Gestión de revisiones y agendamiento de revisión técnica.

SPRINT 2

Misión Tic 2022

Integrantes:

KATERIN VANESSA BLANCO VIVAS

MARY LUZ GARCÍA ECHEVERRI

LEYNER ANDRES PALMA DITA

CÉSAR IVÁN MCNISH FAJARDO

PEDRO DAVID GUERRERO SERRANO

Énfasis en programación web

Universidad de Caldas

Octubre 28 de 2022

CONTENIDO

Paginas

Información general del proyecto …… 3 – 5

Identificación del problema ……..…….3

Objetivo general ……3

Objetivos específicos ……3

Interesados ……3

Restricciones ……..4

Requisitos generales ……4

Casos de uso ……...……5

Modelo entidad relación ……6

Sprint 1 ……7 - 9

Objetivos …… 7

Que debo saber …… 7

Estrategia …….. 7

Logros …… 8

Sprint 2 ……9 - 10

Objetivos …… 9

Que debo saber …… 9

Estrategia …….. 10

Logros …… 10

INFORMACION GENERAL DEL PROYECTO

Identificación del problema

Una empresa que cuenta con diferentes servicios, como revisión mecánica, revisión eléctrica, montallantas entre otros, y que se encuentra en diferentes partes del país, creciendo significativamente, Actualmente está teniendo bastantes inconvenientes en la atención a sus usuarios, ya que no hay orden en la atención, no hay una programación de revisiones y no se tiene un historial de los vehículos que han sido atendidos para tener trazabilidad de las revisiones y para impulsar procesos de fidelización.

Adicional es importante que se tenga un registro histórico de las revisiones, reparaciones y la instalación de repuestos; esto permite luego realizar análisis por parte de los dueños de los vehículos y por el taller.

Objetivo

Desarrollar una pagina web que permita tener control de la atención dentro de los talleres, generando una mejor atención al cliente, permitiendo un crecimiento optimo de estos.

Objetivos específicos

- Permitir a los clientes registrarse y registrar su vehículo, para así poder solicitar las citas para revisiones y recibir las notificaciones después de su atención
- Obtener un registro histórico de las atenciones dentro de los talleres para su análisis, buscando tener una mejor de la atención al cliente
- Mejorar las condiciones laborales para los supervisores y los mecánicos, haciéndoles más ágil el proceso de registro y notificaciones

Interesados

- Los dueños del taller mecánico
- Los clientes
- Los empleados del taller

Restricciones

- El sistema debe tener una base de datos en la nube, para facilitar el despliegue de la aplicación.
- El sistema debe tener separado el front-end y el back-end, para tener las responsabilidades separadas y facilitar los cambios en el futuro. En el back-end se usarán servicios con API REST para facilitar las pruebas
- Los datos de cada vehículo solo pueden ser consultados por el dueño o el mecánico asignado. El jefe de operaciones es el único que puede ver la información de todos los vehículos

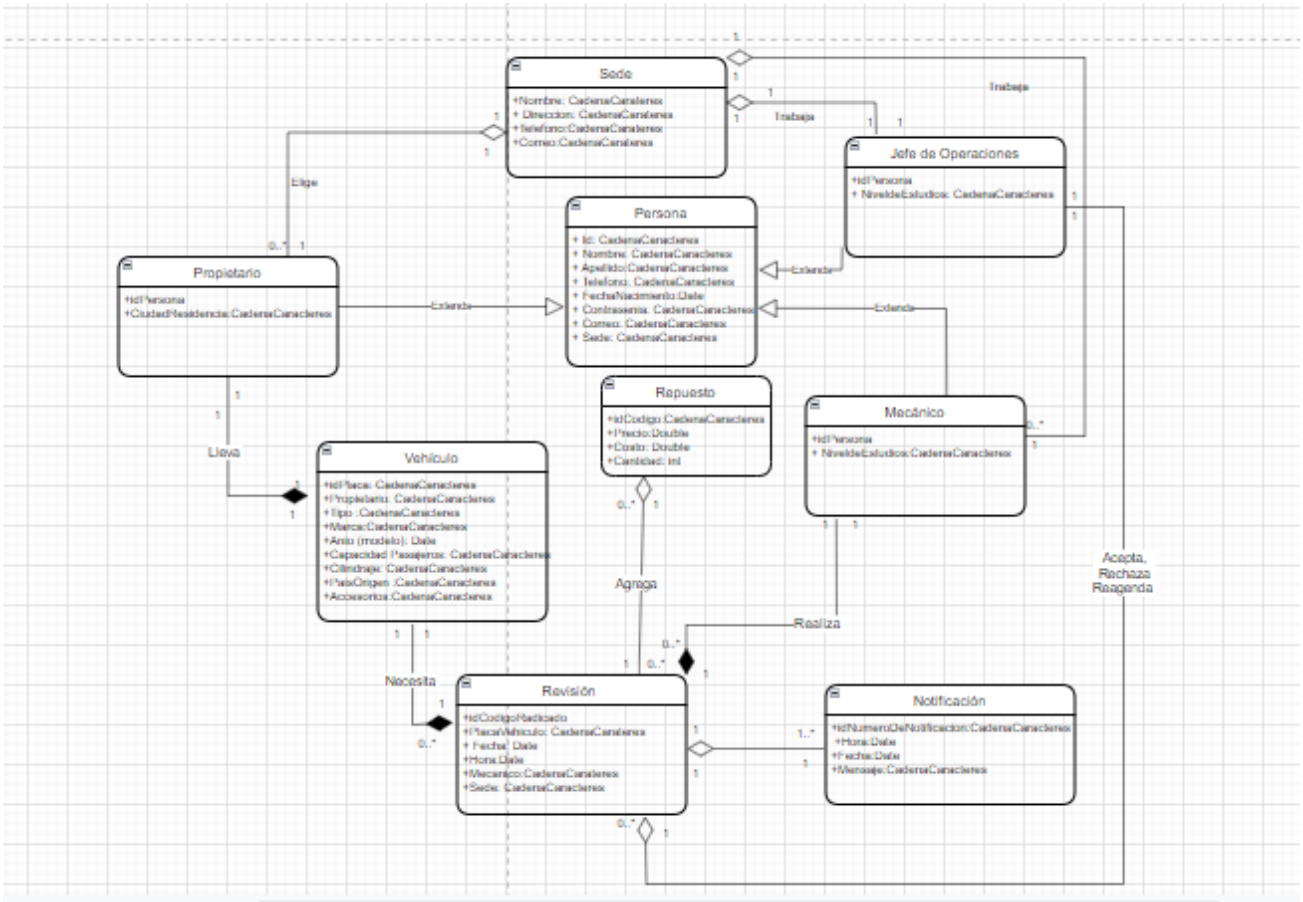
Requisitos generales

Registrar los datos de los vehículos, incluyendo también la información de los dueños.
• Registrar los datos de un mecánico.
• El sistema debe tener un Login.
• Solicitar cita de revisión por parte del usuario.
• Revisar solicitudes de revisiones por parte del jefe de operaciones.
• Permitir aceptar la revisión solicita por el usuario. Esta acción la realiza el jefe de operaciones.
• Asignar un mecánico a cada vehículo para que realice las revisiones.
• Registrar en cada revisión los datos: el nivel de aceite, el nivel del líquido de frenos, el nivel de refrigerante y el nivel de líquido de dirección. Cuando se cambie algún repuesto, esto también se debe registrar y notificar al dueño.
• Llevar un histórico de las revisiones realizadas a cada vehículo.
• Permitir la consulta de la información de las revisiones.
• Permitir reagendar cita por parte del jefe de operaciones y que el usuario pueda aceptar o rechazar el nuevo horario
• Poder consultar las revisiones del día.
• Editar los datos de un mecánico.
• Eliminar un mecánico.

Casos de uso


HISTORIAS DE USUARIOS
Como jefe de operaciones o Supervisor
·Autenticarse en el sistema
·Registrar la información de vehículo nuevo (poder actualizarla) y la del dueño,
·Registrar la información de un nuevo mecánico y poder actualizarla
·Eliminar o inactivar un mecánico del sistema
·Consultar la información general de los vehículos registrados
·Consultar los datos detallados de los vehículos
·Asignar un mecánico a una revisión
·Ver todas las solicitudes de revisiones
Como Mecánico
·Poder autenticarme en el sistema
·Registrar la bitácora de acciones de la revisión y la necesidad de compra de repuestos
·Consultar el listado de los vehículos que tengo asignados, para tener una idea general de mi labor.
·Consultar las revisiones que tengo agendadas para un día en específico
·Registrar la compra de un repuesto
Como jefe de operaciones o Supervisor
·Poder autenticarme en el sistema
·Registrar mi información y la de mi vehículo (poder actualizarla)
·Solicitar la revisión de mi vehículo mediante un calendario
·Consultar la historia de revisiones de realizadas.
·Consultar las notificaciones de una revisión y recibirlas por correo

Modelo entidad relación



Sprint 1

Objetivos

- Generar una base de datos en la nube, en este caso Mongodb, para ser usada en los futuros Sprint y facilitar el despliegue de la aplicación.
- Tener un modelo entidad-relación aprobado por el equipo
- Tener todos los mockups del sistema que correspondan a las historias de usuario.
- Tener modelo conceptual del sistema. Ejercicio en Mongo y Express CRUD de vehículos y usuarios utilizando Express y Postman

Que debo saber

- El manejo básico de la base de datos NoSQL en este caso Mongo DB, Express CRUD y Postman.
- Tener conocimiento de plataformas tipo Draw.io para la creación de los mockups
- Los conceptos de modelo entidad relación
- Saber las relaciones de cardinalidad entre las entidades
- Saber que necesito hacer y cuáles son los objetivos a alcanzar
- Tener claras las restricciones

Estrategia

Para la realización de este Sprint 1se debe:

- Crear una base de datos en la nube, en este caso se realizó la instalación de Mongodb Compass y se registró en Mongodb atlas pata generar en esta la base de datos del taller mecánico.
- Verificar que los mockups estén acordes al objetivo del proyecto y al gusto de los participantes del equipo
- Como se tenia el modelo entidad del ciclo pasado, pero faltaba la relación entre las entidades con el objetivo de más adelante poder hacer uso de esta, se debe concluir esta relación finalmente.
- Realizar el modelo conceptual del sistema, específicamente de vehículos y usuarios utilizando Express y Postman de la siguiente manera.
- Realizar la descarga de Node.js y los respectivos paquetes que permiten utilizar Express y hacer la conexión con MongoDB

- Definir las entidades en la herramienta de acuerdo al modelo discutido anteriormente con el equipo haciendo énfasis en los datos requeridos por el cliente.
- Establecer las rutas de las funciones a utilizar para realizar el CRUD de usuarios y vehículos, en este caso, las funciones son:
    - Agregar Usuario / Agregar Vehículo
    - Buscar Usuario / Buscar Vehículo
    - Eliminar Usuario / Eliminar Vehículo
    - Editar Usuario / Editar Vehículo
    - Listar Usuarios / Listar Vehículos
- Una vez desarrollado el código realizar el CRUD de usuarios y vehículos se procederá a probar las peticiones por medio de la aplicación Postman, utilizando las urls para cada petición.
- Verificar que se guarden los documentos en la base de datos creada por medio de MongoDB Compass, además de poder buscar, editar, eliminar y listar los documentos.

Logros

- El equipo ya estuvo de acuerdo con todos los mockups realizados
- Se concluyo finalmente como iba a ser la relación entre las entidades del modelo entidad relación
- Se realizo el ejercicio del CRUD de Vehículos y usuarios por medio de Express y Postman
- Se creó una base de datos en la nube.

Sprint 2

Objetivo general

- Realizar el BACKEND de diferentes historias de usuario utilizando loopback

Objetivos específicos

- Registrar la información de un nuevo vehículo y la de su dueño para que pueda tener su información actualizada, la información de un nuevo mecánico, para que esté en la lista oficial de la empresa como JEFE DE OPERACIONES
- Consultar la información general de los vehículos registrados, para verificar el registro de la información, y los datos detallados de un vehículo, para verificar la información registrada del mismo como JEFE DE OPERACIONES
- Registrar mi información y la de mi vehículo poder solicitar un servicio automotriz como DUEÑO DEL VEHICULO
- Tener en cuenta todos los criterios de aceptación pata lograr hacer el BACKEND efectivo y útil.

Que debo saber

- El manejo básico Loopback
- Los criterios de aceptación los conceptos modelo entidad
  - relación
    - De cada persona se solicita id, nombre, apellidos, número de teléfono y fecha de nacimiento (este último es opcional).
    - De los mecánicos se solicita (además de los datos de persona) dirección y nivel de estudios (bachiller, técnico o tecnólogo).
    - De los dueños se solicita (además de los datos de persona) la ciudad de residencia y el correo (e-mail).
    - De los vehículos se solicita la placa, el tipo (buseta, microbús), la marca, el año (modelo), la capacidad de pasajeros, la cilindrada de motor, el país de origen y otras características (una descripción general, por ejemplo, si tiene aire acondicionado, televisor, etc.). El país de origen y otras características son opcionales.
    - Se debe validar que se ingresen los campos obligatorios y que cada campo tiene el formato correcto (por ejemplo, caracteres en los nombres, números en el teléfono, formato con @ en el correo).
    - Al consultar los vehículos se debe mostrar una lista con la placa, tipo, marca, modelo y capacidad.
    - Para consultar un vehículo se debe solicitar su placa y se muestra toda su información.

Estrategia

Para la realización de este Sprint 2 se debe:

1. Conectar Loopback con la base de datos, en este caso MongoDB
2. Crear en la carpeta modelos las entidades según se tiene planteado en el modelo Entidad Relación, una por una con sus propiedades
3. En la carpeta repository se crea un repositorio por cada modelo
4. Se crean las relaciones según el modelo Entidad Relación, si es belongs to, has one, hasManyThrough, o hasMany
5. Finalmente, ya se puede evidenciar en la base de datos la información solicitada.

Logros

- Se logro registrar en la base de datos la información que se necesitaba como jefe de Operaciones
- Se puede consultar la información registrada tanto como jefe de e operaciones como Dueño del vehículo revisado
- Se tuvo en cuenta todos los criterios de aceptación para que el registro y consulta de la información se haga de manera exitosa.