por mes. Si se necesita un servidor dedicado, los precios rondan desde \$119.99 por mes por la opción de gama alta.

- **Servidores de Alto Rendimiento**

Con los servidores de alto rendimiento, Bluehost asigna menos usuarios por servidor y cada usuario recibe un recuento de archivos de 300 000; esto significa más recursos informáticos por usuario, velocidades de carga constantes del sitio y menos tiempo de inactividad causado por usuarios deshonestos. Sin embargo sus servidores de alto rendimiento solo están disponibles con su plan Pro, que es mucho más caro que el resto de sus planes de alojamiento web compartido.

- **CDN**

Desafortunadamente, el servicio de hosting de BlueHost no cuenta con ninguna CDN, lo que puede significar una pérdida del tiempo de respuesta en cuanto al manejo de contenido multimedia.

11. Análisis de la aplicación Web

ID	Nombre corto del Requerimiento
<i>RF01</i>	<i>Iniciar Sesión</i>
Descripción: El sistema permitirá iniciar sesión en la aplicación web	
Elementos: <ul style="list-style-type: none">• Amazon Cognito• Administrador• Credenciales• MongoDB	
Solución del Requerimiento: <p>El sistema comprobará que las credenciales ingresadas por parte del usuario administrador se encuentren en la base de datos. En caso contrario, la aplicación web indicará que ese usuario no se encuentra registrado en la base de datos.</p>	

Tabla 10: RF01- Iniciar Sesión

ID	Nombre corto del Requerimiento
<i>RF02</i>	<i>Mostrar el Historial de Reportes de Incidencia</i>
Descripción: El sistema desplegará en forma de lista todas las incidencias que se tengan registradas en la base de datos	
Elementos: <ul style="list-style-type: none">• Conductor• Incidencias• Administrador• MongoDB	
Solución del Requerimiento: <p>El sistema recuperará de la base de datos todas las incidencias que se tengan registradas.</p>	

Tabla 11: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF03</i>	<i>Visualizar Reporte de Incidencia</i>
Descripción: El sistema desplegará los detalles específicos de cada incidencia registrada.	
Elementos: <ul style="list-style-type: none">• Incidencia• Administrador• Conductor	
Solución del Requerimiento: <p>El usuario administrador podrá visualizar los reportes individuales de incidencias de cada conductor al hacer click en cualquiera de las incidencias mostrada en la lista principal. Los reportes contendrán información sobre la fecha, hora, ubicación y un video corto del momento en que fueron detectados síntomas de somnolencia.</p>	

Tabla 12: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF04</i>	<i>Confirmar Incidencia</i>
Descripción: El sistema permitirá al administrador confirmar la incidencia, esto para descartar que se trate de un falso positivo.	
Elementos: <ul style="list-style-type: none">• Administrador• Incidencia• Conductor• MongoDB	
Solución del Requerimiento: <p>El usuario administrador podrá confirmar la incidencia después de haber revisado el videoclip del momento de somnolencia con la intención de descartar falsos positivos. Esto será posible ingresando a una incidencia específica mostrando sus detalles.</p>	

Tabla 13: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF05</i>	<i>Recuperar Contraseña</i>
Descripción: El sistema contará con una opción para recuperar la contraseña del administrador en caso de que sea olvidada la contraseña.	
Elementos: <ul style="list-style-type: none">• Amazon Cognito• Email• Administrador• MongoDB	
Solución del Requerimiento: <p>El sistema requerirá que el usuario administrador ingrese el correo con el que fue registrado. Posteriormente se le enviará un código de recuperación de contraseña a ese correo. El administrador ingresará se código en el apartado de recuperar contraseña y así podrá ingresar una nueva contraseña.</p>	

Tabla 14: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF06</i>	<i>Mostrar Perfil Conductor</i>
Descripción: El sistema permitirá al usuario administrador visualizar los datos de cada conductor registrado en la base de datos.	
Elementos: <ul style="list-style-type: none">• Perfil• Conductor	
Solución del Requerimiento: <p>El administrador podrá consultar cada uno de los perfiles de los conductores registrados en la base de datos dando click en el nombre del mismo. En dicho perfil se mostrarán datos como nombre, apellido, así como el número de incidencias de dicho conductor, con sus respectivos detalles.</p>	

Tabla 15: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
RF07	<i>Mostrar Ubicación Geográfica</i>
Descripción: La aplicación web, con ayuda de los datos proporcionados por el módulo de telemetría, mostrará la ubicación en tiempo real de un conductor.	
Elementos: <ul style="list-style-type: none">• Ubicación en Tiempo Real• Módulo de Telemetría	
Solución del Requerimiento: El sistema permitirá al administrador consultar la ubicación de los conductores en tiempo real. Para esto el administrador deberá de ingresar previamente al perfil del conductor del cual desea consultar dicha ubicación.	

Tabla 16: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
RF08	<i>Descartar Incidencia</i>
Descripción: El sistema permitirá catalogar una incidencia como Falsa si fuera el caso.	
Elementos: <ul style="list-style-type: none">• Incidencia• Administrador	
Solución del Requerimiento: En caso de presentarse una incidencia falsa, después de haber sido revisada por el administrador, esta podrá ser catalogada como falsa incidencia y será eliminada automáticamente de la base de datos.	

Tabla 17: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
RF09	<i>Registrar Usuario</i>
Descripción: El sistema le permitirá al administrador registrar nuevos conductores.	
Elementos: <ul style="list-style-type: none">• Incidencia• Administrador	
Solución del Requerimiento: En caso de presentarse una incidencia falsa, después de haber sido revisada por el administrador, esta podrá ser catalogada como falsa incidencia y será eliminada automáticamente de la base de datos.	

Tabla 18: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF10</i>	<i>Modificar Usuario</i>
Descripción: El sistema le permitirá al administrador modificar los datos del conductor. Elementos: <ul style="list-style-type: none">• Nombre• Apellido• Conductor• Administrador	
Solución del Requerimiento: El administrador podrá editar los datos de los conductores como nombre o apellido.	

Tabla 19: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
<i>RF11</i>	<i>Eliminar Usuario</i>
Descripción: El sistema le permitirá al administrador eliminar los datos del conductor. Elementos: <ul style="list-style-type: none">• Conductor• Administrador	
Solución del Requerimiento: El sistema le permitirá al eliminar los datos de conductores de la base de datos.	

Tabla 20: RF02- Mostrar el Historial de Reportes de Incidencia

ID	Nombre corto del Requerimiento
RF12	<i>Generar Mensaje de Alerta</i>
<p>Descripción:El sistema alertará con un mensaje en caso de que algún componente en alguna Unidad Contenedora no funcione correctamente.</p> <p>Elementos:</p> <ul style="list-style-type: none">• Módulo Central de Procesamiento• Administrador• Alerta	
<p>Solución del Requerimiento:</p> <p>El sistema enviará un mensaje de alerta al administrador indicando que Unidad presenta algún desperfecto, esto con la intención de sea reparada posteriormente.</p>	

Tabla 21: RF02- Mostrar el Historial de Reportes de Incidencia

12. Búsqueda de Lenguajes de Programación Web

Con respecto al desarrollo web, hay distintos parámetros en los que basarnos para poder elegir un lenguaje o lenguajes de programación. El primero, es que tipo de aplicación web se va a desarrollar. El segundo, es el tipo de contenido que tendrá nuestra página web.

■ Javascript

JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, también es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa

A continuación se describen sus características más importantes:

- **Veloz**

Como lenguaje de programación interpretado, no tiene que compilarse cada vez que se ejecuta, lo que hace que el desarrollo y la depuración sean más rápidos para comenzar. Además, JavaScript encuentra más velocidad al ejecutarse como un script del lado del cliente, ejecutándose en el navegador sin conectarse al servidor y ahorrando recursos valiosos para cada usuario adicional.

- **Procesamiento asíncrono**

El procesamiento asíncrono es una de las características más útiles del lenguaje JavaScript. Usando JavaScript, un bloque del script no podrá detenerse o dejar que el otro bloque de código espere a que comience la respuesta. Si se está procesando una solicitud, otras también trabajarán en paralelo con la solicitud anterior en lugar de esperar la respuesta de la solicitud anterior. Ahorra mucho tiempo al ejecutar scripts en paralelo.

- **Poca carga de Procesamiento del servidor**

JavaScript permite realizar funcionalidades básicas en el lado del cliente. Esto significa que el servidor no tendrá que procesar las funcionalidades básicas que mejoran el rendimiento del servidor.

■ PHP

PHP es otro lenguaje de programación del lado del servidor conocido que se lanzó oficialmente en 1997. Este lenguaje de fondo está arraigado en HTML y normalmente se usa para el seguimiento de sesiones, el diseño de sitios de comercio electrónico y la administración de bases de datos y contenido dinámico.

- **Simplicidad**

PHP es uno de los lenguajes de programación back-end más simples. La sencillez de esta programación lo ha convertido en uno de los lenguajes más utilizados, especialmente para el desarrollo web.

- **Independencia de Plataforma** PHP es un lenguaje independiente de la plataforma. Esto significa que puede funcionar para todos los sistemas operativos y plataformas.

- **Tipado débil**

En el lenguaje de programación PHP, no necesita mencionar el tipo de datos de una variable antes de asignar el valor. El tipo de datos de la variable se basará en los datos almacenados en ella durante la ejecución.

- **Flexibilidad**

PHP es un lenguaje flexible, y se puede usar muy fácil y efectivamente con HTML, XML y JavaScript. Este lenguaje se puede incrustar e integrar fácilmente con estos lenguajes de secuencias de comandos para crear una aplicación.

- **Python**

El desarrollo web en Python no siempre es la elección principal para construir el lado del servidor de los sitios web. Este lenguaje es multipropósito, lo que significa que es utilizable en la mayoría de los campos de programación. Sin embargo, Python para el desarrollo web es también una sólida opción, atrayendo a la gente con su sintaxis simple, una variedad de marcos de trabajo, y un fuerte ecosistema general. Instagram, una de las redes sociales dominantes, utiliza Python para su plataforma, recibiendo millones de visitantes al día. A continuación se describen sus ventajas mas importantes:

- **Lenguaje Orientado a Objetos** Python es compatible con el paradigma de programación de la programación orientada a objetos. Esto significa que puede definir clases y objetos para escribir código estructurado para su programa o aplicaciones. Ayuda a comprender mejor el código y también a limpiar el código. El código puede ser reutilizado fácilmente usando los conceptos de programación orientada a objetos. Este permite reducir las líneas de código.

- **Librerías**

Python ofrece una amplia gama de bibliotecas, que incluyen diferentes tipos de funcionalidades. Por ejemplo, OpenCV es una biblioteca de Python para Computer Visions y Selenium se usa ampliamente para la automatización web y *web scraping*.

13. Elección del lenguaje de programación para la aplicación web

14. Diseño de la aplicación web

15. Cotización de materiales

16. Conclusiones

En un principio, se tenía contemplado utilizar la red LoRaWan para la comunicación y transferencia de archivos. Sin embargo, después de las actividades investigación, nos dimos cuenta de que esta opción no era viable, ya que el *bitrate* soportado por LoRaWan es insuficiente para la transferencia de archivos multimedia, que es una parte fundamental de nuestro sistema. Por tanto, se optó por utilizar la interfaz de LTE-M, una opción más costosa, pero que cumple los requisitos de una alta tasa de transferencia, así como una buena integración con la Raspberry Pi 4.

En el caso del Submódulo de Procesamiento de imágenes, se tenía contemplado desarrollarlo de manera separada del submódulo de *machine learning* pero después de haber realizado un análisis más profundo, se decidió integrarlo dentro de este mismo. Esto para el ahorro de recursos de procesamiento y conseguir una mejor comunicación entre procesos. En un siguiente reporte, se terminará por diseñar la Red Neuronal Convolucional, así como el diseño del Módulo de Telemetría y de la Estación Base.

Pará el diseño de la unidad de almacenamiento se obtuvo un bosquejo prototipo, sin embargo, se tiene contemplado rediseñar el modelo con el fin de mejorar y adaptarla los cambios del sistema a futuro.

17. Bibliografía

Referencias

- [1] Gupta , S. (2022) *Best language for machine learning: Which should you learn?*, Springboard Blog. Springboard . Available at: <https://www.springboard.com/blog/data-science/best-language-for-machine-learning/> (Accessed: October 30, 2022).
- [2] Irv Kalb, *Object-Oriented Python*, Primera Edición ,No Starch-Press, 2021
- [3] V.Moret Bonillo, *Fundamentos de Inteligencia Artificial*, Segunda Edición,Santiago de Compostela: Universidad de La Coruña Servicio de Publicaciones, 2005.
- [4] Ian Sommerville, *Ingeniería de Software*, Novena Edición , Pearson Educación de México, 2011
- [5] Kenneth E. Kendall, Julie E. Kendall, *Análisis y Diseño de Sistemas*, Octava Edición , Pearson Educación de México, 2011
- [6] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow*, Segunda Edición , O'Reilly, 2009
- [7] Kurt Demaagd, Anthony Oliver, Nathan Oostendorp y Katherine Scott *Practical Computer Vision with OpenCV*, Tercera Edición, O'Reilly, 2017
- [8] Cuno Plister, *Getting Started with Internet of Things*,Primera Edición, O'Reilly, 2011
- [9] Vilca Espinoza, R.A, *Influencia de un sistema de geolocalización en el control y monitoreo de vehículos con dispositivos GPS en una empresa logística*, 2007