

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR SEDE ESMERALDAS



CARRERA:

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PREVIO AL GRADO ACADÉMICO DE INGENIERÍA
DE SISTEMAS Y COMPUTACIÓN

TEMA DE INVESTIGACIÓN:

COMPARATIVA DE STACKS JAVA Y JAVASCRIPT PARA LA IMPLEMENTACIÓN DE
UN SERVICIO WEB RESTFUL PARA CONSULTA DE DATOS EN LA UNIVERSIDAD
CATOLICA SEDE ESMERALDAS.

LINEA DE INVESTIGACIÓN:

DESARROLLO DE SOFTWARE

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERÍA DE SISTEMA Y COMPUTACIÓN

AUTOR:

LARA ORTIZ PABLO ANDRES

ASESOR:

MGT. SAYAGO HEREDIA JAIME PAUL

ESMERALDAS, 2022

1. TRIBUNAL DE GRADUACIÓN

Trabajo de tesis aprobado luego de haber dado cumplimiento a los requisitos exigidos por el Reglamento de Grado de la PUCESE, previo a la obtención del título de Ingeniería en Sistemas y Computación.

Presidente del Tribunal de Graduación

Lector 1

Mgt. Marc Grob

Lector 2

Mgt. Evelyn Lorena Flores

Director(a) de Escuela

Mgt. Susana Patiño

Director de Tesis

Mgt. Jaime Sayago

2. DEDICATORIA

El siguiente trabajo de investigación se lo dedico a mis padres, Rebeca Ortiz Moran y Pablo Ignacio Lara, quienes me han guiado correctamente por el camino para desarrollarme socialmente como persona y profesional.

A mis abuelos Ángel Rúaless y Zoila Moran quienes me han sabido criar y otorgar sabiduría a través de su experiencia y el amor incondicional de familia que solo un abuelo o una abuela sabe dar. Por guiarme por el camino de la rectitud de la honestidad la reciprocidad, por formarme como persona, ser gracias a ellos la persona que hoy soy, darme el estímulo necesario y tener la paciencia para comprender las metas que me he planteado, gracias.

Pablo Andrés Lara Ortiz

Contenido

1. TRIBUNAL DE GRADUACIÓN	2
2. DEDICATORIA	3
Introducción	6
Presentación del problema	6
Planteamiento del problema	6
Justificación	7
Objetivos	8
Objetivo General	8
Objetivos específicos	8
1. Capítulo 1: Marco Teórico	9
1.1. Bases teóricas conceptuales	9
1.1.1. Cloud Computing	9
1.1.2. SAAS	9
1.1.3. PAAS	10
1.1.4. Arquitectura RESTful	10
1.1.5. Servicios web RESTful	11
1.1.6. Stacks Java EE	12
1.1.7. Java EE	12
1.1.8. Spring boot	13
1.1.9. Spring data REST	13
1.1.10. JSP/HTML/CSS	13
1.1.11. Micro servicios	14
1.1.12. Gradle	14
1.1.13. Maven	14
1.1.14. Docker	14
1.1.15. JavaScript	15
1.1.15.1. REACTJS	15
1.1.15.2. VUEJS	15
1.1.15.3. ANGULARJS	16
1.1.16. Stack MERN	16
1.1.17. JSON	17
1.1.18. GitHub	17
1.1.19. Tecnologías de medición de software	18
1.1.19.1. Load Runner	18
1.1.19.3. Apache Bench	19

1.1.19.4.	Apache JMeter	19
1.1.19.5.	Siege.....	20
1.2.	Antecedentes	20
1.3.	Fundamentación legal.....	21
2.	Capítulo 2: Metodología	22
2.1.	Delimitación espacio temporal	22
2.2.	Tipo de investigación.....	22
2.3.	Métodos	22
2.4.	Variables de investigación	23
2.5.	Población y muestra.....	25
2.6.	Técnicas e instrumentos de recolección de datos.....	25
3.	Capítulo 3: Resultados.....	27
3.1.	Análisis e interpretación de los resultados.....	27
3.2.	Análisis de los datos	27
3.3.	Identificación de las herramientas.....	28
3.4.	Arquitectura de software.....	28
3.5.	Desarrollo de los micro servicios	30
3.5.1.	Aplicación en Spring Boot Java.....	30
3.5.2.	Aplicación Stack MERN.....	32
3.6.	Análisis de variable usabilidad	34
3.7.	Análisis variable eficiencia	35
3.8.	Comparación	38
4.	Capítulo 4: Discusión	41
5.	Capítulo 5: Conclusiones	42
6.	Capítulo 6: Recomendaciones	43
7.	Anexos	44
8.	Referencias.....	64

Introducción

Presentación del problema

En la actualidad el manejo de datos se ha convertido en un ámbito sumamente importante a la hora de desarrollar software, es por ello que hoy se encuentran diferentes tecnologías para el manejo de los mismos. En los tiempos que transcurren, las tecnologías están siendo enfocadas a la nube lo que conlleva a que el manejo de datos sea de manera remota, y como apoyo a ellos surgen los servicios web.

En la presente investigación se presentan los resultados de la comparación entre dos Frameworks los cuales son Spring Boot y MERN. La comparación consistió en analizar los diferentes requerimientos, procesos y resultados al momento de desarrollar software en estos Frameworks, y de esta manera destacar al lenguaje con mayor rendimiento y eficiencia.

Planteamiento del problema

Los servicios web son técnicas para el manejo de datos integrando el software a través de una red. Estas técnicas de manejo de datos remota se realizan a través de API (Application Programming Interface) por sus siglas en inglés, esta herramienta permite enviar y recibir mensajes de tipo (XML y JSON) los cuales servirán para ejecutar todo tipo de sistemas informáticos [1].

Actualmente la arquitectura de servicios web Restful domina el desarrollo web, su popularidad se vio aumentada debido a sus ventajas en cuanto a transferencia de datos de tipo JSON, los cuales eliminan el inconveniente de datos redundantes en comparación con su contraparte XML. Esta arquitectura se basa en una idea avanzada de arquitectura orientada a los recursos [2].

A pesar de las ventajas de la arquitectura ya mencionada la cual es la solución más optada por los desarrolladores al día de hoy, para el desarrollo de aplicaciones que conllevan servicios web, existen un sin número de herramientas, es debido a esto que existen conflictos para llegar a una toma de decisión definitiva de cual tecnología debería utilizarse a la hora del desarrollo de software.

Comúnmente en el desarrollo, se elige el lenguaje de programación JAVA debido a su confiabilidad como tecnología del lado del servidor, su robustez su distribución, su arquitectura neutral, y confiabilidad además de ser gratis y estar bien documentado [3].

Muy a menudo se pueden encontrar varios Stacks para los lenguajes de programación JAVA y JavaScript, estos pueden ser escogidos para el desarrollo de un proyecto. Entre las herramientas a elegir están las bases de datos, los frameworks y la arquitectura de la aplicación a realizar.

Una vez identificada la arquitectura del software, las siguientes incógnitas a resolver son, (i) ¿Cuál será la base de datos a utilizar? ¿Será una base de datos relacional o no?; (ii) ¿Qué framework es el más conveniente?; (iii) ¿Qué tipo de arquitectura se utilizará? Algunos framework como angularjs, reactjs, vuejs, los cuales han visto incrementado su uso desde la llegada de la web 2.0, han obligado a los desarrolladores a implementar a sus webs características de como ser reactivos e interactivos [4] , estas características pueden ser obtenidas gracias al uso de los frameworks ya mencionados.

Entre los gestores de bases de datos existen algunos como mysql, mongoDB y firebase esta última perteneciente a la compañía GOOGLE. Mysql es uno de los gestores de bases de datos más utilizados entre las empresas. Mientras que mongoDB y firebase han tomado gran popularidad debido a su capacidad para rápida transferencia de datos, estos gestores de datos son los más utilizados actualmente a la hora de desarrollar aplicaciones móviles.

Justificación

Debido a la existencia de infinidad de herramientas para el desarrollo de software, se crean conflictos al momento de desarrollar una aplicación. Este trabajo de investigación nace con la idea de contribuir y resolver la problemática en la que se puede encontrar un programador al momento de selección las tecnologías adecuadas para el desarrollo de software dado que, al momento de la elección de las herramientas equivocadas puede generar problemas como la pérdida de

recursos y tiempo en la fase de desarrollo de una aplicación, como consecuencia puede provocar retrasos en los springs o dirigir un proyecto directamente al fracaso. Por ello se realizó la presente investigación la cual con los resultados y datos obtenidos se procederá a realizar una comparativa entre los diferentes stacks de JAVA y JavaScript.

Con los datos obtenidos en la presente investigación, se entrega información confiable que permita la correcta selección de las tecnologías de los lenguajes de programación ya mencionados y así se creó una guía para los desarrolladores de software que ayuda a identificar las herramientas pertinentes para el desarrollo de una aplicación según sea el caso.

Objetivos

Objetivo General

- Comparar los diferentes stacks de JAVA y JavaScript para seleccionar la tecnología más conveniente en el desarrollo de sistemas basados en ecosistemas de servicios web.

Objetivos específicos

- Identificar los patrones característicos de las tecnologías Java y JavaScript para el desarrollo de servicios web.
- Reconocer los principales aspectos relacionados con el desarrollo de arquitecturas orientadas a servicios web.
- Construir una aplicación basada enfocada en herramientas y tecnologías en ambos lenguajes de programación orientada a servicios web.
- Definir una guía para desarrolladores de software basada en los resultados de la investigación.

1. Capítulo 1: Marco Teórico

1.1. Bases teóricas conceptuales

Para ayudar con la comprensión y el conocimiento de la investigación presente, primero se presentan los fundamentos más esenciales. Partiremos por la conceptualización de la arquitectura REST, seguido de las herramientas o stacks que conforman JAVA, para el desarrollo de software. Y finalizando con las herramientas de medición de software las cuales sirvieron para evaluar las herramientas mencionadas en el presente marco teórico.

1.1.1. Cloud Computing

Cloud computing o computación en la nube, permite a los usuarios tener acceso a softwares remotos y almacenamiento de datos, así como también a procesamiento de datos a través de internet, convirtiéndose así en una opción alternativa a las computadoras personales o servidores locales. Muchas empresas están optando por ejecutar servicios Cloud computing debido a los beneficios que esto representa. Este concepto también ha influenciado en campos como el IoT o internet de las cosas, con la implementación de redes de sensores inalámbricos gracias al cloud computing, el internet de las cosas se ha convertido en una realidad de desarrollo. [5]

1.1.2. SAAS

SAAS (Software as a service) o también conocido como software bajo demanda, es un método para distribución software basado en el alojamiento de los servicios en la nube por un proveedor. Los usuarios pueden acceder a estos servicios a través de internet, por lo que no tienen necesidad de instalar softwares adicionales a sus dispositivos.

1.1.3. PAAS

PAAS (Platform as a service) es un modelo de desarrollo de software el cual permite a los desarrolladores crear aplicaciones mediante internet sin tener que preocuparse por la integridad de la infraestructura. A diferencia de SAAS en este caso PAAS puede gestionar la plataforma mas no el servidor.

1.1.4. Arquitectura RESTful

En la actualidad los sistemas están contruidos en base a la arquitectura RESTful, sus características de escalabilidad, interoperabilidad y simplicidad la convierten en una promesa de desarrollo significativa en cuanto a velocidad e integración entre varias aplicaciones [6]. Amazon, Facebook y Mega ofrecen a los desarrolladores APIs basadas netamente en REST, esta arquitectura también se puede encontrar en muchas aplicaciones móviles.

La arquitectura RESTful se utiliza en cualquier cliente HTTP, su simplicidad yace en que utiliza una interfaz web que usa hipermedios para la transición de la información. Al ser escalable ofrece soporte a numerosos componentes y las interacciones entre ellos. Entre otras características de REST tenemos:

- Carece de estados, es decir la información no se almacena en el servidor, toda la información es manejada por el cliente a través de mensajes HTTP.
- Sus operaciones están bien definidas, de las cuales las más importantes son GET, POST, PUT y DELETE.
- La información es representada a través de hipermedios como lo son XML, HTML o JSON.

Los métodos ya mencionados de esta arquitectura los cuales son GET, POST, PUT y DELETE funcionan de la siguiente forma.

- ✓ GET: Es utilizada para acceder a los recursos.
- ✓ POST: Se utiliza para la creación de nuevos recursos.
- ✓ PUT: Modifica los recursos ya existentes.
- ✓ DELETE: Elimina recursos.

Para la utilización de los recursos las URIs deben llevar un ID (identificador único) que debe ser representado de la siguiente manera:

- `http://...../recurso/6`

El invocar un recurso conlleva una serie de características que deben ser tomadas en cuenta:

- El recurso no debe llevar consigo una acción o verbo:
 - × `http://...../recurso/6/eliminar`
- Debe estar independizado del formato del recurso:
 - × `http://...../recurso/6.png`
- Debe llevar jerarquía lógica:
 - × `http://...../recurso/458`
- La filtración no debe ser expuesta en la URI:
 - × `http://...../recurso/6/fecha/mes`

1.1.5. Servicios web RESTful

Los servicios web Restful han adquirido gran popularidad al presentar ventajas en el envío de datos sobreponiéndose a su contraparte SOAP fig1. La composición del servicio web se ha convertido en la forma más prometedora de respaldar la integración de aplicaciones de empresa a empresa [7].

Los desarrolladores se centran cada vez más en los servicios web Restful orientándose a la arquitectura de servicios (ROA). Oponiéndose a los usuarios de servicios web clásicos, los basados en RESTful cumplen con el principio Hypermedia As The Engine Of Application State (HATEOAS) que instiga el uso de enlaces hipermedia para navegar entre los estados de una aplicación [8].

Muchos desarrolladores del estilo REST lo describen como fácil, sin embargo, esto no quiere decir que sea sencillo o simplista, ni tampoco significa que los sistemas REST carezcan de sofisticación. La simplicidad del estilo arquitectónico REST viene del hecho de que no solo define claramente sus compensaciones y limitaciones, sino que también resuelve problemas como la identificación de recursos. En consecuencia, los desarrolladores que diseñan y compilan servicios RESTful consideran mucho más fácil rastrear problemas importantes que pueden afectar el rendimiento del sistema [9].

1.1.6. Stacks Java EE

Al ser uno de los lenguajes de programación más utilizados y que posee diferentes herramientas con diferentes propósitos, java se convierte en uno de los lenguajes de programación más robustos de la actualidad. Para el desarrollo de una aplicación web son necesarias varias herramientas que conforman tanto el front-end como el back-end.

Para el desarrollo del front- end en java se debe utilizar JSP acompañado de HTML y CSS, mientras que en el back-end se utilizan spring boot y una base de datos. A continuación, veremos algunas de los stacks de java para el desarrollo de aplicaciones de web.

1.1.7. Java EE

Java es un lenguaje de programación con el cual se puede desarrollar cualquier tipo de software, una de las principales características por las cuales java ha tomado popularidad es ser un lenguaje de programación independiente del software. Esto quiere decir que al desarrollar software en java este funcionara en cualquier ordenador del mercado. La razón de esto es que java ha creado una máquina virtual que sirve de puente entre el lenguaje de programación y el sistema operativo, es por ello que se puede ejecutar código Java en distintos sistemas operativos.

Java EE también sirve para otros marcos de uso, que exponen información y servicios distribuidos a usuarios remotos. El marco de Java EE proporciona a los desarrolladores los medios para especificar políticas de control de acceso a fin de asegurar las propiedades de confidencialidad e integridad de los recursos expuestos por las aplicaciones web [10].

Las aplicaciones Java EE generalmente están construidas por JSP y Servlets. El mecanismo de control de acceso implementado en este nivel se encarga de controlar el acceso a estos elementos junto con cualquier otro artefacto accesible como paginas HTML puras, documentos multimedia, etc. Las políticas de control de acceso se especifican utilizando dos mecanismos diferentes: seguridad declarativa y seguridad programática, esta última se proporciona para los casos en que se necesita un control de acceso fino, que necesita evaluaciones del contexto

de usuario, Java EE recomienda un uso preferencial de seguridad declarativa siempre que sea posible [10].

1.1.8. Spring boot

Este framework se basa en la técnica de desarrollo spring, por lo cual tiene las bibliotecas necesarias para el desarrollo de software basado en la técnica mencionada.

Este framework orientado en java proporciona flexibilidad para la arquitectura basada en SOA. Facilita la construcción e implementación de aplicaciones spring independientes de nivel de producción con muy poca configuración de spring [11].

1.1.9. Spring data REST

Esta herramienta se implementa de dos maneras con Spring MVC y Spring WebFlux, estos ofrecen la base para construir estos servicios. Debido a lo complicado que resulta implementar el principio más simple de los servicios REST para un sistema de objetos de múltiples dominios, esta herramienta basada en los repositorios de spring data los exporta de manera automática como recursos Rest.

1.1.10. JSP/HTML/CSS

Para el diseño front-end de una página web, tenemos este lenguaje de programación llamado JSP, el cual nos permite diseñar páginas web dinámicas. Este lenguaje se ha vuelto estándar para el diseño de páginas web dinámicas, y está conformado en realidad entre etiquetas HTML y JAVA [12]. Gracias a que está basado en java este lenguaje se convierte en multiplataforma por lo cual puede ser ejecutado en casi todos los entornos. En cuanto a CSS se refiere este código nos permite diseñar el entorno visual de la página web, modificando el código HTML, estas tecnologías son altamente compatibles entre sí.

1.1.11. Micro servicios

En el desarrollo de software existe un enfoque donde el software se comunica con múltiples servicios pequeños. Esta comunicación es posible gracias a las API (Application Programming Interface). Las características de estos micro servicios es que son autónomos lo cual nos permite desarrollar servicios que se pueden implementar sin afectar el funcionamiento del software, y también son especializados, esto quiere decir que cada servicio esta específicamente desarrollado para resolver un problema.

Los micro servicios también son una de las arquitecturas de software más populares de la actualidad, integrados a las tecnologías de java para el desarrollo de aplicaciones [13].

1.1.12. Gradle

Esta herramienta de código abierto está diseñada para ser lo suficientemente flexible para compilar casi cualquier tipo de lenguaje de programación. Nos evita el trabajo innecesario a la hora de ejecutarse tareas que deben ser solo ejecutadas por que sus entradas y salidas han sufrido modificaciones.

1.1.13. Maven

Este artefacto basado en JVM es uno de los repositorios más populares utilizados en la actualidad. Este almacena una gran colección de binarios de softwares junto con sus metadatos, en una estructura eficientemente definida, de manera que los usuarios pueden acceder a las dependencias. Esta herramienta predica el concepto de reutilización y facilidad de dependencia. [14]

1.1.14. Docker

Cuando hablamos de Docker hablamos de contenedores, pero ¿Qué es un contenedor en el área de desarrollo de software? Para poder ejecutar una aplicación previamente desarrollada, un ordenador primero debe cumplir con los requerimientos y especificaciones adecuadas para la ejecución de esta aplicación, es por ello que se desarrollan los contenedores en los cuales se puede ejecutar casi

todo tipo de software independientemente del sistema operativo que tenga instalado, ya que estos contenedores contienen todo tipo de requerimientos ya sea, Java, Tomcat, etc. todo lo necesario para que una aplicación sea desarrollada y ejecutada, esto convierte a la aplicación desarrollada en una aplicación portable.

Esta es la base sobre la cual está construido Docker el cual nos permite construir contenedores potentes, ligeros y que puedan ser ejecutados en diferentes ordenadores gracias a su característica de portabilidad.

1.1.15. JavaScript

JavaScript es un lenguaje de programación que permite a su usuario implementar funciones en páginas web, este lenguaje permite crear contenido con actualización dinámica, en las tecnologías web se describe JavaScript como la tercera capa del pastel el cual se complementa con HTML y CSS. Este lenguaje de programación también permite crear gráficos interactivos a todo tipo de sitios web [15].

1.1.15.1. REACTJS

JavaScript es un lenguaje de programación que ha tomado fuerza en los últimos años, con su creciente popularidad se han creado algunos Frameworks potentes, uno de estos es reactjs creado por Facebook, el cual permite el desarrollo de interfaces interactivas.

Una de las ventajas de reactjs es que utiliza la tecnología DOM virtual el cual le permite crear una copia en memoria y compararla con la versión anterior de esta manera, reactjs evita renderizar toda la aplicación o página web cada vez que surjan cambios.

1.1.15.2. VUEJS

Al igual que reactjs VUE es un framework para JavaScript el cual permite crear SPA (single page applications) de forma dinámica. Estos dos comparten muchas similitudes, debido a que ambos utilizan tecnologías DOM de virtualización. Pero a diferencia de reactjs VUE no contiene un ecosistema tan abundante. En cuanto al rendimiento, esta herramienta es comparable a reactjs, a veces estando un poco

por delante debido a la ligereza en su implementación en el DOM virtual. Este framework está diseñado para ser implementado incrementalmente.

1.1.15.3. ANGULARJS

Otro framework que es muy popular actualmente es AngularJS al igual que VUEJS y ReactJS este framework permite crear proyectos SPA, es de los marcos más utilizados para desarrollar aplicaciones web. Las aplicaciones web poco a poco han tomado camino reemplazando a las aplicaciones de escritorio y angularjs es de las herramientas que han hecho este avance posible. Angularjs permite cargar datos sin tener que recargar las paginas constantemente, esto debido a que funciona dinámicamente al igual que sus competidores reactjs y vuejs, casi todo se hace del lado del cliente, esta herramienta mantenida por google nos permite trabajar de manera front-end para el desarrollo de aplicaciones web [16].

1.1.16. Stack MERN

MERN (MONGODB, EXPRESS, REACTJS, NODEJS) es una herramienta que utiliza JavaScript en ambas partes (cliente, servidor), es decir es FullStack JavaScript. Este Stack comparte similitudes con el Stack MEAN (MONGODB, EXPRESS, ANGULARJS, NODEJS), este último reemplazando a ReactJS por AngularJS.

El creciente número de aplicaciones orientadas a la nube, obliga a tener comunicación rápida entre dispositivos IoT y la nube misma. Para este tipo de comunicación es necesario utilizar tecnologías tales como MEAN o MERN y arquitectura tipo RESTful [17].

En el apartado anterior se conceptualiza lo que es ReactJS uno de los componentes del Stack MERN, a continuación, se definirá las herramientas restantes que componen este Stack.

- **MONGODB:** Es una base de datos no relacional muy sencilla de utilizar y que ha tomado popularidad entre los programadores front-end y desarrolladores móviles en los últimos años. Posee una versión online la cual permite desplegar soluciones de base de datos en poco tiempo para

entornos de desarrollo y Testing. La configuración y conexión a esta base de datos es muy sencilla, realiza la conexión a la base de datos toma pocas líneas de código que esta misma en su versión online ofrece para muchos lenguajes de programación como Nodejs, java, Python, PHP.

- EXPRESS: Es una herramienta que nos permite crear la infraestructura para un proyecto web, con ella se puede gestionar todo el back-end apoyándonos con NodeJS. Con esta herramienta se puede ahorrar el tiempo de configuración de una aplicación server side.
- NODEJS: Es un entorno de ejecución en tiempo real que nos permite ejecutar código JavaScript de ahí sus siglas .JS. NodeJS controla los eventos por lo cual es ligero y eficiente, las entradas (solicitudes) y salidas (respuestas) pueden ser descritas como cualquier operación (leer, escribir) cualquier tipo de archivo hasta solicitudes tipo HTTP.

1.1.17. JSON

JSON (JavaScript Object Notation) es el método de traslado de datos el cual se popularizo por tener ventajas sobre XML, una de estas ventajas es que resulta mucho más fácil crear un analizador sintáctico (PARSER) para JSON. En JavaScript resulta muy sencillo analizar un texto tipo JSON utilizando la función eval(), debido a su extensa adaptación a JSON se lo considera un formato de datos independiente del lenguaje, por lo cual podemos encontrar métodos de análisis de datos tipo JSON en otros lenguajes de programación. Debido a esta llamada adaptabilidad extensa que lo caracteriza, JSON es un tipo de archivo que se complementa perfectamente con el protocolo RESTful, el cual busca la reducción de gastos generales de recursos [18].

1.1.18. GitHub

Para que el desarrollo de software de código abierto sea exitoso, la adquisición de colaboradores externos es un factor importante. Sin embargo, a pesar de la necesidad de aportes externos, muchos proyectos han desaparecido sin poder adquirir cotizantes. GitHub, un servicio de alojamiento de Internet para el

desarrollo de software abierto, se utiliza ampliamente para proyectos de investigación de temas relacionados.[19]

Se caracteriza por ser una herramienta colaborativa donde cualquiera puede tener acceso al código desarrollado por un usuario y así ser mejorado o generar ramificaciones a partir del código original. Esta herramienta utiliza el sistema de control Git que permite observar las diferencias entre las versiones de código, restaurar viejas versiones en casos de emergencias y fusionar los cambios de distintas versiones.

1.1.19. Tecnologías de medición de software

Las herramientas que se utilizaron en este proyecto pasaron por un proceso de medición en los cuales se analizaron la usabilidad, y la eficiencia de la herramienta, para ello existen softwares que ayudaron con las pruebas de rendimiento. Para mencionar algunos tenemos JMeter, LoadRunner, NeoLoad y apachebench, este último el cual se utilizó para las pruebas especificando el porqué de su uso en el apartado 2. Mientras que para las pruebas de concurrencia se utilizó el software Siege, el cual queda detallado en el apartado 2.1.16.2.

1.1.19.1. Load Runner

LoadRunner es una herramienta para realizar pruebas de software, se ha utilizado con éxito para realizar pruebas automatizadas basada en su API de automatización de pruebas [20], esta herramienta está compuesta por tres componentes.

- Virtual User Generator: Permite grabar y construir los scripts de rendimiento.
- Controller: Es donde se especifican las pruebas de rendimiento, se colocan los datos de entrada como: duración de la prueba, la carga, etc.
- Analysis: Esta última parte del software se encarga de generar el análisis de los resultados arrojados por las pruebas de rendimiento.

1.1.19.2. NeoLoad

NeoLoad es una herramienta que permite realizar pruebas de rendimiento a aplicaciones web nativas. Gestiona poblaciones virtuales de usuarios para que realicen las mismas o distintas acciones al mismo tiempo y generar un conjunto de informes una vez realizadas las pruebas. Esta herramienta ha pasado por análisis en sus funcionalidades de test de rendimiento para el despliegue de aplicaciones web [21], junto con otras herramientas de Testing.

1.1.19.3. Apache Bench

Una de las ventajas que ofrece esta herramienta de medición es que al igual que apache es de código abierto. Al ejecutar programas en un servidor apache es muy sencillo ejecutar el comando que permite testear el rendimiento del servidor, este comando se conoce como **ab** al cual se le puede suministrar parámetros para obtener la información deseada. Este comando en ejecución nos proporciona información como la moda y la media de las mediciones, así como también errores subyacentes o datos descargados.

Este software fue previamente utilizado para la medición de rendimiento en la investigación “*Comparative Analysis for Web Applications Based on REST Services: MEAN Stack and Java EE Stack*” [22]. También ha sido utilizado para la medición de pruebas para generar cargas de trabajo de cliente con solicitudes y conexiones simultáneas en el trabajo investigativo “*Performance Evaluation of a Clustered Memcache*”[23]. Las cuales se encuentran debidamente referenciadas en el apartado 2.2. Antecedentes de esta investigación.

1.1.19.4. Apache JMeter

Apache Jmeter es una herramienta desarrollada en JAVA para realizar pruebas de estrés y medición del desempeño de software. Esta herramienta puede ser utilizada en servidores, redes u objetos para probar la compresibilidad de los servicios mediante simulaciones de cargas pesadas o para analizar el rendimiento general bajo diferentes presiones. Esta herramienta puede realizar pruebas de rendimiento en varios campos, como web, base de datos, FTP, LDAP, servicios web, etc. [24]

1.1.19.5. Siege

Este software desarrollado e implementado por “Jeffrey Fulmer” permite realizar pruebas de carga de software las cuales nos ayudan a detectar el rendimiento del software evaluado. Esta herramienta facilita la administración de código fuente, gestión de proyectos, gestión de requisitos, informes, compilaciones y entre otros procesos importantes entre las métricas de medición de software [24]. Al igual que apachebench este software es mencionado en el apartado 2.2. Antecedentes donde es referenciado a su respectiva investigación. Este software es seleccionado para realizar las pruebas de concurrencia, debido a todas las características ya mencionadas.

1.2. Antecedentes

Esta investigación ha sido posible gracias a los diferentes artículos científicos y libros encontrados en bases de datos, utilizando la cadena de búsqueda: (web service OR restful) AND (java) AND (data). De esta manera fueron recabados 25 artículos científicos, utilizando las bases de datos: IEEE Explorer, web science y SCOPUS.

De estas bases de datos se logró recuperar artículos científicos donde se menciona la comparación de distintos stacks de Java, artículos tales como “Comparative Analysis for Web Applications Based on REST Services: MEAN Stack and Java EE Stack” [22]. Donde se mencionan algunos stacks de java y herramientas de medición de software en las cual ha sido previamente definidas dentro del marco teórico en la sección de bases teóricas conceptuales.

La comparación entre lenguajes de programación es más común de lo que se puede pensar, esta investigación también toma en cuenta los resultados recabados en el artículo científico “Differences between programming languages offered by the academy versus demanded by the companies in Costa Rica” [25]. En el cual se realiza la comparación entre lenguajes de programación tales como JAVA, C++, JavaScript, PHP y entre otros.

Otros artículos científicos relevantes para esta investigación fueron “A design of cross-terminal web system based on JSON and REST” [2] y “RESTful service composition at a glance: A survey”[7], donde los distintos autores comparan la arquitectura de software basadas en REST en contraposición con la arquitectura SOAP y las ventajas que esta arquitectura presenta en cuanto al manejo de recursos.

A pesar de las ventajas que presenta la arquitectura tipo RESTful, no quiere decir que la arquitectura de software SOAP para el manejo de datos este muerta u obsoleta, en otros artículos tales como “Design of a simple service oriented supervisory control and data acquisition system” [3], realizado en 2017 donde el objetivo era realizar un sistema SCADA, utilizan un sistema híbrido de SOAP y RESTful para comunicarse con el backend de dicho sistema.

1.3. Fundamentación legal

Este proyecto está fundamentado legalmente en el artículo 322 de la constitución de la república del Ecuador, capítulo sexto: trabajo y producción, sección segunda: tipos de propiedad, donde se esclarece lo siguiente: “Se reconoce la propiedad intelectual de acuerdo con las condiciones que señale la ley. Se prohíbe toda forma de apropiación de conocimientos colectivos, en el ámbito de las ciencias, tecnologías y saberes ancestrales” [26].

Esta investigación científica también se basa en el código 142 tecnologías libres, sección 5 disposiciones especiales sobre ciertas obras, apartado segundo del código orgánico de economía social de los conocimientos o COESC por sus siglas. Donde se especifica: “Se entiende por tecnologías libres al software de código abierto, los estándares abiertos, los contenidos libres y el hardware libre. Los tres primeros son considerados como Tecnologías Digitales Libres“ [27].

En base a las normativas anteriormente mencionadas, el trabajo de investigación redactado, respeta todos los trabajos de investigación, así como las herramientas anteriormente citadas en base a la normativa IEEE para la cita de dichas investigaciones.

2. Capítulo 2: Metodología

2.1. Delimitación espacio temporal

La recuperación de los artículos científicos en los que se basa esta investigación, cumplen con la norma de estar en el periodo de tiempo designado de 5 años, es decir desde el año 2015 al presente año 2020.

Debido a la naturaleza experimental de esta investigación, fue necesaria una institución donde llevar a cabo la experimentación y comparación de los datos resultantes de esta investigación. Este trabajo investigativo tuvo lugar en la institución conocida como la Pontificia Universidad Católica del Ecuador sede Esmeraldas o PUCE-Esmeraldas, en el periodo del segundo semestre del año 2020.

2.2. Tipo de investigación

Este estudio de caso por la naturaleza de los datos y la información a recabar, se define como investigación cualitativa-cuantitativa, es una investigación cualitativa debido a que para este trabajo es de suma importancia la descripción detallada de este tema de investigación.

Se define de igual forma como investigación cuantitativa, ya que los datos recabados para realizar las comparaciones pertinentes entre los distintos stacks de Java exigen un sondeo cuantitativo, donde se extrajo una población y muestra de dichos stacks y se realizó pruebas para comparar la eficiencia de las herramientas.

2.3. Métodos

El objetivo ya definido de esta investigación es la comparativa de los diferentes stacks de Java y JavaScript, para ello se utilizó el método descriptivo el cual en una investigación cualitativa nos ayuda a describir las diferentes características de una población o muestra, siendo en este caso los stacks de ambos lenguajes de programación. Este método el cual ayuda al investigador a orientarse en el método científico, también nos ayuda a categorizar la información que observamos y proporcionar la oportunidad de que esta información sea replicada por otros investigadores.

2.4. Variables de investigación

En esta investigación se han logrado identificar las variables e indicadores que se analizarán y dentro de que entidades están sujetos estos análisis. En total se han logrado recabar 2 variables y 4 entidades a analizar, variables que están sujetas a las métricas de software según ISO/IEC 25000.

VARIABLES	INDICADORES	TIPO	ENTIDAD
EFICIENCIA DE LA HERRAMIENTA	<ul style="list-style-type: none"> Utilización de recursos Comportamiento de tiempos 	de Cualitativa-cuantitativa	Servidor
	USABILIDAD	<ul style="list-style-type: none"> Documentación Comunidad Soporte Satisfacción 	Cualitativa-cuantitativa

Tabla 1 variables e indicadores de investigación

A continuación, se describen las variables y los indicadores planteados en la tabla anterior.

Variable eficiencia: El conjunto de indicadores en esta variable evaluó la eficiencia con la que trabaja la herramienta analizada. Esta variable describió cuán factible y confiable es una herramienta.

Indicadores:

- I. **Utilización de recursos:** Esta variable indica la capacidad del software para manejar los recursos del sistema cuando se ejecuta bajo condiciones establecidos. Estos recursos son memoria, sistema operativo, memoria de almacenamiento.

- II. Comportamiento de tiempos:** JMeter fue una de las herramientas utilizadas para medir el tiempo de respuesta del sistema. De esta manera pudimos percatarnos de cuantas tareas puede realizar una aplicación web soportando una concurrencia alta de usuarios.
- III. Prueba de carga:** La prueba de carga es aquella que realiza objetivos determinados con el fin de determinar la carga que es capaz de soportar el sistema. Entre las pruebas que se pueden realizar son:
 - Descarga de archivos.
 - Asignación de múltiples tareas
 - Sometimiento a alto tráfico de usuarios
 - **Transferencia de datos:** En cuanto a la transferencia de datos y concurrencias se utilizó la herramienta JMeter. La cual indica los tiempos en transferencias de datos, el tiempo de respuesta del servidor y la tasa de transacción.

Variable usabilidad: La usabilidad de una herramienta está determinada por los indicadores, documentación, comunidad, soporte y satisfacción para determinar qué tan fácil resulta utilizar una herramienta.

Indicadores:

- IV. Documentación:** La documentación proporcionara información relevante al momento de medir los niveles de satisfacción de este mismo.
- V. Comunidad:** El crecimiento o disminución de la comunidad del lenguaje indicara que tanto es aceptado y utilizado el lenguaje en la actualidad.
- VI. Soporte:** Basándose en el indicador comunidad, se podrá inferir en si el lenguaje tiene un soporte de calidad que brinde a los usuarios la información necesaria para solucionar inconvenientes.
- VII. Satisfacción:** Este indicador permitió conocer los niveles de satisfacción del usuario, si el usuario encuentra la herramienta analizada fácil de usar y comprender. Este indicador será representado en tres niveles alto, medio y bajo.

Como resultado obtendremos una tabla donde podremos ponderar la evaluación de la variable usabilidad. La cual será evaluada de la siguiente manera:

- De 1 a 1,5 = regular

- De 1,5 a 2 = buena
- De 2,5 a 3 = excelente

Usabilidad	
Documentación:	
Comunidad:	
Satisfacción:	
Soporte:	

El indicador satisfacción estará dictaminado a través de los niveles promediados de los indicadores documentación, comunidad y soporte, y la usabilidad será dictaminada por el promedio entre todos estos indicadores.

2.5. Población y muestra

En esta investigación considero tomar como población, los stacks que componen el desarrollo web de java, de los cuales se tomó como muestra las tecnologías JAVA EE, spring boot, gradle para el desarrollo de una aplicación FRONT-END y en cuanto a JAVASCRIPT, el Stack MERN (MongoDB, Express, ReactJS, NodeJS) por sus siglas en ingles fue la muestra que se utilizó para desarrollar una misma aplicación FRONT-END, basada en las mismas características que la aplicación java. En cuanto a las herramientas de Testing se utilizaron las herramientas Siege, JMeter y Apache Bench para realizar las respectivas pruebas de carga y rendimiento de las aplicaciones.

2.6. Técnicas e instrumentos de recolección de datos

El presente trabajo investigativo obliga al investigador a realizar pruebas para determinar causa y efecto, ya que el objetivo de este trabajo, es comparar los diferentes stacks de java, es necesario realizar pruebas y simulaciones para determinar diferencias y eficiencia de las herramientas. Una vez obtenidos los datos resultantes de la experimentación, se procedió a evaluar cada herramienta

con una ficha de evaluación basada en algunas características de la ISO 9001-2015 bajo los parámetros de la observación científica.

Esta técnica permitió evaluar cada una de las variables anteriormente planteadas en el apartado 2.4. variables de investigación.

Ficha de evaluación de software		
Producto:	Versión:	Compañía/creador:
Descripción General:	Imagen de la herramienta:	
Pesos:		
Eficiencia de la herramienta:		
Usabilidad:		
Requerimientos del sistema:		
Licencia:	Página de descarga:	

Los pesos para la evaluación de la herramienta estarán determinados de la siguiente manera:

- De 0 a 0,5 = deficiente
- De 1 a 1,5 = regular
- De 1 a 2 = bueno
- De 2,5 a 3 = excelente

2.7. Técnicas de procesamiento y análisis de datos.

En la tabla que se puede apreciar a continuación, se especifican las técnicas de procesamiento y análisis de datos utilizados para esta investigación.

TÉCNICA	USO
ANÁLISIS DE CORRELACIÓN	Si
VISUALIZACIÓN DE DATOS	Si
ESTADÍSTICA DESCRIPTIVA	Si

Tabla 2 técnicas de análisis y procesamiento de datos

El análisis de correlaciones permitió tener una mejor percepción de la correlación que existe entre los indicadores curva de aprendizaje y satisfacción y de esta manera obtener resultados óptimos al momento de analizar la variable usabilidad. La visualización de datos otorgó una visión amplia y simplificada de los datos obtenidos al momento de evaluar la variable eficiencia por último la estadística descriptiva permitió crear tablas de frecuencia, dispersión, y graficas las cuales fueron extraídas a partir de las pruebas realizadas a las variables de investigación que se encuentran en el apartado 2.4.

3. Capítulo 3: Resultados

3.1. Análisis e interpretación de los resultados

En el siguiente apartado quedan detallados los resultados obtenidos de la presente investigación, en donde se visualizan las aplicaciones basadas en micro servicios, las cuales fueron realizadas en los dos lenguajes de programación previamente mencionados, estas fueron sometidas a pruebas para realizar la comparación respectiva entre ambas. Esta investigación como ya se ha mencionado fue realizada en el área de TICS de la Pontificia Universidad Católica del Ecuador Sede Esmeraldas, la cual proporciono los datos necesarios para construir dichas aplicaciones.

3.2. Análisis de los datos

Al realizar una rápida observación de los datos proporcionados por la institución, se logró identificar que los datos pertenecían al tipo de dato JSON, el cual estaba subdividido en varios arrays los cuales contenían información relevante de los prospectos de la universidad. Sin embargo, para el desarrollo de los micro servicios propuestos en esta tesis, se tomó en consideración el formato de datos tipo array de los prospectos, los cuales contenían la información personal de cada prospecto.

De este modo se procedió a generar más prospectos con datos personales para ejemplificar la lectura de datos desde las aplicaciones hacia la base de datos mongo en la nube. El número total de prospectos generados para ello fueron 5000,

los cuales contenían así mismo arrays con más datos personales como datos de contacto: datos telefónicos y datos de correos electrónicos. De esta manera las aplicaciones pueden ser sometidas a pruebas con datos realistas en cantidad, y con volúmenes grandes.

3.3. Identificación de las herramientas

Debido a la naturaleza de los datos proporcionados por la institución, los cuales eran una agrupación de datos tipo JSON, se utilizó la base de datos no relacional mongoDB, así como la herramienta mongoDBCompass para el correcto manejo y almacenamiento de estos datos.

Dentro de las herramientas de desarrollo de software se utilizó para el desarrollo de ambas aplicaciones el IDE IntelliJ Idea[28], entre las características que esta herramienta ofrece están una apropiada y rápida inspección del código escrito, inyección del lenguaje utilizado, refactorizaciones de lenguaje cruzado, detección de duplicados, etc.

En cuanto al almacenamiento del código escrito se utilizó la herramienta opensource GitHub[29], la cual se encuentra detallada en el apartado 1.1.19. esta herramienta ofrece la capacidad de versionar código, lleva un registro de incidencias y ofrece compatibilidad con todos los sistemas operativos.

Para la generación de datos de los prospectos utilizados en esta investigación se utilizó la herramienta online Mockaroo[30] la cual es gratuita y permite la creación de datos en grandes volúmenes en diferentes tipos de formatos.

3.4. Arquitectura de software

La arquitectura seleccionada para el desarrollo de ambas aplicaciones, fue la arquitectura MVC (model, view, controller) por sus siglas en inglés. Esta arquitectura fue seleccionada ya que aparte de ser de las más comunes y aprendidas a lo largo del estudio de la carrera, ofreció al desarrollo de estas aplicaciones la facilidad de separar los datos, la interfaz de usuario y la lógica de

control entre los distintos componentes construidos para el procesamiento de datos.

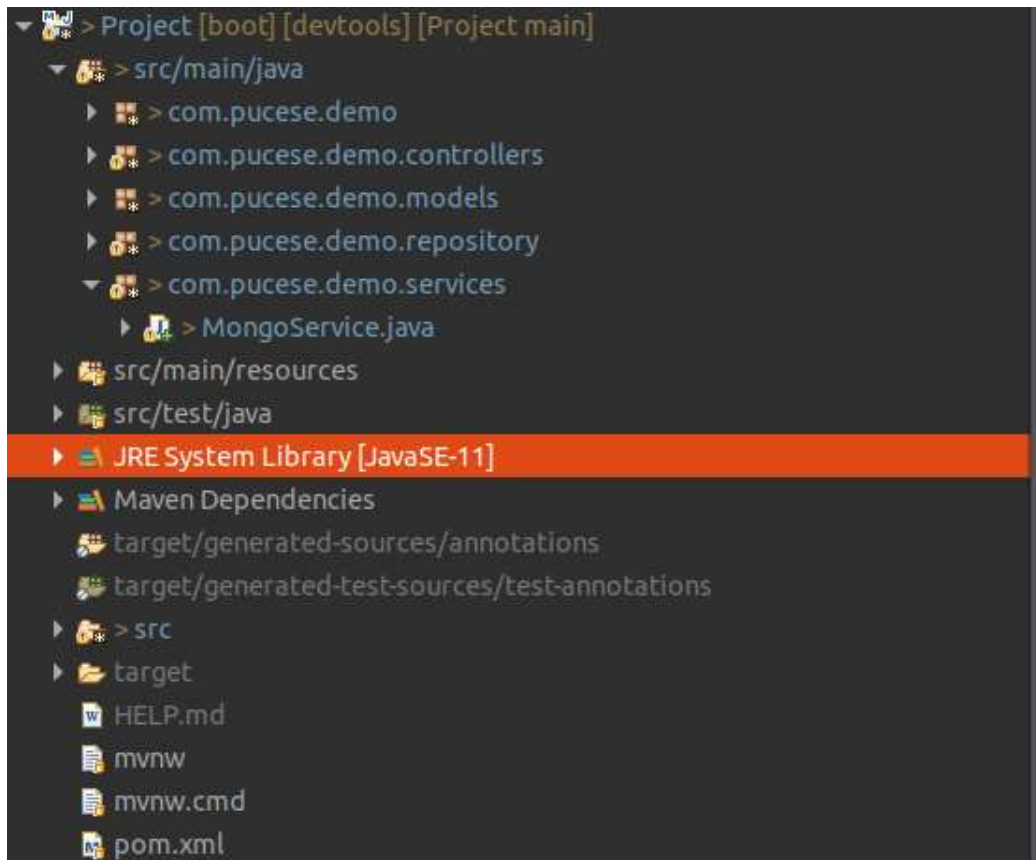


Imagen 1. Arquitectura MVC Stack JAVA

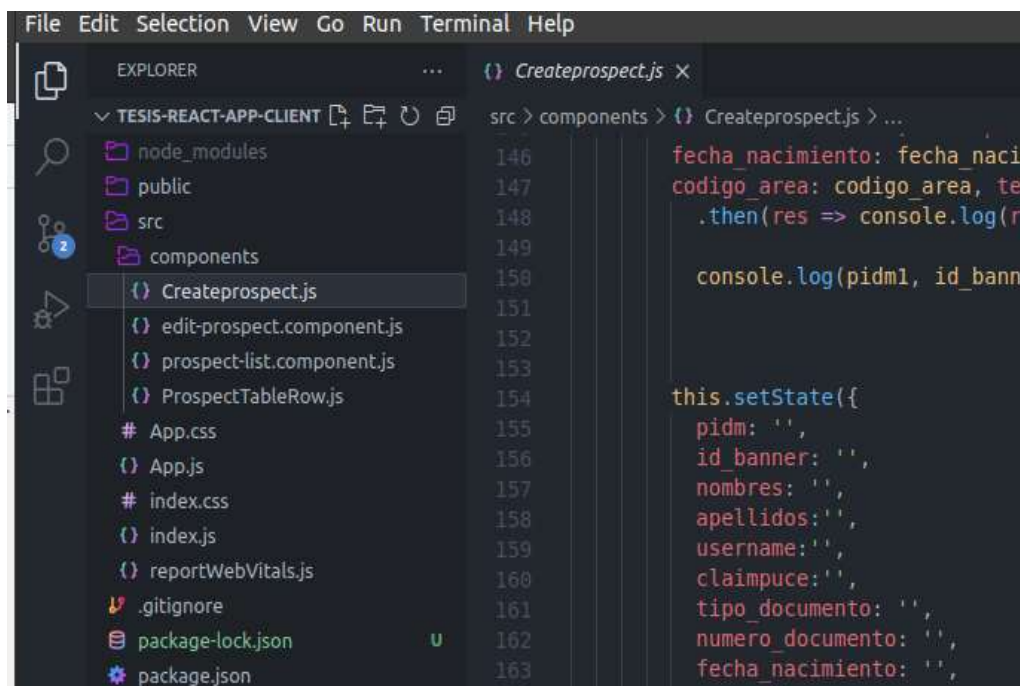


Imagen 2. Arquitectura Stack MERN

3.5. Desarrollo de los micro servicios

Una vez identificados los lenguajes y las herramientas a utilizar, se desarrolló dos aplicaciones basadas en micro servicios con los datos otorgados por la institución, los cuales cumplirían la funcionalidad CRUD (create, read, update and delete) por sus siglas en ingles. A continuación, se detalla con plenitud la estructura de las aplicaciones realizadas empezando con la aplicación realizada en JSP.

3.5.1. Aplicación en Spring Boot Java

Esta aplicación llamada TesisPuceSpringBoot basada en la arquitectura MVC ya mencionada consiste en una aplicación desarrollada en JAVA EE la cual permite al usuario la creación, lectura, modificación y eliminación de datos obtenidos de una base de datos mongoDB. Esta aplicación basada en micro servicios fue desarrollada con el propósito de permitir al usuario realizar los cambios ya mencionados a un prospecto que se encuentre registrado en la base datos o registrarlo en la misma.

Como requerimiento principal para ejecutar esta aplicación se necesita el jdk versión 16 de java el cual se puede descargar de manera gratuita de la página oficial de Oracle y apache Tomcat.

Esta aplicación estructurada bajo la arquitectura MVC cuenta con las carpetas correspondientes a cada componente necesario: Controller, Repository, Model y Views.

En la carpeta Controller podemos encontrar la siguiente clase:

- ProspectController: En esta clase encontraremos los siguientes métodos:
 1. Prospect(): Devuelve todos los prospectos registrados en la BD.
 2. Save(): Se receptan los datos desde el view para ser ingresados en la BD.
 3. Show(): Esta función devuelve los datos de un prospecto en específico buscado.

4. Delete(): En esta clase se busca el id del prospecto a ser eliminado
5. Edit(): Entrega los datos de un prospecto seleccionado
6. Update(): Prepara los nuevos datos a ser actualizados a un prospecto.

Si nos dirigimos a la carpeta Model encontraremos las siguientes clases:

- Prospect: Aquí se encuentran las clases pertinentes para el modelado de un prospecto. Se pueden observar los métodos Getter y Setter junto con el constructor de los datos.

Dentro de la carpeta Repository encontramos la siguiente clase:

- ProspectRepository(): Esta clase se extiende de la clase CrudRepository<>, enviando como parámetros un prospecto y un String que será el id del prospecto. De esta manera se construyen los métodos para él envió de información a la base de datos.

Dentro de la carpeta resource se puede encontrar por defecto el archivo Application.properties, es en este archivo donde se instancia la conexión a la base de datos mongo a través de una única línea de código.

Por ultimo tenemos los Views, estos se encuentran en la carpeta ya mencionada resoucer/templates. Aquí se encuentran todos los componentes que permiten la visualización y manejo de los datos a través de una interfaz, estos componentes responden a los nombres:

- create.html
- default.html
- edit.html
- index.html
- prospect.html
- show.html

Sistema CRUD Spring Boot

Nombres	Apellidos	Username	Numero Documento	Action	
Jacquenette	Gluyas	jgluyas0	0859978096	Edit	Delete
Penny	Sherwen	psherwen1	0839467006	Edit	Delete
Kristian	Wibrow	kwibrow2	0880035965	Edit	Delete
Chilton	Mosdall	cmosdall3	0822622564	Edit	Delete
Percival	McAirt	pmcairt4	0846351993	Edit	Delete
Eberto	Deem	edeem5	0890364755	Edit	Delete
Harper	Blanpein	hblanpein6	0826895275	Edit	Delete
Cortney	Holson	cholson7	0891296790	Edit	Delete
Moises	Tale	mtale8	0892308710	Edit	Delete
Paulette	Greaves	pgreaves9	0886902374	Edit	Delete
Zachariah	Janicki	zjanickia	0829298657	Edit	Delete
Sarine	Norridge	snorridgeb	0852229884	Edit	Delete
Missie	Meenan	mmeenanc	0832333171	Edit	Delete
Nolana	Chaudrelle	nchaudrelled	0899483563	Edit	Delete
Teddy	Petrak	tpetrake	0880761541	Edit	Delete
Beulah	Rubinovitch	brubinovitchf	0828279861	Edit	Delete
Edy	McMillan	emcmillang	0814159699	Edit	Delete
Sully	Giannoni	sgiannonih	0862229485	Edit	Delete
Ferrell	Scholtis	fscholtisi	0838141508	Edit	Delete
Ashil	Affleck	aaffleckj	0872380458	Edit	Delete
Robert	Beenham	rbeenhamk	0817732189	Edit	Delete
Sarena	Stoodley	sstoodleyl	0836360390	Edit	Delete

Imagen 3. Software terminado Spring Boot

3.5.2. Aplicación Stack MERN

La aplicación realizada con el Stack MERN se basa en los mismos principios y objetivos de la aplicación realizada en JAVA, sin embargo, cabe mencionar las diferencias importantes que yacen en esta aplicación y fueron posibles apreciar en comparación con la aplicación en java. Una diferencia clave es que java es un lenguaje de programación orientado a objetos y javascript por otra parte es orientado a funciones. Los requerimientos para poder ejecutar javascript son casi universales, puesto que, al ser un lenguaje que se ejecuta del lado del cliente, solo es necesario contar con un navegador web para ejecutarlo.

Si bien ambas aplicaciones son FRONT-END en el caso de la aplicación realizada con javascript fue necesaria la implementación de un BACK-END el cual ayudo con el procesamiento de los datos antes de ser enviados a la base de datos mongo. A continuación, se detalla la creación de esta aplicación:

En cuanto al desarrollo del BACK-END de esta app se refiere, fue necesario realizar la especificación de puertos por los cuales los métodos Http iban a ser escuchados. Es por ello que dentro de esta aplicación se encuentran dos directorios, uno es el client y el otro es el server.

En el directorio del server se encuentra un archivo llamado index.js en el cual se especifican los métodos http que se realizan hacia la base de datos, estos métodos serán llamados desde el lado del client y corresponden a las acciones del CRUD, aquí también se instancio el puerto 3001 por el cual se escucha la petición de dichos métodos y se realiza la conexión a la base de datos mongoDB.

Lo que respecta el lado del client, maneja así mismo el patrón de arquitectura MVC donde encontramos el directorio components.

En este directorio podemos encontrar los siguientes archivos con extensión .js los cuales realizan cada uno una función:

- Createprospects.js: En este archivo se encuentran instanciados los métodos y clases pertinentes para la creación de un nuevo prospecto.
- edit-prospect.components.js: Aquí se receipta la información previamente enviada por la clase ProspectTableRow, con la cual se procederá a realizar los cambios en un prospecto ya registrado en la base de datos.
- prospect-list.components.js: En esta clase se realiza la visualización de los datos obtenidos de la base de datos.
- ProspectTableRow: Esta clase devuelve una tabla con los datos obtenidos de la base de datos, también se obtienen las acciones de editar y eliminar a un prospecto de esta misma clase.

Todas estas clases envían peticiones http al lado del server en el cual se encuentran instanciados los métodos CRUD a realizar en la base de datos. Para realizar el llamado a estos métodos se utilizó la tecnología Axios la cual se encuentra debidamente especificada en el package.json de la aplicación.

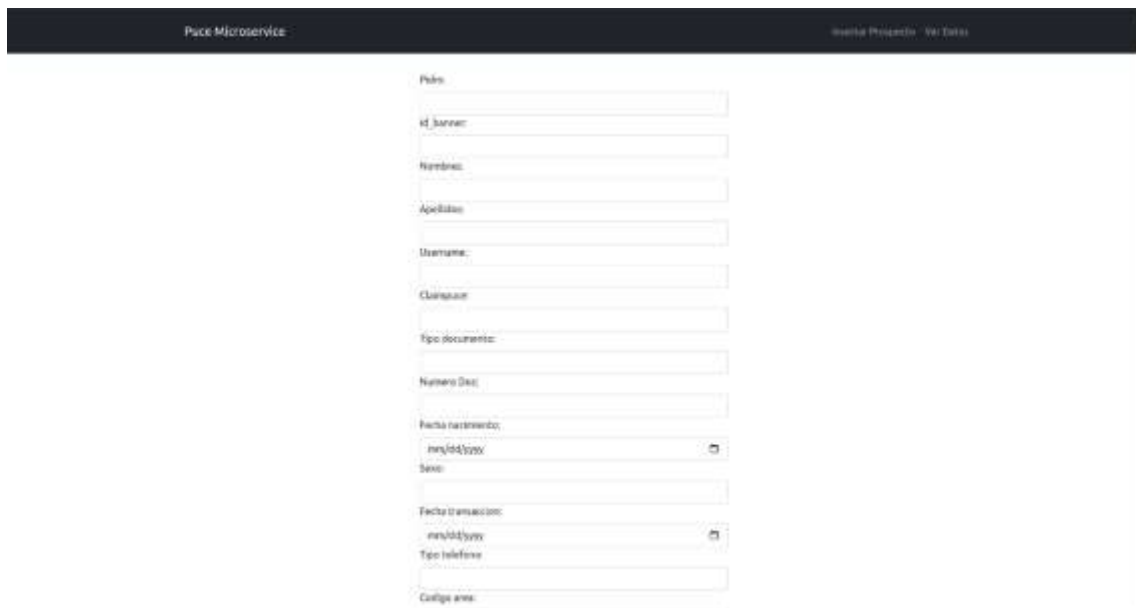


Imagen 4. Software terminado Stack MERN

3.6. Análisis de variable usabilidad

Una vez realizadas las aplicaciones en los dos lenguajes de programación se procedió a hacer el análisis de la variable de investigación usabilidad, en la cual intervienen los indicadores: Documentación, Comunidad, Soporte y Satisfacción.

Para el análisis de estas variables se aplicó el método de investigación no experimental, análisis correlacional, en la cual se infiere que el indicador documentación y comunidad están correlacionados, debido a que la documentación proporcionada del lenguaje de programación java ha provocado el crecimiento de la comunidad de este mismo lenguaje.

Según la fuente State of the Developer Nation de SlashData [31], la comunidad de java ha tenido un aumento siendo en 2020 un total 9,3 millones de usuarios de este lenguaje.

Por lo cual al momento de realizar la aplicación en java fue siempre oportuno encontrar foros, y documentación en los cuales basarse para la resolución de problemas. En cuanto al soporte Oracle se mantiene al día al proporcionar soporte a este lenguaje de programación a través de páginas oficiales y contactos telefónicos.

Realizando la correlación que existe entre estos indicadores se puede evaluar la variable usabilidad para el lenguaje de programación java de la siguiente manera:

Usabilidad:	2,6
Documentación:	3
Comunidad:	2
Soporte:	3
Satisfacción:	2,6

La usabilidad por lo tanto de este lenguaje de programación según las medidas de ponderación, se encuentra en niveles excelentes.

Por otra parte, en cuanto al lenguaje de programación javascript se encuentra de igual manera bien documentado, aunque la curva de aprendizaje de este lenguaje es más compleja, no quiere decir que es más complicado realizar aplicaciones con este lenguaje, se pueden desplegar aplicaciones en menor tiempo que en java y sin conocer o ser un experto en programación. El soporte técnico para este lenguaje es algo complicado de hallar, pero es compensado con su creciente comunidad la cual según la misma fuente State of the Developer Nation de SlashData[31], los usuarios de javascript ya son 14 millones. Por lo cual es siempre posible encontrar información en foros y sitios web para resolver inquietudes.

La tabla de evaluación de usabilidad según estos datos y la experiencia programando en este lenguaje es de la siguiente manera:

Usabilidad:	2,7
Documentación:	3
Comunidad:	3
Soporte:	2
Satisfacción:	2,7

Según el análisis realizado la usabilidad de javascript se encuentra en niveles excelentes y es ligeramente mejor a la usabilidad de java.

3.7. Análisis variable eficiencia

Para poder realizar el análisis de la eficiencia de cada lenguaje se realizaron pruebas de rendimiento: carga y estrés; a cada una de las aplicaciones, con los

resultados obtenidos se realizó la tabulación de los datos y se aplicó el método de investigación estadística descriptiva.

3.7.1. Pruebas de rendimiento

Esta prueba permite conocer los niveles de escalabilidad, fiabilidad y uso de recursos de un sistema, además de dar a conocer mediante comparación de resultados que sistema funciona mejor. JMeter es una herramienta de Testing que permite la tabulación de los datos resultantes de las pruebas, de la tabla resultante generada por este software se consideraron como más relevantes las variables Average la cual indica el promedio de latencia de las solicitudes, porcentaje de errores, rendimiento los cuales sirvieron para realizar la comparación entre ambas aplicaciones. El método de investigación

- Latencia: Cuando nos referimos a latencia hablamos de la diferencia de los tiempos en el cual un usuario envía una solicitud y el momento en que comenzó a recibirse una respuesta. Esta variable sirvió como indicador para evaluar el comportamiento de los tiempos en ambas aplicaciones.
- Porcentaje de Errores: El porcentaje de errores indica el promedio en que una solicitud realizada por un usuario no pudo ser realizada con éxito.
- Rendimiento: La variable rendimiento o Throughput en inglés, indica la cantidad de trabajo realizado con éxito en un periodo de tiempo definido.

Ambas aplicaciones fueron sometidas a pruebas de carga y estrés, en los siguientes apartados se puede apreciar los resultados de dichas pruebas empezando por la prueba de carga.

3.7.2. Prueba de carga

Las aplicaciones realizadas fueron sometidas a una prueba estándar de carga la cual concierne a 1000 usuarios simulados, realizando peticiones cada segundo, esta prueba se hace con la finalidad de constatar cuanta carga de usuarios es capaz de soportar una aplicación antes de devolver errores como resultado. Estas peticiones fueron realizadas para cada uno de los procesos de las aplicaciones, es decir insertar, leer, modificar y eliminar un prospecto de la base de datos. Con el

software JMeter es posible realizar un reporte resumido de las pruebas. Los datos resultantes fueron tabulados y graficados de manera que se puede apreciar el comportamiento de los tiempos, el porcentaje de errores y el rendimiento al realizar las peticiones.

En el Anexo 5. se ven representados los datos resultantes de la prueba de carga en la aplicación realizada con el Stack Mern en javascript. Al realizar la misma prueba en la aplicación creada con Spring Boot nos encontramos con los resultados en el Anexo 6.

Al comparar estos resultados tabulados, es posible percatarse que el margen de error de la herramienta Spring boot 77.78%, fue mucho mayor a la del Stack Mern el cual obtuvo un margen de error de 62.8%, indicando que el Stack Mern de javascript es más eficaz realizando las tareas propuestas.

La herramienta Stack Mern fue capaz de enviar más solicitudes que la herramienta spring boot, obteniendo así el Stack Mern un margen de rendimiento mucho más alto que spring boot.

De igual manera JMeter nos da a conocer de manera gráfica (Anexo 8 y Anexo 9.), el comportamiento de las variables anteriormente mencionadas. Aquí podemos observar que el Stack Mern es capaz de soportar un promedio de 1.156 peticiones por minuto con una desviación estándar de 2,33 segundos.

Mientras que spring boot es capaz de soportar un margen mucho mayor el cual es de 3.114 peticiones por minuto con una desviación estándar de 2.39 segundos.

Las pruebas de carga se realizan normalmente enviando una carga que contenga cierta cantidad de usuarios y luego una segunda carga con un volumen de usuarios mucho mayor, esto se hace con la finalidad de constatar cuanta carga es capaz de soportar una aplicación antes de devolver errores. En este caso solo fue necesaria una carga de 1000 usuarios para percatarse de cuando las aplicaciones empezaron a devolver errores como resultado. Aun así, el Stack Mern resulto ser mucho más eficiente que Spring boot en esta prueba de carga, ya que a pesar de que spring boot es capaz de soportar muchas más peticiones por minuto, el Stack Mern de javascript tiende a sufrir de menos de errores y tener mejor rendimiento que el framework de java.

3.7.3. Prueba de Estrés

El test de estrés es realizado con la finalidad de enviar una carga extrema al sistema y comprobar cuanto es capaz de soportar antes de que algún componente o el mismo sistema deje de responder. Estas pruebas tienen como objetivo averiguar si un componente es capaz de recuperarse por sí mismo o si el error se propaga hacia otros componentes provocando así el fallo total del sistema. Una vez que se identifica el componente que ha fallado, se puede proceder a la mejora del mismo y por lo tanto la mejora del Throughput (rendimiento) de todo el sistema.

Al igual que la prueba de carga, esta prueba fue realizada con un total de 1000 usuarios realizando peticiones cada segundo a cada uno de los métodos de las aplicaciones. La aplicación Java fue la primera en someterse a esta prueba con la cual se obtuvo los resultados tabulados en el Anexo 6.

La aplicación realiza con el Stack Mern sometida a la misma prueba, arrojó los resultados correspondientes a el Anexo 7.

Con esta prueba nos pudimos percatar que el margen de error del Stack MERN fue ligeramente menor al momento de realizar la consulta de datos, la diferencia clave entre ambas, fue el promedio de la latencia y el rendimiento que fueron capaces de llevar a cabo durante dicha prueba. En el cual se puede constatar que el rendimiento de Spring boot fue mucho mayor al rendimiento de Stack Mern y una latencia mucho menor.


3.8. Comparación

Una vez evaluadas ambas herramientas, a través del test de rendimiento y los indicadores establecidos para medir la usabilidad, fue posible realizar la comparación entre ambas y determinar, que Stack resulto más eficiente al momento de procesar datos.

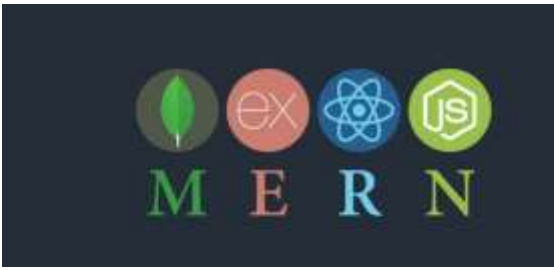
Para esta comparación se utilizó la ficha de evaluación de software establecida en el apartado 2.6 Técnicas de procesamiento y análisis de datos.

Las pruebas de carga y estrés realizadas a cada una de las herramientas, ayudaron a tener una visión más concreta del funcionamiento y eficiencia de ambas. En la prueba de carga el Stack Mern resulto superior a Spring boot, sin embargo, al realizar la prueba de estrés, spring boot obtuvo mejores resultados en comparación a Stack Mern. Bajo criterios de experiencia en la utilización de ambas herramientas y resultados de las pruebas, ambos pesos de las herramientas en cuanto a la variable eficiencia fue de 2,5. Por lo que las fichas de evaluación se establecieron para ambas aplicaciones fueron establecidas con los mismos parámetros de eficiencia, y con una diferencia de 0.1 en los parámetros de usabilidad.

A lo largo de esta investigación ambas aplicaciones mostraron las ventajas y desventajas que proveen al momento de llevar a cabo un proyecto. Sin embargo, las pruebas de rendimiento han demostrado que la herramienta, Stack Mern provee al momento de desarrollo mayor velocidad y mejores tiempos de respuesta al momento de ser ejecutada. Esto no quiere decir que la herramienta de java Spring Boot sea por lo tanto una mala elección o una herramienta deficiente, puesto que, aunque sus tiempos de respuesta fueron inferiores, java sigue siendo un lenguaje muy robusto capaz de proveer a los usuarios herramientas y documentación para llevar a cabo un proyecto de desarrollo de software.

Ficha de evaluación de software		
Producto: Spring Boot	Versión: 2.5.3	Compañía/creador: Rod Johnson
Descripción General: Spring boot es un framework que facilita la creación de aplicaciones independientes basadas en spring.	Imagen de la herramienta: 	
Pesos		
Eficiencia de la herramienta:	2,5	
Usabilidad:	2,6	
Requerimientos del sistema: jdk, Apache Tomcat		
Licencia:	Página de descarga: https://start.spring.io/	

Ficha de evaluación 1 Spring boot

Ficha de evaluación de software		
Producto: Stack Mern	Versión: Unknow	Compañía/creador: None
Descripción General: Este conjunto de herramientas ofrecen un marco de desarrollo framework el cual se hace cada vez más común en la comunidad de desarrolladores web.	Imagen de la herramienta: 	
Pesos		
Eficiencia de la herramienta:	2,5	
Usabilidad:	2,7	

Requerimientos del sistema: Navegador web	
Licencia:	Página de descarga: múltiples webs

Ficha de evaluación 2. Stack MERN

4. Capítulo 4: Discusión

La presente investigación fue enfocada en la comparación de los lenguajes de programación java y javascript. Esta comparación nace con la idea de poner a prueba estos lenguajes y obtener como resultado la selección de un lenguaje de programación y sus stacks para el desarrollo de software, según sea el caso, un proyecto en marcha.

Hay investigaciones existentes en las cuales estos lenguajes son puestos a trabajar en conjunto para realizar un mismo objetivo. Ejemplos de esto se encuentran en investigaciones como la de Milán Vidakovic y compañía en la cual el objetivo fue buscar una solución para la ejecución de javascript en servidores de aplicaciones Java EE. [32]

La investigación “Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics”[33] ofrece una amplia vista de lo que son los servicios web Rest con la cual fue posible tomar el enfoque de basado en servicios web Restful.

Un enfoque similar se obtuvo en la investigación [34] en la cual se realiza la comparación de distintos lenguajes de programación al momento de utilizar el método JSON Parser. Esta investigación evalúa la eficacia de este método en los distintos lenguajes mencionados en la misma. Como resultado obtuvieron que Javascript fue el más eficaz al momento de analizar un archivo JSON. Este resultado se acopla a los resultados de esta investigación en la cual se hace mención a que javascript fue más eficaz al momento de manejar y procesar los datos obtenidos desde la base de datos, los cuales son tipo JSON de igual manera.

5. Capítulo 5: Conclusiones

A lo largo de este trabajo de investigación, se han recopilado datos, los cuales se volvieron pertinentes a la hora de evaluar las herramientas propuestas para la comparación de ambos lenguajes de programación. Con toda la información recopilada se pudo arribar a las siguientes conclusiones:

- Los servicios web Restful se han convertido en el esqueleto de casi todas las aplicaciones y páginas web en la actualidad. Esto, debida a la facilidad de uso y transportación de datos mediante los métodos que estos servicios conllevan consigo.
- Fue posible no solo encontrar diferencias entre ambos softwares, si no también similitudes las cuales forman parte de las buenas prácticas de programación, como son la reusabilidad de código, la arquitectura y la inyección de dependencias.
- La ISO/IEC 25000 o también conocida como SquaRE, sirvió como guía que permitió evaluar de manera satisfactoria ambas aplicaciones desarrolladas en los distintos lenguajes de programación.
- A través de la revisión de otras investigaciones se llegó a la conclusión de que no solo es posible analizar y comparar los distintos lenguajes de programación, sino también es posible el análisis y la evolución que estos sufren a través del tiempo. La aparición de nuevas tecnologías que poco a poco van desplazando a herramientas ya conocidas pueden ser puestas bajo análisis y comparación, tales herramientas como Maven que actualmente es más tendencia que Gradle.
- Al someter a pruebas ambas aplicaciones, se pudo llegar a la conclusión de que, aunque ambas son herramientas muy bien diseñadas y de alta funcionalidad. La herramienta de Stack Mern permitió el manejo más simplificado de datos, permite realizar aplicaciones en muy poco tiempo, veloces, eficientes y cumpliendo con requerimientos actuales de software.

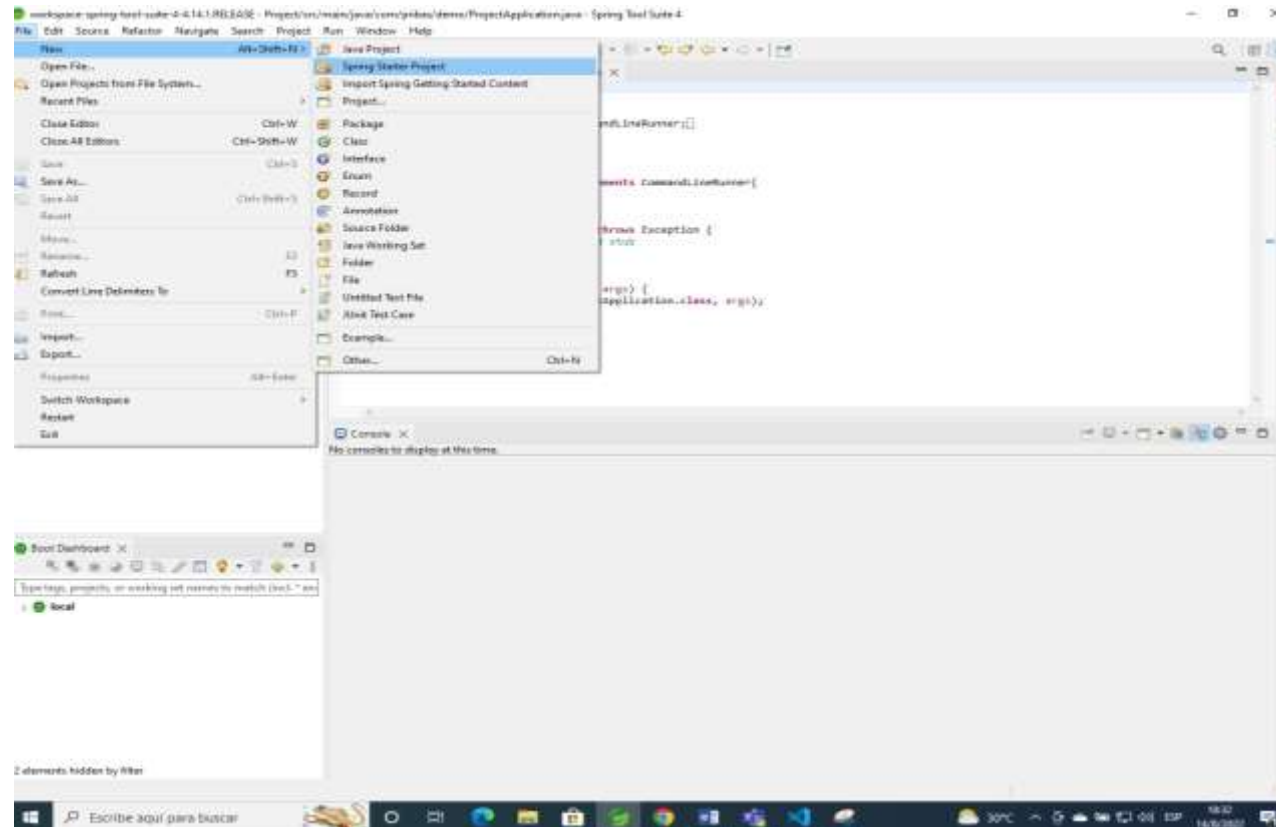
6. Capítulo 6: Recomendaciones

Se recomienda siempre estar al día con los procesos y calidad de software, así como las tendencias a nuevas tecnologías. Las herramientas de Testing también son parte esencial en el desarrollo de software, por lo que es recomendable tener conocimientos de Testing para evaluar el software realizado y medir el rendimiento del mismo. De esta manera es posible verificar bugs y cuellos de botella a lo largo del desarrollo de software.

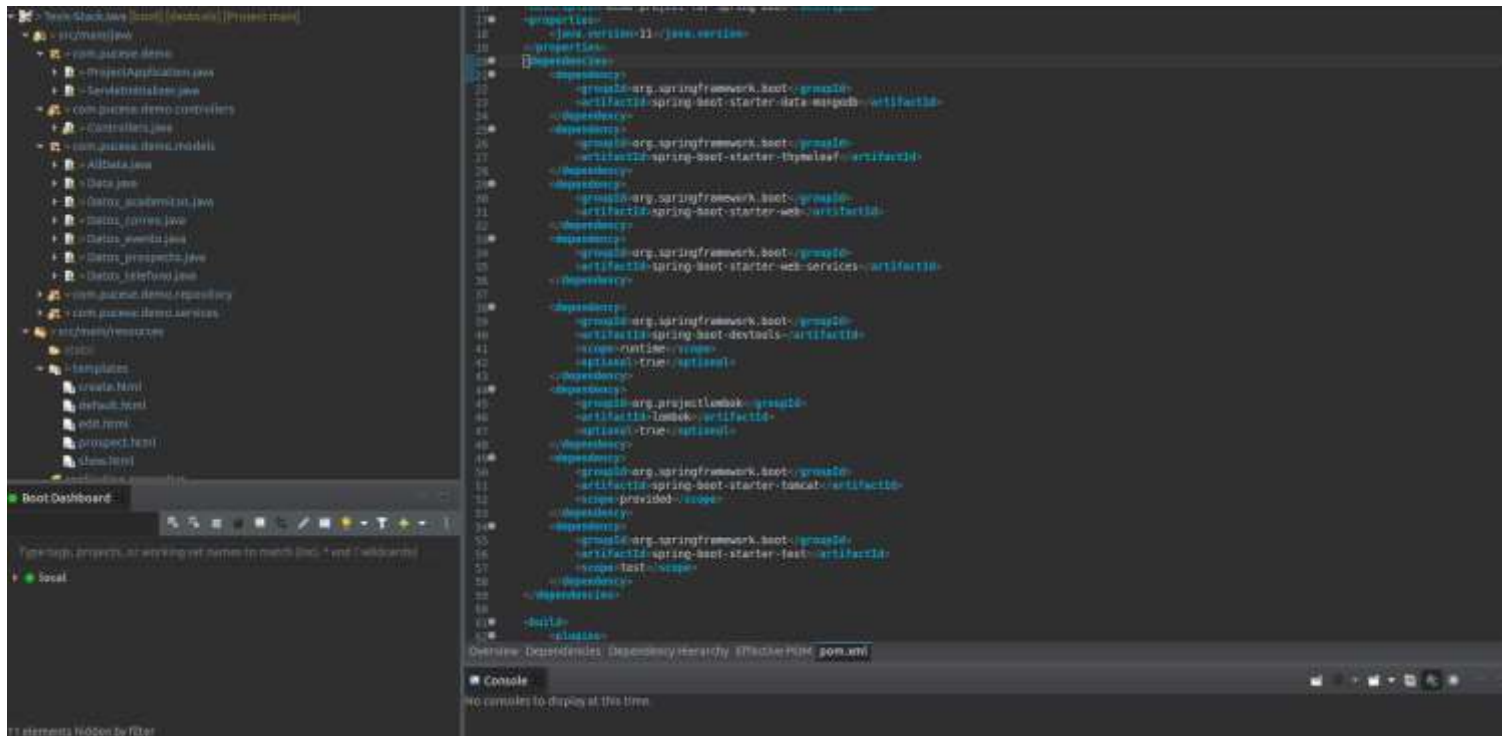
Los test siempre deben ser realizados a medida que el software valla siendo desarrollado y no al final. Pues esto puede generar inconvenientes graves en la etapa final de entrega de un proyecto. La verificación de requerimientos y características de un lenguaje de programación, así como sus Stacks, deben ser cuidadosamente analizados antes de poner en marcha un proyecto, puesto que la selección de las herramientas adecuadas puede mejorar los tiempos de desarrollo o, al contrario, si las herramientas resultan inadecuadas puede llevar a un proyecto al fracaso.

7. Anexos

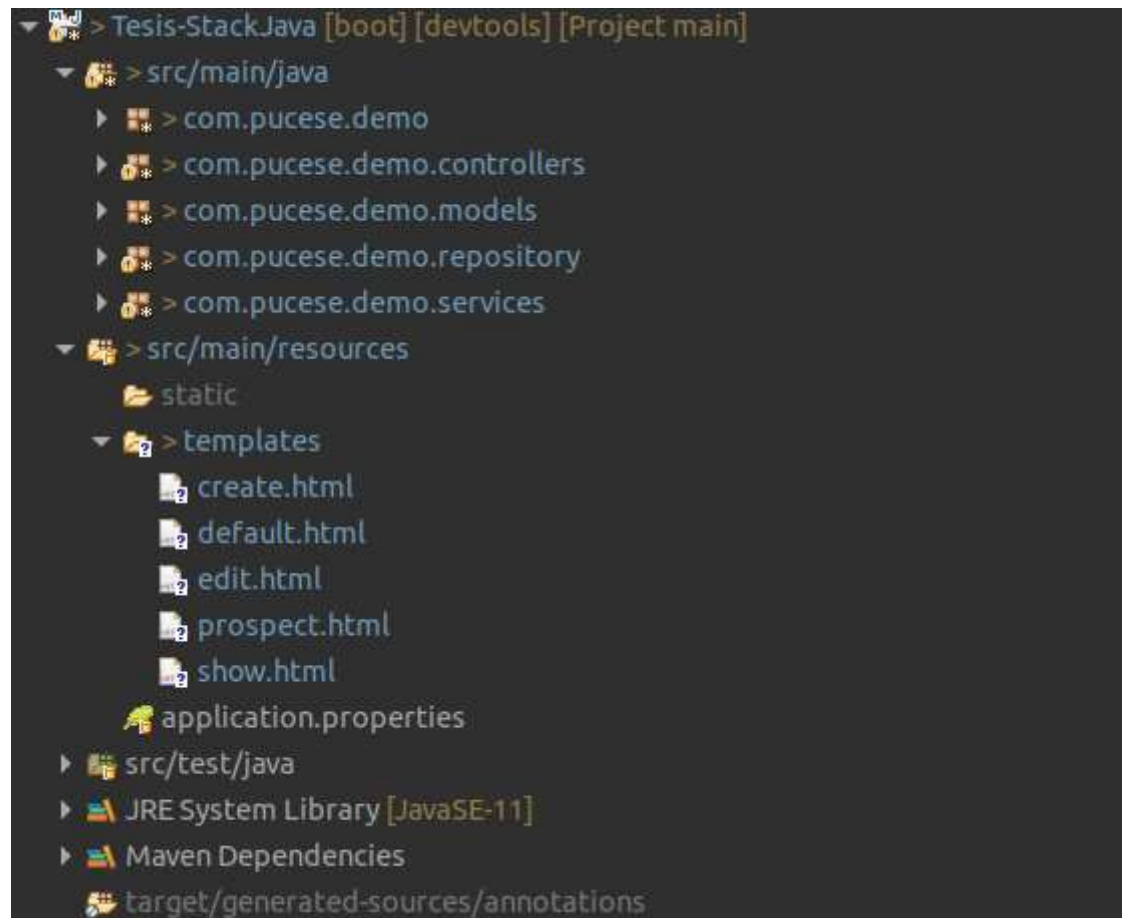
7.1. Construcción del software Spring Boot



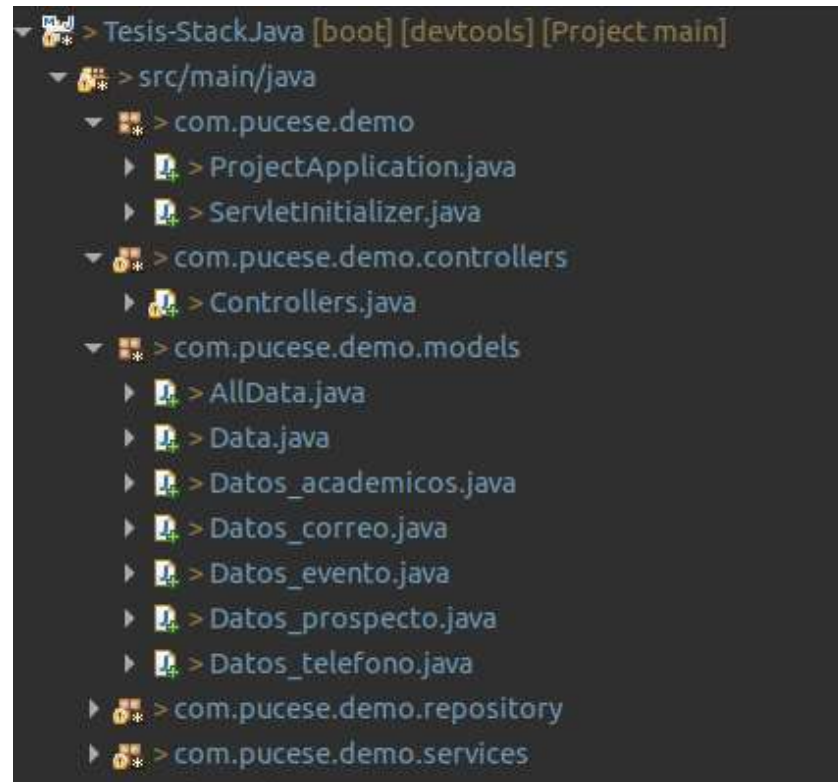
Paso 1 construcción de la aplicación spring boot



Paso 2. Adición de librerías al proyecto en el pom.xml



Paso 3. Arquitectura del software MCV



Paso 4. Definición de los modelos, controladores, servicios y repositorio

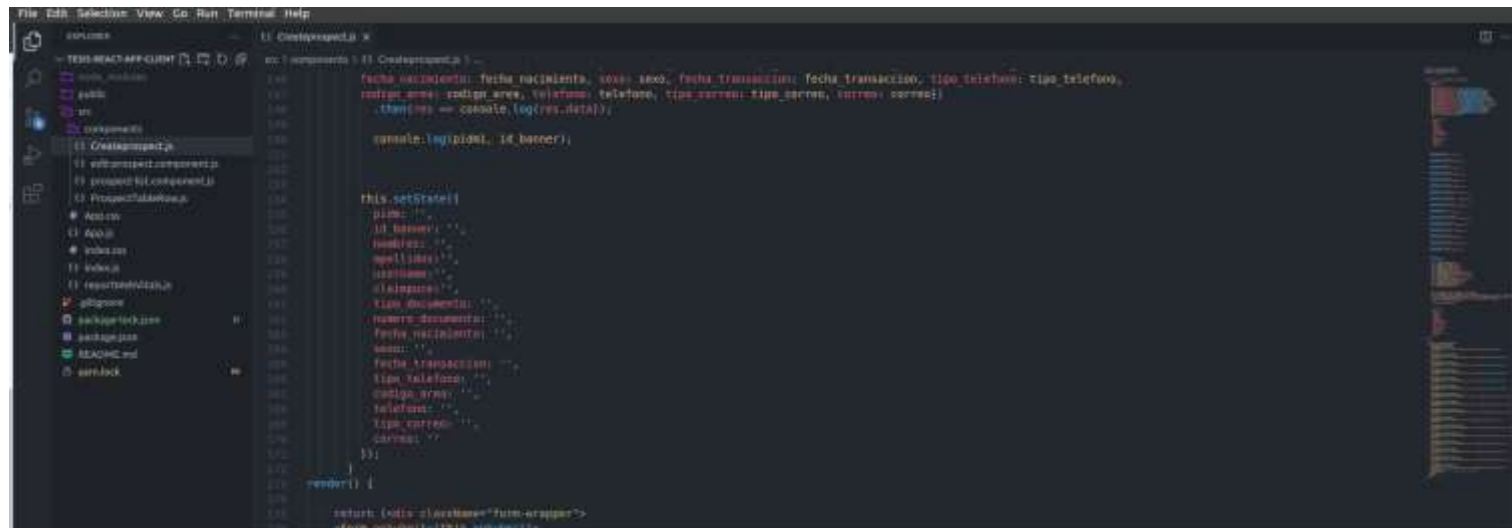
7.2. Construcción del software Stack MERN

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Andres> npx create-react-app tesis-stackmern
```

Paso 1. Comando para creación de la aplicación Stack Mern client Side



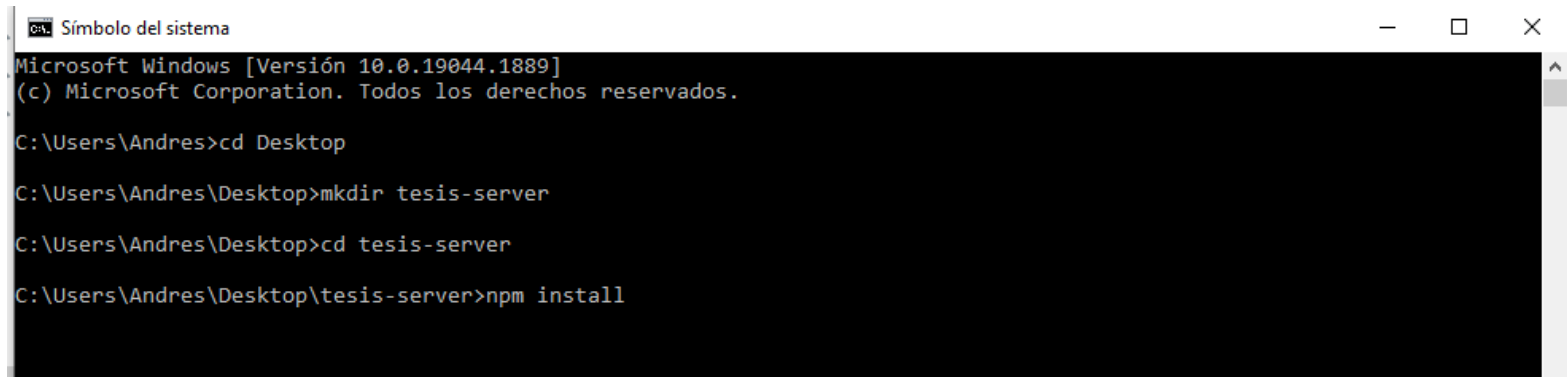
```
File Edit Selection View Go Run Terminal Help
tesis-react-app-client x
  src
    components
      11 CreateComponent.js
      11 subcomponentcomponent.js
      11 propsComponent.js
      11 PropComponent.js
  # src
  11 App.js
  # index.css
  11 index.js
  11 reportWebVitals.js
  # setupTests
  11 setupTests.js
  # package.json
  # README.md
  # yarn.lock

fecha nacimiento: fecha nacimiento, sexo: sexo, fecha transaccion: fecha transaccion, tipo transaccion: tipo transaccion,
codigo_area: codigo_area, telefono: telefono, tipo_correo: tipo_correo, correo: correo);
.then(res => console.log(res.data));

console.log(id, id_banner);

this.setState({
  id_banner: '',
  nombre: '',
  apellido: '',
  documento: '',
  cuit: '',
  tipo_documento: '',
  nombre_documento: '',
  fecha_nacimiento: '',
  sexo: '',
  fecha_transaccion: '',
  tipo_transaccion: '',
  tipo_telefono: '',
  codigo_area: '',
  telefono: '',
  tipo_correo: '',
  correo: ''
});
}
}
render() {
return (<div className="form-wrapper">
  <form onSubmit={this.handleSubmit}>
```

Paso 2. Creación de componentes client Side y funciones para consumo REST



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.1889]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Andres>cd Desktop

C:\Users\Andres\Desktop>mkdir tesis-server

C:\Users\Andres\Desktop>cd tesis-server

C:\Users\Andres\Desktop\tesis-server>npm install
```

Paso 3. Creación de Server Side con nodejs

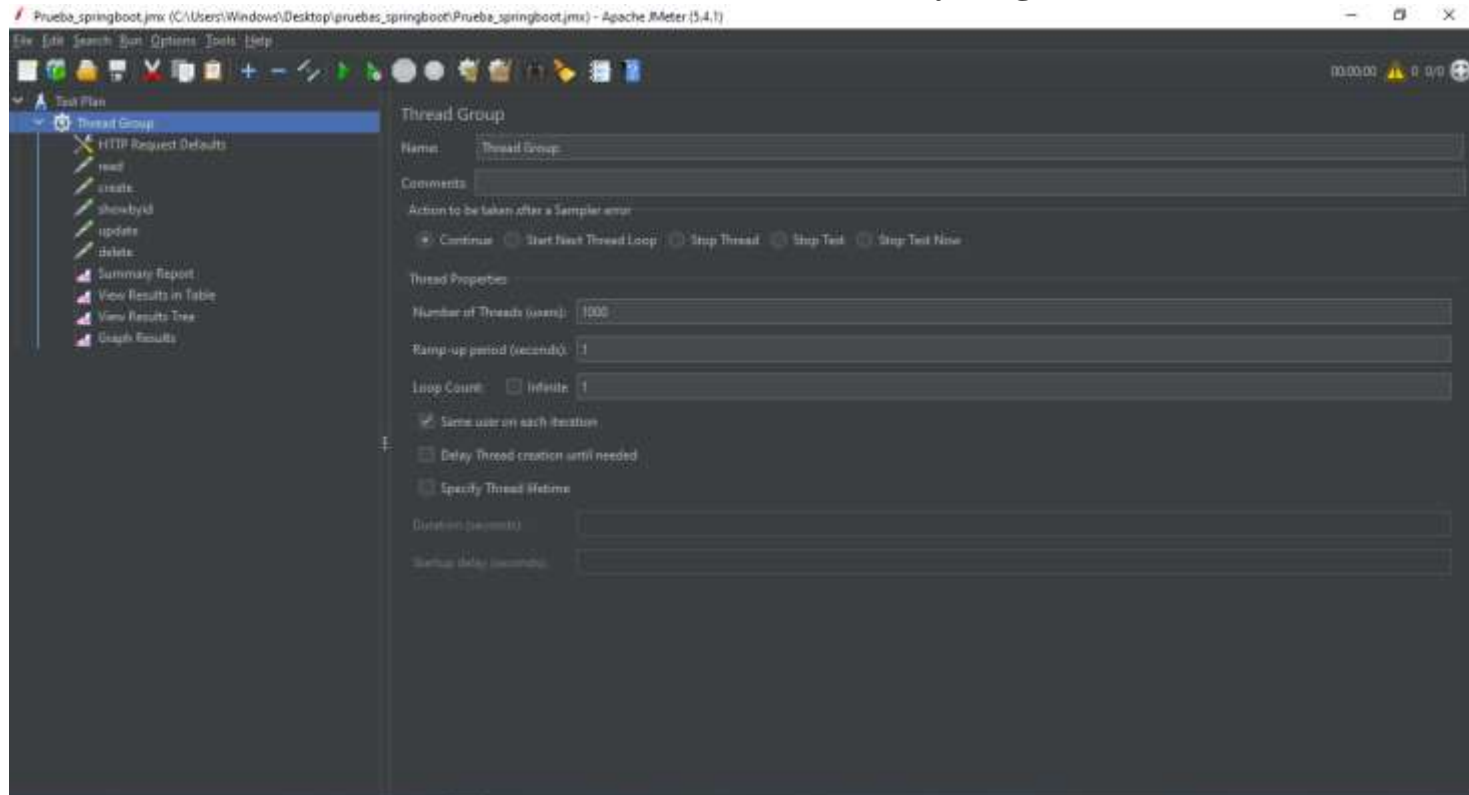
```

1  const express = require('express');
2  const mongoose = require('mongoose');
3  const cors = require('cors');
4  const app = express();
5
6  const PrspectoModel = require('./models/Prspecto');
7
8  mongoose.Promise = global.Promise;
9  mongoose.connect('mongodb+srv://andres2094:845061122aa0@cluster0.mongodb.net/Data_Tesis?retryWrites=true&maxJority', {
10    useNewUrlParser: true,
11  });
12
13  app.use(express.json());
14  app.use(cors());
15
16  app.post('/insert', async (req, res) => {
17    const pidm = req.body.pidm;
18    const id_banner = req.body.id_banner;
19    const nombres = req.body.nombres;
20    const apellidos = req.body.apellidos;
21    const username = req.body.username;
22    const claspape = req.body.claspape;
23    const tipo_documento = req.body.tipo_documento;
24    const numero_documento = req.body.numero_documento;
25    const fecha_nacimiento = req.body.fecha_nacimiento;
26    const sexo = req.body.sexo;
27    const fecha_transaccion = req.body.fecha_transaccion;
28    const tipo_telefono = req.body.tipo_telefono;
29    const codigo_area = req.body.codigo_area;
30    const telefono = req.body.telefono;
31    const tipo_carrea = req.body.tipo_carrea;
32    const correo = req.body.correo;
33    const prospect = new PrspectoModel({
34      pidm: pidm,
35      id_banner: id_banner,
36      nombres: nombres,
37      apellidos: apellidos,
38      username: username,
39      claspape: claspape,
40      tipo_documento: tipo_documento,
41      numero_documento: numero_documento,
42      fecha_nacimiento: fecha_nacimiento,
43      sexo: sexo,
44      fecha_transaccion: fecha_transaccion,
45      datos_telefono: {

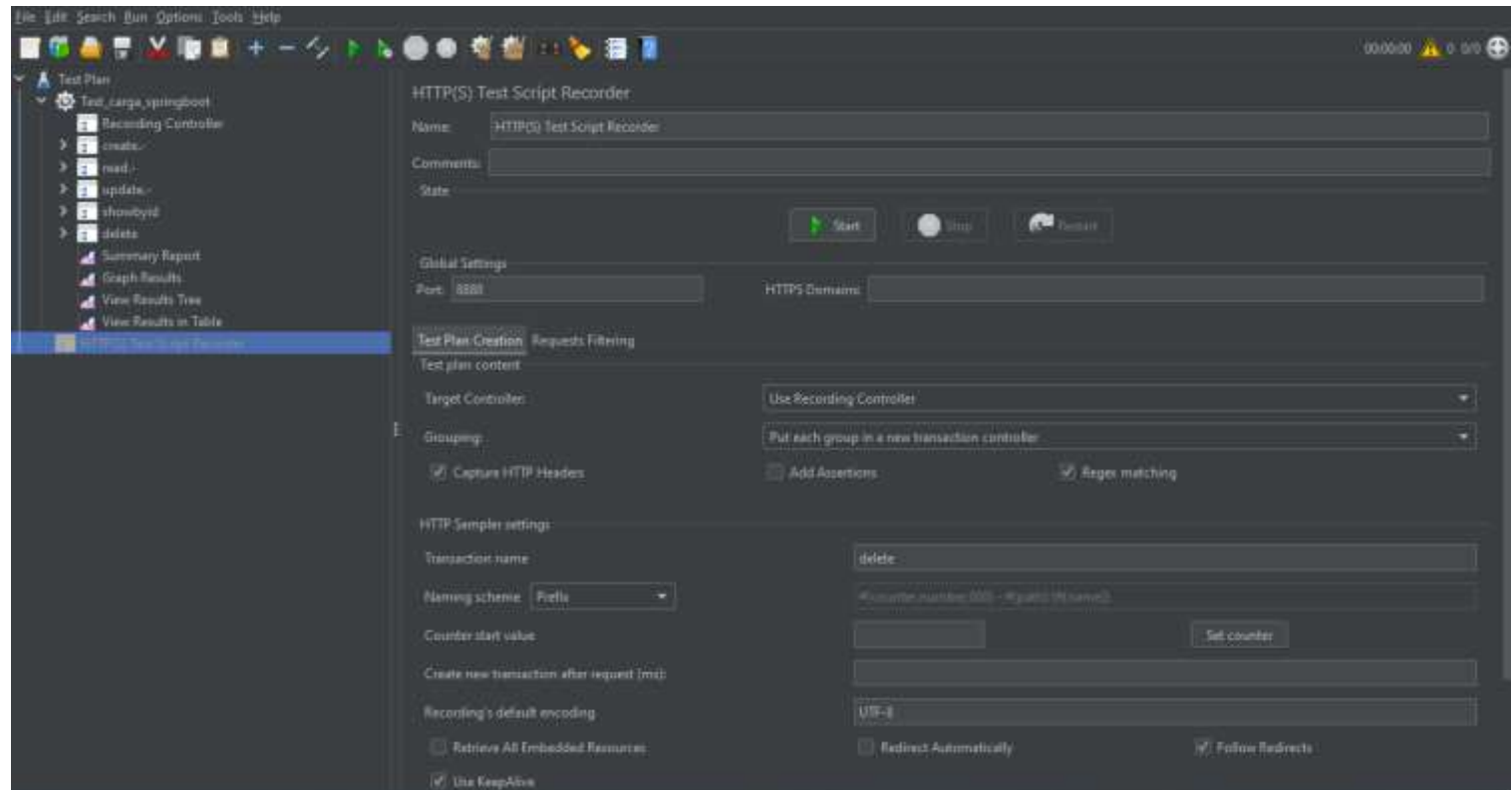
```

Paso 4. Conexión a la base de datos MongoDB y servicios CRUD server Side nodejs

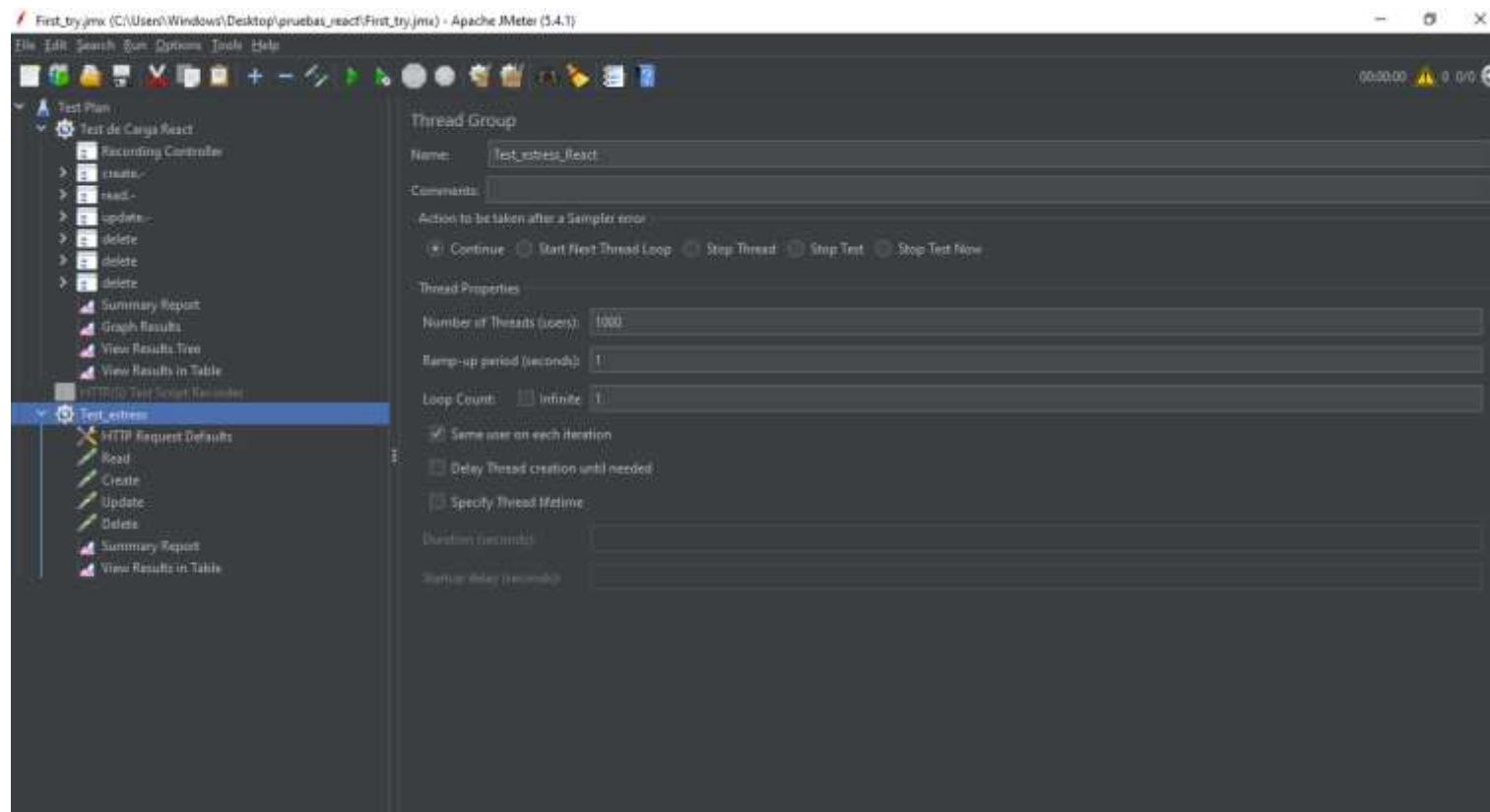
7.3. Pruebas de rendimiento, estrés y carga



Anexo 1. Preparación test de estrés spring boot



Anexo 2. Test de carga Spring Boot



Anexo 3. Preparación test de carga y estrés Stack Mern

First_try.jmx (C:\Users\Windows\Desktop\pruebas_react\First_try.jmx) - Apache JMeter (3.4.1)

File Edit Search Run Options Tools Help

00:04:50 9/2000

Test Plan

- Prueba de Carga Spring Boot
 - Recording Controller
 - read-
 - read-
 - update-
 - delete-
 - Summary Report
 - Graph Results
 - View Results Tree
 - View Results in Table**
 - HTTP(s) Test Script Recorder

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Log/Display Only Errors Successes

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
2963	19:39:10.225	Prueba de Car...	read-	36620	✓	288	332	36620	0
2964	19:39:48.843	Prueba de Car...	update-/upda...	4	✓	329	390	0	0
2965	19:39:09.630	Prueba de Car...	read-/read-5	37230	✓	288	332	37230	0
2966	19:39:09.639	Prueba de Car...	read-	37230	✓	288	332	37230	0
2967	19:39:48.839	Prueba de Car...	update-/upda...	3	✓	329	390	0	0
2968	19:39:13.657	Prueba de Car...	update-/upda...	33545	✓	265	770	33545	0
2969	19:39:13.633	Prueba de Car...	update-/upda...	33549	✓	264	1160	33545	0
2970	19:39:13.863	Prueba de Car...	update-/upda...	33292	✓	265	770	33292	0
2971	19:39:13.862	Prueba de Car...	update-	33292	✓	264	1160	33292	0
2972	19:39:47.182	Prueba de Car...	delete/delete/...	14	✓	265	372	0	0
2973	19:39:47.185	Prueba de Car...	delete/delete/...	11	✓	265	372	0	0
2974	19:39:10.229	Prueba de Car...	read-/read-6	37205	✓	288	332	37205	0
2975	19:39:10.229	Prueba de Car...	read-	37205	✓	288	332	37205	0
2976	19:39:02.811	Prueba de Car...	create-/insert-5	44623	✓	276	776	44623	0
2977	19:39:01.070	Prueba de Car...	create-	44363	✓	605	1167	44623	0
2978	19:39:02.653	Prueba de Car...	create-/insert-5	44482	✓	276	776	44482	0
2979	19:39:01.108	Prueba de Car...	create-	46328	✓	605	1167	44482	0
2980	19:39:03.099	Prueba de Car...	create-/insert-5	44338	✓	276	776	44138	0
2981	19:39:01.151	Prueba de Car...	create-	46285	✓	605	1167	44138	0
2982	19:39:03.170	Prueba de Car...	create-/insert-5	44299	✓	276	776	44299	0
2983	19:39:03.022	Prueba de Car...	create-	46288	✓	605	1167	44299	1523

Scroll automatically? Child samples? No. of Samples: 11199 Latest Sample: 4238

Activar Windows
Ir a Configuración para activar Windows.

Anexo 5. Prueba de carga Spring Boot

Prueba_estres.jmx (C:\Users\Windows\Desktop\pruebas_springboot\Prueba_estres.jmx) - Apache JMeter (5.4.1)

File Edit Search Run Options Tools Help

00:00:18 0 0/1000

Test Plan

- Thread Group
 - HTTP Request Defaults
 - Read
 - Delete
 - Update
 - Create
 - Summary Report
 - Graph Results
 - View Results in Table**

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: Errors Successes

Sample #	Start Time	Thread Name	Label	Sample Time...	Status	Bytes	Sent Bytes	Latency	Connect Time...
352	19:27:46.663	Thread Group ...	Read	4054	✓	2741	0	0	4054
353	19:27:46.666	Thread Group ...	Read	4051	✓	2741	0	0	4051
354	19:27:46.671	Thread Group ...	Read	4046	✗	2741	0	0	4046
355	19:27:46.668	Thread Group ...	Read	4049	✓	2741	0	0	4049
356	19:27:46.658	Thread Group ...	Read	4059	✗	2741	0	0	4059
357	19:27:46.664	Thread Group ...	Read	4053	✗	2741	0	0	4053
358	19:27:46.662	Thread Group ...	Read	4055	✓	2741	0	0	4055
359	19:27:46.659	Thread Group ...	Read	4059	✗	2741	0	0	4058
360	19:27:46.667	Thread Group ...	Read	4051	✓	2741	0	0	4051
361	19:27:46.672	Thread Group ...	Read	4058	✗	2741	0	0	4058
362	19:27:46.680	Thread Group ...	Read	4050	✓	2741	0	0	4050
363	19:27:46.694	Thread Group ...	Read	4046	✓	2741	0	0	4046
364	19:27:46.677	Thread Group ...	Read	4053	✗	2741	0	0	4053
365	19:27:46.682	Thread Group ...	Read	4048	✗	2741	0	0	4048
366	19:27:46.678	Thread Group ...	Read	4053	✗	2741	0	0	4053
367	19:27:46.675	Thread Group ...	Read	4056	✗	2741	0	0	4056
368	19:27:46.673	Thread Group ...	Read	4058	✗	2741	0	0	4058
369	19:27:46.681	Thread Group ...	Read	4050	✗	2741	0	0	4050
370	19:27:46.696	Thread Group ...	Read	4050	✗	2741	0	0	4050
371	19:27:46.690	Thread Group ...	Read	4056	✗	2741	0	0	4056
372	19:27:46.702	Thread Group ...	Read	4044	✗	2741	0	0	4044

Scroll automatically? Child samples? No of Samples: 4000 Latest Sample: 4075 Average: 4000 Duration: 0

Activar Windows
Ve a Configuración para activar Windows.

Anexo 6. Prueba de estrés Spring Boot

First_try.jmx (C:\Users\Windows\Desktop\pruebas_react\First_try.jmx) - Apache JMeter (3.4.1)

File Edit Search Run Options Tools Help

00:04:30 0 0/2000

Test Plan

- Prueba de Carga React
 - Recording Controller
 - create-
 - read-
 - update-
 - delete-
 - Summary Report
 - Graph Results
 - View Results Tree
 - View Results in Table**
 - HTTP(S) Test Script Recorder
 - Test_estructura
 - HTTP Request Defaults
 - Read
 - Create
 - Update
 - Delete
 - Summary Report
 - View Results in Table

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Log/Display Only Errors Successes

Sample #	Start Time	Thread Name	Label	Sample Time(L)	Status	Bytes	Sent Bytes	Latency	Connect Time(L)
1335	19:39:00,750	Prueba de Car...	create-/insert-5	8888	✓	276	776	8888	0
1336	19:39:00,760	Prueba de Car...	create-	9049	✓	605	1167	8888	1
1337	19:39:00,765	Prueba de Car...	create-/insert-5	8874	✓	276	776	8874	0
1338	19:39:00,767	Prueba de Car...	create-	9878	✓	605	1167	8874	1
1339	19:39:00,770	Prueba de Car...	create-/insert-5	8869	✓	276	776	8869	0
1340	19:39:00,753	Prueba de Car...	create-	9865	✓	605	1167	8869	1
1341	19:39:00,775	Prueba de Car...	create-/insert-5	8861	✓	276	776	8861	0
1342	19:39:00,779	Prueba de Car...	create-/insert-5	8861	✓	276	776	8861	0
1343	19:39:00,723	Prueba de Car...	create-	9914	✓	605	1167	8861	0
1344	19:39:00,730	Prueba de Car...	create-	9901	✓	605	1167	8861	1
1345	19:39:00,309	Prueba de Car...	create-/insert-5	8862	✓	276	776	8862	0
1346	19:39:00,717	Prueba de Car...	create-	9952	✓	605	1167	8862	1
1347	19:39:00,814	Prueba de Car...	create-/insert-5	8858	✓	276	776	8858	0
1348	19:39:00,703	Prueba de Car...	create-	9968	✓	605	1167	8858	1
1349	19:39:00,821	Prueba de Car...	create-/insert-5	8852	✓	276	776	8852	0
1350	19:39:00,667	Prueba de Car...	create-	9984	✓	605	1167	8852	0
1351	19:39:00,827	Prueba de Car...	create-/insert-5	8848	✓	276	776	8847	0
1352	19:39:00,673	Prueba de Car...	create-	10001	✓	605	1167	8847	0
1353	19:39:02,815	Prueba de Car...	create-/insert-5	7873	✓	276	776	7873	0
1354	19:39:02,813	Prueba de Car...	create-/insert-5	7874	✓	276	776	7874	0
1355	19:39:02,811	Prueba de Car...	create-/insert-5	7875	✓	276	776	7875	0

Scroll automatically Child samples No of Samples: 1000 Latest Sample: 8238

Activar Windows
 a Configuración para activar Windows.

Anexo 7. Prueba de carga Stack Mern

First_try.jmx (C:\Users\Windows\Desktop\pruebas_react\First_try.jmx) - Apache JMeter (5.4.1)

File Edit Search Run Options Tools Help

00:04:30 0 0/2000

Test Plan

- Prueba de Carga React
 - HTTP(S) Test Script Recorder
 - Test_estress
 - HTTP Request Defaults
 - Read
 - Create
 - Update
 - Delete
 - Summary Report
 - View Results in Table

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file:

Filename: Browse... Log/Display Only: Errors Successes Configure

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	19:39:00.748	Test_estress 2-3	Read	4046	✓	2741	0	0	4046
2	19:39:00.748	Test_estress 2-2	Read	4046	✗	2741	0	0	4046
3	19:39:00.758	Test_estress 2-15	Read	4068	✓	2741	0	0	4068
4	19:39:00.758	Test_estress 2-16	Read	4068	✗	2741	0	0	4068
5	19:39:00.756	Test_estress 2-13	Read	4070	✗	2741	0	0	4070
6	19:39:00.755	Test_estress 2-12	Read	4071	✓	2741	0	0	4071
7	19:39:00.759	Test_estress 2-17	Read	4067	✗	2741	0	0	4067
8	19:39:00.752	Test_estress 2-7	Read	4075	✓	2741	0	0	4075
9	19:39:00.764	Test_estress 2-25	Read	4063	✗	2741	0	0	4063
10	19:39:00.748	Test_estress 2-1	Read	4079	✓	2741	0	0	4079
11	19:39:00.763	Test_estress 2-23	Read	4064	✓	2741	0	0	4064
12	19:39:00.751	Test_estress 2-4	Read	4077	✓	2741	0	0	4077
13	19:39:00.754	Test_estress 2-10	Read	4074	✓	2741	0	0	4074
14	19:39:00.751	Test_estress 2-6	Read	4077	✓	2741	0	0	4077
15	19:39:00.765	Test_estress 2-26	Read	4063	✓	2741	0	0	4063
16	19:39:00.752	Test_estress 2-8	Read	4077	✓	2741	0	0	4077
17	19:39:00.754	Test_estress 2-11	Read	4075	✓	2741	0	0	4075
18	19:39:00.752	Test_estress 2-5	Read	4078	✗	2741	0	0	4077
19	19:39:00.767	Test_estress 2-28	Read	4063	✓	2741	0	0	4063
20	19:39:00.753	Test_estress 2-9	Read	4077	✓	2741	0	0	4077
21	19:39:00.764	Test_estress 2-24	Read	4066	✗	2741	0	0	4066

Scroll automatically? Child samples? No of Samples: 4000 Latest Sample: 4095 Average: 4063 Deviation: 0

Anexo 8. Prueba de estrés Stack Mern

7.4. Lista de Tablas

Label	# Samples	Average	Error %	Throughput
create./insert-4	1000	2812	0.066	17.743.080.198.722.400
create./insert-5	1000	18859	0.0	19.990.804.230.054.100
create.-	1000	21672	0.066	19.340.489.314.379.600
read./read-6	1000	643457	0.814	1.332.091.823.753.590
read.-	1000	643457	0.814	1.332.088.274.825.790
update./update-7	1000	3423	0.814	21.199.825.313.439.400
update./update-8	1000	54938	0.923	21.015.599.879.790.700
update.-	1000	58361	0.923	21.007.829.618.098.600
delete/read-9	1000	38936	1.0	21.279.357.533.637.300
delete	3000	16310	0.666	6.229.986.169.430.700
delete/delete/611959c70d046a0d6e0f76cd-10	1000	4092	1.0	5.570.720.294.134.030
delete/delete/611959c70d046a0d6e0f76cd-11	1000	4086	1.0	55.558.642.146.785.900
delete/prospect-list-12	1000	1815	0.0	583.192.395.171.167
TOTAL	15000	102989	0.628	19.274.507.536.332.400

Anexo 9. Tabla de resultados Prueba de carga Stack Mern.

Label	Sample	Average	Error%	Throughput
create.-/save-25	1000	75115	0.0	1.091.726.893.600.290
create.-	1000	75115	0.0	10.916.792.209.777.000
read.-/show/61100ca33437811fa8175794-31	1000	26100	1.0	79.942.441.442.161.600
read.-/edit/61100ca33437811fa8175794-33	1000	21105	1.0	9.065.196.896.076.580
read.-	1000	47205	1.0	6.956.086.227.644.870
update.-/update-35	1000	18047	1.0	11.380.059.859.114.800
update.-	1000	18047	1.0	11.380.059.859.114.800
delete.-/delete-41	1000	16770	1.0	13.484.721.810.189.000
delete.-	1000	16770	1.0	13.484.721.810.189.000
TOTAL	9000	34919	0.7777777777777778	5.190.042.039.340.510

Anexo 10. Tabla de resultados prueba de carga Spring Boot

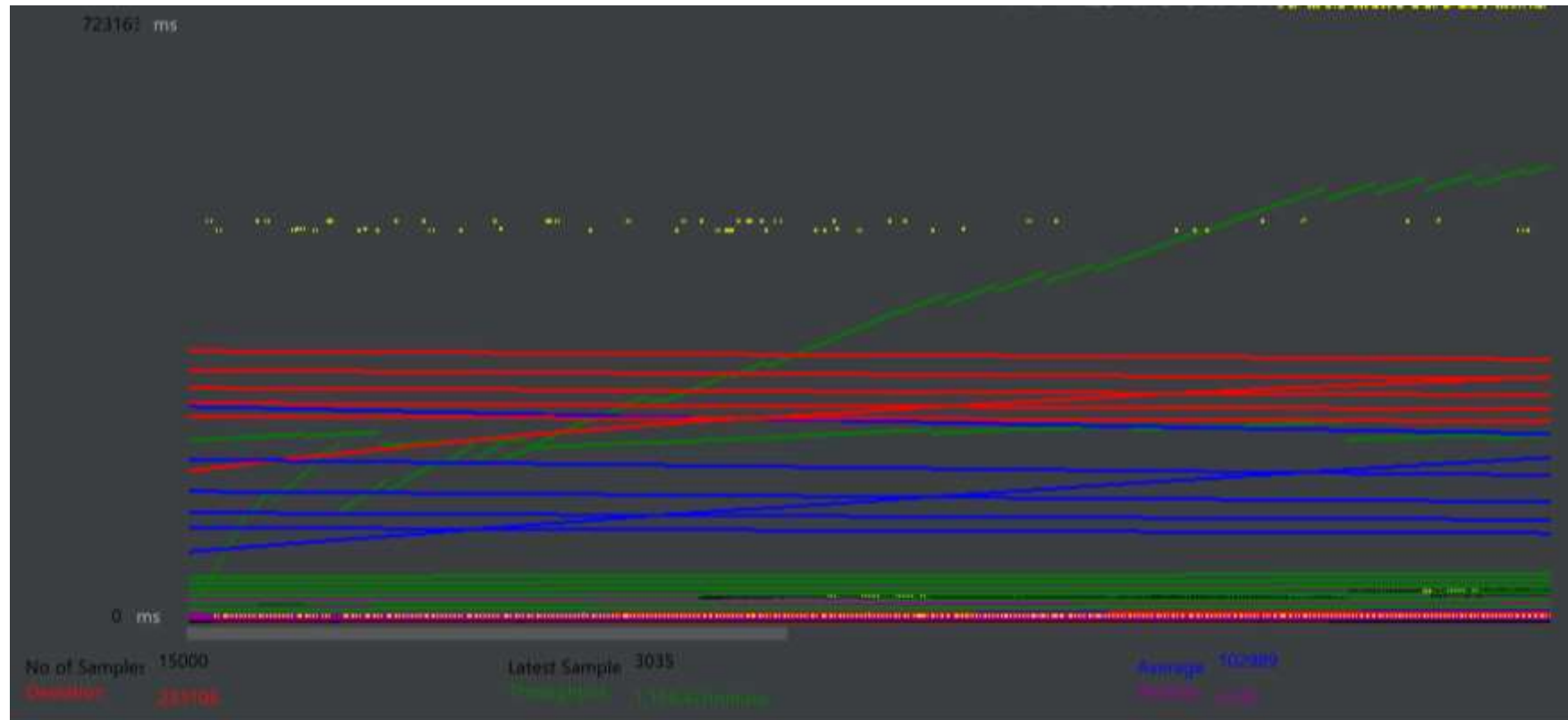
Label	Sample	Average	Error%	Throughput
Read	1000	4085	0.76	18.302.495.708.201.150
Delete	1000	4290	0.0	1.705.087.774.591.800
Update	1000	4083	0.0	10.762.654.123.008.300
Create	1000	4076	0.0	1.456.234.603.146.222
TOTAL	4000	4134	0.0	7.980.750.231.534.600

Anexo 11. Tabla de resultados prueba de estrés Spring Boot

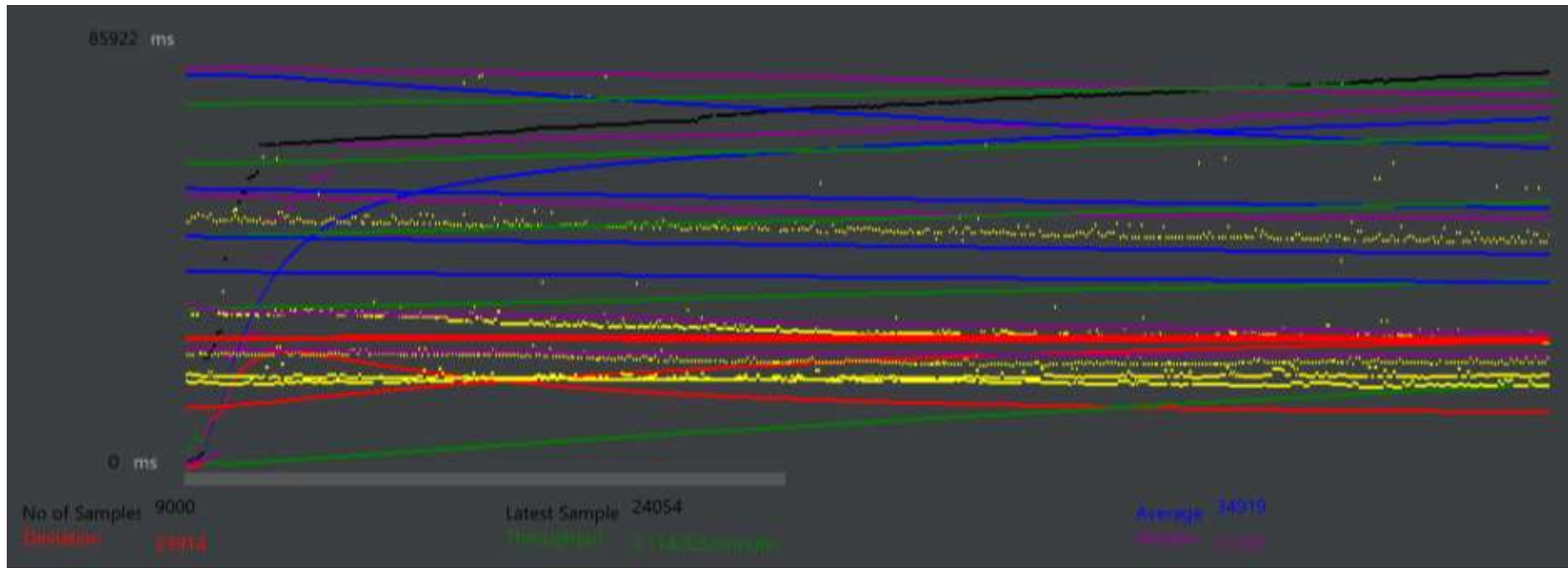
Label	Sample	Average	Error%	Throughput
Read	1000	5122	0.5643	1.155.225.257.346.220
Create	1000	4638	0.0	12.500.543.100.900.100
Update	1000	4127	0.0	1.503.308.222.433.200
Delete	1000	4096	0.0	1.225.365.332.034.654
TOTAL	4000	4496	0.0	4.057.542.321.532.500

Anexo 12. Tabla de resultados prueba de estrés Stack Mern

7.5. Lista de Figuras



Anexo 13. Grafico prueba de carga Stack Mern.



Anexo 14. Grafico prueba de carga Spring Boot

8. Referencias

- [1] M. Grahl *et al.*, “Archive WEB API: A web service for the experiment data archive of Wendelstein 7-X,” *Fusion Eng. Des.*, vol. 123, pp. 1015–1019, Nov. 2017, doi: 10.1016/j.fusengdes.2017.02.047.
- [2] Z. Niu, C. Yang, and Y. Zhang, “A design of cross-terminal web system based on JSON and REST,” in *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, Oct. 2014, pp. 904–907, doi: 10.1109/ICSESS.2014.6933711.
- [3] N. Osmic and J. Velagic, “Design of a simple service oriented supervisory control and data acquisition system,” in *Proceedings Elmar - International Symposium Electronics in Marine*, Nov. 2017, vol. 2017-September, pp. 245–248, doi: 10.23919/ELMAR.2017.8124478.
- [4] N. Yi, X. Feng, and C. Li, “2017 7 th International Conference on Communication Systems and Network Technologies The Design and Implementation of the Front End of the Art Play Library System,” doi: 10.1109/CSNT.2017.54.
- [5] S. Fang *et al.*, “An integrated system for regional environmental monitoring and management based on internet of things,” *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1596–1605, 2014, doi: 10.1109/TII.2014.2302638.
- [6] L. Moron-Rodriguez, B. Longa-Chevarria, and P. Shiguihara-Juarez, “Analysis of image transfer mechanisms in a RESTful API client-server architecture and its application to lane detection,” in *2017 IEEE International Conference on Aerospace and Signals (INCAS)*, Nov. 2017, vol. 2017-November, pp. 1–4, doi: 10.1109/INCAS.2017.8123497.
- [7] M. Garriga, C. Mateos, A. Flores, A. Cechich, and A. Zunino, “RESTful service composition at a glance: A survey,” *Journal of Network and Computer Applications*, vol. 60, Academic Press, pp. 32–53, Jan. 01, 2016, doi: 10.1016/j.jnca.2015.11.020.
- [8] S. B. A. Ben Lamine, H. Baazaoui Zghal, M. Mrissa, and C. Ghedira Guegan, “An ontology-based approach for personalized RESTful Web service discovery,” in *Procedia Computer Science*, Jan. 2017, vol. 112, pp. 2127–2136, doi: 10.1016/j.procs.2017.08.235.
- [9] S. Vinoski, “RESTful web services development checklist,” *IEEE Internet Comput.*, vol. 12, no. 6, 2008, doi: 10.1109/MIC.2008.130.
- [10] S. Martínez, V. Cosentino, and J. Cabot, “Model-based analysis of Java EE web security misconfigurations,” *Comput. Lang. Syst. Struct.*, vol. 49, pp. 36–61, Sep. 2017, doi: 10.1016/j.cl.2017.02.001.
- [11] K. Guntupally, R. Devarakonda, and K. Kehoe, “Spring Boot based REST API to Improve Data Quality Report Generation for Big Scientific Data: ARM Data Center Example,” in *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, Jan. 2019, pp. 5328–5329, doi: 10.1109/BigData.2018.8621924.
- [12] Y. Xiong, “The design of automatically generating drop-down menu on JSP,” in *Proceedings - 2012 International Conference on Computer Science and Information Processing, CSIP 2012*, 2012, pp. 1404–1406, doi: 10.1109/CSIP.2012.6309126.
- [13] L. Coppolino, S. D’Antonio, G. Mazzeo, and L. Romano, “A comparative analysis of emerging approaches for securing java software with Intel SGX,” *Futur. Gener. Comput. Syst.*, vol. 97, pp. 620–633, Aug. 2019, doi:

- 10.1016/j.future.2019.03.018.
- [14] “Sci-Hub | The Maven Dependency Graph: A Temporal Graph-Based Representation of Maven Central. 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR) | 10.1109/MSR.2019.00060.” <https://sci-hub.se/10.1109/MSR.2019.00060> (accessed Aug. 15, 2021).
- [15] Z. Fang *et al.*, “A Java web application based MDSplus data analysis and real-time monitoring system for EAST,” *Fusion Eng. Des.*, vol. 147, p. 111215, Oct. 2019, doi: 10.1016/j.fusengdes.2019.05.034.
- [16] W. Chansuwath and T. Senivongse, “A model-driven development of web applications using AngularJS framework,” Aug. 2016, doi: 10.1109/ICIS.2016.7550838.
- [17] P. Porter, S. Yang, and X. Xi, “The Design and Implementation of a RESTful IoT Service Using the MERN Stack,” in *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, Nov. 2019, pp. 140–145, doi: 10.1109/MASSW.2019.00035.
- [18] S. Colbran and M. Schulz, “An update to the software architecture of the iLab Service Broker,” in *Proceedings of 2015 12th International Conference on Remote Engineering and Virtual Instrumentation, REV 2015*, Apr. 2015, pp. 90–93, doi: 10.1109/REV.2015.7087269.
- [19] N. Kobayakawa and K. Yoshida, “How GitHub Contributing.md Contributes to Contributors,” 2017, doi: 10.1109/COMPSAC.2017.139.
- [20] X. Han, N. Zhang, W. He, K. Zhang, and L. Tang, “Automated Warship Software Testing System Based on LoadRunner Automation API,” doi: 10.1109/QRS-C.2018.00023.
- [21] “Analytical Study and Implementation of Web Performance Testing Tools. 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE) | 10.1109/ICRIEECE44171.2018.9008408.” <https://sci-hub.se/10.1109/ICRIEECE44171.2018.9008408> (accessed Aug. 20, 2021).
- [22] “Comparative Analysis for Web Applications Based on REST Services: MEAN Stack and Java EE Stack | KnE Engineering.” <https://knepublishing.com/index.php/KnE-Engineering/article/view/3647/7623> (accessed Jul. 21, 2020).
- [23] K. A. Bakar, M. H. M. Shaharill, and M. Ahmed, “Performance evaluation of a clustered memcache,” 2010, doi: 10.1109/ICT4M.2010.5971915.
- [24] R. Abbas, Z. Sultan, and S. N. Bhatti, “Comparative analysis of automated load testing tools: Apache JMeter, Microsoft Visual Studio (TFS), LoadRunner, Siege,” in *International Conference on Communication Technologies, ComTech 2017*, Oct. 2017, pp. 39–44, doi: 10.1109/COMTECH.2017.8065747.
- [25] A. Quesada and O. M. Castro, “Differences between programming languages offered by the academy versus demanded by the companies in Costa Rica: Information systems engineering from the universidad nacional,” in *2017 XLIII Latin American Computer Conference (CLEI)*, Sep. 2017, vol. 2017-January, pp. 1–7, doi: 10.1109/CLEI.2017.8226460.
- [26] A. Constituyente, “Republica del Ecuador, ‘Constitucion del Ecuador.’” <https://www.wipo.int/edocs/lexdocs/laws/es/ec/ec030es.pdf> (accessed Jul. 26, 2020).
- [27] Ecuador, “Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación.”

- [28] “IntelliJ IDEA: el IDE de Java eficaz y ergonómico de JetBrains.” <https://www.jetbrains.com/es-es/idea/> (accessed Aug. 20, 2021).
- [29] “GitHub: Where the world builds software · GitHub.” <https://github.com/> (accessed Aug. 20, 2021).
- [30] “Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel.” <https://www.mockaroo.com/> (accessed Aug. 20, 2021).
- [31] “SlashData.” <https://www.slashdata.co/reports/> (accessed Aug. 20, 2021).
- [32] “Sci-Hub | One solution for execution of JavaScript in Java EE application servers. 2018 Zooming Innovation in Consumer Technologies Conference (ZINC) | 10.1109/ZINC.2018.8448646.” <https://sci-hub.se/10.1109/ZINC.2018.8448646> (accessed Aug. 18, 2021).
- [33] J. Tihomirovs and J. Grabis, “Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics,” *Inf. Technol. Manag. Sci.*, vol. 19, no. 1, Jan. 2017, doi: 10.1515/itms-2016-0017.
- [34] H. K. Dhalla, “A Performance Analysis of Native JSON Parsers in Java, Python, MS.NET Core, JavaScript, and PHP,” 2020.