DESARROLLO DE APLICACIÓN WEB PARA LA GESTIÓN DOCUMENTAL Y CONTRACTUAL DE LA FUNDACIÓN GIMNASIO MODERNO DEL CAUCA

ALEJANDRO POLANCO ANDRADE

CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SISTEMAS INFORMÁTICOS
PASANTÍA
POPAYÁN
2022

DESARROLLO DE APLICACIÓN WEB PARA LA GESTIÓN DOCUMENTAL Y CONTRACTUAL DE LA FUNDACIÓN GIMNASIO MODERNO DEL CAUCA

ALEJANDRO POLANCO ANDRADE

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director: Gabriel Ángel Osorio Hoyos

Ingeniero de sistemas

CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA

FACULTAD: INGENIERÍA

PROGRAMA: INGENIERÍA DE SISTEMAS

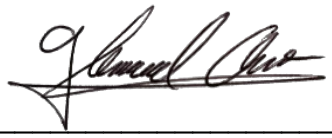
PASANTÍA

POPAYÁN

2022

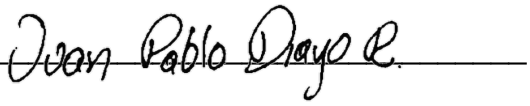
Nota de aceptación

El director y los Jurados han leído el presente documento, escucharon la sustentación del mismo por su autor y lo encuentran satisfactorio.

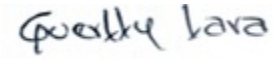


Ing. Gabriel Ángel Osorio Hoyos

Director



Presidente del Jurado



Jurado

Jurado

Popayán, abril de 2022

AGRADECIMIENTOS

Agradezco en primer lugar a mis padres por brindarme la oportunidad de tener una educación de alta calidad en la cual pude aprender, desarrollarme como persona y profesional en un ámbito que me encanta como lo es la ingeniería de sistemas informáticos, seguido deseo destacar a mi universidad la corporación autónoma del cauca, entidad que me abrió las puertas para formarme como profesional y así lograr adquirir mis conocimientos como ingeniero.

CONTENIDO

RESUMEN

El presente documento es producto del trabajo realizado en las instalaciones de la Fundación Gimnasio Moderno del Cauca, las cuales se encuentran ubicadas en la ciudad de Popayán, capital del departamento del Cauca, con el objeto de implementar una aplicación web para la gestión documental y contractual de la Fundación Gimnasio moderno del Cauca; partiendo desde la metodología de desarrollo ágil XP y marco de trabajo SCRUM. Promulgando con ello apoyar los diferentes procesos, procedimientos de sistematización y modernización dentro de la institución anteriormente mencionada.

Para el desarrollo del trabajo, se aplicaron diversos conocimientos los cuales permitieron brindar una solución óptima para la problemática que acusaba la institución. Se elaboraron dos módulos dentro de los cuales se utilizaron librerías y tecnologías diferentes que permitieron una comunión entre los mismos.

El primer módulo fue de gestión documental, aquí se hizo una inyección de dependencias bastante amplia, se usaron tecnologías de Backend tales como Nodejs, Express, Mongo DB y en el frontend se empleó una tecnología vanguardista muy utilizada en la actualidad a nivel mundial como lo es React y sus diversas dependencias y librerías que se especificarán en el transcurso de este documento.

El segundo módulo, enfocado en la contratación institucional a nivel nacional, es de vital importancia para la institución debido a su gran repercusión en el área contractual y legal del Gimnasio Moderno del Cauca, se usaron tecnologías JavaScript, TypeScript, CSS y React enfocándonos en el frontend del aplicativo web.

Palabras claves: Modernización, sistematización, Metodología ágil XP, SCRUM, Desarrollo web

ABSTRACT

This document is the product of the work carried out in the facilities of the Gimnasio Moderno del Cauca Foundation, which are located in the city of Popayán, capital of the department of Cauca, in order to implement a web application for document and contractual management of the Modern Gymnasium Foundation of Cauca; starting from the agile XP development methodology and SCRUM framework. Promulgating thereby supporting the different processes, systematization and modernization procedures within the aforementioned institution.

For the development of the work, various knowledge was applied which allowed to provide an optimal solution for the problem that the institution accused. Two modules were developed within which different libraries and technologies were used that allowed a communion between them.

The first module was document management, here a fairly wide injection of dependencies was made, Backend technologies such as Nodejs, Express, Mongo DB were used and in the frontend an avant-garde technology was used widely used today worldwide, such as React and its various dependencies and libraries that will be specified in the course of this document.

The second module, focused on institutional contracting at the national level, is of vital importance for the institution due to its great impact on the contractual and legal area of the Gimnasio Moderno del Cauca, JavaScript, TypeScript, CSS and React technologies were used focusing on the frontend of the web application.

KEYWORDS: processes, systematization, modernization, SCRUM, agile XP development methodology, Software. Web development.

INTRODUCCIÓN

El siguiente documento presenta el trabajo de pasantía realizado en las instalaciones de la Fundación Gimnasio Moderno del Cauca, ubicada en la ciudad de Popayán, en el transcurso de varios meses dentro y fuera de la institución educativa.

La principal motivación para el desarrollo del proyecto de pasantía, es desarrollar Y aportar una herramienta que brinde la solución de un problema a la institución educativa Fundación Gimnasio Moderno del Cauca, aplicando los conocimientos adquiridos durante la realización de los estudios de la carrera profesional, orientada hacia el desarrollo de aplicaciones, en este caso específico, dirigido al desarrollo web y siendo así, tener la posibilidad de aportar nuevas herramientas tecnológicas y diferentes conocimientos a la institución que le permitan continuar con su proceso de digitalización.

Actualmente, el mundo presenta grandes avances científicos y tecnológicos, teniendo como grandes problemáticas dentro de las empresas, el no estar a la vanguardia de la tecnología, el riesgo de caer rápidamente en la obsolescencia y atraso tecnológico, trayéndoles por ello diversos problemas como baja competitividad en el desempeño de sus funciones en comparación con la competencia y la imposibilidad de acceso a las nuevas máquinas, equipos y tecnologías introducidos en el mercado.

En este momento, la humanidad se encuentra frente a una paradoja cuyas consecuencias son aún difíciles de cuantificar. En efecto, cuando por un lado se dispone de la capacidad tecnológica para fabricar productos duraderos**,** nos encontramos con la “necesidad” de adaptarnos al cambio permanente de las tecnologías.

Ser obsoletos no cabe ahora en el panorama de ninguna compañía, por eso, hallar la respuesta a este dilema, es una obligación para la que tenemos que prepararnos si queremos sacar a flote nuestra manera de hacer negocios y ser competitivos frente al resto de compañías.

La mejor forma de hacerle frente es la digitalización de nuestras empresas, o lo que es lo mismo: modificar tecnología, procesos, habilidades TI e, incluso, los modelos de negocio, para asegurarnos un puesto en la lista de empresas que están interesadas en seguir a la par con los avances y el desarrollo

CAPITULO I: PROBLEMÁTICA

1.1 Planteamiento el Problema

La Fundación Gimnasio Moderno del Cauca (FGMC) fundada en 1983, es una entidad de derecho privado, sin ánimo de lucro, regida por la constitución y normas legales vigentes, con domicilio en la ciudad de Popayán, que tiene entre sus objetivos, la prestación directa de servicios educativos, servicios de asesoría a través de convenios interinstitucionales con entidades estatales o privadas para elaboración, ejecución de planes y proyectos educativos y ofrecer servicio de alimentación en el marco del programa de alimentación escolar en Colombia PAE, entre otros [1].

La FGMC, dentro de su proceso de modernización ha obtenido la certificación del sistema de gestión de calidad EFMQ, dentro de lo cual es importante digitalizar el proceso de gestión documental, con el que no cuenta en este momento.

El no contar con este sistema, trae a la institución los siguientes problemas:

Uso ineficiente del espacio físico: La cantidad de documentos ocupan demasiado espacio físico en la institución, los lugares de almacenamiento no son los indicados y ocupan lugares que podrían ser utilizados para puestos de trabajo más útiles en la empresa.

Pérdida de tiempo: Debido a que no existe orden en el almacenamiento físico de los papeles y la no existencia de bases de datos o inventarios documentales, buscar un documento o información es un proceso muy demorado, incrementando los costos de funcionamiento y la ineficiencia en sus procesos administrativos.

Poca productividad: La demora en obtener la información y los documentos respectivos a tiempo, conlleva a no tomar decisiones oportunas, a la no corrección

de procesos a tiempo y a una dificultosa medición de la productividad de los empleados de la Fundación Gimnasio Moderno del Cauca, lo que hace lenta la respuesta de la empresa a las solicitudes externas y poco adaptable a los cambios que se presentan continuamente en el entorno.

Requerimientos legales: En Colombia por lo general, todas las empresas privadas tienen establecido su sistema propio de gestión documental, permitiéndole optimizar las respuestas a solicitudes externas sobre sus procesos estratégicos, misionales y de apoyo.

En el mundo actual, caracterizado por ser altamente competitivo, globalizado y digitalizado, la automatización de las tareas y los procesos, se perciben como aspectos esenciales para las empresas, permitiéndoles un desarrollo económico y una transformación digital más eficiente, consistente en la aplicación de tecnologías y talentos digitales en el desarrollo de procesos y productos para incrementar la eficiencia y generar valor para la empresa y el cliente e implementar nuevas formas de generar ingresos para la empresa.

Así, el Observatorio de Recursos Humanos. (2020), afirma que: Según estudios realizados por la compañía AIIM, casi el 80% de las organizaciones reconocen que necesitan digitalizar su negocio entero para subsistir. Sin embargo, la creciente marea de información y el caos que ésta conlleva, se convierten en un duro obstáculo para las empresas que inician su transformación digital [2].

Por consiguiente, a medida que se acelera la disrupción digital, el grado de automatización de los procesos se sitúa muy por detrás de lo necesario. Así, dos tercios de las empresas informan que ni siquiera la mitad de sus procesos están automatizados y en las menos eficientes, más del 75% de todos los procesos administrativos siguen siendo completamente manuales [3]

Un proceso se automatiza utilizando tecnologías especializadas disponibles en el mercado y requieren de actividades claras y bien definidas que funcionan con información que ingresa, se genera o se transforma.

Para realizar tareas como contratación, kardex, compras, facturación, contabilidad, inventarios, se utilizan herramientas hardware/software, capaces de manejar y trabajar con grandes volúmenes de información y permiten gestionarla para brindar acciones seguridad, edición, disponibilidad, entre otras.

De igual forma, la realización eficiente del proceso de contratación y la disponibilidad permanente de la información relacionada con el mismo, son de vital importancia para la institución educativa FGMC; la cual desde el año 2006, ha realizado anualmente 580 contratos en promedio, alcanzando un mínimo de 390 en el año 2006 y un máximo de 611 en el año 2019, con cobertura en los departamentos del Cauca y Putumayo [4], en modalidades como, por ejemplo:

a) Contrato individual de trabajo a término fijo inferior a un año para docentes privados
b) Contrato de trabajo por duración de una obra o labor contratada
c) Contrato individual de trabajo a término fijo inferior a un año
d) Contrato individual de trabajo a término fijo a un año.
e) Contratos de tipificación especial.

La FGMC, presenta inconvenientes administrativos en sus procesos de contratación, que le dificultan desarrollar la misión institucional, debido a que los contratos son elaborados en un formato de procesador de texto y la información de los contratistas es gestionada en hojas de cálculo básicas, no interrelacionadas y sin un orden definido.

Estas condiciones dificultan la generación de reportes y presentan inconvenientes, tales como:

a) Aumento en los costos, debido al consumo de papelería y la contratación de personal adicional para su manejo.
b) Incremento de errores manuales, por la diversidad de formatos utilizados.
c) Falta de seguridad y confiabilidad de la información, por la pérdida de la misma debido a la manipulación de los datos realizada por diversas personas.
d) Repetición de tareas, por la ocurrencia de continuos errores humanos.
e) Alto consumo de tiempo en la elaboración de informes, por no estar la información de contratos unificada en un solo aplicativo.
f) Dificultad en la toma de decisiones, debido a que la información no está integrada ni disponible en todo momento para ser analizada por las personas que toman las decisiones.

Los problemas anteriores, generan un proceso de contratación y gestión de informes relacionados ineficiente, lo cual dificulta alcanzar los objetivos, metas estratégicas, lograr mayor productividad y rentabilidad de la empresa.

El contexto anterior, conlleva al planteamiento de la siguiente pregunta de investigación para este proyecto: *¿Cómo sistematizar la gestión documental, la elaboración de contratos y la generación de informes relacionados en la fundación Gimnasio Moderno del Cauca, para mejorar la gestión y eficiencia administrativa de la empresa?*

1.2. Justificación

La motivación para efectuar esta pasantía, se fundamenta en la oportunidad de contribuir con la solución de un problema real en la empresa, apoyar la implementación y mantenimiento del sistema de gestión de calidad obtenido por la institución, al igual que aplicar en la práctica, los distintos conocimientos adquiridos

durante el transcurso de la carrera, utilizando diferentes tecnologías y metodologías, en unión de buenas prácticas para desarrollar una aplicación software, mediante la cual la Fundación Gimnasio Moderno del Cauca(FGMC), pueda alcanzar sus objetivos de digitalización en los procesos administrativos.

Con el proyecto, se sistematizarán los procesos de gestión documental y contratación realizados en la institución. El módulo relacionado con el proceso de gestión documental, estará compuesta por:

Una sección de documentos de consulta, los cuales pueden ser todos los archivos administrativos que la entidad dispone para que las personas puedan obtener información de cualquiera de los procesos y procedimientos de las dependencias.

Una sección de descarga de documentos del sistema de calidad, donde los usuarios internos tendrán disponibles los formatos estandarizados de sus procesos y procedimientos, los cuales podrán ser descargados para su uso de acuerdo con las necesidades.

Una batería de indicadores con la cual se mide el cumplimiento de los objetivos institucionales planteados periódicamente. Estos indicadores para la Fundación Gimnasio Moderno del Cauca están relacionados con:

- Indicadores de gestión directiva y estratégica.
  - Índice de satisfacción de clientes el cual buscará Medir el porcentaje de satisfacción de los clientes de la organización.
  - Índice de Satisfacción de Colaboradores: Garantizar en el buen desempeño y la productividad de los trabajadores.

- Indicadores de gestión Tics.
  - Índice de Gestión de Información Digital: Garantizar que la información siempre sea adecuada y este guardada

- ¬ Índice de Gestión de Fallos: Eliminar las fallas dadas en la página web de la institución

- Indicadores de gestión académica.
  - ¬ Índice de continuidad en educación de los egresados Gimnasitas: Conocer la cantidad de estudiantes que continuaron con sus estudios
- Indicadores de gestión de apoyo académico.
  - ¬ Índice de Calidad y Disponibilidad del Archivo Académico: Dar el apoyo necesario a los procesos de archivo y documentos de carpetas de los estudiantes para lograr un buen funcionamiento de la institución.
  - ¬ Índice de cumplimiento del plan de trabajo anual de SST: Monitorear el estado % de cumplimiento de las actividades planteadas en el periodo.
- Indicadores de gestión financiera.
  - ¬ Índice de Ejecución Presupuestal: Realizar una adecuada gestión del presupuesto de acuerdo a las necesidades y recursos existentes

De igual forma, se implementará una sección para realizar seguimiento de las acciones correctivas, entendidas como acciones que implementa la empresa para eliminar las causas que originan las no conformidades en la institución y así prevenir la recurrencia.

Por último, se realizará una sección relacionada con el tratamiento de los riesgos detectados institucionalmente, en el cual se registrarán los diferentes riesgos detectados por el sistema de gestión de calidad de la institución y el tratamiento dado a los mismos para minimizarlos o evitar su ocurrencia [3].

El sistema de gestión documental apoyará la toma de decisiones y la gestión del conocimiento entre otras actividades. Este se compone de varios procesos entre los cuales se encuentran la adquisición o generación de los documentos, el registro de

los mismos en el sistema, la clasificación de los documentos, el almacenamiento de diversa data, el acceso a los documentos del sistema, la trazabilidad y la disposición de los documentos, una vez agotados los plazos de conservación dentro de su estructura [3].

También se sistematizará el proceso de contratación dentro de la Fundación Gimnasio Moderno del Cauca (FGMC), aplicando tecnologías actualizadas, acordes con los requerimientos de la empresa y las tendencias del mundo actual.

El aplicativo se desarrolla por la necesidad que presenta la institución, de sistematizar los procesos de gestión documental y contratación institucional, los cuales hasta el momento se han realizado de una manera ineficiente, y por lo sensible de los procesos, requieren la implementación de una solución que brinde mayor eficiencia, seguridad de la información, que sea portable y escalable.

El aplicativo brindará a la institución, confianza en la capacidad de los procesos y en la calidad de los productos o servicios prestados, contar con trazabilidad en el proceso documental, implementar mejora continua en la gestión documental, el proceso de contratación y aumentar la satisfacción de clientes e interesados.

Adicionalmente la aplicación contribuirá a que la Fundación Gimnasio Moderno del Cauca continúe en su proceso de transformación digital el cual se viene desarrollando por política institucional.

La solución desarrollada, impactará los procesos administrativos de gestión documental, contratación y generación de informes de la institución, lo cual se verá reflejado en el logro de mayor eficiencia, agilidad, seguridad, fiabilidad, portabilidad y confiabilidad de la información, beneficiando de manera directa, con el desarrollo de la aplicación, al personal administrativo y a los contratistas que pretenden vincular a la institución.

Este proyecto es posible llevarlo a cabo, gracias a la disponibilidad de infraestructura como servidor, clientes y conectividad presentes en las instalaciones de la Fundación Gimnasio Moderno del Cauca, sobre los cuales se implantará la solución a desarrollar.

1.3. Objetivos

1.3.1 Objetivo general

- Desarrollar una aplicación web para la gestión documental, elaboración de contratos y generación de informes, requerida por la Fundación Gimnasio Moderno del Cauca, para mejorar la eficiencia, seguridad y portabilidad de estos procesos administrativos.

1.3.2 Objetivos Específicos

- Identificar los requerimientos de acuerdo con las necesidades de la institución para el desarrollo web.

- Implementar los componentes de la arquitectura del sistema para lograr un desarrollo que cumpla con los requerimientos del proyecto.

- Validar la funcionalidad del software utilizando casos de prueba en la Fundación Gimnasio Moderno del Cauca.

CAPÍTULO II: MARCO TEÓRICO

2.1 MARCO REFERENCIAL.

2.1.1 Antecedentes.

La evolución de los adelantos tecnológicos, la aparición de dispositivos como computadores, tabletas y celulares, han llevado a que se creen nuevos métodos para mejorar su funcionamiento, permitiendo hacer a través de ellos una gran cantidad de actividades que anteriormente eran inimaginables, los cuales no se deben solamente a los dispositivos mencionados sino a una herramienta encargada de darles vida y funcionalidades específicas. Así, (Globalbit, 2019) afirma que: “El software es la parte intangible de todos los dispositivos, un conjunto de comandos que dan órdenes y se encargan del completo funcionamiento de estos. Sin el software los computadores, tabletas, celulares y otros dispositivos no tendrían vida [4].

Es importante señalar, que los avances tecnológicos han causado gran impacto en las personas, pero con el transcurrir del tiempo, las empresas se han dado cuenta cuán importante es el proceso de digitalización de las mismas, volviéndose algo indispensable para el funcionamiento correcto y eficaz de cada empresa o institución que desee estar a la vanguardia en el mundo actual.

Existen diversos desarrollos o aplicaciones ligadas a la gestión documental y la contratación dentro de las empresas o instituciones, brindando diferentes servicios para poder administrar de manera más eficiente todos los temas derivados de la contratación. Como parte de la investigación realizada para este proyecto, a

continuación, se presentan algunas aplicaciones relacionadas con el desarrollo de la solución software que se pretende crear en el transcurso de la pasantía:

OpenDocMan.

OpenDocMan es un DMS de código abierto diseñado para almacenar y acceder a documentos de forma centralizada. Tiene una interfaz fácil de usar que es muy intuitiva y atrae a los usuarios. Tiene una implementación basada en Web y es compatible con Windows y Mac [5]. También puede utilizarse como una aplicación en dispositivos iOS, Android, Windows o BlackBerry.

En cuanto a las desventajas, este software simplemente se encarga de gestionar documentos dentro de lo cual almacena y reorganiza la ubicación de los archivos, tiene acceso limitado y únicamente 1 GB de almacenamiento [5].

OpenKM.

El software de gestión documental OpenKM está desarrollado para gestionar y organizar todos tus archivos y documentos digitales para simplificar tu trabajo diario y mejorar tu eficiencia. Es un software de gestión de contenido empresarial para almacenar, rastrear, editar y gestionar documentos electrónicos [6]

Referente a las desventajas, su licencia tiene un costo alto dado que para desbloquear todo el software se requieren hacer pagos ya sea mensuales o anuales, su plataforma se ha caído varias veces.

OnlyOffice.

El presente es un software gratuito de gestión de documentos personales. Puede editar documentos online y gestionarlos con sus servicios multifuncionales para reducir costes y ahorrar tiempo [7]

En cuanto a las desventajas, se presenta la compatibilidad de documentos específicamente con Microsoft Office no está 100% garantizada. Depende de los servidores del servicio y de una conexión a Internet para acceder a los documentos.

Meet Bitrix24.

Meet Bitrix24 es un software de gestión de documentos online gratuito el cual es desplegado en la web y en la nube. Ofrece a los usuarios la flexibilidad de trabajar en dispositivos Android y iOS a través de su aplicación móvil [8] .En cuanto a las desventajas se puede presentar que solo permite gestionar archivos en formato pdf, tiene accesos de gestión los cuales deben ser pagados para su posterior uso.

Comforce.

Comforce, es un software de gestión el cual administra contratos de forma rápida y sencilla. Almacena información relevante y documentos relacionados en el administrador de archivos por contrato y genera alertas o alarmas para vencimientos, garantías, pólizas, estudios [9]

Dentro de sus desventajas, se considera su implementación es bastante costosa dado que maneja aplicaciones licenciadas las cuales deben ser adquiridas por la empresa, no deja exportar archivos desde su base de datos.

Zoho Recruit.

Zoho Recruit es un servicio de gestión de contratación integral para gestionar todos los aspectos de la empresa en el proceso de contratación, es decir, la gestión de clientes y contactos, gestión de solicitudes de trabajo, seguimiento de candidatos, eventos y programación de entrevistas [10].

Referente a sus desventajas, se encuentra que no cuenta con los diversos modelos de contratación tales como: Contratos a término fijo y contratos a término fijo inferiores a un año.

ContractBook.

Es un desarrollo web que gestiona contratos digitales para empresas modernas, se puede crear contratos con plantillas, permite firmarlos con firma digital. [11]. Dentro de las desventajas, está la de no permitir descargar el contrato de manera gratuita, ya que exige un pago para poder desactivar el conversor a pdf, lo que hace que no sea ideal dado que los contratos se requieren en la base de datos de la empresa.

Vertex.

Es una aplicación diseñada por abogados, la cual brinda a elección gran variedad de contratos con todas las prestaciones legales existentes, se maneja también en GitHub y tiene interfaces modernas y vistosas [12]. Dentro de las desventajas que tiene esta aplicación es que posee demasiada demanda y por esta razón en la actualidad tiene más de 2760 asuntos sin resolver en su repositorio oficial, tiene sistemas de cobro después de cierto tiempo de utilidad. Por otra parte, los tiempos de respuesta en asuntos legales de los contratos son muy demorados.

2.2 MARCO CONCEPTUAL

2.2.1 Metodología de Desarrollo.

2.2.1.1 Scrum.

Es un marco de trabajo de metodologías ágiles que aporta flexibilidad al proceso de desarrollo de software, definiendo requerimientos considerados como pequeñas

partes del desarrollo el cual se lleva a cabo de manera iterativa e incremental. Fue diseñado para incrementar la productividad del desarrollo y puede ser utilizado en proyectos pequeños y grandes.

El proceso de este marco de trabajo consiste en Sprints, los cuales se llevan a cabo en un periodo de dos a cuatro semanas. Cada Sprint es una entrega que define una cantidad historias de usuario como requerimientos funcionales del cliente y otro tipo de tareas como mejoras, arreglo de errores, investigaciones, entre otras. Además, se llevan a cabo prácticas como reuniones diarias de 15 minutos para definir el estado de las tareas, refinamiento de las historias de usuario, planeación del Sprint, retrospectivas y limpieza [8]

2.2.2 Herramientas de integración y entrega continúa.

Un tema importante para el desarrollo de este proyecto fue la automatización de las tareas relacionadas con la integración, el despliegue y la entrega continúa del proyecto de software a la institución educativa [13].

A continuación, se presenta en qué consiste cada una de ellas y cuáles son sus beneficios:

- **Integración continua**: Su objetivo es verificar el correcto funcionamiento del proyecto antes de integrar nuevos cambios a la rama principal. Para ello, realiza tareas automatizadas como construir y ejecutar las pruebas del proyecto en un entorno aislado. Algunas de las ventajas que ofrece es la temprana detección de errores, una menor cantidad de bugs generados y menos tiempo de pruebas en el proceso de aseguramiento de calidad del software.

- **Entrega continua**: Es una extensión de la integración continua, con la diferencia de que se agrega encima de ella una tarea de automatización para el

proceso de lanzamiento de versiones. Reduce el tiempo que gasta en la preparación de un lanzamiento y permite hacerlo de manera más frecuente.

- **Despliegue continuo**: Este punto es muy importante ya que el despliegue va un paso más allá de la entrega continua y consiste en entregar el producto al cliente, en el caso de la aplicación web subir automáticamente las nuevas versiones y/o modificaciones al localhost determinado para así poder visualizar el estado del programa. Permite dar una mejor experiencia al cliente al entregar nuevas funcionalidades con más frecuencia, en lugar de hacer entregas cada semestre o trimestre [13].

2.2.3 Versionamiento en GitHub

El desarrollo de software es una tarea que implica la escritura de código por parte de su o sus integrantes con el objetivo de entregar múltiples funcionalidades, cada sprint bajo un marco de trabajo SCRUM. Existe la posibilidad de que durante el desarrollo del software el programador haga modificaciones en distintas partes del código y surjan conflictos en el momento de integrar el trabajo realizado. Usualmente existen tareas que son desarrolladas primero que otras e integradas en el código fuente del proyecto, por lo que es necesario tener un control de versiones que permita actualizar el código sin perder el trabajo que se ha adelantado. Por esta razón existen proveedores de servicios en Internet basados en GitHub [14], los cuales ofrecen herramientas para almacenar el código en sus servidores, comúnmente llamados repositorios, además de la posibilidad de tener diferentes versiones llamadas ramas y puntos de modificación llamados commits.

Se presenta a continuación la figura 1 y 2.

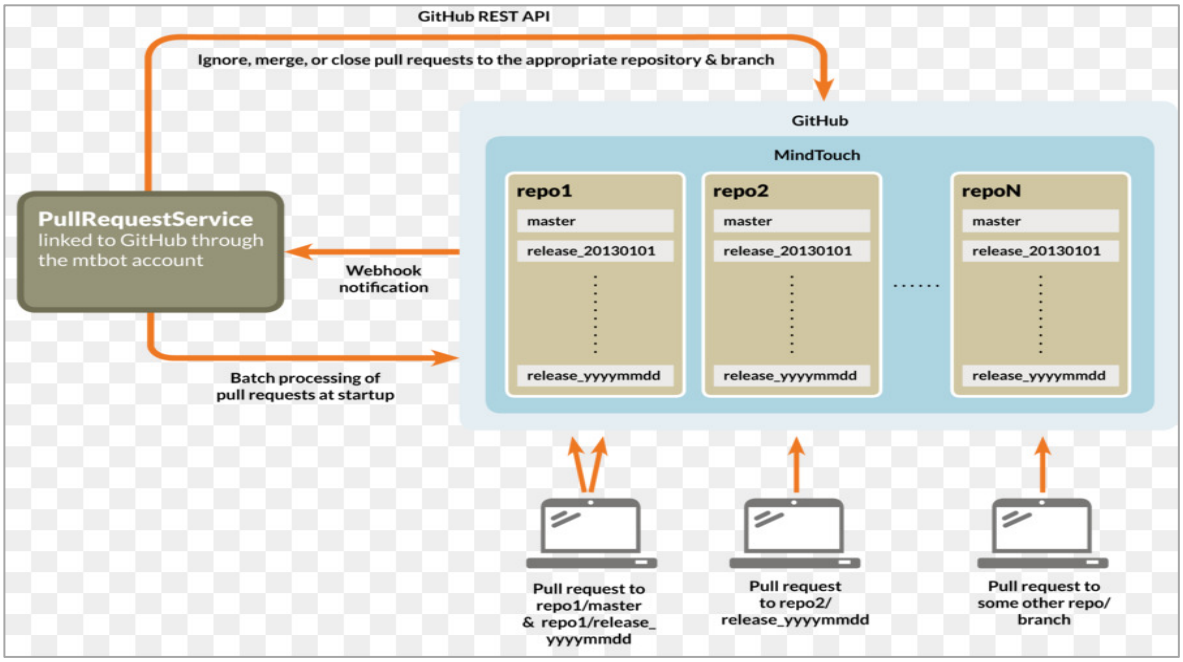


Figura 1. **Versionamiento en GitHub**
Fuente: The GitHub Blog [14] https://www.freepng.es/png-smvi1g/

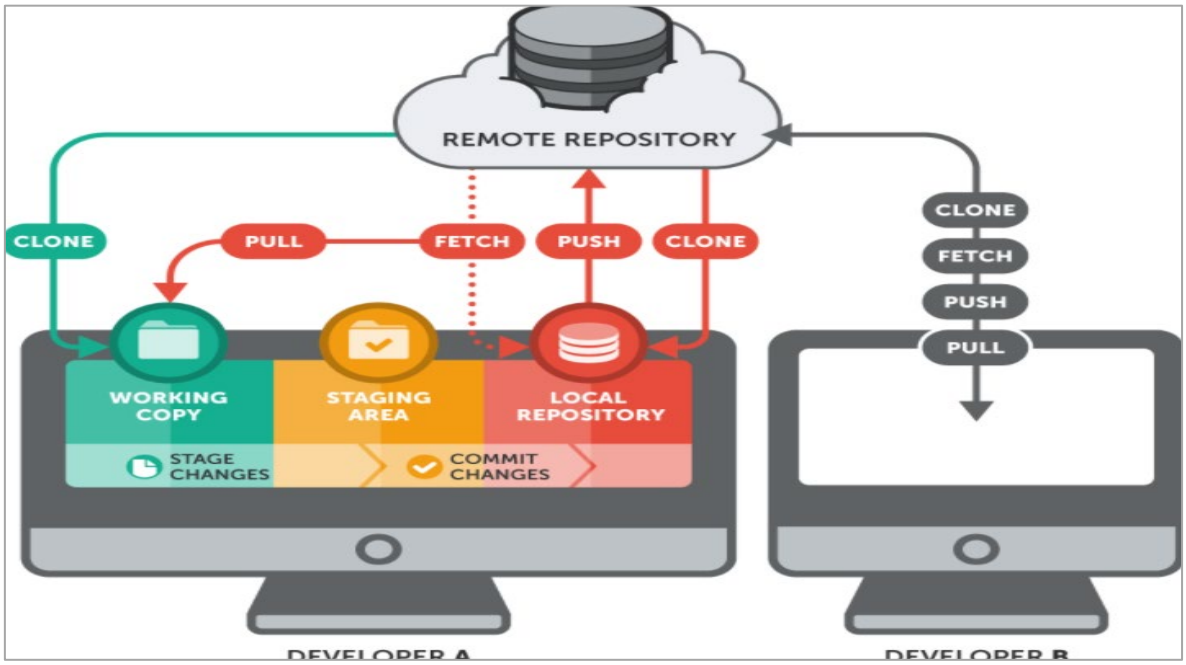


Figura 2. Repositorio en la nube

Fuente [14] Disponible en: https://stackoverflow.com/questions/34844513/why-cant-i-see-commited-files-in-my-shared-repo

2.2.4 Pruebas automatizadas con postman

Postman es una aplicación que nos permite realizar pruebas API. Es un cliente HTTP que nos da la posibilidad de testear 'HTTP requests' a través de una interfaz gráfica de usuario, por medio de la cual obtendremos diferentes tipos de respuesta que posteriormente deberán ser validados [15]

Postman nos ofrece muchos métodos para interactuar con los 'endpoints [15]'.

Los más utilizados y sus funciones son:

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

2.2.5 Códigos de respuestas

Al testear APIs con Postman, normalmente obtendremos diferentes códigos de respuesta. Los más comunes incluyen:

- ✓ Serie 100 > Respuestas Temporales, por ejemplo: '102 Processing'.
- ✓ Serie 200 > Respuestas donde el cliente acepta el request, siendo procesado exitosamente en el server, por ejemplo: '200 Ok'.
- ✓ Serie 300 > Respuestas relacionadas a redireccionamiento URL, por ejemplo: '301 Moved Permanently'.
- ✓ Serie 400 > Respuestas de error del lado del cliente, por ejemplo: '400 Bad Request'.
- ✓ Serie 500 > Respuestas de error del lado del server, por ejemplo: '500 Internal Server Error. [16]

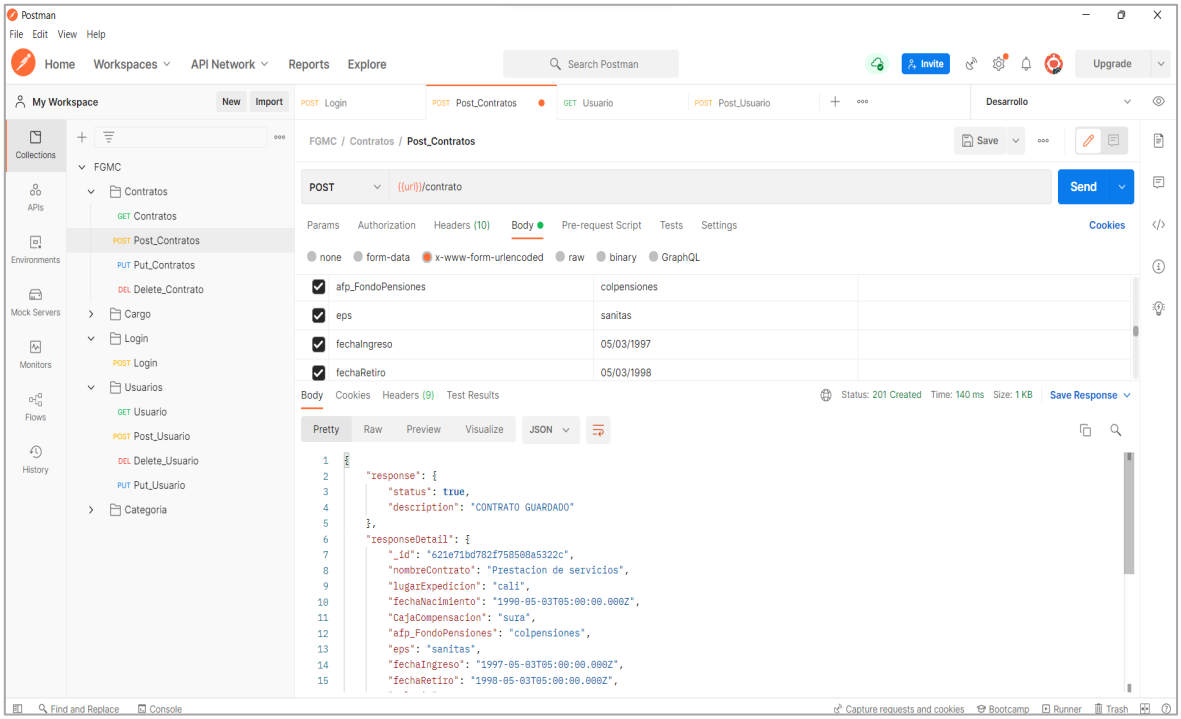
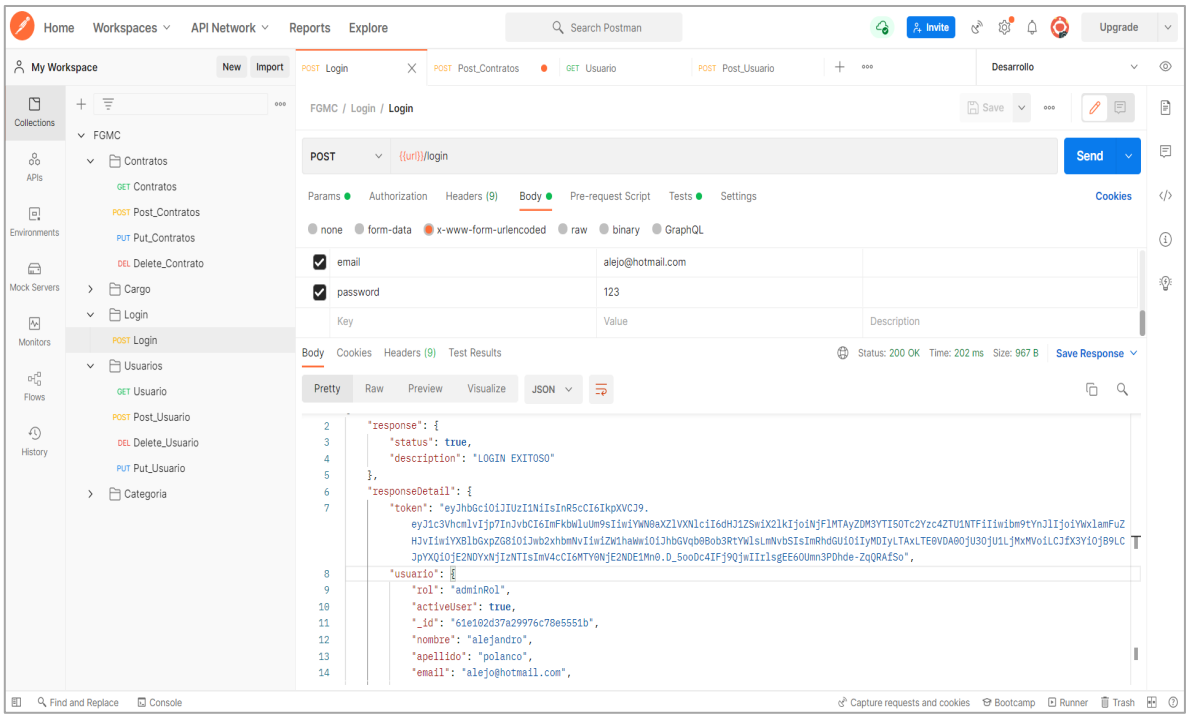
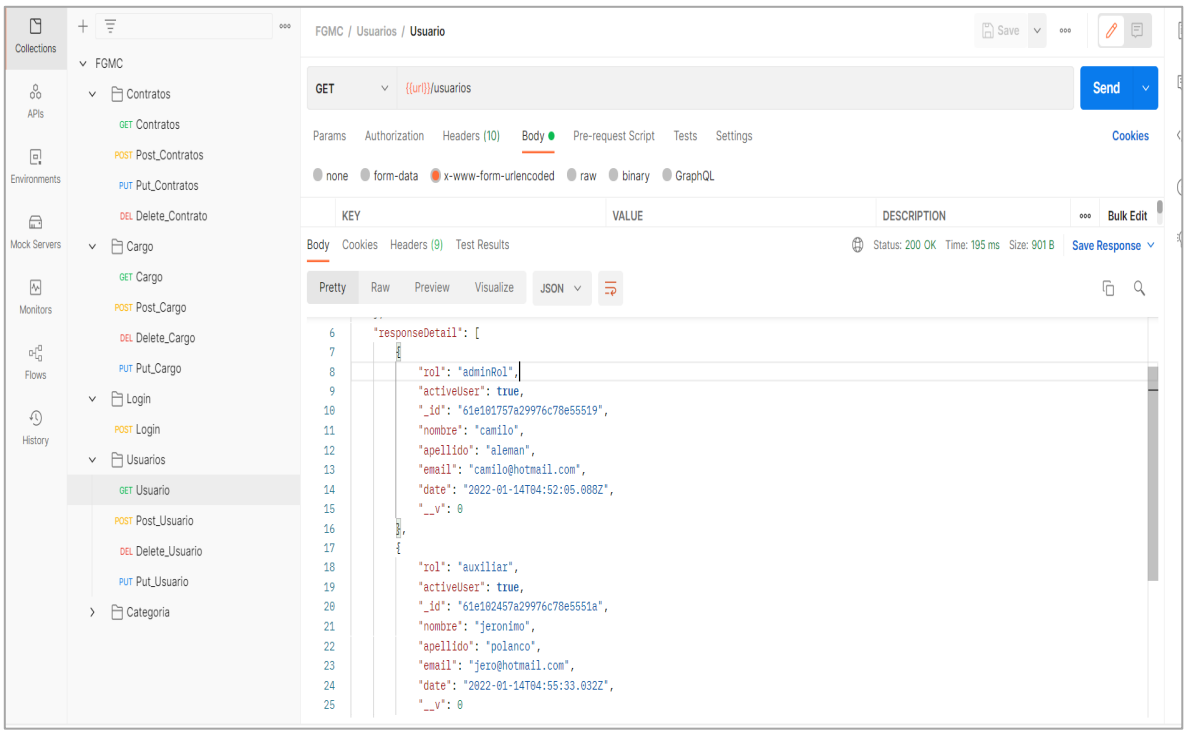


Figura 3. Pantallazos códigos de respuesta.

Fuente: Elaboración propia presente estudio. Año 2022.



Fuente: Elaboración propia presente estudio. Año 2022.



Fuente: Elaboración propia presente estudio. Año 2022.

2.2.6 Código limpio (Clean code)

El código limpio es aquel código que está estructurado de forma compresible, que es claro en sus intenciones, fácil de leer, que es fácilmente mantenible y que está testeado**.** En el libro se van dando algunas ideas para conseguir escribir código limpio, hablando de principios SOLID, de la importancia de dar nombres a variables y clases [17]

2.2.7 Arquitectura limpia

La arquitectura de un proyecto de desarrollo de software es una parte fundamental para garantizar la mantenibilidad del producto, una de las métricas de

calidad definidas como estándar en la norma ISO 9126 [18]: La mantenibilidad cuenta con cuatro características:

- Estabilidad: fortaleza con la que el software responde ante situaciones inesperadas
- Facilidad de análisis: rapidez del software para diagnosticar y mostrar resultados de fallos o circunstancias excepcionales.
- Facilidad de cambio: versatilidad con la que el software puede ser actualizado o modificado.
- Facilidad de pruebas: simplicidad con la que se pueden verificar las diferentes funcionalidades del software.

En el desarrollo del proyecto, se aplicaron además las siguientes características para dar mayor robustes al desarrollo software.

Independiente de los frameworks: Los frameworks deberían ser herramientas, y no obligarnos a actuar de una determinada manera debido a sus restricciones.

Testeable: Debemos poder probar nuestras reglas de negocio sin pensar en base de datos, interface gráfica u otros componentes no esenciales de nuestro sistema.

Independiente de la UI: Si la UI cambia a menudo esto no puede afectar al resto de nuestro sistema, que tiene que ser independiente.

Independiente de la base de datos: Deberíamos poder cambiar de Oracle, a SQL Server, a MongoDB, a Casandra o a cualquier otra base de datos sin que afectara demasiado a nuestro sistema.

Independiente de cualquier entidad externa: No deberíamos saber nada de entidades externas, por lo que no deberemos depender de ellas. [19]

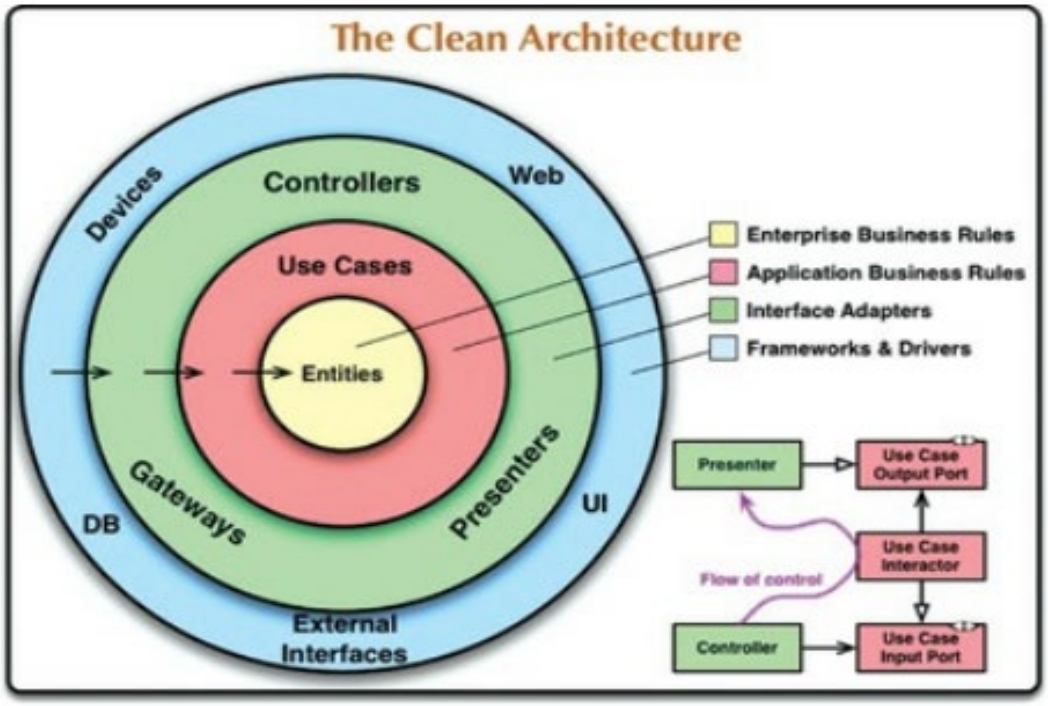


Figura 4. Arquitectura Limpia

Fuente: [20]. Disponible de: https://www.genbeta.com/desarrollo/principios-de-una-arquitectura-limpia-mantenible-y-testeable

2.2.8 Pruebas Software

Las pruebas de software son una parte integral del ciclo de vida del desarrollo de software (SDLC). Las pruebas son la forma en que puede estar seguro acerca de la funcionalidad, el rendimiento y la experiencia del usuario. Ya sea que realice sus pruebas manualmente o a través de la automatización, cuanto antes y más a menudo pueda llevar a cabo pruebas, más probable es que identifique errores y errores, no sólo ahorrándole a usted y a su equipo de posibles simulacros de incendio más adelante, sino también asegurándose de que su aplicación de software haya sido revisada y auditada a fondo antes de que esté frente a sus usuarios. Si los problemas se arrastran al entorno de producción, los más caros y lentos que van a solucionar [21]

Las pruebas de software se pueden dividir en dos tipos diferentes: pruebas funcionales y no funcionales. Diferentes aspectos de una aplicación de software requieren diferentes tipos de pruebas, como pruebas de rendimiento, pruebas de escalabilidad, pruebas de integración, pruebas unitarias y muchos más. Cada uno de estos tipos de pruebas de software ofrece una excelente visibilidad de la aplicación, desde el código hasta la experiencia del usuario. Vamos a entrar en los detalles de algunos de los tipos más comunes de pruebas de software [21].

2.2.8.1. Tipos de pruebas de software: pruebas funcionales y no funcionales

Pruebas funcionales

Las pruebas funcionales se llevan a cabo para comprobar las características críticas para el negocio, la funcionalidad y la usabilidad. Las pruebas funcionales garantizan que las características y funcionalidades del software se comportan según lo esperado sin ningún problema. Valida principalmente toda la aplicación con respecto a las especificaciones mencionadas en el documento Software Requirement Specification (SRS) [21].

Los tipos de pruebas funcionales incluyen pruebas unitarias, pruebas de interfaz, pruebas de regresión, además de muchas, por otro lado, tienen varias ventajas tales como:

- Se asegura de que el sitio web / aplicación está libre de defectos.
- Garantiza el comportamiento esperado de todas las funcionalidades.
- Garantiza que la arquitectura sea correcta con la seguridad necesaria.
- Mejora la calidad y las funcionalidades generales.
- Minimiza los riesgos empresariales asociados con el sitio web/aplicación.

Pruebas unitarias

Las pruebas unitarias se basan en lograr probar unidades, fragmentos, piezas individuales de una aplicación de software al principio del ciclo de vida de un sistema en sus siglas específicas “SDLC”. Cualquier función, procedimiento, método o módulo puede ser una unidad que se someta a pruebas unitarias para determinar su corrección y comportamiento esperado. Las pruebas unitarias son las primeras pruebas que los desarrolladores realizan durante la fase de desarrollo, cabe destacar que cualquier SDLC debe resultar en un sistema de alta calidad que cumple o excede las expectativas del cliente, llega a término en el tiempo y estimaciones de costos, sea barato de mantener y rentable, mostrando características importantes como las que se mencionan a continuación [21].:

- Detección temprana de errores en las nuevas funcionalidades o características desarrolladas.
- Minimiza los costos de las pruebas a medida que se detectan problemas desde el principio.
- Mejora la calidad del código con una mejor refactorización del código.
- Apoya el proceso de desarrollo ágil.
- Simplifica la integración y permite una buena documentación.

Pruebas de integración.

Las pruebas de integración implican probar diferentes módulos de una aplicación de software como grupo. Una aplicación de software se compone de diferentes submódulos que trabajan juntos para diferentes funcionalidades. El propósito de las pruebas de integración es validar la integración de diferentes módulos juntos e identificar los errores y problemas relacionados con ellos. Poseen ventajas as cuales se mencionarán a continuación:

- Se asegura de que todos los módulos de aplicación estén bien integrados y funcionen juntos según lo esperado.
- Detecta problemas y conflictos interconectados para resolverlos antes de crear un gran problema.
- Valida la funcionalidad, fiabilidad y estabilidad entre diferentes módulos.
- Detecta excepciones ignoradas para mejorar la calidad del código.
- Admite la canalización de CI/CD.

Pruebas no funcionales.

Las pruebas no funcionales son como pruebas funcionales; sin embargo, la principal diferencia es que esas funciones se prueban bajo carga para el rendimiento de los observadores, fiabilidad, usabilidad, escalabilidad, etc. Las pruebas no funcionales, como las pruebas de carga y esfuerzo, normalmente se llevan a cabo mediante herramientas y soluciones de automatización, como LoadView. Además de las pruebas de rendimiento, los tipos de pruebas no funcionales incluyen pruebas de instalación, pruebas de confiabilidad y pruebas de seguridad.

Performance Testing

Las pruebas de rendimiento son un tipo de pruebas no funcionales, realizadas para determinar la velocidad, estabilidad y escalabilidad de una aplicación de software. Como su nombre indica, el objetivo general de esta prueba es comprobar el rendimiento de una aplicación con respecto a los diferentes puntos de referencia del sistema y de la red, como la utilización de la CPU, la velocidad de carga de la página, el control de tráfico máximo, la utilización de recursos del servidor, etc. Dentro de las pruebas de rendimiento, hay varios otros tipos de pruebas, como pruebas de carga y pruebas de esfuerzo.
Tienen varias ventajas como:

- ¬ Evalúa la velocidad y escalabilidad del sitio web/aplicación
- ¬ Identifica los cuellos de botella o enfrascamientos para las mejoras de rendimiento.
- ¬ Detecta errores que se pasan por alto en las pruebas funcionales.
- ¬ Optimización del sistema y mejoras de características.
- ¬ Garantiza la fiabilidad del sitio web bajo una gran carga.

2.2.9 Tipo de arquitectura software

2.2.9.1 Patrón de arquitectura micro servicios

Cuando se escribe la solicitud como un conjunto de microservicios, en realidad se están escribiendo múltiples solicitudes que funcionarán juntas. Cada micro servicio tiene su propia responsabilidad y los equipos pueden desarrollarlos independientemente de otros microservicios. La única dependencia entre ellos es la comunicación. A medida que los microservicios se comunican entre sí, se tendrá que asegurar que los mensajes enviados entre ellos sean compatibles con los anteriores [22].

Existen muchas ventajas, las cuales se logran evidenciar dentro del código fuente realizado, para así poder dar alcance y gestionar el aplicativo en su totalidad, algunas de estas ventajas se nombran a continuación:

- ¬ Escribir, mantener y desplegar cada microservicio por separado
- ¬ Escalable
- ¬ Facilidad de reescribir las piezas de la aplicación porque son más pequeñas y menos acopladas a otras partes
- ¬ Los nuevos miembros del equipo deben ser rápidamente productivos
- ¬ La aplicación debe ser fácil de entender y modificar

- Altamente mantenible y comprobable ya que permite por sus distintas características tener un desarrollo, despliegue rápido y frecuente.
- Desplegable de forma independiente, permite a un equipo desplegar su servicio sin tener que coordinar con otros teamworks [22]

2.3 Librerías de desarrollo React

Dentro del desarrollo del aplicativo web, se emplearon tecnologías vanguardistas las cuales están revolucionando el desarrollo web a nivel mundial, una de estas tecnologías se llama Axios, el cual es un cliente http ligero el cual se basa en el servicio http angulas.so y posee una extrema similitud a la api fetch nativa de JavaScript, Axios es un cliente que se basa en promesas (Promises), permitiendo aprovechar las funciones denominadas async y await de JavaScript para así permitir obtener un código asíncrono mucho más entero y legible [23]

En el proyecto desarrollado, todas las peticiones de parte del servidor se implementaron con la librería anteriormente mencionada, dado que es muy efectiva a la hora de realizar peticiones a gran escala, estas labores se ven ejecutadas en solo milisegundos debido a su gran robustes a la hora de su implementación con JavaScript, permitiendo así tener un entorno altamente competitivo y escalable.

CAPÍTULO III: METODOLOGÍA

El marco de trabajo bajo el cual se implementó este proyecto fue la metodología de trabo ágil SCRUM [24]

Institucionalmente, se tuvo un Scrum Master el cual fue el responsable de que las técnicas Scrum sean comprendidas y aplicadas a cabalidad, trabajando conjuntamente con el product owner y finalmente el equipo de desarrollo (Scrum Team) tarea la cual fue exclusivamente de mi responsabilidad.

Se implementó el uso de los sprint cada veinte (20) días el cual tuvo cuatro (4) aspectos importantes que permitieron lograr con éxito cada sprint planteado, siendo estos: sprint planning donde se pactaba que era lo que se deseaba hacer dentro del desarrollo y más importante aún, se definía la manera por la cual se lograría realizar el objetivo principal de este sprint, que se iba a implementar para realizar este objetivo era la tarea más importante en este paso, se implementaros a lo largo del proyecto daily meetings de no más de quince (15) minutos para así ir mirando el paso a paso del proyecto, la manera como se deseaba viera el desarrollo backend, frontend, Bases de datos y así se logró tomar las mejores decisiones respecto al mismo, siguiente, se realizaron sprint reviews donde se mostraba el funcionamiento del software hasta el momento al cliente, para que así los cambios que se pactaban dentro de este ítem fueran paulatinos y se pudiera ir atenuando de manera cómoda el desarrollo de los solicitado y finalmente se realizó el sprint retrospective donde se hace una evaluación de cómo se implementó la metodología scrum en el último sprint [24] .

Dicho lo anterior las fases en las que se desarrolló el trabajo fueron las siguientes:

3.1. Fase 1: Análisis de las distintas características y componentes relacionados con la gestión documental y contractual de la institución.

En esta fase, se buscó y realizó todo el levantamiento de los requerimientos del cliente con el objetivo de conocer y dimensionar el alcance del proyecto y se determinaron las tecnologías que se creyó y pactó eran mejores a implementar para el desarrollo de la aplicación. Esta preparación inicial permitió observar el camino para pactar las actividades a seguir en el desarrollo de la fase de análisis los cuales fueron las siguientes actividades:

1. Analizar el funcionamiento, estado actual y necesidades de sistematización de los procesos de gestión documental y contratación de la Fundación Gimnasio Moderno del Cauca.
2. Establecer los requisitos funcionales y no funcionales, requeridos para el desarrollo de la aplicación Web.
3. Determinar las herramientas tecnológicas a utilizar para el desarrollo del Backend, Frontend y la base de datos.

3.2. Fase 2: Diseño

En esta fase se diseñó la base de datos, las interfaces gráficas y se pudo definir la arquitectura de la aplicación. Para ello, se realizarán las siguientes actividades:

1. Diseñar el modelo de datos de acuerdo con los requerimientos establecidos para el proyecto.
2. Definir la arquitectura base de la aplicación de acuerdo con los requerimientos.
3. Diseñar las interfaces gráficas y el modelo de navegación.

3.3. Fase 3: Codificación y descripción de las tareas de programación e implementación de tecnologías, librerías y arquitecturas.

En esta fase se desarrolló el código de la aplicación correspondientes a la base de datos, el Backend y el Frontend, los cuales se implementaron utilizando las tecnologías determinadas en la fase de análisis, se realizaron las siguientes actividades:

1. Desarrollo de la funcionalidad de la base de datos de acuerdo con los requerimientos funcionales establecidos en la fase de análisis.
2. Se codificó el Backend de acuerdo con los requisitos funcionales y no funcionales establecidos en la fase de análisis.
3. Se unió el Frontend con el backend de la aplicación de acuerdo con los requisitos funcionales y no funcionales previamente obtenidos.
4. Describir los componentes de arquitectura seleccionados.
5. Se explicaron y realizaron las diversas pruebas unitarias del Backend.

3.4. Fase 4: Validación

En esta fase se realizó las pruebas de verificación, para confirmar que todas las etapas del desarrollo se ejecutaron con calidad y conforme a lo establecido en los requerimientos funcionales del proyecto, realizando las siguientes actividades:

1. Diseñar los casos de prueba específicos para la Base de datos, Backend y Frontend desarrollados.
2. Ejecutaron los casos de prueba diseñados para validar el funcionamiento de la aplicación.
3. Realizar ajustes necesarios según los datos obtenidos en el paso anterior.
4. Documentar el proceso llevado a cabo en cada actividad de caso de prueba.

Fase 5: Implantación

En esta fase se utilizó un acercamiento al software, buscando mirar la experiencia y funcionamiento de un módulo del proyecto, así que se pensó denominar piloto esta prueba, que implica la instalación del software en el servidor y la puesta en marcha del desarrollo con un grupo selecto y representativo de usuarios, a fin de comenzar a consolidar la operatividad. Una vez instalada y puesta en marcha la aplicación, se procedió a capacitar a los usuarios que utilizarán el desarrollo dentro de la institución. En esta etapa se realizó lo siguiente:

1. Instalación del software en el servidor.
2. Capacitar el personal en el manejo de la aplicación desarrollada.

CAPÍTULO IV: DEFINICIÓN DE REQUERIMIENTOS

4.1 Características del negocio

Según la información adquirida y analizada en entorno al marco institucional de contratación y gestión documental, basándose en las necesidades del cliente (Gimnasio moderno del cauca) para el que se desarrolló este proyecto, se identificaron las siguientes características [1]:

4.1.1 Características asociadas a la contratación.

Consulta de tipos de contratos: La institución posee diferentes tipos de contratos los cuales ejercen en sus labores cotidianas, estos contratos varían uno del otro en diferentes aspectos reglamentariamente hablando y tienen connotaciones especificas cada uno de ellos que hace que los mismos sean trabajados e implementados de maneras puntuales dado su contenido, los contratos que institucionalmente se manejan son:

1. Contrato de prestación de servicios
2. Contrato a término fijo
3. Contrato de prestación de servicios inferior a un año
4. Contrato individual de trabajo a término fijo inferior a un año para docentes privados
5. Contrato de trabajo por duración de una obra o labor contratada
6. Contrato individual de trabajo a término fijo inferior a un año
7. Contrato individual de trabajo a término fijo a un año.
8. Contratos de tipificación especial.

Dicho esto, existe gente que puede tener múltiples contratos, por lo cual debe ser capaz el aplicativo de consultar los contratos asignados a las personas a buscar sea

por número de contrato, cedula de ciudadanía o tipo de contrato y acceder a ellos cuando se necesite.

Bases de datos relacionadas y no relacionadas: Una manera para facilitar el trabajo de los encargados del área contractual de la institución, es alinear todas sus bases de datos descentralizadas a una base de datos en la nube empleando mongoDB Atlas, mostrando información en tiempo real y a la mano a cualquier hora del día, evitando así obtener retazos en producción de contratos debido a la búsqueda y posterior lectura de hojas de Excel como venía pasando anteriormente.

Seguimiento de contratos vigentes: Es necesario recoger y enviar datos informativos de manera periódica con el objetivo de utilizarlos en otras plataformas de la empresa para el conocimiento de sus clientes, como entidades a las cuales la institución debe mostrar información veraz de la cantidad de contratos anuales o semestrales, su vigencia y otros factores contractuales importantes que les permitan llevar a cabo el proceso de control de la parte contractual institucional de una manera sencilla, visual y verídica, para así poder mostrar a los entes de control sus estadísticas contractuales.

Impresión de contratos: Es de vital importancia para la institución Gimnasio Moderno del Cauca, que el aplicativo pueda generar impresión de sus pdf almacenados en la base de datos, buscando así poder legalizar los contratos realizados por medio del aplicativo.

Por otra parte, esto permitirá que en cualquier momento que se desee realizar algún tipo de informe escrito, ellos puedan realizar las búsquedas y filtrar los contratos que deseen para su posterior impresión.

4.1.2. Características asociadas a la gestión documental

Plan institucional de archivos: Es de vital importancia la Identificación y definición de los aspectos críticos y riesgos empresariales, para así lograr por medio del aplicativo priorizar los aspectos críticos frente a los ejes articuladores.

Formulación de la visión estratégica y formulación de objetivos. Formulación de Planes, Programas y Proyectos es lo que busca desarrollarse dentro del marco de plan institucional de archivos, se llevó a cabo los siguientes ítems dentro de este módulo.

- Los propósitos de la planificación.
- El estado actual de la gestión documental en la empresa.
- Los puntos de mejora y su priorización.
- Las estrategias de gestión a implementar.
- El mapa de ruta para la ejecución de las estrategias.
- Las herramientas de seguimiento.

Estado de los archivos institucionales: Los documentos son la columna y el cimiento de toda empresa, puesto que en ellos quedan registradas las actividades que se ejecutan a diario en cada dependencia y área. Desde estados de cuenta y contratos, hasta manuales, correspondencia y actas de presidencia quedan almacenadas, engrosando el archivo interno a lo largo de los años.

Batería de indicadores: Es de vital importancia para la institución educativa Gimnasio moderno del Cauca, las empresas que sub contratan con la institución, los padres de familia de los estudiantes que ejercen año escolar dentro de la mencionada anteriormente y sus clientes conocer el estado de los procesos más importantes que la institución lleva a cabo, con sistemas de medidas que sean de fáciles de entender e interpretar, que puedan ser utilizados de manera transversal en cualquier proyecto.

Acciones correctivas: Es de bastante prioridad para la institución gimnasio moderno del Cauca, tener conocimiento pleno de sus procesos, requerimientos y estrategias misionales que se encuentran presentando errores sea de tipo operativo

o de ejecución, siendo así requiere un módulo de acciones correctivas en las cuales ellos puedan digitar cuales son las acciones que requieren un estudio más a fondo para ver cuáles son las soluciones a implementar, teniendo en cuenta que la institución ha alcanzado la certificación internacional de calidad FMQ
Algunos de los criterios que puede evaluar para saber si está ante un problema sistémico que justifica una acción correctiva son los siguientes:

- KPI de rendimiento: cuando los indicadores muestran problemas rutinarios. Si los indicadores muestran que los procesos funcionan de forma correcta, no es preciso emprender acciones correctivas.
- Registros: los registros pueden evidenciar problemas que se presentan de forma recurrente y que deben ser investigados.
- Informes de los empleados: informes, quejas, comentarios de los empleados sobre problemas repetitivos son un signo claro de que es preciso implementar acciones correctivas.
- Resultados de auditorías o inspecciones al sistema: cuando los auditores internos o externos incluyen en sus informes las recomendaciones para la implementación de acciones correctivas en ISO 9001.

A continuación se presenta las tablas 1 y 2. Requerimientos funcionales y no funcionales:

4.2. Requerimientos

4.2.1. Requerimientos funcionales

Tabla 1. Requerimientos funcionales

Código	Nombre del requerimiento	Descripción
RF-01	Login personalizado	Implementar un acceso login el cual permita a los trabajadores de la institución acceder al software en todos sus módulos, teniendo en cuenta las restricciones según el cargo al que pertenezca y teniendo sus respectivas animaciones, iconos y comportamientos.
RF-02	Login verificación de campos requeridos	Los campos que se ven en el login (Email y Password) son requeridos para iniciar sesión, por lo tanto el software debe verificar si no hay data o la misma esta incorrecta y ser capaz de denegar acceso
RF-03	Modal de error	El sistema debe mostrar un modal el cual capture el error obtenido en el backend y se lo muestre al usuario Ejemplo: Acceso denegado Con el fin de que el usuario sepa que ha realizado mal y pueda corregir su error, esto brinda buena experiencia de usuario
RF-04	Crear contrato	El sistema permitirá crear un contrato siempre y cuando todos los campos del formulario se encuentren completos

RF-03	Guardar contratos creados	Obtener los contratos realizados por medio del aplicativo y guardarlos en la base de datos de la aplicación.
RF-04	Módulo de contratación con filtro de búsqueda	Dentro del módulo de contratación, tener filtro de búsqueda de contratos por: ¬ Año ¬ Nombre ¬ Fecha creación ¬ Tipo de contrato
RF-04	Estadísticas de Inter operatividad de los contratos realizados	Implementar una vista que permita observar el estado de los contratos en el tiempo, efectuando filtros por tipo de contrato, cedula de ciudadanía, municipio.
RF-05	Notificar el estado de la aplicación cuando se pone lenta	Mostrar un icono traslucido de carga en todas las pantallas de la aplicación que indique si la misma se encuentra en proceso de carga por un error interno falta de conexión.
RF-06	Dashboard o panel de control para cada usuario	Implementar una vista dentro del aplicativo que permita evidenciar los módulos dependiendo el tipo de login realizado: ● Login administrador ● Login auxiliar ● Login talento humano
RF-07	Pantalla de navegación de cada módulo.	Refactorizar la lógica de la pantalla del desarrollo web para así tener un mejor manejo y funcionalidad de cada uno de los módulos realizados.

RF-08	Archivos de descarga-Modulo de gestión documental	Un módulo especial de archivos institucionales para descarga al publico
RF-09	Botón de carga restrictivo	Solo permite cargar formatos (jpeg. Png, xmls, Word, pdf)
RF-10	Archivos solo vista-Modulo gestión documental	Crear sección donde se alojen documentos institucionales de tipo vista sin edición, los cuales hacen parte de la misión y visión de la entidad
RF-11	Indicadores en gráficos	En el módulo denominado batería de indicadores, implementar gráficos estadísticos para su posterior análisis
RF-12	Cargar acciones correctivas institucionales	Formulario con: Selección de responsable encargado de seguir la acción, nivel de prioridad de la acción, descripción del problema, descripción acción de mejora, fecha de solución y botón de crear la acción correctiva
RF-13	Mensaje de bienvenida personalizado dentro del dashboard	En el dashboard personal, mostrar mensaje de bienvenida autenticando quien inicio la sesión
RF-14	Módulos con animación y pointer	Todos los módulos que se encuentran dentro del dashboard deben tener animación y pointer denotando que el mouse se encuentra apuntando en el acceso de un modulo
RF-15	Aplicación totalmente responsive	Desarrollo de aplicativo 100% responsive, si se abre en Tablet, celular o portátil debe verse ajustado al tamaño de la pantalla del dispositivo
RF-16	Logout de aplicativo	Cierre de sesión de módulos o aplicativo, retornando a pantalla principal (Login)

Fuente Elaboración propia presente estudio. Año 2021

4.2.1. Requerimientos no funcionales

Tabla 2. Requerimientos no funcionales

Código	Nombre del requerimiento	Descripción
RNF-01	Versionamiento de código con GitHub	Manejar estrategias de versionamiento de código con GitHub para los diferentes repositorios del proyecto.
RNF-02	Automatización de tareas de integración y entrega continua	Utilizar la herramienta Jenkins para automatizar tareas de verificación de funcionamiento y calidad del proyecto.
RNF-03	TypeScript y JavaScript como lenguaje de programación	Utilizar JavaScript como lenguaje de programación para el proyecto.
RNF-04	Buenas prácticas de programación	Implementar buenas prácticas y principios de programación orientada a objetos para garantizar la calidad del código.
RNF-05	Arquitectura y patrón de arquitectura	Investigar, seleccionar e implementar una arquitectura y patrones arquitectura adecuados para el proyecto y para garantizar su calidad.
RNF-06	Librerías recomendadas y soportadas	Investigar, seleccionar y utilizar librerías robustas que sean de calidad y que reciban soporte continuo para sus futuras actualizaciones.
RNF-07	Pruebas unitarias	Implementar pruebas unitarias de cada funcionalidad para garantizar su correcto funcionamiento y comportamientos esperados.
RNF-08	Reportes de cobertura y calidad	Implementar reportes de cobertura de clases y funciones por pruebas unitarias y verificación de calidad de código.

Fuente Elaboración propia presente estudio. Año 2021

CAPÍTULO V: INGENIERIA DEL PROYECTO

5.1 Trabajo realizado

A continuación, se presenta un diagrama guía que resume el trabajo realizado en esta pasantía y que será explicado durante el transcurso de este capítulo:

Figura 5. Diagrama guía resumen de trabajo de pasantía.

Fuente Elaboración propia presente estudio. Año 20212

5.2 Investigación de librerías, frameworks y herramientas

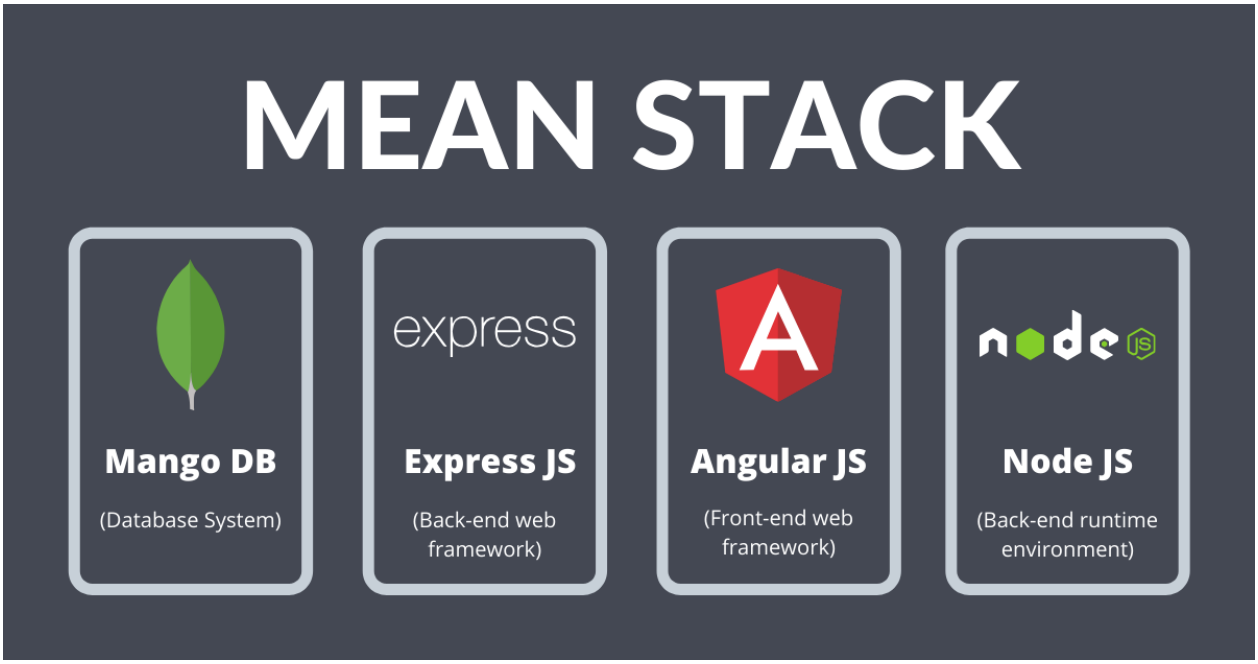


Figura 6 Investigación de librerías, frameworks y herramientas

Fuente: https://www.thirdrocktechkno.com/blog/what-is-mean-stack-mean-stack-components-and-benefits/

Descripción del trabajo: Dentro de esta tarea, se fundamentó en la investigación de diferentes librerías útiles para el desarrollo de aplicaciones de tipo web y pertinentes para el proyecto. Además, con los resultados se realizó una prueba de concepto, junto con las arquitecturas recomendadas, para evaluar su uso de manera más profunda y presentar los resultados obtenidos en un informe.

A continuación, se muestra las librerías usadas para el desarrollo backend, que permitieron el correcto funcionamiento del mismo.

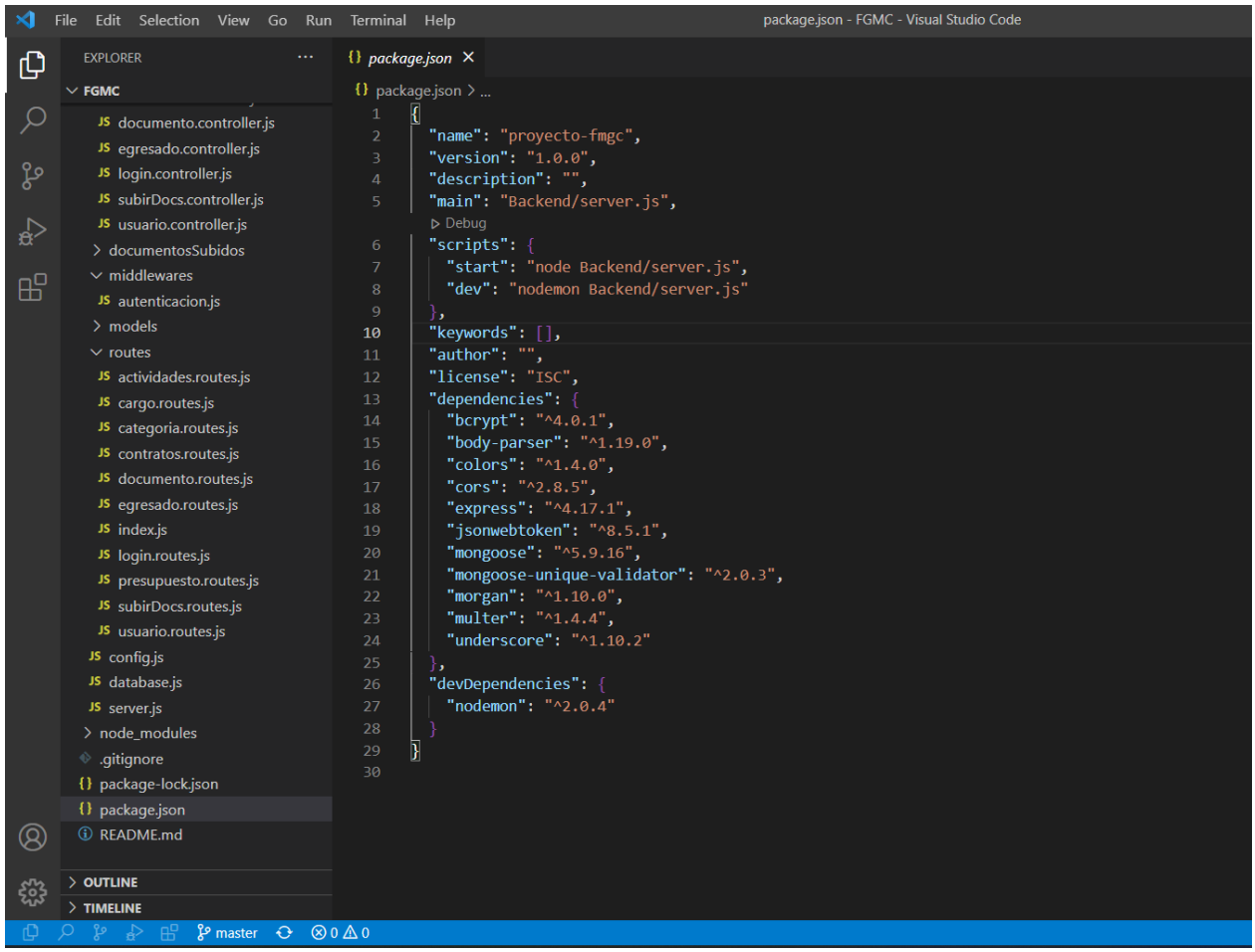


Figura 7 Librerías usadas para el desarrollo backend.

Fuente Elaboración propia presente estudio. Año 2021

Descripción de las librerías: En la evolución del desarrollo, se hizo especial énfasis de investigar las librerías más populares usadas por los desarrolladores de aplicaciones Android y las recomendadas como componentes de arquitectura, las cuales se clasificaron en las siguientes categorías: peticiones HTTP, seguridad en la información, manejo de base de datos, descarga y manejo de estadística gráfica , análisis y manejo de información JSON, pruebas unitarias, de instrumentación y otras herramientas y funcionalidades.

Haciendo especial énfasis para el desarrollo del backend, se emplearon varias tecnologías mundialmente muy mencionadas en la actualidad, con el fin de desarrollar una aplicación vanguardista y funcional.

5.3. Bases de datos.

En el proceso de selección del framework, se buscó que por medio de sus características intrínsecas, este lograra permitir controlar de manera idónea la base de datos seleccionada la cual fue MongoDB Atlas, incluyendo consultas de sus tablas, transacciones, modelamiento. Por otra parte, se debe tener en cuenta que existen variantes en los frameworks más empleados para dicho proceso, por lo mismo se busca que el framework pueda soportar la integración entre las distintas librerías y sus respectivas clases.

Los motivos más importantes para utilizar estos lenguajes de programación con MongoDB son:

- Son lenguajes basados en JavaScript y se complementan muy bien.
- Existen multitud de herramientas de terceros, librerías y lookups.
- Existe bastante documentación y ejemplos de uso de estos lenguajes con MongoDB.
- Profesionalmente, son de los lenguajes más demandados para usar con MongoDB.

MongoDB Atlas: Es un servicio global de base de datos de tipo cloud (Alogamiento en la nube) para las aplicaciones modernas. Implementar una base de datos en la nube empleando MongoDB hace que la BD sea completamente administrada y garantiza la disponibilidad, la escalabilidad, además de brindar gran conformidad

con las normas de seguridad gracias a la automatización inteligente para mantener el funcionamiento y el rendimiento a escala conforme las aplicaciones evolucionan. Por otra parte, amplía sus datos para gestionar cualquier carga de trabajo que use la plataforma de datos de aplicaciones de MongoDB, como la búsqueda de texto completo y el análisis en tiempo real. [25]

Características**:**

- Automatización: la manera más fácil de crear, lanzar y escalar aplicaciones en MongoDB.
- Flexibilidad: La única base de datos como servicio con tofo lo necesario para las aplicaciones modernas.
- Seguridad: disponible varios niveles de seguridad.
- Escalabilidad: gran escalabilidad sin interrumpir la actividad.
- Alta disponibilidad: implementaciones con tolerancia a errores y autoreparación predeterminadas.
- Alto rendimiento**:** el rendimiento necesario para las cargas de trabajo más exigentes.

Ventajas que ofrece trabajar con MongoDB Atlas:

- Ejecución
- Puesta en marcha de un clúster en segundos.
- Implementaciones replicadas y sin interrupción.
- Total escalabilidad: escalado horizontal o vertical en tan solo unos clics y sin interrumpir la actividad.
- Revisiones automáticas y actualizaciones simplificadas para las nuevas funciones de MongoDB.
- Protección y seguridad
- Autenticación y cifrado.

- Copias de seguridad continúas con recuperación en un punto en el tiempo.
- Supervisión detallada y alertas personalizadas.
- Libertad de movimiento
- Modelo de planes de precio según demanda: se factura por hora.
- Compatible con diferentes tipos de servicios de nube *(AWS, GCP, Azure)*.
- Parte de un paquete de productos y servicios para todas las fases de la aplicación.
- Ideal para entornos con pocos recursos de computación
- Tiene coste nulo o en su caso de la versión pro es muy bajo
- Tiene gran documentación
- Es un complejo perfecto para JavaScript.

Desventajas:

- Es una tecnología joven
- No posee joins para realizar consultas
- Falta de estandarización entre las diversas bases de datos no SQL

Express Js: Es una herramienta con un amplio número de dependencias y sus librerias propias la cual permite que se creen aplicaciones de manera rápida y sencilla**.** Esto es de vital importancia ya que permite a aquellas personas que no tienen un gran conocimiento sobre programación y desarrollo de apps, dado que se puede realizar paso a paso y con ayuda de una comunidad de expertos que trae express de manera nativa.

Más específicamente, se trata de un marco o frame de aplicación web para back-end, disponible en la plataforma de JavaScript junto a Node.js. De hecho, actualmente se considera como el frame estándar para esa plataforma.

La gran importancia la cual en los últimos años ha tomado express dentro del mundo del desarrollo web, es que permite que se desarrollen funcionalidades tales como el enrutamiento, que se utiliza para almacenar información sobre redes que se

encuentren conectadas y permiten que el tráfico de información se transmita de una forma sencilla y rápida.

Hay que aclarar que existen formas de mejorar el rendimiento de estas aplicaciones, ya que si bien es cierto su elaboración puede darse en periodos cortos de tiempo, lo ideal es garantizar su funcionalidad. Express.js permite hacer esto con muchas características disponibles, pero dispuestas de forma fácil para aplicar.

Sus principales características son:

- Aplicaciones web: Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles.
- API: Con miles de métodos de programa de utilidad HTTP y middleware a su disposición, la creación de una API sólida es rápida y sencilla.
- Tiene su propio middleware
- Sirve activos estáticos dado que posee su propio app.use
- Rendimiento: Express proporciona una delgada capa de características de aplicación web básicas.

Desventajas:

- Variaciones entre cada actualización
- CPU consume alto recurso si encuentra bucles muy grandes

5.4. Seguridad de acceso

El tema de seguridad en los accesos de una aplicación, cobra mucha importancia en los desarrollos actuales. En el desarrollo web especialmente, se busca que la autenticación de los usuarios sea genuina y que la misma brinde las garantías necesarias al usuario, siendo así el TLS (Transport layer security) o SLS (Secure sockets layer) que es la seguridad de la capa de transporte, esta nos permite cifrar o encriptar la conexión con el servidor de forma que todo el

intercambio de datos vaya tipado y complejo para robar o hackear. Haciendo esta la opción más segura porque protege todo el flujo de datos.

Bcrypt: Es una utilidad de cifrado de archivos multiplataforma. Los archivos cifrados son portátiles en todos los sistemas operativos y procesadores compatibles. Las frases de contraseña deben tener entre 8 y 56 caracteres y se codifican internamente en una clave de 448 bits. Sin embargo, todos los caracteres suministrados son significativos. Cuanto más fuerte sea su frase de contraseña, más segura será su información [26]

Además de encriptar sus datos, bcrypt sobrescribirá por defecto el archivo de entrada original con basura aleatoria tres veces antes de eliminarlo para frustrar los intentos de recuperación de datos por parte de personas que puedan acceder a su computadora.

A continuación, se muestra la implementación de Bcrypt dentro del software, para asi por medio de la librería lograr ocultar todo tipo de información que sea requerida por el usuario, además específicamente como se muestra en la foto a continuación, se implementó para hacer una encriptación dentro del controlador del login, el cual es el único encargado de manejar las contraseñas de los usuarios adscritos a la Institución educativa Fundación Gimnasio del Cauca, ubicada en la ciudad de Popayán.

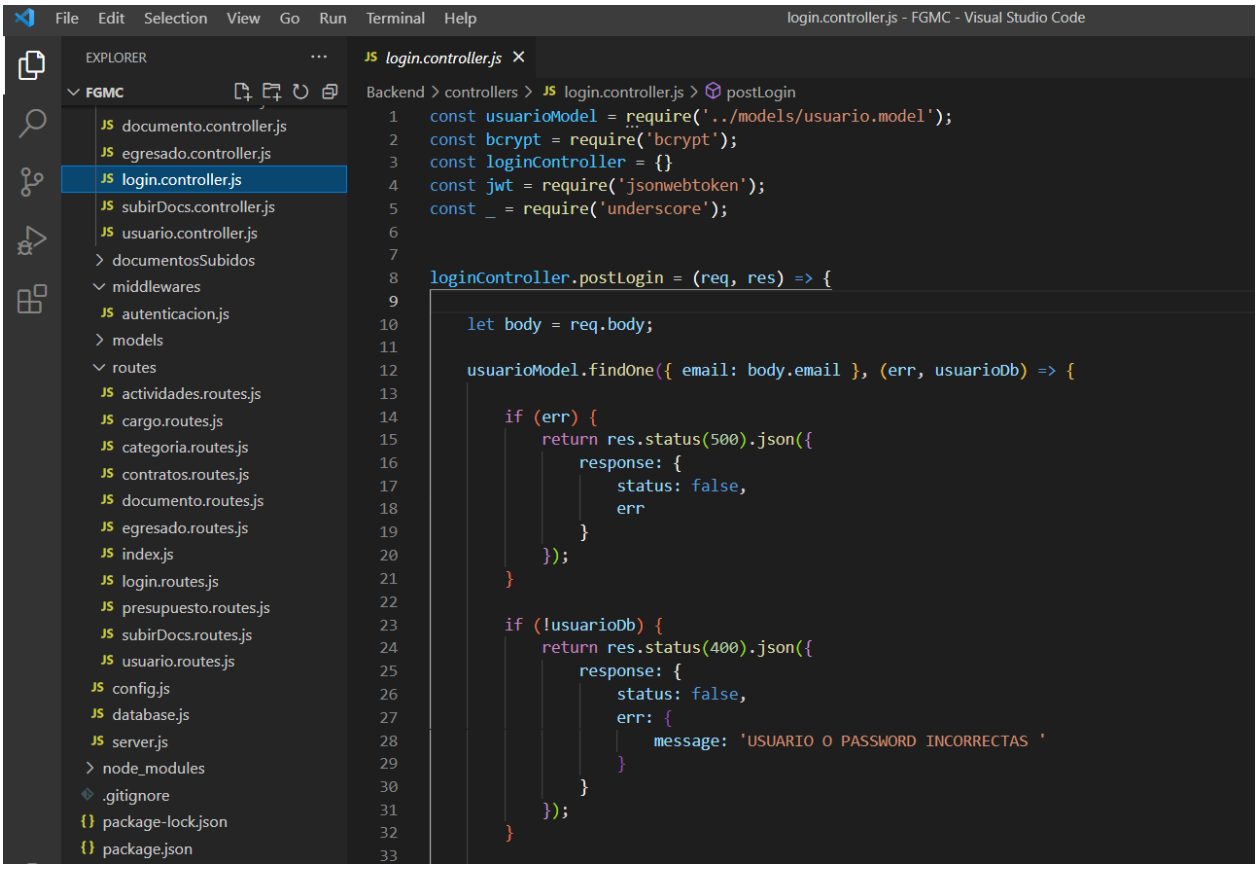


Figura 8. Implementación de Bcrypt

Fuente Elaboración propia presente estudio. Año 2021

Características:

- Función de hash de contraseña específica.
- Eficiencia de 3000 rondas por minuto.
- Protección extrema.

Ventajas:

- Protección contra lookups.
- Refinamiento de intentos de robo con tablas arcoíris.
- Fácil de programar.

Desventajas:

- Utiliza más recursos de máquina.
- Mayor tiempo de procesamiento.

5.4.1. Consultas en MongoDB Atlas

Es importante aclarar que MongoDB es una base de datos NoSQL de código abierto, multiplataforma y orientada a documentos que se utiliza para almacenar datos semi estructurados escritos en C ++.

En este caso, MongoDB en lugar de tablas y filas, almacena datos en pares clave-valor. Para que el aprendizaje sea fácil y sin complicaciones para los desarrolladores y administradores, estos son algunos de los comandos de MongoDB de uso frecuente. Para realizar las consultas avanzadas dentro de Atlas toca realizar la instalación npm de la siguiente librería la cual nos permitirá de manera eficaz realizar todo tipo de consultas en nuestra base de datos determinada para el desarrollo de nuestro sistema.

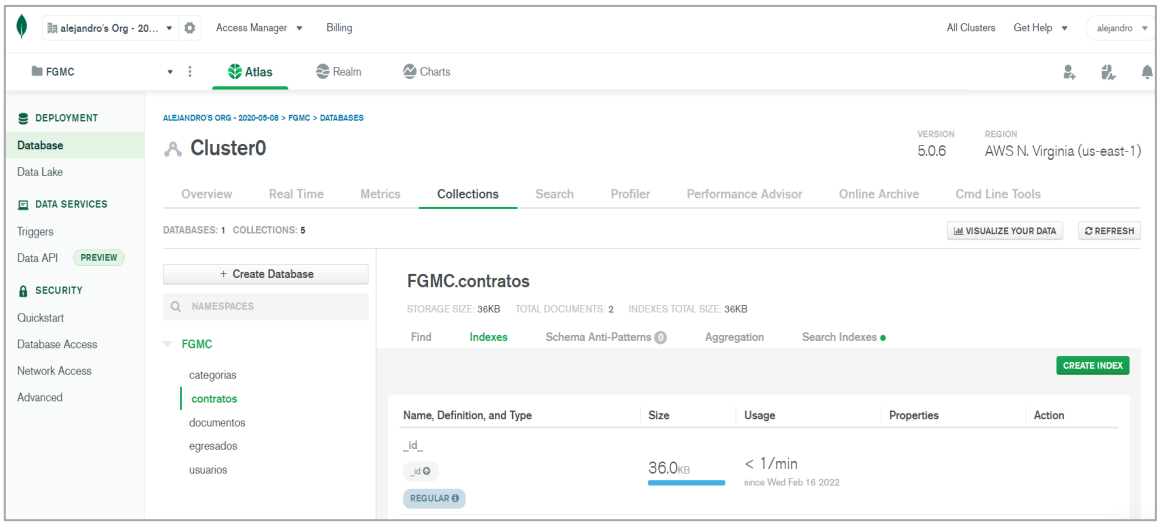


Figura 9. Consultas en nuestra base de datos

Fuente https://geekflare.com/es/mongodb-queries-examples/

Mongoose: Es un object Document Mapper (ODM). Esto significa que Mongoose le permite definir objetos con un esquema fuertemente tipado que se asigna a un documento MongoDB. [27]

Siendo así, proporciona una increíble cantidad de funcionalidades para crear y trabajar con esquemas. Mongoose actualmente contiene ocho SchemaTypes que una propiedad se guarda como cuando se conserva a MongoDB. Son [27]

- ✓ String (Cadena)
- ✓ Number (Número)
- ✓ Date (Fecha)
- ✓ Buffer
- ✓ Boolean (Booleano)
- ✓ Mixed (Mixto)
- ✓ ObjectId
- ✓ Array (Matriz)

Cada tipo de datos le permite especificar:

- ✓ Un valor predeterminado
- ✓ Una función de validación personalizada
- ✓ Indica que se requiere un campo
- ✓ Una función get que le permite manipular los datos antes de que se devuelva como un objeto
- ✓ Una función de conjunto que le permite manipular los datos antes de guardarlos en la base de datos
- ✓ Crear índices para permitir que los datos se obtengan más rápido

Ventajas:

- ➢ Un ODM simplifica el código:
- ➢ Nuestros objetos tendrán por defecto métodos como save o remove (delete es una keyword en JavaScript).
- ➢ Permite usar middlewares.
- ➢ Son hooks que se ejecutan antes o después (pre o post) de determinados eventos (validate, save, remove).
- ➢ Getters, setters, atributos virtuales.

5.4.2. Sistema de tokens

La tokenización o el sistema de tokens, cuando se aplica a la seguridad de los datos, se refiere al proceso de sustitución de un elemento de datos sensible por un equivalente no sensible denominado token, que no tiene un significado o valor extrínseco o explotable. El token es una referencia (un identificador) que regresa a los datos sensibles a través de un sistema de tokenización. El mapeo de datos originales a un token utiliza métodos que hacen que los tokens no sean factibles de revertir en ausencia del sistema de tokenización, por ejemplo, utilizando tokens creados a partir de números aleatorios.

Dicho esto, en el proyecto desarrollado se implementó una librería la cual permitió crear y administrar los tokens generados para así lograr tener un control total sobre la seguridad del aplicativo web desarrollado.

Jsonwebtoken: Es un token de acceso estandarizado en el RFC 7519 que permite el intercambio seguro de datos entre dos partes. Contiene toda la información importante sobre una entidad, lo que implica que no hace falta consultar una base de datos ni que la sesión tenga que guardarse en el servidor [29]

Por este motivo, los JWT son especialmente populares en los procesos de autentificación. Con este estándar es posible cifrar mensajes cortos, dotarlos de información sobre el remitente y demostrar si este cuenta con los derechos de acceso requeridos**.** Los propios usuarios solo entran en contacto con el *token* de manera indirecta: por ejemplo, al introducir el nombre de usuario y la contraseña en una interfaz. La comunicación como tal entre las diferentes aplicaciones se lleva a cabo en el lado del cliente y del servidor.

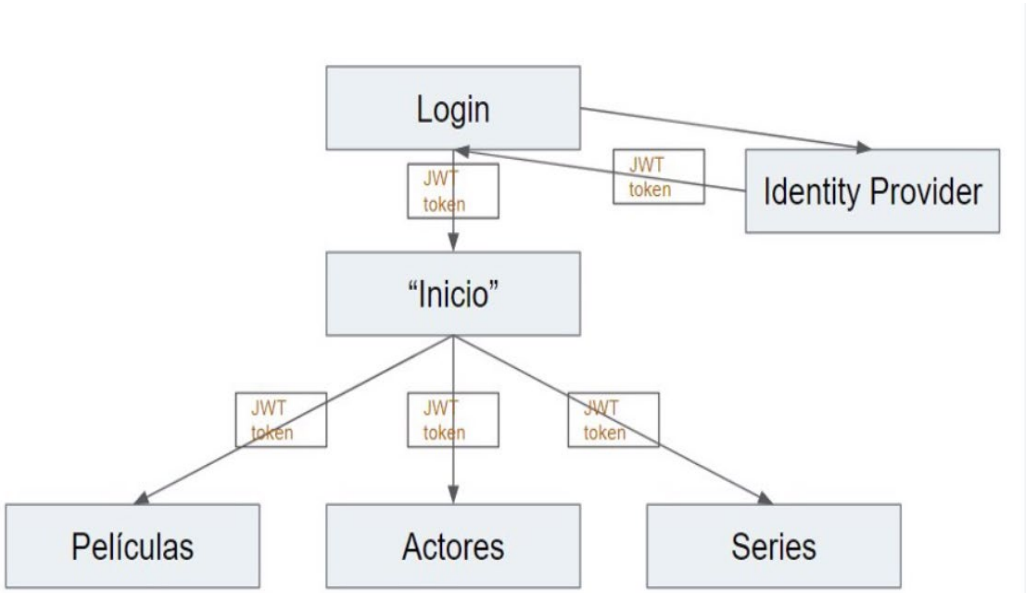


Figura 10 Sistema de Tokens

Fuente [29] . https://www.ionos.es

Ventajas:

- Tiene gran experiencia de usuario.
- Permite asegurar procedencia y validez.
- Todos sus request son legítimos.

Desventajas:

- No tiene

A continuación, se observa la implementación del JWT dentro del backend del aplicativo

Figura 11. Implementaciones del JWT dentro del backend

Fuente: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/json-web-token-jwt/

5.4.3. Middlewares

El middleware es el software que brinda servicios y funciones comunes a las aplicaciones, además de lo que ofrece el sistema operativo. Generalmente, se encarga de la gestión de los datos, los servicios de aplicaciones, la mensajería, la autenticación y la gestión de las API.

Ayuda a los desarrolladores a diseñar aplicaciones con mayor eficiencia. Además, actúa como hilo conductor entre las aplicaciones, los datos y los usuarios.

En el caso de las empresas con entornos de contenedores y multicloud, el middleware puede rentabilizar el desarrollo y la ejecución de las aplicaciones según sea necesario.
En el desarrollo se empleó un middleware muy potente llamado Morgan el cual detallare a continuación.

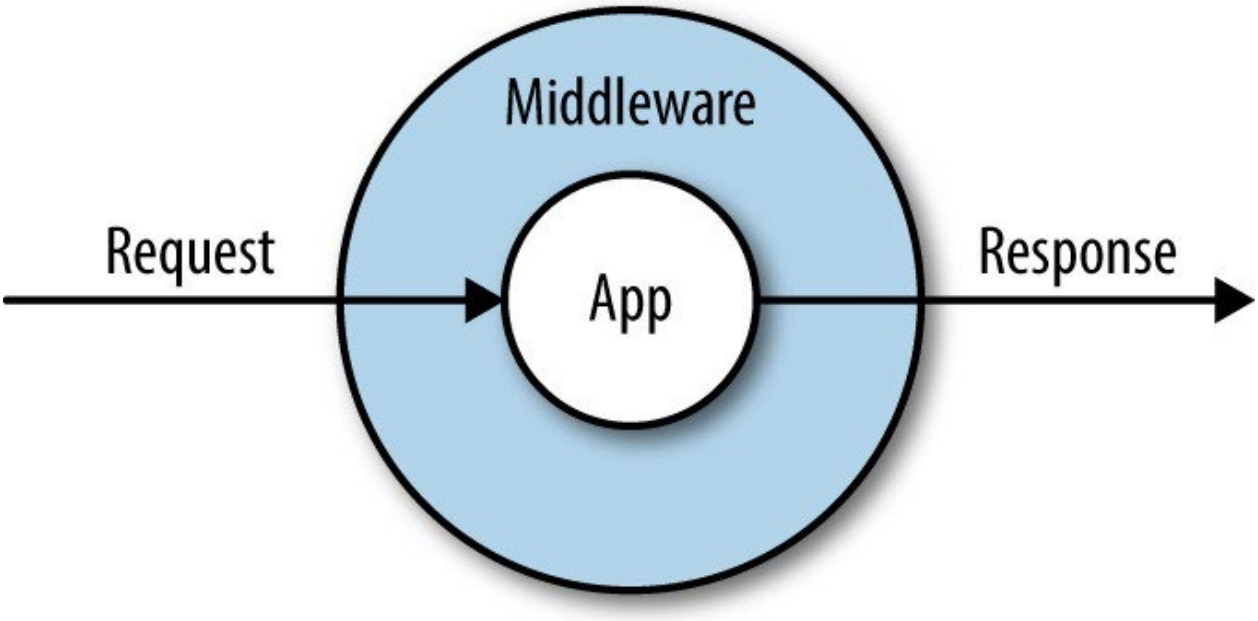


Figura 12. Middleware muy potente llamado Morgan
Fuente:https://medium.com/@aarnlpezsosa/middleware-en-express-js-5ef947d668b

Morgan: Es un Middleware de nivel de solicitud HTTP. Es una gran herramienta que registra las solicitudes junto con otra información dependiendo de su configuración y el ajuste preestablecido utilizado. Resulta muy útil durante la depuración y también si desea crear archivos de registro. https://www.geeksforgeeks.org/what-is-morgan-in-node-js/

Características:

- Escritura de manejadores de peticiones con diferentes verbos HTTP en diferentes caminos URL (rutas).
- Integración con motores de renderización de "vistas" para generar respuestas mediante la introducción de datos en plantillas.
- Establecer ajustes de aplicaciones web como qué puerto usar para conectar, y la localización de las plantillas que se utilizan para renderizar la respuesta.
- Añadir procesamiento de peticiones "middleware" adicional en cualquier punto dentro de la tubería de manejo de la petición.
- Middleware compatible para abordar casi cualquier problema de desarrollo web

Desventajas:

- Middleware bastante pesado.
- Actualización no suele ser compatible con otros entornos de desarrollo.
- Migración compleja.

5.4.4. Plataformas como servicio (PaaS) o (IaaS)

La plataforma como servicio (PaaS) es un entorno de desarrollo e implementación completo en la nube, con recursos que permiten entregar todo, desde aplicaciones sencillas basadas en la nube hasta aplicaciones empresariales sofisticadas habilitadas para la nube. Usted le compra los recursos que necesita a un proveedor de servicios en la nube, a los que accede a través de una conexión segura a Internet, pero solo paga por el uso que hace de ellos.

Al igual que IaaS, PaaS incluye infraestructura (servidores, almacenamiento y redes), pero también incluye middleware, herramientas de desarrollo, servicios de inteligencia empresarial (BI), sistemas de administración de bases de datos, etc.

PaaS está diseñado para sustentar el ciclo de vida completo de las aplicaciones web: compilación, pruebas, implementación, administración y actualización.

En sus características, el PaaS posee muchas situaciones que utilizar PaaS es beneficioso o incluso necesario. Si hay varios desarrolladores trabajando en el mismo proyecto de desarrollo, o si también se deben incluir otros proveedores, PaaS puede proporcionar gran velocidad y flexibilidad a todo el proceso. PaaS también es beneficioso si desea poder crear sus propias aplicaciones personalizadas. Este servicio en la nube también puede reducir en gran medida los costos y puede simplificar algunos desafíos que surgen si desarrolla o despliega rápidamente una aplicación.

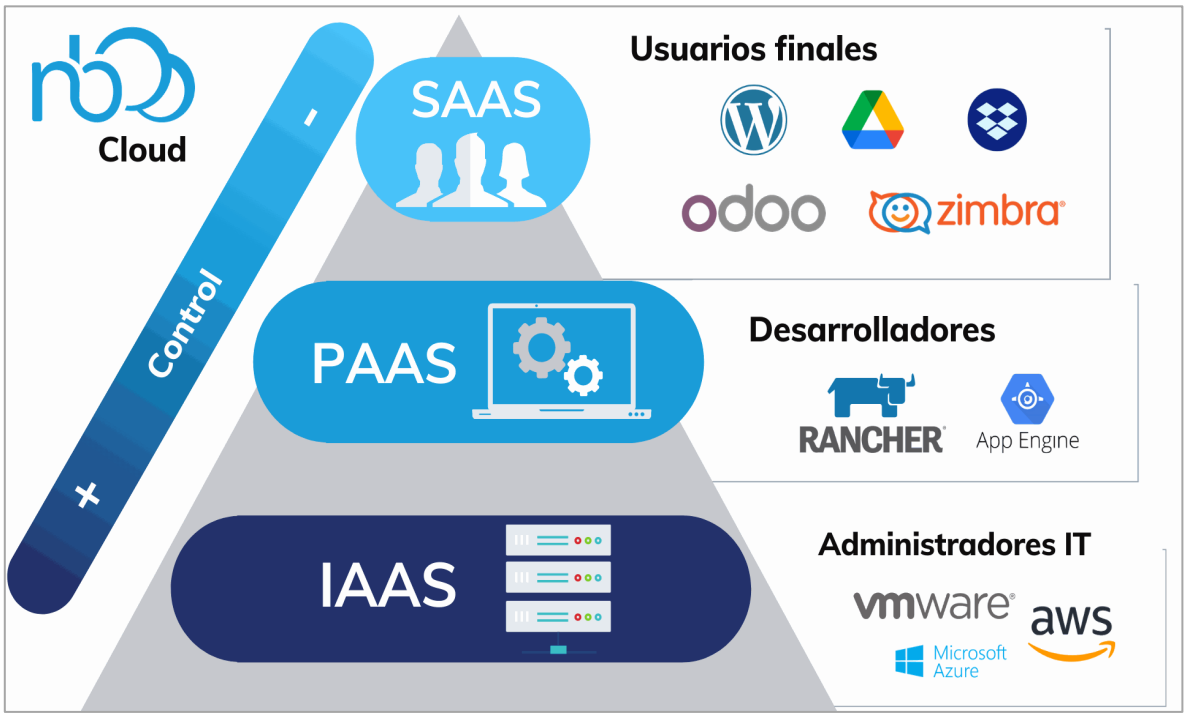


Figura 13. Middleware llamado Morgan

Fuente: https://nanobytes.es/blog/blog-nanobytes-1/cloud-computing-diferencia-entre-iaas-paas-y-saas-7

Ventajas:

Puesto que ofrece infraestructura como servicio, PaaS aporta las mismas ventajas que IaaS. Pero las características adicionales, como herramientas de desarrollo y otras herramientas empresariales, ofrecen más ventajas:

- Reducir el tiempo de programación. Las herramientas de desarrollo de PaaS pueden reducir el tiempo que se tarda en programar aplicaciones nuevas con componentes de aplicación preprogramados que están integrados en la plataforma, como flujos de trabajo, servicios de directorio, características de seguridad, búsqueda, etc.
- Agregar más funcionalidad de desarrollo sin incorporar más personal. Los componentes de plataforma como servicio pueden aportar a su equipo de desarrollo nuevas características sin necesidad de contratar personal especializado.
- Desarrollar para varias plataformas (Multiplataforma) con más facilidad. Algunos proveedores de servicios ofrecen opciones de desarrollo para varias plataformas, como PC, dispositivos móviles y exploradores, lo que agiliza y facilita el desarrollo de aplicaciones multiplataforma.
- Usar herramientas sofisticadas a un precio asequible. Gracias a un modelo de pago por uso, las personas u organizaciones pueden usar software de desarrollo sofisticado y herramientas de inteligencia empresarial y análisis cuya compra no se podrían permitir.
- Colaboración en equipos de desarrollo distribuidos geográficamente. Puesto que al entorno de desarrollo se accede a través de Internet, los equipos de desarrollo pueden colaborar en proyectos incluso si los miembros del equipo se encuentran en lugares diferentes.
- Administrar el ciclo de vida de las aplicaciones con eficacia. PaaS proporciona todas las características necesarias para sustentar el ciclo de vida completo de las aplicaciones web: compilación, pruebas,

implementación, administración y actualización, dentro del mismo entorno integrado.

- No necesitan invertir en infraestructura física: Poder “alquilar” una infraestructura virtual les supone ventajas tanto económicas como prácticas, les evita tener que comprar hardware por su cuenta y dedicar sus conocimientos a administrarlo, lo cual les deja más tiempo libre para concentrarse en el desarrollo de las aplicaciones. Además, los clientes sólo necesitarán alquilar los recursos que realmente necesiten, evitando así malgastar su dinero en la adquisición de capacidad fija que vaya a permanecer sin utilizarse la mayor parte del tiempo.
- Hace posible que incluso usuarios “no expertos” puedan realizar desarrollos: Con algunas propuestas de PaaS, cualquiera puede desarrollar una aplicación o un sitio web, sólo tiene que seguir los pasos necesarios a través de una sencilla interfaz web. Un excelente ejemplo de este tipo de aplicaciones son las instalaciones de software para la gestión de blogs como WordPress.
- Flexibilidad: Los clientes pueden disfrutar de un control total sobre las herramientas que se instalen en sus plataformas, y crear una plataforma perfectamente adaptada a sus necesidades concretas. Sólo tienen que ir seleccionando aquellas funcionalidades que consideren necesarias.
- Adaptabilidad: Las funcionalidades pueden modificarse si las circunstancias así lo aconsejan.
- Permite la colaboración entre equipos situados en varios lugares distintos; como lo único que se necesita es una conexión a Internet y un navegador web, los desarrolladores pueden estar dispersos por varios lugares distintos y aún así colaborar juntos en el desarrollo de la misma aplicación.
- Seguridad: Se ofrecen diversos mecanismos de seguridad, que incluyen la protección de los datos y la realización y recuperación de copias de seguridad.

Desventajas:

- Anteriormente PAAS contaba con ciertas desventajas, pero ya han sido solucionadas, una de sus desventajas era la necesidad de disponibilidad de los datos de la nube, la aplicación o plataforma no disponía de un sistema (web service, una API, etc..) que permita extraer los datos, era un claro inconveniente para no adoptar la aplicación en la nube. Pero estando esto ya solucionado no tendrá ninguna desventaja, y WAAS lo guiará en todo el proceso [29]

Heroku: Es uno de los PaaS más utilizados en la actualidad en entornos empresariales por su fuerte enfoque en resolver el despliegue de una aplicación. En otras palabras, a Heroku solo se le debe mencionar qué lenguaje de backend estás utilizando o qué base de datos vas a utilizar y genera todos los schemas para la aplicación, haciendo que el desarrollador solo se preocupe por el desarrollo del sitio web como fue en este caso.

Es importante mencionar que, si se encuentra desarrollando apps**,** debemos ser conscientes que el impacto inicia cuando comienzas a tener usuarios, por esto es importante lanzar tu aplicación sin tener complicaciones de infraestructura, administrar servidores, las bases de datos y la seguridad que estos deben de tener entre otras cosas.
Heroku posee diversas características tales como:

- Soporta diferentes lenguajes de programación: Node, Ruby, Java, Clojure, Scala, Go, Python, PHP
- Tiene una versión gratuita fácil de usar
- Ejecuta las aplicaciones a través de sus contenedores, también conocidos como Dynos
- Tiene Dynos que pueden ser de tres tipos: web, worker o cron
- Ofrece más de 200 complementos con los que ampliar las aplicaciones al instante

- Ofrece varias características de seguridad, incluyendo SSL, autenticación y cumplimiento de PCI
- Cabe destacar que dentro del desarrollo del proyecto se empleó la computación en la nube implementando esta valerosa APP, la cual me permitió administrar y desarrollar mi software en su hosting de microservicios.

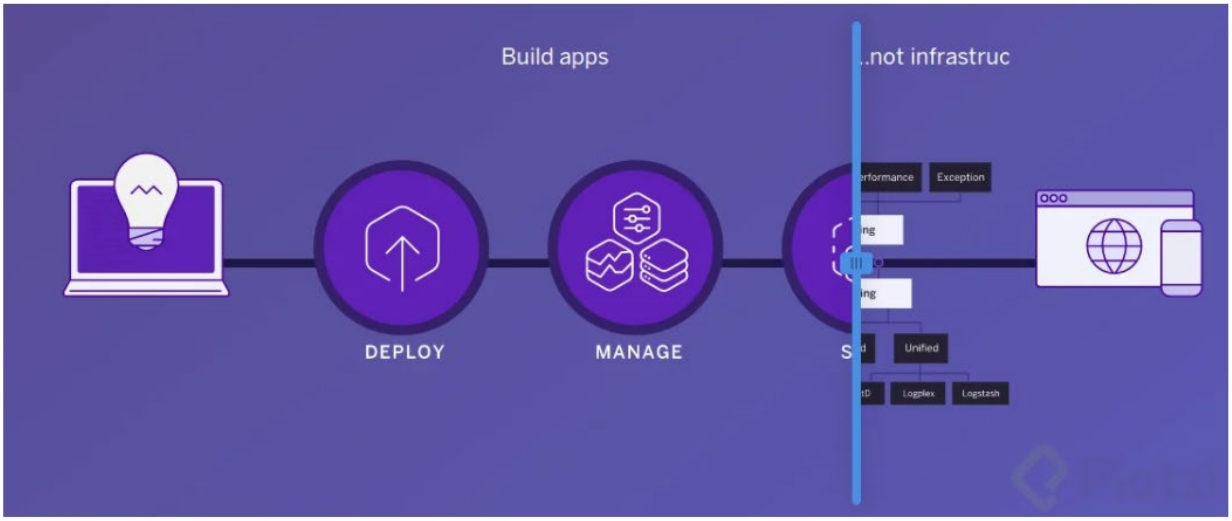


Figura 14. Hosting de microservicios

Elaboración propia presente estudio.

5.4.5. Upload y manejo de estadística grafica

La carga y el manejo de estadística grafica es un tema muy importante dentro del módulo de batería de indicadores institucionales que se desarrolló en el aplicativo, dado que esto permite tener de una manera interactiva las métricas que requiere la institución para poder llevar control de una manera más amena la estadística que ellos tienen en diversos factores.

5.4.6. Pruebas unitarias y de instrumentación

Las librerías seleccionadas en esta categoría se encargan de facilitar el desarrollo de pruebas unitarias y de instrumentación para las diferentes funcionalidades de una aplicación web.

Mocha js: Es el marco de prueba más popular que admite pruebas de backend y frontend. MochaJS es una base flexible para desarrollar pruebas según lo necesite. Ejecuta las pruebas de forma asincrónica en el motor Chrome v8 o en cualquier otro navegador.

Los principales beneficios de Mocha incluyen:

- Funciona tanto para frontend como para backend
- Saporta depurador NodeJS
- Proporciona una base limpia para desarrollar pruebas según la conveniencia del desarrollador.
- Admite cualquier navegador, incluida la biblioteca de Chrome sin cabeza
- Admite la simulación de objetos para realizar pruebas de backend flexibles

Desventajas:

- Relativamente nuevo (comenzó en 2012), lo que significa que hay muchas incógnitas y variabilidad, en algunos aspectos o falta de base de usuarios y soporte.

  Solo se proporciona la estructura de prueba básica, por lo que se requieren ajustes y configuraciones adicionales (lo que puede ser beneficioso para algunos).
- Si está buscando un marco de prueba de unidades independiente para JavaScript

Jazmin: Es un imitador del comportamiento del usuario que le permite realizar casos de prueba similares al comportamiento del usuario en su sitio web. Jasmine es útil para una interfaz de prueba de visibilidad, claridad de clics y capacidad de respuesta de la interfaz de usuario en diferentes resoluciones. Jasmine permite automatizar el comportamiento del usuario con retrasos aduaneros y tiempo de espera para simular el comportamiento real del usuario.

Los principales beneficios de usar Jasmine incluyen:

- Menores gastos generales debido a casi cero dependencias externas

- Viene con casi todas las herramientas necesarias fuera de la caja
- Soporta pruebas de Frontend y Backend
- La codificación es bastante similar a escribir en lenguaje natural.
- Amplia documentación para usarlo con varios frameworks

Desventajas:

- En la mayoría de los casos, requiere un corredor de pruebas (como Karma).
- Es difícil realizar pruebas de forma asincrónica.
- Si está buscando una solución de prueba unitaria unificada (cliente-servidor), Jasmin puede ser poco adecuada.

Karma: Es un entorno de prueba productivo que admite todo el marco de descripción de prueba popular dentro de sí mismo. Brinda a su aplicación el soporte para ejecutar pruebas en diferentes entornos. Tiene un amplio soporte para la ejecución de pruebas en diferentes dispositivos y aplicaciones.

El factor principal para elegir Karma radica en su soporte para integrarse con motores CI / CD y las siguientes características.

- Se puede utilizar para ejecutar pruebas en navegadores, entornos sin cabeza como PhantomJS y en dispositivos.
- Admite pruebas escritas en la mayoría de los marcos populares
- Permite ejecutar pruebas de forma remota en otros dispositivos con solo los archivos que vienen
- Admite la depuración de casos de prueba con Chrome y Webstorm

Desventajas:

- Si necesita habilidad para emular, debe agregar Mockito (biblioteca de simulación).
- Debido a que los nombres de métodos en JUnit están restringidos por las convenciones de Java y otras razones, es difícil para el personal no técnico leer los resultados de la prueba.

Titiritero: Es un excelente marco de ejecución de pruebas creado por un equipo de Google. Proporciona una API de Chrome sin cabeza para aplicaciones NodeJS.

Puppeteer se usa principalmente para aplicaciones específicas del navegador como prueba de rastreo, prueba de estructura de página, toma de capturas de pantalla e incluso captura de contenido pre-renderizado para aplicaciones de una sola página.

Los beneficios adicionales de usar titiritero son:

- Posibilidad de configurar resoluciones y tamaños personalizados para el navegador
- Soporte para probar extensiones de Chrome
- Soporte de automatización para envío de formularios, pruebas de interfaz de usuario y entradas de teclado
- Admite funcionalidades de ES6 como await y async
- Escrito en Java puro.
- Admite el desarrollo basado en pruebas (TDD).
- Permite crear su propia suite de casos de prueba unitaria.
- Puede integrarse bien con otras herramientas (como Maven) e IDE (como IntelliJ).
- Se ha desarrollado durante mucho tiempo y tiene su propio gran grupo de usuarios, por lo que es fácil encontrar los documentos.

Desventajas:

- Los informes HTML personalizados son engorrosos.
- Solo se admite Java y se requieren al menos conocimientos básicos del lenguaje de programación Java.

5.4.7. Herramientas de apoyo para desarrollo

Visual studio code: Actualmente, visual studió code se posiciona como el editor de código más empleado o usado por los desarrolladores a nivel mundial, dado que

posee características muy buenas, brindando un entorno de desarrollo moderno, amigable y óptimo para realizar desarrollos software.

Este editor fue desarrollado por Microsoft para casi todas las plataformas existentes en la actualidad, tales como: macOS, Windows, Linux. Además, es importante señalar que Visual studio code es compatible con diferentes lenguajes de programación.

Hooks: Hay que mencionar que este componente es completamente opcional, se puede probar los hooks en unos pocos componentes sin la necesidad de reescribir algún código existente.
Cabe destacar que no hay que tener conocimientos muy avanzados para poder aprender o usar hooks.
Por otra parte, son 100% compatibles con versiones anteriores y tienes características importantes como:

- Los Hooks no tienen cambios con rupturas con respecto a versiones existentes.
- Disponibles de inmediato.
- Los Hooks ya están disponibles con el lanzamiento de la versión v16.8.0.
- Los Hooks proporcionan una API más directa a los conceptos que se conocen de React: props, estado, contexto, referencias, y ciclo de vida.

Además, este elemento también ofrece una nueva y poderosa forma de combinarlos dentro de la usabilidad del proyecto, permitiendo resolver una amplia variedad de problemas aparentemente desconectados en React que diversos desarrolladores han encontrado durante más de cinco años de codificar y mantener decenas de miles de componentes.
Entre los problemas que presenta el uso de hooks en un proyecto basado en reactJs, se encuentran los siguientes:

- React no ofrece una forma de “acoplar” comportamientos reutilizables a un componente.

Para finalizar, los Hooks, permiten extraer lógica de estado de un componente de tal manera que se pueda probar y reutilizar independientemente. Además, permiten reutilizar lógica de estado sin cambiar la jerarquía de tu componente, facilitando el compartir Hooks entre muchos componentes o incluso con la comunidad.

Se presenta ilustración de empleo de un hook en React de un estado denominado State.

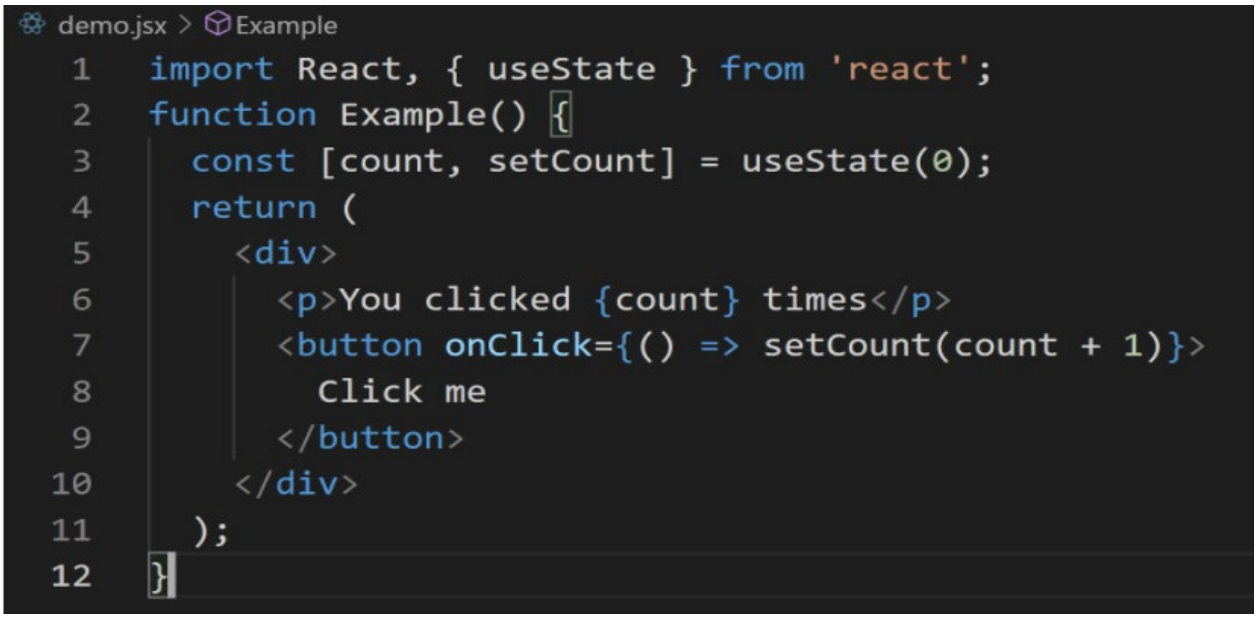


Figura 15. Empleo de un hook en React

Fuente: Elaboración propia presente estudio. Año 2022

5.5. Desarrollo del proyecto.

Basados en la investigación de librerías, frameworks, dependencias realizadas en los capítulos anteriores, se tomó la decisión de usar e implementar las tecnologías que por sus características propias podían brindar mayor eficacia y productividad a la hora de desarrollar los diversos módulos propuestos para lograr culminar con éxito el aplicativo para la Fundación Gimnasio Moderno del Cauca.

La arquitectura general del proyecto se escogió y se diseñó basados en el conocimiento aportado sobre el manejo de los microservicios.

Es importante señalar que se trabajó sobre plataformas libres, en este caso especialmente se desarrolló sobre un conjunto de tecnologías llamado el Stack MER, con un grupo explícito de aplicativos como MongoDB Atlas-Mongo. ExpressJs, React.

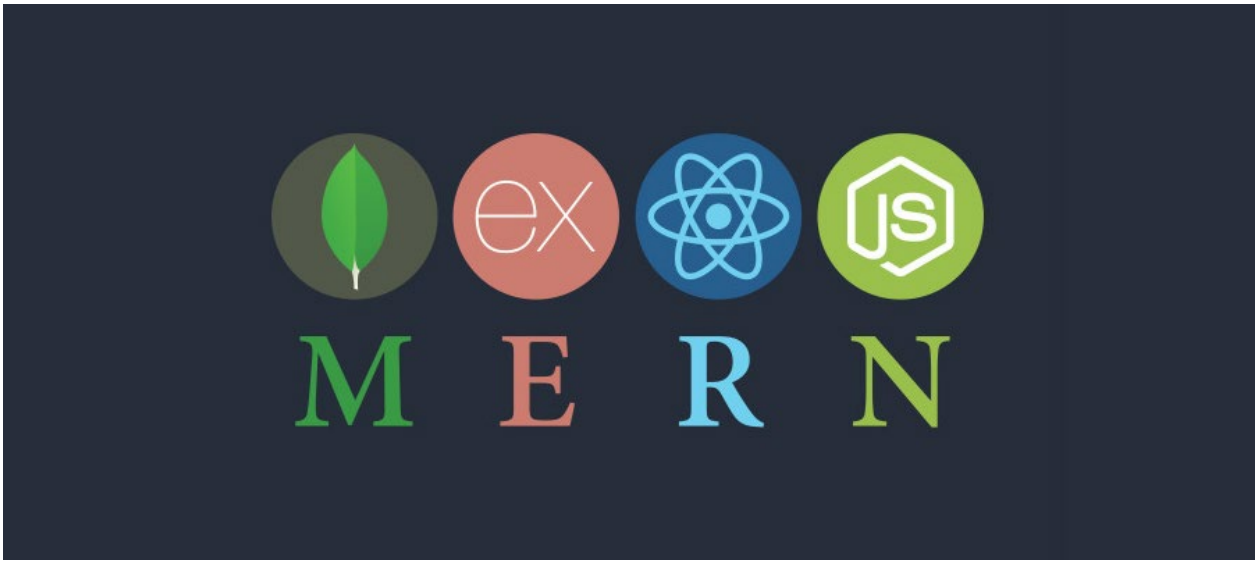


Figura 16. Plataformas libres

Foto: https://platzi.com/blog/que-es-mern-stack-javascript/

5.5.1 Editor de código del proyecto - Visual studio Code

Para empezar con el desarrollo de la aplicación web, se descarga el editor de código Visual studio code en su versión de Windows x64, editor robusto en el cual se trabajaron los dos frentes del aplicativo (Beackend y Frontend).

5.5.2 Bases de datos

Para el desarrollo del proyecto se toma la decisión de usar una base de datos no relacional, la cual permite trabajar con el grupo de tecnologías más vanguardistas de la actualidad en cuanto a desarrollo web, dicha base de datos es MongoDB Atlas en su versión estable.

5.4.3 Conexión BD Mongo atlas + Backend

En este apartado, se toma la decisión de realizar la conexión e integración del backend con MongoDB Atlas, para esto se requiere Mongoose que es una biblioteca empleada en JavaScript la cual permite y tiene por capacidad, definir todos los esquemas que tienen en sus características propias los estados fuertemente tipados que posteriormente son asignados a todos los archivos y distintos documentos MongoDB.

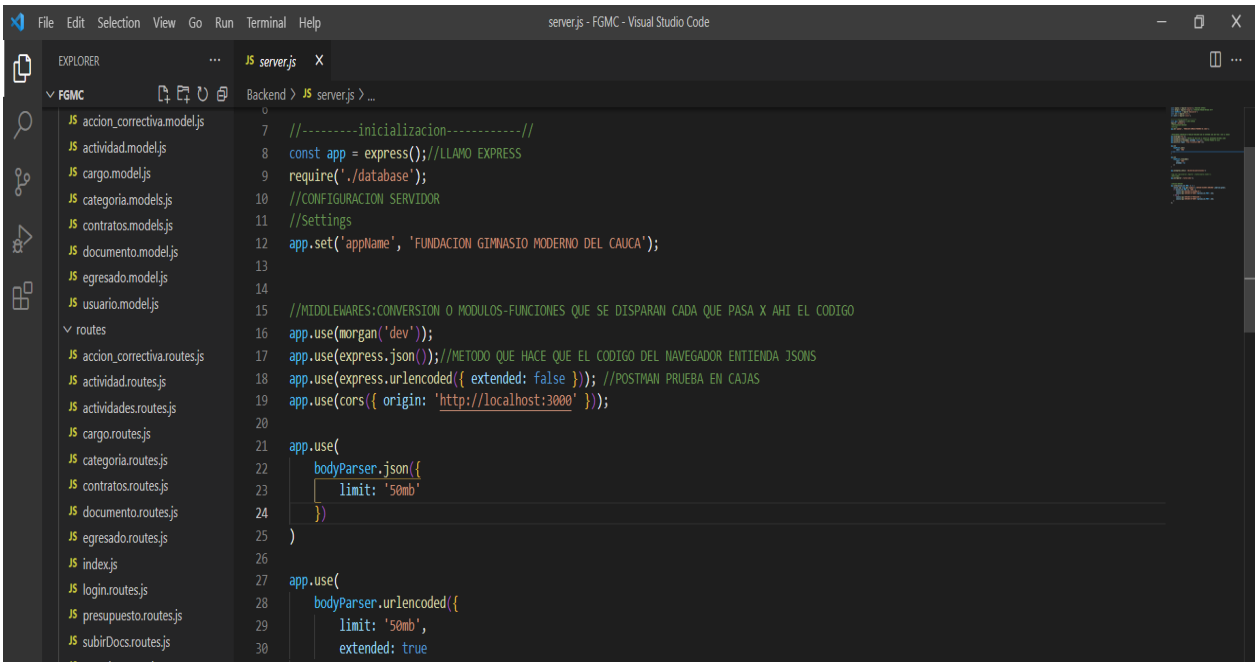
Dado que el software desarrollado tiene en sus principales módulos diversas tipificaciones de schemas (String, Number, Date, Buffer, Boolean, Mixed, ObjectId y Array) que se emplearon para la construcción del software dentro de los diversos endpoint que se hicieron en el sistema.

Siendo así, se realizó la conexión de la siguiente manera, mostrando un estado de alerta si la base de datos en la nube presentaba alguna destructuración o falla a la hora de realizar su conexión empleando o requiriendo la librería Mongoose.
Aquí se presenta su integración con el código siguiente:

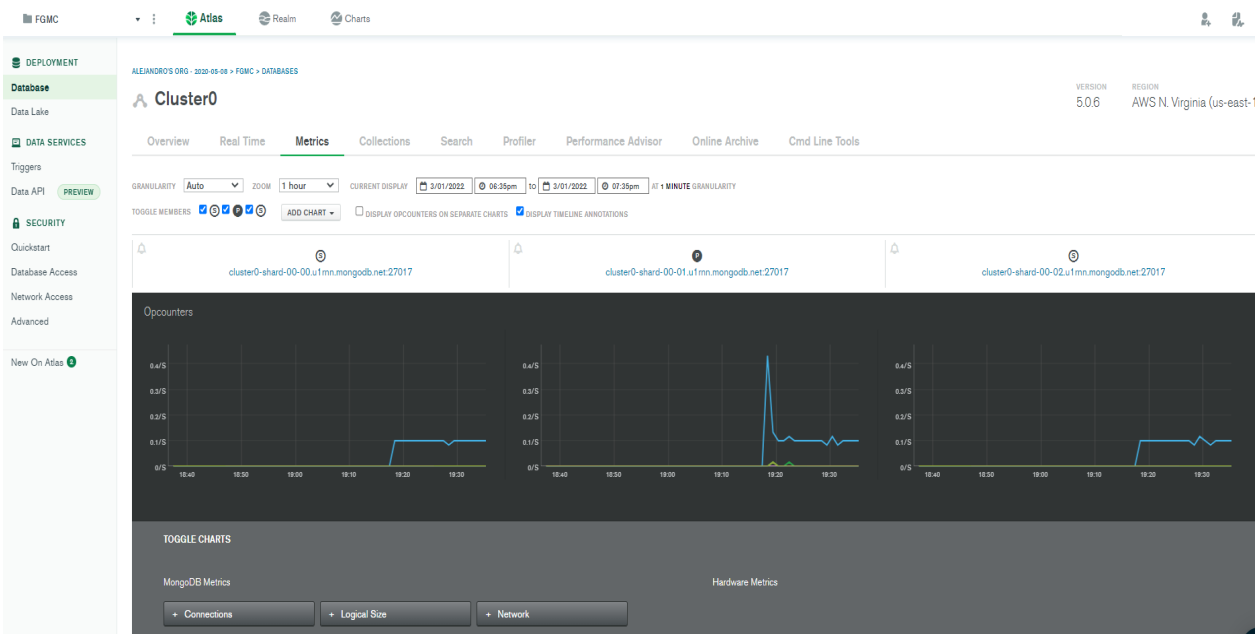
5.4.4. Conexión MongoDB Atlas

Para lograr culminar el proceso de conexión con MongoDB Atlas, se requiere realizar una multiconexion a él aplicativo que por defecto se despliega en la nube, el cual indica que debe generarse primeramente un clúster donde se plantea alojar la base de datos realizada, posteriormente se ejecuta un comand-line en consola o directamente desde la configuración en la página de MongoDB Atlas, dicho comando se emplea para brindar permisos administrables a el clúster creado.

A continuación, se procede a inicializar la base de datos empleando expressJs dentro del backend desarrollado como se muestra en la siguiente imagen.

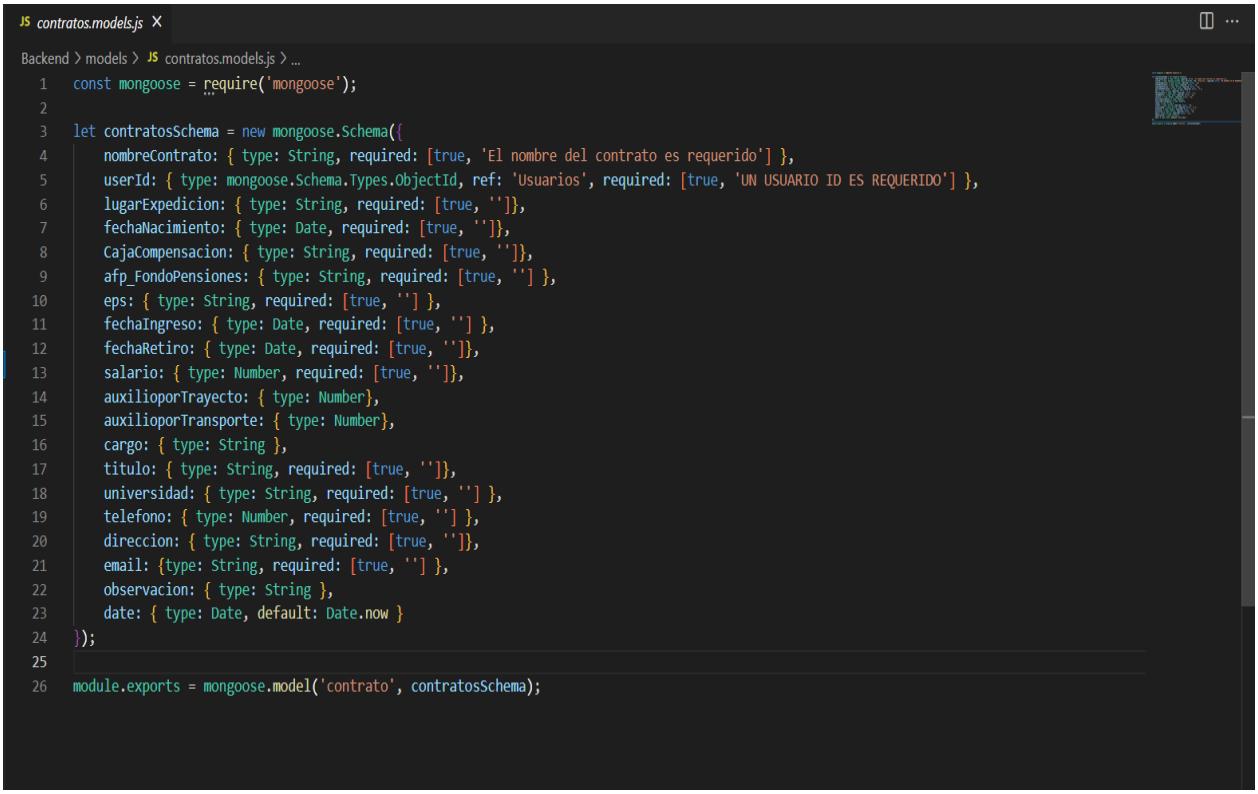


Dicho lo anterior, la conexión se encuentra finalizada y permite tener acceso a la parte grafica del clúster en MongoDB Atlas como se muestra a continuación.



5.4.5 Modelos de las estructuras del backend.

Para el desarrollo de esta tarea, fue necesario plasmar toda la estructura que tendría los diversos endpoints del sistema, empleando y llamando la librería mangosta de manera requerida en el sistema a implementar, es importante en este apartado mencionar los campos que se idealizan o requieren como formatos requeridos dentro del aplicativo y el tipo de data que va a asignarse (Date, Boolean, String. etc.) como se muestra a continuación:



5.5 Creación de controladores

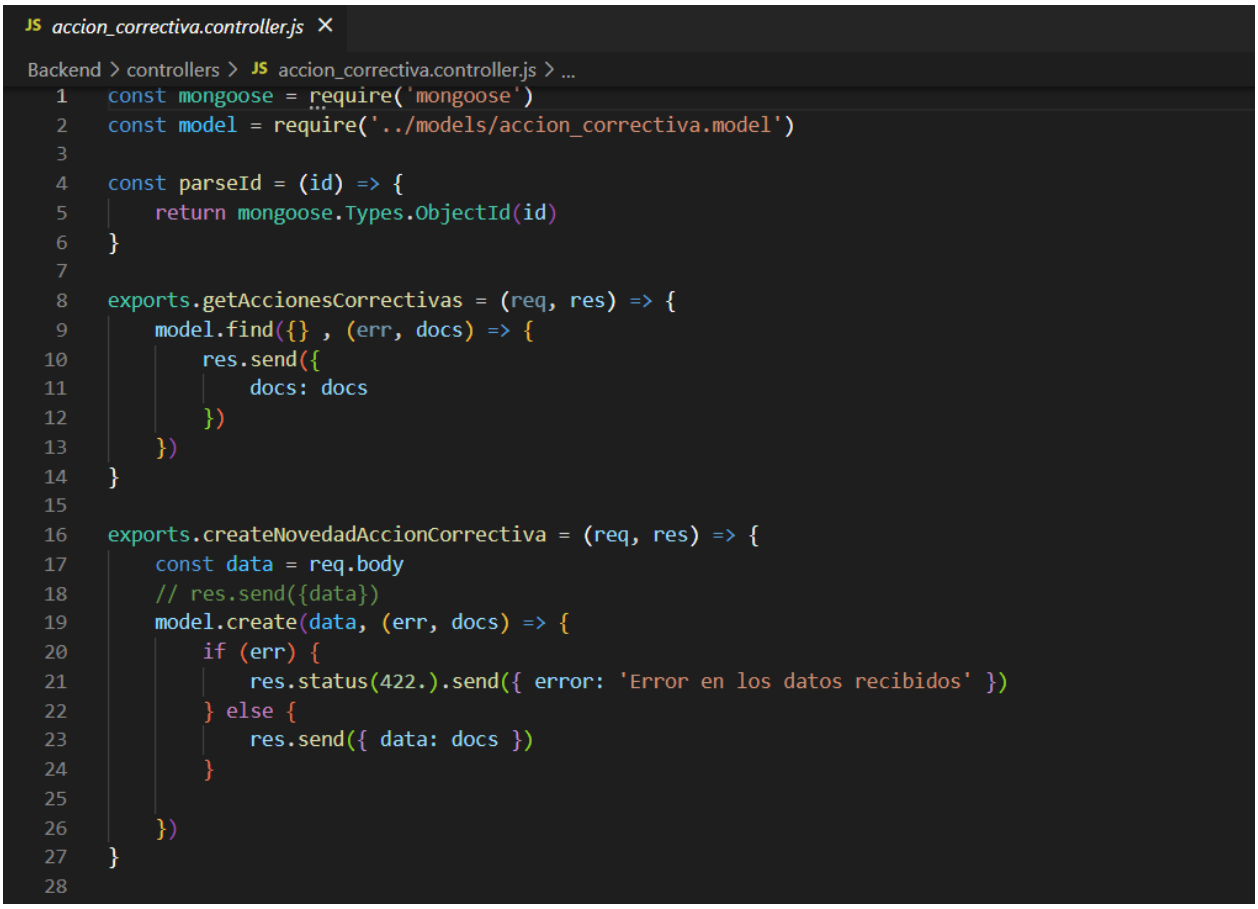
La implementación de los controladores es importante dado que trae toda la lógica de negocio, las jerarquías y el funcionamiento que se emplea en el desarrollo del proyecto, para así brindar las acciones necesarias que requiere el software para la optimización y correcta puesta en marcha de cada uno de los ítems o características que necesita el desarrollo.

Para el desarrollo de las tareas de los controladores, en el transcurso del proyecto fue necesario dividir cada controlador en una vista independiente siguiendo el manual de buenas practicas de desarrollo software como se muestra a continuación:

JS contratos.controller.js

Backend > controllers > JS contratos.controller.js > getContratos > exec() callback

```
75  contratosCtrl.getContrato = (req, res) => {
76      let id = req.params.id;
77      Contratos.findById(id, (err, responseDetail) => {
78          if (err) {
79              return res.status(500).json({
80                  response: {
81                      status: false,
82                      err
83                  }
84              });
85          }
86
87          if (!responseDetail) {
88              return res.status(400).json({
89                  response: {
90                      status: false,
91                      err: {
92                          message: "ID NO ENCONTRADO"
93                      }
94                  }
95              });
96          }
```



```
JS accion_correctiva.controller.js ×
Backend > controllers > JS accion_correctiva.controller.js > ...
1   const mongoose = require('mongoose')
2   const model = require('../models/accion_correctiva.model')
3
4   const parseId = (id) => {
5       return mongoose.Types.ObjectId(id)
6   }
7
8   exports.getAccionesCorrectivas = (req, res) => {
9       model.find({} , (err, docs) => {
10          res.send({
11              docs: docs
12          })
13      })
14  }
15
16  exports.createNovedadAccionCorrectiva = (req, res) => {
17      const data = req.body
18      // res.send({data})
19      model.create(data, (err, docs) => {
20          if (err) {
21              res.status(422.).send({ error: 'Error en los datos recibidos' })
22          } else {
23              res.send({ data: docs })
24          }
25
26      })
27  }
28
```

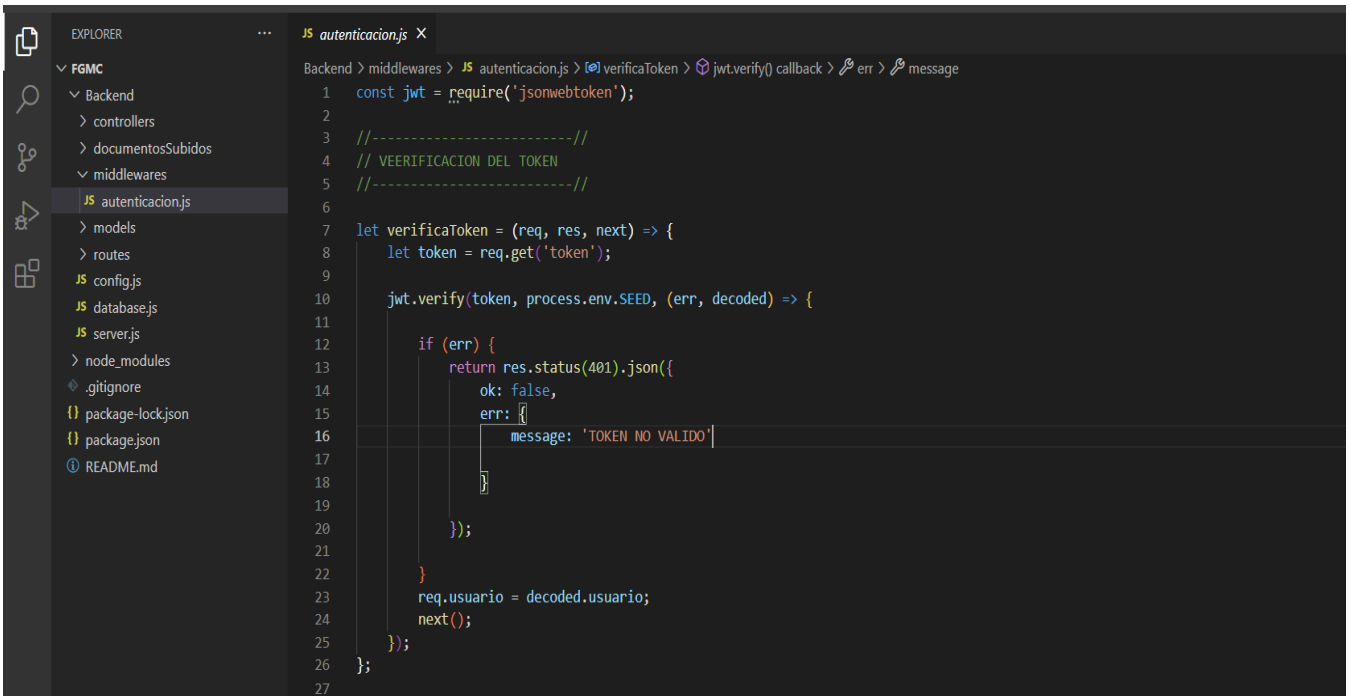
Fuente: Elaboración propia presente estudio. Año 2022

Seguido a esto, se realizaron todos los bucles anidados, las peticiones http y las respuestas que brindaría el software, para lograr la posterior captura de cada uno de los datos en el desarrollo del frontend.

5.6 Middleware Autenticación

El desarrollo de este área tiene implicación en todo el proyecto y por eso fue necesario darle bastante importancia dado que este middleware de autenticación recorre todos los aspectos del software conforme se estipuló en el apartado de los requisitos funcionales, el proceso que se realizó fue crear una constante de tipo

JWT (Jason web token), la cual verifica que la persona que realiza el inicio de sesión tenga una llave única y exclusiva a las otras personas que entren a el aplicativo, haciendo que la persona en base de datos tenga un identificador personal que le permite mirar la experiencia de usuario de una manera u otra.



Fuente: Elaboración propia presente estudio. Año 2022

```
//---------------------------//
// VEERIFICACION ADMIN ROL
//---------------------------//

let verificaAdmin_Rol = (req, res, next) => {

    let usuario = req.usuario;
    console.log(usuario);

    if (usuario.rol === 'adminRol') {
        next();

    } else {
        return res.json({
            ok: false,
            err: {
                message: 'EL USUARIO NO ES ADMINISTRADOR'
            }
        });
    }
}

module.exports = {
    verificaToken,
    verificaAdmin_Rol
}
```

Fuente: Elaboración propia presente estudio. Año 2022

5.7 Rutas

para el desarrollo de esta funcionalidad fue necesario implementar una función dentro de las vistas que serían agregadas en el desarrollo del frontend para poderlas consumir o llamar posteriormente, logrando implementar una forma de comunicar el evento con la actividad para permitir ejecutar la acción deseada, que

en este caso es lograr tener una navegación entre cada una de las rutas desarrolladas dentro del aplicativo web.

Fuente: Elaboración propia presente estudio. Año 2022

5.8. Desarrollo Frontend

Hay que mencionar que este desarrollo se implementó con React, el cual es una biblioteca de JavaScript propia que permite la construcción y manipulación de interfaces de usuarios. React maneja gran variedad de componentes que permiten un desarrollo visual interactivo que permite brindar una experiencia de usuario agradable, practica, responsive y robusta.

Es importante destacar que React posee vistas declarativas que hacen que todo el código desarrollado a lo largo del proyecto sea predecible, lo cual es bueno dado que permite depurarlo de manera ágil y sencilla. Esta biblioteca se basa en componentes asi que la misma crea componentes encapsulados que manejen su propio estado, y conviértelos en interfaces de usuario complejas, en el desarrollo

del frontend de esta aplicación, la lógica de los componentes se encuentra escrita en JavaScript y no en plantillas, permitiendo que se acceda a pasar datos de forma sencilla a través de la aplicación y mantener el estado fuera del DOM (Modelo de objetos del documento html-xml).

5.8.1 Pantalla principal

Se muestra una interfaz de acceso al aplicativo, la cual cumple a cabalidad con los requisitos funcionales del acceso login suministrado en el apartado de requerimientos funcionales, la interfaz presenta dos campos de solicitud de información por la cual el usuario va a digitar su correo y posteriormente su contraseña, seguido a eso la pantalla contiene un botón el cual va a permitir iniciar sesión y lograr continuar con el acceso del aplicativo como se evidencia en la fotografía siguiente:

5.8.2 Validaciones Login

Con el objetivo de brindar una experiencia de usuario conforme a los estándares de desarrollo software, se utilizaron validaciones por medio las cuales se lograría evidenciar que el software explica cual o cuales son los pasos erróneos que el usuario externo que maneja el aplicativo esta cometiendo, mostrando mensajes en los campos respectivos que las validaciones fallan, como se muestra en la siguiente imagen:

1. Usuario presiona botón iniciar sesión sin llenar los dos parámetros
2. Usuario presiona botón iniciar sesión sin llenar campo correo
3. Usuario presiona botón iniciar sesión sin llenar campo contraseña

GIMNASIO MODERNO DEL CAUCA

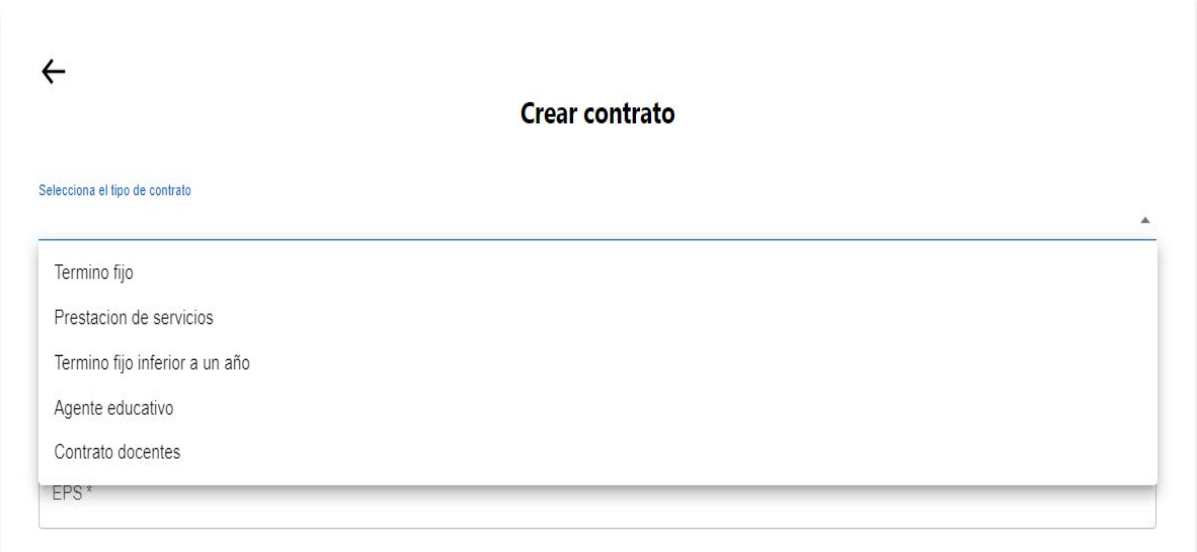
5.9 Contratación Institucional

Al momento que el usuario tiene acceso de login permitido, se evidencia un menú lateral o sidebar en el cual se pueden evidenciar cinco diferentes módulos que posee el desarrollo software o el aplicativo. Al momento que el usuario selecciona la vista llamada “Contratación institucional” y seguido a ello presiona el botón crear contrato, el aplicativo le permite visualizar el formulario que se debe diligenciar para así lograr crear un contrato de cualquiera de los siguientes tipos:

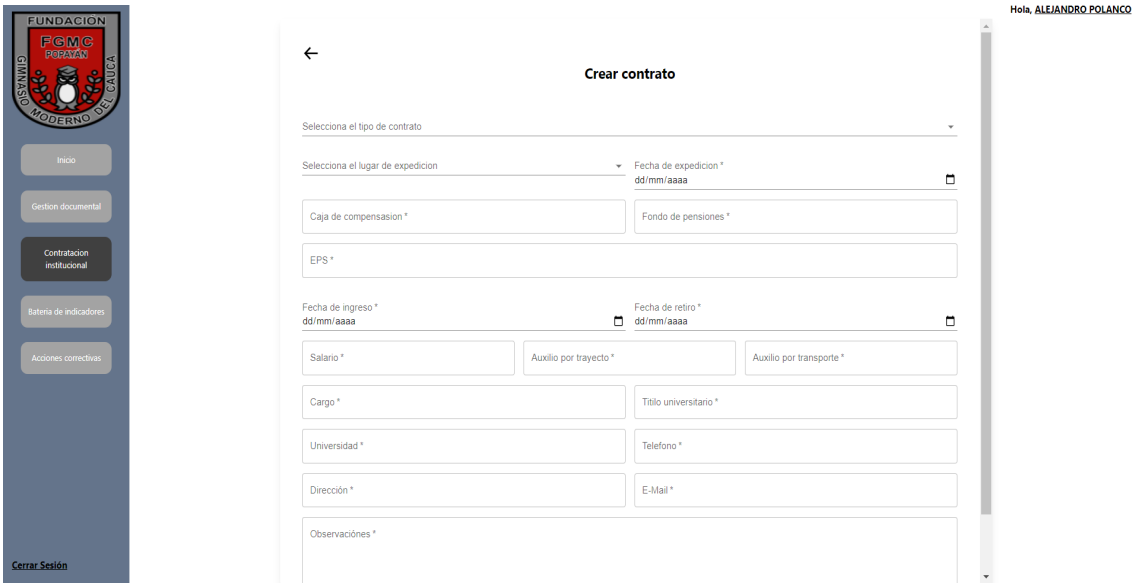
1. Contrato individual de trabajo a término fijo inferior a un año para docentes privados
2. Contrato de trabajo por duración de una obra o labor contratada

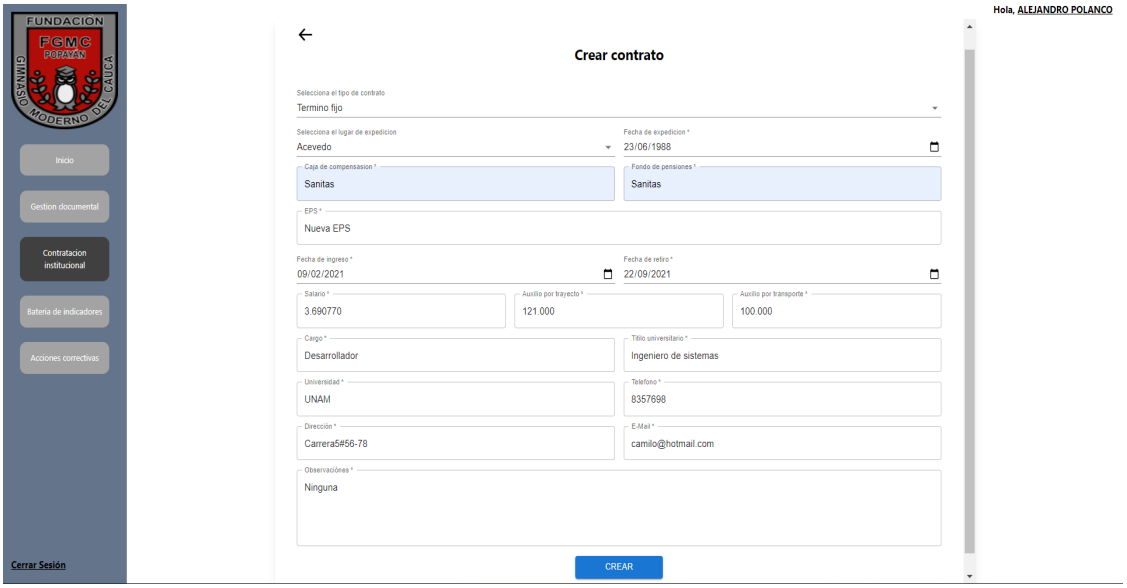
3. Contrato individual de trabajo a término fijo inferior a un año
4. Contrato individual de trabajo a término fijo a un año.
5. Contratos de tipificación especial.

Los tipos de los contratos que se evidencian, fueron programados a través de un DropDownLits o lista despegable la cual se muestra en la siguiente foto.



Seguido a ello se evidencia todo el formulario de contratación institucional el cual debe ser debidamente diligenciado por la persona encargada del área de talento humano de la Fundación gimnasio moderno del Cauca.





A continuación, se logra evidenciar el cargue de la data o información que fue previamente suministrada en el módulo de contratación institucional. Dentro del aplicativo MongoDB Atlas, se puede examinar la manera en la que se ve la

información enviada, dado que se conecta el servicio realizado en el backend con el aplicativo desarrollado en el frontend,

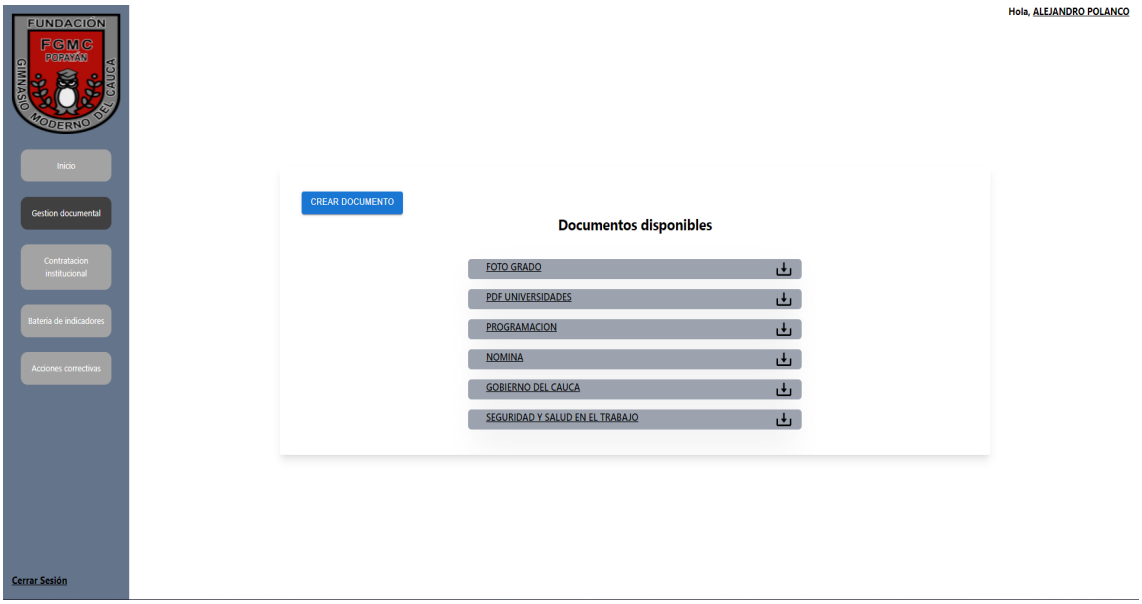
esto sucede cuando el personal de talento humano de la mencionada institución, desea registrar la data escrita en el formulario y desea finalizar el contrato cargándolo al clúster en la nube por medio de la base de datos, esta acción se realiza al dar click en el botón crear contrato del apartado anterior, mostrando la siguiente data en MongoDB Atlas:


```
_id: ObjectId("621e71bd782f758508a5322c")
nombreContrato: "Prestacion de servicios"
lugarExpedicion: "cali"
fechaNacimiento: 1990-05-03T05:00:00.000+00:00
CajaCompensacion: "sura"
afp_FondoPensiones: "colpensiones"
eps: "sanitas"
fechaIngreso: 1997-05-03T05:00:00.000+00:00
fechaRetiro: 1998-05-03T05:00:00.000+00:00
salario: 1200000
auxilioporTrayecto: 121000
auxilioporTransporte: 123000
cargo: "Desarrollador"
titulo: "Ingeniero de sistemas"
universidad: "UNAM"
telefono: 876543212
direccion: "calle 1012"
email: "camilo18@hotmail.com"
observacion: "ninguna"
userId: ObjectId("61e102d37a29976c78e5551b")
date: 2022-03-01T19:19:25.889+00:00
__v: 0
```

6.0 Gestión Documental

El módulo de gestión documental institucional, presenta dos pantallas o vistas interconectadas una a la otra, la primera pantalla nos muestra una sección donde se alojan todos los documentos institucionales tales como misión, visión, métricas de evaluaciones, reportes, requerimientos, etc.

los usuarios quienes tienen acceso a este módulo pueden descargar los archivos que se encuentran alojados en el mismo, como se observa en la imagen siguiente:



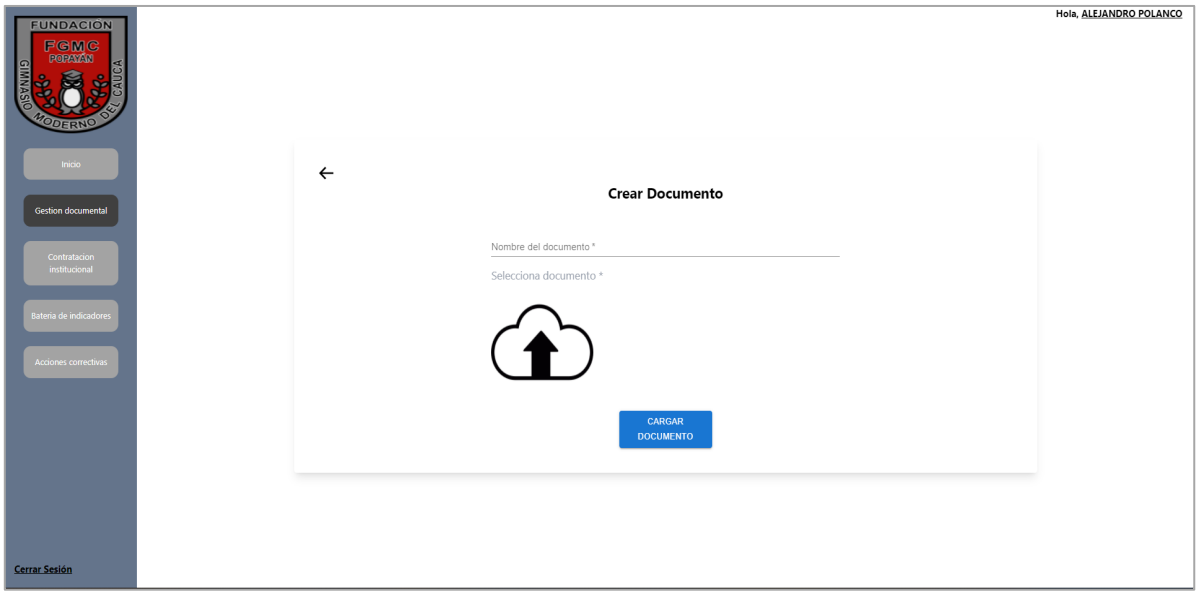
Fuente: Elaboración propia presente estudio. Año 2022

Dentro del módulo se encuentra el botón denominado crear documento el cual redirige a la vista numero dos (2) del módulo de gestión documental, esta vista se desarrolla empleando una librería llamada multer la cual permite hacer upload de archivos, data y diferente tipo de información que requiera el usuario.

Como se puede evidenciar, esta vista se fundamenta en subir archivos al sistema, dichos elementos que se pueden subir son de extensión xml, pdf, png o jpeg, extensiones las cuales fueron expuestas en los requisitos funcionales.

El aplicativo posee un botón de cargar documento, además tiene la acción de poder enviar el archivo seleccionado a la base de datos para su posterior uso.

Se evidencia en la fotografía siguiente:

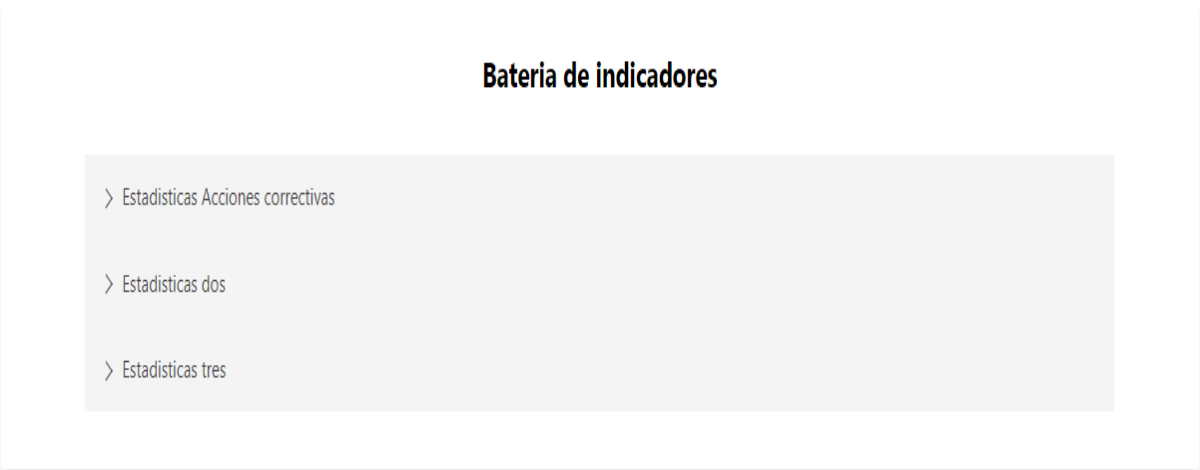


Fuente: Elaboración propia presente estudio. Año 2022

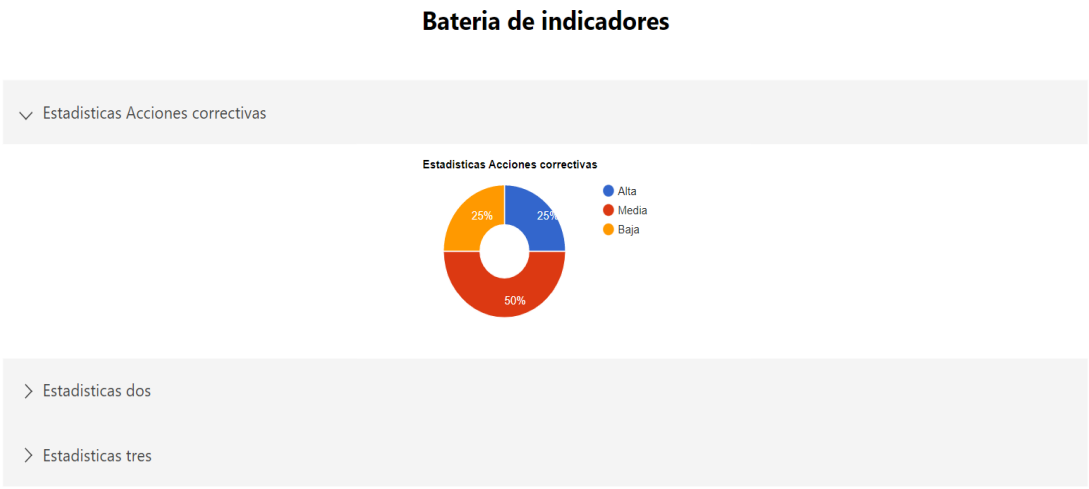
6.1 Batería de indicadores

La batería de indicadores, es un apartado muy importante en el aplicativo dado que permite a los usuarios mirar estadísticas institucionales veraces y dinámicas, dichas estadísticas hacen alusión desde la cantidad de contratos generados al momento hasta la complejidad de las acciones correctivas que se encuentren activas en la Fundación Gimnasio Moderno del Cauca.

El desarrollo de este módulo se implementó con react-google-charts lo cual es una librería que permite visualizar data o información institucional de manera grafica. Cabe mencionar que la estadística grafica expuesta se implementó e introdujo para una mejor visualización y mejor experiencia de usuario, al momento de interactuar en el aplicativo dentro de un acordeón como se logra evidenciar en la foto a continuación:



Cuando el usuario en cuestión, da click sobre una de las pestañas que contiene el acordeón, se logra entrar a visualizar e interactuar directamente con la data o información que el sistema posee, permitiendo identificar cada uno de los parámetros que se evidencia en la gráfica deseada.



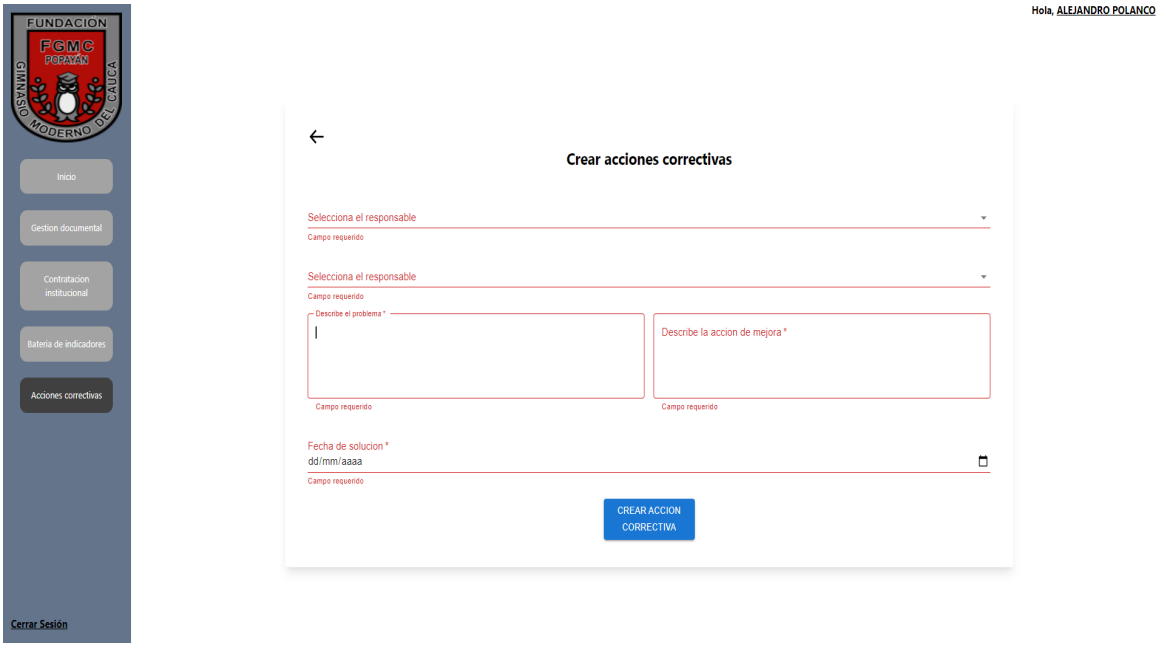
6.2 Acciones correctivas

El módulo de acciones correctivas, permite ser la herramienta empleada dentro de la Fundación Gimnasio Moderno del Cauca para realizar seguimiento a la calidad institucional, mostrándose como el módulo ideal para solucionar cualquier tipo de problemas de forma metódica y sistémica, sin tener que ejercer una vigilancia continua o realizar complejas tareas, ahorrando recursos económicos, humanos y tiempo, dado que en su desarrollo se implementaron métricas de seguimiento diario las cuales darán gran valor a la institución.

6.2.1 Validaciones

A continuación, se logra identificar todas las validaciones que presenta el software para que un usuario pueda crear una acción correctiva dentro del aplicativo, todos los campos son requeridos para así lograr exponer o plasmar la acción correctiva dentro de la tabla que permite evidenciar el número de acciones propuestas por el o los usuarios del sistema.

Se procede a mostrar la imagen con las respectivas validaciones que exige el aplicativo desarrollado para la Fundación Gimnasio Moderno del Cauca



Fuente: Elaboración propia presente estudio. Año 2022

6.2.2 Crear acción correctiva

En este apartado, se identifica el formulario que debe ser completado por cada usuario que tenga acceso o permisos a este módulo del sistema, el formulario

cuenta con los campos necesarios para detallar la alarma o el mal funcionamiento que se evidencia dentro de la Fundación Gimnasio Moderno del Cauca.

Para crear con éxito una acción correctiva, el usuario debe llenar los parámetros requeridos lo cuales son:

1. Seleccionar el responsable
2. Seleccionar la gravedad del problema: Alta, Media, Baja.
3. Describir el problema y seguido a ellos, describir la solución de mejora a esa falla
4. Asignar fecha de solución

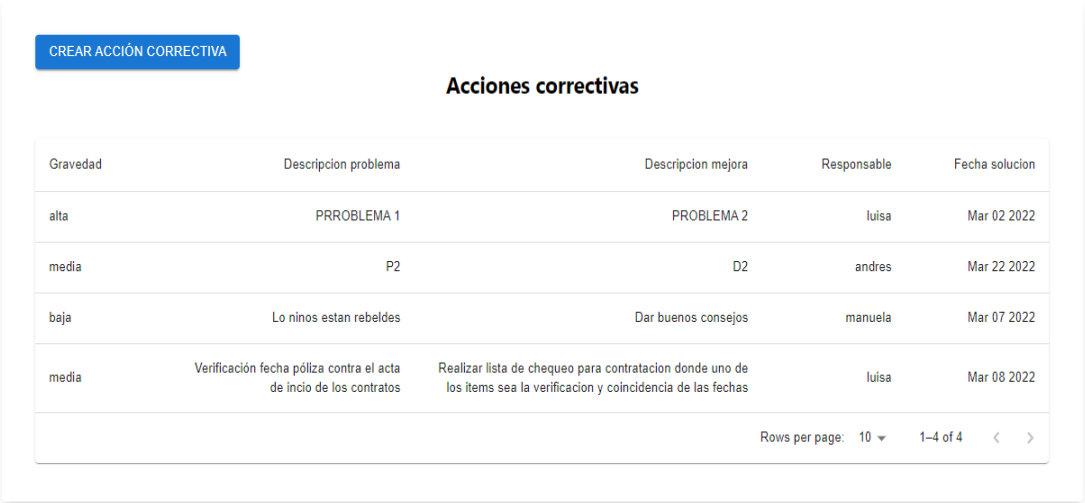
A continuación, se refleja lo expuesto anteriormente en la imagen siguiente:

Fuente: Elaboración propia presente estudio. Año 2022

6.2.3 Tabla de seguimiento

En esta vista del aplicativo, se puede evidenciar toda la información suministrada previamente en el paso anterior, en este apartado, se cargan y evidencian las acciones correctivas detectadas en la Fundación Gimnasio Moderno del Cauca.

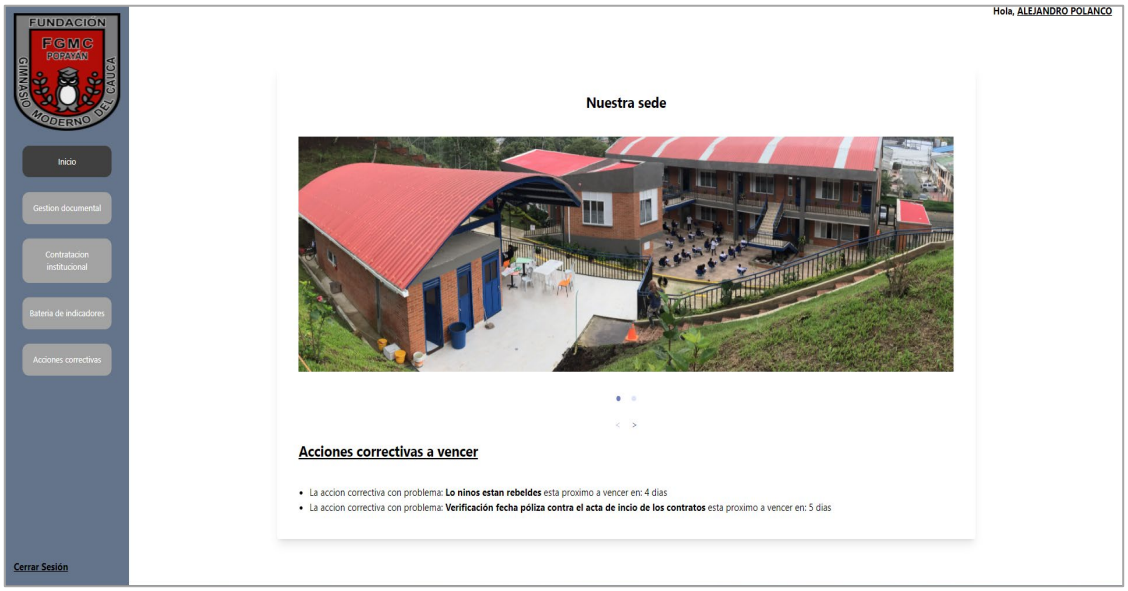
La vista permite paginar la tabla, esto quiere decir que en el momento que las acciones correctivas sean demasiadas, el software permite que el usuario tenga mejor manejo de la tabla en cuestión, logrando que pueda recorrerla de mejor manera, garantizando experiencia de usuario de calidad.



Fuente: Elaboración propia presente estudio. Año 2022

6.2.4 Reminder o recordatorio de acciones correctivas

La vista evidenciada en el módulo de inicio, conlleva una galería elaborada en scroll-image, la cual permite mostrar diversas fotos o videos de la Fundación Gimnasio Moderno del Cauca, seguido de un apartado donde permite mostrar las acciones correctivas que por su tiempo se encuentran próximas a vencer, siendo esto un recordatorio a todas las acciones correctivas que se elaboraron en el capitulo 6.2.2, se destaca que únicamente el software muestra las acciones correctivas que les falta 15 días para llegar a su fecha límite como se puede observar en la imagen a continuación.



Fuente: Elaboración propia presente estudio. Año 2022

Acciones correctivas a vencer

- La accion correctiva con problema: **Lo ninos estan rebeldes** esta proximo a vencer en: 4 dias
- La accion correctiva con problema: **Verificación fecha póliza contra el acta de incio de los contratos** esta proximo a vencer en: 5 dias

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

De acuerdo con los resultados obtenidos en el desarrollo de la pasantía mediante la cual se desarrolló el aplicativo para la Fundación Gimnasio Moderno del Cauca, se pueden sacar las siguientes conclusiones.

La implementación y uso del aplicativo desarrollado, permitió mejorar los procesos administrativos de contratación de la Fundación Gimnasio Moderno del cauca, evidenciándose en la disminución de tiempos, organización del archivo, facilidad de almacenamiento, seguridad de la información y facilidad de uso de la herramienta.

La posibilidad de utilizar tecnologías libres como lo es el stack MER (MongoDB. ExpressJs y React) permitió a la Fundación Gimnasio Moderno del Cauca, abaratar costos de implementación y dependencia de proveedores en la solución del problema que se presentaba.

La aplicación práctica de los conocimientos teóricos adquiridos en el desarrollo de la carrea de ingeniería de sistemas informáticos en temáticas como: bases de datos, buenas prácticas de programación, pruebas unitarias y funcionales, levantamiento de requerimientos, calidad del software, maquetación de aplicaciones, programación orientada a objetos, metodologías agiles como SCRUM, entre otras, permitieron fortalecer el perfil profesional, debido a la aplicación práctica de conocimientos en la resolución de un problema real.

Por las necesidades surgidas para el desarrollo del proyecto y el análisis de sus posibles soluciones, se pudo recrear situaciones de la vida real en la resolución de problemas que obligaron a adoptar conocimientos y plataformas tecnológicas

novedosas no tratadas durante el proceso formativo que ayudaron a actualizar y enriquecer los conocimientos adquiridos por el estudiante.

La aplicación de metodologías agiles como SCRUM, permiten optimizar los tiempos de desarrollo y fortalecer competencias como el trabajo en equipo, optimización, cumplimiento del cronograma, responsabilidad y liderazgo las cuales sin ser técnicas son esenciales para el desarrollo y logro de los objetivos planteados para el proyecto.

Por la arquitectura implantada y las tecnologías utilizadas para el desarrollo del proyecto y solución del problema, la aplicación es de tipo escalable, lo cual permitirá a futuro anexar funcionalidades o módulos adicionales que permitan robustecer los procesos administrativos a la Fundación Gimnasio Moderno del Cauca.

El despliegue de la aplicación en diferentes dispositivos como portátiles, computadores de mesa, tabletas, celulares, permite a los directivos de la fundación Gimnasio Moderno del Cauca, acceso a la información en tiempo real, lo cual facilita el desarrollo de sus actividades diarias y contribuye con el mejoramiento de la competitividad institucional.

La implantación del aplicativo desarrollado en la Fundación Gimnasio Moderno del Cauca, contribuye con el proceso de transformación digital institucional emprendido, por cuanto obliga a la adopción de nuevas tecnologías, actualización digital de sus empleados y mejoramiento de los procesos administrativos fundamentados en tecnologías de la información y la comunicación TIC.

Por la variedad y aparición constante de tecnologías aplicadas en el desarrollo de software, el talento humano que se desempeña en este sector, debe estar en aprendizaje permanente y adaptarse rápidamente a los cambios, para ser competitivos en el desarrollo de soluciones actualizadas a los clientes.

Se recomienda a los usuarios del aplicativo aplicar el manual de usuario acorde a lo establecido y orientado en la capacitación para evitar reprocesos que dificulten el desarrollo administrativo de la Fundación Gimnasio Moderno del Cauca.

Es recomendable que la institución tenga una copia de respaldo del código fuente de la aplicación y evite la manipulación del mismo por personal ajeno a los procesos para evitar un daños o malos funcionamientos del sistema.

RERERENCIAS

[1] Fundación Gimnasio Moderno del Cauca FGMC. (2021). 38 años. Educación de calidad. Información de primera fuente. Disponible en: https://www.fgmc.edu.co .

[2] ORH. Observatorio de Recursos Humanos. (2020). Por qué la automatización de procesos impulsa la transformación digital en las empresas. Disponible en: https://www.observatoriorh.com/actualidad/por-que-la-automatizacion-de-procesos-impulsa-la-transformacion-digital-de-las-empresas.html

[3] TICPYMS. (2020). Emprender, innovar, triunfar. Por qué la automatización de procesos impulsa la transformación digital de las empresas. Disponible en: https://www.ticpymes.es

[3] Información interna con directivos y personal área administrativa de la Fundación Gimnasio Moderno del Cauca FGMC. (2021).

[4] Globalbit. (2019). *El alcance del software en el mundo actual y su impacto en el futuro*. Disponible en: http://www.globalbit.co/2019/07/20/el-alcance-del-software-en-el-mundo-actual-y-su-impacto-en-el-futuro/

[5] OpenDocMan [DMS] Sistema de Gestión de documentos de código abierto. Disponible en: https://www.opendocman.com

[6] OpenKM. Open Document Management System S.L. (2019). Gestión Documental. Disponible en: https://www.openkm.com/es/

[7] OnlyOffice. 7.0. Released. (2020). Disponible **https://www.onlyoffice.com/es/**

[8] Britrix 24 (2021). Sistema para Gestión Documental. Disponible en: https://www.bitrix24.es

[9] Comforce. (2019). Software de Gestión de Contratos Disponible en:: https://www.comforce.co/

[10] Zoho Recruit. (2018). *Software de contratación en línea para reclutadores*. Disponible en: https://www.zoho.com/es-xl/recruit/recruiting-software.html

[11] Kuntz, F. (2019). ContractBook, Making Contracts Easy, Efficient & Automated. Disponible en: https://contractbook.com/

[12] VertexGroup. (2018). *La manera más fácil y rápida de hacer contratos*. Disponible en: https://www.ltvertexmx.com/

[13] Atlassian CI/CD (2019). By Max Rehkopf. Herramientas de Integración continua Disponible en: https://www.atlassian.com/es/continuous-delivery/continuous-integration/tools

[14] The GitHub Blog. Udpates, ideas, and inspiration from GitHub. Disponible en: https://github.blog/. https://www.freepng.es/png-smvi1g/

[15] Izertis (2016. Automatización de pruebas de una API usando POSTMAN. Disponible en: https://www.izertis.com

[16] ENCORA. Blog. Disponible en: https://www.encora.com/es/blog/como-realizar-pruebas-automatizadas-con-postman

[17] Hostgator. Desarrollo web. Tendencias tecnológicas. Clean Code. Código limpio. Disponible en: https://www.hostgator.mx

[18] Norma ISO /IEC. 9126: 2001. Características de la calidad de Software. Disponible en: https://www.verity.cl

[19] Martínez. P. Christian. Devexperto. Arquitectura de Software y sus beneficios. Disponible en: https://devexperto.com

[20] Genbeta: dev. (2017). Principios de una arquitectura limpia mantenible y Testeable. Disponible en: https://www.genbeta.com/desarrollo/principios-de-una-arquitectura-limpia-mantenible-y-testeable.

[21] Loadview. By dotcom.monitor. (2020).Tipos de pruebas de Software, diferencias y ejemplos. Disponible en: https://www.loadview-testing.com.

[22] 54 cuatro (2020). Patrones de Arquitectura para Microservicios. Disponible en: https://go.54cuatro.com

[23] REACT. Desarrollo web.Framework o Librería. Disponible en: https://desarrolloweb.com

[24] Wam, Global Growth Agents. Abellán, Encarna. (2020). Scrum. Qué es y cómo funciona esta metodología. Disponible en línea: https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html

[25] Mongo DB Atlas Database- Cloud DBaaS For MongoDB. Disponible en: https://aws.amazon.com/es/quickstart/architecture/mongodb-atlas/#:~:text=MongoDB%20Atlas%20es%20un%20servicio,nube%20para%20las%20aplicaciones%20modernas.

[26] Bcrypt. Node. Bcrypt. Js.Disponible: https://www.npmjs.com

[27] Envato Tuts. Una introducción a Mongoose para Mongo DB y Node. Js. Disponible en: https://code.tutsplus.com/es/articles/an-introduction-to-mongoose-for-mongodb-and-nodejs--cms-29527

[28] Digital Guide IONOS. Introducción a JSON Web Token (JWT). Disponible en: https://www.ionos.es

[29] WaaS. (2018). Ventajas y desventajas de la plataforma como servicio (PAAS). Disponible en: https://www.waas.com.co/novedades-cloud-computing/ventajas-desventajas-la-plataforma-servicio-paas/#:~:text=Anteriormente%20PAAS%20contaba%20con%20ciertas,un%20claro%20inconveniente%20para%20no