

Prototype on Social Network Storage Technologies Non-relational - Neo4j

Roger Calderón Moreno, Ing.¹, Miguel A. Avendaño Castañeda, Ing.¹, Julián D. Varela Prieto, Ing.¹

¹ Universidad de los Llanos, Colombia, rcalderonmoreno@unillanos.edu.co, miguelavendanoc@gmail.com,
juliandvarelap@gmail.com

Abstract— This article is focused on the development of a prototype of social tourism network, under a non-relational storage engine called Neo4j, including an implementation of an architecture developed specifically for this prototype, which provides a structure that allows the organization of code using triad of classes, with well-defined responsibilities, allowing the interaction between client and server layers. In the final part of this project performance and stress tests are performed, which lead to receive proper response times by non-relational engine on the prototype of social network.

Keywords— Prototype, Vertical social network, Tourism, noSQL, Neo4j, Node.

Digital Object Identifier (DOI): <http://dx.doi.org/10.18687/LACCEI2015.1.1.071>

ISBN: 13 978-0-9822896-8-6

ISSN: 2414-6668

13th LACCEI Annual International Conference: “Engineering Education Facing the Grand Challenges, What Are We Doing?”
July 29-31, 2015, Santo Domingo, Dominican Republic **ISBN:** 13 978-0-9822896-8-6 **ISSN:** 2414-6668
DOI: <http://dx.doi.org/10.18687/LACCEI2015.1.1.071>

PROTOTIPO DE RED SOCIAL SOBRE TECNOLOGIAS DE ALMACENAMIENTO NO RELACIONAL – Neo4J

Roger Calderón Moreno, Especialista en Ingeniería del Software¹, Miguel A. Avendaño Castañeda, Ingeniero de Sistemas², Julián D. Varela Prieto, Ingeniero de Sistemas³

¹ Escuela de Ingeniería, Universidad de los Llanos - Colombia
rcalderonmoreno@unillanos.edu.co

² Programa de Ingeniería de Sistemas, Universidad de los Llanos - Colombia
miguelavendanoc@gmail.com

³ Programa de Ingeniería de Sistemas, Universidad de los Llanos - Colombia
juliandvarelap@gmail.com

Resumen — El presente artículo está enfocado en el desarrollo de un prototipo de red social de turismo, bajo el motor de almacenamiento no relacional Neo4j, incluyendo la implementación de una arquitectura desarrollada específicamente para este prototipo, en la cual se establece una estructura que permite la organización del código bajo triadas de clases, con responsabilidades bien definidas, permitiendo la interacción entre la capa del cliente y la del servidor.

Como parte final de este proyecto se realiza una prueba de rendimiento y otra de estrés, con las cuales se logran percibir tiempos de respuesta apropiados por parte del motor no relacional sobre el prototipo de red social.

Palabras clave — Prototipo, Red social vertical, Turismo, noSQL, Neo4j, Nodo.

Abstract — This article is focused on the development of a prototype of social tourism network, under a non relational storage engine called Neo4j, including an implementation of an architecture developed specifically for this prototype, which provides a structure that allows the organization of code using triad of classes, with well defined responsibilities, allowing the interaction between client and server layers.

In the final part of this project performance and stress tests are performed, which lead to receive proper response times by non-relational engine on the prototype of social network.

Keywords — Prototype, Vertical social network, Tourism, noSQL, Neo4j, Node.

1. INTRODUCCIÓN

El “PROTOTIPO DE RED SOCIAL SOBRE TECNOLOGIAS DE ALMACENAMIENTO NO RELACIONAL – Neo4J”, nace como un trabajo dentro del grupo de investigación GITECX - GRUPO DE INVESTIGACIÓN EN TECNOLOGÍAS ABIERTAS, en donde se abordan nuevos conceptos para el almacenamiento de datos diferentes del enfoque tradicional, con el fin de adaptarlos a nuevas áreas de conocimiento, para nuestro caso una Red Social.

El concepto de Red Social, ha sido un término abordado desde hace tiempo por diversos pensadores en campos de las ciencias humanas como la sociología, la psicología o la antropología, para dar explicación a distintos comportamientos humanos con relación a su entorno social y/o productivo [1]

Con el paso del tiempo, el concepto de red social se distancia de los claustros académicos de las ciencias sociales para ingresar al mundo de la informática, gracias a la evolución de la Web a comienzos del siglo XXI, con el surgimiento de proyectos como Facebook en el 2004 [2], Twitter en el 2006 [3] o Instagram en el 2010 [4]; haciendo que el concepto de redes sociales tome una orientación virtual, abriendo puertas a un nuevo concepto con gran acogida en el diario vivir de las personas.

El despliegue que han tenido las redes sociales en los últimos 10 años es tan grande, que actualmente existen las denominadas redes sociales verticales que se adaptan a los gustos y/o necesidades de mercados específicos de usuarios [5] y por otro lado, las redes

sociales de propósito general como Facebook o Twitter, cuya principal característica es la diversidad o generalidad de gustos de sus usuarios.

El presente trabajo propone el diseño y desarrollo de un prototipo de red social vertical de turismo y cultura para la región de los Llanos Orientales de Colombia denominado “Turismeta”. Este prototipo pretende aplicar el concepto de red social vertical, diseñada y desarrollada desde cero, aplicando un modelamiento de datos orientado a grafos con el motor Neo4j, y todo un desarrollo de tipo web 2.0 con tecnologías como PHP, HTML 5 y JQUERY entre otras.

Con el desarrollo de “Turismeta” los usuarios podrán realizar dinámicas de enlaces sociales enfocadas a fortalecer los canales de promoción y divulgación de información turística de la región, contribuyendo con el desarrollo social y turístico de la misma.

2. ESTADO DEL ARTE

Las redes sociales verticales son aquellas que se enfocan en un tema específico, permitiendo que grupos concretos de personas se agrupen en un único tema, lo que en muchas ocasiones es aprovechado por las diferentes empresas, pues ven en estos sitios un canal de comunicación directo para ofrecer sus productos o servicios a un público objetivo, mientras que en las redes sociales horizontales, captar público objetivo les puede implicar un costo adicional por ser tan diverso el público que accede a dichos sitios.

En la actualidad muchos de los grandes proyectos que se mueven dentro del ámbito de la Web 2.0 como Facebook, Twitter, LinkedIn, Pinterest, Dribbble, Google, Bing, entre otros, manejan grandes volúmenes de información, lo que dio origen a nuevas tecnologías para el almacenamiento de la información, con alto grado de escalabilidad, flexibilidad y rendimiento. Dichas tecnologías se salen del esquema tradicional SQL (relacional), por tal motivo se hace referencia a ellos como los sistemas gestores de base de datos NoSQL. Las bases de datos no relacionales surgen como respuesta a nuevos paradigmas computacionales, en donde la implementación de un modelo relacional no es suficiente; a este concepto se le denominó NoSQL (No Only SQL) y se forjó en los años 90, refiriéndose

a modelos de base de datos que no usaban lenguajes de consultas SQL (Structured Query Language).

Cabe destacar que no todos los proyectos tienen la misma finalidad y es por esto que los motores de bases de datos NoSQL han ido surgiendo en la medida que los motores relacionales, por diferentes motivos, se han quedado cortos a la hora de resolver las demandas exigidas.

Actualmente los motores NoSQL se clasifican según la forma en la que se concibe su estructura interna de almacenamiento de información. Se encuentran las orientadas a documentos (MongoDB o CouchDB), las orientadas a columnas (Cassandra o HBase), las de clave valor (DynamoDB o Redis) y las orientadas a grafos (Infinity Graph o Neo4j) [7].

Estos tipos de motores de almacenamiento no relacional han tenido gran acogida entre proyectos web de la actualidad como es el caso de Google que utiliza BigTable, Amazon trabajando con Dynamo, Apache con HBase y Facebook y Twitter con Cassandra [8].

Claro está que el hecho de contar con un sistema noSQL no evita que sigan implementando motores relacionales para algunas de sus funcionalidades, como es el caso de Facebook que implementa MySQL en muchos de sus servidores para administrar las notificaciones de los usuarios, e incluso han aprovechado su licencia de código abierto para realizar las modificaciones que han visto necesarias para mejorar sus servicios [9], de igual manera Twitter ha implementado MySQL en su plataforma y ha aprovechado su licenciamiento libre para adaptarlo a sus necesidades e incluso han liberado esas modificaciones para que cualquier persona pueda utilizarlas [10] [11] [12].

Para el desarrollo de la red social de Turismo expuesta en el presente trabajo, se toma la decisión de utilizar un motor NoSQL orientado a grafos. En este tipo de base de datos la información se almacena como una estructura estura de nodos relacionados entre sí conformando lo que se denomina como grafo. Los grafos facilitan considerablemente el recorrido de la información, sobre todo a la hora de realizar consultas que impliquen complejas que impliquen relaciones directas entre nodos. En la actualidad podemos destacar algunos motores NoSQL orientados a grafos existentes como: Neo4j, InfoGrid, InfiniteGraph, DEX o Virtuoso.

Finalmente fue seleccionado Neo4j, por sus características de código abierto, desarrollado e implementado en Java, completamente transaccional y con licencia AGPL v3 (Affero General Public License, versión 3). Neo4j gracias a su estructura orientada a grafos, permite realizar recorridos sobre ellos y obtener resultados que un modelo tradicional de almacenamiento relacional le costaría mucho trabajo e incluso ser imposible de realizar. Para un ejemplo de esto, se puede plantear un caso hipotético en donde haya que realizar una consulta para obtener la información sobre los amigos de los amigos de una hija, los gustos de los compañeros de clase más cercanos a ella o incluso conocer el lugar de origen de los padres de los novios de las exnovias del novio de la misma, dado el caso como ejemplo de una investigación de tipo social. Cabe aclarar que el potencial del almacenamiento orientado a grafos no solo se enfoca en modelar estructuras complejas como la descrita anteriormente, sino en la óptima respuesta en la ejecución de consultas.

Dado el potencial del motor Neo4j, podría ser un importante respaldo tecnológico para proyectos de distintas áreas como: administración de redes y datos, geo posicionamiento y redes de contacto [13]; tal y como lo han hecho National Geographic, ebay, Infojobs, Mapillary Meetic, StarHub entre muchos otros ejemplos [14] de empresas que le han apostado a Neo4j como alternativa de almacenamiento para sus proyectos.

3. IMPLEMENTACIÓN

Para el desarrollo del prototipo de red social “Turismeta” fue utilizada la metodología de programación extrema (Extreme Programming - XP), por medio de la cual se establecen 5 fases que permiten alcanzar los objetivos del proyecto.

- 1ª Fase: Planificación del proyecto.
- 2ª Fase: Diseño.
- 3ª Fase: Codificación.
- 4ª Fase: Pruebas.
- 5ª Fase: Implementación.

Estas fases se vieron replicadas en 5 iteraciones, las cuales permitieron el desarrollo óptimo del prototipo. A continuación se listan los resultados finales de cada una de las fases de la metodología.

3.1 PLANIFICACIÓN DEL PROYECTO

En esta fase se logra establecer los actores del sistema, así como también los requerimientos funcionales y no funcionales.

Además se identificaron las tecnologías más apropiadas para el desarrollo del prototipo de red social, logrando definir a Neo4j como el motor de base de datos idóneo para el desarrollo del proyecto.

3.2 DISEÑO DEL SISTEMA

En esta etapa se logró modelar el funcionamiento del sistema mediante el lenguaje de modelado unificado (UML) [15], por medio del cual se consiguió el diagrama de caso de uso que se muestra en la Figura 1.

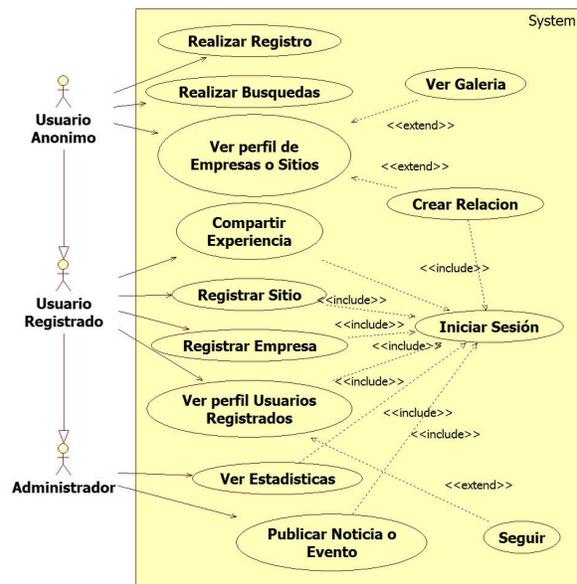


Figura 1. Diagrama de caso de uso general de Turismeta.

Adicionalmente en esta etapa se definieron los diseños a maquetar construidos por medio de la herramienta Balsamiq Mockups [16].

3.3 DESARROLLO DEL SISTEMA

Una vez realizada la planificación y diseño del prototipo de red social Turismeta, se aborda la fase de desarrollo, en donde es diseñada e implementada una arquitectura de software en la que se distribuyeron los diferentes elementos del sistema

logrando un desarrollo limpio, escalable y de fácil migración expuesta en el siguiente literal.

3.3.1 ARQUITECTURA DE SOFTWARE

La figura 2 representa la arquitectura propuesta para el desarrollo del prototipo de red social vertical Turismeta.

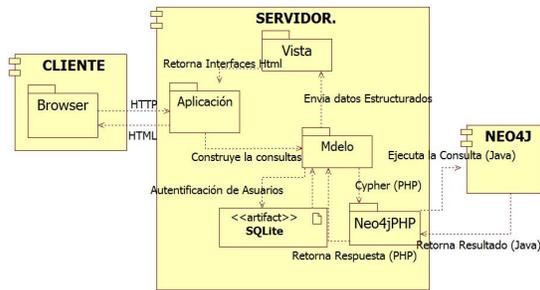


Figura 2. Arquitectura de software implementada.

A continuación se explica de forma detallada cada uno de los componentes de la arquitectura.

Cliente

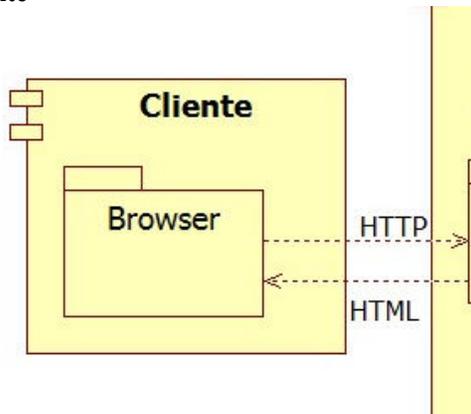


Figura 3: El cliente.

El cliente representa a los usuarios del sistema que por medio de un navegador web realizan peticiones http al sistema.

Servidor

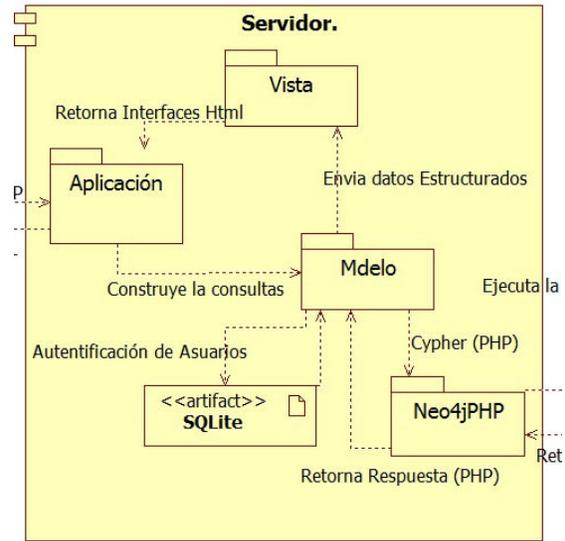


Figura 4: El servidor.

Aplicación: Representa a un conjunto de clases que reciben las peticiones del usuario y realizan una labor de ente validador de condiciones propias para cada interface que será retornada al usuario. Igualmente coordina las acciones que se deben realizar entre el elemento Modelo y el elemento Vista.

Modelo: El elemento modelo representa un conjunto de clases encargadas de construir las diferentes consultas, inserciones y actualizaciones a la base de datos.

Vista: Representa a un conjunto de clases encargadas de construir las interfaces graficas de usuario por medio de los datos que recibe del Modelo y del paquete de plantillas html5 con las que cuenta el sistema.

SQLite: Representa la librería Sqlite implementada para la autenticación de usuarios.

Neo4jPHP [17]: Es una librería desarrollada por *Josh Adell*, que permite la interacción entre el motor de bases de datos no relacional Neo4j desarrollado en Java, y el lenguaje de programación PHP 5 con el que fue escrito el prototipo.

Motor de base de datos no relacional neo4j

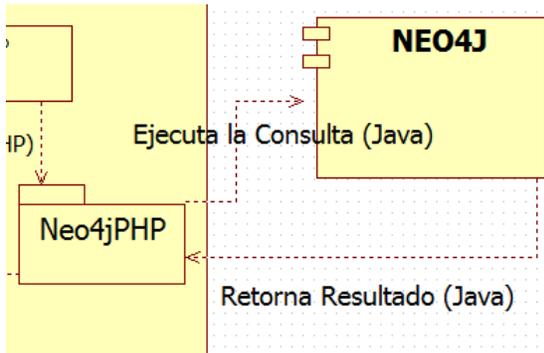


Figura 5: El Motor NoSQL.

Neo4j, representa el motor de bases de datos no relacional orientado a grafos denominado Neo4j, en el cual se centra el modelamiento de datos implementado para el desarrollo del prototipo de red social vertical Turismeta.

3.3.2 DIAGRAMA DE CLASES

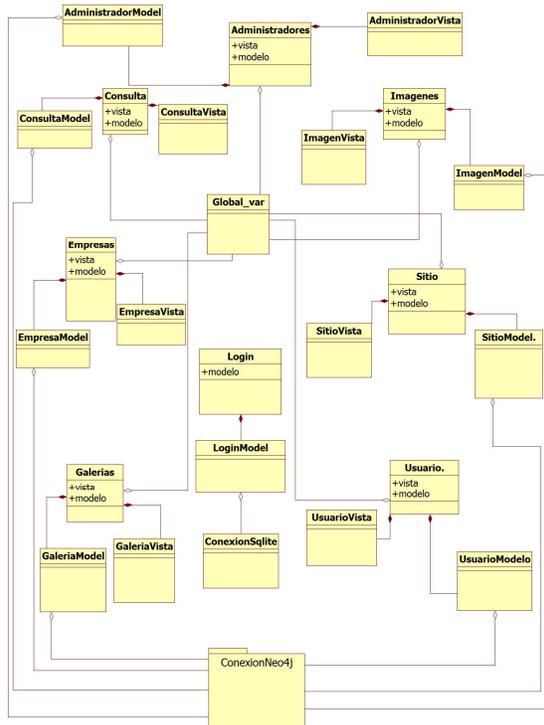


Figura 6: Diagrama de Clases General.

El desarrollo del prototipo de red social fue construido sobre una estructura de clases que permite una efectiva organización del código, denominando esta estructura como “Triadas de clases”¹ donde cada clase cuenta con funciones específicas que permiten la interacción entre las interfaces gráficas de usuario y el modelamiento de los datos.

La Figura 6 representa el diagrama de clases general de forma no detallada, en donde se pueden apreciar las triadas de clases definidas para el funcionamiento de los distintos componentes del proyecto.

Cada una de las triadas de clases que se observa en la Figura 6 está compuesta por una clase central y dos complementarias. En la Figura 7 se expone una de las triadas de clases del proyecto.

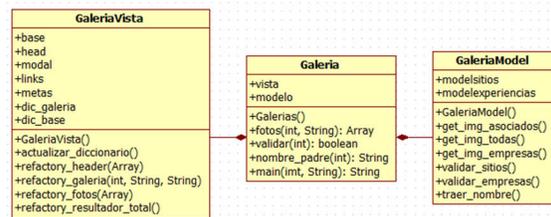


Figura 7: Triada de clases de Galeria.

En la Figura 7 se aprecia la triada de clases correspondientes al funcionamiento de la galería de imágenes, en donde la clase central denominada para este caso Galería, instancia a un objeto de sus dos clases de apoyo, de esta manera se encarga de servir de elemento ordenador para la construcción gráfica de la galería de imágenes.

En este orden de ideas, la clase GaleriaVista se encarga de generar la interfaz de la galería partiendo de las plantillas HTML5 construidas para dicho caso, mientras que la clase GaleriaModelo se encarga de gestionar la comunicación con el motor de almacenamiento de datos para obtener los datos que alimentaran las plantillas de la interfaz de la galería.

¹ Concepto definido por los autores del trabajo.

4. PRUEBAS Y RESULTADOS

4.1 PRUEBA DE RENDIMIENTO DEL MOTOR NEO4J

Con el objetivo de evaluar los tiempos de respuesta del prototipo de red social desde una perspectiva ideal de crecimiento exponencial de datos, son planteados tres escenarios descritos en la tabla 2.

Escenario	Total nodos	Total relaciones
1	123.559	315.471
2	354.846	735.207
3	583.229	1'534.811

Tabla 2: Escenarios para prueba de rendimiento.

Sobre los escenarios descritos se ejecutan tres consultas que implican un nivel de complejidad considerable, debido a que la respuesta de dichas consultas son nodos que se encuentran distanciados del nodo origen de la consulta hasta por tres enlaces seguidos, lo que implica la realización de grandes recorridos entre los nodos.

Las consultas en mención se detallan en la Tabla 3.

CONSULTA	DESCRIPCIÓN
START a=node(idSitio) MATCH s<-[:Asociada]-e-[:Img]->i<-[:Sobre]-c RETURN c;	Retorna los comentarios sobre las imágenes de una experiencia que está asociada un sitio específico.
START a=node(idUsuario) MATCH a-[:Sigue]->b-[:Sigue]->c RETURN c;	Retorna la lista de amigos de los amigos del usuario sobre el que se ejecutó la consulta.
START e=node(idNodo) MATCH e<-[:Cliente]-u1<-[:Sigue]-u2 Return u2;	Retorna la lista de usuarios que siguen a los usuarios que confían en una empresa en particular.

Tabla 3: Consultas realizadas en cada escenario de la prueba de rendimiento.

Finalmente los resultados de la prueba se pueden observar en la Tabla 4 y su respectiva gráfica en la Figura 10.

Escenario	Tiempo ejecución consulta 1 (ms)	Tiempo ejecución consulta 2 (ms)	Tiempo ejecución consulta 3 (ms)
1	3.382	4.049	4.961
2	4.339	4.32	4.926
3	4.887	7.256	7.526

Tabla 4. Tiempos de respuesta para las consultas ejecutadas.

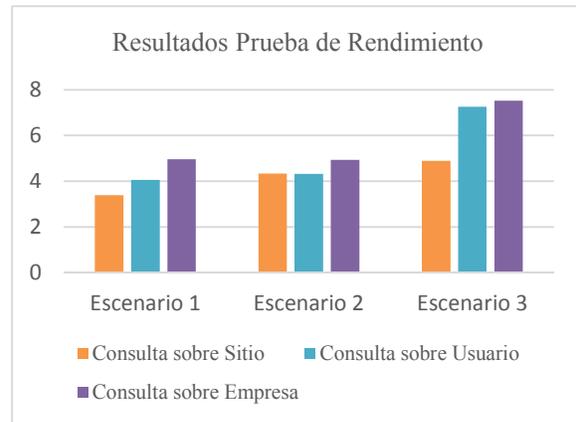


Figura 10. Gráfica de resultados prueba de rendimiento.

La grafica resultado de la Tabla 4 muestra un comportamiento estable en los tiempos de respuesta del motor teniendo en cuenta el crecimiento exponencial de la información al que fue sometido el motor de almacenamiento de datos.

Por otro lado cabe resaltar que en promedio los tiempos de respuesta no superaron los ocho (08) milisegundos, pero estos valores pueden variar dependiendo del volumen de información implícita en cada consulta, pues la misma consulta ejecutada sobre distintos nodos puede requerir más tiempo para dar respuesta, en aquellos casos en donde los nodos implicados están enlazados a un volumen de nodos mayor que otros.

4.1.1 PRUEBA DE ESTRÉS

Con el objetivo de realizar una prueba que permita analizar el comportamiento del prototipo de red social en un ambiente con alto grado de concurrencia, se diseña una prueba de estrés en la cual se simula el ingreso virtual de usuarios de forma simultánea a distintas vistas del prototipo, tal y como se describe en la Tabla 5.

ESCENARIOS	USUARIOS VIRTUALES	VISTAS A CONSULTAR
1	50	Index
2	200	Perfil de usuario
3	500	Perfil de Sitio
4	1000	Galería de Sitio Imagen.

Tabla 5. Escenario de pruebas de estrés

De esta manera para cada uno de los escenarios planeados se realizan peticiones HTTP sobre cada una de las vistas, labor que fue realizada con la herramienta de software libre Apache JMeter, que permite la realización de pruebas de este tipo.

Finalizadas las pruebas se obtienen los resultados descritos en la Figura 11.

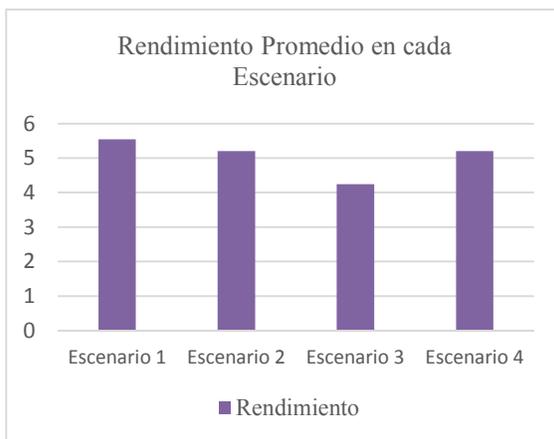


Figura 11: Resultado general de la prueba de estrés.

Analizando el rendimiento promedio del prototipo para cada uno de los escenarios establecidos, se evidencia una normalización en los tiempos de respuesta del mismo, sometido a un incremento exponencial de peticiones concurrentes, obteniendo un rendimiento promedio total de 4,955

milisegundos lo que muestra la eficiencia tanto del prototipo como del motor de bases de datos.

5. CONCLUSIONES

Como resultado del desarrollo del prototipo de red social, se destacan las siguientes conclusiones:

El planteamiento de la arquitectura de los proyectos debe estar enfocada a permitir que se pueda hacer una manipulación transparente del código, una fácil depuración de errores, la migración de tecnologías sin problemas y la escalabilidad del proyecto a un precio mínimo.

Se resalta la importancia de reconocer que cada uno de los motores de bases de datos NoSQL existentes poseen una virtud particular para la cual fueron creados, por lo cual es necesario realizar un estudio previo de las características de cada uno de ellos para así determinar cual se adapta mejor a las necesidades del proyecto a realizar.

El uso de Neo4j, al ser la base fundamental del desarrollo del prototipo, con unas características únicas que lo hacen un poderoso gestor de bases de datos, puede garantizar el éxito de proyectos tanto en procesos de investigación como en ámbitos comerciales.

El uso de motores de bases de datos no relacionales, no excluye al uso de motores relacionales para complementar funcionalidades que pueden funcionar de forma óptima sobre estos.

AGRADECIMIENTOS

En el desarrollo de este trabajo, los autores agradecemos a la Universidad de los Llanos quien a través de los últimos años ha influenciado significativamente en nuestras vidas permitiéndonos crecer como personas y profesionales; además, nos permitió integrarnos en los procesos de investigación que se están realizando en la Escuela de Ingeniería con el grupo de Investigación GITECX, y en nuestro caso, trabajamos buscando soluciones tecnológicas que faciliten el desarrollo de la sociedad a través de las tecnologías abiertas.

REFERENCIAS

- [1] C. Madariaga Orozco, R. Abello Llanos y O. Sierra García, *Redes sociales: infancia, familia y comunidad*, Colombia: Editorial Universidad del Norte, 2003.
- [2] I. Fundación Wikimedia, «Wikipedia,» 21 Julio 2014. [En línea]. Available: http://es.wikipedia.org/wiki/Historia_de_Face_book. [Último acceso: 28 Agosto 2014].
- [3] I. Fundación Wikimedia, «Wikipedia,» 25 Agosto 2014. [En línea]. Available: <http://es.wikipedia.org/wiki/Twitter>. [Último acceso: 28 Agosto 2014].
- [4] I. Fundación Wikimedia, «Wikipedia,» 26 Agosto 2014. [En línea]. Available: <http://es.wikipedia.org/wiki/Instagram>. [Último acceso: 28 Agosto 2014].
- [5] B. Tecnimedios, «Blog Tecnimedios,» 26 Marzo 2013. [En línea]. Available: http://tecnimedios.com/blog/redes_sociales/redes-sociales-verticales/. [Último acceso: 28 Agosto 2014].
- [6] A. J. Urzola. [En línea]. [Último acceso: 16 11 2012].
- [7] R. Fernández, «Genbetadev,» 27 Enero 2014. [En línea]. Available: <http://www.genbetadev.com/bases-de-datos/bases-de-datos-nosql-elige-la-opcion-que-mejor-se-adapte-a-tus-necesidades>. [Último acceso: 23 Agosto 2014].
- [8] D. Martinez, «Working in solutions,» 20 Julio 2010. [En línea]. Available: <http://blog.danielmartinez.info/?p=12..> [Último acceso: 23 Agosto 2014].
- [9] J. Sobel, «Facebook,» 21 Diciembre 2007. [En línea]. Available: <https://www.facebook.com/notes/facebook/keeping-up/7899307130>. [Último acceso: 23 Agosto 2014].
- [10] J. Villugas, «webespacio,» 10 Abril 2012. [En línea]. Available: <http://myspace.wihe.net/twitter-trabajos-codigo-mysql/>. [Último acceso: 23 Agosto 2014].
- [11] C. Aniszczyk, «GitHub,» 26 Marzo 2014. [En línea]. Available: <https://github.com/twitter/mysql/wiki>. [Último acceso: 23 Agosto 2014].
- [12] K. Crane, «GitHub,» 21 Agosto 2014. [En línea]. Available: <https://github.com/kevincrane>. [Último acceso: 23 Agosto 2014].
- [13] I. Neo Technology, «Neo4j,» 2014. [En línea]. Available: <http://neo4j.com/use-cases/geo/>. [Último acceso: 25 Agosto 2014].
- [14] I. Neo Technology, «Neo4j,» 2014. [En línea]. Available: <http://neo4j.com/customers/>. [Último acceso: 25 Agosto 2014].
- [15] J. Milutinovich, «Object Management Group,» 27 Febrero 2014. [En línea]. Available: http://www.omg.org/gettingstarted/what_is_uml.htm. [Último acceso: 23 Agosto 2014].
- [16] Balsamiq, «Balsamiq Mockups,» 2014. [En línea]. Available: <http://balsamiq.com/products/mockups/>. [Último acceso: 23 Agosto 2014].
- [17] J. Adell, «github,» 2012. [En línea]. Available: <https://github.com/jadell/neo4jphp>. [Último acceso: 24 Agosto 2014].
- [18] P. C. Ávila, «Patricia Chávez Ávila: Perfil público,» [En línea]. Available: <http://virtual2.unillanos.edu.co/moodle/user/profile.php?id=2337>. [Último acceso: 05 01 2014].
- [19] F. R. Santos, *Análisis de redes sociales*, Madrid: CIS Centro de Investigaciones Sociológicas, 2003.