



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Plataforma de seguimiento y
rastreo satelital**

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de

Ingeniero Eléctrico Electrónico

P R E S E N T A

Juan Carlos García Hilario

ASESOR DE INFORME

Ing. Gabriel Alejandro Jaramillo Morales



Ciudad Universitaria, Cd. Mx., 2024



**PROTESTA UNIVERSITARIA DE INTEGRIDAD Y
HONESTIDAD ACADÉMICA Y PROFESIONAL**
(Titulación con trabajo escrito)



De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado PLATAFORMA DE SEGUIMIENTO Y RASTREO SATELITAL que presenté para obtener el título de INGENIERO ELÉCTRICO ELECTRÓNICO es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Entidad Académica, citando las fuentes de ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia, acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación.



JUAN CARLOS GARCIA HILARIO

Número de cuenta: 304313635

Agradecimientos

Quiero agradecer a mi familia que siempre estuvo apoyándome en cada etapa de mi vida, como lo es ahora cuando realizo y culmino este trabajo para recibir mi título como ingeniero.

Agradezco a mi madre Anastasia Hilario Mendoza por sus consejos y esfuerzo que hizo para cuidarme y siempre estar ahí para mí.

A Maribel García Hilario que siempre me apoyó y en momentos difíciles no me dejó solo, ayudándome para continuar con mis estudios y orientándome cuando he equivocado el camino.

A Guadalupe García Hilario quien cuidó de mí y me dio mucho cariño, lo cual, fue muy importante para no olvidar mis objetivos y metas.

A Olga Azucena Núñez Valdez quien me alentó a iniciar y terminar este trabajo, si no fuera por ella quizá este trabajo no existiría y no hubiera cerrado este ciclo.

Especialmente agradezco a mi Profesor Ing. Gabriel Alejandro Jaramillo Morales, quien me apoyó y aconsejó durante todo este trabajo, muchas gracias, profesor.

Declaración de autenticidad

Yo Juan Carlos García Hilario, declaro que el contenido de este informe es original y no se ha presentado total o parcialmente en otros informes para la obtención de otro título o grado en esta Universidad.

En este informe no se da publicidad a empresa alguna o marca, por lo tanto, si he de mencionar alguna, no se menciona con el fin de publicidad ni a modo de recomendación, es parte del informe y de las herramientas ya sea de software o hardware utilizados durante mi participación en el proyecto, por lo que, renuncio a cualquier marca, la cual, pertenece y es responsabilidad de su creador o fabricante.

Declaro que este informe es el resultado de mi esfuerzo y experiencia al participar en proyectos desarrollados y puestos en producción sobre rastreo satelital.

Juan Carlos García Hilario. Ciudad de México, México, 2024

Índice

1	INTRODUCCIÓN	1
2	OBJETIVO	3
2.1	OBJETIVOS ESPECÍFICOS	3
3	PARTICIPACIÓN EN LA EMPRESA	4
3.1	FUNCIONES DESEMPEÑADAS.....	4
4	MARCO TEÓRICO	5
4.1	¿QUÉ ES EL GPS?.....	5
4.1.1	<i>Segmentos GPS</i>	5
4.1.2	<i>¿Cómo funciona el GPS?</i>	6
4.2	RASTREO SATELITAL	7
4.3	RASTREADORES GPS.....	8
4.4	APLICACIONES DE LOS RASTREADORES GPS	8
5	DESARROLLO DEL PROYECTO	10
5.1	ANTECEDENTES Y PROBLEMÁTICA	10
5.2	PARTICIPACIÓN Y PRINCIPALES ACTIVIDADES EN EL DESARROLLO	11
5.3	LEVANTAMIENTOS DE REQUERIMIENTOS.....	11
5.4	ALCANCE DEL PROYECTO.....	12
5.5	METODOLOGÍA DE DESARROLLO.....	13
5.5.1	<i>Sprint en Scrum</i>	14
5.5.2	<i>Roles centrales en Scrum</i>	15
5.5.3	<i>¿Qué es el Product Backlog?</i>	15
5.6	EQUIPO DE TRABAJO	15
5.7	TECNOLOGÍA.....	16
5.7.1	<i>Microsoft SQL Server</i>	16
5.7.2	<i>Front End</i>	17
5.7.3	<i>Back End</i>	17
5.8	DESARROLLO E IMPLEMENTACIÓN	18
5.8.1	<i>Product Backlog</i>	18
5.8.2	<i>Arquitectura general de la plataforma</i>	20
5.8.3	<i>Diseño de la base de datos</i>	21
5.8.4	<i>Planeación de Sprint</i>	21
5.8.5	<i>Decodificador de paquetes</i>	21
5.8.6	<i>Portal Web administrador</i>	24
5.8.7	<i>Portal Web de usuarios</i>	32
6	PUESTA A PRODUCCIÓN	44
6.1	REVISIÓN Y PRUEBAS DE CAJA NEGRA.....	44
6.2	PUESTA A PRODUCCIÓN	45
7	CONCLUSIONES	47
8	REFERENCIAS	49

Índice de tablas y figuras

FIGURA. 1 CIFRAS DE INEGI DE VEHÍCULOS DE MOTOR REGISTRADOS EN CIRCULACIÓN.....	1
FIGURA. 2 REPRESENTACIÓN GRÁFICA DEL CONCEPTO DE UN SPRINT	14
FIGURA. 3 EJEMPLO DEL PRODUCT BACKLOG	19
FIGURA. 4 ARQUITECTURA GENERAL DE LA PLATAFORMA	20
FIGURA. 5 DIAGRAMA DE LA BASE DE DATOS.....	21
FIGURA. 6 LAYOUT (MOCKUP) LOGIN	26
FIGURA. 7 LAYOUT (MOCKUP) RECUPERAR CONTRASEÑA	27
FIGURA. 8 LAYOUT (MOCKUP) PANTALLA DE USUARIOS	28
FIGURA. 9 LAYOUT (MOCKUP) VEHÍCULOS VINCULADOS A UN USUARIO	29
FIGURA. 10 LAYOUT (MOCKUP) PERMISOS DE USUARIO	29
FIGURA. 11 LAYOUT (MOCKUP) CAMBIAR CONTRASEÑA A USUARIO	30
FIGURA. 12 LAYOUT (MOCKUP) VINCULAR O DESVINCULAR VEHÍCULO A USUARIO	31
FIGURA. 13 LAYOUT (MOCKUP) PANTALLA PRINCIPAL DE MONITOREO	34
FIGURA. 14 CÓDIGO DE IMPLEMENTACIÓN DE UN MAPA DE GOOGLE EN UN DOCUMENTO HTML	36
FIGURA. 15 LAYOUT (MOCKUP) EDITAR VEHÍCULO	38
FIGURA. 16 LAYOUT (MOCKUP) BUSCADOR DE VEHÍCULOS	39
FIGURA. 17 LAYOUT (MOCKUP) PANTALLA REPORTE HISTÓRICO	40
FIGURA. 18 LAYOUT (MOCKUP) ADMINISTRACIÓN DE GEOCERCAS.....	41
FIGURA. 19 LAYOUT (MOCKUP) NOTIFICACIONES	42
FIGURA. 20 LAYOUT (MOCKUP) REPORTE NOTIFICACIONES	43
FIGURA. 21 TABLA EJEMPLO MATRIZ DE PRUEBA	45

1 INTRODUCCIÓN

El transporte en México es un área de negocio que deja cientos de miles de pesos al año y muchos transportistas y/o dueños de empresas donde su rol principal es la renta, adquisición y compra de vehículos, les preocupa no solo conocer dónde están sus vehículos en todo momento, sino que, buscan y contratan herramientas de software y hardware que les ayude a administrar y mejorar la logística de sus flotillas de vehículos.

Es por ello, que el rastreo de vehículos no sólo está dirigido a transportistas o grandes empresas, hay miles de personas que están preocupadas en cuidar su patrimonio. Según el INEGI (INEGI, s.f.) las cifras en el 2023 de los vehículos de motor registrados en circulación eran en total 58,199,293, según la tabla siguiente.

Año de registro	Total	Automóviles	Camiones para pasajeros	Camiones y camionetas para carga	Motocicletas
2013	36,744,838	24,819,922	347,144	9,704,131	1,873,641
2014	38,027,171	25,543,908	347,720	9,863,980	2,271,563
2015	39,975,998	26,907,994	357,452	10,073,288	2,637,264
2016	42,454,265	28,664,295	367,772	10,384,621	3,037,577
2017	45,853,522	30,958,042	401,120	10,895,817	3,598,543
2018	47,783,926	32,290,067	443,579	10,970,128	4,080,152
2019	49,869,688	33,603,591	453,606	11,022,869	4,789,622
2020	51,215,678	34,425,695	461,089	11,044,684	5,284,210
2021	53,102,838	35,455,026	452,766	11,258,873	5,936,173
2022	55,167,421	36,496,879	468,586	11,395,481	6,806,475
2023	58,199,293	38,039,898	487,243	11,887,666	7,784,486

Figura. 1 Cifras de INEGI de vehículos de motor registrados en circulación

Estas cifras nos indican que existe un mercado creciente, en donde, posiblemente muchas personas buscarán opciones que les ayude a proteger su patrimonio. Aquí es donde, una plataforma de rastreo puede ayudar a dar seguimiento a cualquier activo móvil, por ejemplo:

- Personas
- Vehículos particulares o de carga
- Motos
- E incluso mascotas
- ETC.

Dicho lo anterior, he tenido la oportunidad de trabajar en un par de empresas dedicadas al rastreo satelital, que me han permitido no solo conocer varias plataformas de rastreo sino participar en el desarrollo y puesta en producción de un par de ellas. Donde he

ocupado puestos que van desde programador hasta gerente en el área de desarrollo de software.

El desarrollar una plataforma de rastreo no es tarea fácil, sin embargo, el punto determinante entre el éxito y el fracaso es entender los requerimientos del cliente a la perfección y no dar nada por hecho que pudiera alterar los objetivos del desarrollo, ya que, un error en la planeación puede desencadenar que los módulos no sean eficientes y contengan información de poco valor, pese a eso, esta información innecesaria requiere recursos de cómputo para su procesamiento, lo que se refleja en un gasto en infraestructura extra.

En el presente informe se plasmará de forma general, pero sin poner en riesgo temas de confidencialidad, lo realizado en la creación de una plataforma de rastreo satelital, desde su planteamiento, desarrollo y puesta a producción.

Es importante aclarar que no se mostrará código alguno y que todas las imágenes mostradas son únicamente ilustrativas y no pertenecen al desarrollo original, sino que serán elementos basados en los ejemplos mostrados durante este informe.

2 OBJETIVO

El objetivo general de este informe es dar a conocer el trabajo y participación que he tenido en el diseño de una plataforma de rastreo satelital, la cual, es capaz de soportar aproximadamente 10 mil rastreadores GPS concurrentes.

2.1 OBJETIVOS ESPECÍFICOS

- Crear un software que actúe como servidor TCP para la homologación de equipos de rastreo GPS de la marca Queclink. Dicho software deberá ser capaz de recibir, interpretar y almacenar en una base de datos, la información según la hoja de datos del equipo de rastreo GPS proporcionada por el fabricante, dónde, se indica cómo debe realizarse la de decodificación de la información transmitida por los equipos GPS a través del protocolo TCP.
- Diseñar una página web de administración que permita el acceso a través de un usuario y contraseña, así como crear, editar y/o dar de baja usuarios, además, de contar con la opción de modificar los permisos de cada usuario y vincular o desvincular los vehículos a los que tiene acceso.
- Diseñar una página web, que permita el acceso a través de un usuario y contraseña, además de la visualización de los vehículos asignados al usuario y la consulta de la información histórica de las posiciones reportadas por cada rastreador GPS.
- Crear una base de datos escalable, que permita almacenar la información proveniente de los equipos de rastreo GPS y que permita consultar la información desde los diferentes softwares que conforman la plataforma de rastreo.

3 PARTICIPACIÓN EN LA EMPRESA

3.1 FUNCIONES DESEMPEÑADAS

La empresa a la que hace referencia este informe me ha permitido desenvolverme en dos áreas que me apasionan, electrónica y software, aunque, he desarrollado más actividades dentro del área de desarrollo de software tal como se verá a continuación.

- He participado dando mi opinión en el diseño electrónico que incluye la adaptación de sensores a los equipos de rastreo, por ejemplo, extensión de puertos u homologación de sensores de gasolina, no obstante, no son de mi responsabilidad las pruebas y el diseño final de estos aditamentos.
- Pruebas de homologación de equipos de rastreo con diferentes plataformas de rastreo, donde mi principal trabajo era revisar la información que llega a los servidores proveniente de los equipos y crear el software necesario para poder interpretar dicha información.
- Mi actividad principal es estar a cargo de todo el equipo de desarrollo de la empresa, que incluye:
 - Administración del área de desarrollo de software.
 - Contratación y selección de programadores independientes y eventuales.
 - Seguimiento a proyectos en desarrollo de software.
 - Levantamiento de requerimientos para la planeación, implantación y puesta en producción de desarrollos de software.
 - Líder de proyecto en la creación, desarrollo y puesta en producción de una plataforma de rastreo.

4 MARCO TEÓRICO

4.1 ¿QUÉ ES EL GPS?

GPS significa, Global Positioning System, que en español es un Sistema de Posicionamiento Global, el cual es en principio propiedad de los Estados Unidos de América, quien tiene el control y mantenimiento de dicho sistema (EUA, s.f.).

El sistema GPS basa su funcionamiento en los principios de los Sistemas Globales de Navegación por Satélite (GNSS) permitiéndonos determinar la posición, navegación y cronometría de un objeto en cualquier parte de la superficie de la Tierra.

Aunque el sistema GPS no es el único sistema en el mundo que proporciona servicios de posicionamiento global, sí es uno de los sistemas con la mayor cuota de mercado hoy en día.

4.1.1 Segmentos GPS

El sistema GPS se puede dividir en tres segmentos que permiten su correcto funcionamiento:

- Segmento espacial
- Segmento de control
- Segmento de usuarios

4.1.1.1 *Segmento espacial*

Está conformado por 24 satélites operativos que orbitan la Tierra a 20200 km de altura y pueden transmitir señales de radio unidireccionales con la posición y hora de cada satélite. Según lo indicado por los Estados Unidos, la disponibilidad de estos 24 satélites GPS es estar operativos el 95% del tiempo.

Los satélites GPS se distribuyen en 6 planos orbitales con la misma distancia entre ellos, que rodean a la Tierra y cada plano orbital se divide en 4 zonas con su propio satélite de referencia, esta distribución de los satélites garantiza que desde cualquier lugar de la Tierra se tenga a la vista al menos 4 satélites.

4.1.1.2 *Segmento de control*

El segmento de control consiste en una red global de instalaciones en tierra que rastrean los satélites GPS, supervisan sus transmisiones, realizan análisis, envían comandos y

datos a la constelación de satélites. Estas estaciones están distribuidas en todo el mundo e incluyen:

- Una estación de control maestra.
- Una estación de control maestro alternativa.
- 11 antenas de comando y control.
- 16 sitios de monitoreo.

4.1.1.3 Segmento de usuario

El Sistema de Posicionamiento Global es utilizado en cientos de aplicaciones y sectores, por ejemplo, se tiene una sección de usuarios en general que permite su utilización con ciertas limitaciones, se puede utilizar para incorporarlos en rastreadores GPS en vehículos, para obtener la posición en teléfonos celulares, etc.

Otra sección es para uso exclusivo de la milicia de los Estados Unidos de América, de lo cual, se sabe que cuenta con mayores prestaciones como la exactitud.

4.1.2 ¿Cómo funciona el GPS?

Los satélites que conforman el sistema GPS emiten señales conocidas como efemérides de 1500 bits de tamaño, las cuales son emitidas en paquetes de 50 bits por segundo a una frecuencia aproximada de 1600 MHz durante 30 segundos.

Cada efeméride es una onda electromagnética que viaja a través del espacio, aproximadamente a la velocidad de la luz 299,792 km/s y tienen que viajar a través de las diferentes capas de la Tierra, por ejemplo, la ionosfera donde debido a las condiciones atmosféricas de esta capa sufren un retardo, debido a esto las señales portadoras son transmitidas en dos frecuencias de modulación diferentes, para contrarrestar dicho retardo (García, ¿Qué es y cómo funciona el GPS?, 2023).

Cuando los receptores, que en este caso son rastreadores GPS, reciben las señales o efemérides del satélite, saben el tiempo exacto en que fue emitida dicha señal por el satélite, ya que, esta información se encuentra en cada efeméride transmitida, y el receptor quien tiene un reloj de cuarzo interno, sabe la hora exacta en que recibió esta señal, por lo tanto, podemos calcular el tiempo que tardó en llegar la señal al receptor, y dado que, la velocidad a la que viaja la señal es la velocidad de la luz, podemos calcular la distancia entre el receptor y el satélite (García, Sistemas Global de Navegación por Satélite GNSS, 2024).

Por lo tanto, para determinar la posición de un objeto en la superficie de la Tierra, se requiere que el rastreador GPS reciba señales de al menos cuatro satélites, ya que:

- Si el rastreador GPS recibe efemérides de un solo satélite se determina que el receptor se encuentra en algún punto de una esfera de radio igual a la distancia entre el satélite y el receptor, con centro en el mismo satélite.
- Si el rastreador GPS recibe efemérides de dos satélites, podemos indicar que el rastreador se encuentra en cualquiera de las intersecciones entre las dos esferas creadas por cada satélite.
- Si el rastreador GPS recibe efemérides de tres satélites, se reduce a dos el número de intersecciones entre las tres esferas creadas por los satélites, por lo tanto, aún no tenemos claro cuál es su verdadera posición, ya que, una de estas intersecciones no es verdadera.
- Para resolver este problema, necesitamos un cuarto satélite como referencia o control cruzado, con el cual se elimina la intersección no verdadera y podemos determinar la posición del receptor GPS.

4.2 RASTREO SATELITAL

El rastreo satelital en México está dominado por empresas que proporcionan servicios de localización de vehículos y otros activos móviles, además, proporcionan servicios de valor agregado como centros de monitoreo y/o servicios de recuperación de vehículos por mencionar algunos.

Sin importar cuál sea el tipo de servicio o factores que hagan mejor a una empresa que a otra, el rastreo satelital es lo mismo.

Dar seguimiento a activos móviles, como vehículos, motocicletas, personas, mascotas, barcos, etc., recuperando su última posición cada determinado tiempo, la cual, se muestra en una plataforma web o aplicación móvil, que proporciona información útil en reportes.

Muchas empresas basan su negocio en la compra y venta de equipos de rastreo, lo importan y distribuyen dentro de México en soluciones completas que incluyen servicios de valor agregado como los mencionados anteriormente, no obstante, una de las principales ganancias que se tiene en este tipo de empresa es a través de la renta de plataforma de rastreo, ya que, en muchos casos es una renta mensual, por equipo o por volumen de equipos.

Como usuarios y consumidores, no necesariamente se requiere contratar un servicio con estas empresas, se pueden adquirir dispositivos GPS en tiendas Online, los cuales, nosotros mismos podemos instalar en nuestros vehículos y rastrearlos vía mensajes de texto o en plataformas de rastreo gratuitas, las cuales suelen tener funciones limitadas y la mayoría en idioma inglés o chino.

4.3 RASTREADORES GPS

Un rastreador GPS es un receptor capaz de captar las señales de los sistemas de navegación como lo es el sistema GPS y nos permiten recuperar la última posición del receptor y transmitirla vía GPRS (Servicio General de Paquetes de Radio).

Estos receptores los podemos encontrar y conseguir en una infinidad de fabricantes, como es el caso de Queclink o genéricos (chinos). Cada rastreador viene con especificaciones dadas por cada fabricante tanto en Firmware como en Hardware.

No existe algún estándar que controle y/o regule el protocolo de comunicación en los rastreadores GPS, cada fabricante crea su propio protocolo y decide si lo expone a creadores de plataformas de rastreo o lo reserva únicamente para sus productos y servicios.

Dado que no hay un estándar en el protocolo de comunicación, cada equipo que se utiliza en la construcción y desarrollo de una plataforma de rastreo es único, y requiere de una homologación en particular.

Homologar un equipo significa que se examinará a detalle la hoja de datos del equipo y se creará un software que pueda interpretar cada trama o paquetes que el dispositivo emita con la información de posición y/o puertos de entrada y salida según sea el caso, además del tipo de equipo, por lo que, crear este software que interpreta los datos es parte fundamental de mis actividades en este proyecto.

4.4 APLICACIONES DE LOS RASTREADORES GPS

Hoy en día el posicionamiento de un objeto vía GPS se ha vuelto algo tan común que quizá se nos olvida que está presente en nuestra vida todos los días.

Una de las aplicaciones más importantes y de actualidad es en los Smartphone, los celulares inteligentes cuentan en su diseño con un radio GPS que sirve de receptor, y con el cual, se puede determinar la posición del dispositivo móvil tal como lo haría un rastreador GPS dedicado (aunque en un celular se tienen ciertas limitaciones), y que, si lo combinamos con algunas aplicaciones móviles como Google Maps, no solo tenemos un rastreador GPS sino todo un navegador GPS que nos permite visualizar nuestra posición con información extra de tráfico, velocidad e información de lugares cercanos.

También podemos encontrar rastreadores GPS en tracto camiones que transportan mercancía, ubicando dichos rastreadores en el tracto, caja y/o mercancía.

Con un rastreador podemos localizar personas, a través de un mini GPS que puede llevar en el bolsillo o mochila.

Existen localizadores que se pueden ocultar en los vehículos y no requieren tipo alguno de instalación, ya que, cuentan con poderosos imanes que se adhieren a la carrocería del vehículo, además de ser impermeables y cuentan con batería de larga vida.

Podemos rastrear motocicletas, y mediante una instalación adecuada evitar que encienda sin autorización o enviar un comando que apague la motocicleta vía remota, incluso esto también es posible en autos particulares y tractos.

Otras aplicaciones comunes son para no perder de vista a personas mayores o con alzhéimer, cuidado de mascotas, transporte marítimo, etc.

En este informe nos centramos en el seguimiento y localización de vehículos, ya que, es el principal objetivo de la plataforma de rastreo desarrollada.

5 DESARROLLO DEL PROYECTO

En esta sección se desglosan los puntos más importantes referente a cómo fue creada la plataforma de rastreo satelital en cuestión, explicaré a grandes rasgos los antecedentes y problemática que permitieron el desarrollo y cómo es que fue resuelta cada situación.

5.1 ANTECEDENTES Y PROBLEMÁTICA

El inicio de este proyecto comienza con una reunión que tuve con el director general y el director de investigación y desarrollo de esta empresa, me comentaron que tienen un cliente que presta servicios de renta de vehículos en todo México, dicho cliente cuenta con una plataforma de rastreo y que desea renovarla y mejorarla.

Anteriormente su cliente operaba con una plataforma de rastreo que fue desarrollada por otro equipo de trabajo quienes crearon y pusieron en producción dicha plataforma, sin embargo, esta plataforma contaba con varias deficiencias que afectan directamente la operación de su cliente debido a caídas constantes, lentitud en los servicios, reportes y tiempos de respuesta muy largos por parte del equipo de desarrollo anterior.

El desarrollo no inició inmediatamente, por lo que, mis primeras tareas eran, revisar la documentación de los equipos de rastreo GPS que posteriormente serían homologados en esta nueva plataforma, analizaba y estudiaba periféricos y sensores para su documentación y homologación de los datos.

Entender las configuraciones y el diseño electrónico que la empresa ya tenía respecto a los rastreadores GPS, como, por ejemplo, las integraciones que contaban con sensores de combustibles, contadores de pasajeros, etc., me fue fácil comprender y dar una opinión respecto a lo que ya tenían desarrollado debido a mi perfil como electrónico y haber trabajado con microcontroladores y electrónica de potencia en otra empresa anteriormente.

Por otro lado, lo que también me permitió ser parte de este proyecto fue la experiencia adquirida durante mi estancia en otra empresa que operaba dentro del mismo sector del rastreo satelital, en la cual, llegué a ser gerente en el área de desarrollo de software y desarrollar una plataforma de rastreo similar.

Avanzando en el proyecto, se presenta la documentación y contratos al cliente, lo cual, nos lleva un tiempo en negociación y ponernos de acuerdo, durante el tiempo de revisiones y reuniones, se me asigna la tarea de realizar un “demo” que nos ayude a cerrar la negociación y que además afiance la confianza del cliente al demostrar la capacidad del equipo de desarrollo.

El demo lo desarrollé en 20 días aproximadamente, el cual cuenta con:

- La primera versión del decodificador de paquetes provenientes de los equipos GPS, esta primera versión del software no cuenta con tipo alguno de optimización, sólo

permite la recepción de un solo tipo de paquete con los datos de posición del GPS y no era multihilos.

- Se crea una base de datos en SQL Server, la cual únicamente tiene 2 tablas, una que sirve para almacenar un usuario y contraseña y otra que almacena la decodificación del paquete proveniente del equipo GPS.
- Se crea una página web en el ambiente de C# .NET conectada a SQL Server, esta página se conforma de:
 - Una pantalla de acceso con usuario y contraseña.
 - Una pantalla que muestra la cartografía de Google, en donde, podemos visualizar el equipo GPS directamente en el mapa y permite su seguimiento en tiempo real.
 - En esta misma pantalla podemos observar datos de velocidad, longitud, latitud y fecha provenientes de la decodificación hecha previamente.

El demo fue mostrado al cliente y funcionó correctamente, hubo comentarios, recomendaciones, modificación en el documento de requerimientos y contrato, pero finalmente se consiguió cerrar el proyecto y se dio luz verde para el inicio del desarrollo de la plataforma, con la premisa de desarrollar una plataforma desde cero y hacer a un lado la plataforma de rastreo satelital que le habían desarrollado al cliente previamente.

El inicio del proyecto fue un poco complicado, el equipo anterior no dejó código de la plataforma, fue muy difícil tener acceso a la base de datos, por lo que, no se pudo tener una referencia anterior ni partir de algo que ayudará a homologar los equipos que tenían ya en operación, por lo que se creó todo desde cero bajo el ambiente de NET Framework (Microsoft, Información general de la biblioteca de clases de .NET, 2023) el cual es un entorno de ejecución para Windows (Windows, s.f.) y el motor de base de datos SQL Server (Microsoft, Prueba SQL Server en el entorno local o en la nube, s.f.).

5.2 PARTICIPACIÓN Y PRINCIPALES ACTIVIDADES EN EL DESARROLLO

Una vez iniciado el proyecto, mi rol principal es ser responsable del desarrollo de software, por lo tanto, mi actividad principal es crear todos los componentes de software necesarios para el funcionamiento de la plataforma de rastreo.

La configuración de equipos, instalación de equipos, diseño electrónico y desarrollo de periféricos o aditamentos extras y pruebas físicas o en sitio, son realizadas por el equipo de investigación y desarrollo de la empresa que me contrató y de lo cual no cuento con injerencia ni decisión en cuanto a sus actividades.

5.3 LEVANTAMIENTOS DE REQUERIMIENTOS

El levantamiento de requerimientos es una parte fundamental para este tipo de desarrollos, ya que, permite comprender las necesidades de los usuarios que finalmente utilizarán la plataforma de rastreo.

Se llevaron a cabo varias visitas a las diferentes áreas que conforman la empresa del cliente, **el proceso de levantamiento de requerimiento se completó en poco más de un mes**, ya que, las instalaciones del cliente se encuentran en varios puntos de la Ciudad de México y el Estado de México.

Además, algunas áreas tenían resistencia al cambio debido a que anteriormente se tenía una plataforma de rastreo y ciertas funciones las querían exactamente igual, ahora bien, para lograr la participación de las diferentes áreas se les indica que en esencia podrán hacer lo mismo, sólo que, con una interfaz de usuario diferente, e incluso, contarán con nuevas funcionalidades.

5.4 ALCANCE DEL PROYECTO

Concluidas las negociaciones y reuniones con los usuarios se establecen los alcances generales e iniciales de la plataforma de rastreo, los cuales en resumen son:

- Debe soportar un aproximado de 10 mil equipos concurrentes, de la marca Queclink.
- Debe existir un administrador de usuarios que permita dar de alta, editar o desactivar usuarios, así como editar los permisos y los vehículos a los que tendrán acceso.
- Debe permitir visualizar la información general de GPS:
 - Velocidad
 - Posición (latitud, longitud y altitud)
 - Odómetro
 - Fecha hora del paquete enviado por el equipo
 - Estado de ignición
 - Orientación del equipo
 - Batería del vehículo
 - IMEI (Identidad Internacional de Equipo Móvil), placas, modelo, etc.
- Debe contar con geocodificación inversa, lo cual, es la transformación de una ubicación latitud y longitud en una dirección postal.
- Debe permitir editar los parámetros de un vehículo como son las placas, marca, modelo, versión, etc.
- Debe contar con la opción de poder enviar comandos a los equipos de rastreo que permitan la activación y desactivación de los puertos del equipo GPS, con la intención de activar o desactivar el bloqueo de motor, claxon o solicitar posición del equipo en cualquier momento.
- Debe contar con un buscador general de vehículos, que permita realizar búsquedas por placas e IMEI.

- Debe contar con un reporte histórico que permita visualizar el recorrido de un vehículo en un rango de fechas preseleccionadas, debe mostrar el resultado en una tabla y en un mapa.
- Geocercas, debe poder permitir la creación de zonas georreferenciadas, es decir, poder delimitar una zona desde un mapa, y que el sistema alerte de la entrada y salida de un vehículo en esta zona.
- Notificaciones, debe contar con un módulo que permite avisar a los usuarios sobre algunos eventos reportados por los equipos GPS, como son:
 - Entrada y salida de Geocercas.
 - Botón de pánico.
 - Batería baja del vehículo.
 - Batería baja del equipo GPS.
 - Exceso de velocidad.
 - La notificación debe visualizarse dentro del portal web y notificación vía correo electrónico.

Los requerimientos anteriores que conforman el alcance inicial del proyecto representan a una plataforma básica de rastreo, donde, lo importante es establecer la arquitectura general y la planeación del proyecto para su entrega en tiempo y forma.

5.5 METODOLOGÍA DE DESARROLLO

Para el desarrollo de este proyecto se plantea el uso de la metodología Scrum (Gallego, 2012), la cual, es un marco de trabajo para la gestión ágil de proyectos y ¿Por qué se decidió utilizar Scrum? Pues bien, esta metodología tiene varias ventajas, por ejemplo:

- Se puede utilizar en cualquier tipo de proyecto, software (principalmente), hardware, etc.
- Se ha utilizado desde los 90 's por varias empresas importantes.
- Permite dividir un proyecto grande en etapas pequeñas y sencillas con objetivos alcanzables rápidamente, permitiendo crear entregables más pequeños con muchísimo valor para los clientes.

Me gustaría señalar que no soy experto en la metodología Scrum, sólo utilicé algunos aspectos de dicha metodología que me ayudaron a organizar, distribuir y revisar las actividades a realizar, por lo tanto, este escrito no se basa en la implementación de la metodología Scrum, se centra en explicar a grandes rasgos el desarrollo de los diferentes componentes de software utilizados en la creación de una plataforma de rastreo satelital.

Scrum tiene algunos conceptos importantes que debemos conocer y que a continuación ejemplifico para comprender mejor el por qué se decidió utilizar esta metodología.

5.5.1 Sprint en Scrum

En proyectos que siguen una metodología clásica como la metodología en cascada (Gallego, 2012), se tienen algunos pasos como:

- Análisis
- Diseño
- Desarrollo
- Pruebas
- Cierre del proyecto

Por mencionar algunas etapas, no obstante, en una metodología clásica el cliente no podrá ver algún entregable del proyecto, ya que, no existen entregas parciales que permita al cliente validar que el proyecto realmente cumplirá sus expectativas, por lo tanto, el cliente verá un entregable hasta el final del proyecto.

En Scrum existe el concepto de Sprint que nos permite dividir un proyecto grande en varias iteraciones conocidas como Sprint, por lo tanto, un Sprint es un periodo de trabajo en el cual se debe completar un requerimiento específico que en esta metodología es conocida como historias de usuario.



Figura. 2 Representación gráfica del concepto de un Sprint

Entonces cada Sprint es una iteración que se puede conformar de:

- **Épica**, es una funcionalidad por decirlo de alguna manera grande o compleja, que puede o debe descomponerse en historias de usuario pequeñas o menos complejas.
- **Historia de Usuario**, funcionalidad o requisito que puede ser finalizada en un tiempo definido ya sea en horas o días.
- **Tareas**, son pequeños incrementos de trabajo, es decir, una historia de usuario la podemos descomponer a su vez en múltiples tareas.

Por lo tanto, la jerarquía es Épicas que se descompone en Historias de usuario y una historia de usuario se descompone a su vez en tareas.

Creo que para comprender mejor esta metodología haré un ejemplo de cómo crear una historia de usuario, así que voy a tomar uno de los requerimientos, el cual, es el buscador de vehículos que mencioné en el alcance del proyecto.

Para crear una historia de usuario se sigue un formato de redacción para que todos los miembros del equipo comprendan el requerimiento y sepan qué hacer, el formato es el siguiente:

Cómo [rol del usuario] quiero [objetivo] para poder [beneficio]

Ahora en un ejemplo real tendríamos:

Cómo usuario de la página **quiero** buscar un vehículo por medio de IMEI o Placas, **para poder** conocer el último estado de un vehículo y sus datos de posición.

Esta historia de usuario sigue el formato de redacción, aunque, estrictamente no debe contener las mismas palabras **cómo**, **quiero**, **para poder**, lo que importa es que la intención se cumpla y que la historia establezca claramente el objetivo que debe cumplir.

5.5.2 Roles centrales en Scrum

En Scrum existen ciertos roles que permiten organizar al equipo, por ejemplo:

- Product Owner o dueño del producto el cual es el encargado de que el entregable final posea el mayor valor posible para el cliente, ya que, crea y mantiene el Product Backlog y es el intermediario con el cliente.
- Scrum Master es quien lidera al equipo de trabajo e identifica y elimina complicaciones u obstáculos que puedan afectar la entrega del proyecto.
- Equipo Scrum es el equipo de trabajo conformado por distintos roles como, el Product Owner, Scrum Master y el equipo de desarrollo como son programadores, diseñadores, etc.

5.5.3 ¿Qué es el Product Backlog?

El Product Backlog es uno de los artefactos pertenecientes a Scrum y es una lista ordenada y dinámica de todas las funcionalidades, requisitos, mejoras y correcciones que se deben realizar en un producto. Sirve como un único punto de referencia para todo el trabajo que debe hacerse durante el desarrollo.

5.6 EQUIPO DE TRABAJO

El equipo de trabajo se llevó a cabo con un número reducido de personas debido a temas de presupuesto, durante el desarrollo desempeñé varios roles como son:

- Levantamiento de requerimientos, visita a diferentes usuarios para comprender sus necesidades y principales requerimientos.
- Scrum Master, aun cuando la participación de otras personas fue esporádica, yo era quien los coordinaba y revisaba el trabajo realizado para incorporarlo a los componentes principales.
- Diseñador de la base de datos, creando la estructura relacional de las diferentes tablas.
- Programador, encargado de escribir el código necesario para crear los diferentes componentes de software.

En el equipo de trabajo el director de investigación y desarrollo tuvo la figura del Product Owner y es quien, hacía las pruebas de caja negra (Tejero, 2016) también, las cuales son pruebas donde el evaluador verifica que el desarrollo funcione tal como lo espera el usuario final, aún sin conocer cómo está desarrollado.

Eventualmente se contrató programadores independientes para ayudar en ciertas partes del código o crear algunas funciones en particular, aun así, la participación de dichos programadores era muy reducida y ninguno conocía el sistema en su totalidad y se enfocan en sólo desarrollar lo solicitado.

En cuanto a la administración de la base de datos y la administración de servidores, se llevó a cabo por un proveedor nacional contratado por el cliente. Su proveedor de infraestructura nos dio acceso para instalación de los diferentes componentes en producción, y después, la administración de los servidores, la administración de la base de datos y la seguridad quedó a cargo de dicho proveedor.

5.7 TECNOLOGÍA

Después de analizar y revisar a detalle el requerimiento inicial y el alcance del proyecto se decide realizar el desarrollo con las siguientes tecnologías.

5.7.1 Microsoft SQL Server

Para el desarrollo de la base de datos se utiliza este motor dado el buen rendimiento con el que cuenta, además, el cliente ya contaba con algunos desarrollos previamente creados en tecnologías Microsoft, lo que, facilitaba la integración y compatibilidad con algunos software y componentes internos de su negocio.

Microsoft SQL Server es un sistema de gestión de base de datos relacional desarrollado por la empresa de Microsoft, existen varias licencias de este gestor de base de datos que permiten su utilización, por ejemplo: Enterprise, Developer, Standard, Express, SQL

Azure, de las cuales se opta por la versión Standard (Microsoft, Prueba SQL Server en el entorno local o en la nube, s.f.).

La versión Standard cuenta con excelentes prestaciones y a pesar de que cuenta con algunas limitaciones en cuanto al número de procesadores y la agregación de memoria y CPUs, no representaba un problema dado el número de equipos GPS iniciales, esta versión es más que suficiente para iniciar el desarrollo.

5.7.2 Front End

El término Front End hace referencia a la interfaz de usuario, es decir, es la parte gráfica expuesta en el portal web a la cual todo usuario con acceso podrá ver para la utilización y manejo de la plataforma de rastreo y para poder realizar dicha página web se establece la utilización de:

- CSS, hojas de estilo en cascada, permiten crear y diseñar la presentación gráfica de archivos HTML dentro de un sitio web (Durango, 2015).
- SASS, hojas de estilo sintácticamente asombrosas permite llevar a cabo un tipo de programación en hojas de estilos para hacer el diseño de una página web aún más dinámico (SASS, s.f.).
- Bootstrap, biblioteca de elementos gráficos para la creación de interfaces de usuario (Bootstrap, s.f.).
- JavaScript, lenguaje del lado del cliente con el cual se puede crear la lógica y operaciones realizadas en una página web que se ejecuta del lado del navegador del usuario, por ejemplo, la solicitud de la información en los reportes o exportar documentos a Excel (Maza, 2012).
- JQuery es una biblioteca de JavaScript que facilita y resume el código de JavaScript en el desarrollo de algunas acciones (jQuery, s.f.).
- Google Maps, uno de los puntos más importantes en este desarrollo; esta herramienta permite a los usuarios realizar acciones sobre los equipos GPS directamente desde el mapa (Google, Maps JavaScript API, 2024).

En la primera fase no se utilizan bibliotecas de JavaScript salvo JQuery, Google Maps, toda la lógica del lado del usuario se lleva a cabo en JavaScript nativo.

5.7.3 Back End

El término Back End hace referencia a todas las operaciones que se llevan a cabo del lado de los servidores y que están restringidas para los usuarios que hacen uso de una página web, las operaciones más comunes que se llevan a cabo del lado del back end son, conexiones a base de datos, desarrollo de algoritmos sensibles que no pueden ser revelados por razones de negocio.

Dado que el cliente previamente trabajaba en tecnologías Microsoft, se decide trabajar en el ambiente de .NET, específicamente con C# .NET para el desarrollo de la lógica del lado del servidor, con lo cual, se crea una API (Interfaz de Programación de Aplicaciones), la cual, permite la comunicación entre distintos componentes de software, ya que, recibe todas las solicitudes enviadas por el Front End y devuelve los datos e información según el tipo de solicitud (Microsoft, Información general de la biblioteca de clases de .NET, 2023).

5.8 DESARROLLO E IMPLEMENTACIÓN

La implementación de los alcances de este proyecto se llevó aproximadamente 8 meses y medio, contando desde las negociaciones y cierres de contrato, demo y hasta su puesta en producción, no obstante, a la fecha de la creación de este escrito, aún continuo dando soporte y creando funcionalidades nuevas para dicha plataforma.

Dado que existe información sensible y por razones de contratos de confidencialidad en esta sección presentaré parte de la planeación y desarrollo del proyecto, sin incluir código ni imágenes de la plataforma real, los diagramas de base de datos sólo ejemplifican las relaciones de las tablas, pero no se exponen campos ni el nombre real de las tablas utilizadas, por lo tanto, el número de tablas presentadas tampoco son igual al diseño real, así como la optimización de estas.

5.8.1 Product Backlog

Para iniciar con la descripción de cómo fue creada la plataforma de rastreo, partamos revisando el Product Backlog del proyecto, cabe mencionar que el siguiente Product Backlog es un micro resumen del original y sólo muestra una Épica y algunas historias de usuario con sus tareas y tiempos estimados.

La razón de esto es que no es fácil en este escrito leer una tabla con esta información, pero es un ejemplo de cómo se fue construyendo el original, de igual manera, deseo aclarar que como yo construí prácticamente todos los componentes de software, no se trabajó en paralelo, fui avanzando y creando componentes tras componentes.

Sprint	Épicas	Historias de Usuario	Tareas	Tiempo estimado
1	Creación de un software intérprete capaz de decodificar la información recibida de los equipos a través del protocolo TCP.	La plataforma de rastreo requiere de un Servicio de Windows en .Net que siempre esté en ejecución para poder atender las conexiones de los equipos de rastreo y decodificar su información.	Crear proyecto Servicio de Windows (SW): <ul style="list-style-type: none"> Configurar y establecer variables en el archivo app.config Configurar clase Program.cs Configurar el archivo ProjectInstaller.cs 	2 h
			Crear biblioteca de clases (DLL): <ul style="list-style-type: none"> Ligar el archivo app.config del Servicio de Windows. 	1 h

			<ul style="list-style-type: none"> Agregar esta biblioteca de clases al proyecto del Servicio de Windows. 	
			Creación de clase para el manejo de Logs	2 h
		El Servicio de Windows requiere de un servidor TCP que reciba de forma concurrente la información de 10 mil equipos aproximadamente y decodificar su información en forma paralela.	Crear servicio TCP a través de TcpListener	1 h
			Implementar la recepción de datos multihilos.	2 h
			Decodificación primaria: obtención de la información básica que identifica al equipo.	1 h
		El Servicio de Windows requiere de la creación de tablas que almacenen los catálogos de los vehículos, tipos de vehículos, configuración de vehículos y usuarios para poder identificar a cada equipo que se conecta al socket TCP y validar su existencia en sistema.	Creación de las tablas, cada tabla debe contar con llave primaria, índices y deben estar normalizadas al menos hasta la 3ra forma normal. <ul style="list-style-type: none"> Crear tabla para el catálogo de usuarios. Crear tabla para el catálogo de los vehículos. Crear una tabla intermedia de usuarios-vehículos. Crear tabla configuración-vehículos. 	2 h

Figura. 3 Ejemplo del Product Backlog

Como podemos observar el exponer el Product Backlog es un poco complicado, pese a eso, lo anterior es una muestra de cómo se conformó el Product Backlog del proyecto, por lo que, optaré en explicar en los siguientes apartados las Épicas, historias de usuario, y quizá, algunas tareas, pero no en tabla, ya que es muy difícil leerlas por temas de espacio y tamaños de letra.

Durante todo el desarrollo revisaba continuamente el Product Backlog en conjunto con el Product Owner, ya que, en muchas ocasiones al desarrollar ciertas tareas nos dábamos cuenta de que había dependencias y optamos por separar y volver a definir el orden de las tareas o juntar en una tarea aquellas tareas que estaban fuertemente relacionadas.

El Product Backlog lo revisaba todos los días y realizaba mis anotaciones, pero, los cambios se actualizaban al menos una vez a la semana en conjunto con el Product Owner, así las tareas e historias de usuarios subsiguientes quedaban definidas.

5.8.2 Arquitectura general de la plataforma

A grandes rasgos el proyecto se puede diagramar de la siguiente manera

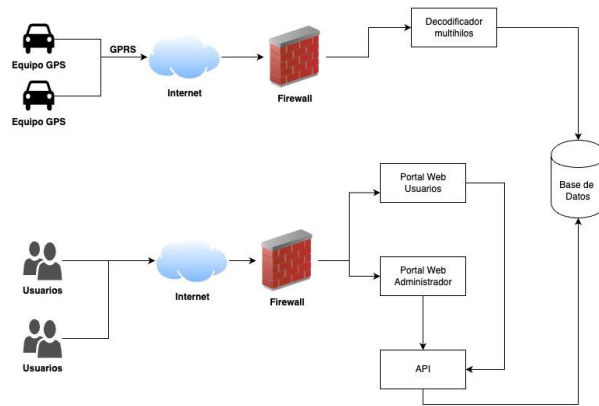


Figura. 4 Arquitectura general de la plataforma

En este diagrama se puede ver claramente cómo se comunicarán cada uno de los componentes del sistema:

- Los equipos GPS se comunicarán a través de GPRS a Internet, pasarán por un Firewall, serán decodificados por un software que se desarrollará y permitirá cientos de equipos conectados vía TCP y una vez que decodifique los paquetes de información los almacenará en una base de datos.
- Por otra parte, los usuarios se conectan a través de Internet, pasan por un Firewall y a través de sus accesos pueden conectarse ya sea al Portal Web de Usuarios o Administrador según los permisos y tipo de usuario.
- Ambos Portales Web serán capaces de comunicarse con una API y se deben autenticar a través de un token que tendrá una caducidad de cierto tiempo, una vez que el token caduque el usuario tendrá que volver hacer login para generar un nuevo token y los portales puedan volver a interactuar con el API.
- El API es el componente que tendrá acceso a la base de datos, creando una brecha de seguridad, ya que, ningún componente al que los usuarios tienen acceso puede comunicarse con la base de datos directamente.

A grandes rasgos la plataforma funciona de esta manera y se decidió esta arquitectura debido a temas de tiempos y costos, si bien es cierto, existen arquitecturas más complejas y que proporcionan un mayor rendimiento, esta arquitectura proporciona escalabilidad y rápida integración de más componentes que permitan el crecimiento del proyecto y adición de nuevas funcionalidades en el futuro.

5.8.3 Diseño de la base de datos

La base de datos se fue diseñando conforme a las historias de usuario y tareas establecidas en el Product Backlog, en los siguientes apartados explicaré un poco más este proceso, por lo pronto, muestro el diseño general de la base de datos, con las limitantes antes mencionadas, el nombre y la cantidad de las tablas no son las del proyecto original y no se muestran los campos que las conforman.

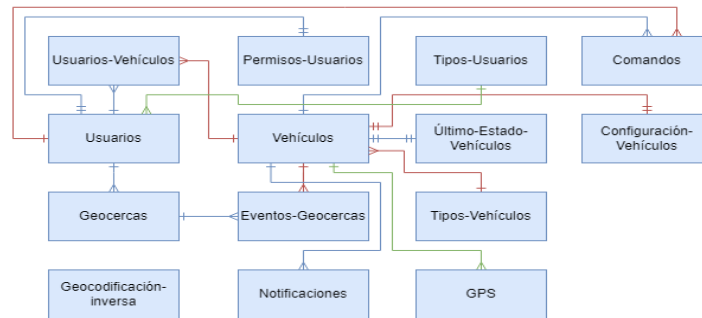


Figura. 5 Diagrama de la base de datos

5.8.4 Planeación de Sprint

El desarrollo lo he dividido en los siguientes Sprint:

- Decodificador de los paquetes enviados por los equipos de rastreo.
- Portal administrador
- Portal de usuarios

Los Sprint anteriores no son dependientes del todo, esto quiere decir que podrían hacerse en paralelo y probar en conjunto al final de cada iteración, sin embargo, el equipo de programadores no desarrollaba funcionalidades completas, es decir, solo tenían ciertas tareas asignadas, esto quiere decir que, no hubo mucho trabajo en paralelo por cuestiones de presupuesto, así que, a grandes rasgos los diferentes Sprint se desarrollaron de la siguiente manera.

5.8.5 Decodificador de paquetes

Como vimos en la arquitectura general, los equipos de rastreo se conectan a nuestro sistema por internet vía GPRS mediante el protocolo TCP, por lo tanto, este componente de software debe implementar un servidor TCP que acepte varias conexiones concurrentes y sea capaz de interpretar los paquetes, decodificarlos según el protocolo y documentación proporcionada por el fabricante y enviar la información a diferentes tablas.

5.8.5.1 Product Backlog decodificador

A continuación, muestro la parte del Product Backlog referente al decodificador de paquetes, se han hecho algunos cambios para que se entiendan mejor las historias de usuario, ya que, para resumir, se han omitido las tareas de cada historia de usuario.

Épica:

- Creación de un software intérprete capaz de decodificar la información recibida de los equipos a través del protocolo TCP.

Historias de usuario:

- La plataforma de rastreo requiere de un Servicio de Windows en .Net (Microsoft, Introducción a las aplicaciones de servicios de Windows, 2023) que siempre esté en ejecución para poder atender las conexiones de los equipos de rastreo y decodificar su información.
- El Servicio de Windows requiere de un servidor TCP que reciba de forma concurrente la información de 10 mil equipos aproximadamente y decodificar su información en forma paralela.
- El Servicio de Windows requiere de la creación de tablas que almacenen los catálogos de los vehículos, tipos de vehículos, configuración de vehículos y usuarios para poder identificar a cada equipo que se conecta al socket TCP y validar su existencia en sistema.
- El Servicio de Windows requiere decodificar el identificador del equipo conectado y validar que exista en sistema, para mantener el equipo conectado y proceder con la decodificación del paquete completo.
- El Servicio de Windows requiere decodificar cada paquete recibido en el socket TCP y recuperar la información GPS para poder calcular la geocodificación inversa.
- El Servicio de Windows requiere actualizar el último estado de un vehículo siempre y cuando la información GPS sea más reciente que el estado actual del vehículo.
- El Servicio de Windows requiere analizar la información GPS recibida para identificar si es un evento que debe tratarse como notificación o como información GPS, si es notificación almacenarla en notificaciones y enviar correo electrónico, y si no, únicamente almacenarla en GPS.
- El Servicio de Windows requiere analizar la información GPS para identificar si un equipo entra o sale de una geocerca y almacenar dichos eventos en la base de datos y enviarlos por correo electrónico.
- El Servicio de Windows requiere recibir la solicitud de un comando y canalizarlo hacia el equipo correcto, además, debe esperar la respuesta del equipo según el comando canalizado y decodificar dicha respuesta.

Tiempo estimado total:

- El desarrollo de este componente fue aproximadamente de dos meses.

5.8.5.2 Desarrollo decodificador de paquetes

Para desarrollar cada historia de usuario se requiere crear un Servicio de Windows y una biblioteca de clases para separar y tener ordenado el código, en la biblioteca de clase se escribe el código necesario para cumplir con cada historia de usuario.

5.8.5.2.1 Servidor TCP

En primer lugar, se crea un servidor TCP multihilos capaz de soportar cientos de conexiones concurrentes, el cual, obtiene el identificador del equipo:

- Realiza una primera búsqueda en memoria para saber si el equipo ya se encuentra conectado y actualiza la información del socket.
- De lo contrario, busca en la base de datos para validar si existe dicho equipo de rastreo, ¿si existe?, crea una instancia con toda la información del equipo y lo mantiene en memoria con un socket TCP abierto que sigue recibiendo paquetes provenientes del equipo.

5.8.5.2.2 Decodificación de paquetes

Cada equipo en memoria cuenta con un proceso que corre en un hilo capaz de procesar los paquetes recibidos y decodificarlos:

- Decodifica la información GPS.
- Cruza dicha información GPS mediante un algoritmo para obtener la geocodificación inversa a partir de un par de coordenadas.
- Compara la fecha de la información GPS recibida y si es más reciente que la fecha del estado actual del equipo: actualiza el último estado del vehículo con la información GPS recibida.
- Analiza el tipo de paquete recibido para saber si corresponde a un evento de botón de pánico, exceso de velocidad, etc., si corresponde, lo trata como una notificación y almacena dicha información en la tabla de notificaciones y manda un correo electrónico.
- Utilizando el paquete `Microsoft.SqlServer.Types` (Microsoft, `Microsoft.SqlServer.Types`, 2022) se crea un algoritmo capaz de realizar comparaciones entre el catálogo de Geocercas y un par de coordenadas para detectar si existe una entrada o salida a una Geocerca, además, se envía un correo para notificar dichos eventos y se almacenan en la tabla de Eventos-Geocercas.
- Si la información GPS no corresponde a ningún evento de notificación, almacena dicha información en la tabla GPS.

5.8.5.2.3 Envío de comandos desde el decodificador

Cada equipo en memoria cuenta con un segundo proceso capaz de recibir solicitudes de comandos, dicho proceso recibe un comando e identifica a qué equipo va dirigido, realiza una búsqueda en los equipos en memoria para saber si puede o no enviar el comando:

- Si el equipo se encuentra conectado, crea un nuevo comando según la información del fabricante y lo manda a una cola de datos para poder ser enviado posteriormente a través del socket TCP. En este punto el comando se marca en la tabla Comandos como “En proceso”.
- Una vez que el comando es enviado al equipo a través del socket TCP, el sistema marca el comando como “Enviado” en la tabla de Comandos, de existir un error de envío, el comando es marcado como “No enviado.”
- El sistema queda al pendiente a la espera de la respuesta del equipo según el tipo de comando enviado, no obstante, no afecta a la decodificación de paquetes. Si recibe respuesta según el tipo de comando, marca el comando en la tabla de Comandos como “Confirmado”.
- Si el equipo no se encuentra conectado, simplemente desecha el comando y marca en la tabla de Comandos que el equipo no se encuentra conectado.

Lo anterior resume a grandes rasgos cómo funciona el decodificador de paquetes, obviamente cada historia de usuario tiene algunas tareas que fueron dividiendo el trabajo en tareas más pequeñas, por ejemplo:

- A uno de los programadores eventuales se le compra parte de la información que conforma las tablas de la geocodificación inversa y se le solicita la información para poder explotar dichos datos.
- Dicho programador nos da un respaldo de la información, el cual, es montado en un ambiente de desarrollo local, se analiza se adecua y se procede a montarlo en la infraestructura del cliente.
- Se procede a crear el algoritmo que permite crear la geocodificación inversa.
- Durante el desarrollo veo que muchos resultados de la geocodificación inversa son poco precisos, por lo que procedo manualmente a cargar nuevas direcciones, a través de diversas fuentes, algunas de pago y otras gratuitas.

En resumen, el decodificador de paquetes escribe la información que interpreta y analiza en la base de datos, para posteriormente ser consultada por los demás componentes de la plataforma de rastreo.

5.8.6 Portal Web administrador

Como muestra la arquitectura general de la plataforma, los usuarios pueden acceder al portal web administrador desde internet, por lo que, se expone una página web a través de un servidor web Internet Information Services (IIS) (Microsoft, Microsoft Learn

Challenge, 2024), ahora bien, para poder acceder al portal se debe contar con los permisos necesarios.

5.8.6.1 Product Backlog administrador

A continuación, muestro la parte del Product Backlog referente al administrador, se han hecho algunos cambios para que se entiendan mejor las historias de usuario, ya que, para resumir, se han omitido las tareas de cada historia de usuario.

Épica:

Creación de página web que permita el acceso sólo a ciertos usuarios y les permita dar de alta, editar o desactivar usuarios, así como editar los permisos y los vehículos a los que tendrán acceso.

Historias de usuario:

- Cómo usuario del administrador quiero una pantalla de login para poder ingresar al administrador a través de un usuario y contraseña para poder administrar usuarios y sus vehículos vinculados.
- Cómo usuario del administrador quiero en la pantalla de login la opción de poder recuperar mi contraseña en caso de que no la recuerde, y así poder recuperar el acceso al portal.
- Cómo usuario del administrador quiero que al ingresar muestre sólo los usuarios que he creado y me permita visualizar los vehículos que tienen asignados.
- Cómo usuario del administrador quiero crear usuarios nuevos, así como poder editar su información y darlos de baja.
- Cómo usuario del administrador quiero editar los permisos de mis usuarios para permitir o limitar el acceso a los distintos módulos de la plataforma de rastreo.
- Cómo usuario del administrador quiero cambiar la contraseña de mis usuarios.
- Cómo usuario del administrador quiero vincular o desvincular los vehículos a los que tendrán acceso los usuarios.
- Cómo usuario del administrador quiero una opción para poder cerrar mi sesión actual y salir del portal.

Tiempo estimado total:

- El desarrollo de este componente fue aproximadamente de un mes.

Para más detalles de cómo se desarrolló cada historia de usuario, el portal web administrador se dividió en dos componentes:

- Una página web

- Y una API REST (Microsoft, Crear API RESTful con ASP.NET Web API, 2024) la cual permite la comunicación entre dos componentes de software, en este caso, permite la comunicación entre la página web del administrador y la base de datos.

5.8.6.2 Desarrollo portal administrador

La página web del administrador se creó a partir de tecnologías como Javascript, JQuery, Bootstrap, etc., tal como mencioné en el apartado 5.7.2 de Front End.

La API se creó a partir de tecnologías .NET con el lenguaje C# y se expusieron una serie de endpoints, los cuales, son distintas URL que se encargan de responder, para este caso, a una solicitud originada en la página web.

5.8.6.2.1 Login del administrador

Ahora vayamos a enumerar algunas de las tareas que debe contemplar el desarrollo de este Login:

1. La creación del diseño e interfaz de usuario debe contemplar:
 - a. Una entrada (input) para ingresar el correo electrónico del usuario.
 - b. Una entrada (input) para ingresar la contraseña del usuario registrado.
 - c. Un botón para que el usuario al dar clic acceda al sitio.
 - d. Debe contar con label (o etiquetas) que dé información al usuario y ayude a comprender para qué sirve cada elemento.
2. El usuario debe acceder al dar clic en el botón de acceder o al oprimir la tecla enter.
3. Se debe contar con la opción de recuperar la contraseña en caso de que el usuario llegue a extraviarla.

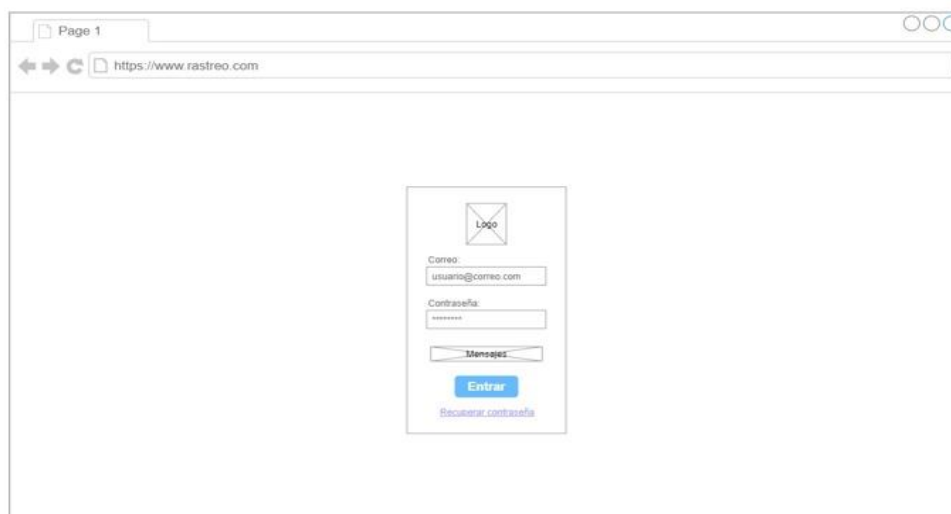


Figura. 6 Layout (Mockup) Login

Respecto al funcionamiento del login, lo podemos describir de la siguiente manera:

- Se crea una pantalla donde el usuario podrá ingresar un usuario y contraseña y al dar clic en un botón o presionar la tecla enter, la página lanza una solicitud tipo POST hacia un endpoint de la API, enviando el usuario y contraseña que se han ingresado, para que éstos sean validados en la API y regrese un token.
- La API recibe en su endpoint la solicitud originada desde la página web y toma el dato usuario y contraseña para realizar una consulta en la tabla de usuarios, si los datos son correctos y el usuario se encuentra activo, regresa un token hacia el cliente.
- La página web recupera el token retornado por la API y lo utiliza en todas las siguientes solicitudes que realice hacia la API, es decir, en toda solicitud subsiguiente envía este token en la cabecera de cada solicitud y la API se encargará de revisar si el token es válido.
- La pantalla de login también cuenta con un link para recuperar la contraseña, cuando el usuario da clic sobre el link o enlace, lo lleva a una página donde se le solicita que ingrese su correo. Después de que el usuario ingresa a su correo, la página lanza una nueva solicitud a la API con el token en la cabecera de la solicitud y con el dato del correo.
- La API recibe la solicitud para la recuperación de contraseña y revisa que el token sea válido y que el correo está registrado en la tabla Usuarios. Si el correo es válido, se envía a dicho correo un link para que el usuario pueda establecer una nueva contraseña.
- Cuando el usuario da clic en link recibido en su correo, éste lleva al usuario a la pantalla para establecer contraseña, la página con la información de la URL proporcionada por el link realiza una solicitud a la API para comprobar que el link es válido y el usuario existe en el sistema.
- Si la API regresa que el usuario existe, la página le permite al usuario ingresar una nueva contraseña y se solicita que la confirme, después da enter o clic en el botón del formulario y la página lanza a la API el cambio de contraseña, la API recibe esta solicitud, valida la información y actualiza la contraseña en la base de datos.

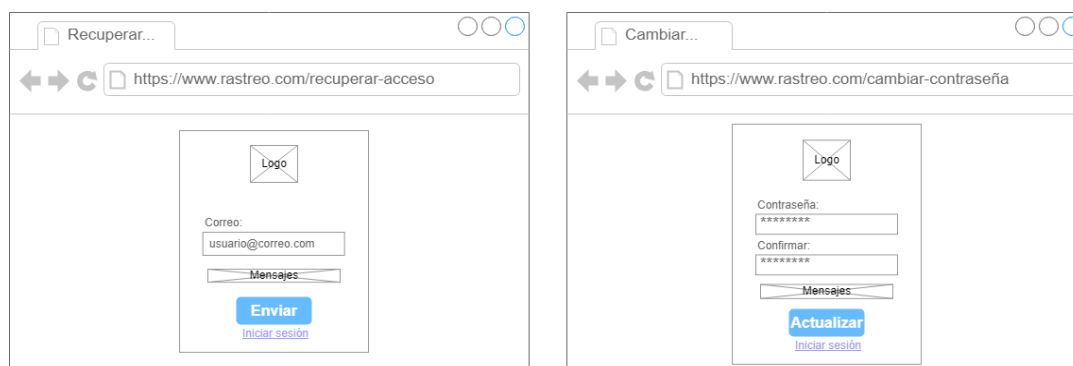


Figura. 7 Layout (Mockup) recuperar contraseña

5.8.6.2.2 Administración usuarios

En cuanto a la pantalla que permite visualizar la lista de usuarios creados, podemos describirla de la siguiente manera:

- Al igual que en la pantalla del login, toda solicitud que lanza la página web hacia la API se agrega en la cabecera de la solicitud el token generado durante el inicio de sesión.
- Pantalla de usuario:

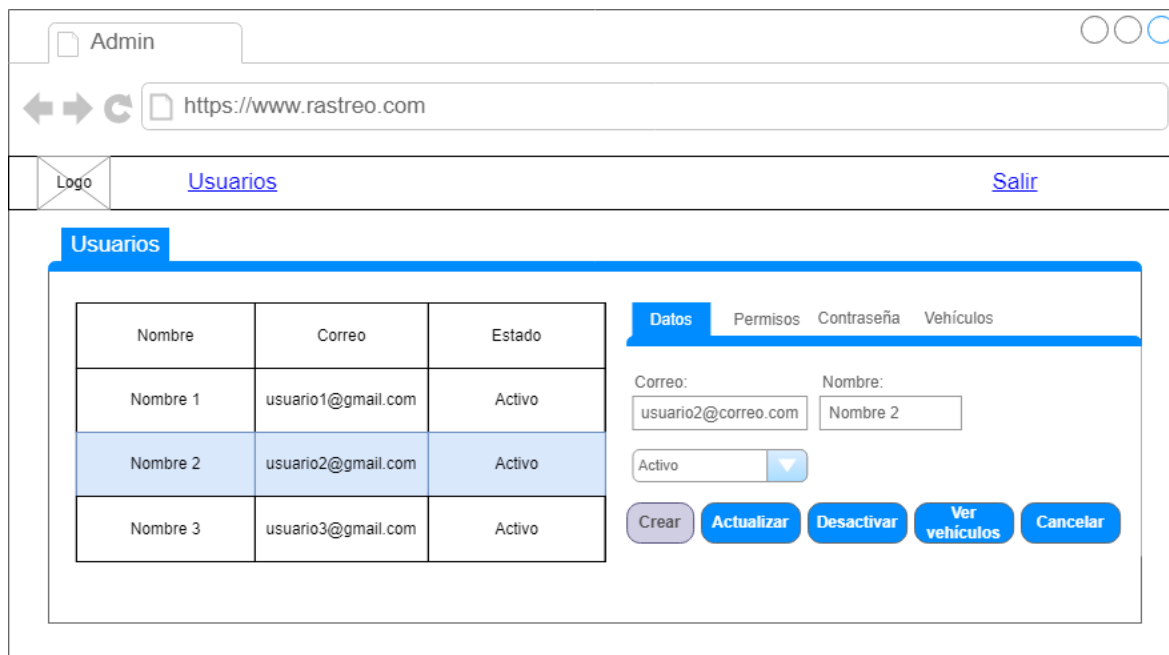


Figura. 8 Layout (Mockup) Pantalla de usuarios

- La pantalla de usuarios cuenta con una tabla que muestra una lista de usuarios perteneciente al usuario actual.
- La tabla permite seleccionar un usuario de la lista y permite editar su información o desactivarlo.
- Para crear un usuario no debemos tener ningún usuario seleccionado, se puede dar clic en el botón de Cancelar, en dado caso que se tenga un usuario seleccionado, introducir los datos que se solicitan y dar clic en Crear, después de ser creado, puedes editar los permisos, contraseña y vincular vehículos.
- Al dar clic en el botón Ver vehículos se muestra una tabla con los vehículos que el usuario actualmente tiene vinculados.

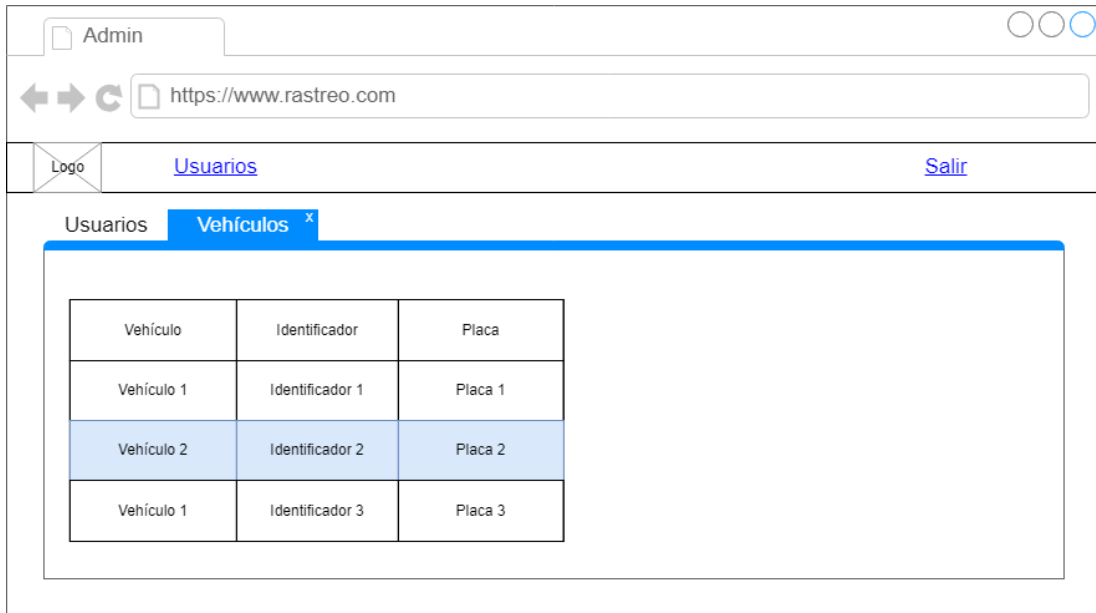


Figura. 9 Layout (Mockup) Vehículos vinculados a un usuario

- En esta misma pantalla podemos modificar los permisos que tendrá el usuario, esta información se almacena en la tabla Permisos-Usuarios.

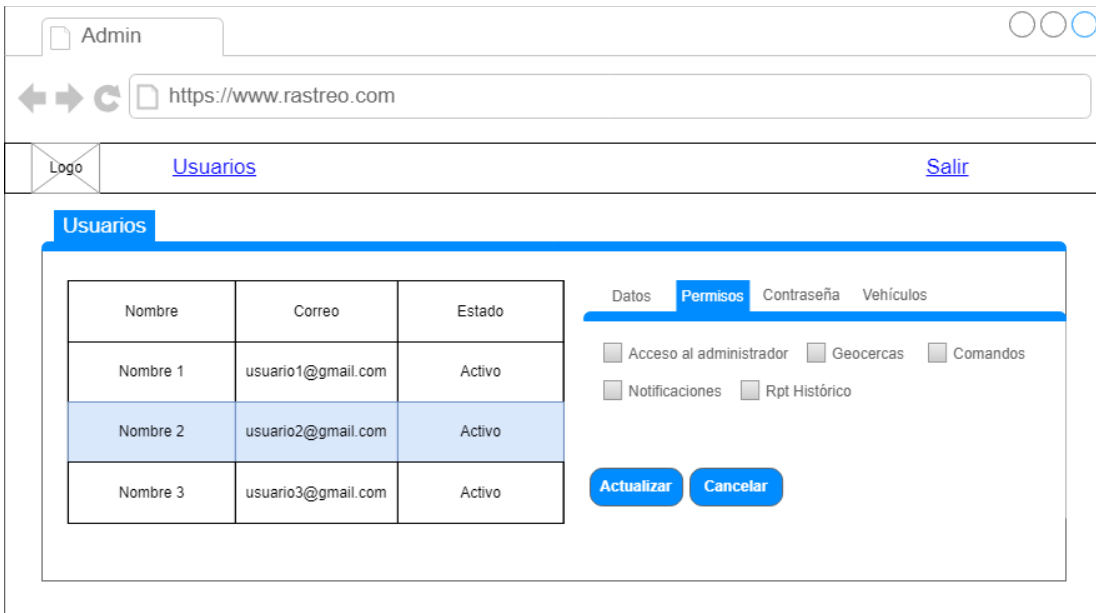


Figura. 10 Layout (Mockup) Permisos de usuario

- Otra función importante es que permite la actualización de la contraseña para un usuario, el usuario introduce una contraseña y su confirmación, finalmente la manda a la API para realizar el cambio en la base de datos, claro que hay validaciones tanto en Front End como en Back End.

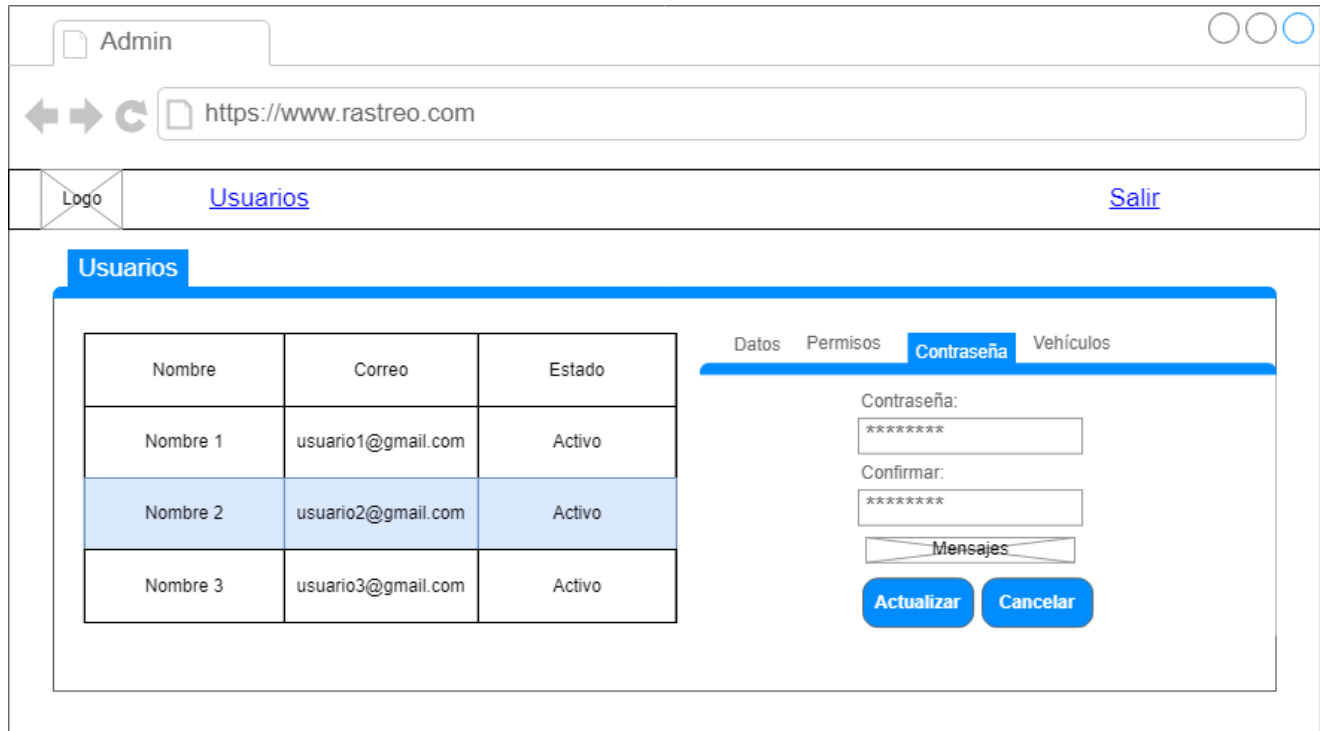


Figura. 11 Layout (Mockup) Cambiar contraseña a usuario

- Finalmente podemos llevar a cabo la vinculación de vehículos al usuario, para poder llevar a cabo esta función, previamente se llevó a cabo una carga manual del catálogo de vehículos.
- Por lo tanto, el usuario solo debe cargar los identificadores de los vehículos y vincularlos o desvincularlos del usuario.
- El sistema valida si el usuario que realiza la acción tiene acceso a los vehículos que desea vincular o desvincular, después existen validaciones para saber si el usuario que se desea alterar ya cuenta con los vehículos vinculados o no, dependiendo de la acción solicitada, entre muchas más validaciones.

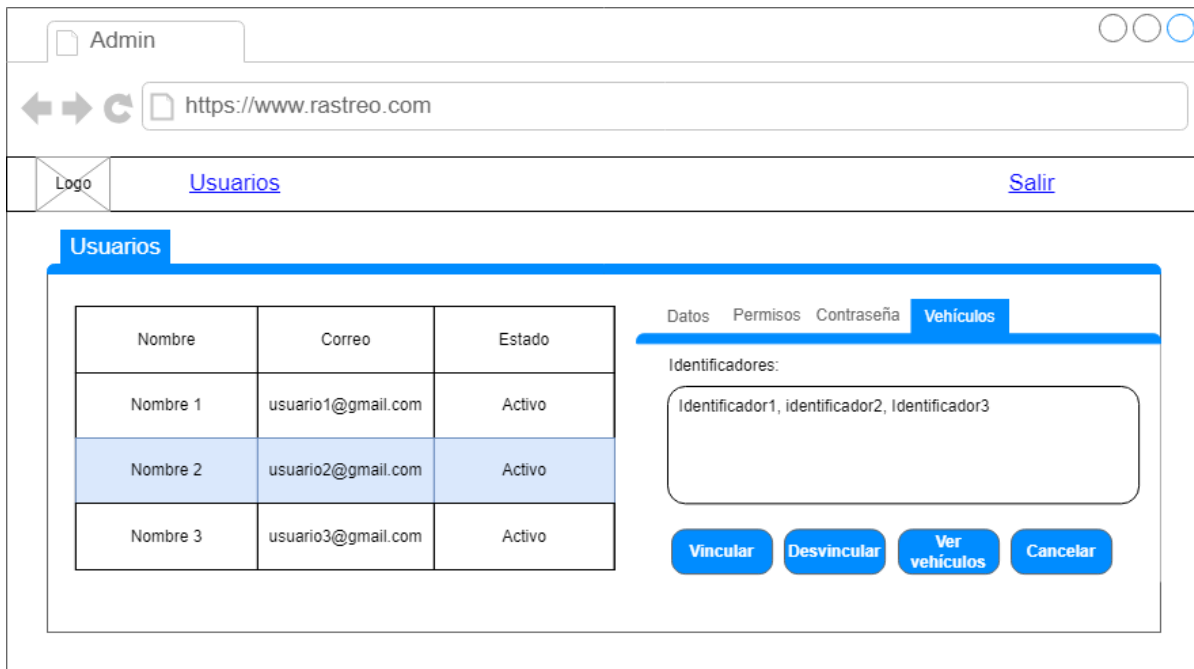


Figura. 12 Layout (Mockup) Vincular o desvincular vehículo a usuario

En esencia es todo lo que permite hacer el portal administrador, claro que cada funcionalidad en el administrador requiere de un endpoint del lado de la API, por lo tanto, fui desarrollando la página y la API al mismo tiempo, así como la creación de las tablas:

- Usuarios-Vehículos
- Permisos-Usuarios
- Tipos-Usuarios

Para cada funcionalidad se llevaron a cabo pruebas de caja negra, de las cuales el director de investigación y desarrollo era el responsable de aplicarlas y ejecutarlas.

5.8.7 Portal Web de usuarios

La segunda página web que compone esta plataforma de rastreo es el portal web de usuarios a la que accederán los usuarios finales, dicha página debe ser capaz de permitir realizar la mayor parte las tareas que se plantearon en el alcance del proyecto. Cabe mencionar que cada módulo se mostrará según los permisos del usuario.

5.8.7.1 Product Backlog portal de usuarios

A continuación, muestro la parte del Product Backlog referente al portal de usuarios, se han hecho algunos cambios para que se entiendan mejor las historias de usuario, ya que, para resumir, se han omitido las tareas de cada historia de usuario.

Épica:

- Creación de página web que permita a los usuarios dar seguimiento a su flota de vehículos, debe permitir visualizar los vehículos en un mapa, además de poder visualizar la información GPS, envío de comandos a los equipos de rastreo, un buscador de vehículos, reporte de recorrido histórico por vehículo, creación de geocercas, visualizar las entradas y salidas de geocercas, así como las notificaciones.

Historias de usuario:

- Cómo usuario de la página quiero una pantalla de login para poder ingresar a través de un usuario y contraseña para poder dar seguimiento a mis vehículos.
- Cómo usuario de la página quiero en la pantalla de login la opción de poder recuperar mi contraseña en caso de que no la recuerde, y así poder recuperar el acceso al portal.
- Cómo usuario de la página quiero que al ingresar solo muestre aquellas secciones y pantallas a las que tenga acceso según mi rol de usuario para poder tener acceso únicamente a las opciones, información y pantallas adecuadas a mi perfil de usuario.
- Cómo usuario de la página quiero que al seleccionar un vehículo pueda visualizar la información del equipo GPS, Posición (latitud, longitud), información del vehículo (placas, modelo, marca, etc.), estado de ignición, etc, para poder validar y comprobar la última posición del vehículo y el estado del mismo.
- Cómo usuario de la página quiero que al seleccionar un vehículo muestre la dirección postal del lugar donde se encuentra el vehículo, para poder reconocer fácilmente el sitio geográfico y alrededores.
- Cómo usuario de la página quiero seleccionar un vehículo y poder editar su información, respecto a placas, marca, modelo, versión, etc.

- Cómo usuario de la página necesito poder enviar comandos a los equipos instalados en los vehículos, para poder habilitar o deshabilitar sistemas del vehículo.
- Cómo usuario de la página quiero buscar un vehículo por medio de IMEI o Placas, para poder conocer el último estado de un vehículo y sus datos de posición.
- Cómo usuario de la página quiero que al seleccionar un vehículo me permita visualizar su recorrido histórico, para poder validar y revisar todos los lugares por los que ha transitado en un rango de fechas específicas, el resultado lo necesito visualizar en una tabla de datos y en un mapa.
- Cómo usuario de la página necesito poder limitar un área geográfica a través de puntos georreferenciados, para poder saber si un vehículo entra o sale de dicha área geográfica.
- Cómo usuario de la página quiero recibir notificaciones vía correo electrónico y cuando esté dentro del portal web, para poder saber en qué momento exacto se presenta la activación de un evento de exceso de velocidad, entrada y salida de geocercas, batería baja, etc.

Tiempo estimado total:

- El desarrollo de este componente fue aproximadamente de tres meses.

Para más detalles de cómo se desarrolló cada historia de usuario, el portal web de usuarios se dividió en dos componentes:

- Una página web
- Y una API REST, la cual es la misma API REST que utiliza el portal web administrador, sólo se agregan nuevos endpoints para atender las solicitudes de este nuevo componente.

5.8.7.2 Desarrollo portal administrador

El portal web de usuarios se creó con la misma tecnología del portal web administrador, por lo que, se reutilizaron componentes de la interfaz de usuario, como botones, tablas, etc.

5.8.7.2.1 Login del portal de usuarios

El login del portal usuario tiene el mismo comportamiento que el login del portal administrador, por lo que, omitiré la descripción de su funcionamiento en esta sección.

5.8.7.2.2 Pantalla principal de monitoreo

El portal de usuarios cuenta con una pantalla de monitoreo, y que, a su vez, es la pantalla principal, donde los usuarios pueden realizar prácticamente todas las acciones disponibles, ya que, a través de JavaScript y JQuery, se crea dinámicamente la interfaz de usuario de las diferentes secciones según cómo los usuarios lo vayan requiriendo:

- Al igual que en el portal administrador, toda solicitud que lanza la página web hacia la API se agrega en la cabecera de la solicitud el token generado durante el inicio de sesión, así como los parámetros que requiere cada endpoint.
- Pantalla principal de monitoreo:

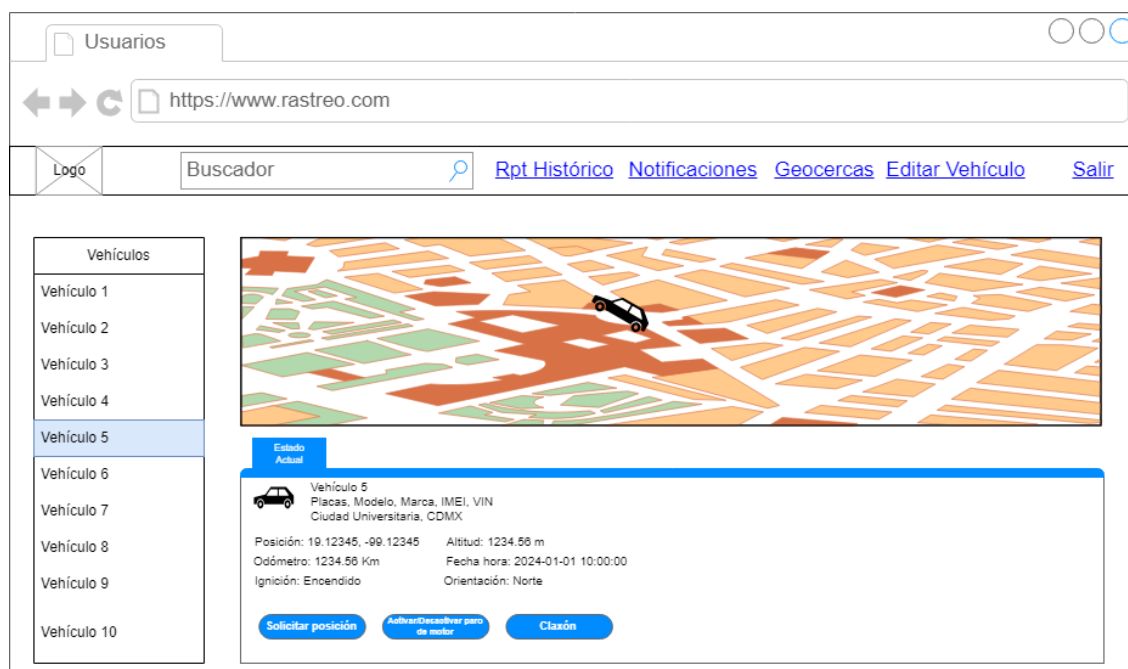


Figura. 13 Layout (Mockup) Pantalla Principal de monitoreo

- La pantalla principal de monitoreo es lo primero que se le muestra al usuario cuando realiza un inicio de sesión, la página valida los permisos que posee el usuario y solo muestra aquellas secciones permitidas según sus permisos configurados desde el portal web administrador.
- En esta pantalla el usuario puede seleccionar un vehículo de la lista y en el panel inferior se mostrará la información GPS del vehículo, la geocodificación inversa y las opciones para enviar comandos. La página tiene un timer que cada minuto hará un llamado a la API para actualizar la información en pantalla.
- De igual forma el vehículo seleccionado se mostrará en el mapa y su posición será actualizada con el llamado a la API. Con esto el usuario podrá dar seguimiento a sus vehículos, ya que, podrá conocer la última posición reportada.

5.8.7.2.3 Envío de comandos desde el portal de usuarios

Desde la pantalla principal podemos enviar comandos el vehículo seleccionado, si observamos el Mockup de la figura 13, en el panel inferior, existen tres botones:

- Uno para solicitar la última posición del vehículo.
- Otro para realizar un paro de motor o desactivar el paro de motor.
- Y por último activar el claxon.

Cuando la página envía un comando, hace una llamada a la API:

- La API valida el tipo de comando y lo almacena en la tabla de comandos.
- Después, se comunica con el decodificador de paquetes y le envía el comando.
- Finalmente, el encargado de enviar los comandos a los equipos es el decodificador de paquetes, y eso lo explico en el punto 5.8.5.2.3 Envío de comandos desde el decodificador.

5.8.7.2.4 Manejo de mapas y cartografía

El manejo de mapas es uno de los puntos más relevantes en este tipo de plataformas, es por ello que mostraré un ejemplo de cómo se puede incrustar un mapa en un sitio web, ya que es la base que se ha tomado para la creación de esta plataforma de rastreo y que es parte fundamental de la pantalla principal de esta página.

Para poder utilizar la cartografía de Google y sus APIs (Google, Maps JavaScript API, 2024) debemos tener una licencia, en este caso el cliente ha tramitado su licencia que le permite acceder a prácticamente todas las funcionalidades de Google Maps, sin embargo, para los desarrolladores se puede conseguir una licencia a través de la obtención de una Key que debemos agregar a nuestro código para poder hacer uso de la cartografía, toda la información se puede encontrar en la siguiente URL. <https://developers.google.com/maps/documentation/javascript/get-api-key?hl=es>

Siguiendo los pasos que se establecen en la anterior URL, Google nos proporciona una clave con la cual podemos hacer pruebas e incluso poner en producción nuestro desarrollo bajo ciertas limitaciones (Google, Usa claves de API, 2024).

```
<script Async Defer  
src=https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&callback=initMap  
type="text/javascript"></script>
```

La anterior línea de código nos permite tener acceso a la cartografía de Google Maps en donde:

- **YOUR_API_KEY**, es la clave de acceso que nos proporciona Google.

- **iniMap**, es el nombre que elegimos a la función que iniciará nuestro mapa.
- Para que quede más claro aquí dejo un ejemplo completo de cómo se implementa en un archivo HTML (Google, Cómo agregar un mapa de Google Maps con marcadores usando HTML, 2024).

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
5      <meta charset="utf-8" />
6      <title>Simple Markers</title>
7      <style>
8          /* Estilos para el contenedor del mapa */
9          #map {
10             height: 100%;
11         }
12         /* Opcional: Estilos generales del documento HTML */
13         html,
14         body {
15             height: 100%;
16             margin: 0;
17             padding: 0;
18         }
19     </style>
20 </head>
21 <body>
22     <div id="map"></div>
23     <script>
24         function initMap() {
25             var myLatLng = { lat: 19.432614, lng: -99.132313 };
26
27             var map = new google.maps.Map(document.getElementById("map"), {
28                 zoom: 4,
29                 center: myLatLng
30             });
31
32             var marker = new google.maps.Marker({
33                 position: myLatLng,
34                 map: map,
35                 title: "Hello World!"
36             });
37         }
38     </script>
39     <script
40         async
41         defer
42         src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
43     ></script>
44 </body>
45 </html>

```

Figura. 14 Código de implementación de un Mapa de Google en un documento HTML

En la Figura 14 podemos visualizar una estructura HTML básica de una página web y lo que podemos destacar de dicho código es que muestra un mapa y lo hace de la siguiente manera:

- Cuando la página carga se ejecuta la función **initMap**, dentro de esta función se crea el objeto de mapa que se mostrará al usuario.
- Este objeto se crea a partir de la línea de código **google.maps.Map(id,opciones)**, donde id es el identificador único del div donde queremos embeber el mapa y opciones es un objeto JSON que contiene las características con las que deseamos que se muestre el mapa.
- En el ejemplo el id es un div que gracias a los estilos tomará todo el ancho y alto de la pantalla del navegador.
- Las opciones son un objeto JSON con las características iniciales que en este caso son:
 - ZOOM: que tan alejada carga la vista del mapa al inicio.
 - Center: es un par de coordenadas que indican el centro del mapa.
- Por último, se tiene un objeto más que recibe el nombre de marker, el cual crea un objeto sobre el mapa, que en nuestro caso serán los vehículos para rastrear y cada marker representa la posición que envían los rastreadores GPS.
- Un vehículo lo podemos representar en el mapa a través de la línea de código **new google.maps.Marker(opciones)**, que recibe como opciones:
 - Position: par de coordenadas latitud y longitud
 - Map: objeto del mapa donde se mostrará el marker
 - Title: texto que describa e identifique al marker

El ejemplo anterior muestra la forma más simple en que se incrusta un mapa en un archivo HTML, para la creación de la plataforma de rastreo se hizo lo anterior, pero de una manera dinámica; es decir, en vez de un solo marker se muestran varios marker que son solicitados a la API según el usuario que haya iniciado sesión, la API regresa la información que se debe pasar a los objetos de Google Maps y a través de JavaScript se manipulan para solicitar información adicional o generar reportes de un vehículo en particular.

5.8.7.2.5 Editar información de un vehículo

Desde la pantalla principal lo usuarios pueden editar cierta información como las placas, el modelo, la marca y versión de un vehículo de la siguiente forma:

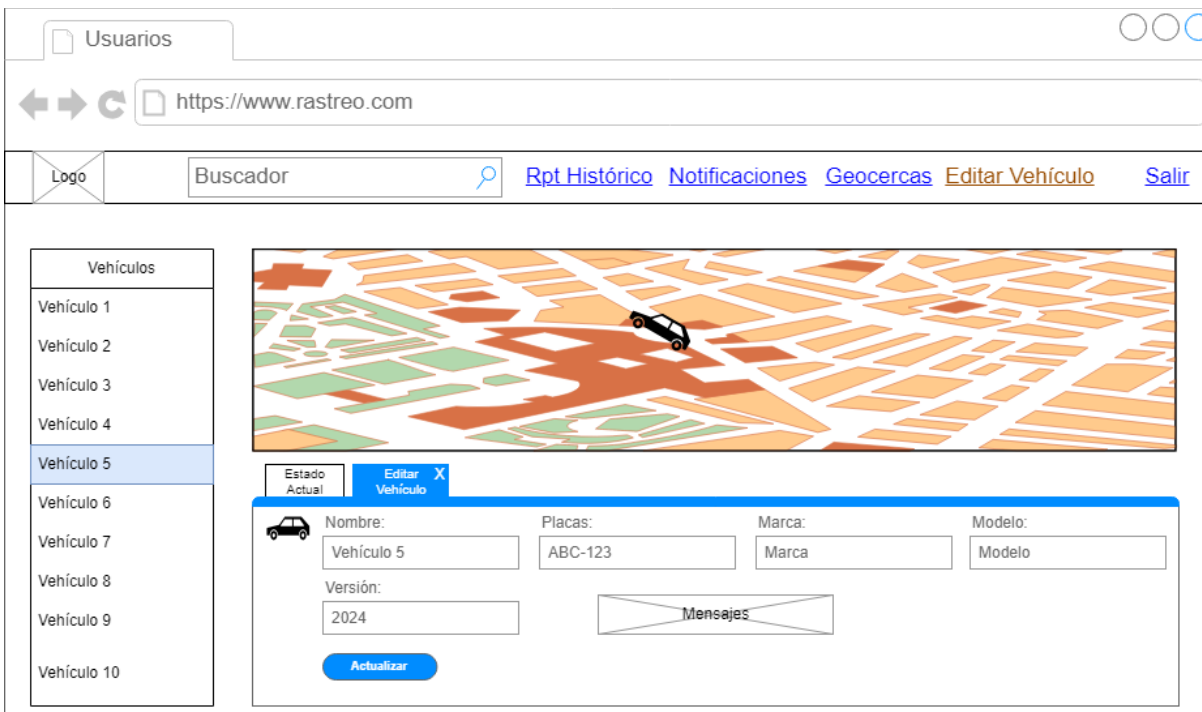


Figura. 15 Layout (Mockup) Editar Vehículo

- Sólo se debe dar clic en el menú superior que indica editar vehículo.
- Se abrirá un nuevo TAB en el panel inferior con la información actual del vehículo y podrás editar su información.
- Una vez que el usuario haya modificado la información que desea, sólo da clic en actualizar y la página enviará la nueva información a la API.
- La API llevará a cabo las validaciones necesarias y actualizará la información en la tabla de Vehículos en la base de datos.
- La opción de creación de nuevos vehículos no existe, ya que, se llevó a cabo una carga masiva de vehículos directamente en la base de datos que el cliente tenía en su anterior plataforma de rastreo, y se planteó que deseaba crear los vehículos desde otro sistema independiente que utiliza en la operación de su negocio, aunque, esta integración no forma parte en la creación de la plataforma de rastreo, ya que, su sistema debía desarrollar esta nueva funcionalidad y una vez desarrollada, se tenía que comunicar con nuestra API.

5.8.7.2.6 Buscador de vehículo

El buscador se encuentra en la parte superior de la pantalla, aquí podemos introducir un IMEI o una placa para realizar la búsqueda:

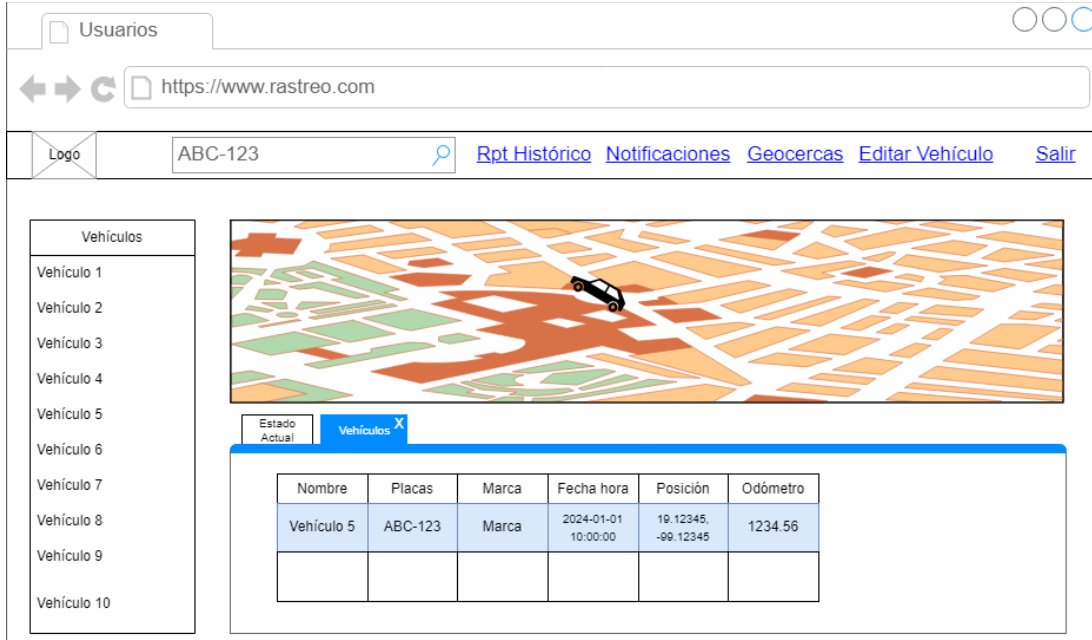


Figura. 16 Layout (Mockup) Buscador de Vehículos

- La página envía el parámetro de búsqueda a la API.
- La API hace un cruce de información en la base de datos y realiza una búsqueda de los vehículos vinculados al usuario en las tablas Vehículos y Usuarios-Vehículos, realizando una condicional mediante el operador LIKE de SQL Server para seleccionar aquellos vehículos cuyo IMEI o Placa coincida con el parámetro de búsqueda.
- Una vez que obtiene los datos, regresa las coincidencias a la página web de usuarios.
- La página web recibe la respuesta de la API y muestra los resultados en un nuevo TAB del panel inferior, así el usuario puede elegir el vehículo, y esta selección funciona igual que si seleccionará un vehículo de la lista de vehículos del panel izquierdo.

5.8.7.2.7 Reporte histórico

La página permite generar un reporte de las posiciones históricas de un vehículo que se encuentra actualmente seleccionado:

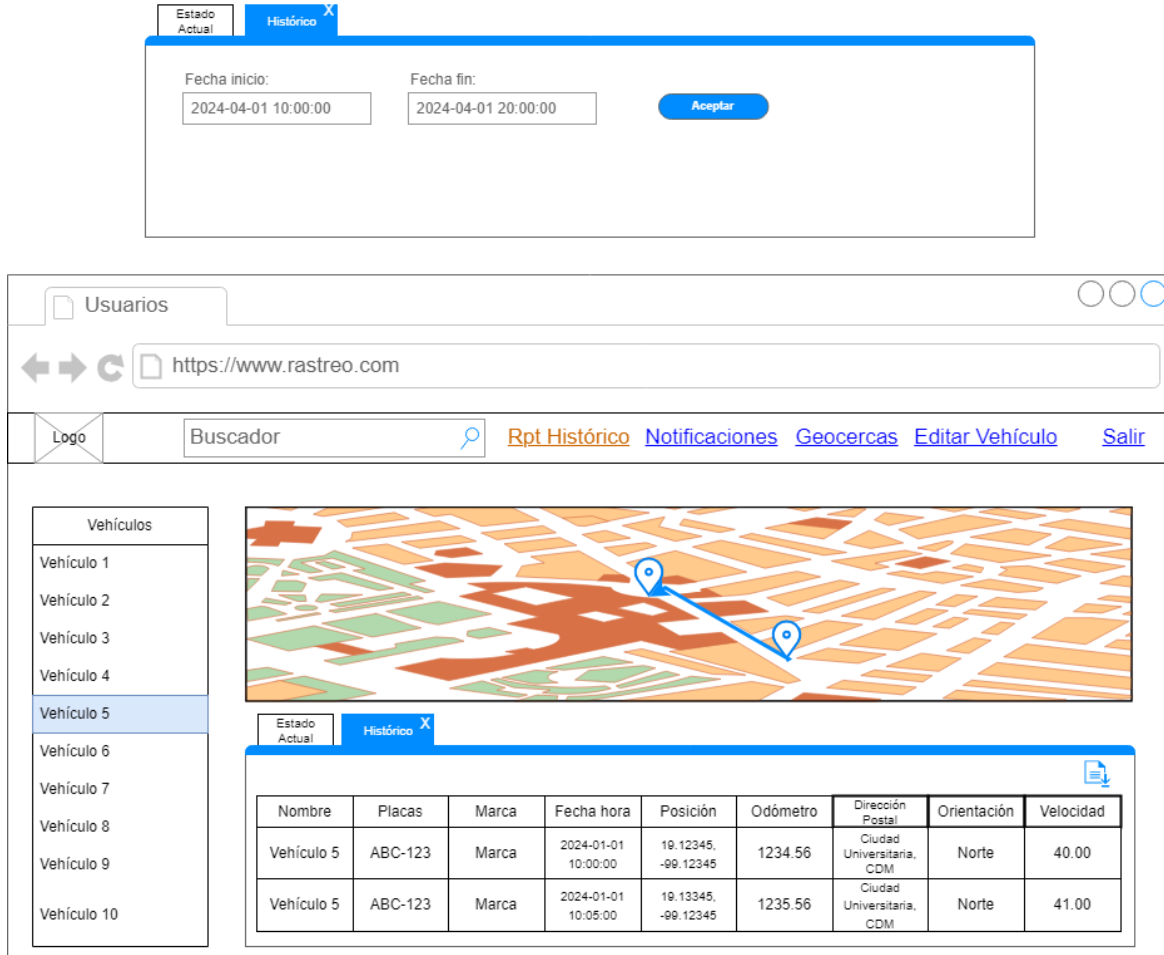


Figura. 17 Layout (Mockup) Pantalla Reporte Histórico

- Sólo se debe dar clic en el menú superior Rpt Histórico y se abrirá un nuevo TAB en el panel inferior.
- El nuevo TAB te solicitará una fecha y hora inicial y una fecha y hora final.
- La página llama a la API con los parámetros introducidos y realiza ciertas validaciones, por ejemplo, valida si el equipo le pertenece al usuario, fechas válidas, etc.
- Lo importante es que al final de las validaciones se realiza una consulta en la tabla GPS, donde se encuentran los datos históricos de la información GPS, extrayendo los datos del vehículo en el rango de fechas indicado.

- La API regresa los datos a la página web y ésta presenta los datos en el mismo panel en forma de una tabla y permite al usuario exportar esta tabla a Excel. También le permite cerrar el panel para poder realizar otro reporte y seleccionar un nuevo vehículo.
- Además, los datos regresados son mostrados en el mapa para que el usuario pueda ver fácilmente cómo se estaba trasladando su vehículo.

5.8.7.2.8 Administración de geocercas

En la pantalla principal el usuario puede crear áreas georreferenciadas o geocercas que representan zonas, lugares o centros de distribuciones; depende de cada usuario, lo importante es que cada geocerca le permitirá saber al usuario cada vez que un vehículo entra o sale de ella.

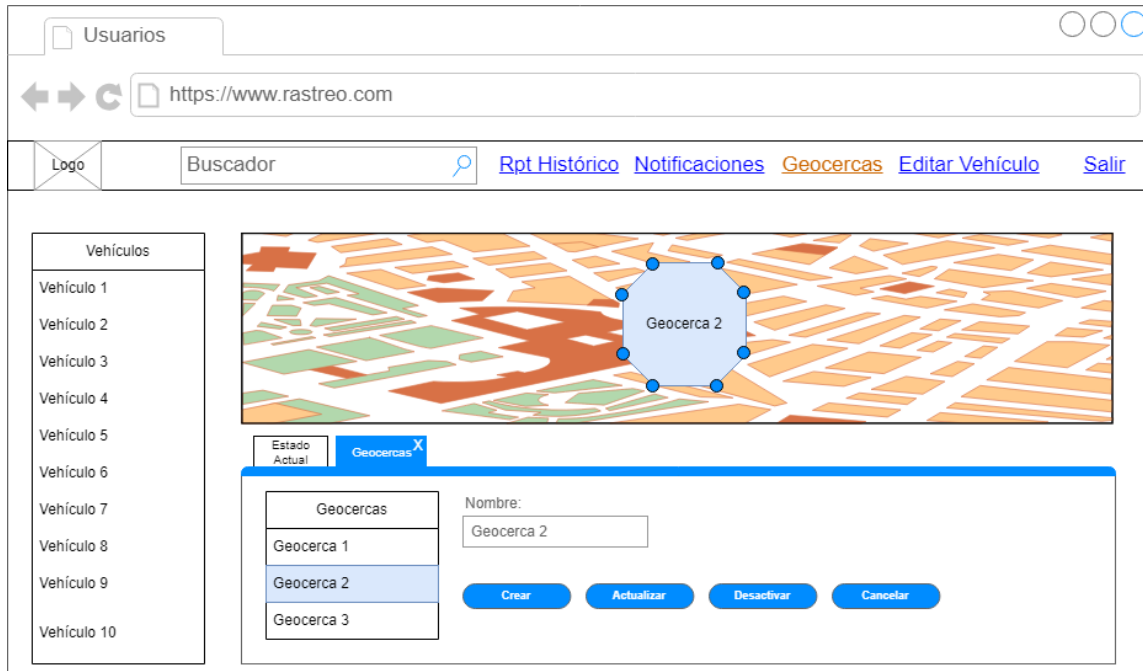


Figura. 18 Layout (Mockup) Administración de Geocercas

- Si el usuario da clic en el menú superior de Geocercas, se abrirá un nuevo TAB en el panel inferior, donde podrá administrar sus geocercas, ya sea: crear nuevas, editarlas o desactivarlas.
- El nuevo TAB llamará a la API para solicitar las geocercas que el usuario haya creado previamente o puede comenzar a crear otras.
- Tanto en el proceso de creación como en la edición, el usuario puede crear o modificar la geocerca directamente en el mapa.

- Una vez que el usuario tenga lista la geocerca, la página llamará a la API y le pasará los parámetros de la geocerca, la API llevará a cabo ciertas validaciones antes de almacenar la geocerca en la tabla de Geocercas.
- Las geocercas creadas por los usuarios son utilizadas por el decodificador de paquetes para que cada posición recibida de los equipos sea comparada con el catálogo de geocercas y obtener los eventos de entrada y salida de geocercas.

5.8.7.2.9 Notificaciones

En la pantalla principal los usuarios pueden visualizar los eventos pertenecientes a una notificación, como son:

- Entrada y salida de Geocercas.
- Botón de pánico.
- Batería baja del vehículo.
- Batería baja del equipo GPS.
- Exceso de velocidad.

Por lo tanto, si el usuario da clic en el menú superior Notificaciones, se abrirá un nuevo TAB en el panel inferior donde podrá visualizar los eventos recibidos:

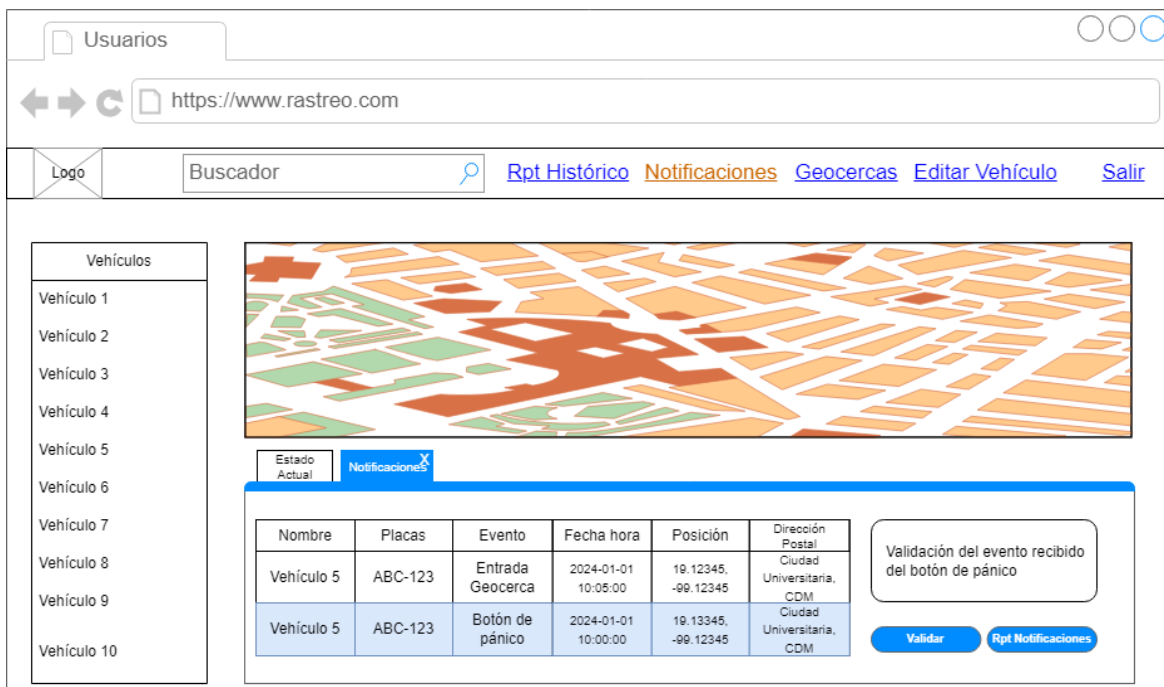


Figura. 19 Layout (Mockup) Notificaciones

- Las notificaciones se presentan desde las más recientes a las más antiguas, esto quiere decir que la página solicita las notificaciones a la API y la API regresa aquellas que le pertenecen al usuario.
- Además, el usuario puede seleccionar una notificación de la tabla y validar dicha notificación para que ya no aparezca en esta lista.
- Cuando un usuario revisa una notificación puede agregar un mensaje para indicar que la notificación ha sido validada y cuando da clic en el botón de Validar, se lanza una llamada a la API con la información de la notificación seleccionada y el mensaje de la validación, la API realiza validaciones y finalmente escribe el mensaje en la tabla de Notificaciones y marca la notificación como validada.
- Desde este panel se cuenta con un botón que permite solicitar un reporte histórico de las notificaciones que han sido validadas, solo debemos ingresar un rango de fechas y la página nos mostrará la información en una tabla que podemos exportar a Excel.

Estado Actual | Notificaciones | Rpt. Notificaciones

Fecha inicio: 2024-04-01 10:00:00 Fecha fin: 2024-04-01 20:00:00 **Aceptar**

Usuarios | [Rpt Histórico](#) | [Notificaciones](#) | [Geocercas](#) | [Editar Vehículo](#) | [Salir](#)

Logo | Buscador

Vehículos

- Vehículo 1
- Vehículo 2
- Vehículo 3
- Vehículo 4
- Vehículo 5
- Vehículo 6
- Vehículo 7
- Vehículo 8
- Vehículo 9
- Vehículo 10

Nombre	Placas	Evento	Fecha hora	Posición	Dirección Postal	Mensaje
Vehículo 5	ABC-123	Entrada Geocerca	2024-01-01 10:00:00	19.12345, -99.12345	Ciudad Universitaria, CDM	Validado
Vehículo 5	ABC-123	Botón de pánico	2024-01-01 10:05:00	19.13345, -99.12345	Ciudad Universitaria, CDM	Se atendió emergencia

Figura. 20 Layout (Mockup) Reporte Notificaciones

6 PUESTA A PRODUCCIÓN

6.1 REVISIÓN Y PRUEBAS DE CAJA NEGRA

Antes de llegar a la puesta en producción de la plataforma se deben cumplir una serie de pruebas que validan el correcto funcionamiento del proyecto, para dejar evidencia de dicho proceso de verificación y pruebas.

Cabe mencionar que las pruebas realizadas fueron de caja negra (Tejero, 2016); es decir, el revisor solo se preocupa en verificar que el funcionamiento de cada caso de uso sea tal como lo solicitó el usuario final y no conoce cómo está desarrollado el sistema ni la calidad del código que lo conforma.

Por temas de presupuesto, el encargado de realizar las pruebas de caja negra era el director de investigación y desarrollo, y la siguiente matriz de prueba es un ejemplo de la que originalmente se utilizó, sólo se han omitido los campos referentes a la información del cliente.

Comprendo que existen formatos referentes a pruebas de caja negra mucho más completos que los expuestos en este escrito y que las pruebas son realizadas por expertos en el área, sin embargo, el siguiente ejemplo fue el utilizado y nos fue suficiente para corroborar el funcionamiento de la plataforma de rastreo.

Aquí un ejemplo de la matriz de prueba que valida el funcionamiento del login del portal de usuario, cabe mencionar que:

- Para realizar las pruebas se creó un usuario que cuenta con acceso al administrador y se le compartieron los accesos al responsable de las pruebas.
- Con este usuario, el responsable de las pruebas creó diferentes usuarios hijos que utilizó para validar los diferentes casos de uso a su discreción.

No se exponen todos los casos de uso que componen el portal, ya que, sería algo muy extenso, pero en esencia, el proceso que se llevó a cabo es prácticamente el mismo para cada caso de uso.

Núm. de caso	Caso de uso	Pruebas realizadas	Tiempo de prueba	Descripción del error	Estado de la prueba
1	Como usuario del portal web quiero que al ingresar solo muestre aquellas secciones y pantallas a las que tenga acceso según mi rol de usuario para poder tener acceso únicamente a las opciones, información y pantallas adecuadas a mi perfil de usuario.	Ingresar correo y usuario correctos.	10 min	Sin error	Correcto
		Ingresar correo y/o contraseña incorrecta	10 min	Sin error	Correcto
		Recuperación de contraseña	20 min	Sin error	Correcto
		Validación del rol, solo debe mostrar las pantallas ligadas al perfil	20 min	Sin error	Correcto

Figura. 21 Tabla ejemplo matriz de prueba

En la matriz anterior se establece:

- **Núm. de caso**, es el identificador de la prueba.
- **Caso de uso**, es la historia de usuario o tarea a la que se le hará la prueba.
- **Pruebas realizadas**, se establecen las acciones a revisar que garanticen el correcto funcionamiento del módulo.
- **Tiempo de prueba**, se ingresa el tiempo que tardó en ejecutarse la prueba.
- **Descripción del error**, si hubiera error durante la ejecución de la prueba, se escribe en esta columna con los detalles e incluso se puede agregar una imagen o referencia del error.
- **Estado de la prueba**, se establece como correcto o incorrecto según el resultado de la prueba.

Si la matriz anterior es satisfactoria al 100% se declara que el funcionamiento del sistema es correcto y se puede proceder con su liberación y puesta a producción, de haber algún error, se me reportaba y procedía a corregir el error que indica el responsable de las pruebas y se repetían las pruebas para validar el funcionamiento. **Las pruebas de caja negra se llevaron a cabo en aproximadamente 20 días.**

6.2 PUESTA A PRODUCCIÓN

La puesta a producción se lleva a cabo una vez que las pruebas hayan pasado en su totalidad, para después, acondicionar e instalar todo lo necesario para que la plataforma de rastreo funcione sin problema.

El cliente cuenta con un proveedor de infraestructura que se encarga de crear y administrar los servidores del cliente, dicho proveedor acondicionó una serie de servidores con sistema operativo Windows; además, fue el encargado de instalar SQL Server, controladores, Firewall, etc., y es el responsable del mantenimiento y administración de todos los servidores y base de datos.

Los servidores quedaron distribuidos de la siguiente manera:

- Un servidor de ocho núcleos dedicado para la base de datos central.
- Dos servidores de ocho núcleos que sirven para la geocodificación inversa.
- Un servidor web de cuatro núcleos que permite el despliegue de la API, portal de usuarios y portal administrador.
- Dos servidores de ocho núcleos a los que se les instala el decodificador de paquetes.

Por lo tanto, una vez que el proveedor de infraestructura nos entrega los servidores nos brinda un usuario de Windows y uno de base de datos para poder instalar los diferentes componentes de la plataforma de rastreo, lo cual incluye lo siguiente:

- Se crearon las diferentes bases de datos y se crearon diferentes usuarios de base de datos para cada componente de software.
- Utilizando Internet Information Services (IIS), el cual es, un servidor web, se publicaron:
 - La API.
 - El portal web administrador.
 - Y el portal web de usuarios.
- Se instalaron diferentes instancias del servicio de Windows del decodificador de paquetes.

Una vez instalado todos los componentes de la plataforma de rastreo, lo cual, me llevo aproximadamente 15 días, se informa al cliente para que su proveedor de infraestructura tome el control total de los servidores y de la base de datos.

Por mi parte, al finalizar el proyecto quedé como responsable del mantenimiento de la plataforma de rastreo, pero solo en lo que respecta a los componentes de software creados por mí; es decir, revisión de archivos de registro (LOGs) y corrección de errores o cambios solicitados por los usuarios.

De igual forma quedé como usuario de soporte técnico para resolver las dudas y ayudar a los usuarios, con respecto al portal de usuarios y administrador.

Con esta distribución de servidores se logra tener una plataforma de rastreo que atiende aproximadamente 10, 000 equipos de rastreo conectados al mismo tiempo.

Además, es capaz de soportar todo el estrés generado por los mismos usuarios en el día a día, esto incluye la búsqueda de unidades y la generación de reportes de varios días.

7 CONCLUSIONES

La plataforma de rastreo surgió dada la necesidad de un cliente de la empresa que me contrató, dicho cliente requiere una herramienta capaz de administrar su flota de vehículos y que esté a la altura con los estándares de calidad que ofrece a sus clientes.

La plataforma fue un rotundo éxito, su funcionamiento no sólo ayudó al cliente a tener contentos a sus usuarios que se quejaban demasiado de la plataforma anterior, sino que evitó multas de varios miles de pesos que anteriormente había tenido que pagar debido al mal desempeño y funcionamiento de la plataforma anterior.

El éxito de la plataforma se logró no sólo gracias a la experiencia que adquirí trabajando en otra empresa pionera en el sector de rastreo satelital, y en la cual, empecé a laborar como desarrollador de software y terminé siendo gerente del área de desarrollo, sino al equipo que me apoyó y a los conocimientos adquiridos durante mi estancia en la Universidad Nacional Autónoma de México, de la cual estoy muy orgulloso.

Saber electrónica me permitió tener una visión global del proyecto, ya que, pude comprender cómo funciona el hardware de los equipos GPS y cómo es que estos transmiten la información, de lo cual agradezco a la empresa por permitirme estar presente en algunos procesos del área de investigación y desarrollo, a pesar de que no era mi principal función para este proyecto. Gracias a todo esto pude dar mi opinión en la configuración y homologación de los equipos GPS, lo cual, me facilitó el desarrollo y creación de cada componente de software.

Hoy en día esta plataforma de rastreo sigue siendo una de mis principales responsabilidades y proyectos que tengo a cargo. La plataforma ha mejorado y se han creado más funcionalidades y nuevos modelos de equipo se han ido agregando; incluso, ha crecido tanto que se tienen cerca de 18 mil equipos conectados hoy en día, y se espera que en un futuro el número de equipos conectados se incremente aún más, lo que representará un rediseño de infraestructura y usabilidad en los portales web.

Es por ello, por lo que se está negociando una actualización de la plataforma, para mejorar la arquitectura a través de aplicar diferentes patrones de diseño (diseño, s.f.), los cuales, permiten estructurar de una mejor manera el código, facilitan el mantenimiento, ya que, para este proyecto en específico, en el decodificador de paquetes nos ayudaría a integrar nuevos equipos de distintos fabricantes.

También se plantea introducir nuevas tecnologías como React.js (Meta Platforms, s.f.), el cual, es una biblioteca de JavaScript que permite crear aplicaciones web basadas en el uso de componentes, con ello, se pueden crear páginas web dinámicas que requieren actualizar su contenido constantemente.

En cuanto al backend se plantea integrar NET 8.0 (Microsoft, .NET | Crear. Probar. Implementar., s.f.), el cual, incluye mejoras a la plataforma .NET como el soporte a

múltiples plataformas o sistemas operativos, además, de un mejor rendimiento, y para este proyecto, nos ayudaría a actualizar la API y el decodificador de paquetes con las tecnologías más recientes, garantizando su compatibilidad y ejecución en servidores y sistemas operativos recientes, sobre todo aquellos basados en Windows.

Hoy en día sólo es una propuesta, pero, así como en un su momento este proyecto sólo fue una propuesta con un demo simple, espero que en el futuro esta actualización sea una realidad y pueda seguir disfrutando de hacer este trabajo como lo he hecho durante todo este tiempo.

8 REFERENCIAS

- Bootstrap. (s.f.). *Build fast, responsive sites with Bootstrap*. Obtenido de Getbootstrap: <https://getbootstrap.com/>
- diseño, P. d. (s.f.). *Patrones de diseño*. Obtenido de Refactoring Guru: <https://refactoring.guru/es/design-patterns>
- Durango, A. (2015). *Diseño Web con CSS: 2ª edición*. IT Campus Academy.
- EUA. (s.f.). *GPS*. Obtenido de GPS.gov: <https://www.gps.gov/systems/gps/spanish.php>
- Gallego, M. T. (2012). *Metodología Scrum*. Barcelona, España: Universitat Oberta de Catalunya.
- García, J. C. (14 de 11 de 2023). *¿Qué es y cómo funciona el GPS?* Recuperado el 11 de 2024, de GPS Total EWebik: <https://gpstotal.org/es/gps>
- García, J. C. (03 de 11 de 2024). *Sistemas Global de Navegación por Satélite GNSS*. Recuperado el 11 de 2024, de GPS Total EWebik: <https://gpstotal.org/es/gps/gnss>
- Google. (21 de 05 de 2024). *Cómo agregar un mapa de Google Maps con marcadores usando HTML*. Obtenido de Google Maps Platform: <https://developers.google.com/maps/documentation/javascript/add-google-map-wc-tut?hl=es-419>
- Google. (26 de 06 de 2024). *Maps JavaScript API*. Obtenido de Google Maps Platform: <https://developers.google.com/maps/documentation/javascript?hl=es>
- Google. (13 de 08 de 2024). *Usa claves de API*. Obtenido de Google Maps Platform : <https://developers.google.com/maps/documentation/javascript/get-api-key?hl=es>
- INEGI. (s.f.). *INEGI*. Obtenido de INEGI: <https://www.inegi.org.mx/sistemas/olap/Proyectos/bd/continuas/transporte/vehiculos.asp?s=est>
- jQuery. (s.f.). *jQuery*. Obtenido de jQuery: <https://jquery.com/>
- Maza, M. Á. (2012). *JavaScript*. Innovación Y Cualificación.
- Meta Platforms, I. (s.f.). *React The library for web and native user interfaces*. Obtenido de React.js: <https://react.dev/>
- Microsoft. (s.f.). *.NET | Crear. Probar. Implementar*. Obtenido de .NET: <https://dotnet.microsoft.com/es-es/>
- Microsoft. (21 de 11 de 2022). *Microsoft.SqlServer.Types*. Obtenido de Nuget: <https://www.nuget.org/packages/microsoft.sqlserver.types>
- Microsoft. (26 de 04 de 2023). *Información general de la biblioteca de clases de .NET*. Obtenido de Microsoft Learn Challenge: https://learn.microsoft.com/es-es/dotnet/standard/class-library-overview?WT.mc_id=dotnet-35129-website

- Microsoft. (09 de 05 de 2023). *Introducción a las aplicaciones de servicios de Windows*. Obtenido de Microsoft Learn Challenge: <https://learn.microsoft.com/es-es/dotnet/framework/windows-services/introduction-to-windows-service-applications>
- Microsoft. (19 de 03 de 2024). *Crear API RESTful con ASP.NET Web API*. Obtenido de Microsoft Learn Challenge: <https://learn.microsoft.com/es-es/aspnet/web-api/overview/older-versions/build-restful-apis-with-aspnet-web-api>
- Microsoft. (10 de 07 de 2024). *Microsoft Learn Challenge*. Obtenido de Introducción al servidor web IIS: <https://learn.microsoft.com/es-es/iis/get-started/introduction-to-iis/iis-web-server-overview>
- Microsoft. (s.f.). *Prueba SQL Server en el entorno local o en la nube*. Recuperado el 11 de 2024, de Microsoft: <https://www.microsoft.com/es-mx/sql-server/sql-server-downloads>
- SASS. (s.f.). *Sass Basics*. Obtenido de SASS: <https://sass-lang.com/guide/>
- Tejero, A. G. (01 de 07 de 2016). *Pruebas de caja negra. Técnica de partición equivalente*. Obtenido de Universitat Politècnica de València: <https://riunet.upv.es/handle/10251/66886>
- Windows, M. (s.f.). *Este es Windows*. Obtenido de Microsoft Windows: <https://www.microsoft.com/es-mx/windows?r=1>