

	UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA			
	Documento	Código	Fecha	Revisión
	FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO	F-AC-DBL-007	08-05-2023	B
Dependencia	Aprobado		Pág.	
DIVISIÓN DE BIBLIOTECA	SUBDIRECTOR ACADEMICO		1(75)	

AUTORES	Sebastian Ortiz Pérez		
FACULTAD	Facultad de Ingeniería		
PLAN DE ESTUDIOS	Plan de Estudios de Ingeniería de Sistemas		
DIRECTOR	Luis Anderson Coronel Rojas		
TÍTULO DE LA TESIS	Desarrollo de la plataforma web (versión 2.0) mediante el Framework Angular para la empresa Compara Motors, ubicada en la ciudad de Ocaña, Norte de Santander		
TITULO EN INGLES	Development of the web platform (version 2.0) using the Angular Framework for the company Compara Motors, located in the city of Ocaña, Norte of Santander.		
RESUMEN			
Un tema actual es la tecnología y sus grandes avances, además de esto su gran contribución a que las empresas lleguen a muchos más clientes a nivel nacional o mundial y esto se debe gracias al desarrollo, la cual ha sido una herramienta fundamental para este crecimiento, y no es una idea descabellada que las empresas modernas tengan o piensan tener una plataforma web para brindar mejores servicios a sus clientes, quienes también serán sus usuarios.			
RESUMEN EN INGLES			
A current issue is technology and its great advances, in addition to this its great contribution to companies to reach many more customers nationally or globally and this is due to the development, which has been a fundamental tool for this growth, and it is not a crazy idea that modern companies have or plan to have a web platform to provide better services to its customers, who will also be its users.			
PALABRAS CLAVES	Actualización, Angular, Desarrollo, Framework, Software		
PALABRAS CLAVES EN INGLES	Update, Angular, Development, Framework, Software		
CARACTERÍSTICAS			
PÁGINAS: 74	PLANOS: 6	ILUSTRACIONES: 47	CD-ROM:



**Desarrollo de la plataforma web (Versión 2.0) mediante el Framework Angular para
la empresa Compara Motors, ubicada en la ciudad de Ocaña, Norte de Santander**

Sebastian Ortiz Pérez

Facultad de Ingenierías, Universidad Francisco de Paula Santander Ocaña

Ingeniería de Sistemas

Mgc. Luis Anderson Coronel Rojas

Febrero de 2023

Dedicatoria

Primeramente, deseo dedicarle este trabajo a Dios por ser tan bueno y misericordioso conmigo y permitirme estar un poco más cerca de cerrar un ciclo en mi vida y de alcanzar una meta propuesta hace unos años.

A mis padres que siempre estuvieron pendientes de mis estudios y que me acompañaron en este proceso en todos los aspectos, anímicamente, económicamente y creyendo fielmente en mis capacidades y motivándome cada día por salir adelante.

A mi director de plan de trabajo y todos los profesores que aportaron para que en este tiempo yo me convierta en un profesional y colega de muchos de ellos.

Agradecimientos

Quiero agradecer a Dios por ser mi guía y estar en todo momento recordándome: “que todo lo puedo en el que me fortalece” en este proceso y hacer realidad un sueño que desde muy niño anhelaba.

A mi mamá Liliana Pérez Pineda por apoyarme y siempre estar pendiente de mi y mis estudios, dándome el aliento necesario para seguir adelante.

A mi papá que en tiempos difíciles siempre tiene un consejo sabio para darme las fuerzas necesarias y no bajar mis brazos.

A mi hermano y mis primas que siempre me dijeron que siguiera con mis estudios y no me quedara solo con una educación media.

A mi tía que no me vio ingresar a una universidad, pero su sueño siempre fue que estudiara y murió con esa expectativa de que algún día yo sería un profesional más de este país.

Por último, a mis compañeros de estudio con quienes posiblemente compartiré esta fecha especial y con los cuales compartí agradables y difíciles momentos durante este periodo, gracias por enseñarme el valor del trabajo en equipo.

Índice

Capítulo 1. Desarrollo de la plataforma web (Versión 2.0) mediante el Framework Angular para la empresa Compara Motors, ubicada en la ciudad de Ocaña, Norte de Santander	12
1.1. Descripción de la empresa	12
1.1.1. Misión	12
1.1.2. Visión	13
1.1.3. Objetivos de la empresa	13
1.1.4. Descripción de la estructura Organizacional	13
1.1.5. Descripción de la dependencia y/o proyecto asignado	14
1.2. Diagnóstico Inicial de la dependencia asignada	15
1.2.1. Planteamiento del Problema	16
1.3. Objetivos de la Pasantía	17
1.3.1. Objetivo General	17
1.3.2. Objetivos Específicos	17
1.4. Descripción de las actividades a Desarrollar	18
1.5. Cronograma de Actividades	19
Capítulo 2. Marco de Referencia	20
2.1. Marco Conceptual	20
2.2. Marco Legal	22
Capítulo 3. Informe cumplimiento de Trabajo	24

3.1. Realizar estudios sobre el Framework Angular	5 24
3.2. Familiarización con la plataforma web existente	25
3.3. Manejo del repositorio GitLab	26
3.4. Desarrollo de Código	28
3.5. Desarrollo FrontEnd	29
3.5.1. Cmwebadmin Modulo Team	31
3.5.2. Plataforma Original (Team)	32
3.5.3. Cambios realizados en la página principal de team	33
3.6. Desarrollo BackEnd	44
3.6.1. Instalación de dependencias necesarias	46
3.6.2. Inicialización del admin	48
3.6.3. Uso de la plataforma Postman	50
3.6.4. Creación de la componente Comparamotors.controller.js	51
3.6.5. Creación de la componente Comparamotors.service.js	53
3.6.6. Endpoints creados	54
3.6.7. Primer Endpoints	55
3.6.8. Segundo Endpoints	58
3.7. Creación de tokens	60
Capítulo 4. Entrega de modulo finalizado	66
4.1. Pruebas Realizadas	66

Capítulo 5. Diagnostico Final	6 68
Capítulo 6. Conclusiones	70
Capítulo 7. Recomendaciones	71
Capítulo 8. Referencias	73

Lista de Tablas

Tabla 1 Estructura Organizacional	14
Tabla 2 Análisis FODA de la dependencia asignada	15
Tabla 3 Descripción de Actividades	18
Tabla 4 Cronograma de Actividades	19
Tabla 5 Antes y Después Modulo Team	39
Tabla 6 Antes y Después Galería	41

Lista de Figuras

Figura 1	Estudios Realizados del Framework	24
Figura 2	Reunión Equipo de Trabajo	25
Figura 3	Repositorio Proyecto Cmwebadmin	26
Figura 4	Repositorio Proyectos Empresa	27
Figura 5	Id de ramas en Visual Studio Code	27
Figura 6	Comando para subir cambios al Repositorio	28
Figura 7	Proyectos Trabajados	29
Figura 8	Código modulo asignado	30
Figura 9	Importación de utilidades de PrimeNg	31
Figura 10	Plataforma Modulo Asignado (Team)	32
Figura 11	Spinner Incorporado	33
Figura 12	Clases para crear el Spinner	34
Figura 13	Declaración de las funciones del Spinner, NotFound y NotResult	35
Figura 14	Muestra del NotResult	36
Figura 15	código insertando ToastService	37
Figura 16	ToastService en el Modulo	37
Figura 17	Spinner Insertado en el formulario	38
Figura 18	Código Validación de Campos	42
Figura 19	Validación, Pagina	42
Figura 20	Código Validación Galería	43
Figura 21	Validación Galería, Pagina	44
Figura 22	Documentación Estudiada Firestore	45
Figura 23	creación de Carpeta Cmapiuser	45

Figura 24	Verificación de Usuario en la base de Datos	46
Figura 25	Instalación Componente Express	47
Figura 26	Instalación Componente Cors	48
Figura 27	Código Para inicializar el Admin	49
Figura 28	Lanzamiento del Servicio	50
Figura 29	Configuración Postman	51
Figura 30	Petición Get	52
Figura 31	Peticiones Post usando Try Catch	53
Figura 32	Eliminación de Datos a la consulta	54
Figura 33	Código Endpoint (Comparamotors.service.js)	56
Figura 34	Código Endpoint (Comparamotors.controller.js)	56
Figura 35	Código Endpoint (Comparamotors.model.js)	57
Figura 36	Consulta Realizada por Postman	57
Figura 37	Código Endpoint 2 (Comparamotors.service.js)	58
Figura 38	Código Endpoint 2 (Comparamotors.controller.js)	59
Figura 39	Código Endpoint 2 (Comparamotors.model.js)	59
Figura 40	Consulta 2 Realizada por Postman	60
Figura 41	Token Creado visto en Firebase	61
Figura 42	Validación de Campos para iniciar Sesión	61
Figura 43	Código validación Campos inicio de Sesión	62
Figura 44	Generación del UserToken	63
Figura 45	Código TeamController.js	64
Figura 46	Código AuthService.js	65
Figura 47	Entregas ensambladas con la rama develop	67

Resumen

Un tema principal en nuestra actualidad es la tecnología y sus grandes avances, además de esto su gran contribución a que las empresas lleguen a muchos más clientes a nivel nacional o mundial y esto se debe gracias al desarrollo, la cual ha sido una herramienta fundamental para este crecimiento, y no es una idea descabellada que las empresas modernas tengan o piensan tener una plataforma web para brindar mejores servicios a sus clientes, quienes también serán sus usuarios. Por lo tanto, en la empresa Comparamotors hace años están con esta gran idea de expandirse por medio de una plataforma y que con la ayuda de esta poder ayudar y satisfacer las necesidades planteadas en sus objetivos, por lo demás con el fin de hacer cumplir este proyecto de grado bajo la modalidad de pasantías desarrollado en la empresa mencionada anteriormente, tiene como objetivo general Desarrollar la plataforma web (Versión 2.0) mediante el Framework de Angular, para la empresa COMPARA MOTORS, ubicada en la ciudad de Ocaña, Norte de Santander. Mediante la aplicación de conocimientos adquiridos en la rama de la programación, donde se logró realizar con satisfacción una nueva actualización propuesta de la plataforma web que ya existía.

Introducción

Las plataformas web y el desarrollo de estas son una meta propuesta de la empresas al iniciar, las cuales pueden hacerlas realidad por medio de una variedad de lenguajes de programación, ya es decisión del ingeniero encargado o de la empresa le parezca mejor propuesta, en este caso bajo código de los lenguajes como HTML, CSS, JAVA, JAVA SCRIPT, TYPESCRIPT, y con la inclusión de framework para hacer de este un desarrollo más eficiente y manejando en su mayoría el framework de Angular, evaluando las entregas anteriores de esta plataforma, en la que se tomó la decisión de no dejar la plataforma tal cual y como esta, en cambio lanzar actualizaciones periódicamente para hacer que esta no se vuelva obsoleta.

Por lo anterior este trabajo de grado se enfoca en una actualización para la plataforma web que ya existe en la empresa, teniendo en cuenta que la última entrega se realizó a mitad del presente año (2022), en la cual la organización se siente satisfecha pero desea que esta no quede solo en eso y trabajan par que como la tecnología este en constante actualización , la cual se realiza en el lapso de 16 semanas, primeramente estudiando la plataforma y su desarrollo por un tiempo, y de esta manera tomar decisiones en conjunto con las autoridades de esta entidad, para que en conjunto con el ingeniero o jefe inmediato procedamos al desarrollo de este, finalmente presentando los resultados del cumplimiento de cada una de las tareas y compromisos en las fechas estipuladas para cada una de estas entregas.

Capítulo 1. Desarrollo de la plataforma web (Versión 2.0) mediante el Framework Angular para la empresa Compara Motors, ubicada en la ciudad de Ocaña, Norte de Santander.

1.1. Descripción de la empresa

Compara Motors se basa principalmente en ofrecer una herramienta que presente resultados sobre los productos seleccionados y se encuentra liderado por un gran equipo de trabajo, encargado de implementar nuevas técnicas e intercambios de ideas con el fin de generar confiabilidad y la satisfacción de nuestros clientes (*Compara Motors, 2022*).

1.1.1. Misión

Compara Motors es una empresa legalmente constituida con el propósito de ofrecer a los usuarios información relacionada con autos nuevos, comparación de vehículos de todos los segmentos, y ofrecer a nuestros clientes pautas publicitarias mediante página web, aplicación iOS, Android y canal de YouTube.

1.1.2. Visión

Ser un sitio web reconocido a nivel nacional, que ofrezca a los usuarios acceso a la información verídica, referente al conocimiento técnico, tecnológico y novedoso de los vehículos.

1.1.3. Objetivos de la empresa

Estamos dispuestos a generar contenido informativo del mundo automotriz con el fin de mostrar las mejores marcas y modelos en el mercado colombiano. Es por esto, que presentamos 4 las más actualizadas noticias o novedades de este sector, proporcionando acceso a la información a todos nuestros usuarios y generar el contenido de nuestros clientes. Principalmente nos enfocamos en que se encuentren satisfechos y cómodos con el servicio que brindamos para cubrir las necesidades de nuestros clientes garantizando su seguridad y confianza (*Compara Motors, 2022*).

1.1.4. Descripción de la estructura Organizacional

En la actualidad la empresa no posee una estructura organizacional, sin embargo, este es el personal que está laborando actualmente con su respectivo cargo.

Tabla 1*Estructura Organizacional*

Nombre	Función
Javier Antonio Gamboa Rivera	Representante legal
Walter Arley Duran Quintero	Diseñador web
Karen Mejía Amaris	Diseñador web
Diego Alejandro Gamboa Rivera	Marketing y logística
Haminton Arévalo Álvarez	Programador web
Marta Patricia Sanjuán Sarabia	Área administrativa, Recursos humanos
María Paula Granados	Relaciones empresariales
Leidys Dalina Barrientos Téllez	comunicadora social

Nota. Autor (2022).

1.1.5. Descripción de la dependencia y/o proyecto asignado

Compara Motors no posee dependencias ya que no cuenta con un organigrama claro, aun así y tras un primer contacto con la empresa están requiriendo pasantes de ingeniería de sistemas ya que el desarrollo y actualización de su plataforma digital (página web) lo demandan. Es por ello que el plan de trabajo estará enfocado en el área de desarrollador web con énfasis en un desarrollador full stack (Front end y Back end).

1.2. Diagnóstico Inicial de la dependencia asignada

Tabla 2

Análisis FODA de la dependencia asignada

	Fortalezas (F)	Debilidades (D)
Matriz DOFA	F1. Posee una plataforma digital ya establecida. F2. Tiene las herramientas tecnológicas para trabajar. F3. Posee personal capacitados en los temas.	D1. Poco tiempo de la empresa en el mercado. D2. Poca disposición de modelos de autos. D3. El catálogo de autos no va dirigido a toda la población.
Oportunidades (O)	FO	DO
O1. Establecer relaciones con grandes empresas del sector automotriz. O2. Venta de espacios publicitarios. O3. Fácil acceso a contenido informativo. O4. Fácil comunicación con la empresa.	Promocionar por medio de la plataforma digital las marcas con las cuales se pueden hacer alianzas, dando así una información exacta a los clientes	Servirse de la viralización del sector y así aprovechar el fácil acceso del contenido informativo.
Amenazas (A)	FA	DA
A1. Alta competencia en el mercado. A2. Los modelos por marca son limitados A3. Existen pocas alianzas con empresas del sector. A4. Es limitado los modelos por año.	Utilizar las herramientas para implementar mejoras en los medios tecnológicos y lograr concretar alianzas potenciales en el sector donde se encuentra ejerciendo la empresa.	Implementar mejoras en los servicios digitales, con el fin de brindar una mejor experiencia a los usuarios

Nota. Autor (2022)

1.2.1. Planteamiento del Problema

Estamos en un tiempo donde lo podemos llamar la era de la tecnología, dado que vemos cantidad de plataformas digitales, también aplicativos móviles para cualquier cosa, es decir para lo que usted busque posiblemente encuentre un App, y sabemos que estos ayudan a un crecimiento llegando a cualquier parte del país y del mundo.

Por otra parte, al contar con una página sabemos que están deben estar en constante actualización con el fin de conservan un ambiente seguro y confiable, y no permitir que la pagina se vuelva obsoleta, de tal manera que es de suma importancia mantenerla activa dado que de esta manera puede crecer constantemente y obtener los resultados esperados a la hora de realizarse.

En Comparamotors ya cuenta con una plataforma digital establecida mediante la cual ofrecen sus servicios, sin embargo, como se mencionó anteriormente con el fin de mantener activa la página y crecer se desea seguir mejorando mediante una actualización (Versión 2.0), dando solución a posibles errores, además de usar tecnologías que actualmente son muy demandadas en este ámbito, y de esta manera brindar mejores servicios a sus clientes.

1.3. Objetivos de la Pasantía

1.3.1. Objetivo General

Desarrollar la plataforma web (Versión 2.0) mediante el Framework de Angular, para la empresa COMPARAMOTORS, ubicada en la ciudad de Ocaña, Norte de Santander.

1.3.2. Objetivos Específicos

- Analizar el estado actual de la plataforma digital, determinando así las funciones que requieran actualización.
- Aplicar el diseño, con lenguaje de programación, en la plataforma web, con sus respectivas funcionalidades.
- Desarrollar las funciones de la plataforma digital, con los conocimientos previamente adquiridos del Framework Angular.
- Realizar las respectivas pruebas para verificar el funcionamiento correcto de la plataforma.

1.4. Descripción de las actividades a Desarrollar

Tabla 3

Descripción de Actividades

Objetivo General	Objetivos Específicos	Actividades a desarrollar
Desarrollar la plataforma digital (Versión 2.0) utilizando el Framework de Angular, la cual permita a los usuarios conocer información detallada y de calidad sobre vehículos automotores en la empresa COMPARAMOTORS, ubicada en la ciudad de Ocaña, Norte de Santander.	Analizar el estado actual de la plataforma digital, determinando así las funciones que requieran actualización.	<ul style="list-style-type: none"> • Maquetación y desarrollo web (Front End) • Desarrollo API user (Back End) • Desarrollo Web admin (Front End y Back end)
	Aplicar el diseño, con lenguaje de programación, en la plataforma web, con sus respectivas funcionalidades.	<ul style="list-style-type: none"> • Manejo del Framework Angular para el desarrollo de la plataforma • Desarrollar código de programación en HTML, CSS, JAVASCRIPT
	Desarrollar las funciones de la plataforma digital, con los conocimientos previamente adquiridos del Framework Angular Realizar las respectivas pruebas para verificar el funcionamiento correcto de la plataforma.	<ul style="list-style-type: none"> • Implementar conocimientos adquiridos (Angular y Node.js) • Realizar test de la plataforma por medio de software testing.

1.5. Cronograma de Actividades

Tabla 4

Cronograma de Actividades

Objetivos Específicos	Actividades	Mes 1				Mes 2				Mes 3				Mes 4				
		S1	S2	S3	S4													
Analizar el estado actual de la plataforma digital, determinando así las funciones que requieran actualización.	Maquetación y desarrollo web (Front End)	X	X															
	Desarrollo API user (Back End)			X	X													
	Desarrollo Web admin (Front End y Back end)			X	X													
Aplicar el diseño, con lenguaje de programación, en la plataforma web, con sus respectivas funcionalidades.	Manejo del Framework Angular para el desarrollo de la plataforma					X	X	X										
	Desarrollar código de programación en HTML, CSS, JAVASCRIPT								X	X	X	X	X					
Desarrollar las funciones de la plataforma digital, con los conocimientos previamente adquiridos del Framework Angular	Implementar conocimientos adquiridos (Angular y Node.js)													X	X	X	X	
Realizar las respectivas pruebas para verificar el funcionamiento correcto de la plataforma.	Realizar test de la plataforma por medio de testing software.																	X

Nota. Autor (2022)

Capítulo 2. Marco de Referencia

2.1. Marco Conceptual

Para llevar a cabo la creación de un software es necesario establecer un marco conceptual haciendo uso de algunos conceptos contenidos en libros y en la web, para el desarrollo adecuado de una aplicación o un software. Las páginas web se han vuelto indispensables gracias a los beneficios y utilidades que le aportan a una organización, como lo es el llegar a más personas y de una manera más rápida, Actualmente, muchos desarrolladores de software posicionan sus aplicaciones como una solución a un problema en particular. Como resultado, a medida que surgen nuevos requisitos, deben volver a trabajarse en diferentes etapas de desarrollo, lo que resulta en cambios de código entre aplicaciones. Duplicación y altos costos asociados con las horas-hombre requeridas para crear nuevas soluciones sin utilizar el trabajo de diseño anterior (Ortega, Guevara , & Benavides, 2017).

En el desarrollo de software, un framework es una estructura de soporte definida sobre la cual se pueden organizar y desarrollar otros proyectos de software. En programación, es un conjunto de funciones o código común que realiza tareas comunes y habituales (crear objetos, conectarse a bases de datos, etc.) en todo tipo de aplicaciones. Esto proporciona una base sólida para desarrollar aplicaciones específicas y evita componentes de desarrollo más triviales y genéricos, lo que permite a los diseñadores y

programadores dedicar más tiempo a definir los requisitos del software (Martínez Villalobos , Camacho Sánchez, & Biancha Gutierrez , 2010).

En una computadora, los programas usados para construir otros programas de computadora se llaman lenguajes de programación. Recibe su nombre porque contiene un lenguaje formal diseñado para organizar procesos algorítmicos y lógicos que luego son ejecutados por una computadora o sistema informático, lo que permite controlar su comportamiento físico y lógico y la interacción con los usuarios. Dicho lenguaje consta de símbolos, reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, con cuya ayuda se construye el código fuente de una determinada aplicación o software (Zuñiga Vasquez, 2020).

El desarrollo FrontEnd trabaja la interfaz visible para los usuarios, y hace que este pueda interactuar con el sitio web o sistema, está orientado al lenguaje de programación web de ejecución en equipos clientes. El desarrollo BackEnd se encarga de la manipulación de los datos, cabe aclarar que este necesita de un desarrollo FrontEnd de por medio, el desarrollador BackEnd debe de conocer de bases de datos, framework y aspectos de seguridad. Él debe encargarse de que la información que llega desde el FrontEnd, sea almacenada a una base de datos, también, se encarga de crear API's para que sus datos puedan consumirse de manera cómoda y pueda mejorar la experiencia del usuario (Pérez Ibarra, Quispe, Mullicundo, & Lamas, 2021).

2.2. Marco Legal

Es importante saber que en el mundo de la seguridad informática es muy común encontrar grandes cantidades de criminales informáticos, estos suelen atacar con mayor frecuencia la integridad, disponibilidad y confidencialidad de los datos, principalmente en aplicaciones y sitios web, para tranquilidad de los desarrolladores existen cantidades de normas y leyes que ayudan a combatir todos estos aspectos negativos.

Ley 1581 de 2012

Mediante la Ley 1581 de 2012, se expidió el Régimen General de Protección de Datos Personales, el cual, de conformidad con su artículo 1, tiene por objeto desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma (Artículo 1, Ley 1581 de 2012).

Ley 1341 de 2009

Mediante esta ley se determina el marco general para la formulación de las políticas públicas que regirán el sector de las Tecnologías de la Información y las Comunicaciones, su ordenamiento general, el régimen de competencia, la protección al usuario, así como lo

concerniente a la cobertura, la calidad del servicio, la promoción de la inversión en el sector y el desarrollo de estas tecnologías (Artículo 1, Ley 1341 de 2009).

Decreto 1360 de 1989

Mediante el cual se da conformidad con lo previsto en la ley 23 de 1982 sobre Derechos de Autor, el soporte lógico (software) se considera como una creación propia del dominio literario, el cual comprende uno o varios de los siguientes elementos: el programa de computador, la descripción de programa y el material auxiliar (Artículo 1 y 2, Decreto 1360 de 1989).

Ley 527 de 1999

Por medio de la cual se define y reglamenta el acceso y uso de los mensajes de datos, del comercio electrónico y de las firmas digitales, y se establecen las entidades de certificación y se dictan otras disposiciones. Por medio de la cual se define y reglamenta el acceso y uso de los mensajes de datos, del comercio electrónico y de las firmas digitales, y se establecen las entidades de certificación y se dictan otras disposiciones (Artículo 2, Ley 527 de 1999).

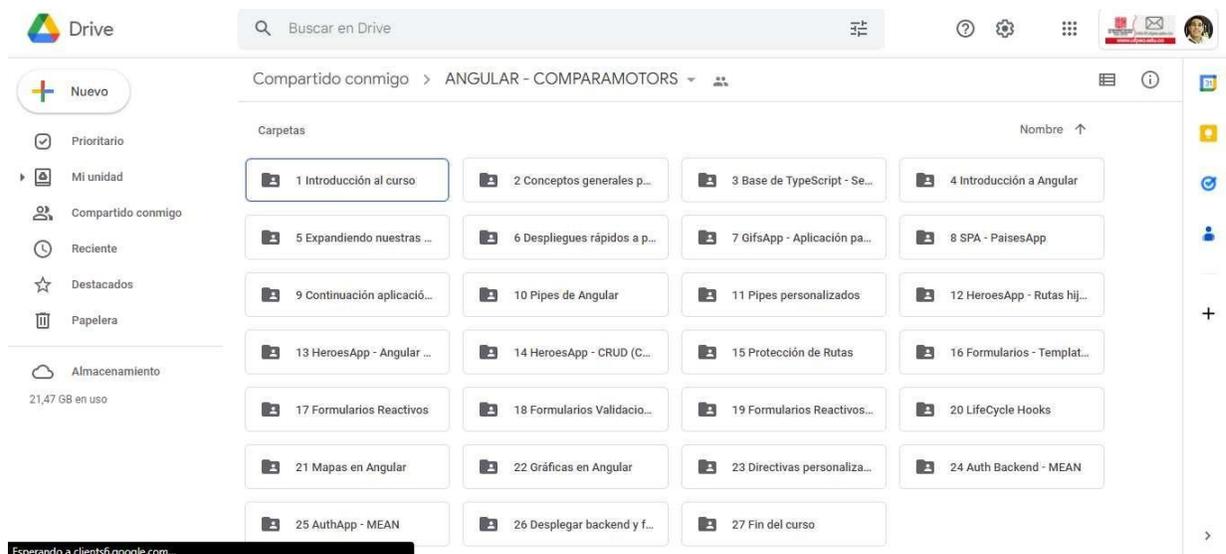
Capítulo 3. Informe cumplimiento de Trabajo

3.1. Realizar estudios sobre el Framework Angular

Al iniciar con el proceso en la empresa COMPARAMOTORS nos facilitaron un curso para profundizar en el tema de Angular, dado que su plataforma web está hecha bajo estos parámetros y de esta manera llegar con un conocimiento más amplio, y así poder ejecutar código de una manera más fluida y sin contratiempos, además de no tener muchas dudas o inquietudes durante el proceso de programación de la plataforma web, en el cual tuvimos alrededor de 2 semanas en las cuales pudiéramos profundizar en lo relacionado al framework Angular.

Figura 1

Estudios Realizados del Framework



Nota. Autor (2022).

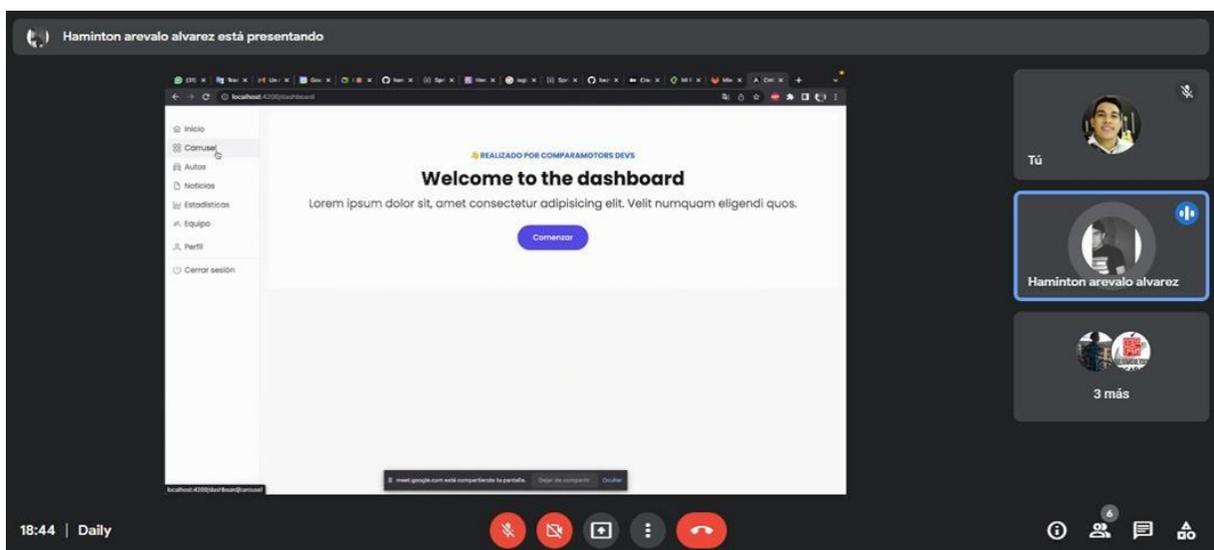
3.2. Familiarización con la plataforma web existente

Luego de estudiar lo relacionado con el Framework nos disponemos a realizar un sondeo de la página, ver sus funciones y su interfaz gráfica, de esta manera podemos lograr definir a lo que queremos llegar con esta actualización, revisando cada uno de sus módulos y procediendo a otorgar el módulo correspondiente, en mi caso el módulo team.

Observando y profundizando en las funciones de la página y de haber estudiado el código fuente, y ya siendo el responsable de un módulo, procedemos a hacerle pruebas para revisar que funciones no están cumpliendo con su objetivo y que funciones se le pueden agregar.

Figura 2

Reunión Equipo de Trabajo



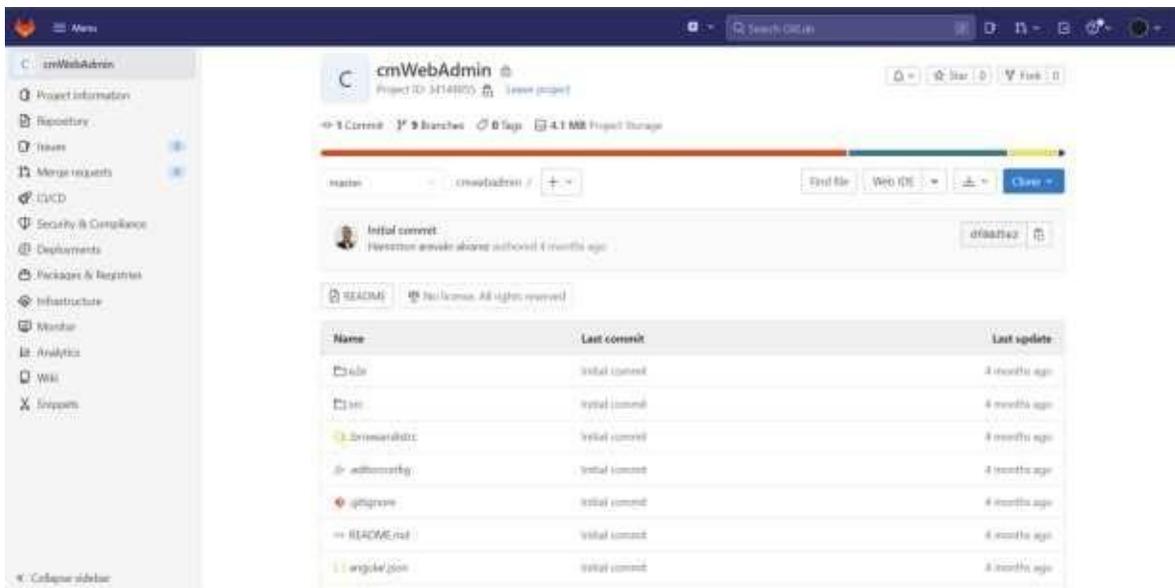
Nota. Autor (2022)

3.3. Manejo del repositorio GitLab

En esta actividad reforzamos un poco más sobre repositorios, y definimos como íbamos a trabajar para que todo fuera de manera ordenada, debido a que había varias personas editando el código, y como lo mencione anteriormente cada quien trabajaba en un módulo, y este repositorio nos ayudó de gran manera debido a que con ella pudimos crear una rama para cada desarrollador y podíamos programar sin interrumpir a otros y sin generar conflictos en el código.

Figura 3

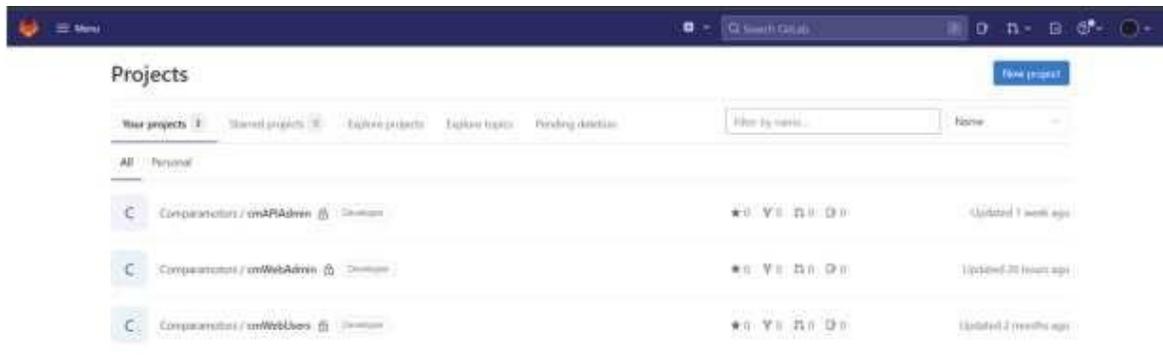
Repositorio Proyecto Cmwebadmin



Nota. Autor (2022)

Figura 4

Repositorio Proyectos Empresa

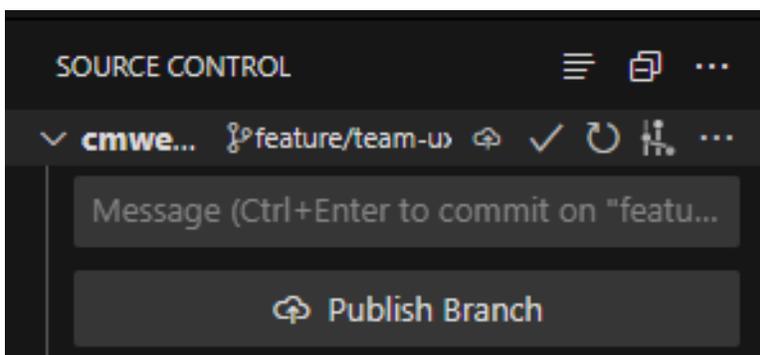


Nota. Autor (2022)

Aclarando que el repositorio lo manejamos más desde el id de Visual Studio Code, haciendo la mayoría de cosas con la consola de este, en el cual damos un nombre al trabajo que hemos realizado y lo subíamos al repositorio por medio del id.

Figura 5

Id de ramas en Visual Studio Code

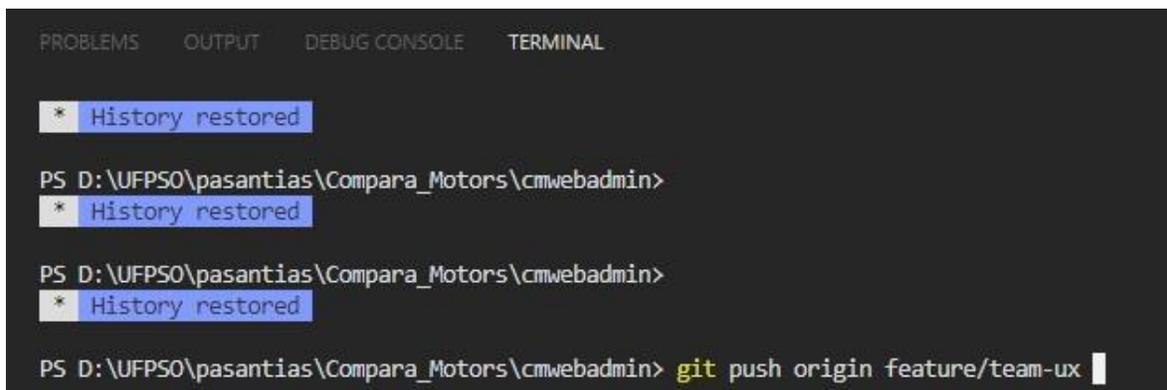


Nota. Autor (2022)

Luego de publicar una nueva actualización o modificación de código nos dirigimos a la consola de nuestro editor de texto y usábamos el comando “Git push origin feature/nombre de tu rama” y de esta manera lográbamos que estuviera en el repositorio para cuando lo fuéramos a lanzar a producción desde la rama develop.

Figura 6

Comando para subir cambios al Repositorio



```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

* History restored

PS D:\UFPSO\pasantias\Compara_Motors\cmwebadmin>
* History restored

PS D:\UFPSO\pasantias\Compara_Motors\cmwebadmin>
* History restored

PS D:\UFPSO\pasantias\Compara_Motors\cmwebadmin> git push origin feature/team-ux

```

Nota. Autor (2022)

3.4. Desarrollo de Código

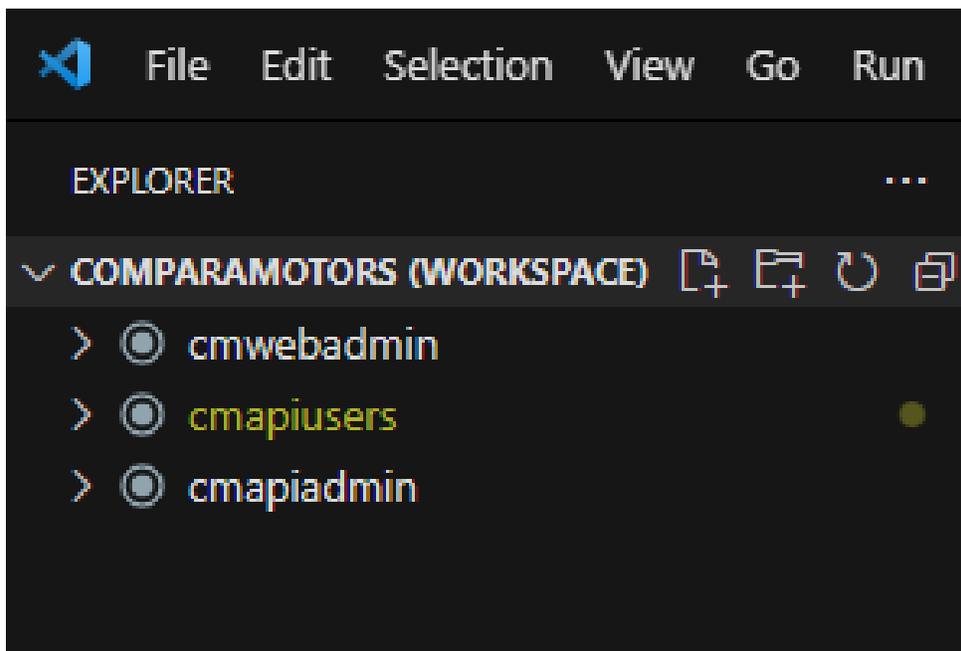
En esta actividad se evidencio que la página esta desarrollada en tres proyectos distintos los cuales estas nombrados y son utilizados de la siguiente manera:

- ✓ Cmwebusers: el cual cuenta con las vistas que serán mostradas al público.
- ✓ Cmwebadmin: la cual constara de los elementos proporcionados a los administradores de la empresa

- ✓ Cmapiadmin el cual constara de los códigos necesarios para la conexión a la base de datos y los métodos CRUD de cada elemento de Cmwebadmin
- ✓

Figura 7

Proyectos Trabajados



Nota. Autor (2022)

3.5. Desarrollo FrontEnd

En este punto iniciamos a desarrollar en el módulo asignado a cada desarrollador, en el cual desarrollamos lo que tiene que ver con la experiencia de usuario, los componentes de la página para los administradores y formularios para la creación de datos, actualización de datos, a la consulta de la base de datos, y eliminación de estos mismos.

Figura 8

Código modulo asignado

The screenshot shows a code editor with the following TypeScript code for a team component:

```

1 import { Component, OnInit } from '@angular/core';
2 import { ToastService } from '@core/services/toast.service';
3 import { AuthService } from '@modules/auth/services/auth.service';
4 import { Team } from '@modules/services/interfaces/team.interface';
5 import { ConfirmationService } from 'primeng/api';
6 import { TeamService } from '../../services/team.service';
7
8 @Component({
9   templateUrl: './team.component.html',
10  styleUrls: ['./team.component.scss'],
11 })
12 export class TeamComponent implements OnInit {
13   teams: Team[] = [];
14
15   public team: Team = {
16     id: '',
17     image: '',
18     name: '',
19     rol: '',
20     status: 'INACTIVO',
21   };
22
23   public isOpen: boolean = false;
24   private canEdit: boolean = false;
25
26   public notFound: Boolean = false;
27   public spinner: Boolean = true;
28   public result: Boolean = true;
29
30   constructor(
31
32

```

Nota. Autor (2022)

Cabe aclarar que se hizo uso de componentes de PRIMENG en el cual encontramos una serie de componentes de interfaz de usuario para manejar en Angular, del cual nos valimos para hacer el desarrollo más ameno y ágil, además de tener menos líneas de código, logrando así una plataforma más eficiente.

Figura 9

Importación de utilidades de PrimeNg

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'modules', 'shared', 'components', 'directives', 'pipes', 'primeng', 'assets', 'environments', and 'test.ts'. The code editor shows the following TypeScript code:

```

1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3
4 import { ButtonModule } from 'primeng/button';
5 import { InputTextModule } from 'primeng/inputtext';
6 import { ColorPickerModule } from 'primeng/colorpicker';
7 import { ToolbarModule } from 'primeng/toolbar';
8 import { TableModule } from 'primeng/table';
9 import { FileUploadModule } from 'primeng/fileupload';
10 import { InputSwitchModule } from 'primeng/inputswitch';
11 import { TagModule } from 'primeng/tag';
12 import { CardModule } from 'primeng/card';
13 import { TooltipModule } from 'primeng/tooltip';
14 import { DialogModule } from 'primeng/dialog';
15 import { InputNumberModule } from 'primeng/inputnumber';
16 import { DropdownModule } from 'primeng/dropdown';
17 import { RadioButtonModule } from 'primeng/radiobutton';
18 import { CalendarModule } from 'primeng/calendar';
19 import { ConfirmDialogModule } from 'primeng/confirmdialog';
20 import { ToastModule } from 'primeng/toast';
21 import { CheckboxModule } from 'primeng/checkbox';
22 import { FieldsetModule } from 'primeng/fieldset';
23 import { TabViewModule } from 'primeng/tabview';
24 import { SidebarModule } from 'primeng/sidebar';
25 import { InputTextareaModule } from 'primeng/inputtextarea';
26 import { StepsModule } from 'primeng/steps';
27 import { EditorModule } from 'primeng/editor';
28 import { ChipModule } from 'primeng/chip';
29 import { ProgressSpinnerModule } from 'primeng/progressspinner';
30 import { MenuModule } from 'primeng/menu';
31 import { DividerModule } from 'primeng/divider';
32

```

Nota. Autor (2022)

3.5.1. Cmwebadmin Modulo Team

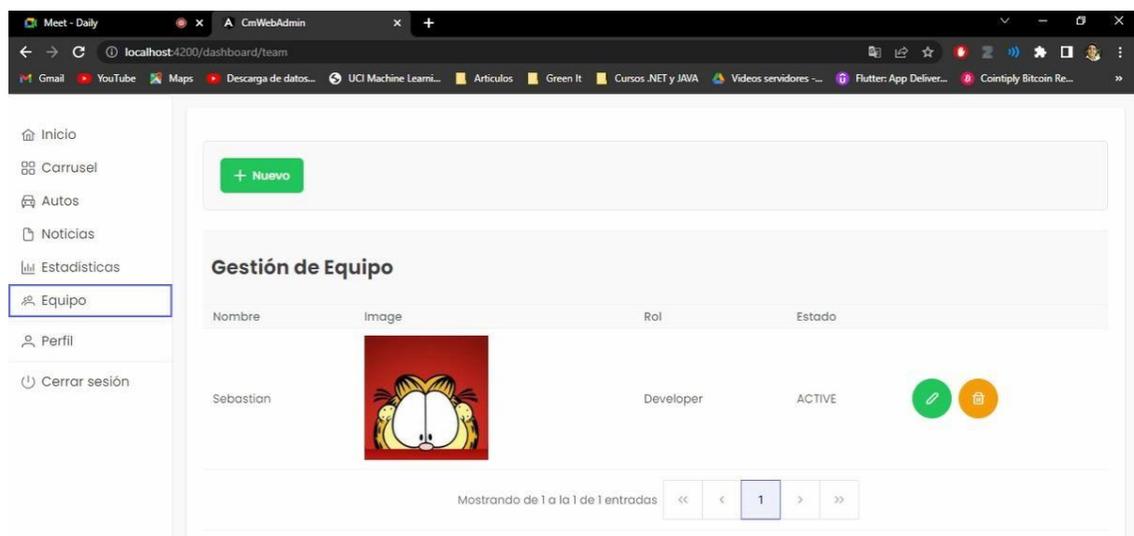
Luego de haber hecho pruebas de la anterior versión del módulo team, procedemos a ver que funciones necesitan cambios y/o ser eliminadas ya sea porque no están funcionando adecuadamente o simplemente no se está usando, en el cual observamos que a la plataforma le hacen falta algunas cosas para que la experiencia de usuario sea mejor las cuales estaremos viendo a continuación:

3.5.2. Plataforma Original (Team)

se puede observar cómo está actualmente el módulo de team desde la versión anterior, en la cual vemos que tenemos un botón de nuevo el cual sirve para crear un nuevo usuario que al accionarlo este nos abre una slider o un formulario en una ventana flotante, en la cual podemos ver los campos que debemos llenar para crear un usuario, también vemos que hay una tabla donde se nos almacena lo que hemos creado, la cual está conformada por cinco columnas (nombre, imagen, rol, estado y botones de editar y eliminar).

Figura 10

Plataforma Modulo Asignado (Team)



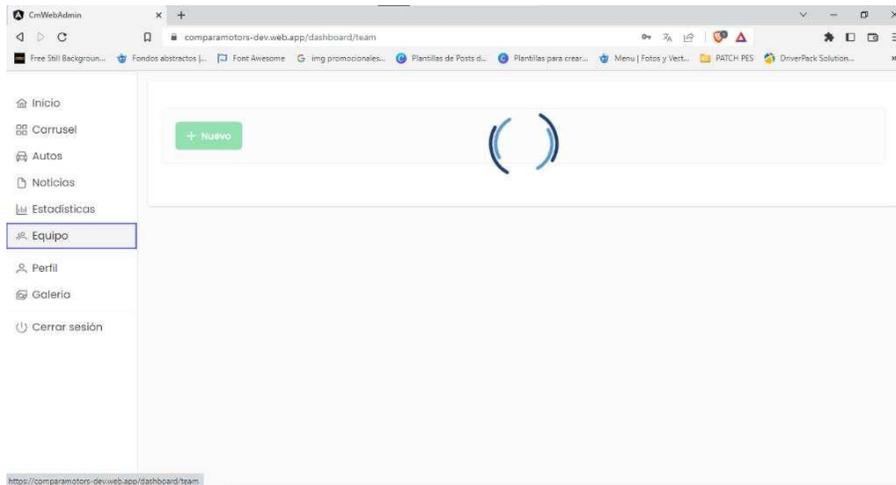
Nota. Autor (2022)

3.5.3. Cambios realizados en la página principal de team

Primeramente, se adiciono un spinner de carga que hiciera la experiencia de usuario mucho mejor, ya que mientras la página cargaba se quedaba en blanco.

Figura 11

Spinner Incorporado



Nota. Autor (2022)

El cual se realizó creando unas nuevas componentes en una carpeta del proyecto llamada Shared, dentro de la cual estaba la carpeta Spinner la cual está compuesta de 3 componentes (*Spinner.component.css*, *Spinner.component.html*, *Spinner.component.ts*), en donde cada uno hace una función

- *Spinner.component.ts* le definimos el nombre, el HTML y el CSS que deseamos que muestre.

- *Spinner.component.html* y *Spinner.component.css*, solo adecuamos lo que queremos que nos muestre y le damos los estilos respectivos.

Figura 12

Clases para crear el Spinner

```

cmwebadmin > src > app > shared > components > spinner > spinner.component.ts
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-spinner',
5    templateUrl: './spinner.component.html',
6    styleUrls: ['./spinner.component.css']
7  })
8  export class SpinnerComponent implements OnInit {
9
10   constructor() { }
11
12   ngOnInit(): void {
13   }
14
15 }
16

cmwebadmin > src > app > shared > components > spinner > spinner.component.css
1  .spinner {
2    width: 100%;
3    height: 100%;
4    display: flex;
5    justify-content: center;
6    align-items: center;
7    padding: 20px;
8    background-color: rgba(255, 255, 255, 0.507);
9    position: absolute;
10   top: 0;
11   left: 0;
12   z-index: 20;
13 }
14
15 .img{
16   width: 200px;
17   height: 200px;
18 }
19
20

cmwebadmin > src > app > shared > components > spinner > spinner.component.html
Go to component
1  <div class="spinner">
2    
6  </div>
7

```

Nota. Autor (2022).

El mismo proceso visto anteriormente fue realizado para la creación de las funciones del NOT FOUND y NOT RESULT, luego de la creación de estas componentes procedemos a llamarlas en la página principal de nuestro modulo team, el cual se realizó declarando las variables antes del constructor como variables de tipo Booleano decidiendo si estas estarían activadas o desactivadas, de la siguiente manera.

Figura 13

Declaración de las funciones del Spinner, NotFound y NotResult

```
public notFound: Boolean = false;
public spinner: Boolean = true;
public result: Boolean = true;

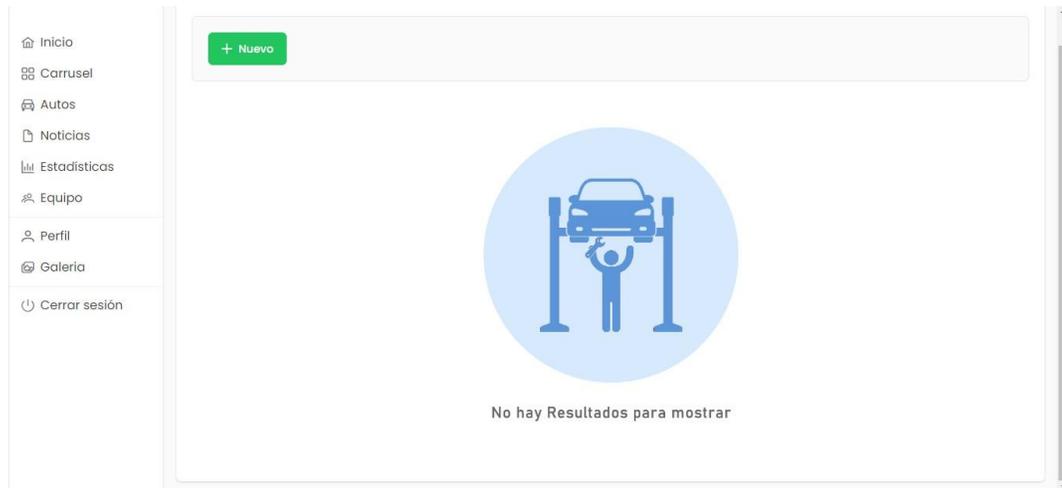
constructor(
  private confirmationService: ConfirmationService,
  private toastService: ToastService,
  private teamService: TeamService,
  private authService: AuthService
) {}
```

Nota. Autor (2022)

Las cuales serían llamadas en cada una de las funciones en las que queremos que el Spinner salga, esta fue agregada en la página principal de team a la hora de cargar todos sus elementos, también fue agregada en el módulo de crear cuando le indicábamos que creara o editara un usuario.

Figura 14

Muestra del NotResult



Nota. Autor (2022)

Otro cambio realizado en la plataforma principal de team, fueron los mensajes de alerta (ShowError, ShowSuccess, ShowInfo, ShowWarn) los cuales nos indicaban si todo se realizó correctamente, si hubo un error, advertencia y para dar información a nuestros usuarios lo cual fue implementado en esta versión luego de ver las falencias que tenía la plataforma de la versión Anterior.

Figura 15

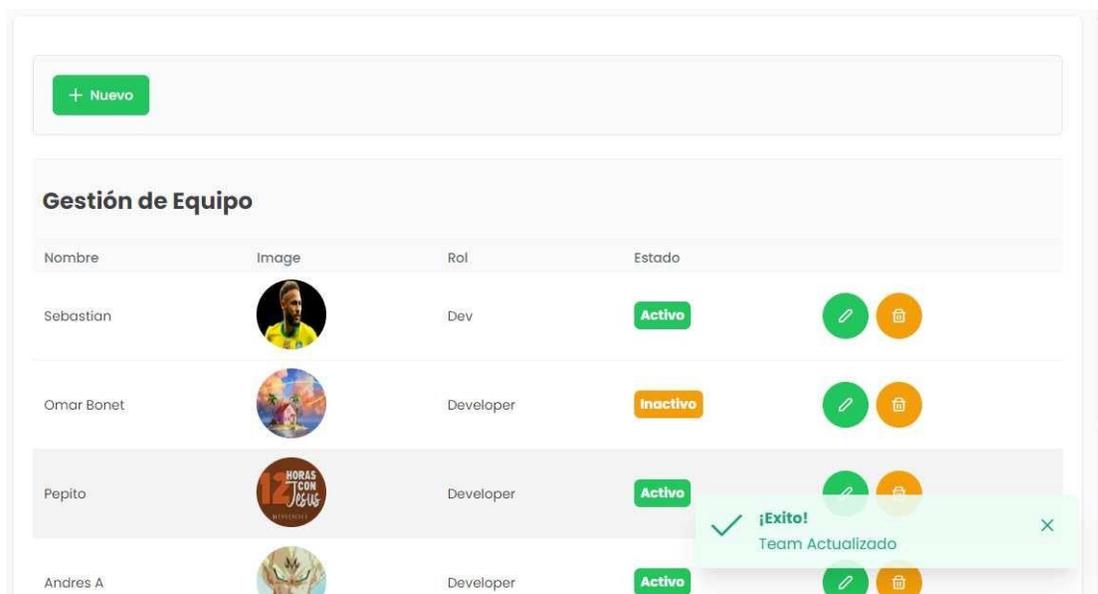
código insertando ToastService

```
78 private updateTeam(team: Team): void {
79     this.teamService.updateTeam(team).subscribe(
80         (response) => {
81             if (response.status) {
82                 this.toastService.showSuccess('Team Actualizado');
83                 this.getAllTeams();
84                 this.isOpen = false;
85                 this.teamService.setSpinnerState(false);
86                 this.resetTeam();
87             }
88         },
89         (error) => {
90             console.error(error);
91             this.teamService.setSpinnerState(false);
92             if(error.status == 403) this.authService.logout();
93         }
94     );
95 }
96 }
```

Nota. Autor (2022)

Figura 16

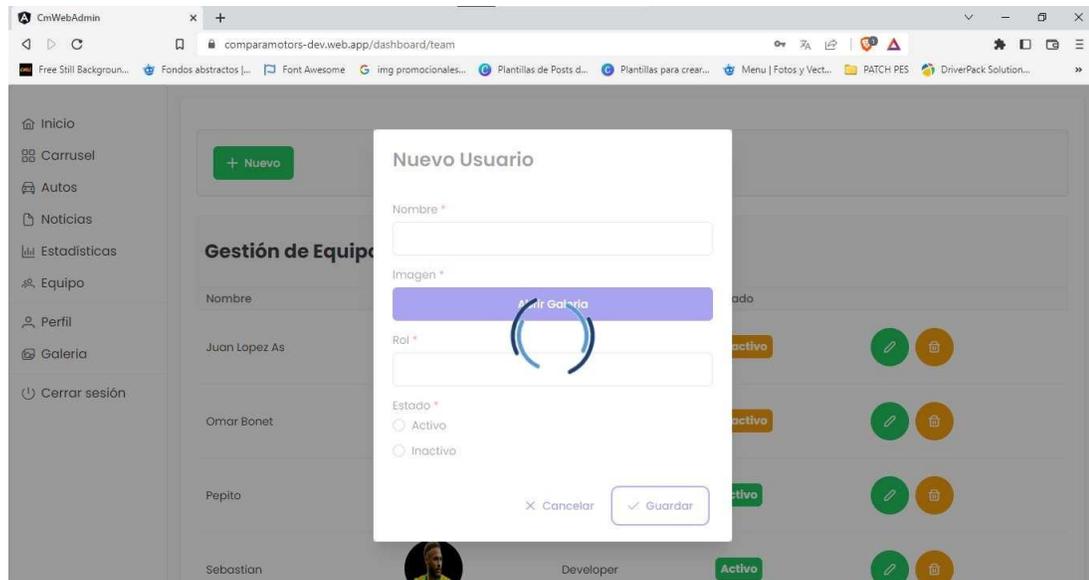
ToastService en el Modulo



The screenshot displays a web application interface for team management. At the top left, there is a green button labeled '+ Nuevo'. Below it, the title 'Gestión de Equipo' is centered. The main content is a table with columns for 'Nombre', 'Image', 'Rol', and 'Estado'. The table lists four team members: Sebastian (Dev, Activo), Omar Bonet (Developer, Inactivo), Pepito (Developer, Activo), and Andres A (Developer, Activo). Each row includes a profile picture and a set of action icons (edit and delete). A green toast notification is overlaid on the bottom right, displaying a checkmark, the text '¡Éxito!', and 'Team Actualizado'.

Nombre	Image	Rol	Estado
Sebastian		Dev	Activo
Omar Bonet		Developer	Inactivo
Pepito		Developer	Activo
Andres A		Developer	Activo

Nota. Autor (2022)

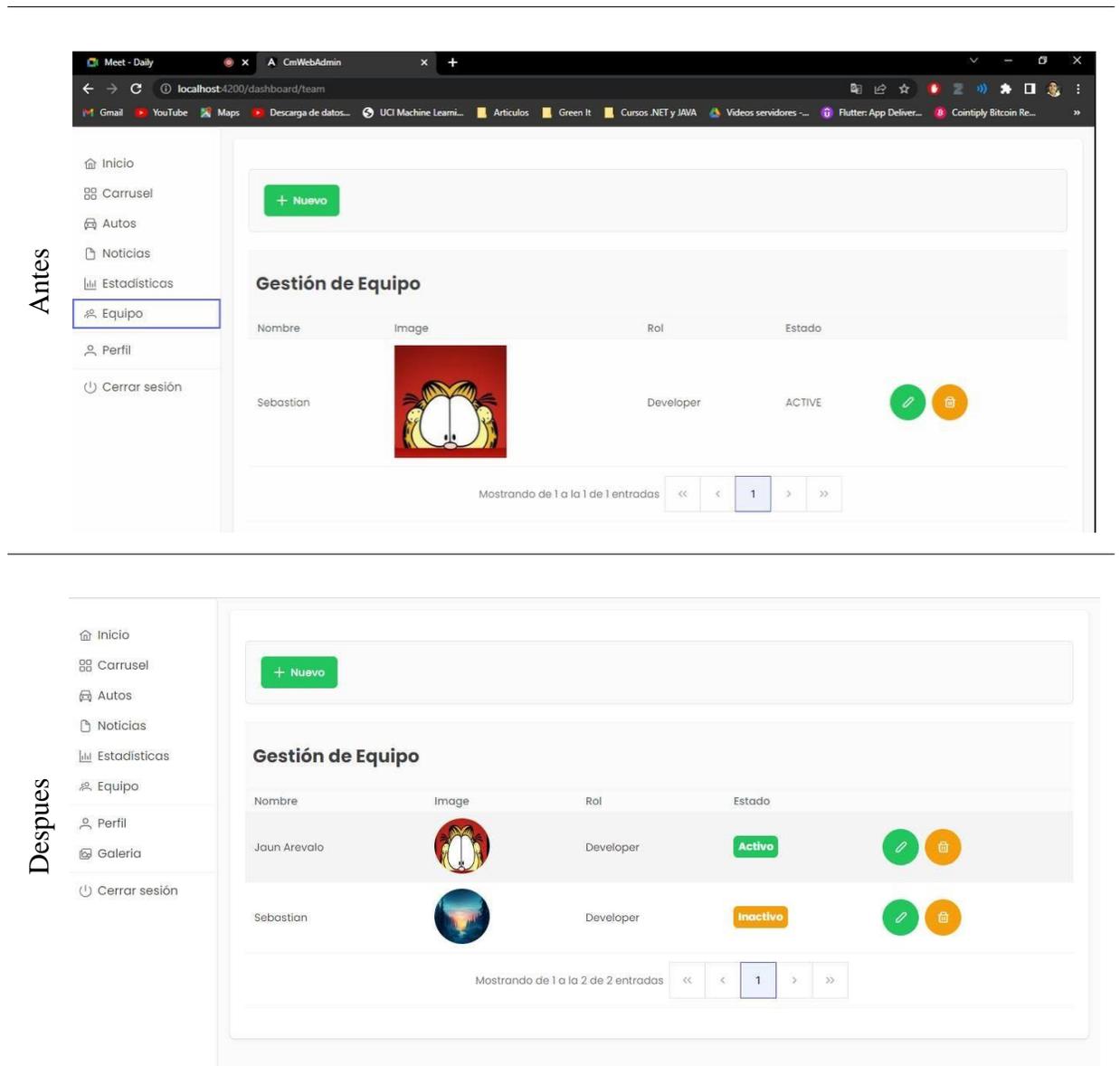
Figura 17*Spinner Insertado en el formulario*

Nota. Autor (2022)

Otra actualización de este módulo fue la vista de las imágenes, las cuales primeramente se agregaban por medio de un enlace y la plataforma las tomaba como una imagen, pero esta no se podía editar o cambiar, primeramente se agregó un botón para agregar imágenes desde una galería que el ingeniero encargado del proyecto ya tenía lista previamente, la cual agregamos cada quien a su modulo y de esta manera ya podíamos agregar fotos y también la podemos editar cuando se desee, además se le dio un estilo más adecuado a un perfil, la imagen la redondeamos y la disminuimos a un tamaño más pequeño, también el formulario donde captamos los datos tenía una función de activo e inactivo la cual no se mostraba de la mejor manera, es decir solo mostraba el texto.

Tabla 5

Antes y Después Modulo Team

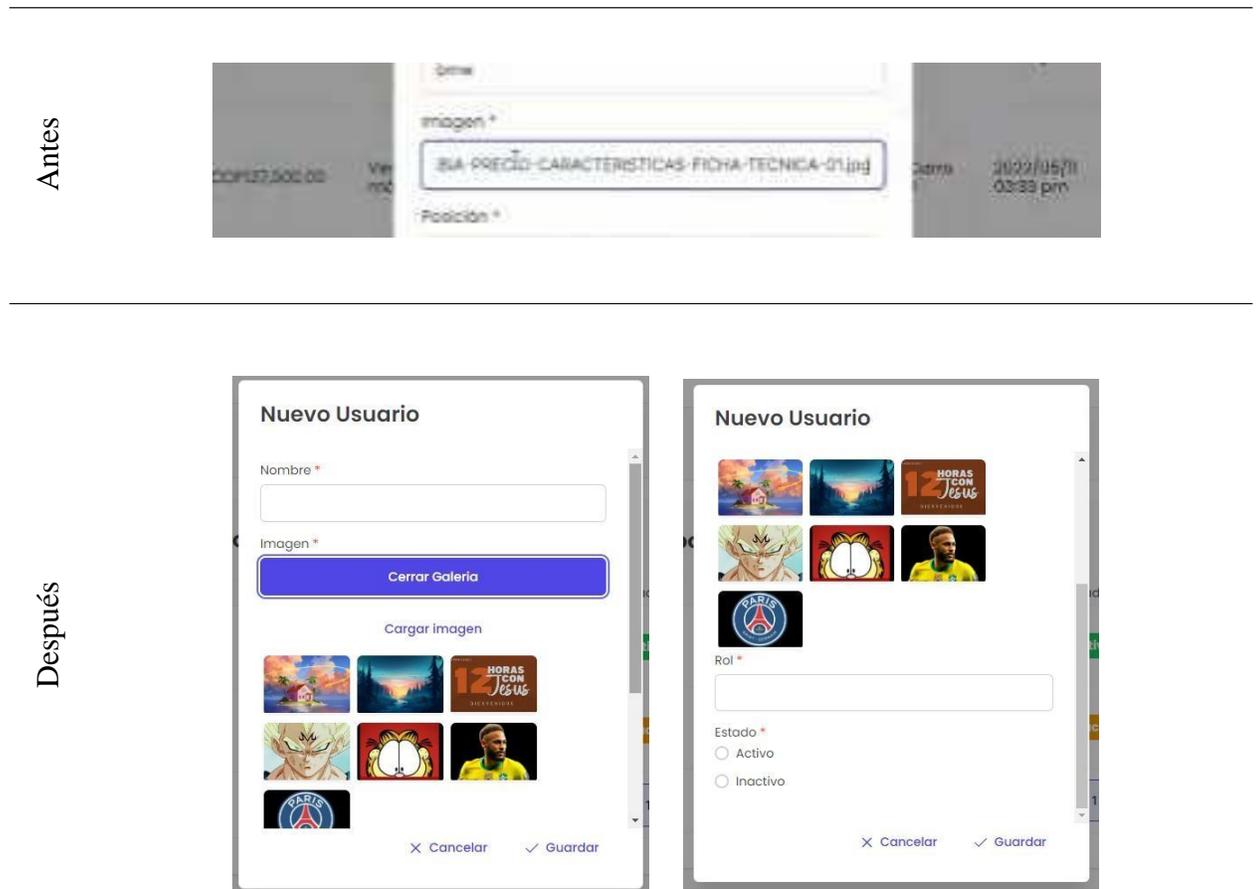


Nota. Autor (2022)

Como se puede observar hay un cambio en la interfaz en la cual lo hacemos más amigable para el usuario, una vista más agradable del módulo, además de ahorrar espacio

de la plataforma, antes un usuario ocupaba una cantidad considerable y luego de las adecuaciones dos usuarios usan el mismo que el primero.

A continuación, vemos el antes y el después de formulario, que como se mencionó anteriormente es donde captamos los datos registrados por un nuevo usuario, en el cual se agregó la galería completando así el CRUD de este módulo.

Tabla 6*Antes y Después Galería*

Nota. Autor (2022)

Otro de los cambios realizado en el módulo, es la validación de los campos la cual nos ayuda a verificar que los datos están completos y son correctos, es decir que al no colocar algún dato la plataforma no dejara que siga avanzando o no guardara o editara los datos, el cual se realiza en la función de LoadForm, cuando cargamos el formulario le decimos que verifique que los campos estén llenos, de lo contrario no deje guardar y muéstrele un mensaje de que debe llenar todos los campos

Figura 18

Código Validación de Campos

```
loadForm(): void {  
  this.form = this.fb.group({  
    id: [],  
    name: ['', [Validators.required]],  
    image: [''],  
    rol: ['', [Validators.required]],  
    status: [false, [Validators.required]],  
  })  
}
```

Nota. Autor (2023)

Figura 19

Validación, Pagina

The image shows a web application interface for creating a new user. The form is titled "Nuevo Usuario" and contains several fields: "Nombre*" (required), "Imagen*" (with an "Abrir Galeria" button), "Rol*" (required), and "Estado*" (radio buttons for "Activo" and "Inactivo"). The "Nombre*" field is empty, and a red error message "Campo requerido" is displayed below it. At the bottom of the form, there are "Cancelar" and "Guardar" buttons. In the background, a list of users is visible, and a yellow error message box at the bottom right states "Error: llena todos los campos (*)".

Nota. Autor (2023)

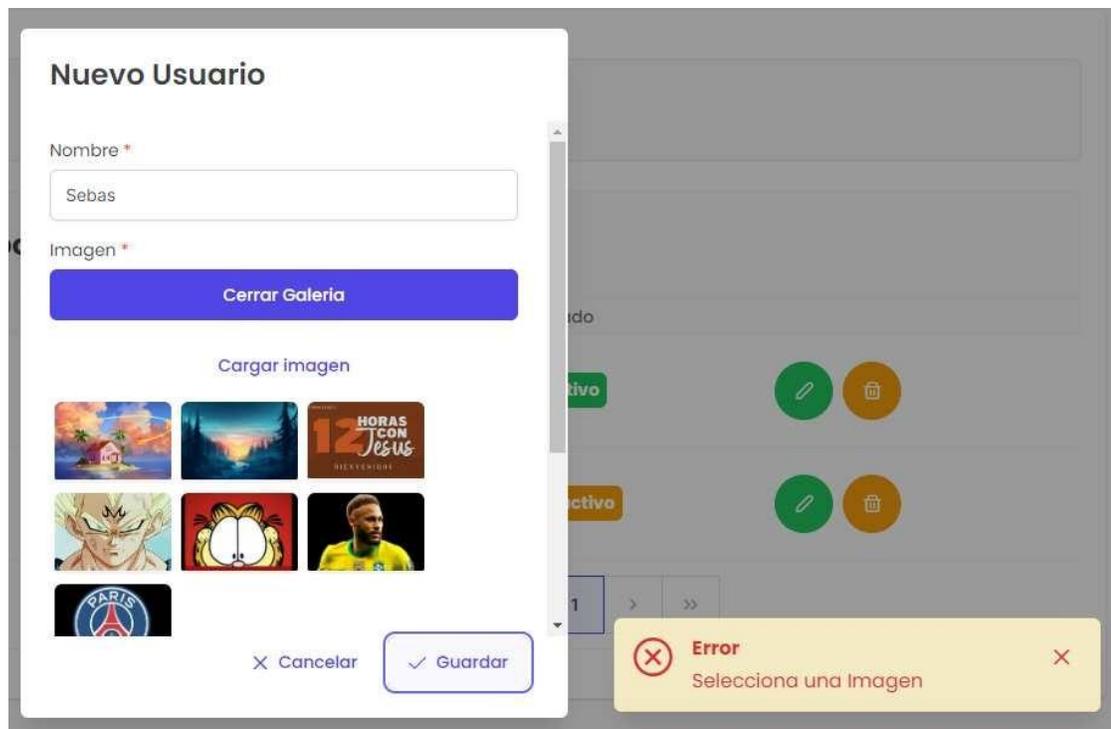
La imagen como la galería es un componente distinto se realizó la verificación de una manera distinta a los demás campos, es decir se debe escoger una imagen para poder registrar al nuevo usuario, la cual se hizo creando un nuevo método en el que estipulábamos con un IF que si el usuario tenía en imagen un campo vacío le enviara un mensaje de error diciendo que debe escoger una imagen.

Figura 20

Código Validación Galería

```
public validateForm(): void {  
    if(this.team.id && this.imageSelect==''){  
        this.imageSelect = this.team.image;  
    }  
    if(this.imageSelect == ''){  
        this.toastService.showError('Selecciona una Imagen');  
        return;  
    }  
}
```

Nota. Autor (2023)

Figura 21*Validación Galería, Pagina*

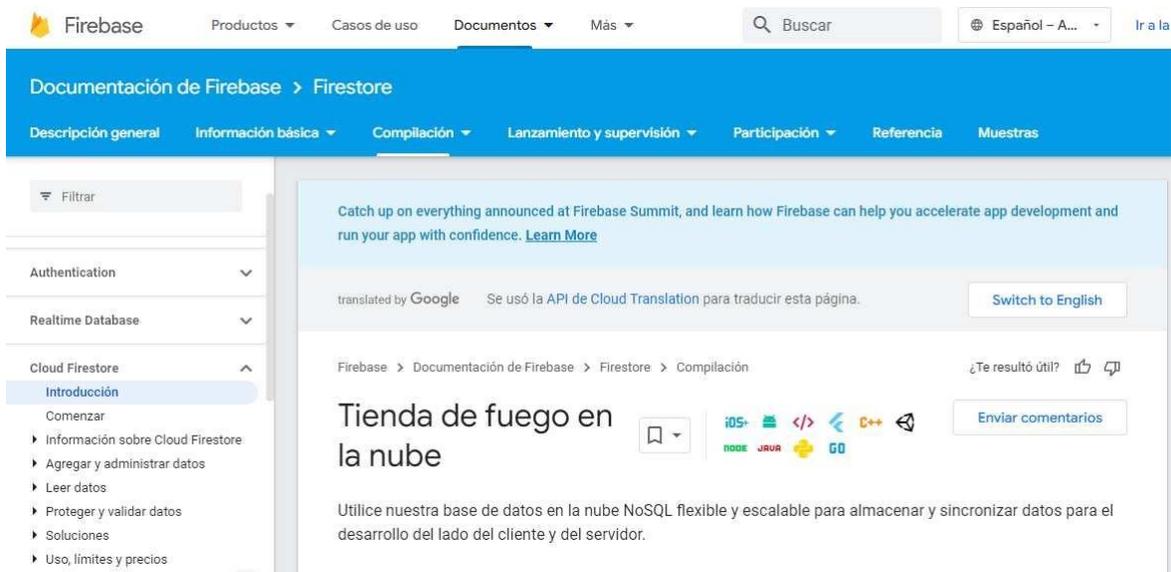
Nota. Autor (2023)

3.6. Desarrollo BackEnd

Primeramente, iniciamos este proceso con el estudio de la librería de cloud firestore, de firebase, donde empezamos con la introducción de esta librería. Donde prendimos cosas nuevas y fue de gran ayuda para la terminación de este proyecto.

Figura 22

Documentación Estudiada Firestore

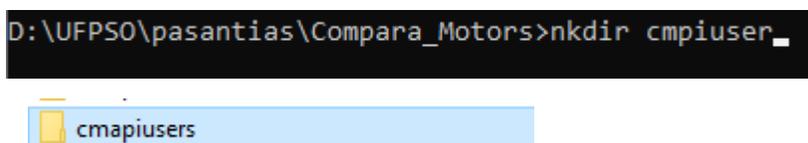


Nota. Autor (2023)

Seguidamente, vamos a crear el proyecto, el cual lo hacemos desde la consola por medio de comandos, creando así una carpeta donde vamos a guardar el proyecto del Backend para ello usamos el siguiente comando: `Nkdir nombre de la carpeta.`

Figura 23

creación de Carpeta Cmapiuser



Nota. Autor (2023)

Luego entramos a la carpeta y verificamos con que usuario estamos trabajando en Firebase y arrancamos con el servicio, donde nos mostrara que queremos instalar donde escogeremos lo que necesitamos para poder realizar este proyecto, le damos enter y escogemos el proyecto donde queremos trabajar.

Figura 24

Verificación de Usuario en la base de Datos

```
D:\UFPSO\pasantias\Compara_Motors\cmapiusers>firebase login
Already logged in as sortizp@ufpso.edu.co

D:\UFPSO\pasantias\Compara_Motors\cmapiusers>firebase init

#####  #####  #####  #####  #####  #####  #####
##      ##      ##      ##      ##      ##      ##      ##
#####  ##  #####  #####  #####  #####  #####  #####
##      ##      ##      ##      ##      ##      ##      ##
##      #####  ##  #####  #####  ##      ##  #####  #####

You're about to initialize a Firebase project in this directory:

  D:\UFPSO\pasantias\Compara_Motors\cmapiusers

Before we get started, keep in mind:

  * You are initializing within an existing Firebase project directory

? Are you ready to proceed? Yes
? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your choices. (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
( ) Remote Config: Configure a template file for Remote Config
( ) Realtime Database: Configure a security rules file for Realtime Database and (optionally) provision default instance
>( ) Firestore: Configure security rules and indexes files for Firestore
( ) Functions: Configure a Cloud Functions directory and its files
( ) Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
( ) Hosting: Set up GitHub Action deploys
(Move up and down to reveal more choices)
```

Nota. Autor (2023)

3.6.1. Instalación de dependencias necesarias

Lo que hicimos en este punto es instalar dependencias necesarias para trabajar en el proyecto, esto lo hacemos desde la terminal en la cual abrimos y entramos a la ruta de funciones donde instalaremos todas estas. Para ello hacemos uso del comando `npm install`.

Dependencias instaladas inicialmente.

- ✓ **Express:** Le decimos a nuestro programa que vamos a utilizar métodos http, es decir que me acepte código de tipo JSON, cualquier tipo de llamado como los GET, POST, DELETE

Figura 25

Instalación Componente Express

```
PS D:\UFPSO\pasantias\Compara_Motors\cmapiusers\functions> npm install express
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

up to date, audited 241 packages in 3s

15 packages are looking for funding
  run `npm fund` for details

2 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details
```

Nota. Autor (2023)

- ✓ **Cors:** habilitamos que desde cualquier ip se puedan hacer peticiones a la base de datos.

Figura 26*Instalación Componente Cors*

```
PS D:\UFPSO\pasantias\Compara_Motors\cmapiusers\functions> npm install cors
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

up to date, audited 241 packages in 1s

15 packages are looking for funding
  run `npm fund` for details

2 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
PS D:\UFPSO\pasantias\Compara_Motors\cmapiusers\functions> |
```

Nota. Autor (2023)

3.6.2. Inicialización del admin

Teniendo como base código de java script hacemos este proceso aplicando los conocimientos adquiridos durante la carrera y además bajo la guía de nuestro jefe inmediato, editamos el archivo llamado index.js.

Figura 27

Código Para inicializar el Admin

```
1  const functions = require("firebase-functions");
2  const admin = require('firebase-admin');
3
4  const express = require("express");
5  const cors = require("cors");
6  const api = require("../app/comparamotors/comparamotors.controller");
7
8  const app = express.Router();
9  app.use(express.urlencoded({extended: true}));
10 app.use(express.json());
11 app.use(cors({origin: true}));
12
13 app.use("/api", api);
14
15 admin.initializeApp();
16
17
18 exports.apiUsers = functions.https.onRequest(app);
19
```

Nota. Autor (2022)

Luego de hacer nuestra base y verificar que no tenga errores, lo siguiente que hacemos es con el comando `npm run dev` lanzamos el servicio y de esta manera y podemos hacer consultas a la base de datos.

Figura 28

Lanzamiento del Servicio

```
PS D:\UFPSO\pasantias\Compara_Motors\cmapiusers\functions> npm run dev
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

> dev
> firebase serve

=== Serving from 'D:\UFPSO\pasantias\Compara_Motors\cmapiusers' ...

+ functions: Using node@16 from host.
i functions: Watching "D:\UFPSO\pasantias\Compara_Motors\cmapiusers\functions" for Cloud Functions...
i hosting: Serving hosting files from: public
+ hosting: Local server: http://localhost:5000
! functions: package.json indicates an outdated version of firebase-functions. Please upgrade using npm install --save firebase-functions@latest in your functions directory.
! functions: Please note that there will be breaking changes when you upgrade.
+ functions: Loaded functions definitions from source: apiUsers.
+ functions[us-central1-apiUsers]: http function initialized (http://localhost:5001/comparamotors-dev/us-central1/apiUsers).
```

Nota. Autor (2023)

3.6.3. Uso de la plataforma Postman

Postman es una aplicación que nos permite hacer pruebas API. Es un cliente HTTP que nos permite probar "solicitudes HTTP" a través de una interfaz gráfica de usuario de la cual recibiremos diferentes tipos de respuestas que luego serán validadas.

Nos ofrece muchos métodos para interactuar con los 'endpoints'. Los más utilizados y sus funciones son:

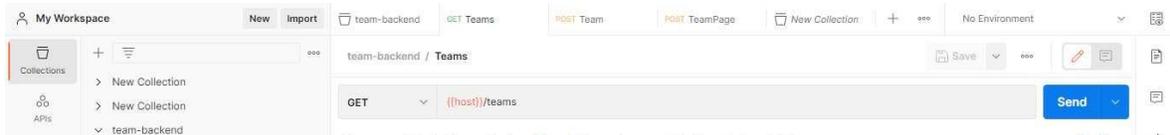
- ✓ GET: Obtener información
- ✓ POST: Agregar información
- ✓ PUT: Reemplazar la información
- ✓ PATCH: Actualizar alguna información

- ✓ DELETE: Borrar información

Aplicación en la cual crearemos una colección en la cual estaremos haciendo todas nuestras consultas, donde definimos nuestra variable y le agregamos la url que nos muestra al lanzar el servicio.

Figura 29

Configuración Postman



Nota. Autor (2023)

3.6.4. Creación de la componente Comparamotors.controller.js

En este punto no adentramos a todo lo que tiene que ver con node.js el cual es un uso del lenguaje java, donde crearemos la funciones las cuales nos ayudaran a hacer las consultas a la base de datos, en las que nos basamos en peticiones GET y POST.

Figura 30

Petición Get

```
8
9  app.get("/conexion", (req, res) => {
10     return res.status(200).send({
11         message: "Conexion exitosa"
12     });
13 });
14
```

Nota. Autor (2023)

En la anterior imagen podemos visualizar una petición de tipo GET, la cual nos indica que tenemos conexión con nuestra base de datos, esta nos permite saber si tenemos una buena conexión indicándonos un error si no tenemos éxito y en el caso contrario nos muestra un mensaje de éxito.

Como lo mencione anteriormente esta componente me sirve para plasmar los métodos tipo POST y GET, o lo que también llamamos endpoints, en la cual se crearon tres de estos, donde verificamos que teníamos una conexión exitosa con nuestra base de datos y además podíamos consultar usuarios de nuestra plataforma web.

Figura 31

Peticiones Post usando Try Catch

```

9  app.get("/conexion", (req, res) => {
10     return res.status(200).send({
11         message: "Conexion exitosa"
12     });
13 });
14
15 app.post("/team", (req, res) => {
16     try {
17         logInfo(`${ENUM_MESSAGES.START_SERVICE} CONSULTA DE TEMAS`);
18         const valid = ModelId.validate(req.body);
19         if(valid.error != null) return APIResponseError(res, valid.error.message);
20         return team(req, res);
21     } catch (error) {
22         logError("Error consultando team => " + error);
23         return APIResponseError(res, error);
24     }
25 });
26 app.post("/teamPage", (req, res) => {
27     try {
28         logInfo(`${ENUM_MESSAGES.START_SERVICE} CONSULTA DE TEMAS`);
29         const valid = ModelPage.validate(req.body);
30         if(valid.error != null) return APIResponseError(res, valid.error.message);
31         return teamPage(req, res);
32     } catch (error) {
33         logError("Error consultando team => " + error);
34         return APIResponseError(res, error);
35     }
36 });

```

Nota. Autor (2023)

3.6.5. Creación de la componente *Comparamotors.service.js*

En este punto como en todo lo que tiene que ver con el BackEnd estamos trabajando con Node.js en donde crearemos las consultas, que deseamos que nos muestre y además podemos filtrar las consultas, además siempre usaremos un TRY y CATCH, lo cual nos sirve para manejar las excepciones, en otras palabras el TRY es para que haga la consulta, pero en caso de que exista un error es donde el CATCH entra en función para indicarme cual es el error que se presentó.

Además, cuando no quiera que me muestre un dato por seguridad de nuestros usuarios, le definimos los datos que deseamos que elimine de la consulta.

Figura 32

Eliminación de Datos a la consulta

```
18      logInfo(`${ENUM_MESSAGES.FINISH_SERVICE} CONSULTA DE TEAM`);
19      const Myteam = team.data();
20      delete Myteam.createDate;
21      delete Myteam.updateDate;
22      return APIResponseSucessFull(res, Myteam);
23
```

Nota. Autor (2023)

3.6.6. Endpoints creados

Son una parte esencial de los procesos de negocio y del desarrollo de aplicaciones en general es una pasarela que conecta los procesos del servidor de la aplicación con una interfaz externa. En otras palabras, es la dirección a la que se envían las peticiones.

Es importante mencionar cómo funciona la API para entender qué son los endpoints, API es la abreviatura de una interfaz de programación de aplicaciones y estas utilizan las API para interactuar con servicios de terceros y sus usuarios.

La API procesará la solicitud y enviará la respuesta recibida del servidor. La respuesta contendrá un código de estado, que es una parte del mensaje de respuesta HTTP

que informa al cliente del resultado de la solicitud. Por ejemplo, el código 200 se utiliza si la operación se ha completado con éxito, 500 si se ha producido un error interno del servidor.

3.6.7. Primer Endpoints

El primer endpoints o consulta realizada a la base de datos es la de consultar a un usuario por su id, cabe aclarar que para que se logre hacer esta consulta es necesario hacer usos de todas las componentes creadas, y para hacer esta consulta lo primero que hicimos fue crear la función para ello esta parte la realizamos en la componente

Comparamotors.service.js, utilizando el TRY y el CATCH para que esta sea una consulta ordenada, en la componente Comparamotors.controller.js lo que hacemos es invocar la clase exportada en el método anterior y plasmamos mensajes por medio de la consola para saber en qué fallo, en caso de que la consulta no haya tenido éxito, seguidamente hacemos uso de la componente Comparamotors.models.js en la cual le estipulamos los datos que sean requeridos.

Figura 33

Código Endpoint (Comparamotors.service.js)

```

7  exports.team = async function (req, res) {
8  try {
9      logInfo(`${ENUM_MESSAGES.START_SERVICE} CONSULTA DE TEAM`);
10     const teamget = await firestore().collection(ENUM_COLLECTIONS.TEAMS).doc(req.body.id);
11     const team = await teamget.get();
12
13     if (!team.exists) {
14         logWarn("No existen teams en la coleccion");
15         return APIResponseSucessFullEmpty(res);
16     }
17
18     logInfo(`${ENUM_MESSAGES.FINISH_SERVICE} CONSULTA DE TEAM`);
19     const Myteam = team.data();
20     delete Myteam.createDate;
21     delete Myteam.updateDate;
22     return APIResponseSucessFull(res, Myteam);
23
24 } catch (error) {
25     logError("Error consultando team => " + error);
26     return APIResponseError(res, error);
27 }
28 }

```

Nota. Autor (2023)

Figura 34

Código Endpoint (Comparamotors.controller.js)

```

15 app.post("/team", (req, res)=> {
16     try {
17         logInfo(`${ENUM_MESSAGES.START_SERVICE} CONSULTA DE TEMAS`);
18         const valid = ModelId.validate(req.body);
19         if(valid.error != null) return APIResponseError(res, valid.error.message);
20         return team(req, res);
21     } catch (error) {
22         logError("Error consultando team => " + error);
23         return APIResponseError(res, error);
24     }
25 });

```

Nota. Autor (2023)

Figura 35

Código Endpoint (Comparamotors.model.js)

```
1  const Joi = require("joi");
2
3  > exports.ModelPage = Joi.object({ ...
8    })
9
10  v exports.ModelId = Joi.object({
11    |   id: Joi.string().required()
12    | })
13
```

Nota. Autor (2023)

Figura 36

Consulta Realizada por Postman

The screenshot shows a Postman interface for a POST request to the endpoint `{{host}}/team`. The request body is a JSON object with the following content:

```
1  {
2    |   "id": "FmbZSGvT0733GjZgq581"
3    | }
```

The response status is 200 OK, with a time of 584 ms and a size of 642 B. The response body is a JSON object:

```
1  {
2    |   "status": true,
3    |   "message": "Consulta exitosa",
4    |   "data": {
5    |     |   "id": "FmbZSGvT0733GjZgq581",
6    |     |   "name": "Jaun Arevalo",
7    |     |   "status": "ACTIVE",
8    |     |   "image": "https://firebasestorage.googleapis.com/v0/b/comparamotors-dev.appspot.com/o/team%2F4931d280-38e8-9266-4be1-2185645bb84e.jpeg?alt=media&token=a4963db4-9607-40a4-8df4-d480aec4a650",
9    |     |   "rol": "Developer"
10    |   }
11  }
```

Nota. Autor (2023)

3.6.8. Segundo Endpoints

El segundo endpoints o consulta realizada a la base de datos es la de paginación, es decir que yo consultaba por algunos parámetros requeridos y el me mostraba los 2 siguientes o los 2 anteriores según como yo se lo especificase, donde se realizó el mismo proceso mencionado anteriormente, pero con parámetros y distintos los cuales veran en las siguientes imágenes.

Figura 37

Código Endpoint 2 (Comparamotors.service.js)

```

29 exports.teamPage = async function (req, res) {
30   try {
31     const limit = req.body.limit;
32     const docData = req.body.docData;
33     let teamRef = firestore().collection(ENUM_COLLECTIONS.TEAMS)
34       .where("status", "=", "ACTIVE")
35       .orderBy("name", "asc");
36     if (req.body.next) {
37       teamRef = teamRef.startAfter(docData.toString()).limit(limit);
38     } else if (req.body.back) {
39       teamRef = teamRef.endBefore(docData.toString()).limit(limit);
40     } else {
41       teamRef = teamRef.limit(limit);
42     }
43     const teamPage = await teamRef.get();
44     if (teamPage.empty) {
45       logWarn("No existen team en la coleccion");
46       return APIResponseSucessFullEmpty(res);
47     }
48     let response = [];
49     await teamPage.forEach(team => {
50       response.push({
51         idTeam: team.id,
52         image: team.data().image,
53         name: team.data().name,
54       });
55     });
56     logInfo(`${ENUM_MESSAGES.FINISH_SERVICE} CONSULTA DE TEAMS`);
57     return APIResponseSucessFull(res, response);
58   } catch (error) {
59     logError("Error consultando carro => " + error);
60     return APIResponseError(res, error);

```

```

58     } catch (error) {
59         logError("Error consultando carro => " + error);
60         return APIResponseError(res, error);
61     }
62 }
63
64

```

Nota. Autor (2023)

Figura 38

Código Endpoint 2 (Comparamotors.controller.js)

```

26 app.post("/teamPage", (req, res) => {
27     try {
28         logInfo(`${ENUM_MESSAGES.START_SERVICE} CONSULTA DE TEMAS`);
29         const valid = ModelPage.validate(req.body);
30         if(valid.error != null) return APIResponseError(res, valid.error.message);
31         return teamPage(req, res);
32     } catch (error) {
33         logError("Error consultando team => " + error);
34         return APIResponseError(res, error);
35     }
36 });

```

Nota. Autor (2023)

Figura 39

Código Endpoint 2 (Comparamotors.model.js)

```

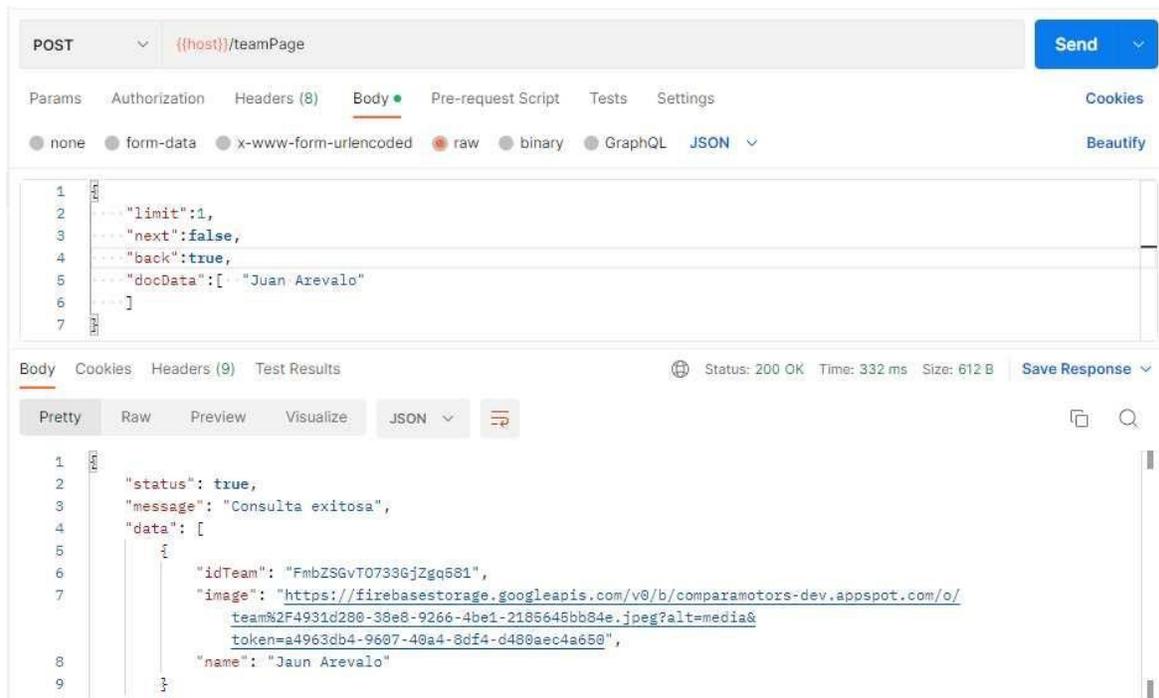
1  const Joi = require("joi");
2
3  exports.ModelPage = Joi.object({
4      limit: Joi.number().required(),
5      next: Joi.boolean().required(),
6      back: Joi.boolean().required(),
7      docData: Joi.array().required()
8  })
9

```

Nota. Autor (2023)

Figura 40

Consulta 2 Realizada por Postman



Nota. Autor (2023)

3.7. Creación de tokens

Teniendo en cuenta las recomendaciones de estudiantes que pasaron por la empresa anteriormente en la cual sugerían una mayor seguridad, dado que cualquiera podía acceder al dashboard por medio de url, lo cual quiere decir que no había una autenticación a la hora de ingresar, optando así en este proceso realizarle ese plus de seguridad a la plataforma.

Figura 41

Token Creado visto en Firebase

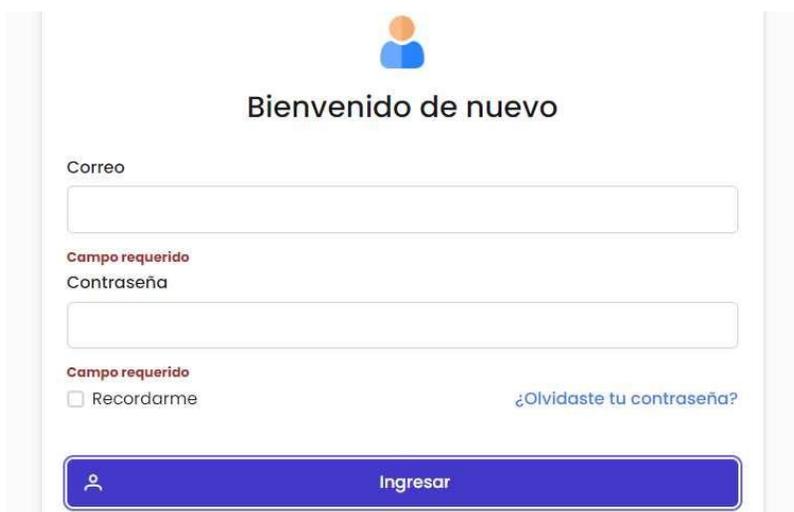
Función	Activador	Versión	Solicitudes (24 h)	Cantidad mínima o máxima de instancias	Tiempo de espera
apiAdmin us-central1	HTTP Solicitud https://us-central1-compparamotors-dev.cloud...	v1	0	0 / —	1 min
apiUsers us-central1	HTTP Solicitud https://us-central1-compparamotors-dev.cloud...	v1	0	0 / —	1 min

Nota. Autor (2023)

Se modifico en el parte del FrontEnd el login para que este hiciera una verificación primeramente que no pasara de modulo a otro por medio de la url, si no que se ingresara un usuario y contraseña, al ingresar un usuario este nos creaba un token el cual se almacenaba en la sesión Storage lo que nos indica que es un usuario autenticado.

Figura 42

Validación de Campos para iniciar Sesión




Bienvenido de nuevo

Correo

Campo requerido

Contraseña

Campo requerido

Recordarme [¿Olvidaste tu contraseña?](#)



Nota. Autor (2023)

Figura 43

Código validación Campos inicio de Sesión

```
45     this.authService.login(this.auth).then(  
46       async(response) => {  
47  
48         console.log("Login correcto");  
49         console.log(response);  
50         const user = await this.authService.currentUser();  
51         const token = await user?.getIdTokenResult();  
52         sessionStorage.setItem(AuthEnum.USERTOKEN, token?.token ? token.token : '');  
53         //this.toastService.showSuccess("Login Exitoso");  
54         this.router.navigate(["/dashboard"]);  
55  
56       }).catch(error => {  
57         //this.toastService.showError("El correo o la contraseña no son validos");  
58       })  
59     }  
60   }
```

Nota. Autor (2023)

Esto se realizó con métodos asíncronos, no permite que se ejecute una línea después de la otra, si no que al tener respuesta de la línea de código pasa a la siguiente línea de código, es decir hasta no obtener respuesta a la consulta no continua.

Desarrollo del Código

Figura 45

Código TeamController.js

```
1 import {Body, Controller, Delete, Get, Param, Post, Put, Query, UseGuards}
2 from '@nestjs/common';
3 import { CreateTeamDto, UpdateTeamDto } from './dto';
4 import { TeamService } from './team.service';
5 import { AuthGuard } from '@core/guards/auth.guard';
6
7 @Controller('team')
8 @UseGuards(AuthGuard)
9 export class TeamController {
10   constructor(private readonly teamService: TeamService) {}
11
12   @Post()
13   create(@Body() createTeamDto: CreateTeamDto) {
14     return this.teamService.create(createTeamDto);
15   }
16
17   @Get()
18   findAll(@Query('id') id: string) {
19     return this.teamService.findAll(id);
20   }
21
22   @Put()
23   update(@Body() updateTeamDto: UpdateTeamDto) {
24     return this.teamService.update(updateTeamDto);
25   }
26
27   @Delete('/:id')
28   remove(@Param('id') id: string) {
29     return this.teamService.remove(id);
30   }
31 }
32
```

Nota. Autor (2023)

Figura 46*Código AuthService.js*

```
1 import { Injectable } from '@angular/core';
2 import { AngularFireAuth } from '@angular/fire/compat/auth';
3 import { Router } from '@angular/router';
4 import { Auth } from '../interfaces/auth.interface';
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class AuthService {
10
11   constructor(
12     public auth: AngularFireAuth,
13     private route: Router
14   ) { }
15
16   login(auth: Auth) {
17     return this.auth.signInWithEmailAndPassword(auth.email, auth.password);
18   }
19
20   currentUser() {
21     return this.auth.currentUser;
22   }
23
24   authState() {
25     return this.auth.authState;
26   }
27
28   logout() {
29     sessionStorage.clear();
30     this.route.navigate(['/']);
31     return this.auth.signOut();
32   }
33 }
```

Nota. Autor (2023)

Capítulo 4. Entrega de modulo finalizado

Antes de realizar esta entrega final, se le hizo pruebas a la plataforma que todas las funciones que se nos habían delegado, se cumplieran en su totalidad, tanto el FrontEnd como el BackEnd, aclarando que las pruebas se le realizaron al desarrollo Front, en el cual se realizaron cambios y se quitaron funciones que no estaban bien implementadas o no eran necesarias.

4.1. Pruebas Realizadas

- ✓ Se verifico la seguridad, que no se pudiera pasar del login al módulo dashboard por cambio de url, si no con un usuario autenticado y registrado en la base de datos.
- ✓ Que en el módulo asignado el spinner cargara los servicios
- ✓ En el module de Team que la imagen de perfil fuera más llamativa tanto en el módulo como el formulario de editar.
- ✓ Que al Crear, editar y eliminar la plataforma nos mostrara por medio de mensajes que la acción se realizó con éxito o error al realizarla.
- ✓ Al crear un usuario, mientras la plataforma lo hacia el spinner apareciera cargando sobre el formulario.
- ✓ Que cada botón del módulo este cumpliendo con su respectiva función
- ✓ En el formulario de creación se verificará que los campos estaban completos y si no era así que enviara mensaje de por favor rellene los campos obligatorios.

Al realizarse estas pruebas y verificar que la plataforma estaba funcionando se procedió a subir los cambios realizados a la rama de cada desarrollador por medio de la consola de nuestro editor de código al repositorio de la empresa Comparamotors, donde el ingeniero o jefe inmediato merchaba con la rama develop para así lanzar a producción la nueva actualización de esta plataforma web.

Figura 47

Entregas ensambladas con la rama develop

```

haminton@MBP-de-Haminton cmWebAdmin % ng build
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

Initial Chunk Files | Raw Size | Names
vendor.js           | 6.94 MB | vendor
scripts.js          | 429.43 kB | scripts
styles.css          | 360.92 kB | styles
polyfills.js        | 123.39 kB | polyfills
main.js             | 63.81 kB | main
runtime.js          | 13.10 kB | runtime
                    | 7.91 MB | Initial Total

Lazy Chunk Files   | Raw Size | Names
src_app_modules_dashboard_pages_cars_cars_module_ts.js | 210.61 kB | pages-cars-cars-module
arya-orange.css    | 183.91 kB | arya-orange
tailwind-light.css | 175.25 kB | tailwind-light
default-src_app_modules_dashboard_pages_gallery_gallery_module_ts.js | 137.94 kB | pages-carousel-carousel-module
src_app_modules_dashboard_dashboard_module_ts.js        | 119.70 kB | modules-dashboard-dashboard-module
default-node_modules_firebase_compat_auth_dist_index_esm_js.js | 86.40 kB | firebase-compat-auth
src_app_modules_dashboard_pages_carousel_carousel_module_ts.js | 67.95 kB | pages-carousel-carousel-module
src_app_modules_dashboard_pages_news_news_module_ts.js  | 53.75 kB | pages-news-news-module
src_app_modules_dashboard_pages_team_team_module_ts.js  | 52.96 kB | pages-team-team-module
src_app_modules_dashboard_pages_stats_stats_module_ts.js | 42.42 kB | pages-stats-stats-module
src_app_modules_auth_auth_module_ts.js                  | 16.78 kB | modules-auth-auth-module
src_app_modules_dashboard_pages_profile_profile_module_ts.js | 9.37 kB | pages-profile-profile-module
src_app_modules_dashboard_pages_home_home_module_ts.js  | 8.46 kB | pages-home-home-module
src_app_modules_dashboard_pages_cars_pages_create_car_create_car_module_ts-src_app_modules_da-afb124.js | 6.57 kB | modules-dashboard-pages-cars-pages-create-car-create-car-module
common.js                                                 | 5.73 kB | common

Build at: 2022-12-19T23:50:07.625Z - Hash: 97b05a108e8eb02c - Time: 8470ms

./src/styles.css.webpack[javascript/auto]!./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[5].rules[0].oneOf[0].use[1]!./node_modules/postcss-loader/dist/cjs.js??ruleSet[1].rules[5].rules[0].oneOf[0].use[2]!./src/styles.css - Warning: Module Warning (from ./node_modules/postcss-loader/dist/cjs.js):
Warning
(1:1) postcss-import: /Users/haminton/Documents/proyectos/Comparamotors/cmWebAdmin/src/assets/styles/colors.css is empty

./src/styles.css.webpack[javascript/auto]!./node_modules/css-loader/dist/cjs.js??ruleSet[1].rules[5].rules[0].oneOf[0].use[1]!./node_modules/postcss-loader/dist/cjs.js??ruleSet[1].rules[5].rules[0].oneOf[0].use[2]!./src/styles.css - Warning: Module Warning (from ./node_modules/postcss-loader/dist/cjs.js):
Warning

```

Nota. Autor (2023)

Capítulo 5. Diagnostico Final

En este apartado se evidenciará la parte crítica y objetiva del plan de trabajo realizado, de igual manera se observará una descripción detallada de las temáticas y énfasis abordados, así mismo los logros que se pudieron obtener con el desarrollo de la página web.

La empresa es un escenario en el cual se nos permitió enriquecer y profundizar en el conocimiento del estudiante de ingeniería de sistemas en el área de desarrollo de software (desarrollo Web). fortaleciendo las habilidades y competencias previamente adquiridas en la creación, actualización de páginas web, codificación y manejo de base de datos. Logrando una interacción directa con distintos framework y lenguajes de programación.

Como se mencionó anteriormente al ser una página la cual oferta servicios en el campo automotriz se logró detectar diversas problemáticas como la escalabilidad de la antigua página, así mismo la página no contaba con la creación de usuarios para los consumidores. Por otra parte, es importante mencionar que se realizó un manejo de una estructura modularizada para la escalabilidad del sitio web, se realiza un manejo de base de datos (firebase) para la información que se quiera agregar por parte de los administradores de la página del mismo modo se manejó para la creación de usuarios.

En base a lo anterior, el ingeniero en formación detecta la necesidad de continuar con la implementación de la página con su detallada escalabilidad puesto que permite la maximización del sitio web, en el cual se realizó un acompañamiento constante por parte de los jefes para dar un seguimiento al trabajo realizado, así mismo se realizó una labor bastante enriquecedora, y se logró tener un acercamiento a la vida laboral, de igual manera se obtuvo una experiencia muy valiosa con cada uno de los trabajos realizados para la empresa.

Capítulo 6. Conclusiones

Es importante mencionar que se dio cumplimiento a los objetivos establecidos dentro del plan de trabajo para el proceso de pasantías, el cual se elaboró en base a la detección de necesidades orientadas por el jefe inmediato, así mismo se logró cumplir con el objetivo inicial planteado, el cual tuvo como fin: Desarrollar la plataforma web (Versión 2.0) mediante el Framework de Angular, para la empresa COMPARA MOTORS, ubicada en la ciudad de Ocaña, Norte de Santander, permitiendo un enfoque óptimo e integral de su razón social y su sostenimiento a lo largo de los años.

Se realizó un análisis el estado de la plataforma digital de la empresa Compara Motors para determinar que funciones demandan actualización y que funciones requieren ser incorporadas realizando reuniones con el equipo de trabajo y llegando a un acuerdo de ambas partes, logrando así una excelente comunicación y un buen ambiente laboral.

Se logro determinar el diseño de las plataformas y sus funcionalidades codificando distintas vistas y funciones que se vieron reflejadas en los encuentros diarios con el jefe inmediato para la entrega del trabajo final.

Para finalizar cabe resaltar que el conocimiento compartido con el equipo de trabajo permitió adquirir aprendizajes habilidades y competencias que aportan a un crecimiento tanto personal como profesional.

Capítulo 7. Recomendaciones

Primeramente, se recomienda seguir dando continuidad a los temas propuestos en este plan de trabajo, haciendo énfasis en la experiencia de usuario y se siga implementando la escalabilidad de la plataforma web, junto al manejo de framework los cuales ayudan a un mejor funcionamiento y a un desarrollo más eficiente.

Se sugiere que en el módulo de team, se pueda guardar información sin necesidad de colocar una imagen, que la plataforma si no escoge u sube imagen esta le otorga una por defecto.

Se recomienda que se trabaje en el módulo de perfil para que el usuario pueda cambiar su contraseña y también sus datos, es decir que no todo se haga desde el mismo modulo team, que se le dé una buena funcionalidad al módulo mencionado al inicio de esta recomendación.

Es importante mencionar el buen uso de la información que se añada a los distintos campos de la página para que esta sea más comercial y atractiva a la vista de los consumidores.

De igual manera se sugiere que se dé continuidad con el convenio ya que se evidencian resultados positivos en cuanto al desarrollo web y que las actividades realizadas contribuyeron positivamente a la empresa a nuestra vida como profesionales.

Se sugiere que se mire como está cargando los servicios la plataforma, de manera que esta lo haga de una manera más eficiente, ya que demora al menos 30 segundos para cargarlos y esto puede hacer que los usuarios pierdan interés dado que deben esperar.

Capítulo 8. Referencias

Comparamotors. (2022). Comparamotors website. <https://comparamotors.com/services/>

Decreto 1360 de 1989. Por el cual se reglamenta la inscripción de soporte lógico (software) en el Registro Nacional del Derecho de Autor. Artículo 1. Artículo 2. 23 junio 1989.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=10575>

Ley 1581 de 2012. Por la cual se dictan disposiciones generales para la protección de datos personales. Artículo 1. 17 octubre 2012.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>

Ley 1341 de 2009. Por la cual se definen principios y conceptos sobre la sociedad de la información y la organización de las Tecnologías de la Información y las Comunicaciones TIC, se crea la Agencia Nacional de Espectro y se dictan otras disposiciones. Artículo 1. 30 julio 2009.
<https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=36913#:~:text=Por%20la%20cual%20se%20definen,y%20se%20dictan%20otras%20disposiciones.>

Ley 527 de 1999. Por medio de la cual se define y reglamenta el acceso y uso de los mensajes de datos, del comercio electrónico y de las firmas digitales, y se establecen las entidades de certificación y se dictan otras disposiciones. Artículo 2. 18 agosto de 1999.
https://www.redjurista.com/Documents/ley_527_de_1999_congreso_de_la_republica.aspx#/

Martínez Villalobos , G., Camacho Sánchez, G. D., & Biancha Gutierrez , D. A. (Abril de

2010). *DISEÑO DE FRAMEWORK WEB PARA EL DESARROLLO DINÁMICO*

DE APLICACIONES. Obtenido de Universidad Tecnológica de Pereira:

<http://ojs.urbe.edu/index.php/telematique/article/view/117/111>

Ortega, D., Guevara , M., & Benavides, J. (17 de Mayo de 2017). *ELEMENTARY: UN*

FRAMEWORK DE PROGRAMACIÓNWEB. Obtenido de Revista Electronica de

Estudios Telematicos:

<http://ojs.urbe.edu/index.php/telematique/article/view/117/111>

Pérez Ibarra, S., Quispe, J., Mullicundo, F., & Lamas, D. (17 de Junio de 2021). *repositorio*

institucional de la UNLP. Obtenido de

<http://sedici.unlp.edu.ar/handle/10915/120476>

Zuñiga Vasquez, L. R. (11 de Enero de 2020). *Desarrollo de aplicaciones web utilizando*

Angular como framework. Obtenido de UNIVERSIDAD POLITÉCNICA DE

SINALOA:

<http://repositorio.upsin.edu.mx/formatos/A021ZUNIGAVAZQUEZLUISRAMON9>

716.pdf