



**UNIVERSIDAD TÉCNICA DE BABAHOYO  
FACULTAD DE ADMINISTRACIÓN, FINANZAS E INFORMÁTICA**

**PROCESO DE TITULACIÓN**

**NOVIEMBRE 2020 – MAYO 2021**

**EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA**

**PRUEBA PRÁCTICA**

**INGENIERÍA EN SISTEMAS**

**PREVIO A LA OBTENCION DEL TITULO DE INGENIERO(A) EN SISTEMAS**

**TEMA:**

**ANÁLISIS DEL STACK MERN Y SU USO EN LAS APLICACIONES WEB  
BASADAS EN SERVICIOS REST.**

**EGRESADA(O):**

**STEEVEN JOSSUEPH NARANJO HEREDIA**

**TUTOR:**

**ING. ERICK MAGNO RICAURTE ZAMBRANO, MSIG, MBA**

**AÑO 2021**

## INTRODUCCIÓN

El presente caso de estudio tiene como título análisis del stack MERN y su uso en las aplicaciones web basadas en servicios REST, su objetivo es analizar el stack MERN y realizar pruebas de rendimiento a una aplicación desarrollada utilizando el conjunto de herramientas que lo componen, para determinar sus principales características, ventajas y desventajas.

Con el paso de los años las empresas han migrado sus servicios a la web, estas buscan aplicaciones escalables que con el paso de los años permitan agregar más funciones, haciendo que el desarrollo de aplicaciones web sea una tarea complicada, es por eso que en la actualidad el trabajo es distribuido para cada una de las partes de la aplicación, es decir un programador se encarga del backend, otro de la base de datos y otro del frontend. Cuando un programador realiza una aplicación web, ya sea para una empresa, blog, o un proyecto personal, debe elegir un stack adecuado que se ajuste a los requerimientos de la aplicación. La elección no es sencilla, ya que se deben considerar varios aspectos, como sistema operativo, base de datos o intérprete de órdenes. El stack MERN es un conjunto de herramientas, que está compuesto por MongoDB, ExpressJS, ReactJS y NodeJS, de allí su nombre. Todas estas herramientas trabajan con el lenguaje de programación JavaScript lo cual supone una ventaja ya que se puede aplicar la misma lógica a toda la aplicación.

Para esta investigación se utilizó la metodología documental y experimental, misma que consistió en realizar indagaciones por internet, consultas bibliográficas sobre las bases teóricas del stack MERN y su uso en aplicaciones web, además se empleó como técnica de investigación la observación ya que al no existir una herramienta que nos permita analizar cada una de los frameworks que conforman el stack se optó por el

desarrollo de una aplicación web que cumpla con las funciones básicas, creada con el stack MERN a la cual se realizó pruebas de rendimiento para así determinar si es recomendable el desarrollo de una aplicación web utilizando este conjunto de herramientas y además con una guía de observación se pudo determinar sus principales ventajas y desventajas.

La línea de investigación presente en este proyecto está encaminada al desarrollo de sistemas de información y comunicación, emprendimiento e innovación correspondiente a la sub línea redes y tecnologías inteligentes de software y hardware.

## DESARROLLO

Los servicios web son aquellos que permiten la comunicación entre el usuario y la máquina, uno de los servicios web más usados es REST, este servicio dio paso a nuevos stacks de desarrollo, siendo el stack MERN uno de los más utilizados en la actualidad, es por eso que el presente caso de estudio tiene como objetivo analizar del stack MERN en una aplicación web basada en servicios REST.

Desde la aparición de las tecnologías web, estas han formado parte de nuestra vida, ya que diariamente visitamos páginas e interactuamos con ellas. Motivando así a las empresas o pequeños negocios a ubicar sus servicios en la web, ya que como indica (Alexandre, 2020) “Muchas empresas que no ofrecían soluciones digitales han tenido que adaptarse a esta nueva realidad. Aquellas que ya lo hacían, junto con las que nacieron en este contexto, vieron un notable aumento del volumen de datos”.

Cuando un programador realiza una aplicación web desde cero, ya sea para una empresa, blog, o un proyecto personal. Es necesario definir de forma correcta y detallada los requerimientos, entender la aplicación a desarrollar, su funcionalidad y escalabilidad, es por eso que se debe elegir un stack adecuado para el desarrollo de la aplicación que se ajuste a sus requerimientos. La elección no es sencilla, ya que se deben considerar varios aspectos.

Con el paso de los años han ido surgiendo diferentes stacks para desarrollar aplicaciones web, actualmente el más conocido es el stack MERN, que se presenta como un conjunto de herramientas para el desarrollo de una aplicación en todos sus niveles utilizando como lenguaje de programación o interprete de ordenes JavaScript.

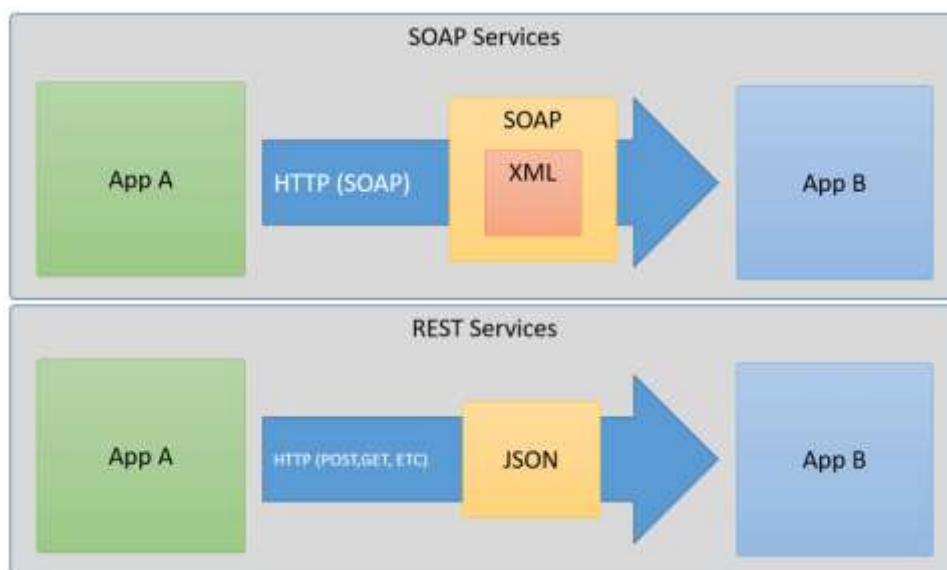
La investigación presente en este proyecto es de tipo Documental ya que va realizar indagaciones por internet, consultas bibliográficas sobre las bases teóricas sobre los servicios REST, además también se consultará teoría sobre el stack MERN, el conjunto de herramientas que lo componen y su uso en aplicaciones web.

Esta investigación también es Experimental, debido a que al no existir una herramienta que nos permita analizar cada uno de los frameworks que conforman el stack MERN se optó por el desarrollo de una aplicación web que cumpla con las funciones básicas, creada con el stack MERN a la cual se realizó pruebas de rendimiento, y se realizaron cálculos de los resultados obtenidos para así poder determinar si es recomendable considerar el desarrollo de una aplicación web y además con una guía de observación determine sus principales ventajas y desventajas.

Por último, esta investigación tiene un enfoque Cuantitativo, ya que al obtener los datos de las pruebas de rendimiento estos fueron examinados de forma matemática para evidenciar cual es el comportamiento y la capacidad que tiene este conjunto de herramientas.

Para hablar del stack MERN partiremos de la base que son los servicios web, estos son un elemento muy importante en la integración de aplicaciones de diferentes plataformas, además estos servicios han facilitado la comunicación entre el usuario y el servidor intensificando su uso con el tiempo. La empresa IONOS da una visión sobre los servicios web y dice lo siguiente: “A día de hoy, usar diferentes servicios a través de la web es una actividad habitual. Comprar online, leer el periódico, reservar una mesa en un restaurante o ver películas son solo algunos ejemplos de las muchas interacciones que se producen a diario entre el usuario y la máquina” (IONOS by 1&1, 2020).

Con esto podemos decir que cada interacción que realizamos en una aplicación web estamos haciendo uso de un servicio web, debido a que por cada acción que ejecutemos, estamos realizando una petición al servidor y este es encargado de procesarla para luego responderla. Cuando se crearon los servicios web la mayor parte funcionaban con el servicio SOAP. IBM lo define como “Un formato de mensaje XML utilizado en interacciones de servicios web. Los mensajes SOAP habitualmente se envían sobre HTTP” (IBM, 2019). Es decir que es un protocolo que permite el intercambio de información entre el usuario y el servidor.



**Figura 1** Servicio SOAP y Servicio REST

**Fuente:** <https://www.oscarblancarteblog.com/2017/03/06/soap-vs-rest-2/>

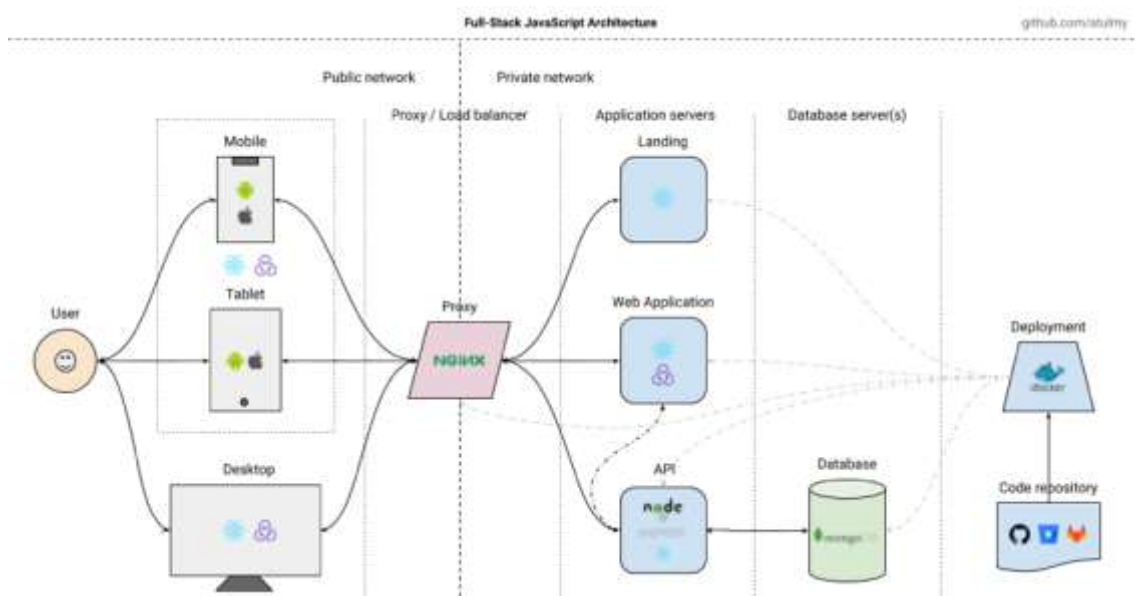
Como se lo puede observar en la *figura 1*, SOAP es muy complejo de implementar, así como también es complicado de gestionar, como solución a esto nace el servicio REST que es uno de los más utilizados en la actualidad, que como en la imagen se puede apreciar tiene menos restricciones.

REST fue propuesto por Roy Fielding en el año 2000, su significado es REpresentational State Transfer o traducido al español Transferencia de estado representacional, se puede definir al servicio REST como un conjunto de restricciones o

normas que se deben aplicar al diseño de una aplicación web. Este servicio posee los siguientes principios:

- Debe tener la arquitectura cliente-servidor.
- Debe ser Stateless es decir sin estado.
- Los datos pueden ser almacenados en caché.
- El sistema debe estar basado en capas.
- Debe poseer una interfaz uniforme
- Puede enviar código OnDemand es decir según el cliente lo solicite.

Además de esto el autor (Moncayo, 2018) define al servicio REST como “Una interfaz para conectar varios sistemas basados en el protocolo HTTP y nos sirve para obtener y generar datos y operaciones, y a partir de esto devuelve los datos en formato JSON”.



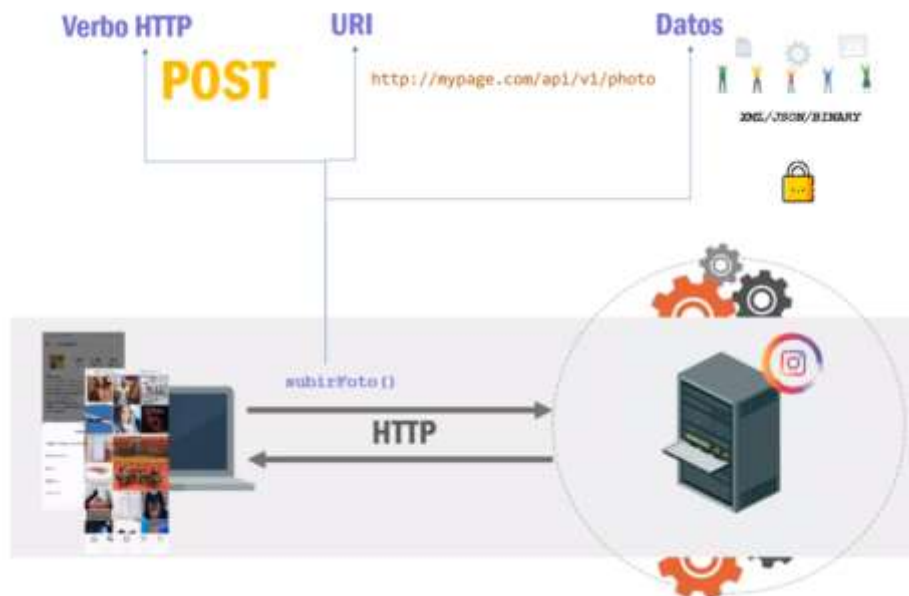
**Figura 2** Arquitectura REST con JavaScript

**Fuente:** <https://github.com/atulmy/fullstack-javascript-architecture>

En la *figura 2* se puede observar el funcionamiento de REST con una aplicación que utiliza como lenguaje de programación JavaScript como podemos ver esta

arquitectura puede ser aplicada para las diferentes plataformas ya sea escritorio, tablet o móvil, el backend o lógica de la aplicación es desarrollada como una API REST y esta es la encargada de interactuar con la base de datos y responder a las peticiones realizadas por el usuario.

Una API ya establecida que sigue las reglas establecidas por el servicio REST se le conoce como RESTful API. En la siguiente figura podemos observar el funcionamiento de una aplicación RESTful API.



**Figura 3** Funcionamiento de una RESTful API  
**Fuente:** Steeven Naranjo Heredia

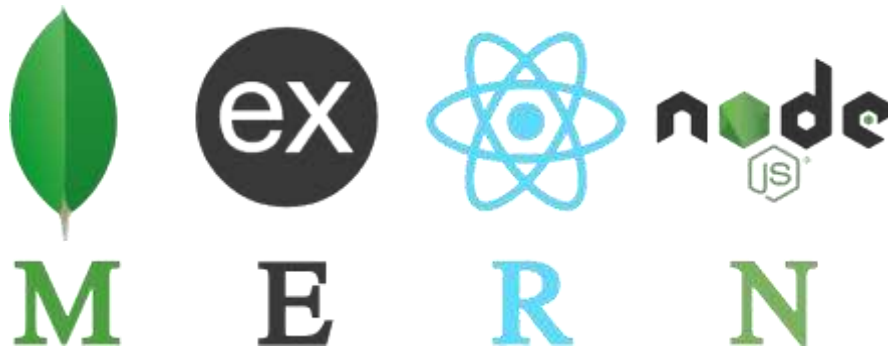
Algunas empresas tales como Facebook, Twitter, Google y muchas más tienen sus propias API REST, ya que esto permite dividir la parte lógica o backend del frontend de la aplicación, haciendo que el desarrollo sea más rápido y fluido, ya que los programadores se enfocan únicamente en la parte lógica, este servicio es muy utilizado para aplicaciones multiplataforma, ya que podemos aplicar el mismo backend para cualquier aplicación de diferente plataforma.



La aparición del servicio REST dio paso a la creación de los llamados stack de desarrollo que pueden ser definidos como un conjunto de herramientas que permiten realizar aplicaciones web que se ajusten a los requerimientos establecidos por las empresas o clientes, la traducción de la palabra stack al español es pila, que hace referencia a agrupar un conjunto de herramientas o frameworks que nos faciliten la tarea del desarrollo o nos den la solución a una problemática en específico.

La empresa (1&1 IONOS España S.L.U., 2016) define los stacks de desarrollo como, “Los componentes de software de un web stack pueden combinarse de forma individual, por lo que existe un gran número de versiones, cada una para diferentes fines.”, entonces se puede decir que un stack no es creado como tal, sino que este puede ser creado a partir de las necesidades de cada programador al momento de crear una aplicación web, se debe tener en cuenta que un stack está conformado 4 componentes básicos y necesarios para el desarrollo de una aplicación web que son: Sistema operativo, servidor web, base de datos, intérprete de órdenes.

Uno de los stacks de desarrollo que ha ido tomando mayor importancia con el paso de los años gracias al crecimiento de JavaScript en el desarrollo web es el stack MERN, que es definido como “Un conjunto de tecnologías utilizadas para el desarrollo web de aplicaciones que consta de MongoDB, React JS, Express JS y Node JS como sus componentes” (Parada, 2020). Este conjunto de herramientas son opensource es decir de código abierto, además tiene una gran comunidad que lo respalda y cada una de las herramientas mencionadas nos permite trabajar con JavaScript como intérprete de órdenes.



**Figura 4** Herramientas del stack MERN

*Fuente:* <https://academicwritingforstudents.com/mern-stack/>

Es decir que gracias a este conjunto de herramientas podemos crear aplicaciones web desde el backend y el frontend, utilizando como único lenguaje de programación JavaScript, lo que supone una ventaja ya que permite trabajar en cualquier parte de la aplicación usando la misma lógica y sin preocuparnos en las herramientas que se debe utilizar.

Las principales características del stack MERN son las siguientes:

- Permite construir una aplicación entera con JavaScript
- Cuenta con una gran comunidad
- Eficiencia de MongoDB en el mantenimiento y la organización de datos
- Eficiencia de NodeJS

MERN es un stack se acopla bien a todo tipo de proyectos o aplicaciones, como ya se mencionó, en la actualidad la arquitectura más utilizada es REST. El backend de la aplicación web es desarrollado como un servicio REST, respondiendo peticiones realizadas por el usuario entregando los resultados en formato JSON al cliente. Este conjunto de datos es consumido desde el frontend a través de ReactJS. Esta arquitectura permite producir aplicaciones web más sencillas de realizar además de hacerlas más ligeras y rápidas, asemejándose a la experiencia que nos daría una aplicación de escritorio.

Entonces se puede decir que el stack MERN es un tipo de arquitectura REST que permite el modularidad de un proyecto, lo que facilita el mantenimiento de la aplicación.

## **MongoDB**

“Es una base de datos de documentos de código abierto utilizada en muchas aplicaciones web modernas. Se clasifica como una base de datos NoSQL porque no se basa en una estructura de base de datos relacional basada en tablas tradicional, en su lugar, Mongo utiliza documentos similares a JSON con esquemas dinámicos.” (Drake, 2021). Entonces podemos decir que MongoDB a diferencia de las bases de datos relacionales no requieren un esquema para poder empezar a agregar datos.

## **Express**

Aarón López Sosa define a Express como “Un framework rápido, minimalista y flexible de Node.js. Permite crear APIs y aplicaciones web fácilmente, provee un conjunto de características como manejo de rutas, archivos estáticos, uso de motor de plantillas, integración con bases de datos, manejo de errores, middlewares entre otras” (Sosa, 2019). Cabe recalcar que NodeJS puede funcionar solo, sin embargo, al momento de realizar una aplicación se debe construir la estructura y este proceso se deberá repetir en cada proyecto a realizar, además esta es una tarea muy complicada.

## **ReactJS**

La empresa NextU define a ReactJS como “Una biblioteca escrita en JavaScript, que facilita la creación de componentes interactivos, reutilizables, para interfaces de usuario. Uno de sus puntos más destacados, es que no sólo se utiliza en el lado del cliente, sino que también se puede representar en el servidor, y trabajar juntos” (NextU, 2017). React.js es una biblioteca que nos permite crear aplicaciones dinámicas del frontend de la

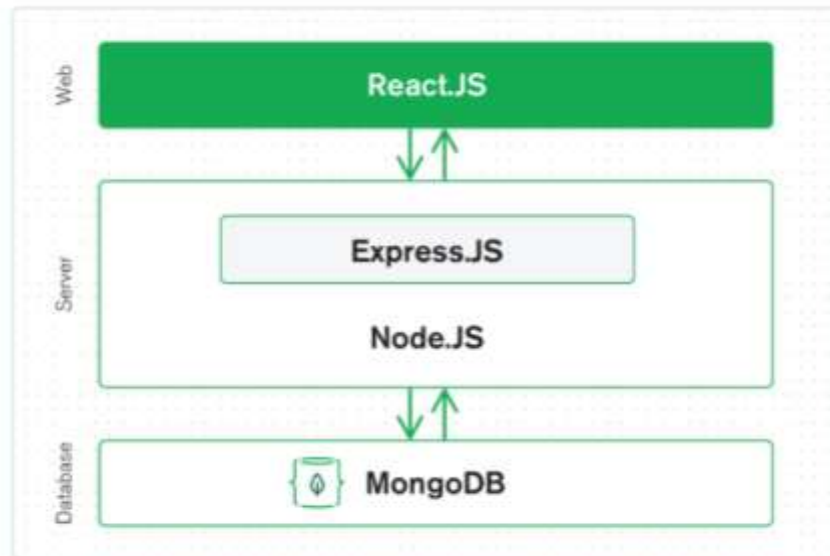
aplicación web, es la que interactúa con el cliente ya que este se encarga de enviar las peticiones al servidor y presentar los resultados en formato HTML.

## **NodeJS**

Jesús Lucas nos da una definición de Node.js e indica lo siguiente “Es un entorno de tiempo de ejecución de JavaScript. Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript. También aporta muchos beneficios y soluciona muchos problemas” (Lucas, 2019). Con esta definición de Node.js se podría decir que a diferencia de los demás frameworks de JavaScript existentes, fue pensado para crear servidores web y la ejecución de funciones en tiempo real, además tiene un manejador de paquetes NPM que funciona en base a módulos que pueden agregados por el programador a la ejecución de la aplicación.

## **Funcionamiento del stack MERN**

La *figura 5* muestra de manera simple el funcionamiento de una aplicación web desarrollada con el stack MERN y las funciones de cada una de sus herramientas, en la misma se pudo identificar 6 pasos al momento que se realiza una petición al servidor de la aplicación web.

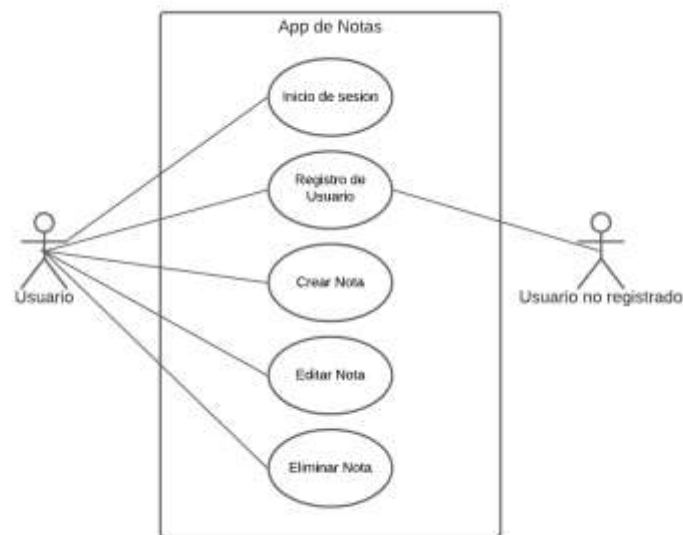


**Figura 5** Funcionamiento del stack MERN  
**Fuente:** <https://www.mongodb.com/mern-stack>

1. El usuario envía una petición al servidor a través de la aplicación web, es decir que este interactúa desde el frontend mediante ReactJS.
2. Node.js recibe la petición del cliente, la analiza y la maneja.
3. Express es el encargado de enviar la petición a MongoDB.
4. Lo mismo pasa cuando Express recibe los datos de MongoDB es el encargado de devolver la respuesta.
5. Al igual que Express, Node.js recibe la respuesta y es el encargado de devolver la respuesta de la petición hecha por el cliente al frontend de la aplicación web.
6. Por último, ReactJS recibe la respuesta y la presenta al cliente.

Debido a que no existe una herramienta que permita analizar cada uno de los componentes que conforman el stack MERN se optó por desarrollar una aplicación web que cumpla con las funcionalidades básicas y así con esto se pudo evidenciar la complejidad de utilizar este conjunto de herramientas y mediante la ejecución de pruebas de rendimiento y la elaboración de una guía de observación pudimos determinar si es recomendable utilizarla y se definió sus principales ventajas y desventajas.

La aplicación consiste en la creación de notas, pero además de esto el usuario podrá registrarse e iniciar sesión en la misma, para tener acceso a las notas que haya guardado en su cuenta. Esta aplicación posee características básicas, es decir permite el inicio de sesión y registro de usuarios, además también permite crear, modificar y eliminar notas, con la peculiaridad que toda la aplicación tanto en el backend como en el frontend fue desarrollada únicamente con frameworks de JavaScript y la base de datos fue diseñada con MongoDB, que son el conjunto de herramientas que conforman el stack MERN.



**Figura 6** Diagrama de caso de uso de la aplicación web

**Fuente:** Steeven Jossueph Naranjo Heredia

En la *figura 6* podemos observar el diagrama de caso de uso de la aplicación web, en el que se detalla las actividades que podrá realizar los usuarios, por ejemplo, un usuario puede registrarse, iniciar sesión y luego administrar notas, mientras que un usuario que no esté registrado en la aplicación únicamente tendrá acceso a la página de registro.

Entre los requerimientos funcionales de la aplicación web tenemos los siguientes:

- Inicio de sesión
- Registro de usuario

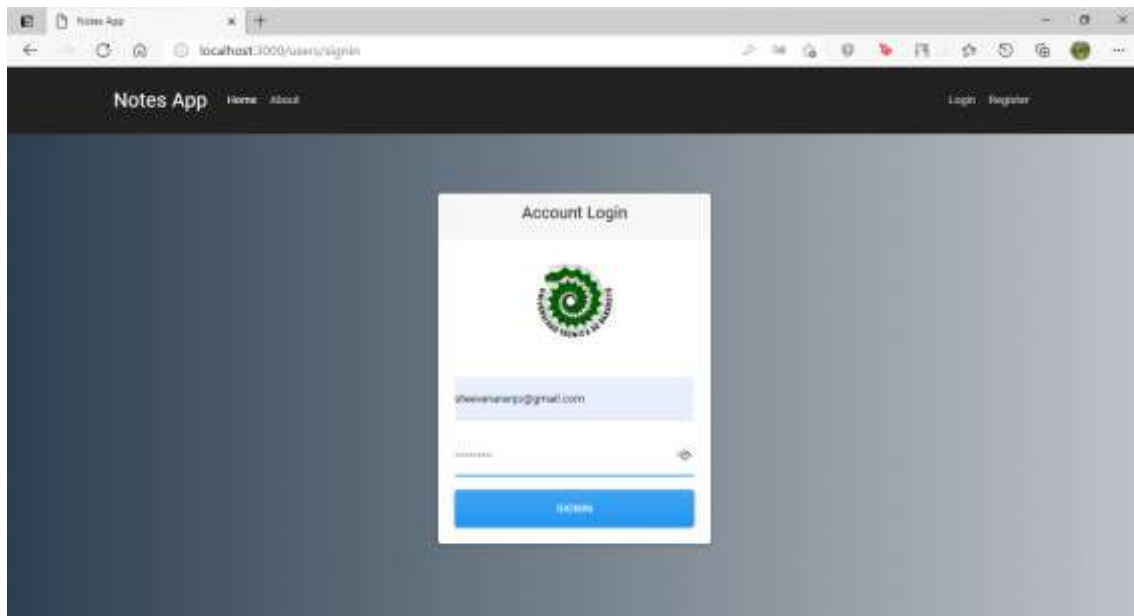
- Crear notas
- Editar notas
- Eliminar notas

<b>Número de requerimiento:</b>	RF01
<b>Nombre del requerimiento:</b>	Autenticación del usuario
<b>Tipo:</b>	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
<b>Característica:</b>	Para poder agregar notas en la aplicación el usuario primero tendrá que iniciar sesión.
<b>Descripción:</b>	El sistema tendrá que validar los datos ingresados por el usuario y permitir el ingreso a ver notas, caso contrario el sistema redireccionará al usuario a la ventana de registro.
<b>Prioridad del requerimiento:</b>	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

**Tabla 1:** Requerimiento funcional para inicio de sesión  
**Fuente:** Steeven Jossueph Naranjo Heredia **Ver ANEXO 2**

Mientras que los requerimientos no funcionales de la aplicación son los siguientes:

- Sistema con interfaz minimalista
- Diseño accesible o de fácil manipulación
- El sistema confirma todas las acciones realizadas por el usuario
- Sistema escalable



**Figura 7** Interfaz de la aplicación web

**Fuente:** Steeven Jossueph Naranjo Heredia Ver ANEXO 3

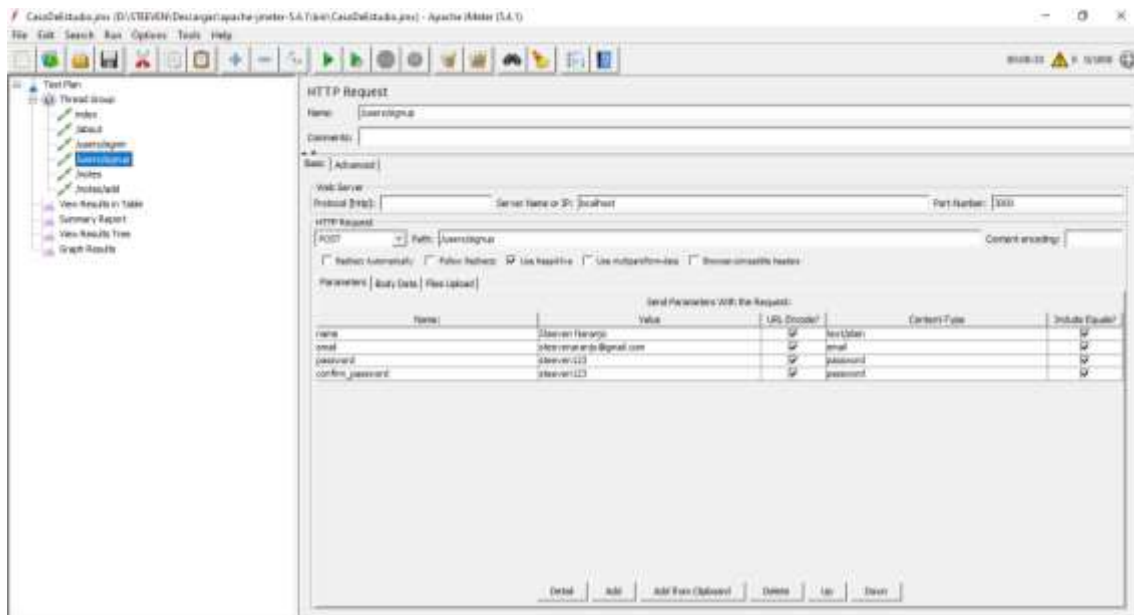
## Pruebas realizadas

Para verificar el comportamiento de la aplicación y las herramientas que conforman el stack MERN se tomó como referencia el estudio realizado por Marcelo Sánchez en el (2019). Se realizaron pruebas de rendimiento a la aplicación web desarrollada con el stack MERN, para esto se utilizó la herramienta de testing llamada Apache JMeter que permite realizar peticiones al servidor para poder evidenciar los tiempos de respuesta, el funcionamiento de la aplicación, observar errores presentes en la aplicación web y cuando se obtenga los resultados de las pruebas de rendimiento, serán analizados de acuerdo a la norma ISO/IEC 9126 para la Calidad del Software en las métricas presentes en la característica de Eficiencia.

Para la ejecución de las pruebas realicé un script en la herramienta ya mencionada que simule la navegación de un usuario normal, interactuando con cada una de las páginas de la aplicación, luego para someter a carga el servidor y las herramientas del stack MERN se realizó la prueba con 1000 usuarios realizando peticiones al servidor. Para ver



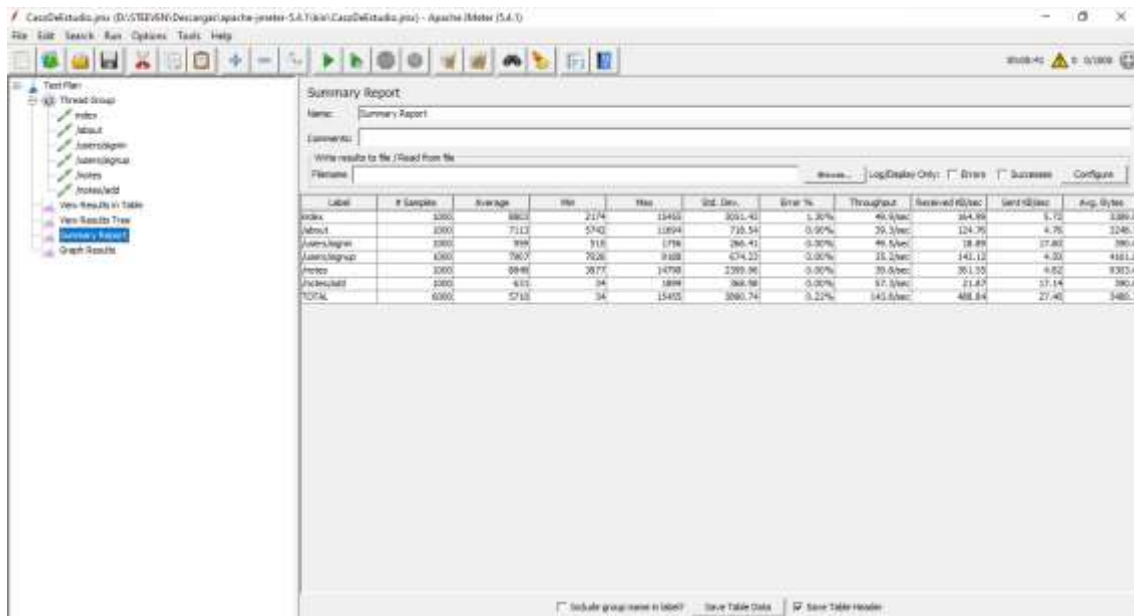
los resultados JMeter nos proporciona funciones que nos permite visualizar los resultados para su posterior análisis.



**Figura 4** Interfaz de JMeter con petición de inicio de sesión al servidor

**Fuente:** Steven Jossueph Naranjo Heredia

Para poder visualizar los resultados de la prueba de rendimiento se agregó un Summary Report o Informe Resumido que nos devuelve todos los resultados de las pruebas realizados en forma de tabla y además se agregó un Graph Results o Gráfico de resultados que nos permiten ver las estadísticas de los tiempos totales de respuestas a las peticiones realizadas para poder llevar a cabo el análisis.



**Figura 8** Resultados de las pruebas realizadas  
**Fuente:** Steeven Jossueph Naranjo Heredia

Al revisar la tabla de resultados de la prueba de rendimiento encontramos que al sobrecargar el servidor se presentó un Error del 0.22% en el total de peticiones realizadas al servidor. Del total de peticiones, podemos ver la cantidad de peticiones que presentaron errores, calculando el porcentaje de error obtenido en la prueba con el total de peticiones realizadas. Una vez que se obtuvieron los resultados se procedió a realizar el cálculo para determinar la confiabilidad de la aplicación en base a la eficiencia y según la norma ISO/IEC 9126 en donde nos indica que:

$$X = 1 - A/N$$

En donde:

- X = Confiabilidad de la aplicación.
- A = Cantidad de veces que ocurrió un error en la respuesta de la petición
- N = Cantidad de peticiones realizadas para generar un error.

Reemplazamos con los resultados obtenidos y decimos que

$$X = 1 - (132/6000)$$

$$X = 1 - (0.022)$$

$$X = 0.978$$

Con los resultados obtenidos de la prueba podemos llegar a la conclusión que una aplicación web básica desarrollada con el conjunto de herramientas que conforman el stack MERN es un 97,8% confiable, bajo las condiciones descritas.

Luego de obtener los resultados de las pruebas y la experiencia de haber realizado una aplicación web con el stack MERN y mediante una guía de observación en la que se anotaron ciertos aspectos al momento de desarrollar la aplicación se pudo determinar las ventajas de utilizar este stack, las cuales son las siguientes:

- El stack MERN abarca todo el desarrollo de la aplicación web.
- Podemos desarrollar toda la aplicación con el lenguaje de programación JavaScript que es uno de los más utilizados en la programación web.
- El stack de desarrollo MERN es escalable, modular y completo.
- Tiene buenos tiempos de respuesta cuando es sometido a la sobrecarga.
- Todas las herramientas o frameworks que lo componen son de código abierto.
- Este stack de desarrollo está respaldado por una gran comunidad.

En su estudio sobre el stack MERN Antonio Morales nos dice que “Los beneficios del stack MERN provienen de la robustez de Node, ya que este nos proporciona su API abierta en tiempo real” (Morales, 2015). Además de esto también nos dice que podemos usar este stack para aplicaciones que requieran transferir datos, chats, actualización de datos, o cualquier otra aplicación que requiera presentar datos en tiempo real, entre las cuales tenemos:

- Chat
- Una Red Social
- RSS feed

- Tienda online

Así mismo como todas las aplicaciones, el stack MERN puede usarse en ciertos proyectos que requieran información en tiempo real, sin embargo, hay que tener en cuenta que este presenta ciertos inconvenientes al momento de utilizarlo y son los siguientes:

- MongoDB guarda grandes cantidades de datos desnormalizados.
- Node.js tiene un bajo rendimiento en caso de consultas con uso intensivo de CPU.
- El stack tiene un bajo rendimiento con bases de datos relacionales.
- Complejidad de las herramientas que componen el stack MERN.
- Requiere de un alto grado de aprendizaje que requiere.

## CONCLUSIONES

Luego de haber analizado el stack MERN y su uso en las aplicaciones web basadas en servicios REST se concluyó lo siguiente:

- En base al desarrollo de una aplicación web y a la guía de observación en la que se registró aspectos importantes al momento del desarrollo de la aplicación, se determinó que el stack MERN posee las características y ventajas suficientes para poder ser tomado en cuenta al momento de desarrollar una aplicación web basada en servicios REST como lo son: su escalabilidad, modularidad y flexibilidad, sin embargo, posee ciertas desventajas, que deben ser consideradas como la dificultad del uso de sus componentes y el alto grado de aprendizaje que requiere.
- Aplicando las métricas de características de Eficiencia descritas en la norma ISO/IEC 9126 para la Calidad del Software, se consiguió como resultado que una aplicación web basada en REST desarrollada con el stack MERN presenta un buen porcentaje de confiabilidad al obtener un 97.8% en las pruebas de rendimiento.
- En base al análisis de los resultados obtenidos de las pruebas de rendimiento realizadas utilizando la herramienta Apache JMeter se pudo determinar que una aplicación web desarrollada con el stack MERN basada en servicios REST presenta buenos tiempos de respuesta a las peticiones realizadas por los usuarios cuando este es sometido a condiciones complejas.

## BIBLIOGRAFÍA

1&1 IONOS España S.L.U. (17 de octubre de 2016). *Digital Guide IONOS*. Obtenido de <https://www.ionos.es/digitalguide/servidores/know-how/desarrollo-web-con-stacks-de-software/>

Academic Writing for Students. (9 de diciembre de 2020). *Academic Writing for Students*. Obtenido de <https://academicwritingforstudents.com/mern-stack/>

Alexandre, L. S. (11 de noviembre de 2020). *Hotmart*. Obtenido de <https://blog.hotmart.com/es/mercado-digital/>

Alvarez, M. A. (25 de febrero de 2019). *Qué es React. Por qué usar React*. Obtenido de <https://desarrolloweb.com/articulos/que-es-react-motivos-uso.html>

Blancarte, O. (6 de marzo de 2017). *SOAP vs REST*. Obtenido de <https://www.oscarblancarteblog.com/2017/03/06/soap-vs-rest-2/>

Drake, M. (4 de enero de 2021). *What is MongoDB?* Obtenido de <https://www.digitalocean.com/community/tutorials/what-is-mongodb>

IBM. (21 de julio de 2017). Obtenido de [https://www.ibm.com/support/knowledgecenter/SSMKHH\\_9.0.0/com.ibm.etools.mft.doc/ac55710\\_.htm](https://www.ibm.com/support/knowledgecenter/SSMKHH_9.0.0/com.ibm.etools.mft.doc/ac55710_.htm)

IBM. (20 de diciembre de 2019). *IBM Integration Bus*. Obtenido de [https://www.ibm.com/support/knowledgecenter/es/SSMKHH\\_10.0.0/com.ibm.etools.mft.doc/ac55770\\_.htm](https://www.ibm.com/support/knowledgecenter/es/SSMKHH_10.0.0/com.ibm.etools.mft.doc/ac55770_.htm)

IONOS by 1&1. (15 de abril de 2020). *Digital Guide IONOS*. Obtenido de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/web-services/>

Jaime Sayago Heredia, I. A. (07 de septiembre de 2018). *Análisis comparativo para aplicaciones web basados en servicios REST: stack MEAN y stack Java EE*.

Obtenido de

[https://www.researchgate.net/publication/328293361\\_Analisis\\_comparativo\\_para\\_aplicaciones\\_web\\_basados\\_en\\_servicios\\_REST\\_stack\\_MEAN\\_y\\_stack\\_Java\\_EE](https://www.researchgate.net/publication/328293361_Analisis_comparativo_para_aplicaciones_web_basados_en_servicios_REST_stack_MEAN_y_stack_Java_EE)

Lucas, J. (4 de septiembre de 2019). *Qué es NodeJS y para qué sirve*. Obtenido de <https://openwebinars.net/blog/que-es-nodejs/>

Moncayo, J. M. (7 de mayo de 2018). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-rest-conoce-su-potencia/#:~:text=REST%20es%20una%20interfaz%20para,espec%C3%ADfic os%2C%20como%20XML%20y%20JSON.>

MongoDB. (16 de noviembre de 2020). *What is the MERN Stack?* Obtenido de <https://www.mongodb.com/mern-stack>

Morales, A. (12 de enero de 2015). *FunnyFrontend*. Obtenido de <https://funnyfrontend.com/introduccion-stack-mean-parte-1/#comments>

NextU. (25 de agosto de 2017). *QUÉ ES Y CÓMO FUNCIONA REACT.JS*. Obtenido de <https://www.nextu.com/blog/que-es-y-como-funciona-react-js/>

Parada, M. (21 de octubre de 2020). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/mern-stack-que-es-y-que-ventajas-ofrece/>

Sanchez, M. (enero de 2019). **USO DEL STACK MEAN PARA EL DESARROLLO DE UNA APLICACIÓN WEB ORIENTADA A LA TECNOLOGIA BIG DATA**. Esmeraldas, Ecuador.

Sosa, A. L. (22 de mayo de 2019). *Medium*. Obtenido de <https://medium.com/@aarnlpezsosa/introducci%C3%B3n-a-express-js-a1ebe16dbcf4>

Yadav, A. (25 de febrero de 2021). *Git Hub*. Obtenido de <https://github.com/atulmy/fullstack-javascript-architecture>

# **ANEXOS**



## ANEXO 1

### Guía de observación

No.	ASPECTO A EVALUAR	SI	NO	OBSERVACIONES
1	Se puede desarrollar la aplicación completa usando el stack MERN	X		Se pudo desarrollar una aplicación web completa utilizando este stack.
2	Todas las herramientas son de código abierto	X		Las herramientas se encuentran disponibles de manera gratuita para cualquier persona que desee utilizarlas.
3	Este stack de desarrollo cuenta con una comunidad dispuesta a apoyar a las personas que se encuentran aprendiendo	X		En el desarrollo de la aplicación se presentaron inconvenientes en la compatibilidad de ciertas herramientas, al buscar los errores en la web la comunidad dio solución a estos problemas.
4	Una aplicación desarrollada con este conjunto de herramientas es escalable	X		La aplicación primero se desarrolló la administración de notas, luego se agregó el registro de usuarios de una manera sencilla.
5	La aplicación puede estar separada en módulos	X		En el desarrollo de la aplicación se separó la parte lógica o backend del frontend.
6	Las herramientas del stack MERN son capaces de funcionar sin otras herramientas que lo complementen		X	Este fue un problema que me encontré ya que para cada framework era necesario instalar librerías para su correcto funcionamiento o para facilitar el trabajo.
7	El diseño y construcción de la BD con MongoDB fue sencilla	X		La base de datos se crea automáticamente, al momento de desarrollar MongoDB verifica si la base de datos que se requiere existe si no es el caso, crea la base de datos.
8	El manejo de todas las herramientas es fácil		X	Aunque todas comparten el mismo lenguaje de programación es complejo manejar las herramientas por la cantidad de funciones que maneja.
9	La curva de aprendizaje de la aplicación es fácil		X	Esto va acorde al punto anterior, ya que es complicado dominar todas las herramientas hace que la curva de aprendizaje sea lenta

## ANEXO 2

### Requerimientos funcionales

<b>Número de requerimiento:</b>	RF01
<b>Nombre del requerimiento:</b>	Autenticación del usuario
<b>Tipo:</b>	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
<b>Característica:</b>	Para poder agregar notas en la aplicación el usuario primero tendrá que iniciar sesión.
<b>Descripción:</b>	El sistema tendrá que validar los datos ingresados por el usuario y permitir el ingreso a ver notas, caso contrario el sistema redireccionará al usuario a la ventana de registro.
<b>Prioridad del requerimiento:</b>	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

<b>Número de requerimiento:</b>	RF02
<b>Nombre del requerimiento:</b>	Registro de usuario
<b>Tipo:</b>	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
<b>Característica:</b>	El usuario podrá registrarse en la aplicación en caso de no tener una cuenta.
<b>Descripción:</b>	El sistema permite crear cuentas para agregar notas, para esto es necesario que el usuario ingrese datos como: Nombres, Email, Password.
<b>Prioridad del requerimiento:</b>	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

<b>Número de requerimiento:</b>	RF03
<b>Nombre del requerimiento:</b>	Crear notas
<b>Tipo:</b>	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
<b>Característica:</b>	El usuario podrá crear notas, a las cuales puede acceder ingresando en su cuenta.

<b>Descripción:</b>	El sistema permite crear notas para esto el usuario tendrá que haber iniciado sesión y deberá enviar datos como: el título de la nota y la descripción
<b>Prioridad del requerimiento:</b>	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

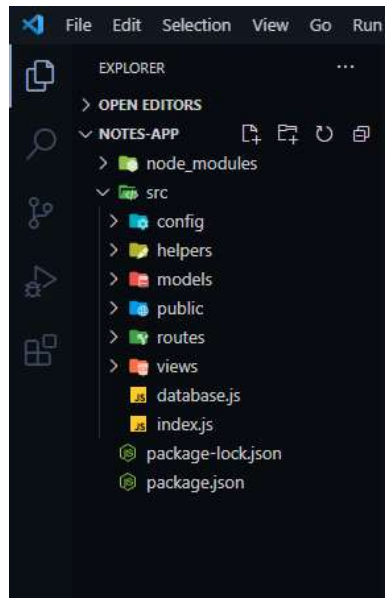
<b>Número de requerimiento:</b>	RF04
<b>Nombre del requerimiento:</b>	Editar notas
<b>Tipo:</b>	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
<b>Característica:</b>	El usuario podrá editar las notas que ha guardado con anterioridad en la aplicación
<b>Descripción:</b>	El sistema permite editar las notas de un usuario para esto el usuario tendrá que enviar los nuevos datos, el sistema automáticamente presenta al usuario los datos de la nota a actualizar.
<b>Prioridad del requerimiento:</b>	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

<b>Número de requerimiento:</b>	RF05
<b>Nombre del requerimiento:</b>	Eliminar notas
<b>Tipo:</b>	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
<b>Característica:</b>	El usuario podrá eliminar notas que haya guardado
<b>Descripción:</b>	El sistema permitirá al usuario eliminar sus notas, si este elimina todas las notas de su cuenta, el sistema presentará un mensaje indicando si desea crear una nueva nota.
<b>Prioridad del requerimiento:</b>	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional

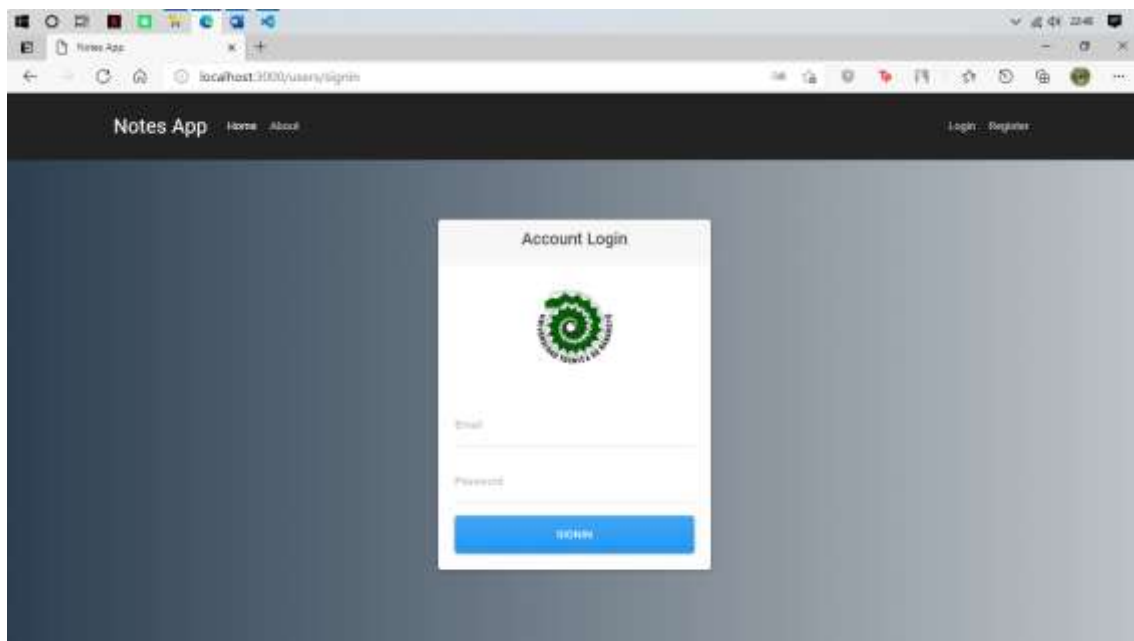
## Requerimientos no funcionales

ID	NOMBRE	DESCRIPCION
<b>RNF1</b>	Sistema con interfaz minimalista	El sistema debe tener una interfaz que sea fácil de entender para el usuario.
<b>RNF2</b>	Diseño accesible o de fácil manipulación	La aplicación tiene que ser muy sencilla y básica para que el usuario pueda manipularla sin problemas.
<b>RNF3</b>	El sistema confirma todas las acciones realizadas por el usuario	El sistema tendrá que notificar al usuario la confirmación o un posible error al momento de manipular la aplicación.
<b>RNF4</b>	Sistema escalable	El sistema permitirá agregar nuevas funciones

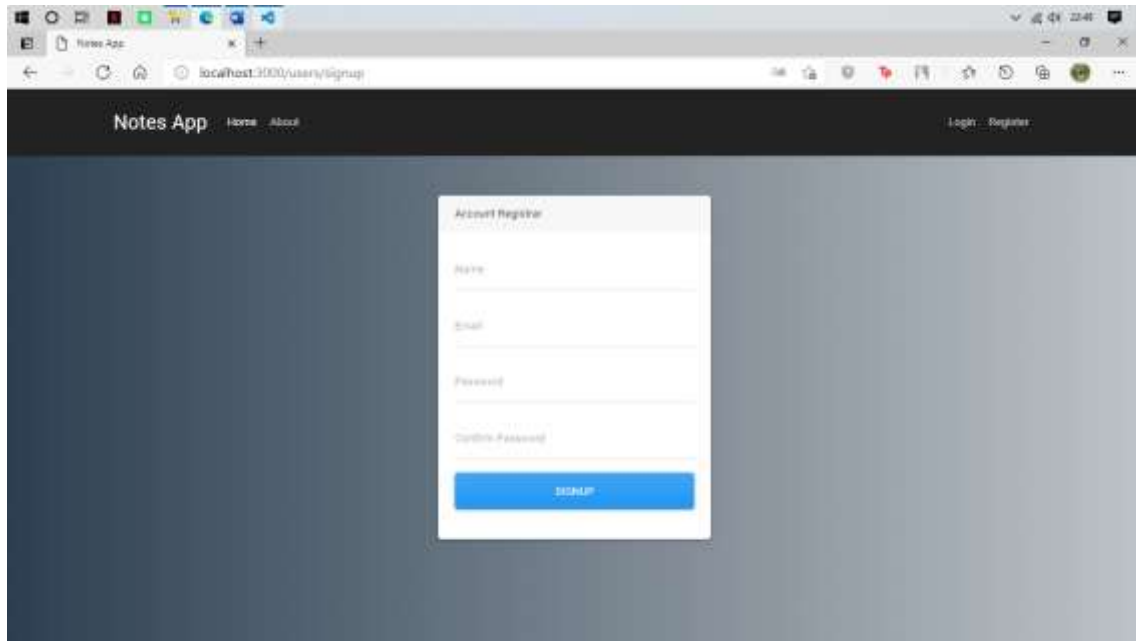
### ANEXO 3



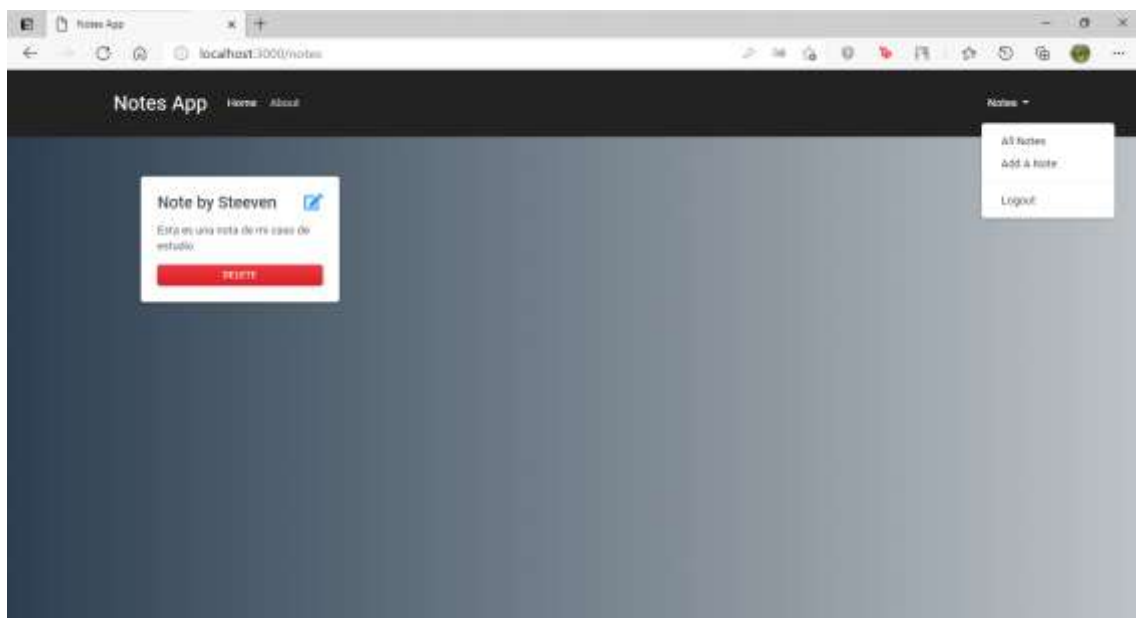
*Figura 9 Estructura de la aplicación web creada con el stack MERN*  
**Fuente:** Steeven Jossueph Naranjo Heredia



*Figura 10 Página de inicio de sesión*  
**Fuente:** Steeven Jossueph Naranjo Heredia



*Figura 11* Página de registro de usuarios  
**Fuente:** Steeven Jossueph Naranjo Heredia



*Figura 12* Página de gestión de notas de un usuario  
**Fuente:** Steeven Jossueph Naranjo Heredia