

F-P-EA-07-01/04

CARTA DE ACEPTACIÓN DEL INFORME DE ESTADÍA

Aceptamos de conformidad el Informe Final de Estadía denominado: Plataforma web y móvil para administrar solicitudes de recuperación de piezas y refacciones producto de siniestros, presentado por el (la) C. Antonio Dimas Fernández, con número de matrícula 21302127, quien puede dar inicio al trámite de titulación.

ASESOR (A) EMPRESARIAL

ASESOR (A) INDUSTRIAL (UNIDAD ECONÓMICA)		
NOMBRE	FIRMA Y SELLO	FECHA DE AUTORIZACIÓN
Juan Manuel Alcívar Gama		31/08/2023
ASESOR (A) ACADÉMICO (UTFV)		
NOMBRE	FIRMA Y SELLO	FECHA DE AUTORIZACIÓN
Juan Pablo Ibáñez Bautista		31/08/2023
ENCUESTA DE SATISFACCIÓN DE EGRESADOS		
NOMBRE	FIRMA Y SELLO	FECHA DE AUTORIZACIÓN
Len D. Jazmin Ivonne Madrigal Reyes DEPARTAMENTO DE DESEMPEÑO DE EGRESADOS I 23-3/0076		28 Septiembre-2023

"Por una Educación Tecnológica de Excelencia"

Universidad Tecnológica
Fidel Velazquez

RECIBIDO

DIVISIÓN ACADÉMICA DE INFORMÁTICA

ING. OSCAR ARMANDO BAEZA HUICOCHEA
DIRECTOR DE LA DIVISIÓN ACADÉMICA DE
INFORMÁTICA

SECRETARÍA DE EDUCACIÓN
SUBSECRETARÍA DE EDUCACIÓN SUPERIOR Y NORMAL
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR



Universidad Tecnológica
Fidel Velázquez



Universidad Tecnológica Fidel Velázquez

División Académica de Informática

“Sistema de localización y recuperación de refracciones
siniestradas”

Informe final de estadía que para obtener el de
Técnico Superior Universitario en Tecnologías de la
Información, Área Desarrollo de Software Multiplataforma

Presentan:

Sánchez Bernal Monserrat
Escobedo González Cristian
Serrano Moreno Israel
Dimas Fernández Antonio.

Asesor Industrial: Lic. Juan Manuel Alcívar Gama.

Asesor Académico: Juan Pablo Ibáñez Bautista.

Septiembre, 2023

DEDICATORIA

Para mis padres que espero y logré cumplir con las metas que les prometí.

Quiero dedicar este proyecto a mi familia, en especial a mis padres y abuelos que siempre me han ayudado y visto por mí desde que tengo memoria, es gracias a ellos que puedo salir adelante y pude terminar este proyecto gracias a su inmenso apoyo y amor incondicional que siempre me brindan.

Se lo dedico a mis padres que han sido las personas que más me han motivado a seguir continuando con esta carrera, así como me han impulsado a mejorar día tras día, dándome su apoyo en cualquier situación que se presente, en poder disfrutar esta nueva fase en la carrera y así poder concluir con éxito.

Este proyecto lo dedico a toda mi familia por apoyarme en todos los sentidos al igual que mi novia e internamente igual a mí mismo por el esfuerzo y dedicación que le tome al proyecto.

AGRADECIMIENTOS

Quiero agradecer a mis padres y abuelos que fueron mi soporte para poder seguir adelante y motivándome a terminar esta etapa de la universidad la cual fue todo un reto, pero gracias a ellos, su apoyo y amor pude salir adelante.

Quiero agradecer a mi familia por el apoyo y la confianza que me han tenido durante este tiempo a la hora de tomar las decisiones que me han llevado a este punto

Quiero agradecer a mis padres que hasta la fecha son en las únicas personas de las cuales me dado todo su apoyo y confianza en el cual a pesar del tiempo que he dedicado a esta etapa no han dejado de creer en mí, y me ayudan en cualquier situación que se presente y que se puede presentar en el futuro.

Agradezco la atención y apoyo de nuestros asesores y compañeros por lo cual pudimos completar el proyecto.

INDICE

INTRODUCCIÓN.....	5
RESUMEN.....	10
Capítulo 1. Marco Contextual.....	12
1.1 Planteamiento del problema.	12
1.2. Objetivo General	12
1.3. Objetivos Específicos.	13
1.4. Justificación.....	13
1.5. Alcance.	13
Capítulo 2. Marco Teórico.....	15
2.1 Metodología	15
2.1.1 Metodología de trabajo (scrum)	15
2.2. Herramientas de desarrollo	19
2.2.1. ¿Qué es Node?	19
2.2.2. ¿Qué es React?.....	20
2.2.3. ¿Qué es MySQL?	24
Capítulo 3. Desarrollo	25
3.1 Desarrollo	26
3.2 Diseño del sistema.....	27
3.3 Codificación	34
3.4 Prueba	37
Apéndice.....	42
Conclusiones	44
Glosario de Términos.....	45
Bibliografía.....	46

INTRODUCCIÓN

Descripción de la empresa

Somos especialistas en servicios de tecnologías de información y consultoría empresarial registrada ante la STPS con Folio: 85cb2511-fd5e-4154-add1-e492652d29d6 como prestadora de servicios especializados, contamos con más de 10 años de experiencia en la prestación de servicios tanto para empresas nacionales como trasnacionales otorgando a nuestros clientes una amplia gama de soluciones y experiencia de nuestros colaboradores

Tipo de organización:

Con base en su información actual y con el apoyo del departamento de TI realizamos análisis de datos e identificamos operaciones de excepción inusuales o poco comunes basado en políticas y procedimientos de la organización.

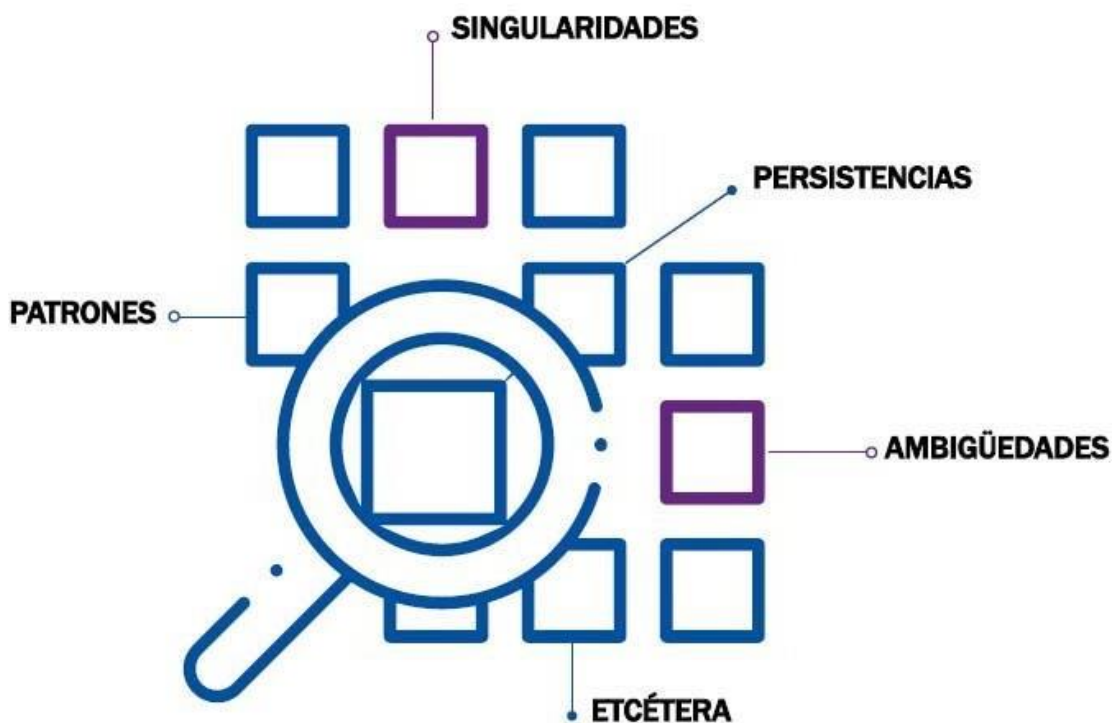


Ilustración 1. Organización

Historia

Grupo DITEC fue creado por cuatro alumnos salidos de la Universidad Tecnológica Fidel Velázquez, los cuales se juntaron para formar una empresa , conseguimos algunos clientes y uno de ellos nos brindó la oportunidad de colaborar en un proyecto y a partir de ahí se creó una relación de negocios eventualmente el actual director le compro sus acciones a dos de nuestros socios convirtiéndonos en tres y trabajando para met life por más de 10 años y adquiriendo otras cuentas como Monterrey new York life , eden red editorial planeta Zurich, seguros prevem entre otros brindando trabajo a más de 70 personas a lo largo de 13 años.

Misión

Somos un equipo interdisciplinario con experiencia y pasión, que busca el bienestar social de nuestros clientes y colaboradores en ámbito tecnológico y humano, diseñando soluciones integrales que les genere información de valor permitiéndoles conocer mejor su organización.

Visión

Ser una empresa de referencia positiva y con reconocimiento en el servicio al cliente a nivel nacional e internacional

Organigrama

Director General: Humberto Hernández Vázquez

Director TI: Juan Manuel Alcivar Gama

Gerente de Recursos Humanos: Marco Antonio Enríquez Álvarez

Gerente de Cuentas: Guillermo López Ledezma

1.2. Descripción de proceso general o específico

Desarrollo de sistemas y software a la medida: Creamos desarrollos de software a la medida para adaptar sus sistemas a las nuevas funcionalidades que requiere el negocio o bien, para crear un sistema único a las necesidades de su operación, nuestra fábrica de software creará soluciones magníficas para su empresa cumpliendo sus expectativas en todo momento.



Ilustración 2. Primer Producto

DITEC se mantiene a la Vanguardia acerca de las nuevas plataformas de desarrollo, así como mantiene vigencia de aquellas clásicas y robustas.



Ilustración 3. Lenguajes de programación.

1.3. Descripción de productos

Asignación por tiempo de expertos en desarrollo (Outsourcing): Si su departamento de tecnologías de la información requiere integrar expertos en las distintas áreas, somos una empresa de servicios especializados registrada ante la STPS con no. Folio: 85cb2511-fd5e-4154-add1-e492652d29d6 Actividad: a) Elaboración y maquila de programas y sistemas operativos (software) a la medida, así como su almacenaje, conservación, transportación y comercialización.



Ilustración 4. Asignaciones

Nuestro personal tiene experiencia probada en el campo laboral y de servicio, así como en el sector Bancario, Seguros, Gobierno e Iniciativa privada.

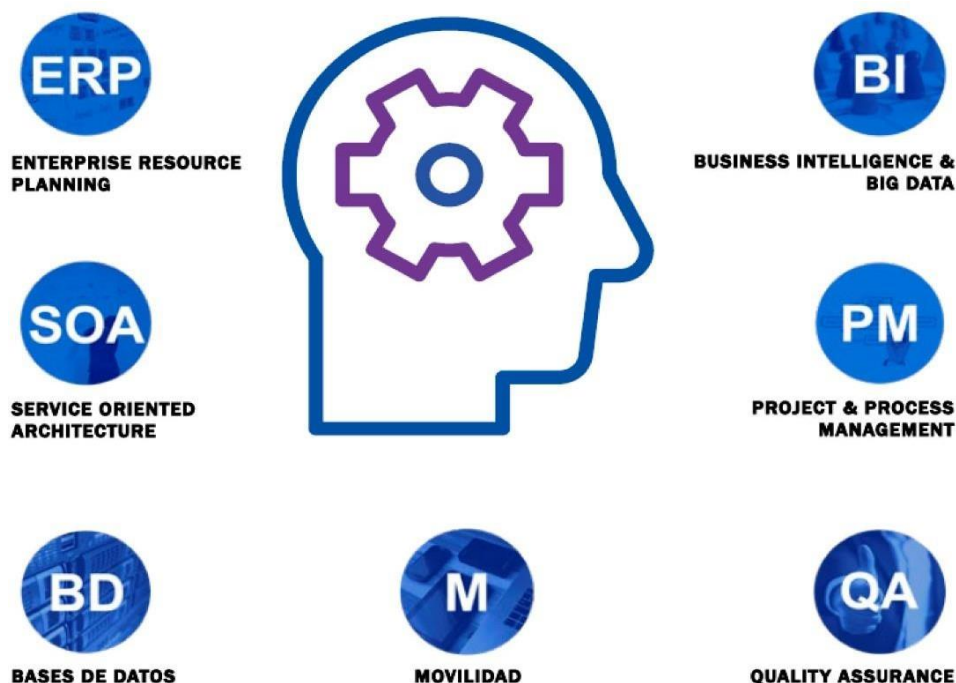


Ilustración 5. Capacidad del personal

1.4. Descripción de proveedores

Grupo DITEC cuenta con un solo proveedor que es el que se encarga de sus principales funciones dentro de la empresa como, la contabilidad, nomina, etc. El proveedor es conocido como: SOLANO VALLEJO Y ASOCIADOS, S.C.

Descripción de dicha firma: Desde sus inicios hasta ahora, ha brindado servicios de calidad, eficiencia y puntualidad a sus clientes, a través de estos años ha adquirido y desarrollado experiencia y destreza, lo cual es reflejado en la cantidad de clientes que hoy atiende. Préstamos servicios en toda la república mexicana.

1.5. Descripción de clientes

Los principales clientes asociados con el grupo DITEC son algunas empresas de seguros de auto y una de ellas está relacionada con un banco, los cuales son:

- ❖ MetLife: Más de 7 años dando servicio y proporcionando recursos de alto nivel. Logros: Migración de sistemas legados a ALICO a los sistemas de MetLife, Implementación de la Facturación electrónica de los sistemas legados a la ver3.3, Implementación de servicios en Portal clientes y agentes.
- ❖ Principal.
- ❖ Prevem seguros.
- ❖ AIG.
- ❖ Bancomer.
- ❖ Kuo.
- ❖ Grupo VIDANTA
- ❖ GIANT
- ❖ Seguros Monterrey: Desde el 2014 proporcionando recursos de especializados y de alto nivel. Logros: Soluciones tecnológicas hechas a la media en todo momento.
- ❖ Edenred: Desde el 2014 proporcionando recursos de especializados y de alto nivel. Logros: Soluciones tecnológicas hechas a la media en todo momento.



Ilustración 6. Clientes del grupo DITEC

RESUMEN

En el presente documento se llevará a cabo explicar a cabo de un sistema hecho por 4 personas para una empresa de TI con el objetivo de mejorar la interfaz y el sistema base con el cual se está trabajando de esta manera se busca que sea más eficiente a la hora de usar y más dinámico con el usuario.

El trabajo está dividido en 3 capítulos. El primer capítulo menciona información relevante a la empresa tales como su historia y objetivos. El segundo capítulo habla sobre el tipo de metodología en la cual se ha estado trabajando, en la cual, se menciona toda la información correspondiente a este mismo; que es, como funciona, que hace y también las herramientas de desarrollo explicando los procesos que este mismo utiliza. En el tercer capítulo se explica el desarrollo del sistema, así como los requerimientos de software y hardware que se necesita para instalarlo

En la última parte del documento se habla sobre la conclusión del proyecto en donde tratamos de explicar la seguridad y eficiencia a la hora de recuperar las partes dañadas de los siniestros, dando a entender que se buscó que el sistema fuera eficiente y seguro para todo tipo de cliente que fuera a trabajar con nosotros.

Capítulo 1. Marco Contextual

Capítulo 1. Marco Contextual

1.1 Planteamiento del problema.

El grupo DITEC maneja diferentes actividades y relaciones con algunas aseguradoras de automóviles como: MetLife, Seguros Monterrey y Edenred, al igual que con el banco Bancomer, en los cuales pueden tener un diseño de sistemas y software, así como auditorías y análisis de datos, de igual forma como diseño gráfico, entre otras diferentes actividades.

El ingeniero a cargo de este nuevo proyecto tiene como objetivo crear un software para las aseguradoras en el cual mediante un siniestro puedan recuperar las piezas pérdidas o cambias de un automóvil que estuvo en cualquier situación y haya ocupado dicho cambio, esto con el fin de tener de manera más detalla los principales elementos que se necesitan para dicha recuperación de las piezas tales como: Fecha, No. Proveedor, No. Siniestro, No. Reporte, Año, Marca, Tipo, Modelo, Color, Placas, Serie y Refracción a cambia, una vez que él estos datos estén contemplados puedan alojarse en un base de datos en el cual se pueda tener mayor organización para que en el momento que se requieran recuperar dichas refacciones se pueda hacer de formas precisa y sin complicaciones.

1.2. Objetivo General.

Desarrollar una aplicación para las aseguradoras para la recuperación de piezas de automóviles, usando Note.js, Visual Studio, React, y MySQL para brindar una mayor información al momento de localizar las piezas, con lo cual se gana mayor organización y seguridad del sistema

1.3. Objetivos Específicos.

- Análisis del software a desarrollar.
- Implementación y relación con una base de datos.
- Creación de la interfaz que se pueda abrir en diferentes dispositivos y no tenga algún cambio en el diseño.
- Tener control específico de lo que se va a realizar en el proceso de desarrollo.

1.4. Justificación.

Con el desarrollo de dicho software se tiene las expectativas que podamos tener mayor acceso a los datos necesarios para la recuperación de las refacciones pérdidas o cambiadas mediante un siniestro ocurrido.

1.5. Alcance.

Al desarrollar este software solo lo realizamos a nuestro asesor empresarial todo con el objetivo de haber adquirido más experiencia y pulir nuestras habilidades aprendidas en la carrera y meramente para cumplir con el propósito de nuestra primera estadía la cual es el aprendizaje y una idea más clara del mundo laboral.

Capítulo 2.

Marco Teórico

Capítulo 2. Marco Teórico

2.1 Metodología

2.1.1 Metodología de trabajo (scrum)

Scrum es un marco de gestión de proyectos de metodología ágil que ayuda a los equipos a estructurar y gestionar el trabajo mediante un conjunto de valores, principios y prácticas. Al igual que un equipo de rugby (de donde proviene su nombre) cuando entrena para un gran partido, scrum anima a los equipos a aprender a través de las experiencias, a auto organizarse mientras aborda un problema y a reflexionar sobre sus victorias y derrotas para mejorar continuamente.



Ilustración 7. Procesamiento del Scrum

La metodología Scrum pasa por diferentes fases que hacen posible que se lleve a cabo con éxito:

ROLES EN EL EQUIPO:

Para que el equipo de trabajo sea capaz de cumplir su objetivo, es decir, ofrecer resultados de calidad y entregar valor al cliente, es necesario que cada miembro sea capaz de organizarse de forma multifuncional y autodidacta. Cada trabajador es responsable de unas tareas y tiene que finalizarlas en los tiempos acordados para que se produzca la entrega de valor del equipo completo y no se precise de la supervisión exhaustiva de otros miembros de la empresa.

Los tres roles principales en un equipo de trabajo Scrum son:

- 🚦 Propietario de producto o Product Owner: Se asegura de que el equipo trabaje adecuadamente desde la perspectiva del negocio. Es la única

persona que habla de forma continua con el cliente, por lo que tiene que conocer el negocio y el mercado a la perfección. Este perfil solo puede ser ocupado por una persona. Además, se responsabiliza de maximizar el valor del trabajo, que viene unido a la buena gestión del Product Backlog (que explicaremos más adelante).

- ✚ Facilitador o Scrum Master: Es el encargado de que se comprendan y se cumplan las reglas del marco scrum, es decir, es mayor especialista de scrum en el equipo. Elimina los inconvenientes o impedimentos para que se desarrolle el objetivo del sprint. Da la información necesaria y asesora al Product Owner y al equipo de desarrolladores.
- ✚ Equipo de desarrolladores: Formado por profesionales, recomendablemente de 3 a 9 personas, que se ocupan de realizar las tareas a las que ha dado prioridad el Product Owner. Son los únicos que estiman las tareas del product backlog. Es un equipo autoorganizado y multifuncional que dispone de habilidades transversales para realizar el trabajo, como diseño, pruebas, análisis, documentación, etc.

ARTEFACTOS

Para adentrarnos en este marco de trabajo, vamos a empezar identificando los tres artefactos de scrum. Artefacto se refiere a algo que fabricamos como, por ejemplo, una herramienta para solucionar un problema. Es necesario que todos los miembros del equipo tengan una misma visión de los artefactos, de lo que ocurre dentro del proyecto. Los artefactos son tres:

✚ PRODUCT BACKLOG O PILA DE PRODUCTO

Es el listado dinámico de tareas, funciones y mejoras que reúne todo un proyecto, es decir, engloba los requisitos del mismo y suelen llamarse 'historias de usuario'. Examina y prioriza los requisitos desde la perspectiva del cliente. Comienza con una visión principal del producto y evoluciona durante el desarrollo del mismo. Aunque es un documento abierto, solo puede modificarlo el product owner, o lo que es lo mismo, el propietario o gestor de productos, porque es el que está siempre en contacto con el cliente.

✚ Sprint backlog o pila del sprint

Es el grupo de tareas o requisitos que elige el equipo de desarrollo en el Sprint Planning junto con el plan para poder llevarlas a cabo durante un sprint. Este artefacto puede ser flexible y evolucionar durante un sprint.

✚ Product Increment o Incremento de scrum

Es el producto final utilizable de un sprint, es decir, el resultado de sumar todos los elementos del Product Backlog completados durante el sprint actual, teniendo en cuenta también el valor de los sprints anteriores.

EVENTOS DE SCRUM:

En este marco de trabajo existe un conjunto de eventos secuenciales o reuniones que los equipos de scrum deben realizar de forma periódica. Estos eventos son:

SPRINT:

Se trata del periodo de tiempo en el que el equipo trabaja para finalizar una cantidad de tareas establecidas. El sprint es el corazón de este framework y hacerlo de forma correcta potenciará la eficacia del trabajo y reducirá los problemas a la hora de lanzar el mejor software.

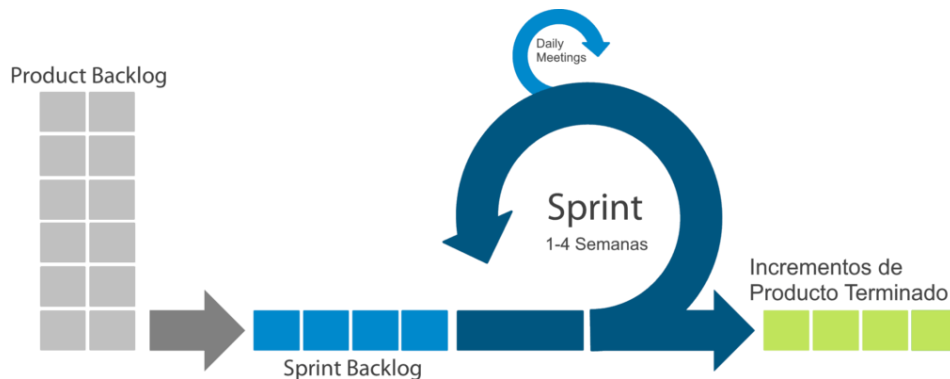


Ilustración 8. Explicación del Sprint

La duración de los sprints debe ser constante y estar definida por el equipo en función de su experiencia. La duración normal del sprint es de 2 a 3 semanas y se va ajustando al ritmo que lleve el equipo, siempre siendo lo más eficiente posible y sin relajarse. El máximo tiempo para un sprint es de 4 semanas porque si se alarga, el proyecto puede peligrar debido a que se corre el riesgo de perder feedback valioso con el cliente. Cuando llega el final del sprint, el equipo debe presentar los avances del proyecto y el resultado, que debe ser un producto que pueda entregarse al cliente.

PLANIFICACIÓN DE SPRINT:

El equipo de scrum tiene una reunión de planificación de sprint justo al comienzo para identificar y establecer las tareas y el objetivo (Sprint Goal) que van a ser capaces de llevar a cabo durante el sprint. La primera reunión que se hace en el sprint puede durar hasta 8 horas para sprints de un mes. El equipo, por un lado, decide qué se va a hacer en el sprint y se eligen las tareas del Product Backlog; y, por otro lado, se decide cómo se va a hacer, para lo que se definen las tareas necesarias para poder completar cada ítem elegido del Product Backlog.

SCRUM DIARIO:

Reunión fija dentro del sprint que tiene como máximo 15 minutos de duración para ver cómo va el estado del proyecto. El objetivo es que los empleados estén

informados del trabajo de los otros, cuáles son los problemas que van surgiendo o que prevén encontrar y cómo lo resolverán.



Ilustración 9. Scrum Diario

La reunión se hace siempre en el mismo sitio y a la misma hora. Los perfiles que tienen que acudir a esta reunión de forma obligatoria son los desarrolladores y el Scrum Master, mientras que el Product Owner puede asistir de forma opcional. Si falta algún empleado de asistencia obligatoria, se pospone la reunión.

Si se quisiera ampliar algún tema deberá hacerse tras el scrum de ese día, sin interrumpir la dinámica. Además, en esta reunión se hacen tres preguntas:

- 🚦 ¿Qué hice ayer?
- 🚦 ¿Qué voy a hacer hoy?
- 🚦 ¿Hay algún impedimento que necesito que me ayuden a solucionar?

REVISIÓN DE SPRINT:

Cuando finaliza el sprint, el equipo lo revisa para determinar el valor que se le va a entregar al cliente. Por lo tanto, en esta revisión se muestran los trabajos finalizados y su duración no debería ser mayor a 4 horas para un sprint de 1 mes y es la única a la que puede asistir el cliente. Esta reunión es la única a la que puede asistir el cliente. El Product Owner hace una presentación de lo desarrollado hasta el momento y el equipo de desarrolladores muestra el funcionamiento. De esta manera, el cliente puede validar los cambios realizados y dar su opinión sobre nuevas tareas que el Product Owner tendrá que añadir al Product Backlog.

RETROSPECTIVA DEL SPRINT:

Después de la revisión del sprint, se hace una retrospectiva como último evento de scrum para que los miembros del equipo comuniquen sus impresiones sobre el sprint recién superado con el objetivo de mejorar la implementación del scrum. La reunión tiene un tiempo fijo de 3 horas para sprints de un mes. Este evento le sirve de ayuda al equipo de desarrolladores para inspeccionarse a sí mismo y proponer mejoras para el siguiente sprint. Cuando acaba la retrospectiva comienza un nuevo sprint, por lo que las mejoras acordadas se aplican al día siguiente.

2.2. Herramientas de desarrollo

2.2.1. ¿Qué es Node?

Node.js®, Node.js, es un entorno en tiempo de ejecución multiplataforma para la capa del servidor (en el lado del servidor) basado en JavaScript. Node.js es un entorno controlado por eventos diseñado para crear aplicaciones escalables, permitiéndote establecer y gestionar múltiples conexiones al mismo tiempo. Gracias a esta característica, no tienes que preocuparte con el bloqueo de procesos, pues no hay bloqueos.

¿Cómo funciona Node.js?

El diseño de Node.js está inspirado en sistemas como el Event Machine de Ruby o el Twisted de Python. Sin embargo, Node.js presenta un bucle de eventos como una construcción en tiempo de ejecución en lugar de una biblioteca. Este bucle de eventos es invisible para el usuario.

Características principales de Node.js:

- ✚ Velocidad. Node.js está construido sobre el motor de JavaScript V8 de Google Chrome, por eso su biblioteca es muy rápida en la ejecución de código.
- ✚ Sin búfer. Las aplicaciones de Node.js generan los datos en trozos (chunks), nunca los almacenan en búfer.
- ✚ Asíncrono y controlado por eventos. Como hemos dicho anteriormente, las APIs de la biblioteca de Node.js son asíncronas, sin bloqueo. Un servidor basado en Node.js no espera que una API devuelva datos. El servidor pasa a la siguiente API después de llamarla, y un mecanismo de notificación de eventos ayuda al servidor a obtener una respuesta de la llamada a la API anterior.
- ✚ Un subproceso escalable. Node.js utiliza un modelo de un solo subproceso con bucle de eventos. Gracias al mecanismo de eventos, el servidor responde sin bloqueos, como hemos dicho. Esto hace que el servidor sea altamente escalable comparando con los servidores tradicionales como el Servidor HTTP de Apache.

¿Para qué sirve Node.js?

Puedes utilizar Node.js para diferentes tipos de aplicaciones. Los siguientes son algunos de los ejemplos:

- ✚ Aplicaciones de transmisión de datos (streaming)
- ✚ Aplicaciones intensivas de datos en tiempo real
- ✚ Aplicaciones vinculadas a E/S

- Aplicaciones basadas en JSON: API
- Aplicaciones de página única

¿Quién usa Node.js?

Cómo debes imaginar, casi todas las marcas más importantes del mercado utilizan Node.js. A continuación, tienes algunos ejemplos:

- GoDaddy
- Microsoft
- eBay
- General Electric
- PayPal
- Uber
- NASA
- Netflix
- LinkedIn
- Medium



Ilustración 10. Node.js

2.2.2. ¿Qué es React?

React es una librería para crear interfaces de usuarios con código abierto, es decir, que está a disposición de cualquier programador para que use sus recursos e incluso haga colaboraciones. Esta biblioteca fue lanzada en el año 2013, por los desarrolladores de Facebook, con base en el lenguaje JavaScript.

¿Para qué sirve React?

La función principal de React es desarrollar páginas web de una manera gratuita y sencilla gracias a sus componentes reutilizables. Estos hacen posible usar un mismo elemento en varias partes del sitio o en otros sitios sin necesidad de volver a escribir todo el código. Como resultado, los programadores web ahorran gran cantidad de tiempo y trabajo.

React cuenta con un óptimo desempeño que se encarga de actualizar y renderizar los cambios realizados de forma automática. Esta cualidad permite a los programadores desarrollar sus códigos sin mayores contratiempos en el modelo en objetos para la representación de documentos (DOM).



Características de React:

1. Lenguaje JSX

La sintaxis que usa React es el JavaScript XML (JSX), el cual es una combinación del lenguaje HTML y el JavaScript, por lo que también se considera una extensión. Así que la relación que hay entre estos es bastante estrecha. Algunos programadores han afirmado que solamente se diferencian por detalles y que puede coincidir en más de un 90 % el código del JSX con el de JavaScript. Así que las personas que ya dominen este lenguaje se adaptarán muy fácil a React.



2. Componentes

Regularmente dentro de la programación web se manejan tres carpetas: la de la estructura de la página web (HTML), la de la presentación de la interfaz (CSS) y la del comportamiento de la misma (JS). Por su lado, React decidió juntar todas estas en un mismo lugar, de manera dinámica, para dar lugar a los componentes. Estos son partes de la interfaz del usuario que es independiente y que se puede reutilizar en otro lugar del sitio web. Esta es una de las características más sobresalientes. Asimismo, existen dos tipos de componentes:

-  Componentes funcionales. Son los más usados: representan una función de JavaScript, retornan a un elemento JSX de React, comienzan con mayúscula y pueden recibir valores.
-  Componentes de clase. Se refiere a los componentes que están escritos en JavaScript moderno de una manera más elaborada: retornan a un elemento JSX a través de un método render y también es posible que se les asigne valores.

3. Ciclos de vida

Dentro de React los componentes pasan por una serie de etapas durante su creación. Regularmente se divide en tres fases esenciales: montaje, utilización y desmontaje. A través de ellas suceden los siguientes métodos que se ejecutan para programar una interfaz de usuario.

-  Constructor. Es donde se crea el estado de un componente, enlaza eventos y setea variables globales.
-  ComponentWillMount. Se ejecuta antes que el componente sea montado, permite modificarlo y limpia la demanda de dependencias. No hace llamados a API ni suscripción a eventos.

- ✚ Render. Aquí se toman las propiedades, el estado y el contexto. Es cuando ya se empieza a mostrar la imagen de la interfaz y refleja los cambios realizados en ella.
- ✚ componentDidMount. Este método se ejecuta cuando el componente ya se ha renderizado; es ideal para llamar a API e iniciar la carga remota de datos, enlazar eventos y modificar el estado.
- ✚ Actualización. Se puede repetir varias veces para realizar cambios en las propiedades de los estados, renderizar los componentes y preparar actualización del diseño de interfaz de usuario (UI).
- ✚ Desmontar. Se aplica con el método componentWillUnmount para dejar de escuchar eventos, quitarse de un WebSocket y cancelar peticiones HTTP. Es una especie de limpieza.
- ✚ componentDidCatch. También conocido como control de errores, se activa cuando detecta equivocaciones al montar un componente.

4. Virtual DOM

El virtual DOM es una característica de React que permite que el rendimiento de la página web no se vea afectada por las actualizaciones y que estas, a su vez, se mantengan bien reflejadas en el modelo en objetos para la representación de documentos (DOM). Normalmente, para que los cambios realizados en los estados de una página web se vean en la interfaz gráfica es necesario un proceso complejo, largo y costoso.

Requerimientos de React

1. Tener conocimientos de JavaScript y HTML

Como ya hemos mencionado React tiene su propio lenguaje: el JSX, el cual es una extensión de JavaScript. Así que para familiarizarse rápidamente con esta librería de interfaces de usuario lo más recomendable es tener conocimientos previos de la sintaxis de JavaScript o HTML.

2. Contar con un sistema operativo y navegador moderno

Uno de los requisitos básicos e indispensables es que tu ordenador o laptop cuente con el sistema operativo más reciente, ya sea Windows, macOS o Linux. Lo mismo aplica para los navegadores web. Se recomienda usar los más actualizados; en este caso Firefox, Google Chrome o Safari.

3. Iniciar con Create-react-app

La misma librería cuenta con su propia aplicación llamada Create-react-app para que puedas practicar con el código con el que trabaja y aprendas a configurar el ambiente para desarrollar la aplicación de tu página web. Es importante aclarar que esta no se encarga de la base de datos, solo hace posible el flujo de construcción

para el frontend. Para su correcto funcionamiento es necesario instalar primero Node.js.

4. Instalar un administrador de paquetes

Un gestor o administrador de paquetes te brinda la posibilidad de usar los paquetes que otros programadores ya crearon, por lo que también puedes instalarlo de manera sencilla. Un par de opciones son Npm o Yarn.

5. Adicionar un empaquetador y un compilador

La función de un software empaquetador es poder escribir en código modular para guardarlo en paquetes más pequeños que permitirán que el tiempo de carga se reduzca. Algunos buenos ejemplos son Parcel, Webpack o Neutrino. Por otro lado, un compilador facilita escribir en JavaScript moderno, el cual funciona en los navegadores pasados; una buena opción es Babel.

6. Usar un framework de renderizado en servidor

Es necesario que cuentes con un programa que te permita escribir, mantener y escalar el código de tu página web. Para ello están los frameworks de renderizado en servidor. Algunas buenas opciones son el famoso Next.js o el también eficaz Razzle, que no necesita configuración y es muy flexible.

7. Dar paridad desde Windows

Solo en el caso de que estés trabajando con un sistema Windows es necesario que uses un software que te ayude a dar paridad o evitar errores de transmisión con la terminal de macOS. Algunos de los programas mejor calificados para esta tarea son Gitbash o el Subsistema de Windows para Linux.

8. Probar Gatsby

Ahora, si tu sitio web está enfocado en el contenido estático, la mejor forma de crearlo es con Gatsby, al mismo tiempo que usas los componentes de React; te genera el código CSS y HTML para agilizar el tiempo de carga.

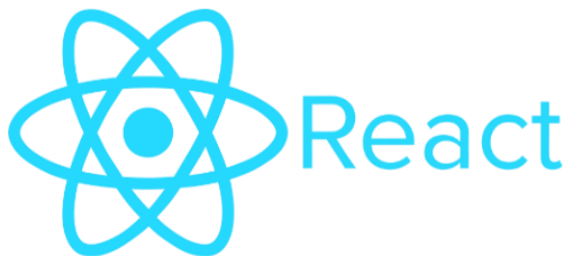


Ilustración 11.React

2.2.3. ¿Qué es MySQL?

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun Microsystems en 2008 y está su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL. MySQL es un sistema de gestión de bases de datos que cuenta con una doble licencia. Por una parte, es de código abierto, pero por otra, cuenta con una versión comercial gestionada por la compañía Oracle.

Características de MySQL:

MySQL presenta algunas ventajas que lo hacen muy interesante para los desarrolladores

- ✚ Arquitectura Cliente y Servidor: MySQL basa su funcionamiento en un modelo cliente y servidor. Es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros, por ejemplo.
- ✚ Compatibilidad con SQL: SQL es un lenguaje generalizado dentro de la industria. Al ser un estándar MySQL ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a MySQL.
- ✚ Vistas: Desde la versión 5.0 de MySQL se ofrece compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos SQL.
- ✚ Procedimientos almacenados. MySQL posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.
- ✚ Desencadenantes. MySQL permite además poder automatizar ciertas tareas dentro de nuestra base de datos.
- ✚ Transacciones. Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas.



Ilustración 12. MySQL

Capítulo 3.

Desarrollo

3.1 Desarrollo

A continuación, se mostrará el proceso que se llevó a cabo para realizar el proyecto siguiendo la metodología señalada la cual es SCRUM.

3.1.1 Requisitos del sistema

Hardware

- Servidor Appserv
- Navegador web Chrome
- 12 GB mínimo de almacenamiento
- Procesador mínimo Intel Core i3
- 8 GB mínimo de RAM

Software

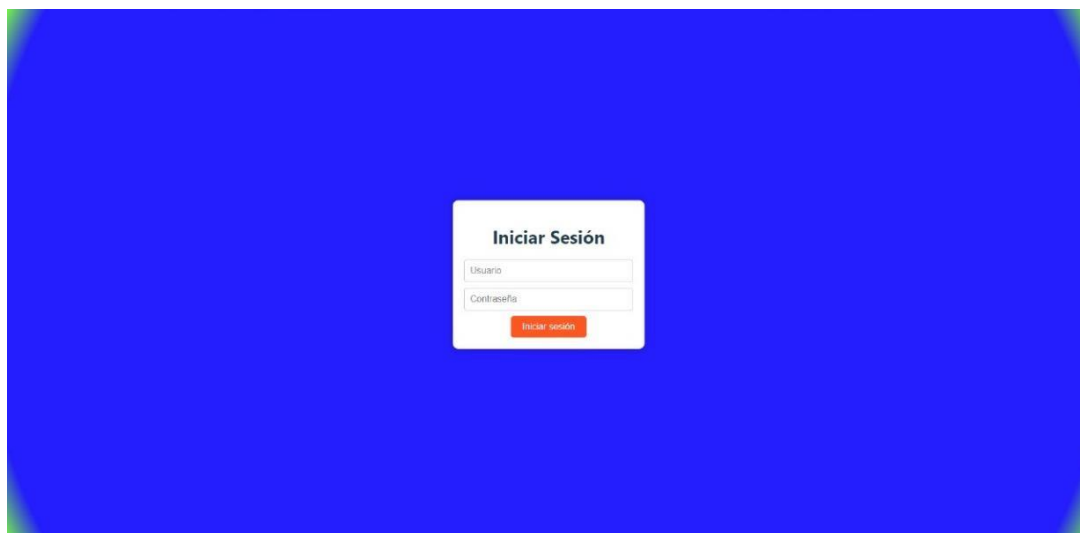
- Instalación del editor de código Visual Studio Code.
- Instalación de React.
- Sistema operativo Windows 8,8.1,10,11
- Entorno de ejecución Node.js

Sistema

- Debe guardar registros de datos de los clientes, siniestros y refacciones
- Debe guardar registros de los clientes los siniestros y las refacciones que va a ir almacenado en una base de datos
- Tener la disponibilidad para brindar apoyo de servicio y soporte a la empresa
- Eficiencia a la hora de buscar registros en la base
- Seguridad y privacidad al guardar los datos
- Debe utilizar los recursos disponibles (como hardware, software, tiempo y personal) de manera eficiente para evitar desperdicios y costos innecesarios.

3.2 Diseño del sistema

Se realizó la interfaz gráfica del programa de acuerdo a los campos necesarios para llenar (captura 1).



captura 1

Posteriormente se realizó esta interfaz que tiene la función de un menú en el cual se podrá desplazar de una sección a otra (captura 2).

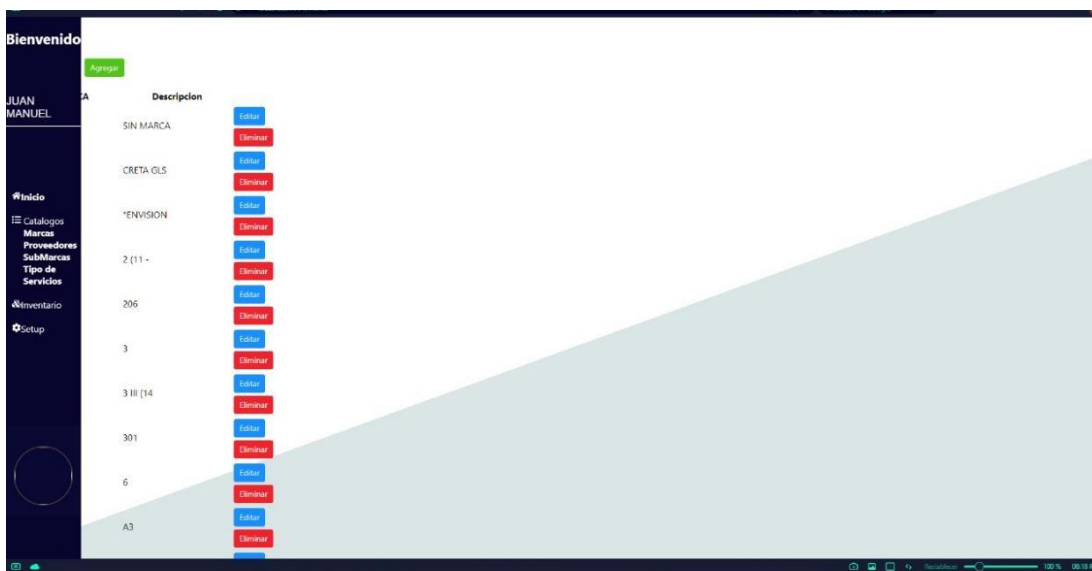


captura 2

En el menú tenemos estas secciones que nos desglosan todas las sub secciones que tiene aquí como lo es catalogo que contiene Marcas, proveedores, submarcas y tipo de servicio (captura 3,4,5,6,7).



captura 3



captura 4

Bienvenido vendedores

[Agregar](#)

Acción Social	Nombre Comercial	Servicio	Entidad	Marca	Calle	Colonia	Poblacion	Estado	Telefono	Codigo Postal	Celular	Correo	Rfc	Cia
JUAN MANUEL	ONIO ZALEZ S	TALLER	CIUDAD DE MEXICO		CAM A DESIERTO DE SANTA ROSA LOS LEONES S/N	XOCHIALC	CUAJIMALPA	ESTADO DE MEXICO		9000			GORA-580705-XXX	CAF
	AS SCANIA DE C.V.	GRUAS SCANIA	GRUAS	GUANAJUATO	CARR. PANAMERICANA KM SILAO 63	SILAO, GTO.	GUANAJUATO		0147-11-58-61	0			GSC-800604-452	CAF
	AZAR DOSO BANDO	AUTO CLUB	TALLER	GUANAJUATO	5 DE FEBRERO NO. 717	ZONA CENTRO	LEON	GUANAJUATO	(477) 7138289	37000		autoclubleon@prodigy.net.mx	SACF-340507-LA4	CAF
	MO MOTRIZ DE C.V.	FERRO AUT. (HUIPULCO)(CABIO DE RS CLAVE ACT.25285)	TALLER	CIUDAD DE MEXICO	CALZADA MEXICO XOCHIMILCO, #61	HUIPULCO.	TLALPAN	CIUDAD DE MEXICO	(55) 55133141/730022	14370	(55) 50149004 /105	ferro@prodigy.net.mx;ferro.tlalpan@ferromotriz.com	FAU-911022-HX6	CAF

[Inicio](#)
[Catalogos](#)
[Inventario](#)
[Setup](#)

captura 5

Bienvenido SubMarca

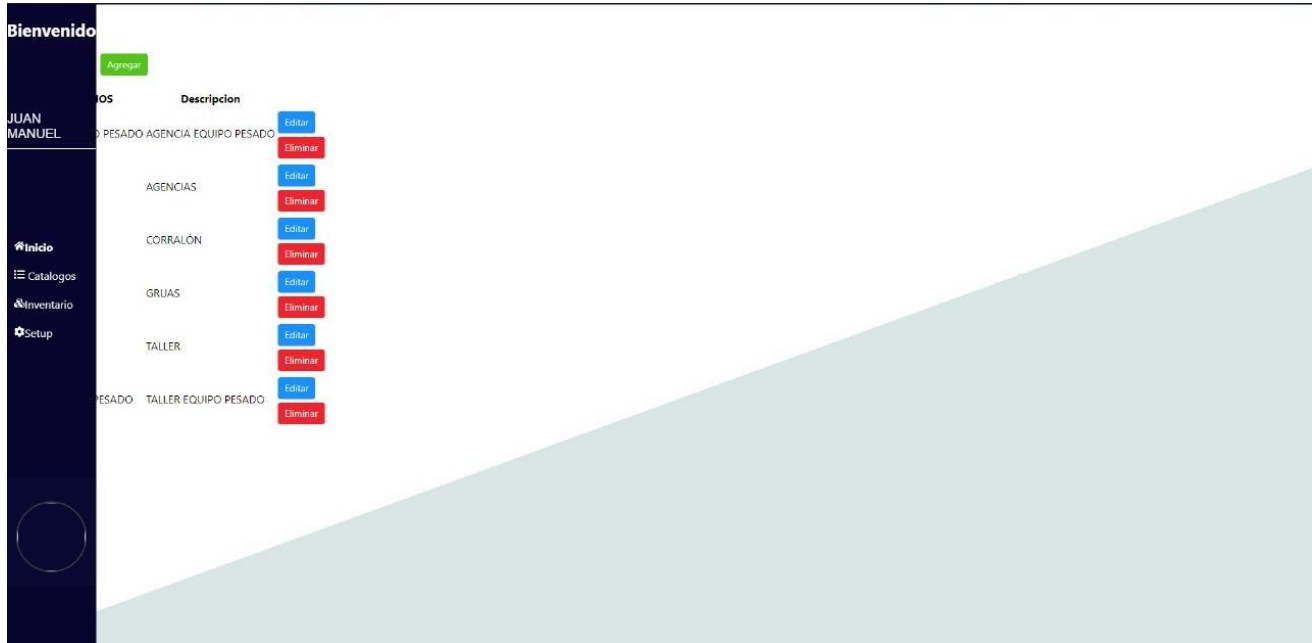
[Agregar](#)

ID SUBMARCA	MARCA	Acciones
	JEEP COMPA	Editar Eliminar
	CRETA GLE	Editar Eliminar
	FATT 110 R	Editar Eliminar
	YCHIVAN	Editar Eliminar

[Inicio](#)
[Catalogos](#)
[Inventario](#)
[Setup](#)

Página 1 de 100 Registros por página: 1 Mostrando 1 a 4 de 398 elementos

captura 6



captura 7

Ahora en la sección de inventario nos dará la información de todas las piezas disponibles de los siniestros (captura 8).

The screenshot shows the 'Lista de Inventario' section. It includes a search bar, a 'Cargar Lote' button, and a detailed table of inventory items. The table has columns for 'ID Expediente', 'Id lote', 'cve proveedor', 'del', 'al', 'Numero', 'Año', 'Reporte', 'Siniestro', 'Oficina', 'Fecha Valuación', 'Marca', 'Tipo', 'Modelo', 'Color', 'Placas', 'Serie', 'Cia', and 'Acciones'. The 'Acciones' column contains 'Eliminar Registro' and 'Expandir Registro' buttons. Below the table is a pagination bar showing 'Página 1 de 2' and 'Registros por página: 5'. The status bar at the bottom indicates 'Mostrando 1 a 4 de 5 elementos'.

ID Expediente	Id lote	cve proveedor	del	al	Numero	Año	Reporte	Siniestro	Oficina	Fecha Valuación	Marca	Tipo	Modelo	Color	Placas	Serie	Cia	Acciones
8219	1	26399			0	17	1400572	1288127	EDOMEX	2043-04-05T06:00:00.000Z	PEUGEOT	307	2017	GRIS	52AKUJ	VF3DCNFFBHU509827	QUALITAS	Eliminar Registro Expandir Registro
8344	2	26396			0	17	1400572	1288128	EDOMEX	2043-04-05T06:00:00.000Z	VW	JETTA	2017	ROJO	VUT1245	VF3DCNFFBHU509828	QUALITAS	Eliminar Registro Expandir Registro
8245	2	26396			0	17	1400572	1288127	EDOMEX	2043-04-05T06:00:00.000Z	PEUGEOT	307	2017	GRIS	53AKUJ	VF3DCNFFBHU509827	QUALITAS	Eliminar Registro Expandir Registro
8247	3	223			0	23	16007300	59347628	CDMX	2045-02-01T06:00:00.000Z	VW	VENTO	2018	ROJO	GR7954	VF3DCNFFBHU425671	ZURICH	Eliminar Registro Expandir Registro

captura 8

En esta parte del menú nos despliega varias sub secciones del apartado de setup el cual contiene: Cia, Estados, Menú, Oficinas, Perfil, Permisos, Sub menú, Tipo, Miembros y Usuarios (captura 9,10,11,12,13).



captura 9

Bienvenido Proveedores

[Agregar](#)

Razón Social	Nombre Comercial	Servicio	Entidad	Marca	Calle	Colonia	Poblacion	Estado	Telefono	Codigo Postal	Celular	Correo	Rfc	Cia
ONIO GONZALEZ AS.		TALLER	CIUDAD DE MEXICO		CAM A DESIERTO DE LOS LEONES S/N	SANTA ROSA XOCHIAC	CUAJIMALPA	ESTADO DE MEXICO		9000			GORA-580705-XXX	CAT
AS SCANIA S.A. C.V.	GRUAS SCANIA	GRUAS	GUANAJUATO		CARR. PANAMERICANA KM 63	SILAO	SILAO, GTO.	GUANAJUATO	0147-11-58-61	0			GSC-800804-452	CAT
HAZAR CARDOSO VANDQ	AUTO CLUB	TALLER	GUANAJUATO		5 DE FEBRERO NO. 717	ZONA CENTRO	LEON	GUANAJUATO	(477) 7138289	37000		autoclubleon@prodigy.net.mx	SACF-540507-LA4	CAT
RO AUTOMOTRIZ DE C.V.	FERRO AUT.(HUIPULCO) (CABIO DE RS CLAVE ACT.25283)	TALLER	CIUDAD DE MEXICO		CALZADA MEXICO XOCHIMILCO, #61	HUIPULCO	TALPAN	CIUDAD DE MEXICO	(55) 55133141/730022	14370	(55) 50149004 /105	ferro@prodigy.net.mx;ferro.talpan@ferromotriz.com	FAU-911022-HX6	CAT

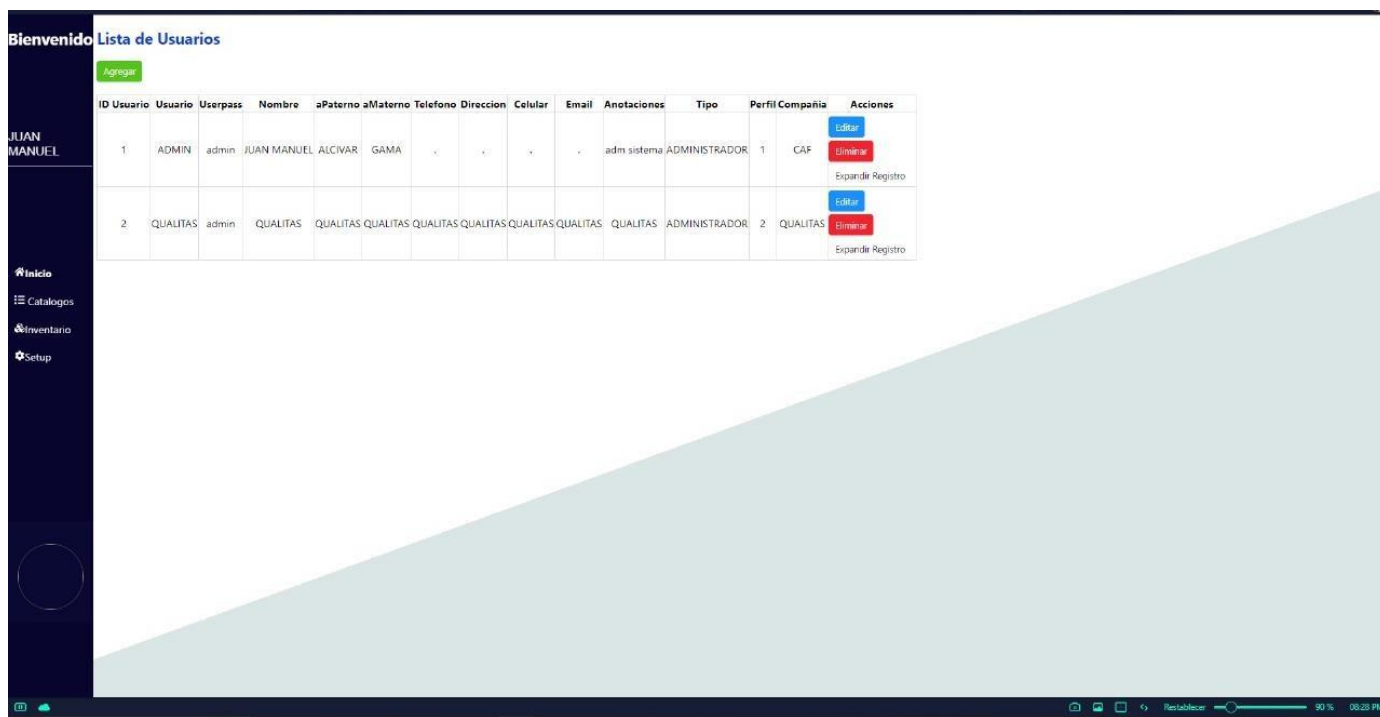
captura 10



captura 11



captura 12



captura 13

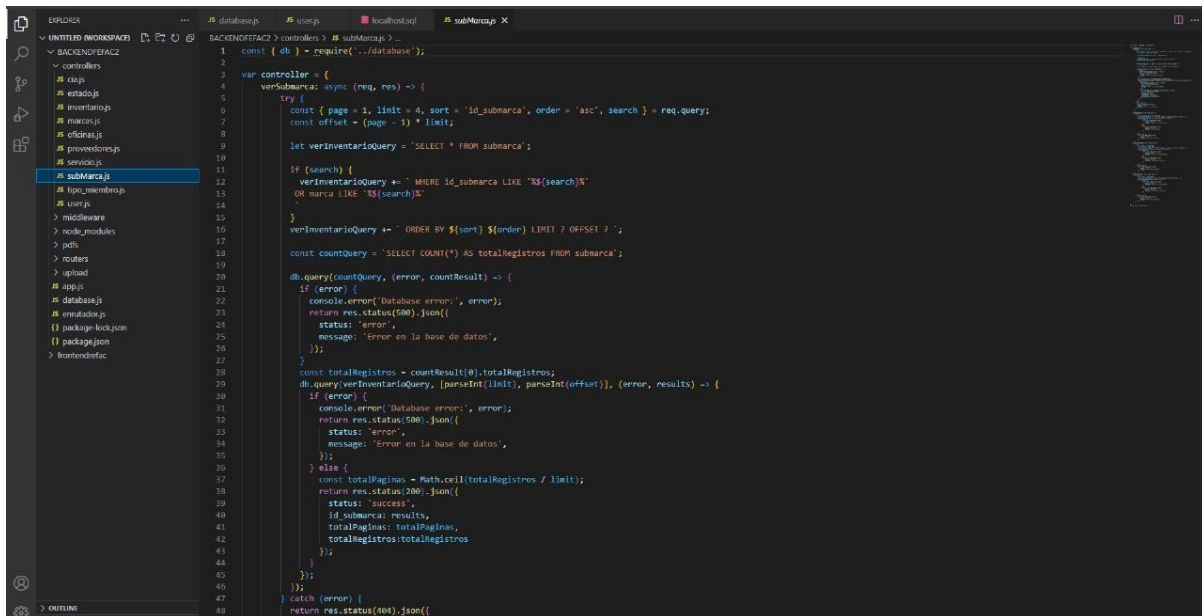
Por último, el apartado final es para poder cerrar sesión de la página como se muestra en la imagen (captura 14).



captura 14

3.3 Codificación

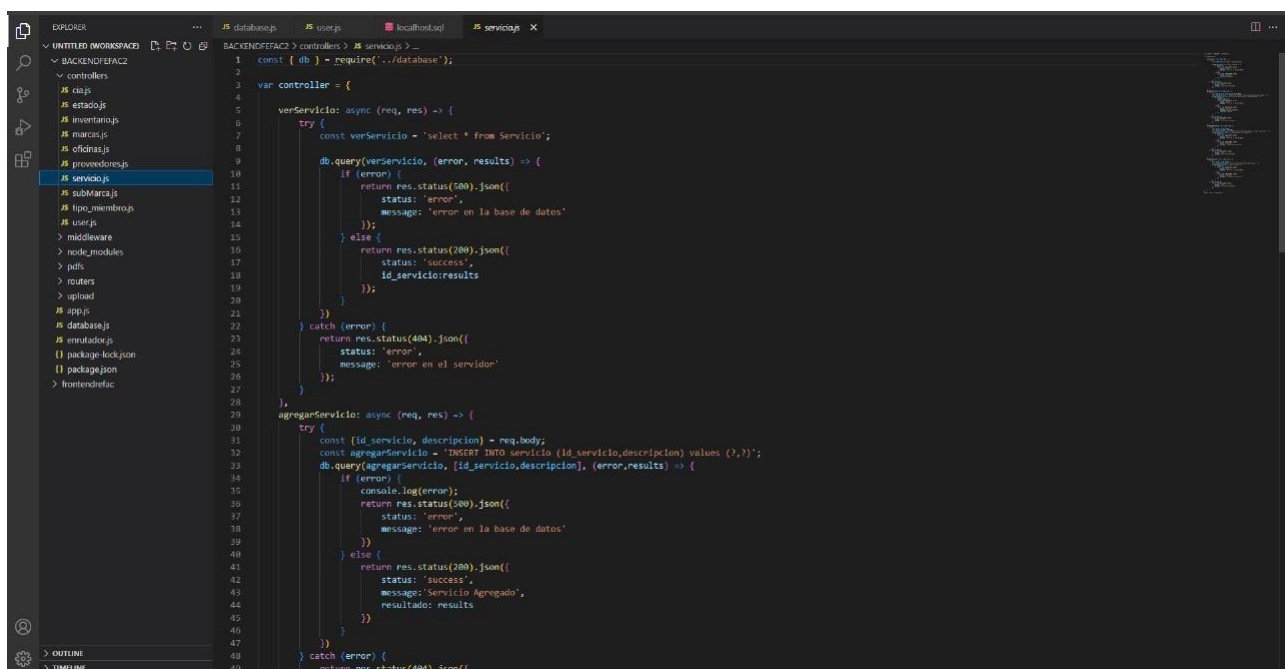
El siguiente código que se muestra es el que hizo posible que la sección de submarcas funcionara (captura 15).



```
1 const { db } = require('../database');
2
3 var controller = {
4   verSubmarcas: async (req, res) => {
5     try {
6       const { page = 1, limit = 4, sort = 'id_submarca', order = 'asc', search } = req.query;
7       const offset = (page - 1) * limit;
8       let verInventarioQuery = 'SELECT * FROM submarca';
9
10      if (search) {
11        verInventarioQuery += ' WHERE id_submarca LIKE "%'+search+'%"';
12        OR marca LIKE "%'+search+'%"';
13      }
14      verInventarioQuery += ' ORDER BY '+sort+' '+order+' LIMIT ? OFFSET ? ';
15      const countQuery = 'SELECT COUNT(*) AS totalregistros FROM submarca';
16
17      db.query(countQuery, (error, countResult) => {
18        if (error) {
19          console.error('Database error:', error);
20          return res.status(500).json({
21            status: 'error',
22            message: 'Error en la base de datos',
23          });
24        }
25        const totalRegistros = countResult[0].totalRegistros;
26        db.query(verInventarioQuery, [parseInt(limit), parseInt(offset)], (error, results) => {
27          if (error) {
28            console.error('Database error:', error);
29            return res.status(500).json({
30              status: 'error',
31              message: 'Error en la base de datos',
32            });
33          }
34          const totalPaginas = Math.ceil(totalRegistros / limit);
35          return res.status(200).json({
36            status: 'success',
37            id_submarca: results,
38            totalPaginas: totalPaginas,
39            totalRegistros: totalRegistros
40          });
41        });
42      });
43    } catch (error) {
44      return res.status(404).json({
45        status: 'error',
46        message: 'Error en el servidor'
47      });
48    }
49  }
50 }
```

captura 15

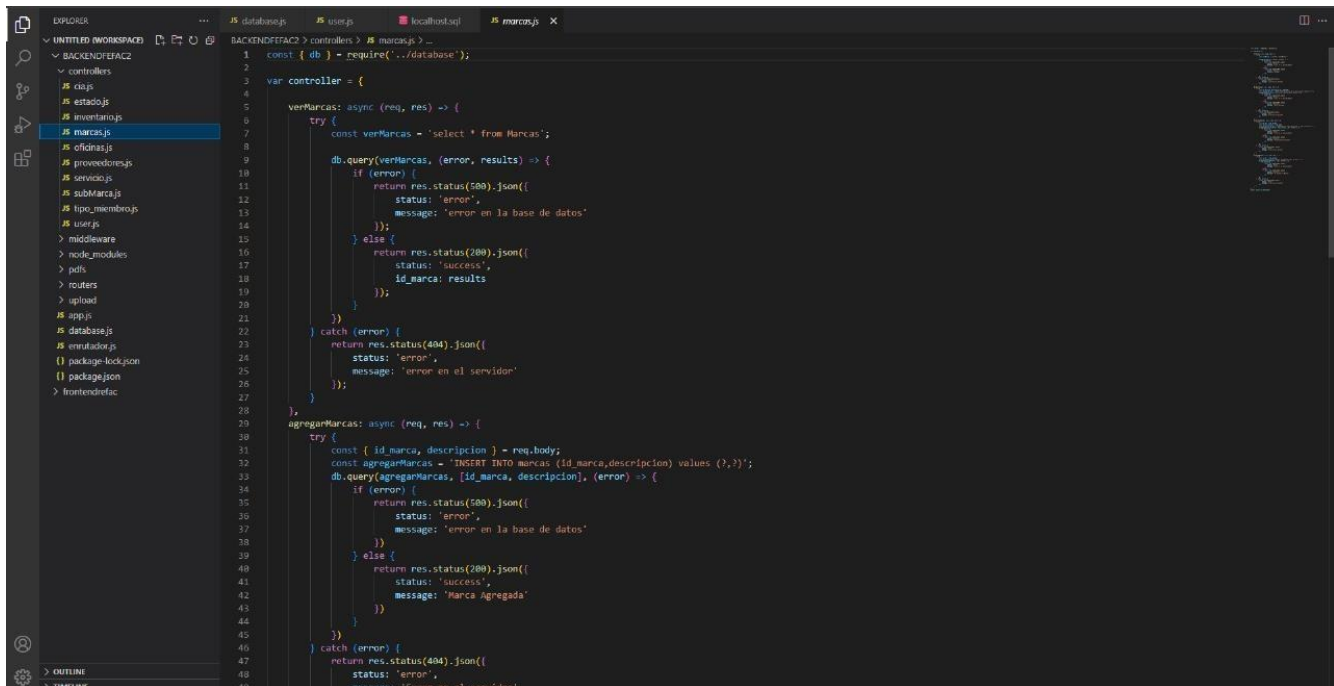
El código siguiente se utilizó para programar la sección de tipo de servicio para dar la información de los servicios que realiza la aseguradora (captura 16).



```
1 const { db } = require('../database');
2
3 var controller = {
4   verServicio: async (req, res) => {
5     try {
6       const verServicio = 'select * from servicio';
7
8       db.query(verServicio, (error, results) => {
9         if (error) {
10           return res.status(500).json({
11             status: 'error',
12             message: 'Error en la base de datos'
13           });
14         }
15         return res.status(200).json({
16           status: 'success',
17           id_servicio: results
18         });
19       });
20     } catch (error) {
21       return res.status(404).json({
22         status: 'error',
23         message: 'Error en el servidor'
24       });
25     }
26   },
27   agregarServicio: async (req, res) => {
28     try {
29       const { id_servicio, descripcion } = req.body;
30       const agregarServicio = 'INSERT INTO servicio (id_servicio,descripcion) values (?,?)';
31       db.query(agregarServicio, [id_servicio,descripcion], (error,results) => {
32         if (error) {
33           console.log(error);
34           return res.status(500).json({
35             status: 'error',
36             message: 'Error en la base de datos'
37           });
38         }
39         return res.status(200).json({
40           status: 'success',
41           message: 'Servicio Agregado',
42           resultado: results
43         });
44       });
45     } catch (error) {
46       return res.status(404).json({
47         status: 'error',
48         message: 'Error en el servidor'
49       });
50     }
51   }
52 }
```

captura 16

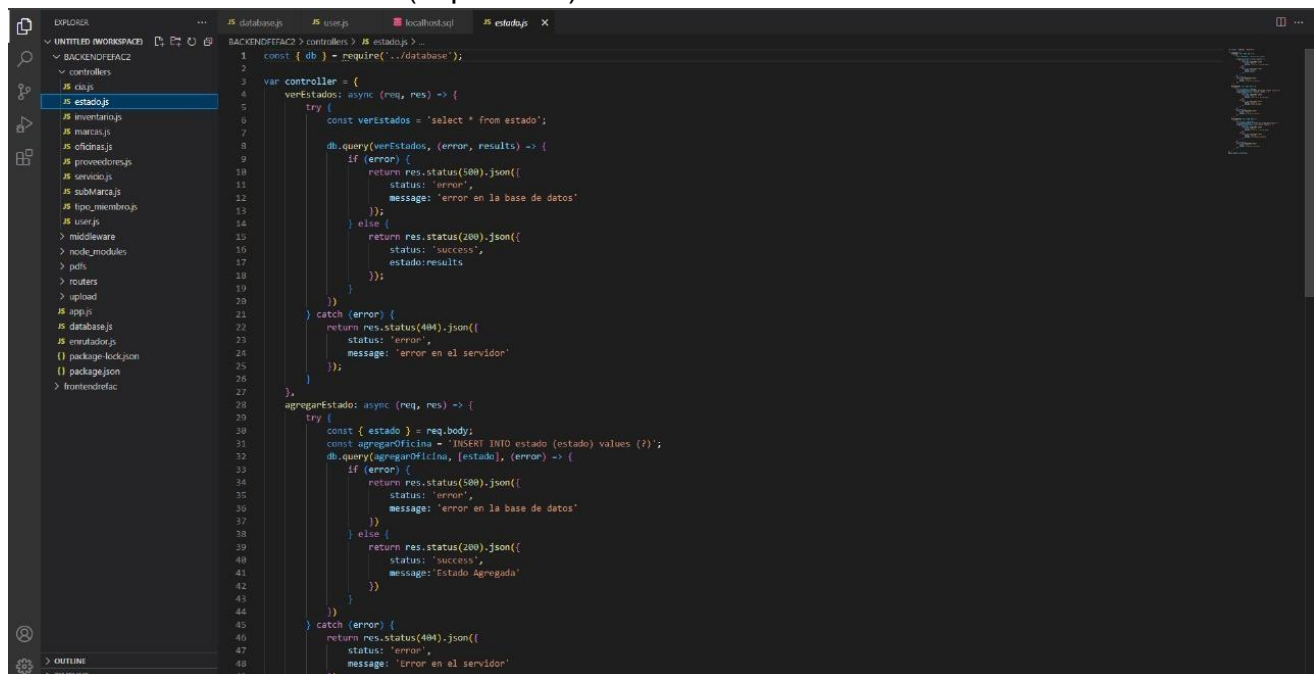
Este código se utilizó para crear la sección de marcas la cual nos muestra la información del auto del siniestro (captura 17).



```
1 const { db } = require('../database');
2
3 var controller = {
4
5   verMarcas: async (req, res) => {
6     try {
7       const verMarcas = 'select * from Marcas';
8
9       db.query(verMarcas, (error, results) => {
10         if (error) {
11           return res.status(500).json({
12             status: 'error',
13             message: 'error en la base de datos'
14           });
15         } else {
16           return res.status(200).json({
17             status: 'success',
18             id_marca: results
19           });
20         }
21       })
22     } catch (error) {
23       return res.status(404).json({
24         status: 'error',
25         message: 'error en el servidor'
26       });
27     }
28   },
29   agregarMarcas: async (req, res) => {
30     try {
31       const { id_marca, descripcion } = req.body;
32       const agregarMarcas = 'INSERT INTO marcas (id_marca,descripcion) values (?,?)';
33       db.query(agregarMarcas, [id_marca, descripcion], (error) => {
34         if (error) {
35           return res.status(500).json({
36             status: 'error',
37             message: 'error en la base de datos'
38           });
39         } else {
40           return res.status(200).json({
41             status: 'success',
42             message: 'Marca Agregada'
43           });
44         }
45       })
46     } catch (error) {
47       return res.status(404).json({
48         status: 'error',
49         message: 'Error en el servidor'
50       });
51     }
52   }
53 }
```

captura 17

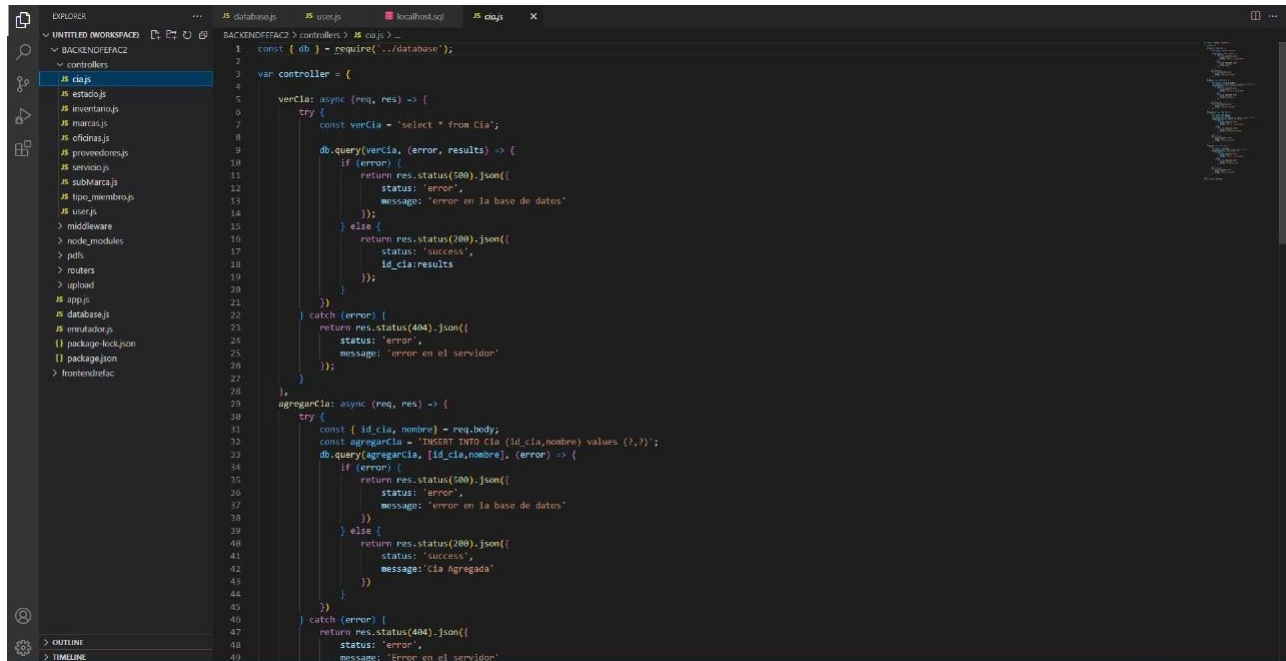
En este código nos dará la información de los estados del cual se recolecto la información del siniestro (captura 18).



```
1 const { db } = require('../database');
2
3 var controller = {
4
5   verEstados: async (req, res) => {
6     try {
7       const verEstados = 'select * from estado';
8
9       db.query(verEstados, (error, results) => {
10         if (error) {
11           return res.status(500).json({
12             status: 'error',
13             message: 'error en la base de datos'
14           });
15         } else {
16           return res.status(200).json({
17             status: 'success',
18             estado: results
19           });
20         }
21       })
22     } catch (error) {
23       return res.status(404).json({
24         status: 'error',
25         message: 'error en el servidor'
26       });
27     }
28   },
29   agregarEstado: async (req, res) => {
30     try {
31       const { estado } = req.body;
32       const agregarOficina = 'INSERT INTO estado (estado) values (?)';
33       db.query(agregarOficina, [estado], (error) => {
34         if (error) {
35           return res.status(500).json({
36             status: 'error',
37             message: 'error en la base de datos'
38           });
39         } else {
40           return res.status(200).json({
41             status: 'success',
42             message: 'Estado Agregada'
43           });
44         }
45       })
46     } catch (error) {
47       return res.status(404).json({
48         status: 'error',
49         message: 'Error en el servidor'
50       });
51     }
52   }
53 }
```

captura 18

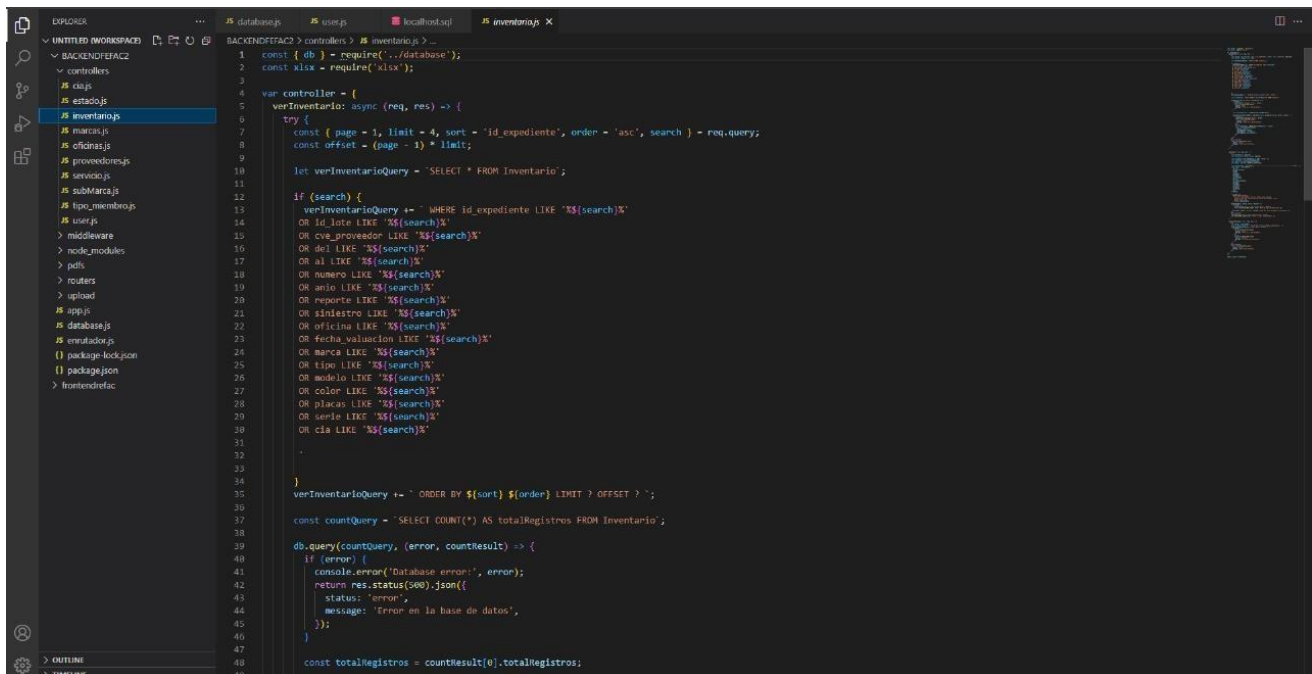
En el código de la sección de cia nos accedemos a las empresas afiliadas a la aseguradora (captura 19).



```
1 const { db } = require('../database');
2
3 var controller = {
4
5   verCia: async (req, res) => {
6     try {
7       const verCia = 'select * from Cia';
8       db.query(verCia, (error, results) => {
9         if (error) {
10           return res.status(500).json({
11             status: 'error',
12             message: 'error en la base de datos'
13           });
14         } else {
15           return res.status(200).json({
16             status: 'success',
17             id_cia: results
18           });
19         }
20       });
21     } catch (error) {
22       return res.status(404).json({
23         status: 'error',
24         message: 'error en el servidor'
25       });
26     }
27   },
28   agregarCia: async (req, res) => {
29     try {
30       const { id_cia, nombre } = req.body;
31       const agregarCia = 'INSERT INTO cia (id_cia,nombre) values (?,?)';
32       db.query(agregarCia, [id_cia,nombre], (error) => {
33         if (error) {
34           return res.status(500).json({
35             status: 'error',
36             message: 'error en la base de datos'
37           });
38         } else {
39           return res.status(200).json({
40             status: 'success',
41             message: 'Cia Agregada'
42           });
43         }
44       });
45     } catch (error) {
46       return res.status(404).json({
47         status: 'error',
48         message: 'Error en el servidor'
49       });
50     }
51   }
52 }
```

captura 19

Por último, en la sección de inventario como su nombre indica tenemos acceso a las piezas disponibles según su color, modelo, etc. (captura 20).



```
1 const { db } = require('../database');
2 const mongoose = require('mongoose');
3
4 var controller = {
5
6   verInventario: async (req, res) => {
7     try {
8       const { page = 1, limit = 4, sort = 'id_expediente', order = 'asc', search } = req.query;
9       const offset = (page - 1) * limit;
10
11       let verInventarioQuery = 'SELECT * FROM Inventario';
12
13       if (search) {
14         verInventarioQuery += ' WHERE id_expediente LIKE "%${search}%"';
15         OR id_lote LIKE "%${search}%"
16         OR cve_proveedor LIKE "%${search}%"
17         OR del LIKE "%${search}%"
18         OR al LIKE "%${search}%"
19         OR numero LIKE "%${search}%"
20         OR anio LIKE "%${search}%"
21         OR reporte LIKE "%${search}%"
22         OR siniestro LIKE "%${search}%"
23         OR oficina LIKE "%${search}%"
24         OR fecha_valuacion LIKE "%${search}%"
25         OR marca LIKE "%${search}%"
26         OR tipo LIKE "%${search}%"
27         OR modelo LIKE "%${search}%"
28         OR color LIKE "%${search}%"
29         OR placa LIKE "%${search}%"
30         OR serie LIKE "%${search}%"
31         OR cia LIKE "%${search}%"
32       }
33
34       verInventarioQuery += ' ORDER BY ${sort} ${order} LIMIT ? OFFSET ? ';
35
36       const countQuery = 'SELECT COUNT(*) AS totalRegistros FROM Inventario';
37
38       db.query(countQuery, (error, countResult) => {
39         if (error) {
40           console.error('Database error:', error);
41           return res.status(500).json({
42             status: 'error',
43             message: 'Error en la base de datos'
44           });
45         }
46       });
47
48       const totalRegistros = countResult[0].totalRegistros;
49
50     } catch (error) {
51       return res.status(404).json({
52         status: 'error',
53         message: 'Error en el servidor'
54       });
55     }
56   }
57 }
```

captura 20

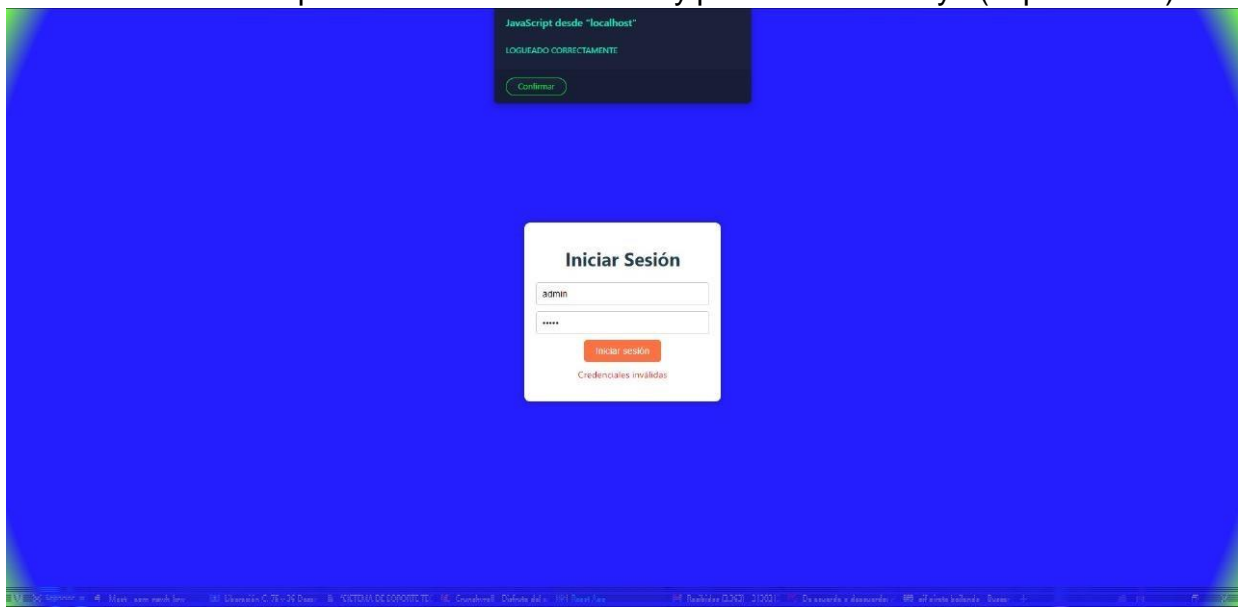
3.4 Prueba

Ahora en el log in para poder iniciar sesión es importante colocar los datos válidos para el ingreso de lo contrario al estar equivocados mostrara el mensaje que se muestra en la parte inferior del recuadro del log in (captura 21).



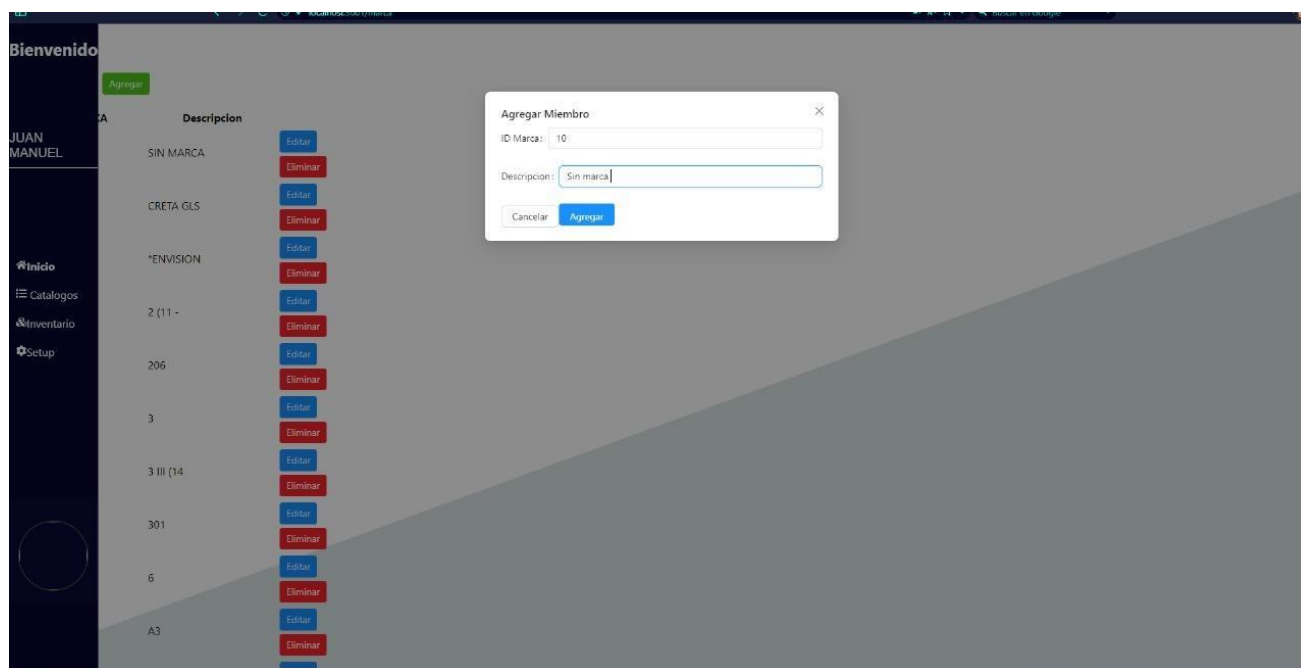
captura 21

Al introducir los datos correctos del log in nos mostrara esta ventana emergente la cual nos da el aviso que los datos son verídicos y podremos entrar ya (captura 21.2).



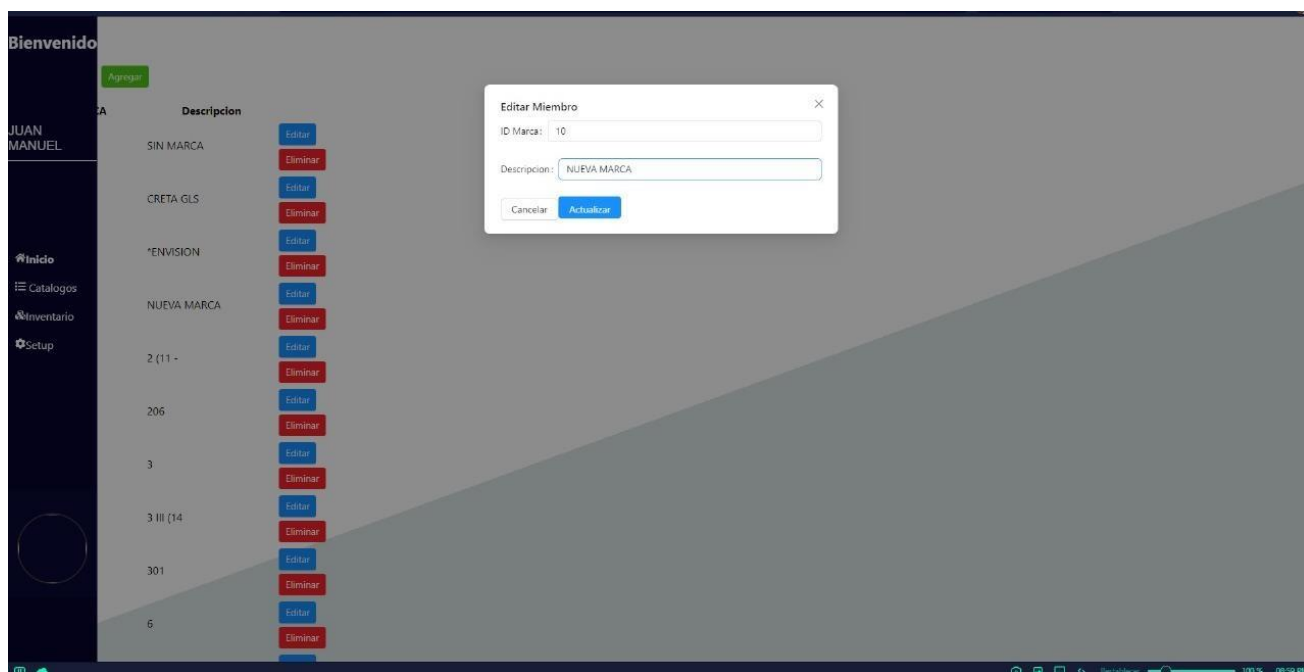
captura 21.2

Para agregar los datos nos arroja esta ventana en la cual podremos realizar el nuevo registro a "marcas" del catálogo (captura 22).



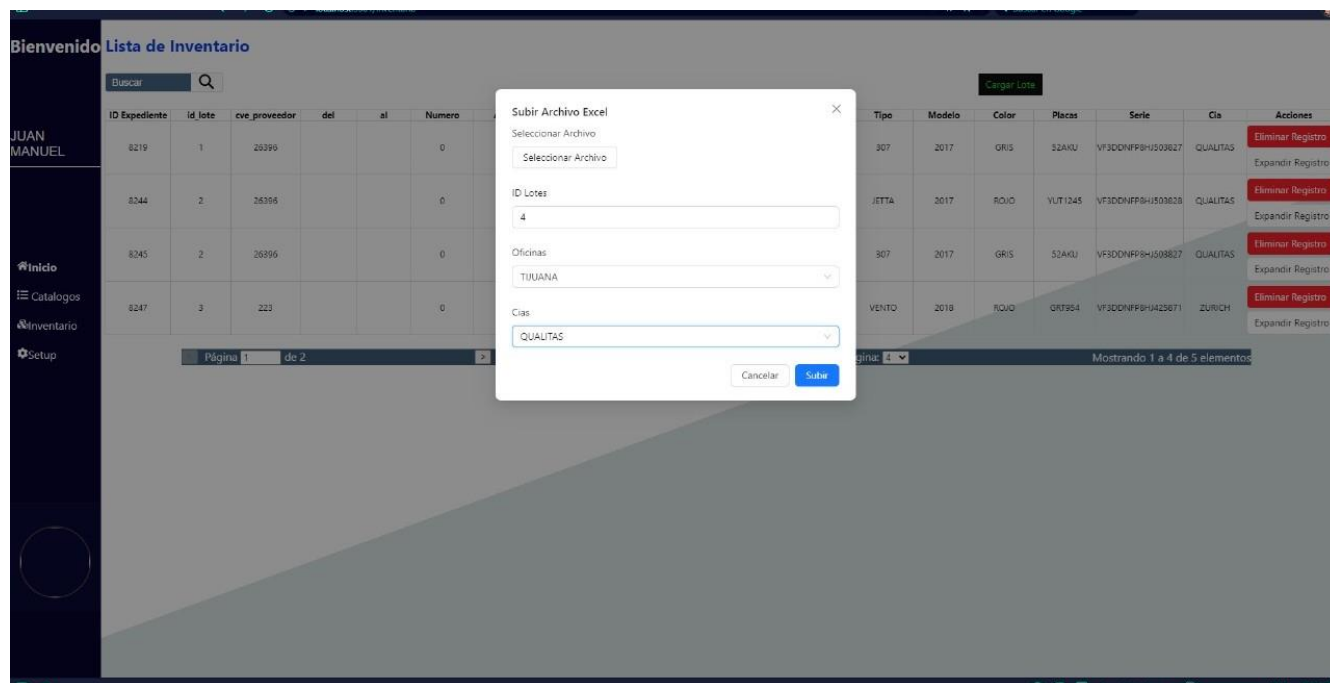
captura 22

Esta ventana nos sirve para editar un registro de "marca" (captura 23).



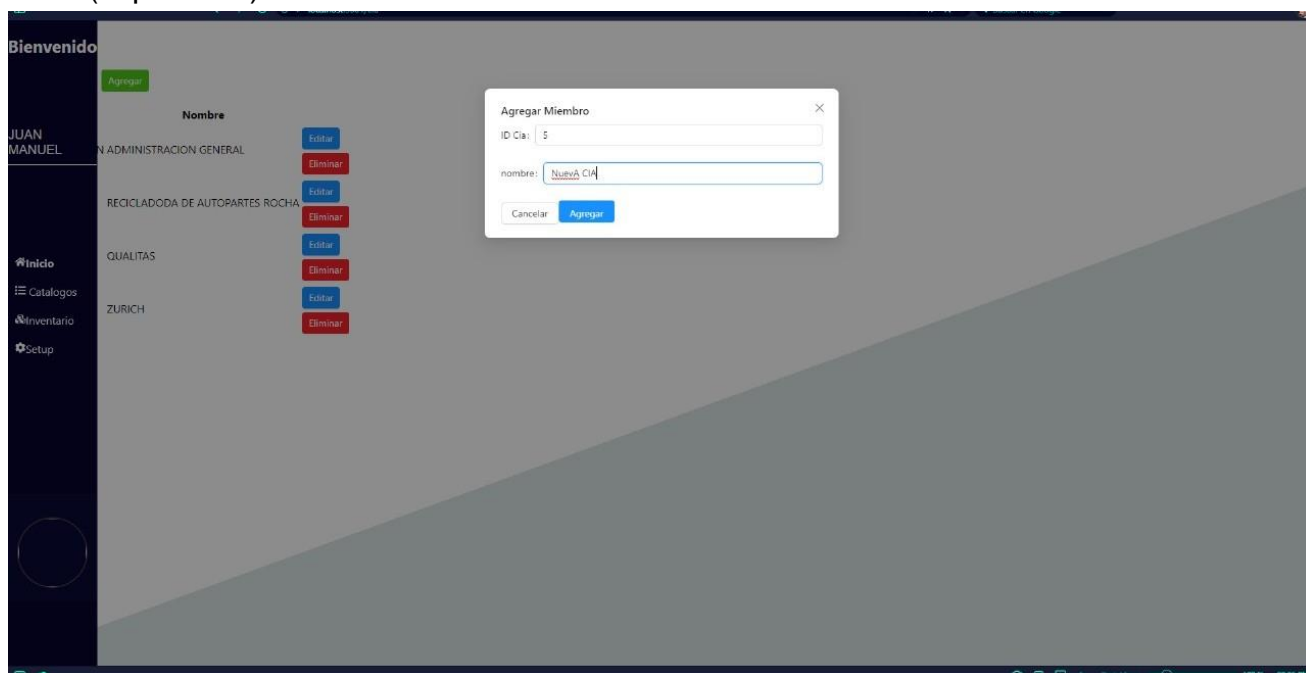
captura 23

En la siguiente ventana en el apartado de inventario nos da la opción de subir un nuevo lote de refacción (captura 24).



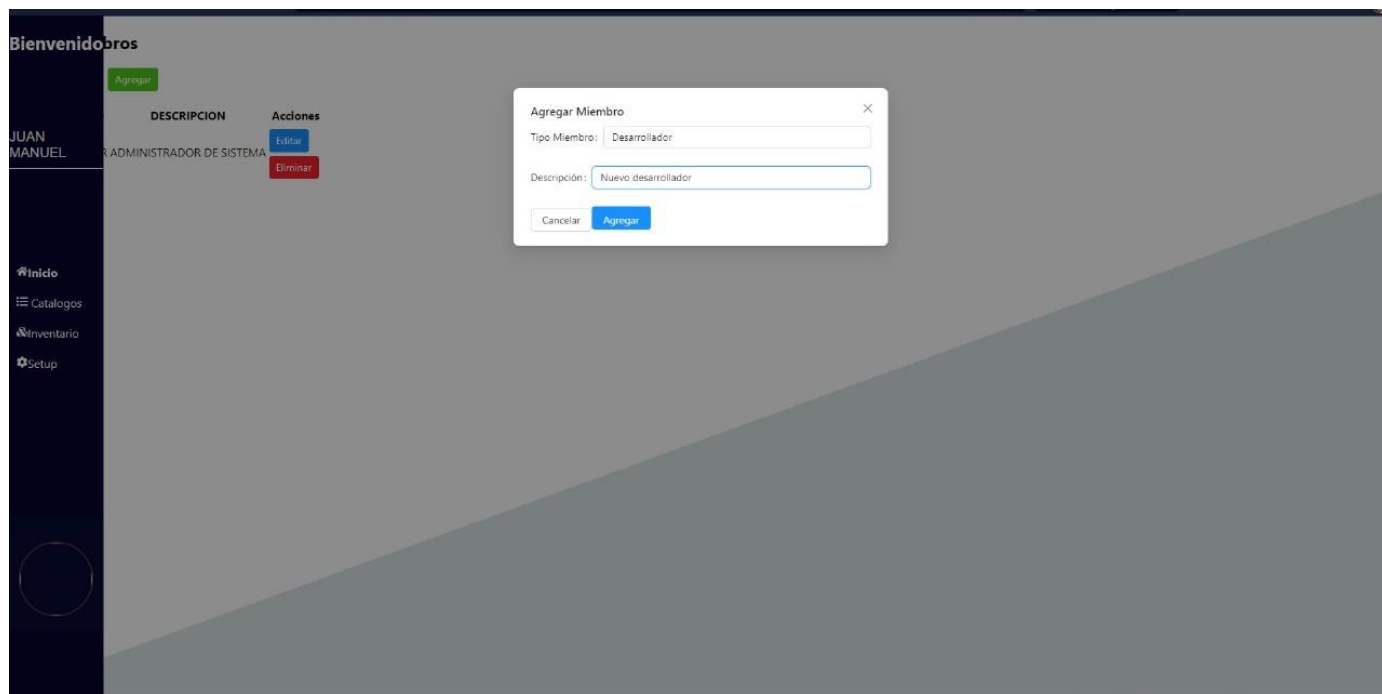
captura 24

Posteriormente en el apartado de "setup" en la sección de cia podemos agregar un nuevo registro a través de la ventana que nos aparece para poder hacerlo (captura 25).



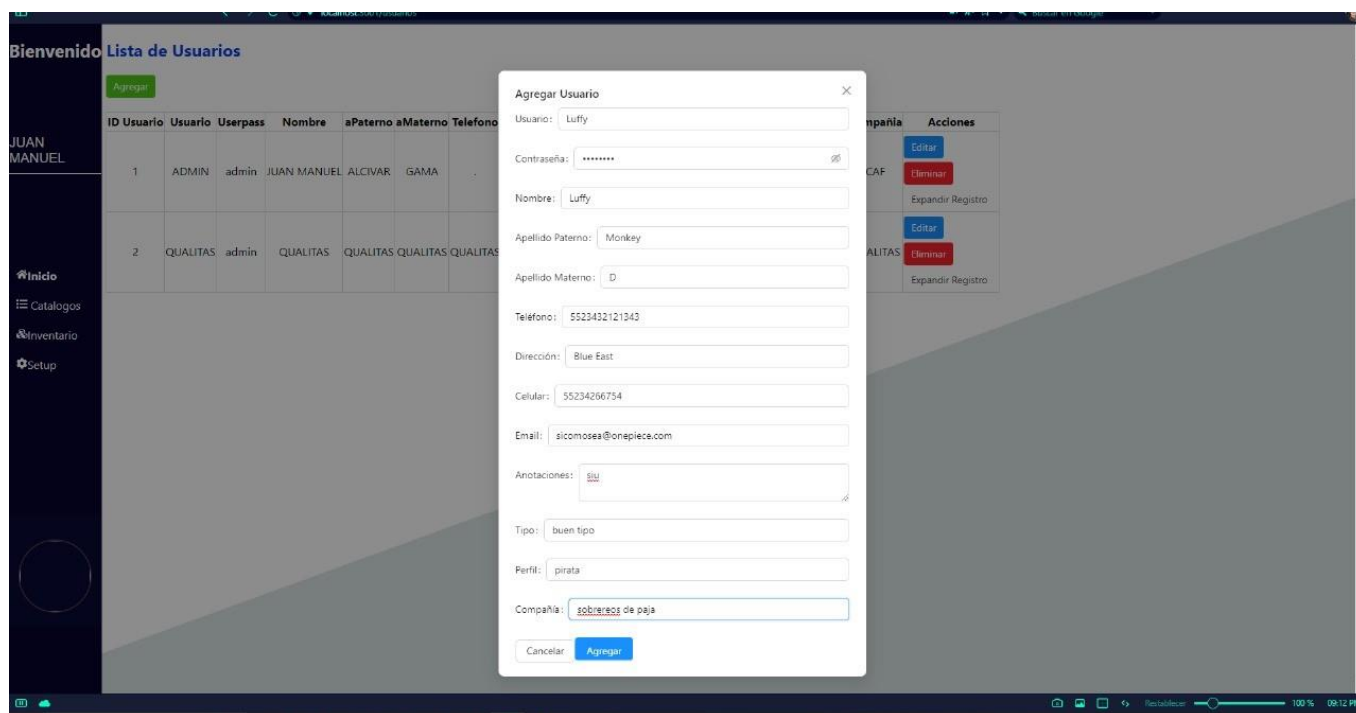
captura 25

En el mismo apartado de setup podremos agregar un nuevo miembro en la sección tipo de miembros (captura 26).



captura 26

En la sección de usuarios del apartado setup podremos agregar un nuevo usuario llenando los campos que nos pide completar el siguiente recuadro (captura 27).



captura 27

Dentro de la unidad del disco local C: se busca la ruta en la cual se están los archivos con los cuales se está trabajando el nodejs, siguiendo la ruta se introduce el código npm start el cual da inicio al nodejs y muestra el host que se utiliza (captura 28).

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19044.2965]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\Prueba>cd C:\Users\Prueba\Estadia\BACKENDEFFAC2
C:\Users\Prueba\Estadia\BACKENDEFFAC2>npm start
> backendrefaci@1.0.0 start
> node app.js
(node:1116) 3.0.11
(node:1116) to restart at any time, enter 'rs'
(node:1116) watching path(s): *.*
(node:1116) watching extensions: js,mjs,cjs,json
(node:1116) startup: node app.js
Servidor corriendo por el puerto: http://localhost:3000
```

captura 28

Una vez iniciado el backend se sigue la misma ruta del disco local C: para dar inicio al frontend de esta forma ambos inician y automáticamente salta la página de inicio del sistema (captura 29).

```
Microsoft Windows [Versión 10.0.19044.2965]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\Prueba>cd C:\Users\Prueba\Estadia\frontendrefaci
C:\Users\Prueba\Estadia\frontendrefaci>npm start
> frontendrefaci@0.1.0 start
> react-scripts start
> Something is already running on port 3000.

Would you like to run the app on another port instead? ... yes
(node:18428) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:18428) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

You can now view frontendrefaci in the browser.

  Local:            http://localhost:3001
  On Your Network:  http://192.168.100.232:3001

Note that the development build is not optimized.
To create a production build, use npm run build.

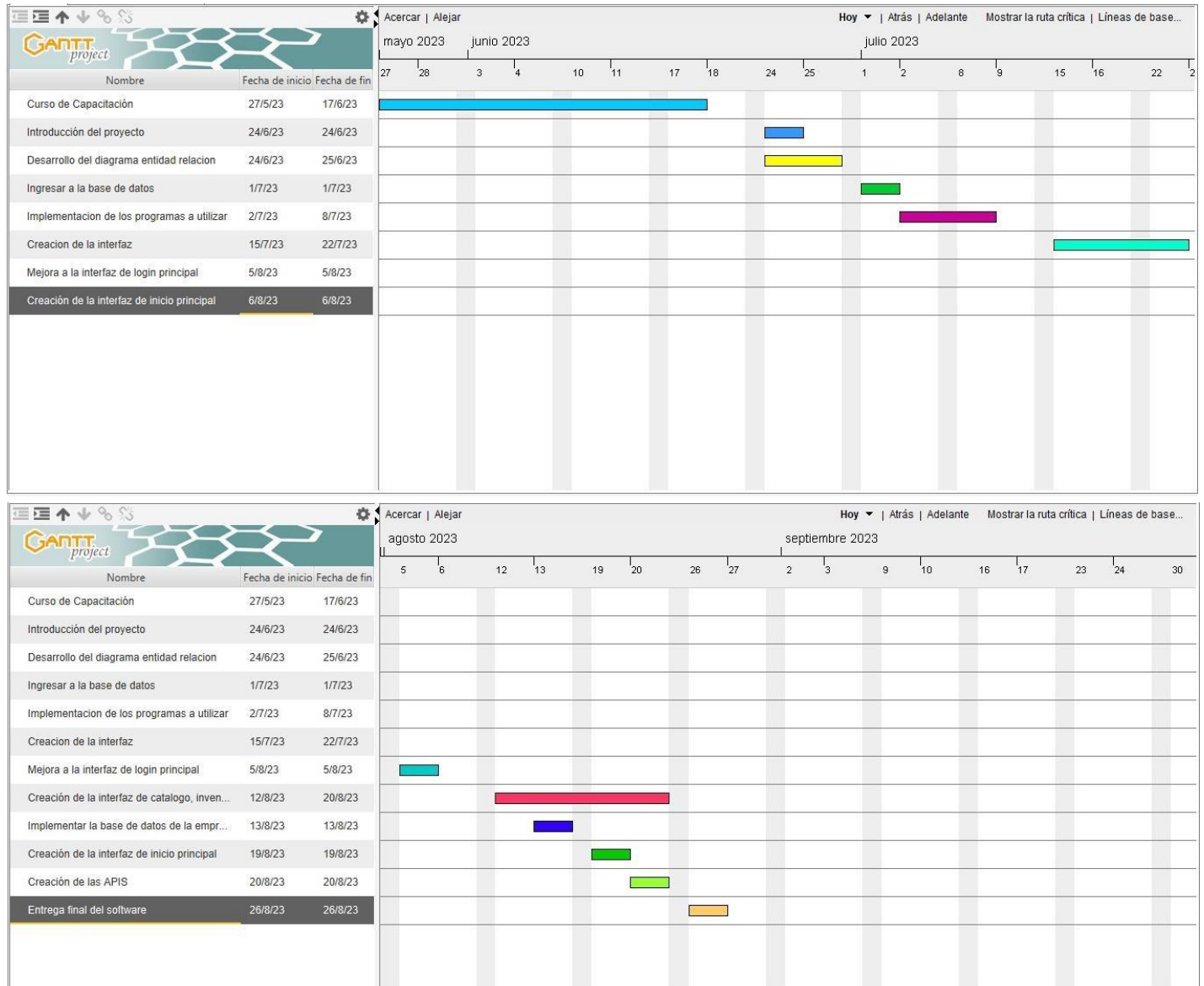
webpack compiled successfully
```

captura 29

Apéndice

Diagrama de Gantt

Aquí se adjuntará la gráfica que se utilizó para la realización de todo el proyecto y que fue de ayuda para organizarnos con el trabajo



Conclusiones

En resumen, el proyecto de la aseguradora de automóviles enfocado en la creación de un sistema orientado a la seguridad y garantía en la recuperación de refacciones ha sido un esfuerzo esencial y visionario. Este proyecto no solo busca proteger los activos de los asegurados, sino que también promueve la cultura de la seguridad vial y la calidad en las reparaciones.

A través de este enfoque, la aseguradora ha demostrado su compromiso con la satisfacción del cliente y la construcción de relaciones a largo plazo basadas en la confianza. Además, este proyecto contribuye al bienestar general de la comunidad al fomentar una conducción más segura. En última instancia, representa un paso importante hacia un futuro más seguro y confiable en la industria de seguros de automóviles.

Queremos agradecer a nuestros compañeros de equipo por su compromiso y colaboración constante. Cada uno de ustedes aportó habilidades únicas y conocimientos valiosos al proyecto, lo que nos permitió abordar problemas de manera efectiva y encontrar soluciones creativas.

También deseamos agradecer a nuestros patrocinadores y supervisores por su apoyo y orientación a lo largo de este proyecto. Sus consejos y retroalimentación fueron fundamentales para nuestro progreso y éxito.

En resumen, este proyecto fue un logro colectivo, y estamos agradecidos por cada aporte que lo hizo posible. Trabajar en equipo nos permitió alcanzar nuestros objetivos y superar obstáculos, y esperamos continuar colaborando en futuros proyectos. Gracias a todos por su compromiso y contribución a nuestro éxito.

Glosario de Términos

Base de datos. -Conjunto organizado de información que se almacena de manera estructurada y que puede ser gestionada, accedida y actualizada de manera eficiente. Está diseñada para permitir el almacenamiento, recuperación y manipulación de datos de manera efectiva

Login. - Es el proceso mediante el cual un usuario se autentica o identifica a sí mismo en un sistema de computadora o una aplicación para acceder a sus recursos o funciones protegidas.

Software. -Programas de computadora, datos y sistemas que se utilizan para realizar diversas tareas en una computadora o dispositivo electrónico. Es la parte no tangible de un sistema informático y abarca todas las instrucciones y códigos de programación que permiten que una computadora realice funciones específicas.

Hardware. -Se refiere a las partes físicas y tangibles de una computadora o dispositivo electrónico, como la CPU, la memoria, el disco duro, la pantalla y el teclado. Estos componentes interactúan para ejecutar programas de software y realizar tareas computacionales.

Interfaz de gráfico. - Es una forma visual y gráfica de interactuar con una computadora o software. Ofrece elementos visuales como ventanas, iconos y botones para que los usuarios puedan realizar acciones de manera intuitiva, en lugar de depender de comandos de texto.

Node.js - Sirve para crear sitios web dinámicos muy eficientes, escritos con el lenguaje de programación JavaScript

React. - La función principal de React es desarrollar páginas web de una manera gratuita y sencilla gracias a sus componentes reutilizables.

MySQL. - MySQL permite almacenar y acceder a los datos a través de múltiples motores de almacenamiento.

Bibliografía

1. Cavero, J. (2023, 29 marzo). Metodología Scrum - MentorDay WikiTips. mentorDay WikiTips. https://mentorday.es/wikitips/metodologia-scrum/?gclid=Cj0KCQjw6KunBhDxARIsAKFUGs9ozfDshtzbCPAmaEvecUqc8tGIT6YUUtKYiu3it8q2WjiYR1qsw-0aAuC2EALw_wcB
2. Martins, J. (2023, 19 junio). Scrum: conceptos clave y cómo se aplica en la gestión de proyectos [2023] • Asana. Asana. <https://asana.com/es/resources/what-is-scrum>
3. Atlassian. (s. f.). Scrum: qué es, cómo funciona y cómo empezar | Atlassian. <https://www.atlassian.com/es/agile/scrum>
4. Apd, R. (2022). Cómo aplicar la metodología Scrum y qué es el método Scrum. APD España. <https://www.apd.es/metodologia-scrum-que-es/>
5. Hurtado, J. S. (2023, 15 febrero). Metodología Scrum: qué es y cómo utilizarla para acometer proyectos. Thinking for Innovation. <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>
6. Pr, N. (2023). ¿Qué es la metodología Scrum? y gestión de proyectos ScRum. Nimblework. <https://www.nimblework.com/es/agile/que-es-scrum/>