



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Estudios de Posgrado
Maestría en Artes en Tecnología de la Información y
Comunicación

**PROTOTIPO DE UNA ARQUITECTURA MULTISERVIDOR BASADA EN EL
MODELO UTILIZADO EN LOS JUEGOS EN LÍNEA TIPO MMO (*MASSIVELY
MULTIPLAYER ONLINE*), PARA GARANTIZAR LA EFICIENCIA Y
ESCALABILIDAD A USUARIOS DE APLICACIONES EDUCATIVAS,
MEDIANTE EL USO DE DISPOSITIVOS MÓVILES**

Lic. Marco Antonio Ramírez Molina

Asesorado por el Ing. MA. Héctor Alberto H. Mendía Arriola

Guatemala, noviembre de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

PROTOTIPO DE UNA ARQUITECTURA MULTISERVIDOR BASADA EN EL
MODELO UTILIZADO EN LOS JUEGOS EN LÍNEA TIPO MMO (*MASSIVELY
MULTIPLAYER ONLINE*), PARA GARANTIZAR LA EFICIENCIA Y
ESCALABILIDAD A USUARIOS DE APLICACIONES EDUCATIVAS,
MEDIANTE EL USO DE DISPOSITIVOS MÓVILES

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

MARCO ANTONIO RAMÍREZ MOLINA

ASESORADO POR EL ING. MA. HÉCTOR ALBERTO H. MENDÍA ARRIOLA

AL CONFERÍRSELE EL TÍTULO DE

**MAESTRO EN ARTES EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIÓN**

GUATEMALA, NOVIEMBRE DE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

| | |
|------------|--|
| DECANO | Ing. Pedro Antonio Aguilar Polanco |
| VOCAL I | Ing. Ángel Roberto Sic García |
| VOCAL II | Ing. Pablo Christian De León Rodríguez |
| VOCAL III | Ing. José Milton De León Brán |
| VOCAL IV | Br. Jurgen Andoni Ramírez Ramírez |
| VOCAL V | Br. Oscar Humberto Galicia Núñez |
| SECRETARIA | Inga. Lesbia Magalí Herrera López |

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

| | |
|-------------|--|
| DECANO | MSc. Ing. Pedro Antonio Aguilar Polanco |
| EXAMINADOR | MSc. Ing. Murphy Olimpo Paiz Recinos |
| EXAMINADORA | M.A. Ing. Marlon Antonio Pérez Turk |
| EXAMINADORA | M.A. Ing. Héctor Alberto H. Mendía Arriola |
| SECRETARIO | M.A. Ing. Hugo Humberto Rivera Pérez |

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROTOTIPO DE UNA ARQUITECTURA MULTISERVIDOR BASADA EN EL MODELO UTILIZADO EN LOS JUEGOS EN LÍNEA TIPO MMO (*MASSIVELY MULTIPLAYER ONLINE*), PARA GARANTIZAR LA EFICIENCIA Y ESCALABILIDAD A USUARIOS DE APLICACIONES EDUCATIVAS, MEDIANTE EL USO DE DISPOSITIVOS MÓVILES

Tema que me fuera asignado por la Dirección de la Escuela de Estudios de Posgrado, con fecha noviembre de 2016.

Marco Antonio Ramírez Molina



FACULTAD DE
INGENIERÍA - USAC
ESCUELA DE
ESTUDIOS DE POSTGRADO

Escuela de Estudios de Postgrado
Facultad de Ingeniería
Teléfono 2418-9142 / 24188000 Ext. 86226

APT-2017-031

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Postgrado, al Trabajo de Graduación de la Maestría en Artes en Tecnología de la Información y la Comunicación titulado: **"PROTOTIPO DE UNA ARQUITECTURA MULTISERVIDOR BASADA EN EL MODELO UTILIZADO EN LOS JUEGOS EN LÍNEA TIPO MMO (MASSIVELY MULTIPLAYER ONLINE), PARA GARANTIZAR LA EFICIENCIA Y ESCALABILIDAD A USUARIOS DE APLICACIONES EDUCATIVAS, MEDIANTE EL USO DE DISPOSITIVOS MÓVILES"** presentado por el Licenciado en Administración de Empresas Marco Antonio Ramírez Molina, procede a la autorización para la impresión del mismo.

IMPRÍMASE.

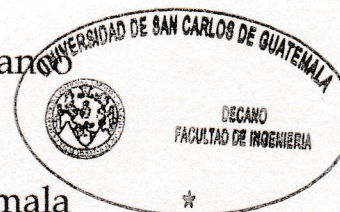
"Id y Enseñad a Todos"

MSc. Ing. Pedro Antonio Aguilar Polanco

Decano

Facultad de Ingeniería

Universidad de San Carlos de Guatemala



Cc: archivo/la

Doctorado: Sostenibilidad y Cambio Climático. **Programas de Maestrías:** Ingeniería Vial, Gestión Industrial, Estructuras, Energía y Ambiente Ingeniería Geotécnica, Ingeniería para el Desarrollo Municipal, Tecnologías de la Información y la Comunicación, Ingeniería de Mantenimiento. **Especializaciones:** Gestión del Talento Humano, Mercados Eléctricos, Investigación Científica, Educación virtual para el nivel superior, Administración y Mantenimiento Hospitalario, Neuropsicología y Neurociencia aplicada a la Industria, Enseñanza de la Matemática en el nivel superior, Estadística, Seguros y ciencias actuariales, Sistemas de información Geográfica, Sistemas de gestión de calidad, Explotación Minera, Catastro.



FACULTAD DE
INGENIERÍA - USAC
EP
ESCUELA DE
ESTUDIOS DE POSTGRADO

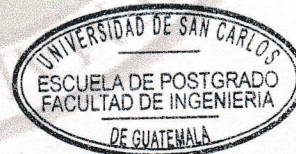
Escuela de Estudios de Postgrado
Facultad de Ingeniería
Teléfono 2418-9142 / 24188000 Ext. 86226

APT-2017-031

El Director de la Escuela de Estudios de Postgrado de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen y dar el visto bueno del revisor y la aprobación del área de Lingüística del Trabajo de Graduación titulado **"PROTOTIPO DE UNA ARQUITECTURA MULTISERVIDOR BASADA EN EL MODELO UTILIZADO EN LOS JUEGOS EN LÍNEA TIPO MMO (MASSIVELY MULTIPLAYER ONLINE), PARA GARANTIZAR LA EFICIENCIA Y ESCALABILIDAD A USUARIOS DE APLICACIONES EDUCATIVAS, MEDIANTE EL USO DE DISPOSITIVOS MÓVILES"** presentado por el Licenciado en Administración de Empresas **Marco Antonio Ramírez Molina**, correspondiente al programa de Maestría en Artes en Tecnologías de la Información y la Comunicación; apruebo y autorizo el mismo.

Atentamente,

"Id y Enseñad a Todos"



MSc. Ing. *[Signature]* Murphy Olympto Paiz Recinos
Director
Escuela de Estudios de Postgrado
Facultad de Ingeniería
Universidad de San Carlos de Guatemala

Guatemala, noviembre de 2017.

Cc: archivo/la

Doctorado: Sostenibilidad y Cambio Climático. **Programas de Maestrías:** Ingeniería Vial, Gestión Industrial, Estructuras, Energía y Ambiente Ingeniería Geotécnica, Ingeniería para el Desarrollo Municipal, Tecnologías de la Información y la Comunicación, Ingeniería de Mantenimiento. **Especializaciones:** Gestión del Talento Humano, Mercados Eléctricos, Investigación Científica, Educación virtual para el nivel superior, Administración y Mantenimiento Hospitalario, Neuropsicología y Neurociencia aplicada a la Industria, Enseñanza de la Matemática en el nivel superior, Estadística, Seguros y ciencias actuariales, Sistemas de información Geográfica, Sistemas de gestión de calidad, Explotación Minera, Catastro.



FACULTAD DE
INGENIERÍA - USAC
ESCUELA DE
ESTUDIOS DE POSTGRADO

Escuela de Estudios de Postgrado
Facultad de Ingeniería
Teléfono 2418-9142 / 24188000 Ext. 86226

APT-2017-031

Como Coordinador de la Maestría en Artes en Tecnología de la Información y la Comunicación del Trabajo de Graduación titulado **"PROTOTIPO DE UNA ARQUITECTURA MULTISERVIDOR BASADA EN EL MODELO UTILIZADO EN LOS JUEGOS EN LÍNEA TIPO MMO (MASSIVELY MULTIPLAYER ONLINE), PARA GARANTIZAR LA EFICIENCIA Y ESCALABILIDAD A USUARIOS DE APLICACIONES EDUCATIVAS, MEDIANTE EL USO DE DISPOSITIVOS MÓVILES"** presentado por el Licenciado en Administración de Empresas **Marco Antonio Ramírez Molina**, apruebo y recomiendo la autorización del mismo.

Atentamente,

"Id y Enseñad a Todos"

Marco Antonio Pérez Türk
Coordinador

MSc. Marco Antonio Pérez Türk
Coordinador de Maestría
Escuela de Estudios de Postgrado
Facultad de Ingeniería
Universidad de San Carlos de Guatemala



Guatemala, noviembre de 2017.

Cc: archivo/la

Doctorado: Sostenibilidad y Cambio Climático. **Programas de Maestrías:** Ingeniería Vial, Gestión Industrial, Estructuras, Energía y Ambiente Ingeniería Geotécnica, Ingeniería para el Desarrollo Municipal, Tecnologías de la Información y la Comunicación, Ingeniería de Mantenimiento. **Especializaciones:** Gestión del Talento Humano, Mercados Eléctricos, Investigación Científica, Educación virtual para el nivel superior, Administración y Mantenimiento Hospitalario, Neuropsicología y Neurociencia aplicada a la Industria, Enseñanza de la Matemática en el nivel superior, Estadística, Seguros y ciencias actuariales, Sistemas de información Geográfica, Sistemas de gestión de calidad, Explotación Minera, Catastro.

ACTO QUE DEDICO A:

- | | |
|------------------|---|
| Dios | Me ha bendecido en todo momento y prueba de ello es la culminación de este proyecto. |
| Mi esposa | Tu apoyo es incondicional y el ánimo que me transmites me han motivado para seguir avanzando. |
| Mis hijos | Fuente de inspiración en todo proyecto por realizar. |
| Mi madre | Quien siempre ha manifestado su confianza en mí. |

AGRADECIMIENTOS A:

Dios

Fuente de todas mis dichas, del conocimiento y sabiduría.

**Universidad de
San Carlos de
Guatemala**

Recinto, dentro del cual he adquirido los conocimientos sin limitaciones, especial agradecimiento a la Facultad de Ingeniería y la Escuela de Postgrado, así como a sus catedráticos que me transmitieron sus conocimientos sin reservas.

ÍNDICE GENERAL

| | |
|--|--------|
| ÍNDICE DE ILUSTRACIONES..... | V |
| GLOSARIO..... | XI |
| RESUMEN..... | XVII |
| PLANTEAMIENTO DEL PROBLEMA Y FORMULACIÓN DE PREGUNTAS ORIENTADORAS..... | XIX |
| OBJETIVOS..... | XXIII |
| MARCO METODOLÓGICO..... | XXV |
| INTRODUCCIÓN..... | XXXIII |
| | |
| 1. ANTECEDENTES | 1 |
| | |
| 2. JUSTIFICACIÓN | 11 |
| | |
| 3. ALCANCES DEL ESTUDIO | 13 |
| 3.1. Alcances investigativos..... | 13 |
| 3.2. Alcances técnicos..... | 14 |
| 3.3. Resultados esperados..... | 15 |
| | |
| 4. MARCO TEÓRICO..... | 17 |
| 4.1. Infraestructura tecnológica | 17 |
| 4.2. Modelos de arquitecturas tecnológicas..... | 18 |
| 4.2.1. Modelo cliente / servidor..... | 19 |
| 4.2.2. Modelo multiservidor | 20 |
| 4.3. Balanceo de carga..... | 21 |
| 4.4. Infraestructura EC2 | 22 |

| | | |
|--------|---|----|
| 4.5. | Arquitecturas para e-learning basadas en la nube | 23 |
| 4.6. | Arquitectura de entornos virtuales con agentes | 25 |
| 4.7. | Experiencia del usuario | 26 |
| 4.8. | Juegos masivos en línea | 27 |
| 4.9. | Sistemas de gestión de aprendizaje..... | 28 |
| 5. | PRESENTACIÓN DE RESULTADOS | 31 |
| 5.1. | Descripción de la arquitectura MMO (Massively Multiplayer Online) y los componentes que la integran | 31 |
| 5.2. | Descripción de la infraestructura necesaria para soportar la arquitectura MMO | 32 |
| 5.3. | Servidor web | 35 |
| 5.4. | Servidor de base de datos..... | 38 |
| 5.5. | Estructura de red..... | 40 |
| 5.6. | Descripción del experimento | 42 |
| 5.6.1. | Escenario con arquitectura cliente / servidor..... | 43 |
| 5.6.2. | Escenario con arquitectura multiservidor | 44 |
| 5.6.3. | Determinación de las capacidades | 44 |
| 5.6.4. | Funcionalidades de la aplicación | 45 |
| 5.6.5. | Limitaciones en la experimentación con cliente / servidor..... | 50 |
| 5.6.6. | Limitaciones en la experimentación con cliente / multiservidor..... | 51 |
| 5.7. | Implementación de la infraestructura en EC2..... | 52 |
| 5.7.1. | Implementación | 52 |
| 5.7.2. | Balanceo de carga | 53 |
| 5.7.3. | Webservers | 54 |
| 5.7.4. | Base de datos | 55 |
| 5.7.5. | Estructura de costos..... | 55 |

| | |
|---|-----|
| 6. DISCUSIÓN DE LOS RESULTADOS | 59 |
| 6.1. Tiempo de respuesta | 60 |
| 6.1.1. Evaluación de la arquitectura cliente / servidor | 60 |
| 6.1.2. Evaluación del rendimiento de la arquitectura cliente / multiservidor | 70 |
| 6.2. Rendimiento de las arquitecturas | 82 |
| 6.2.1. Evaluación del rendimiento de la arquitectura cliente / servidor | 82 |
| 6.2.2. Evaluación del rendimiento de la arquitectura cliente / multiservidor | 88 |
| 6.3. Proyección de usuarios concurrentes en la implementación | 96 |
| 6.4. Costos incurridos | 98 |
| 6.5. Disminución de costos de implementación | 98 |
| 6.5.1. Implementación local | 99 |
| 6.5.2. Implementación en nube | 100 |
| | |
| CONCLUSIONES | 103 |
| | |
| RECOMENDACIONES | 107 |
| | |
| REFERENCIAS BIBLIOGRÁFICAS | 109 |

ÍNDICE DE ILUSTRACIONES

FIGURAS

| | |
|--|----|
| 1. Arquitectura del modelo de objetos acoplados para interfaces multiusuario | 3 |
| 2. Arquitectura de Cloud Computing..... | 24 |
| 3. Diagrama de la arquitectura cliente – multiservidor basada en MMO..... | 32 |
| 4. Panel de configuración del clúster denominado vstudent..... | 34 |
| 5. Configuración de balanceo de carga en los nodos del clúster..... | 35 |
| 6. Panel de administración de Internet Information Server | 36 |
| 7. Panel de parámetros de php..... | 37 |
| 8. Distribución del software en los nodos miembros del clúster | 38 |
| 9. Panel del control del gestor de la base de datos | 39 |
| 10. Distribución de software en servidor de base de datos | 40 |
| 11. Estructura de red | 42 |
| 12. Interfase web de la aplicación de apoyo en la evaluación del prototipo .. | 46 |
| 13. Actividad de identificación de números del 0 al 9 en el nivel de primaria. | 48 |
| 14. Actividad de selección de números pares/impares en el nivel segundo primaria | 49 |
| 15. Actividad de lectura y descomposición de cifras en el nivel de tercero primaria | 50 |
| 16. Panel de configuración del clúster | 53 |
| 17. Interfase de acceso a la aplicación..... | 54 |
| 18. Vista del panel principal de la aplicación | 55 |

| | |
|---|----|
| 19. Milisegundos para carga de 50 usuarios concurrentes en cliente / servidor | 60 |
| 20. Despliegue de 50 usuarios concurrentes..... | 62 |
| 21. Milisegundos para carga de 100 usuarios concurrentes en cliente/servidor | 63 |
| 22. Despliegue de 100 usuarios concurrentes..... | 64 |
| 23. Milisegundos para carga de 150 usuarios concurrentes en cliente / servidor | 65 |
| 24. Despliegue de 150 usuarios concurrentes..... | 66 |
| 25. Milisegundos para carga de 250 usuarios concurrentes en cliente / servidor | 67 |
| 26. Despliegue de 250 usuarios concurrentes..... | 68 |
| 27. Milisegundos para carga de 50 usuarios concurrentes en cliente / multiservidor | 71 |
| 28. Milisegundos para 50 usuarios distribuidos por balanceo de carga | 72 |
| 29. Despliegue de 50 usuarios concurrente en arquitectura multiservidor | 72 |
| 30. Milisegundos para carga de 100 usuarios concurrentes en cliente / multiservidor | 73 |
| 31. Milisegundos para 100 usuarios distribuidos por balanceo de carga | 74 |
| 32. Despliegue de 100 usuarios concurrente en arquitectura multiservidor .. | 75 |
| 33. Milisegundos para carga de 150 usuarios concurrentes en cliente / multiservidor | 76 |
| 34. Milisegundos para 150 usuarios distribuidos por balanceo de carga | 77 |
| 35. Despliegue de 150 usuarios concurrente en arquitectura multiservidor .. | 77 |
| 36. Milisegundos para carga de 250 usuarios concurrentes en cliente / multiservidor | 78 |
| 37. Milisegundos para 250 usuarios distribuidos por balanceo de carga | 79 |
| 38. Despliegue de 250 usuarios concurrente en arquitectura multiservidor .. | 80 |
| 39. Asignación de memoria RAM por cantidad de usuarios | 85 |

| | |
|---|----|
| 40. Distribución de los datos en el escenario de 100 usuarios | 86 |
| 41. Distribución de los datos en el escenario de 250 usuarios | 87 |
| 42. Comparación del tiempo de respuesta entre las arquitecturas evaluadas | 91 |
| 43. Comparación del consumo de memoria RAM en las arquitecturas evaluadas | 92 |
| 44. Comportamiento de los procesadores en las arquitecturas evaluadas... | 93 |
| 45. Comportamiento en los adaptadores de red en las arquitecturas evaluadas | 94 |
| 46. Distribución de los datos de 250 usuario en arquitectura cliente / Multiservidor | 95 |
| 47. Distribución de 250 usuarios en nodos de la arquitectura | 96 |

TABLAS

| | |
|--|-------|
| I. Variables, subvariables e indicadores considerados en la investigación. | XXVII |
| II. Estructura de red | 41 |
| III. Actividades dentro de la aplicación..... | 47 |
| IV. Costos para implementación de servidor local | 56 |
| V. Costos para implementación de servidor en nube | 57 |
| VI. Costo por usuario de acuerdo a la arquitectura..... | 58 |
| VII. Recursos evaluados del servidor con 50 usuarios | 61 |
| VIII. Recursos evaluados del servidor con 100 usuarios | 63 |
| IX. Recursos evaluados del servidor con 150 usuarios | 65 |
| X. Recursos evaluados del servidor con 250 usuarios | 67 |
| XI. Resumen de la actividad por escenarios en la arquitectura | 69 |
| XII. Recursos evaluados de los servidores con 50 usuarios..... | 71 |
| XIII. Recursos evaluados de los servidores con 100 usuarios..... | 74 |
| XIV. Recursos evaluados de los servidores con 150 usuarios | 76 |
| XV. Recursos evaluados de los servidores con 250 usuarios | 79 |
| XVI. Resumen de la actividad por escenarios en la arquitectura cliente / multiservidor | 82 |
| XVII. Cantidad de usuarios concurrentes para establecer rendimiento de la arquitectura cliente / servidor | 83 |
| XVIII. Cantidad de partidas identificadas en la actividad del curso modelo . | 84 |
| XIX. Rendimiento de la arquitectura cliente / servidor | 85 |
| XX. Cantidad de usuarios concurrentes para establecer rendimiento de la arquitectura cliente / multiservidor..... | 89 |

XXI. Requerimientos para servidores de propuesta de implementación96

XXII. Proyección de cobertura de usuarios.....97

XXIII. Proyección de ahorro en la implementación de servidor en la nube....99

GLOSARIO

| | |
|----------------------|---|
| Anaya | Grupo español dedicado a dar apoyo en educación y que ha incursionado en la creación de cursos abiertos de educación vía <i>web</i> |
| Back End | Capa de procesamiento de los datos que son presentados en la capa <i>Front End</i> . |
| b-Learning | <i>Blended Learning</i> ; se refiere a la combinación de actividades presenciales de actividades de educación y el uso medios digitales e <i>Internet</i> . |
| EC2 | <i>Elastic Cloud Computing</i> |
| e-Learning | Consiste en la modalidad de educación a través de Internet |
| Fast Ethernet | Es un estándar de comunicaciones de redes con velocidad de 100 <i>megabits</i> por segundo. |
| Front End | Capa de presentación de un <i>software</i> y que interactúa con los usuarios. |

| | |
|--------------------------------|--|
| <i>Gigabit Ethernet</i> | Es un estándar de comunicaciones de redes con velocidad de 1 <i>gigabit</i> por segundo. |
| <i>HTML</i> | <i>Hyper text markup language</i> ; Lenguaje de marcas de hipertexto. |
| <i>IP</i> | Dirección única con la que se puede identificar a un dispositivo perteneciente a una red. |
| <i>Latencia</i> | Suma de retardos temporales dentro de una red, en el momento de generar una comunicación |
| <i>LMS</i> | <i>Learning management system</i> ; Sistema de gestión de aprendizaje. |
| <i>Middleware</i> | Es un <i>software</i> que tiene la función de comunicarse con una o varias aplicaciones o sistemas operativos con el objetivo de facilitar la labor de comunicar y sincronizar dispositivos dentro de una red. |
| <i>m-Learning.</i> | Metodología de enseñanza y aprendizaje que se basa en el uso de dispositivos móviles y con acceso a plataformas educativas en Internet. |
| <i>MMO</i> | <i>Massively multiplayer online</i> ; Multijugador masivo en línea. |

| | |
|----------------|---|
| MMOFPS | <i>Massively multiplayer online first person shooter;</i> Multijugador masivo en línea en primera persona. |
| MMOG | <i>Massively multiplayer online games;</i> juegos multijugador-masivos en línea. |
| MMORPG | <i>Massively multiplayer online role-playing game;</i> videojuego de rol multijugador masivo en línea. |
| MMORTS | <i>Massively multiplayer online real-time strategy;</i> Multijugador Masivo de Estrategia en Tiempo Real. |
| MongoDb | Sistema de base de datos NoSQL que se orienta a la gestión de documentos y es de código abierto. |
| MOOC | <i>Massively online open course;</i> Curso masivo en línea. |
| MPP | <i>Massively parallel procesing;</i> Procesamiento masivo paralelo. |
| ms | Milisegundos |
| MsSQL | Sistema de gestión de bases de datos relacional de código abierto. |

| | |
|--------------------------|--|
| <i>Multicast</i> | Multidifusión; se refiere al envío de información, de manera simultánea, a diferentes redes. |
| <i>NAS</i> | <i>Network attached dtorage</i> ; Almacenamiento vonectado en red. |
| <i>NLB</i> | <i>Network load balanced</i> ; Balance de carga de Red. |
| <i>PHP</i> | Lenguaje de programación en ambiente <i>web</i> con énfasis a sistemas operativos de servidores. |
| <i>Proxy</i> | Servidor Proxy; Gestiona las peticiones de recursos dentro de una red. |
| <i>RAM</i> | <i>Random access memory</i> ; Memoria de acceso aleatorio, es la memoria que se utiliza para ejecutar instrucciones en un sistema operativo. |
| <i>r-Learning</i> | Se refiere a la educación que se apoya en el uso de artefactos robóticos que es altamente interactivo y personalizado. |
| <i>s</i> | Segundos |
| <i>SAN</i> | <i>Storage area network</i> ; Red de área de almacenamiento. |

WPI

Web platform installer; Instalador de plataforma
Web.

RESUMEN

El uso de la tecnología como medio de aprendizaje se incrementa en la actualidad de manera rápida y variada a tal punto que el uso de dispositivos móviles, como *tablets* y teléfonos, es una tendencia marcada en los años recientes lo cual puede representar una nueva modalidad en el proceso de enseñanza y aprendizaje. Las tecnologías de información han evolucionado y dan respuesta a necesidades sociales por medio *software* y *hardware* especializado que se refleja en diversas actividades económicas y especialmente en la educación.

Como objetivo central, en este trabajo se desarrolló un prototipo de arquitectura que tuvo como modelo la utilizada en los juegos masivos multijugador en línea MMO y tomó en cuenta parte de sus componentes para lograr que funcionase sobre una plataforma educativa que otorgara servicios a cientos de usuarios concurrentes y pretendiendo que el trabajo realizado demostrase que es posible garantizar eficiencia, escalabilidad y disponibilidad del modelo propuesto.

Lo anterior pretende resolver el problema que se puede presentar en la utilización del modelo cliente / servidor debido a limitaciones que han sido identificadas en investigaciones anteriores, entre las que destaca el nulo crecimiento horizontal.

Para determinar la conveniencia de este prototipo se diseñó una investigación de tipo descriptiva, que evaluó la experiencia del usuario validando las siguientes variables: cantidad en milisegundos para la carga de la aplicación web implementada, cantidad de usuarios concurrentes que soporta la arquitectura y partidas efectuadas, que tal como se puede apreciar son aspectos cuantitativos.

Como resultado de lo anterior se determinó que la arquitectura evaluada responde de manera adecuada a las peticiones de entre 50 y 150 usuarios concurrentes, consumiendo servicios de procesador y memoria RAM a niveles de 24% y 30% en un nodo y 20% y 66% en el segundo, respectivamente.

Se concluye que la arquitectura cubre los siguientes aspectos: eficiencia, disponibilidad y escalabilidad.

PLANTEAMIENTO DEL PROBLEMA Y FORMULACIÓN DE PREGUNTAS ORIENTADORAS

El creciente uso de los dispositivos móviles ha provocado un cambio en el manejo de las tecnologías de la información y comunicaciones y particularmente en el ámbito de la educación vienen a representar una tendencia en la que la movilidad se vuelve preponderante.

Se tomó como referencia la publicación de Aguilar y De La Vega, (2014) “Las propiedades técnicas deseables en las plataformas educativas y herramientas de autor como productoras de contenido estandarizado” actualmente 6.7 millones de estudiantes han tomado uno o más cursos en línea, 33% de las escuelas públicas se han integrado a la educación a distancia y el 74% de los estudiantes adolescentes, en países como Estados Unidos, Japón o Finlandia, tienen acceso a servicios educativos desde dispositivos móviles como tabletas o teléfonos. Alier, (2014) indica que en 2013 se alcanzó a vender 195.4 millones de tabletas en todo el mundo.

No obstante, lo anterior, se identifican algunos problemas relacionados con la tecnología que soporta, tanto las aplicaciones como las plataformas en las que funcionan, entre los cuales se tienen la baja reutilización y adaptación de los contenidos, o bien la pérdida de los mismos por cambios en los factores como puede ser la arquitectura seleccionada, (Fernández, 2006) sin dejar de

mencionar que la actualización de los contenidos, entre servidor y dispositivos, suele ser asíncrona.

Por lo general, las aplicaciones educativas fueron desarrolladas sobre arquitecturas cliente servidor y ello implica que se dependa de un equipo centralizador, provocando una alta dependencia y tiempos de respuesta altos en función de la cantidad de usuarios que se autentican.

Para Vidal Rojas, (sf) el paradigma cliente servidor, presenta problemas de escalabilidad y eficiencia porque a medida que aumentan los usuarios éstas se degradan, debido a que todas las comunicaciones tienen que transitar por el servidor central ocasionando así un cuello de botella.

Por lo anterior, el presente trabajo de graduación propone la siguiente pregunta principal:

¿Cómo se puede adaptar una arquitectura multiservidor utilizada en los juegos en línea de tipo MMO (*Massively Multiplayer Online*), para garantizar eficiencia y escalabilidad en el uso de una aplicación educativa?

Complementariamente se plantean las siguientes preguntas auxiliares:

1. ¿Cómo se puede implementar una arquitectura multiservidor que garantice eficiencia y escalabilidad para los usuarios de aplicaciones educativas?

2. ¿Qué aplicación educativa puede ser la adecuada para llevar a cabo pruebas de eficiencia y escalabilidad en una arquitectura multiservidor?
3. ¿Cómo se puede incrementar la eficiencia de la experiencia del usuario en términos de tiempo de respuesta y de formato?
4. ¿Qué aspectos se deben tomar en cuenta para saber si una arquitectura multiservidor otorga eficiencia y escalabilidad?

OBJETIVOS

General

Desarrollar un prototipo de arquitectura multiservidor que tenga como base el modelo utilizado en los juegos en línea de tipo MMO (*Massively Multiplayer Online*) para reducir el tiempo de respuesta en el cliente y que soporte el aumento gradual de la carga transaccional que generan los usuarios de manera colaborativa, a través del uso de una aplicación educativa.

Específicos

1. Implementar una infraestructura en EC2 que soporte la arquitectura MMO, definiendo el esquema mínimo necesario y disminuyendo los costos.
2. Desarrollar un prototipo de aplicación educativa con enfoque en matemáticas, para primaria inicial, que sirva de apoyo para determinar las capacidades de la arquitectura.
3. Utilizar el procesamiento horizontal de la arquitectura MMO, para garantizar el balanceo de carga de las solicitudes, así lograr reducir el

delay de la respuesta en el cliente, así como la carga transaccional que se produce por el acceso de usuarios y su actividad colaborativa.

4. Determinar la capacidad que tiene una arquitectura multiservidor, del tipo MMO, en cuanto a su rendimiento, haciendo uso de las herramientas adecuadas.

MARCO METODOLÓGICO

Tipo de investigación

Se define como una investigación descriptiva, apoyada en un prototipo funcional, con el objetivo de detallar el comportamiento de las variables que fueron seleccionadas las que se detallan así:

sola variable, experiencia del usuario, que es posible dividirla en aspectos cuantitativos de la siguiente manera:

- Rendimiento de la arquitectura, lo cual es medible a través de los indicadores cantidad de usuarios concurrentes, y número de partidas efectuadas.
- Tiempo de respuesta, cantidad de milisegundos en el tiempo de carga de la aplicación.

Diseño de investigación

La presente investigación se define como descriptiva con el objetivo de desarrollar un prototipo de arquitectura multiservidor que tenga como base el modelo usado en los juegos de tipo MMO para incrementar el nivel de

eficiencia y escalabilidad, en la que se pueda visualizar diferencias en los comportamientos de arquitecturas de tipo cliente – servidor y cliente – multiservidor.

Otra de las características del estudio realizado fue el describir los elementos que intervienen en la investigación global.

Alcance de la investigación

Para el logro de los objetivos planteados, el prototipo desarrollado tomó en cuenta lo siguiente:

Uso de la tecnología EC2, elementos del modelo de arquitectura cliente – servidor, balanceador de carga, plataforma educativa de la modalidad LMS, así como herramientas que permitieron la detección de las capacidades de rendimiento de la propuesta en cuestión.

Dentro de la experimentación que se llevó a cabo, en la presente investigación, se definieron las variables que se detallan en el cuadro siguiente:

Tabla I. **Variables, subvariables e indicadores considerados en la investigación**

| Variable | Definición | Sub variables | Indicadores | Dimensiones |
|-------------------------|---|--------------------------------|---|--------------------|
| Experiencia del usuario | Factores que inciden en la percepción de los usuarios, que puede ser positiva o negativa, relacionada a un producto o servicio tecnológico. | Tiempo de respuesta | Cantidad de milisegundos en la carga de la aplicación | Cuantitativa |
| | | Rendimiento de la arquitectura | Cantidad de usuarios concurrentes | Cuantitativa |
| | | | Número de partidas efectuadas | Cuantitativa |

Fuente: elaboración propia.

Fases del estudio

En lo siguiente se detallan las fases que fueron llevadas a cabo en la presente investigación, tomando en cuenta desde el aspecto documental, así como el desarrollo de la propuesta y la experimentación

- **Fase I** Investigación documental: En esta fase se hizo acopio de información documental que permitiera tener una visión general sobre arquitecturas, y en particular se buscó identificar el funcionamiento de las mismas, por medio de la comprensión de los actuales modelos, así

como el uso del procesamiento horizontal en arquitecturas multiservidor del tipo MMO. Se hizo uso de fuentes secundarias, habiendo consultado tesis, libros, artículos científicos y revistas con énfasis en tecnología.

- **Fase II** Evaluación documental: Teniendo como base la fase I se procedió a clasificar los documentos que por su contenido aportaban mejores datos a la investigación, realizando una discriminación por temas, en especial por aquellos orientados a arquitecturas cliente – servidor.

- **Fase III** Desarrollo de la arquitectura prototipo: Teniendo la información teórica seleccionada se procedió a diseñar el prototipo de arquitectura, obteniendo
 - Diagrama de los componentes
 - El diseño del prototipo de la arquitectura basada en la de los juegos tipo MMO.

- **Fase IV** Desarrollo de la aplicación prototipo:
 - Determinación de los requerimientos, funcionales y no funcionales.
 - Elección de la plataforma educativa LMS
 - Integración de aplicaciones para uso combinado con plataforma educativa.

- **Fase V** Implementación: para esta fase se hizo uso de todos los elementos que integran los prototipos de las fases anteriores, con lo cual fue posible llevar a cabo lo siguiente:

- Configuración del ambiente cliente / servidor como primer escenario de pruebas.
 - Configuración del ambiente cliente / multiservidor como segundo escenario de pruebas.
 - Instalación de plataforma educativa y aplicaciones específicas para determinación de las capacidades de ambas arquitecturas.
- **Fase VI Pruebas:**
 - En esta fase se ejecutaron las pruebas necesarias para la obtención de resultados, la actividad consistió en utilizar el primer escenario, cliente / servidor, e instalar y configurar la plataforma educativa seleccionada, así como las aplicaciones específicas.
 - De la misma forma se procedió con el escenario de cliente / multiservidor que reflejó el modelo MMO.
 - Se utilizaron herramientas de simulación para realizar pruebas con escenarios de 50, 100, 150 y 250 usuarios para el caso del modelo cliente servidor, en un tiempo constante de diez minutos de corrida, con 25 segundos de espera entre cada click efectuado.
 - Seguidamente se validaron los escenarios, de usuarios anteriores, y gradualmente se fue incrementando hasta encontrar la cantidad límite para lo cual tuvo capacidad la arquitectura propuesta, basada en el modelo utilizado en los juegos de tipo MMO.
 - **Fase VII Recolección y análisis de datos:**
 - Para la obtención de datos se utilizaron las siguientes herramientas:

- *Webserver stress tool*, una herramienta *freeware* con la cual fue posible obtener datos sobre la cantidad máxima de usuarios que soportaron las plataformas en cuestión, cantidad de milisegundos de carga por tiempo de respuesta. Disponible en <https://www.es.paessler.com/tools/webstress>
- Monitor de rendimiento, herramienta nativa del sistema *Windows server*, con ella fue posible determinar tiempo de respuesta de los diferentes servidores, uso de memoria ram, comportamiento de los discos duros y procesadores. Se ejecutó en consola con el comando `perfmon.exe` y posteriormente se realizaron las configuraciones necesarias.

Técnicas de recolección de la información

El uso de distintos instrumentos fue necesario, entre los que se identifican aquellos de naturaleza primaria, secundaria y el auxilio de herramientas tecnológicas que permitieron la obtención de datos.

- Fuentes primarias:
 - Recolección de datos a través del uso de herramientas tecnológicas que permitieron identificar el rendimiento del prototipo implementado, entre ellas: *Webserver stress tool*, Monitor de rendimiento de Windows.
 - Equipo de computación, consistente en: computadoras portátiles y servidores.

- Fuentes secundarias:
 - Artículos y/o ensayos científicos.
 - Revistas con énfasis en tecnologías de información y comunicaciones
 - Tesis de postgrado de universidades.
 - Tutoriales y videos que aportaron al desarrollo de prototipo
 - Tutoriales y videos que aportaron al uso de las herramientas tecnológicas de medición de rendimiento de servidores.

INTRODUCCIÓN

La educación del siglo XXI dista mucho de ser la misma con la que muchos estuvieron relacionados algunos años atrás, varios de sus elementos fueron complementados y probablemente en otros casos sustituidos, de tal cuenta libros, cuadernos y agendas se ven fuertemente amenazados de ser desplazados por dispositivos electrónicos que al día de hoy vienen a ocupar un amplio espacio en la actividad de enseñanza y aprendizaje.

Las tecnologías de la información y las comunicaciones (TICS) han evolucionado de una manera vertiginosa y de la misma manera han cambiado la forma en que el ser humano accede a la información, estudia, aprende, conoce y enseña y en este proceso las TIC's han hecho aportes importantes, mediante herramientas que facilitan el propósito de aprender.

En tal sentido, este documento desarrolla varios capítulos en los que se podrá encontrar información del desarrollo de un prototipo de arquitectura que ha tomado el modelo de los juegos en línea de tipo Massive Multiplayer Online (MMO) y pone a la vista algunos de los antecedentes evaluados en el capítulo primero.

Dentro del capítulo dos se podrá leer la justificación que ha motivado este estudio, dejando algunos elementos con los que se espera aclarar las

razones que se tienen para llevar a cabo la investigación, seguido de los alcances que se pretenden cubrir, mismos que se detallan en el capítulo tres.

El marco teórico contiene varios de los conceptos, que se consideran necesarios tomar en cuenta para la solución que da origen a la propuesta del prototipo de arquitectura planteada, lo cual se podrá encontrar en el capítulo cuatro.

En el capítulo cinco, podrá verse cómo la parte de mayor relevancia en esta investigación, en él se detalla la presentación de los resultados obtenidos, y se conforma con las secciones: desarrollo de la propuesta e implementación de la misma para luego apreciar una discusión de los mismos; en el capítulo final se podrá encontrar los aportes de la arquitectura propuesta, análisis de la misma, sus componentes, así como el comparativo que se pudo llevar a cabo.

1. ANTECEDENTES

“El futuro de la educación estará fuertemente condicionado por la tecnología, y se caracterizará por no tener límites geográficos o temporales” indicaba R. Buckminster Fuller en 1962 y es citado por Papillon Cesteros (2010) en su trabajo *Las plataformas e-learning para la enseñanza y el aprendizaje universitario en Internet*.

La educación es sin duda uno de los valores mayores estimados que puede tener la humanidad y todo esfuerzo por hacerla llegar a quien la necesite debe ser considerado primordial, en tal sentido, los recursos tecnológicos que se disponen en la actualidad juegan un papel determinante para este propósito.

A partir de la publicación titulada: “Nuevo modelo de objetos acoplados para interfaces multiusuarios” de Juan Carlos Vidal Rojas (s.f.) se identifica el problema de escalabilidad y eficiencia por el acceso a un único servidor, así mismo, se puede apreciar la evolución y los intentos por mejorar las plataformas tecnológicas que han surgido con el paso de los años, así como las técnicas que han sido implementadas.

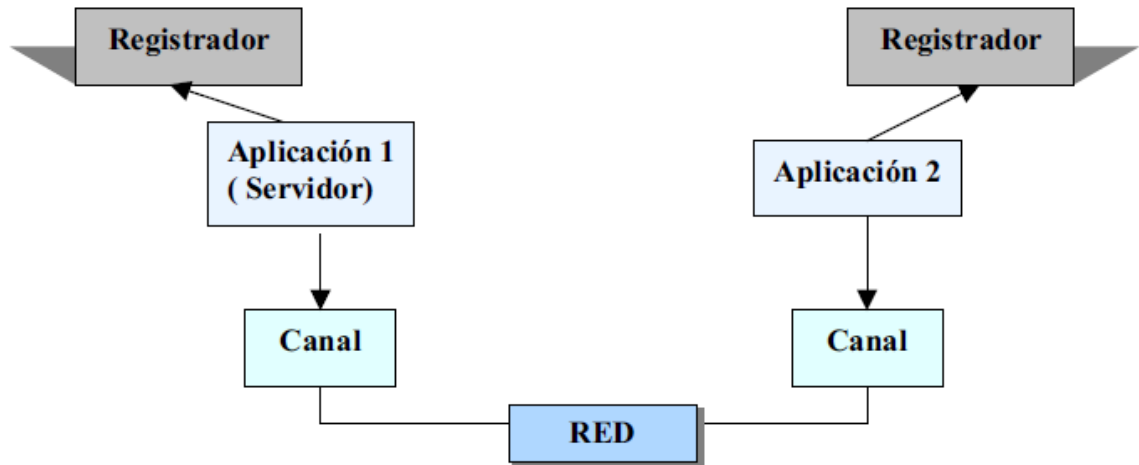
En dicho documento, se evalúan las capacidades de arquitecturas multiusuario y se resalta sus ventajas como la replicación de la funcionalidad de aplicaciones contra la centralización de un servidor y se analiza cómo un modelo basado en IP *multicast* y ordenamiento de mensajes apoyado en un secuenciador

y sincronización por token, puede liberar a un servidor y así eliminar los cuellos de botella que ocasionan los ambientes cliente – servidor (Vidal Rojas, S.F.). El autor divide su estudio en comunicación, sincronización y colaboración.

Como principal aporte, este documento presenta un modelo que, a decir de Vidal Rojas es una “Combinación de la arquitectura de colaboración totalmente replicada y una arquitectura de comunicación distribuida que utiliza como mecanismo de transmisión IP Multicast” y para ello realiza un análisis de las arquitecturas Cliente/servidor, Distribuida, Multiplexada y Replicada con lo que concluye en su propuesta final que consiste en “Objetos acoplados para interfaces multiusuario”

Sin embargo, la técnica aplicada recae en una arquitectura cliente – servidor, que si bien ha sido mejorada, no implica un crecimiento horizontal en cuanto a sus componentes, como se puede observar en la figura 1:

Figura 1. **Arquitectura del modelo de objetos acoplados para interfaces multiusuario**



Fuente: Nuevo Modelo de Objetos Acoplados para interfaces Multiusuario, Vidal Rojas. J.C. Universidad de Chile. (SF).

Por su lado, Díaz Pineda (2007), en su publicación “Análisis de la arquitectura de CDN aplicada a MMOG” aborda la problemática que representa la cada vez mayor cantidad de usuarios simultáneos que acceden a las aplicaciones *web* y que pueden ser soportados por las arquitecturas de red, la variable latencia que esto representa y la consecuente saturación de los servidores, así como el ancho de banda que es requerido para garantizar un adecuado servicio a los clientes finales, con lo cual realiza un estudio de los juegos en línea y hace ver la diversidad de tipos existentes, para resaltar la arquitectura con la cual son soportados.

De este trabajo se puede extraer las características de la arquitectura de los juegos en línea del tipo *Massively Multiplayer Online Role-Playing Game*, MMORPG, que consiste en una topología multiservidor o bien topología de servidor *proxy*, como lo destaca el ponente del tema, en el que los usuarios son distribuidos a través de los servidores para acceder al mismo contenido, incluso

a pesar de que geográficamente se ubiquen en distintas localidades. También se destaca de esta topología su interconexión multicast haciendo ver sus capacidades de reducción del tráfico mediante la entrega de un solo stream a miles de usuarios.

Una investigación colaborativa sobre LMS (del inglés *Learning Management System*) realizada por Silvina Mariel Castro, Claudio Ariel Clarenc, Carmen López de Lenz, María Eugenia Moreno y Norma Beatriz Tosco, titulada: “Analizamos 19 plataformas de *E-Learning*” (2013) define a un sistema de gestión de aprendizaje como un software que es instalado en un servidor web y hacen ver, entre otros aspectos, que estos sistemas otorgan la facilidad, de comunicarse tanto de manera síncrona o asíncrona.

Dentro de este libro es posible observar cómo estos sistemas han evolucionado y en consecuencia prestado, cada vez, mayores y mejores servicios, habiéndose iniciado con materiales elaborados con editores HTML para luego ir madurando con actividades *OnLine*.

En lo que se refiere a este trabajo de investigación, conviene tomar en cuenta lo expresado por los autores en cuanto a que “para responder a las demandas de un mercado educativo en expansión es necesario abaratar los costes de producción de cursos, reducir el tiempo requerido para su desarrollo, facilitar su gestión y simplificar su actualización” que vienen a ser varios de los aspectos que se pretende identificar en el recorrido de varias publicaciones antecedentes.

Como uno de los atributos específicos de los LMS, este documento cita a Clarenc (2013) refiriéndose al de escalabilidad y resalta que estos debiesen funcionar con la misma calidad independientemente de la cantidad de usuarios registrados y activos y hace ver que las capacidades nominales del sistema que lo soporta no deben verse comprometidas, sino al contrario podrían crecer sin ver afectada la calidad de sus servicios.

Lo anterior conduce al aspecto tecnológico que necesariamente debe ser evaluado y de lo cual se puede mencionar que en la opinión de Badillo (2007), otro autor citado en este análisis, existen entre otras, la necesidad de evaluar el ancho de banda que las aplicaciones han de requerir pues es un elemento de importancia al momento de querer descargar o visualizar contenidos, igualmente propone que el tipo y capacidad del servidor se definirá en función de la cantidad de usuarios conectados.

De la parte central de esta publicación se extraen algunas características de algunas de las aplicaciones analizadas, entre ellas:

Aquellas que funcionan sobre servidores apache y motor de base de datos MySQL, que por lo general son aplicaciones de código abierto.

Otras más que se alojan en Windows server 2008, con gestor de base de datos Microsoft SQL Server u Oracle 10g que suelen ser las propietarias.

Con esta información se deduce que en su mayoría los sistemas de gestión de aprendizaje que fueron evaluados operan bajo el paradigma cliente / servidor y el mejor de los casos con arquitectura multicapa. Una de las ventajas para uno de los LMS analizados da a conocer que en el mismo servidor se alojan un millón de usuarios.

De las ventajas que algunos presentan se pueden mencionar: las que no tienen límites de usuario en función del dimensionamiento del servidor y la conexión a internet, alta modularidad y tecnologías plug In, diseños con visión de alta escalabilidad, arquitectura flexible, velocidad de carga de datos y acceso a través de dispositivos móviles.

En contraparte las desventajas que se pueden enumerar son: dificultad para abrir archivos, herramientas débiles para creación de contenidos, prescinde de herramientas de roles (role playing), aquellos basados en tecnología PHP deben tomar mayores cuidados al configurar un servidor con muchos usuarios, se puede comprometer la performance del servidor al tener muchos usuarios y algunos sistemas se basan en la comunicación asíncrona.

Otra de las propuestas evaluadas es la de Octavio Ortiz, Pedro Alcover, Francisco Sánchez, Juan Ángel Pastor y Ruth Herrero, que consiste en la implantación de una herramienta *m-Learning* y lo plasman en su documento “Herramienta *m-learning* para el Aprendizaje de Programación Estructurada en los Primeros Cursos de Ingeniería”

En este se hace una evaluación de las tendencias hacia el uso de dispositivos móviles y los avances que se han manifestado en la tecnología inalámbricas y las comunicaciones, aspectos que les han servido de base para realizar su tesis y sus objetivos son: “desarrollar una aplicación en la que el alumno pueda realizar ejercicios de programación desde su móvil, y obtener una evaluación automática e inmediata; y estudiar el grado de aceptación de dicha aplicación y la valoración que hacen los alumnos sobre su efectividad como herramienta de aprendizaje.”

En el plano técnico, hacen ver cómo el diseño de la aplicación lo enfocan en el uso de smartphones, por ser este el dispositivo de mayor uso entre la población estudiantil de universidades y su implementación la llevan a cabo sobre una arquitectura de tres capas Ortiz, et al (2014) siendo: cliente, servidor y base de datos, esta última MongoDB.

Tal implementación tuvo como mercado objetivo un total de 308 estudiantes, la solución aportó una salida inmediata con resultados favorables en cuanto a mejoras de aspectos docentes sobre los cursos que fueron electos.

De este trabajo, se puede comentar, que la movilidad para aspectos de aprendizaje sigue en creciente aumento, los autores hacen notar que el 90 % de la población mundial tienen acceso a las redes móviles y lo ven como una razón para hacer su propuesta; sin embargo, técnicamente se circunscriben únicamente a un reducido número de usuarios, en comparación con la tendencia que hacen saber y de esta cuenta su propuesta se basa en una aplicación sobre el paradigma cliente – servidor que dará respuesta a la necesidad del momento

y no así a la que se visualiza surgirá por el consumo progresivo de dispositivos móviles

Por su lado, Barri Vilardell, I. (2012) hace una propuesta dentro de su tesis doctoral titulada: “Modelado y Planificación de aplicaciones de juegos masivos multijugador en red en entornos distribuidos”, la cual consiste en una arquitectura híbrida que toma en cuenta las ventajas del modelo cliente-servidor y el modelo peer to peer y la denomina *On Demand Game Service* (OnDeGas).

La idea de Barri Vilardell tiene como base el análisis de las arquitecturas de los diferentes tipos de juegos masivos multijugador en red y toma en cuenta, al menos tres de ellos, MMOFPS (*Massively Multiplayer Online First Person Shooter*), MMORPG (*Massively Multiplayer Online Role-Playing Games*) y MMORTS (*Massively Multiplayer Online Real-Time Strategy*) de los cuales hace notar los modelos de servicio y características técnicas.

De igual forma, se analiza el modelo cliente – servidor desde su forma más simple hasta llegar a combinaciones mejoradas que permitan “posibles propuestas de optimización” haciendo ver que se pueden agrupar más de un servidor para dar respuesta a demandas de los clientes, que incluso puedan ser de ubicaciones geográficas diferentes a través de reglas que previamente han sido configuradas para que los usuarios, indistintamente de su ubicación, logren acceso de forma transparente.

Dentro de las formas mejoradas del modelo cliente – servidor, Barri Vilardell destaca los clústeres de servidores, que viene a ser la arquitectura sobre

la que operan los juegos en línea, principalmente los del tipo MMORPG, pues los proveedores guardan sus contenidos en uno o varios servidores para garantizar disponibilidad debido al alto flujo mundial de jugadores.

La propuesta presentada en esta tesis presenta ventajas técnicas sobre otras ya evaluadas y si bien el modelo cliente – servidor es la base de la arquitectura se debe resaltar que el autor propone una mejora considerable en función de la disponibilidad para los clientes.

Otro de los aportes importantes del escritor, es el análisis que lleva a cabo del modelo centralizado cliente – servidor, del modelo descentralizado *Peer to Peer* y del modelo híbrido cliente – servidor y *Peer to Peer* y a su vez hace ver posibles soluciones para cada uno de los modelos indicados, sin embargo, resalta también que existen grandes retos que superar para llevar a la práctica una combinación de arquitecturas del modelo híbrido.

De las conclusiones que se toman de esta publicación se enfatizan las siguientes:

El paradigma cliente-servidor es el más utilizado en la gestión de los juegos MMOG (*Massively Multiplayer Online Games*), dentro de la propuesta híbrida, el servidor central sigue teniendo el control del juego y la opción *Peer to Peer* viene a representar una alternativa para proporcionar servicio a aquellos jugadores que no puede gestionar el servidor.

Como se puede apreciar todas las publicaciones dotan de elementos clave a la presente investigación, sobre todo dejan claro que existe interés por la mejora continua para llevar a cabo implementaciones de nuevas formas de comunicación y colaboración, aspectos que serán de alto valor para este trabajo de graduación y por lo tanto serán tomados en cuenta.

2. JUSTIFICACIÓN

El presente trabajo tiene el propósito de contribuir en la línea de investigación de la Tecnología de la Información y la Comunicación, para apoyo a la Educación, proponiendo adaptar un prototipo de una arquitectura multiservidor utilizada en los juegos en línea de tipo MMO (*Massively Multiplayer Online*) para el uso de aplicaciones educativas y haciendo uso de un prototipo de aplicación con orientación al estudio de matemáticas de primaria inicial.

Con la propuesta anterior se desea dar solución a los siguientes aspectos que fueron identificados en arquitecturas cliente servidor, tales como: actualización de contenidos de manera asíncrona y altos tiempos de respuesta en función de la cantidad de usuarios que se autentican, así como la ineficiencia que se provoca gradualmente en función de las transacciones que son generadas por actividades colaborativas de los clientes, afectando así la eficiencia y escalabilidad por transitar en un único servidor.

Una de las razones tecnológicas, por la cual proponer este tipo de arquitectura, es su capacidad de albergar a miles de usuarios que podrían interactuar, (González González y Blanco Izquierdo, 2011) y obtener contenidos actualizados sin ocasionar una disminución de las capacidades del sistema de servidores y la mejora en el tiempo de respuesta para el cliente.

Adicional se toma en cuenta la tendencia al uso de dispositivos móviles en la educación, lo que se constata en los datos que dan a conocer Aguilar y De la Vega (2014) quienes afirman que en países como Estados Unidos, Japón o

Finlandia el 33 % de las escuelas públicas ya integran servicios educativos a distancia y el 74 % de los jóvenes hacen uso de ellas, Guatemala, en el futuro cercano, no será la excepción y será alcanzada por la influencia de la tecnología educativa.

Por lo anterior, se considera que la innovación de los servicios escolares se dirige hacia el uso de la tecnología como apoyo para el desarrollo de currículos escolares y teniendo en cuenta que a nivel mundial se han vendido más de 195 millones de dispositivos móviles (Alier, 2014) resulta importante contar con plataformas estables que permitan que las aplicaciones educativas puedan ser usadas por miles de estudiantes a la vez.

Dentro de este proyecto se visualizan beneficios no solo tecnológicos sino también de carácter social, ya que aportaría facilidad e inmediatez al acceso de recursos educativos por la vía de un sitio web.

Finalmente, con la adaptación de esta arquitectura de servidores, se estima que existirá un mejor apoyo a la educación porque se permitiría el acceso de incontables estudiantes a aplicaciones educativas.

3. ALCANCES DEL ESTUDIO

El estudio realizado ha sido catalogado como descriptivo, desde luego que se hicieron pruebas, dentro de un ambiente controlado y con un prototipo funcional de lo que se detallan los componentes que fueron utilizados.

3.1. Alcances investigativos

A través de la implementación de este prototipo se abarcaron temas relacionados a la arquitectura en cuestión, sin embargo, también se definen los aspectos a los que se tuvo alcance.

- La investigación se refiere únicamente a la implementación del tipo de arquitectura multiservidor, es decir, toma en cuenta más de un nodo, pero no más allá de cuatro.
- Se limita a la implementación de un sistema de balanceo de cargas, lo cual conlleva dos servidores de servicios web y uno de base de datos, como prototipo para esta investigación.
- La investigación no se enfoca, específicamente, en la funcionalidad del software elegido como plataforma educativa, más bien se hace uso de la misma como elemento auxiliar para determinar las capacidades de la arquitectura.

- La actividad de creación de software no es parte de esta investigación, así como tampoco se entra en detalle en los componentes que puedan integrar cada uno de los equipos seleccionados para esta experimentación.
- No se define la metodología de implementación de las tecnologías involucradas, pero se da a conocer cada una de las que fueron utilizadas.

3.2. Alcances técnicos

Dentro del prototipo de arquitectura implementado, se han homologado funcionalidades que permiten el incremento gradual de los usuarios concurrentes en una plataforma de actividades, para el caso de los juegos, la tecnología utilizada permite una alta concurrencia de jugadores que interactúan en distintos niveles y escenarios de juego, mientras que para el caso de las actividades educativas los usuarios acceden a niveles educativos y cursos, según sean sus intereses o necesidades.

Dentro de este trabajo se ha dado cobertura a las siguientes funcionalidades:

- Implementación de una plataforma educativa, dentro de la cual pueden interactuar usuarios.
- Integración de aplicaciones educativas específicas de contenido

3.3. Resultados esperados

Se integró el prototipo de arquitectura, como base para el alojamiento de plataformas educativas, la misma tiene capacidad para el incremento gradual de usuarios concurrentes, de manera que se puede tener escalabilidad de acuerdo a la demanda que se presente. El modelo se basa en la arquitectura *Massively Multiplayer Online*, MMO y se logró adaptarlo para el uso de una de las plataformas educativas existentes.

4. MARCO TEÓRICO

4.1. Infraestructura tecnológica

Definir una infraestructura tecnológica podría tener más de una acepción pues en cada organización sus componentes pueden variar de acuerdo al giro del negocio, alcance de información que se desea, cantidad en el espacio de almacenamiento, o bien niveles de seguridad que se deseen lograr. Sin embargo, es posible identificar elementos básicos que permitan la conformación del concepto.

De manera general, se puede entender que la infraestructura tecnológica es conformada por *hardware* y *software*, dentro del primer componente todo lo referente a computadoras con funciones de servidor, PC's, dispositivos de comunicación, tales como *routers*, *switches*, puntos de acceso y cableado estructurado, mientras tanto; para el segundo, se identifican aplicaciones que son utilizados sobre la base de los primeros y pueden ser sistemas operativos, sistemas de gestión de bases de datos, protocolos de comunicación, y todo lo que un usuario podría manipular haciendo uso de los equipos de computación.

González Méndez (2011) hace ver como el concepto de infraestructura tecnológica en realidad no obedece a una definición preestablecida sino más bien hace referencia a la etimología de ambas palabras con apoyo de lo indicado en

el Diccionario de la Real Academia Española y a pesar de ello ofrece un aporte al respecto diciendo que “la infraestructura tecnológica de información es el conjunto de elementos que soportan los servicios compartidos que el área de tecnología proporciona a una organización”

Atendiendo a los componentes que pueden conformar la infraestructura tecnológica se tienen algunos elementos esenciales que se ven equiparados en los servidores y la infraestructura de red que toda organización llega a tener, a partir de ello es posible la interconexión de áreas de trabajo y procesos empresariales siendo la función de los servidores el procesamiento y almacenamiento mientras que el transporte de los datos se lleva a cabo en los dispositivos y enlaces de red. (González Méndez, 2011).

Como se observa en el párrafo anterior, lo esencial de las infraestructuras tecnológicas descansa en cómo se procesa, transmite y se almacena la información.

4.2. Modelos de arquitecturas tecnológicas

En la actualidad la existencia de sistemas de gestión educativa es variada y de acuerdo a su evolución se les ha conocido, en su momento, como *b-Learning*, *r-Learning*, *e-Learning* y más recientemente *m-Learning*.

Cada uno de ellos tiene sus propias características tecnológicas, pero en general se pueden identificar aquellas que se desempeñan sobre el modelo

simple cliente /servidor y multiservidor, de los cuales se detallan sus aspectos técnicos:

4.2.1. Modelo cliente / servidor

En términos generales, se refiere a un conjunto de computadoras que mantienen comunicación bidireccional, en el que uno de los equipos solicita información a otro de ellos, donde el primero es denominado cliente y el segundo servidor.

Por lo regular, las capacidades, relacionadas con el hardware, de los equipos cliente son menores al compararlas con los de servidores; sin embargo, son los que inician el enlace de comunicación.

“Desde el punto de vista funcional, se puede definir la computación cliente – servidor como una arquitectura centralizada que permite a los usuarios finales obtener acceso a la información de forma transparente aun en entornos multiplataforma” (Barri Vilardell, 2012).

Villar Monserrat (2011), identifica en este modelo tres componentes básicos, el cliente, el servidor y la infraestructura de las comunicaciones y hace notar que el primero es quien tiene a su cargo procesos auxiliares para el establecimiento de la conexión y comunicación con el servidor; para el segundo elemento, se refiere al resultado que este otorga al cliente y en tal sentido la carga computacional, de este equipo, es mayor; en cuanto al tercer elemento, es basado a protocolos orientados a la conexión como lo es TCP o no orientados a la conexión para el caso de UDP.

De las ventajas que este autor propone se pueden listar las siguientes:

- Gestión centralizada de los recursos en los procesos de servidor.
- El uso de componentes, indistintamente del fabricante, es posible ya sea hardware o software, lo cual contribuye en la reducción de costos.
- Las interfaces que se utilizan por lo general residen en el cliente de manera que el ancho de banda que se consume, en el momento de la transmisión es mínimo.
- Este modelo propicia una estructura modular que favorece la escalabilidad.

Desventajas identificadas:

- Se requiere niveles de seguridad que eviten uso malicioso del sistema y de los recursos disponibles en el servidor.
- Si la comunicación falla o existen congestionamientos por las peticiones ocasionará un problema.

4.2.2. Modelo multiservidor

La alta demanda que en la actualidad manifiestan los usuarios, de cualquiera de los sistemas de información existentes, ha ejercido presión en el ámbito de la tecnología y en consecuencia se obtienen respuestas tecnológicas que favorecen a quienes esperan resultados inmediatos, prueba de ello es posible observar la evolución que se ha generado en cuanto al uso de arquitecturas centralizadas hacia modelos multiservidor o bien clúster de servidores.

Un clúster es un conjunto de computadoras que se une mediante una red de alta velocidad y que tiene la característica de hacerse ver como un único equipo funcional de mayor capacidad, (Sapiña Rubio, 2012), o como también lo define Mesa Múnera (2009), es una arquitectura computacional de alto rendimiento que es capaz de emular el comportamiento de un sistema MPP (del inglés Massively Parallel Processing) para proveer una determinada solución.

Para que un sistema de servidores de este tipo sea efectivo se debe tomar en cuenta cada uno de sus elementos, Sapiña Rubio (2012) indica que consisten en la agrupación de nodos con características de servidores dedicados interconectados por una red; sistemas operativos de tipo servidor con características de multiproceso y multiusuario; *middleware* de clúster, como medio que actúa entre los sistemas operativos y las aplicaciones finales de usuario; conexiones de red, que pueden ser simples como las *fast Ethernet* o avanzadas como las *Gigabit Ethernet*; protocolos de comunicación para definir la intercomunicación de los nodos y finalmente sistemas de almacenamiento, que igualmente pueden ser simplemente lo que ofrece un servidor como aquellos más sofisticados como una NAS o SAN.

4.3. Balanceo de carga

De acuerdo a la definición de Diaz Henao, T.M.; Ruiz López y E.G.; Sinisterra, M.M. (2012) un balanceo de carga consiste en un “Clúster que permite que un conjunto de servidores compartan la carga de trabajo y de tráfico a sus clientes. Está compuesto por uno o más ordenadores (llamados nodos) que actúan como *front-end* del clúster y se ocupa de repartir las peticiones de servicio que reciba el clúster a otros ordenadores que forman su *back-end*.”

El propósito de este servicio es garantizar la disponibilidad de los servicios que ofrece un sistema de información en particular, previene la disminución de funcionalidad que pudiera ocasionar los altos volúmenes de peticiones de usuarios, las altas cargas de procesamiento y un elevado tráfico de red. La característica de este tipo de clúster son la robustez que implica sus componentes y su alto grado de escalabilidad, (Rubio Sapiña, 2012)

4.4. Infraestructura EC2

La modernidad informática trae consigo nuevas opciones para la implementación de soluciones, considerando a su vez una mejora considerable en el costo que esto implica. El concepto de computación en la nube no solo favorece la reducción de las inversiones sino también ofrece las mismas o mejores capacidades que si se tratara de un equipo físicamente instalado, es decir, se cuenta con la potencia de los servidores que se conocieron hasta antes de la aparición de este modelo ubicado en la nube.

Bocchio, (2014) indica que el concepto de infraestructura EC2 (del inglés Elastic Compute Cloud) se refiere a “un servicio web que proporciona capacidad informática con tamaño modificable en la nube, diseñado para facilitar a los desarrolladores recursos informáticos escalables” en el que se puede instalar cualquier sistema operativo, sistema de gestión de base de datos, así como aplicaciones específicas.

La gestión de estos sistemas operativos, instalados en esta plataforma, seguirá siendo administrada tal como si se tratara de un servidor en sitio, pudiendo otorgar permisos de acceso, cuotas de disco, restricciones, etc.

Otro factor innovador es el hecho de que sea modificable lo cual le identifica como elástica, pues en cuestión de minutos las capacidades instaladas pueden aumentarse o disminuirse de acuerdo a la necesidad del momento, se le considera como totalmente controlable, flexible, fiable, seguro, y económico (Ibid).

4.5. Arquitecturas para e-learning basadas en la nube

Otro de los modelos tecnológicos que surgen a partir del crecimiento de la cantidad de usuarios que demandan servicios educativos, y que viene en apoyo a los sistemas de gestión del aprendizaje es el denominado *cloud computing*.

Silva-López, Méndez-Gurrola y Herrera Alcántara (2015) definen al *cloud computing* como “un modelo que permite compartir un conjunto de recursos computacionales en el que intervienen servidores, almacenamiento, aplicaciones y servicios, en línea a través de redes. Emplea técnicas innovadoras a gran escala para acceder a recursos que se encuentran distribuidos geográficamente.”

De acuerdo a esta definición se puede saber que el modelo presta una serie de servicios con características que cumplen los requerimientos de alta demanda y se puede percibir que la arquitectura responde a altas tasas de flujo de información generadas por una masiva autenticación de usuarios.

Adicional, el *cloud computing* se ha diversificado en la forma en que puede ser contratado, es decir, otorga distintas modalidades de contratación de acuerdo a la necesidad que se requiera solucionar y en tal sentido ofrece tres servicios a saber:

Software como servicio, SaaS, aquí se cuentan las aplicaciones para usuarios finales; Infraestructura como servicio, IaaS, se provee de un espacio en el que el usuario alojará sus herramientas para sus propósitos y plataforma como servicio, PaaS, que vienen a ser las utilerías de las que puede disponer el usuario.

En conjunto con lo anterior las formas de implementación también son variadas, cuatro modelos específicamente, Nube pública, nube privada, nube híbrida y nube comunitaria. (Ibid)

Figura 2. **Arquitectura de Cloud Computing**



Fuente: Silva-López, Méndez-Gurrola y Herrera Alcántara (2015) p. 176.

4.6. Arquitectura de entornos virtuales con agentes

El apartado de arquitecturas multiservidor da ya una noción de los componentes que intervienen en este modelo, adicional al concepto de *cloud computing* que se define, también en este documento, nos orienta al conocimiento de estructuras complejas y dinámicas que se disponen para la ejecución de proyectos educativos.

Existen otros elementos que resultan importantes describir, pues aportan valor a la investigación, tal es el caso de los Agentes, que tienen una función dinamizadora de los objetos ubicados en uno o varios repositorios o servidores para el logro de una comunicación ágil. “Los agentes son una tecnología surgida de la Inteligencia Artificial que utiliza el concepto de agente como componente principal para el diseño y desarrollo de sistemas inteligentes y distribuidos.” (Celorio Aguilera, 2010) como lo indica Méndez Pozo (2008), es una herramienta de abstracción para desarrollar software que permite situarnos un escalón por encima de los objetos y componentes y que, utilizados de forma adecuada, facilitan la construcción de sistemas complejos.

Enfocado en los entornos virtuales, los agentes pueden ser capaces de aportar mayores capacidades de respuesta a sistemas por la alineación que pueden otorgar entre dispositivos móviles cliente y el servidor. (Celorio Aguilera, 2010)

4.7. Experiencia del usuario

La evolución de las tecnologías, sin duda, sería diferente sin la intervención de los usuarios, son los humanos quienes han dado características especiales a todo tipo de sistemas a tal punto que se ha llegado a interactuar con ellos de manera que pueden reflejarse en las personas aspectos emotivos como resultado del uso continuo de *hardware* y *software*.

La experiencia del usuario es entonces una forma de definir la práctica que tiene un usuario cuando interactúa con tecnología en general, “hasta definirse como la evaluación de las interacciones entre los usuarios y los productos tecnológicos con la finalidad de determinar una experiencia de gran calidad en la utilización de cualquier sistema” (Cordoba-Cely, 2013), no obstante, los resultados de percepción también podrían ser negativos.

Garret (2011) define cinco áreas básicas para que la experiencia del usuario sea satisfactoria: estrategia, en el que evalúa las necesidades del usuario y los objetivos del sitio; alcance, toma en cuenta las especificaciones funcionales y los requerimientos de contenido; estructura, el diseño de la interacción de usuarios y la arquitectura de la información, skeleton, para definir el diseño de la interface, el diseño de la navegación que conforman el diseño de la información; finalmente contempla la superficie que viene a ser el diseño visual.

Así, tomando en cuenta estos elementos, la experiencia del usuario se torna más objetiva y menos subjetiva, pues implica el seguimiento de una metodología que permitirá garantizar resultados positivos.

4.8. Juegos masivos en línea

Uno de los tipos de juegos en línea que actualmente goza de gran popularidad son los *Massively Multiplayer Online Role-Playing Game* o MMORPG como mejor se les conoce en este ámbito.

La notoriedad de estos juegos no se refiere solo al hecho de que otorgan diversión a sus usuarios sino a la capacidad de que los jugadores que interactúan simultáneamente se cuenten por miles, por lo que el aspecto tecnológico que lo soporta resulta de gran importancia porque están preparados y elaborados de tal manera que admiten cualquier número de autenticaciones de manera simultánea, (Barri Vilardell, 2012).

De las características principales se puede hacer ver que se ejecutan en un modelo tecnológico de base, cliente – servidor, sin embargo, para lograr responder a las múltiples peticiones a nivel mundial, los proveedores almacenan el contenido en varios servidores centrales que son los responsables de garantizar el acceso, así como permitir una experiencia satisfactoria para el usuario.

Este esquema de servicio suele ser mejorado con otro modelo al que se le conoce como clúster de servidores, de manera que el volumen de peticiones concurrentes pueda ser soportado por esta arquitectura.

Barri Vilardell (2012), también hace notar algunas posibles desventajas de este tipo de juegos y menciona “que se necesita de una gran cantidad de recursos

de desarrollo, requieren la constante gestión de los mundos virtuales, significativos recursos de hardware, tales como servidores y ancho de banda, además de un departamento de soporte dedicado”

4.9. Sistemas de gestión de aprendizaje

Hablar de la educación desde la perspectiva de las TIC´s conlleva conocer sobre las diferentes modalidades que para el fin se tiene a disposición, cada una define su forma de acción de acuerdo a su función principal o bien la especialidad a la que es orientada, en este apartado se define a uno de ellos.

La Fundación para el Desarrollo de la Función Social de las Comunicaciones (FUNDESCO), citada por Suclla Ardiles (2015), define el *e-Learning* como “Un sistema de impartición de formación a distancia, apoyado en las TICs que combina distintos elementos pedagógicos: instrucción clásica (presencial o autoestudio), las prácticas, los contactos en tiempo real (presenciales, videoconferencias o chats) y los contactos diferidos (tutores, foros de debate, correo electrónico)”

Como actividad el *e-Learning* se lleva a cabo sobre plataformas tecnológicas que permiten a los usuarios acceder a servicios que incluso van más allá de la acción propia de aprender o enseñar, para ello existen sistemas de gestión de aprendizaje o como bien se les conoce en este ámbito LMS (del inglés *Learning Management System*).

Los sistemas de gestión del aprendizaje son diseñados sobre arquitecturas de comunicación que toma en cuenta modelos como el cliente/servidor para soportar las peticiones que múltiples usuarios hacen de manera concurrente, además de contar con sistemas de gestión de bases de datos, repositorios de información y software que permiten su función en general.

Definiendo un LMS se puede decir que es un software que presta servicios para el logro de la gestión de actividades de educación que permite la coordinación de usuarios, profesores y administradores, además de organizar contenidos para el desarrollo de cursos, la comunicación entre todos los involucrados, así como la administración de actividades de apoyo.

Suclla Ardilles (2015) amplia diciendo “es un programa que permite organizar materiales y actividades de formación en cursos, gestionar la matrícula de los estudiantes, hacer seguimiento de su proceso de aprendizaje, evaluarlos, comunicarse con ellos mediante foros de discusión, chat o correo electrónico, etc...”

Otras de las tecnologías que se identifican dentro de un LMS son: las transmisivas, que se refieren al hecho de transmitir presentaciones dentro de un curso; Interactivas, aquellas que proporcionan medios de actividades, cuestionarios, evaluaciones y por último las colaborativas que se ven a través de los foros de discusión, espacios de consulta como wikis. (Suclla Ardiles, 2015).

5. PRESENTACIÓN DE RESULTADOS

5.1. Descripción de la arquitectura MMO (Massively Multiplayer Online) y los componentes que la integran

La propuesta comprende varios componentes para lograr su funcionamiento, se basa en la ejecución de un clúster de servidores gestionados por un servicio de balanceo de carga, quien determina la carga de red, los elementos son los siguientes:

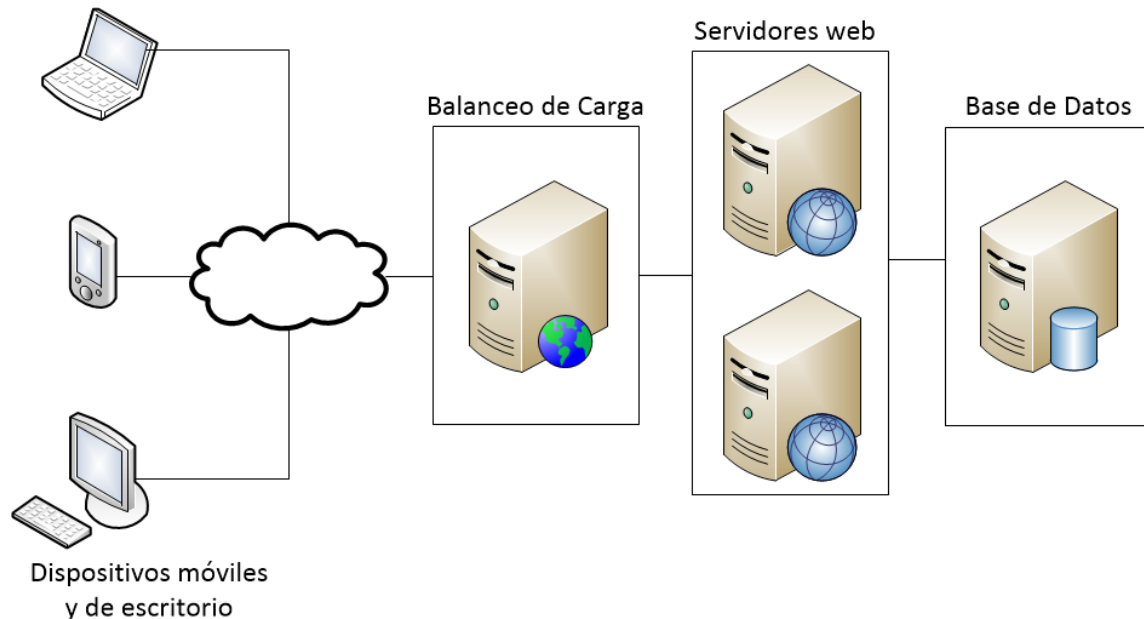
Modelo multiservidor distribuido en cuatro capas, la primera la constituyen los dispositivos que acceden vía web a los servicios, la segunda el servicio de balanceo de carga, una tercera que aloja los servicios de la plataforma educativa a través de dos servidores web y una cuarta que la constituye el servidor de base de datos.

Balanceo de carga, como se ve en la figura 3 es el primero en recibir las peticiones de los usuarios y quien se encarga de distribuir a los nodos, para este caso *webserver1* y *webserver2*, los requerimientos de dispositivos que desean acceder a la plataforma educativa.

Servidores *web*, el prototipo se diseñó con dos servidores, siendo ellos los que alojan a los usuarios y gestionan las actividades que llevarían a cabo, es en estos que se ejecuta la aplicación educativa denominada *vstudent*.

Servidor de base de datos, dentro de este servidor se almacenan todos los resultados que se generan del uso de la aplicación educativa.

Figura 3. Diagrama de la arquitectura cliente – multiservidor basada en MMO



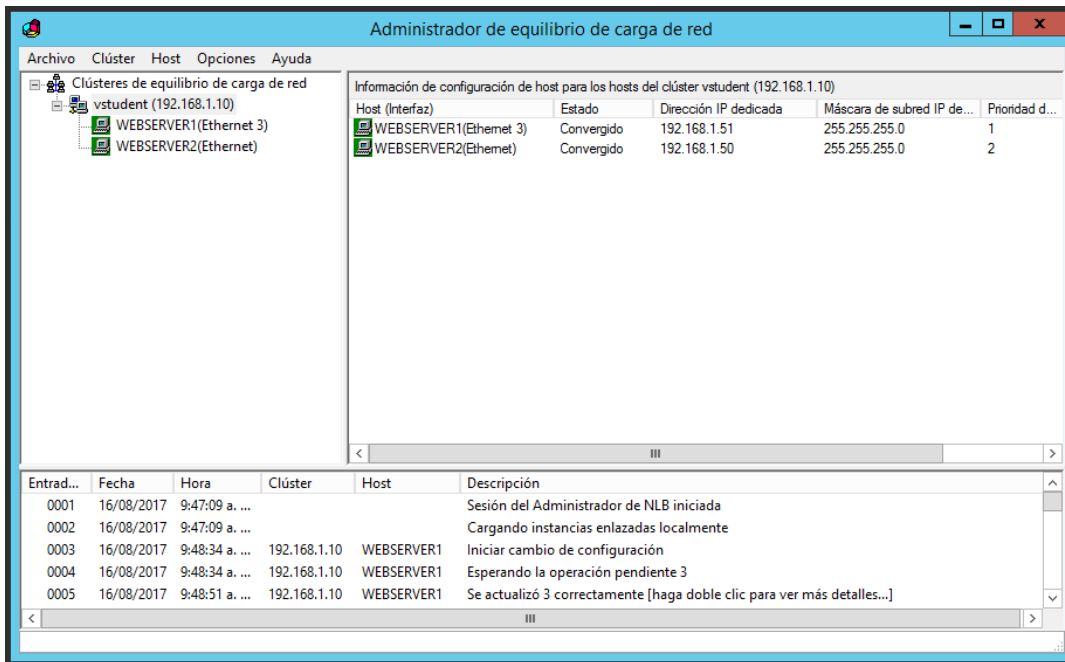
Fuente: elaboración propia.

5.2. Descripción de la infraestructura necesaria para soportar la arquitectura MMO

Este ejercicio de implementación se apoyó en las características del balanceo de carga que ofrece el sistema operativo Windows server 2012, consistente en una tecnología de clúster que permite la escalabilidad y la disponibilidad de los recursos que son configurados, los mismos se identifican como nodos y para el caso de esta propuesta se definieron como webserver1 y webserver2.

La función del servicio NLB (del inglés *Network Load Balance*) se caracteriza por garantizar la permanencia activa de los servicios que presta la arquitectura que pueden llegar a ser de misión crítica, su desempeño se da por medio de la distribución del tráfico de red, dentro de los diferentes hosts que integran el grupo de servidores disponibles, como es el caso de la propuesta que se refleja en la figura 4 en la que se ve las configuraciones que se realizaron y que involucra a los nodos que intervienen en este clúster.

Figura 4. Panel de configuración del clúster denominado vstudent

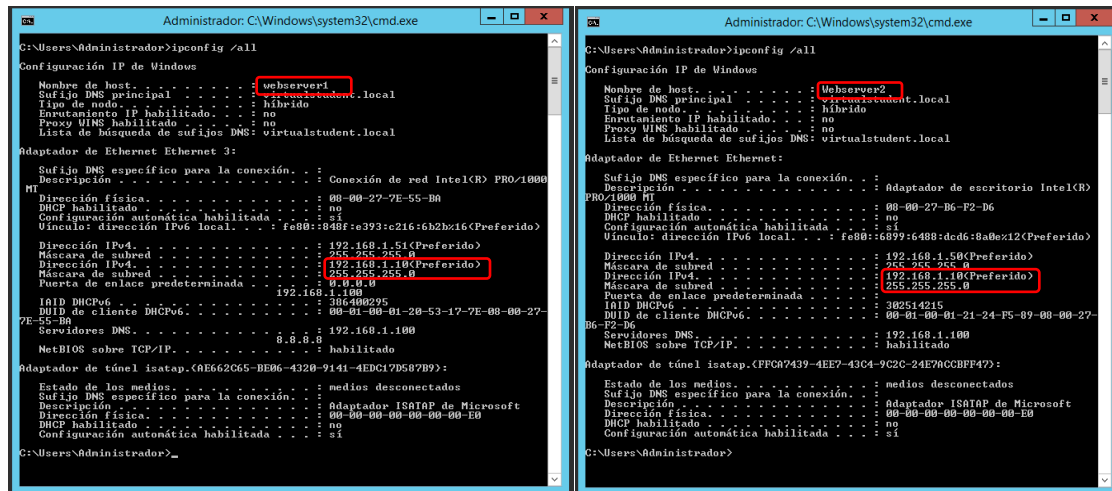


Fuente: elaboración propia.

Dentro de los elementos que integran este conjunto de servidores se identifica la presencia de un servidor que se identifica con la IP 192.168.1.10 que es quien hace las veces de controlador de las peticiones y a su vez quien gestiona la asignación de los servidores.

En cada uno de los servidores se asigna este servicio, de manera que permitan ser controlados por el balanceador de carga, en la figura 5 se puede observar la forma en que este servicio se integra a cada uno de los servidores.

Figura 5. Configuración de balanceo de carga en los nodos del clúster

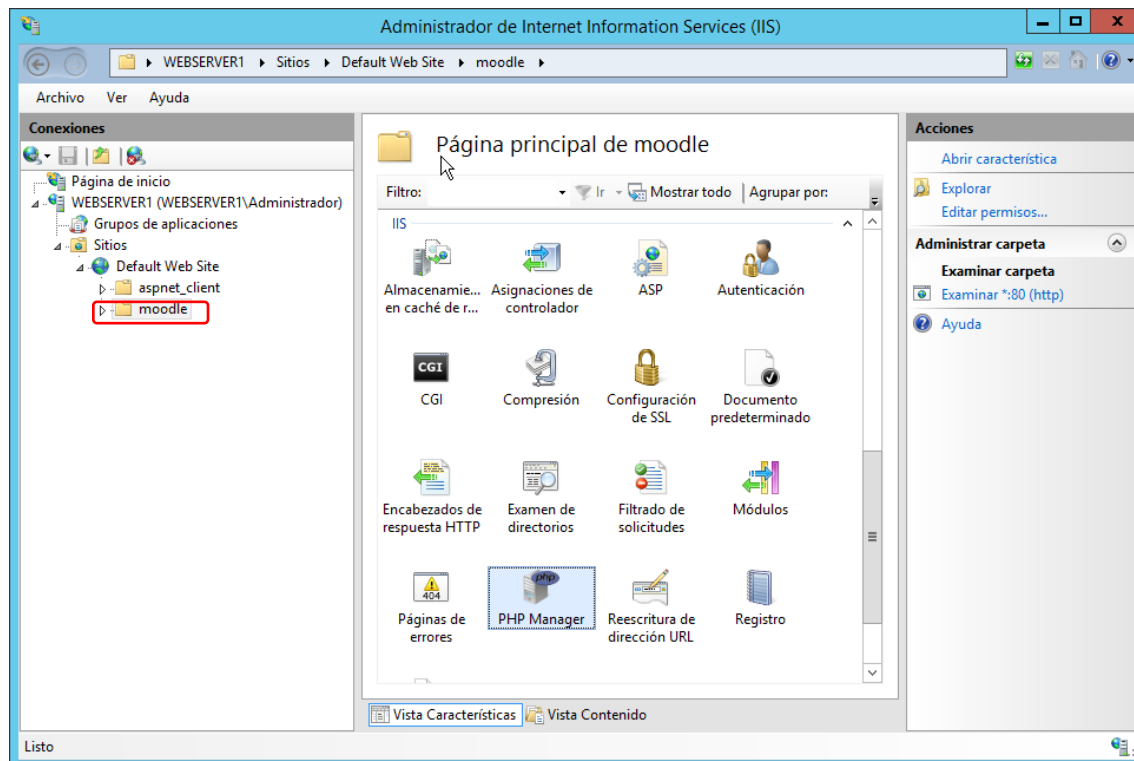


Fuente: elaboración propia

5.3. Servidor web

La configuración que se realizó aplica a cada uno de los nodos miembros del clúster, los servidores *web* han sido configurados con sistema operativo Windows server 2012 Estándar R2 y dentro de este el servicio de IIS 8.5 (*Internet Information Server*) para la configuración, administración y publicación del sitio *web* reflejado en la aplicación educativa, este a su vez contó con ayuda de un intérprete de contenido, PHP versión 5.6.30, para soportar la funcionalidad de Moodle 3.2.2 que fuera instalado en cada uno de los nodos como plataforma educativa. Una visión gráfica de los servicios configurados se ve en las figuras 6 y 7.

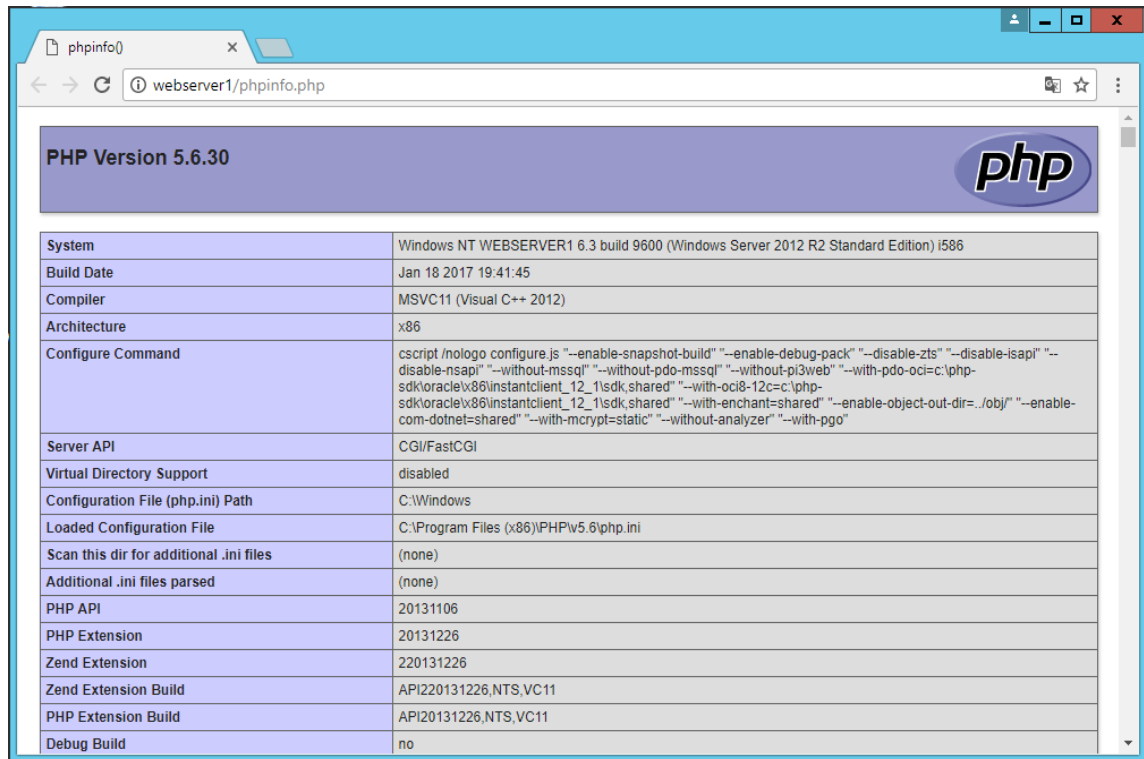
Figura 6. Panel de administración de Internet Information Server



Fuente: elaboración propia.

Para el acceso a la aplicación, desde un navegador, es posible a través del puerto 80 que es el configurado por defecto en el IIS.

Figura 7. Panel de parámetros de php

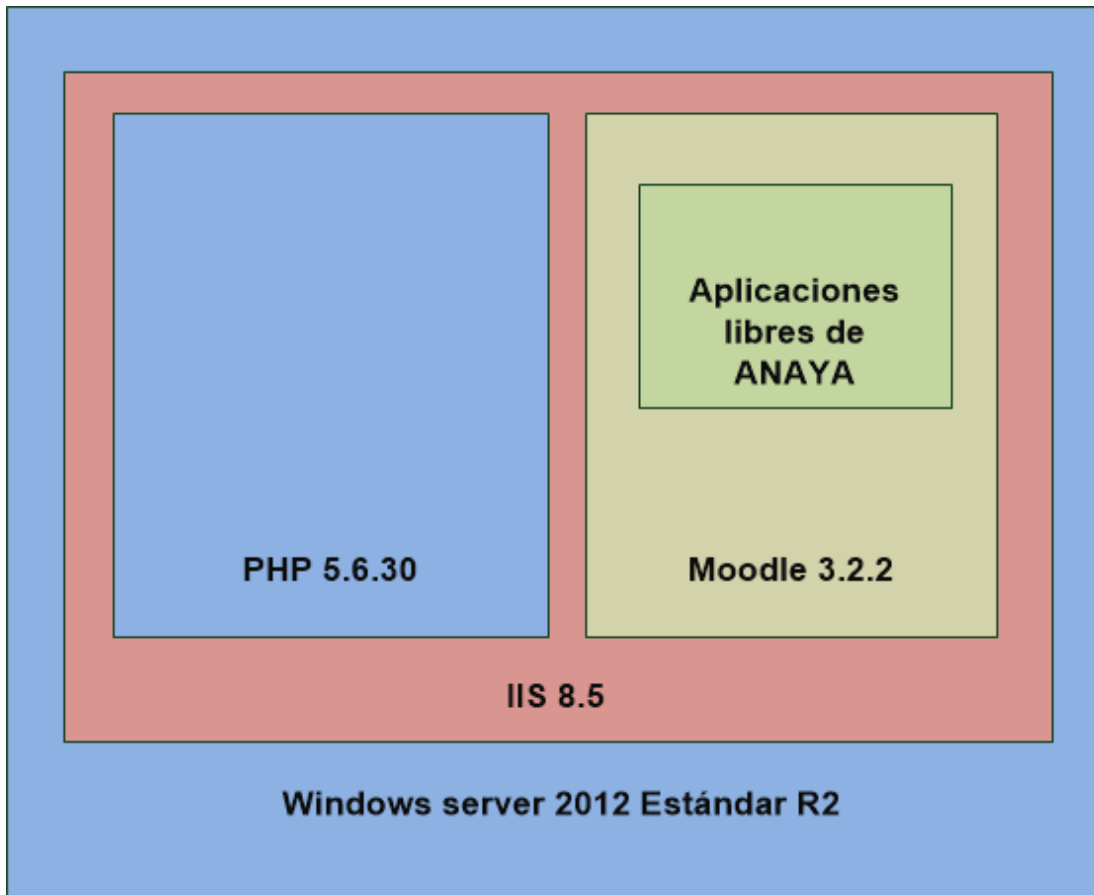


| PHP Version 5.6.30 | |
|---|---|
| System | Windows NT WEBSERVER1 6.3 build 9600 (Windows Server 2012 R2 Standard Edition) i586 |
| Build Date | Jan 18 2017 19:41:45 |
| Compiler | MSVC11 (Visual C++ 2012) |
| Architecture | x86 |
| Configure Command | ccscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--disable-zts" "--disable-isapi" "--disable-nsapi" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sd\oracle\86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sd\oracle\86\instantclient_12_1\sdk,shared" "--with-enchant=shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo" |
| Server API | CGI/FastCGI |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | C:\Windows |
| Loaded Configuration File | C:\Program Files (x86)\PHP\5.6\php.ini |
| Scan this dir for additional .ini files | (none) |
| Additional .ini files parsed | (none) |
| PHP API | 20131106 |
| PHP Extension | 20131226 |
| Zend Extension | 220131226 |
| Zend Extension Build | API220131226,NTS,VC11 |
| PHP Extension Build | API20131226,NTS,VC11 |
| Debug Build | no |

Fuente: elaboración propia.

Como se puede observar cada miembro del clúster configurado conto con la tecnología mínima para el funcionamiento de la aplicación que sirvió de apoyo en la determinación de capacidades de la arquitectura seleccionada, y en términos generales la distribución del *software*, se implementó como se puede ver en la figura 8.

Figura 8. **Distribución del software en los nodos miembros del clúster**

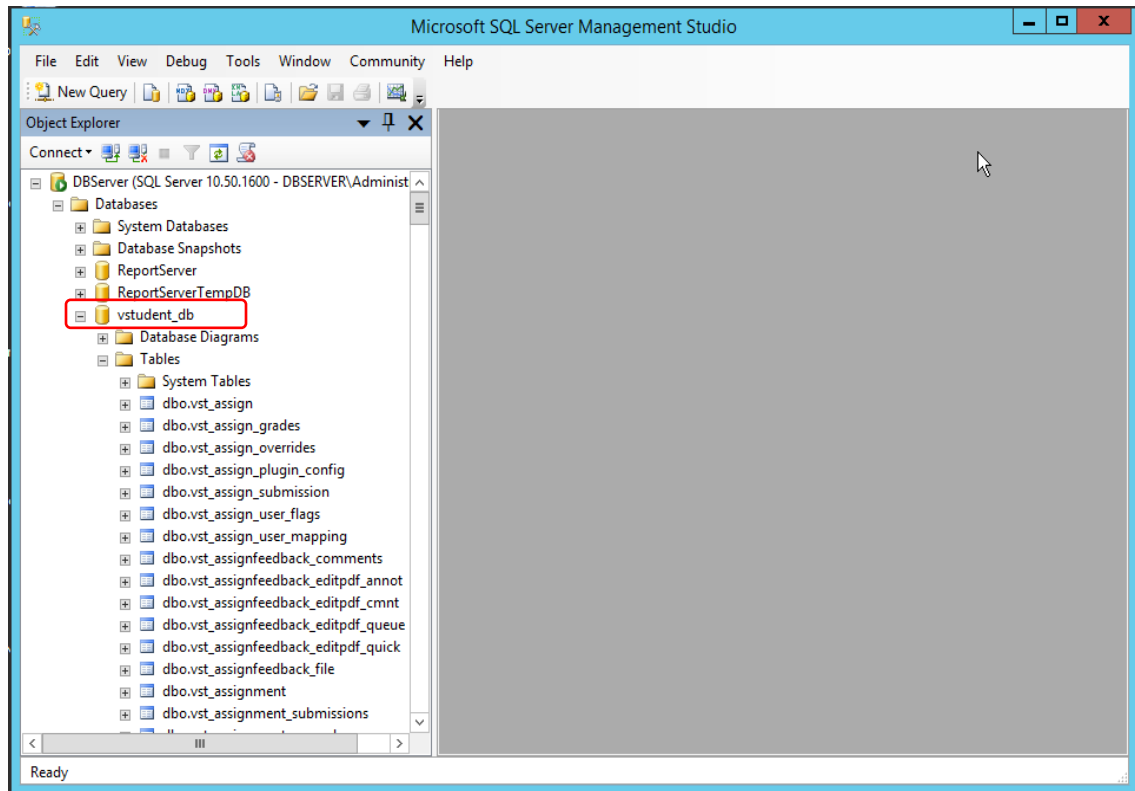


Fuente: elaboración propia.

5.4. Servidor de base de datos

En lo que se refiere a los datos, se configuró una base de datos denominada `vstudent_db` sobre un servidor con sistema operativo Windows server 2012 Estándar R2, para la gestión de la misma se seleccionó SQL server 2008 R2 con acceso a ella a través del puerto 1433. La figura 9 muestra el gestor, así como la base creada y parte de las tablas para su funcionamiento.

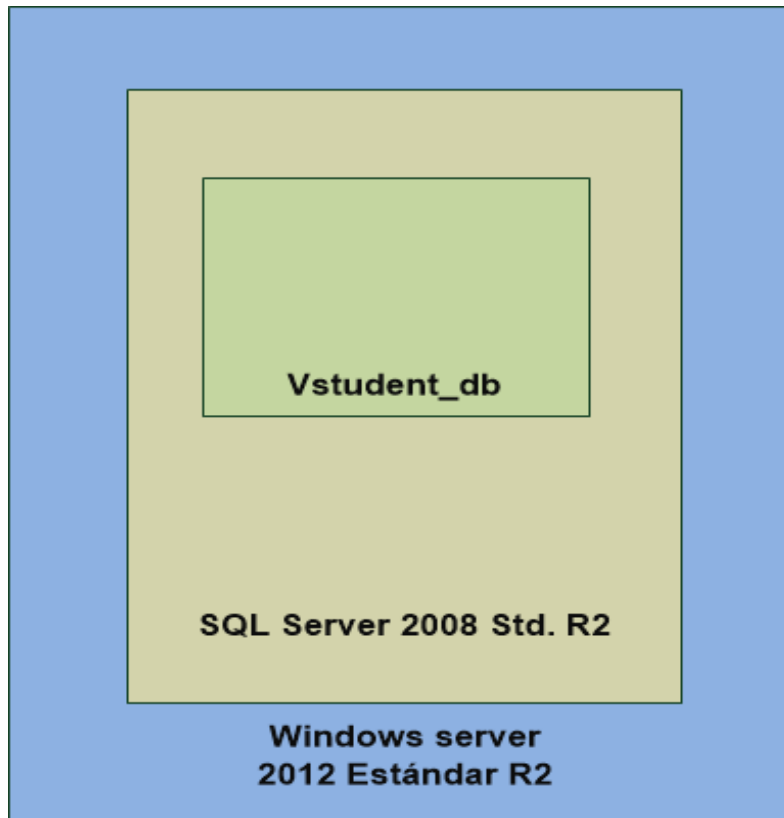
Figura 9. Panel del control del gestor de la base de datos



Fuente: elaboración propia.

La figura 10 muestra la distribución del software dentro del nodo DBserver.

Figura 10. **Distribución de software en servidor de base de datos**



Fuente: elaboración propia.

5.5. Estructura de red

La estructura de red que se diseñó para este prototipo tomó en cuenta lo siguiente:

Tabla II. Estructura de red

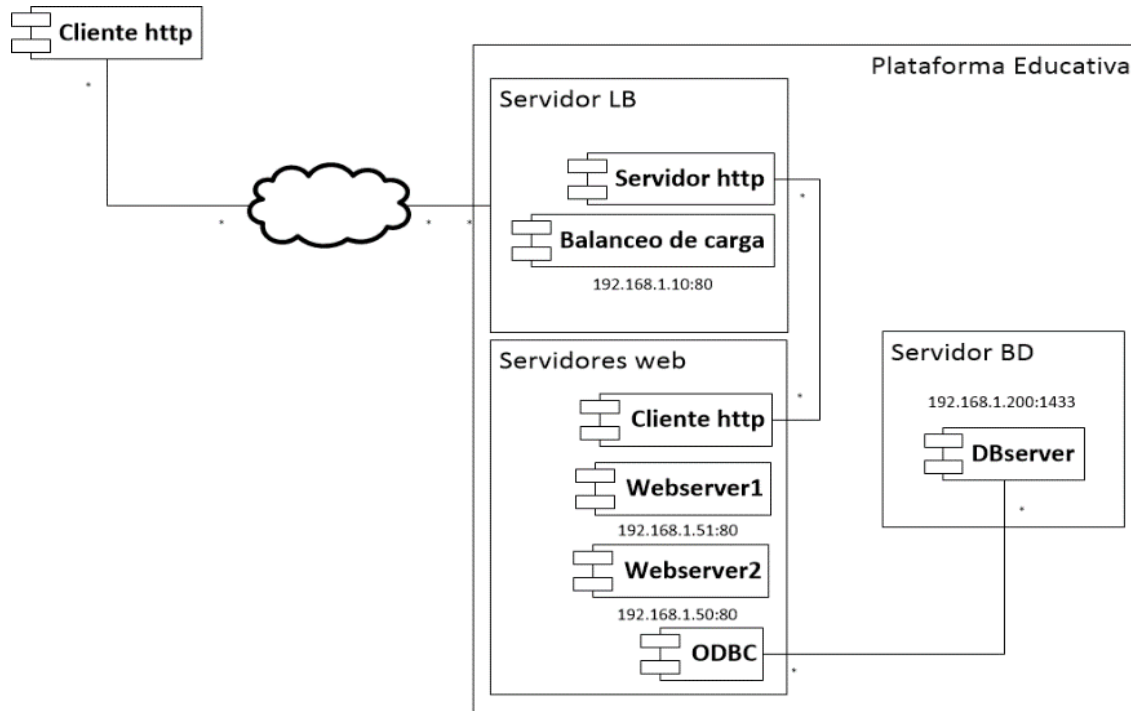
| No. | Nodo | IP | Puerto de acceso |
|-----|----------------------|---------------|------------------|
| 1 | Balanceador de carga | 192.168.1.10 | 80 |
| 2 | Webserver1 | 192.168.1.51 | 80 |
| 3 | Webserver2 | 192.168.1.50 | 80 |
| 4 | Dbserver | 192.168.1.200 | 1433 |

Fuente: elaboración propia.

La función de balanceo redirige las peticiones de los clientes a los nodos, a través del puerto 80, quienes por medio de las configuraciones realizadas en el servicio de IIS escuchan en el mismo puerto.

Los nodos de servicio web trasladan datos generados en la aplicación vstudent a la base de datos vstudent_db, servidor que escucha peticiones en el puerto 1433. Vea la figura 11.

Figura 11. Estructura de red



Fuente: elaboración propia.

5.6. Descripción del experimento

El trabajo realizado se llevó a cabo tomando en cuenta los siguientes escenarios:

- Arquitectura cliente / servidor
- Arquitectura cliente / multiservidor

en ambos casos se utilizaron cuatro grupos de usuarios, en cantidades de 50, 100, 150 y 250, con el objetivo de determinar capacidades en el tiempo de respuesta y rendimiento de las arquitecturas evaluadas.

La obtención de los datos se hizo a través de la simulación de la carga de los usuarios y concurrencia de los mismos, dentro de las arquitecturas, con las herramientas *Webserver stress tool* y *Monitor de rendimiento de Windows server*.

Para el caso de la herramienta de pruebas de estrés los parámetros establecidos para cada escenario consistieron en determinar las capacidades de los equipos durante 10 minutos de corrida con carga creciente para cada grupo de usuarios y un tiempo de espera, entre cada clic, de 25 segundos.

La herramienta utilizada para determinar el rendimiento de las arquitecturas tuvo a cargo monitorear el consumo de recursos de memoria RAM, procesadores y ancho de banda de los adaptadores de red.

5.6.1. Escenario con arquitectura cliente / servidor

Las características para recrear este escenario fueron las siguientes:

Sistema operativo *Windows server 2012 Estándar R2* con 4 Gb. De memoria Ram, 2 procesadores de 2.20 GHz y disco duro de 100 Gb. En la figura 3 se observan los componentes seguido de un diagrama de la arquitectura con equipos que intervienen en el modelo.

5.6.2. Escenario con arquitectura multiservidor

Las características para recrear este escenario fueron las siguientes:

Cliente web, plataforma educativa, servidor de aplicaciones web, servidor de la base de datos y el balanceo de carga. Las capacidades de los equipos involucrados son: sistema operativo *Windows server 2012* estándar R2 con 4 Gb. de memoria Ram, 2 procesadores de 2.20 GHz y disco duro de 100 Gb.

5.6.3. Determinación de las capacidades

Para esta actividad se integraron las arquitecturas que fueron objeto de comparación, por medio de la herramienta de medición, *Webserver Stress Tool*, se llevó a cabo pruebas de estrés en cada una de ellas.

Para cada uno de los ambientes implementados se tuvieron al menos cuatro escenarios de comparación, el primero con carga de 50 usuarios concurrentes durante diez minutos de manera incremental y una interacción continua con 25 segundos entre cada una de ellas.

Para el segundo escenario se incrementaron los usuarios a 100, para el tercero a 150 y para el último de ellos a 250 usuarios, manteniendo los mismos parámetros de tiempos, en la duración de la prueba y los segundos entre cada interacción.

Para todos los escenarios el tiempo de respuesta, para acceder a la aplicación, se estableció en 5,000 milisegundos.

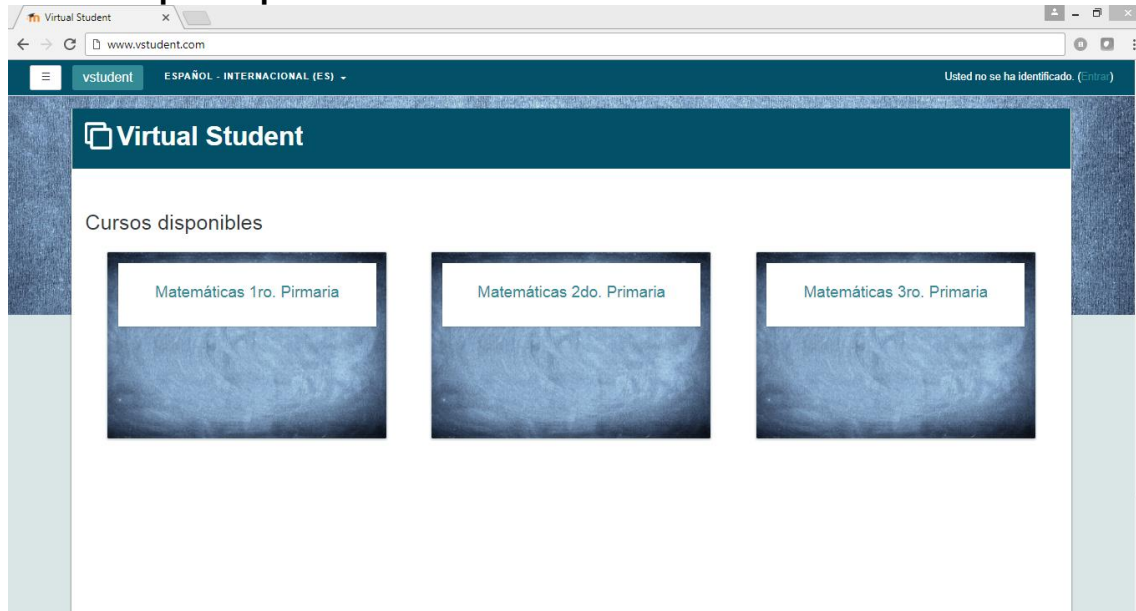
Dentro de los resultados se tiene varias modalidades de representación de los datos obtenidos, gráficas de las frecuencias establecidas en las pruebas del escenario cliente / servidor, además, tablas conteniendo información de los recursos que se utilizaron del equipo afectado. Del servidor, en cada uno de los modelos de arquitectura se identificó el rendimiento del procesador, consumo de memoria RAM, así como el comportamiento de los adaptadores de red.

5.6.4. Funcionalidades de la aplicación

La aplicación contó con un curso de matemáticas para primaria inicial, que corresponde a los niveles de 1º, 2º y 3º. Identificados como módulos, en cada uno de ellos se insertaron aplicaciones de acuerdo a los niveles en las que se desarrollaron actividades. La figura 12 muestra la interfase de la aplicación configurada.

Se utilizó una serie de herramientas tecnológicas *open-source* y de tipo propietario para desarrollar el experimento, con la integración de todos los componentes necesarios se obtuvo un escenario que permitió evaluar todas las variables que se definieron en la investigación para lo cual se tomó en cuenta Moodle versión 3.2.2 como plataforma educativa, PHP 5.6.30 como lenguaje de programación para HTML, aplicaciones educativas de ANAYA, con orientación matemática que se integraron a la plataforma Moodle, IIS 8.5 como servidor web, SQL Server, para la gestión de base de datos.

Figura 12. Interfase web de la aplicación de apoyo en la evaluación del prototipo



Fuente: elaboración propia.

Las actividades que se lograron llevar a cabo, dentro de la aplicación configurada, se detallan en tabla III para cada uno de los niveles del curso.

Tabla III. **Actividades dentro de la aplicación**

| Nivel | Actividad |
|------------------|--|
| 1o. Primaria | Identifica cual es el número y escribe en letras |
| | Coloca el número que sigue en la fila y completa la carga del tren. |
| | Llena los cuadros con los números que van antes y después del número que aparece en medio. |
| 2do. Primaria | Selecciona pares o impares |
| | Ordenamiento numérico |
| 3ro. Primaria | Descomposición de cifras |
| | Valor de las unidades |
| | Comparación |

Fuente: elaboración propia.

De manera gráfica, de algunas de las actividades que se detallan en la tabla III, se pueden observar las figuras 13, 14 y en la que se aprecia la interface realizada.

Figura 13. Actividad de identificación de números del 0 al 9 en el nivel de primero primaria



Fuente: elaboración propia.

Figura 14. Actividad de selección de números pares/impares en el nivel segundo primaria

The screenshot shows a web application interface for a math activity. The page title is "Selección de pares o impares". The interface includes a sidebar with navigation options such as "2do. Primaria", "Participantes", "Insignias", "Competencias", "Calificaciones", "General", and "Actividades". The main content area displays a list of numbers: 133, 76, 347, 233, 124, 147, 43, 474, 168, 18. Below the numbers are two columns labeled "PARES" and "IMPARES" for selection. The interface also includes a top header with the user's name "Marco Ramirez" and a logo for "ANAYA".

Fuente: elaboración propia.

Figura 15. Actividad de lectura y descomposición de cifras en el nivel de tercero primaria

Descomposición de cifras

3 PRIMARIA / SEGUNDO CICLO ANAYA

Actividad 1. Lectura y descomposición

EJEMPLO

| NÚMERO | DESCOMPOSICIÓN | DESCOMPOSICIÓN | SE LEE |
|--------|----------------|----------------|-----------------------------|
| 375 | 3C + 7D + 5U | 300 + 70 + 5 | Trescientos setenta y cinco |

?

| | | | |
|-----|--------------|--------------|-------------------------------|
| 946 | 9C + 4D + 6U | 900 + 40 + 6 | Cuatrocientos sesenta y nueve |
| 496 | 6C + 4D + 9U | 600 + 40 + 9 | Cuatrocientos noventa y seis |
| 469 | 4C + 9D + 6U | 400 + 60 + 9 | Seiscientos cuarenta y nueve |

| NÚMERO | DESCOMPOSICIÓN | DESCOMPOSICIÓN | SE LEE |
|--------|----------------|----------------|-----------------------------|
| 649 | | | |
| | 4C + 6D + 9U | | |
| | | 400 + 90 + 6 | |
| | | | Novecientos cuarenta y seis |

Fuente: elaboración propia.

5.6.5. Limitaciones en la experimentación con cliente / servidor

- Se tuvo la limitante de llevar a cabo pruebas en un ambiente de nube debido a la utilización de herramientas de simulación como Webserver Stress Tool que generaron carga incremental en el servidor, lo cual fue identificado como un ataque a la plataforma y con ello se provocó el cierre de los puertos de acceso a la arquitectura.
- El uso de herramientas de simulación fue necesario para determinar si la arquitectura soportaba altas cantidades de usuarios.

5.6.6. Limitaciones en la experimentación con cliente / multiservidor

Al inicio del experimento se pudo identificar algunas limitaciones que afectaron la implementación de este tipo de arquitectura, y se ha considerado importante poder enumerarlas a efecto de que puedan servir como indicios para investigaciones futuras.

- La interacción de los usuarios no siempre será dentro del mismo servidor, debido al funcionamiento del sistema de balanceo de cargas.
- Los usuarios no pueden elegir a que servidor ingresar y en caso que los contenidos sean distintos, en los servidores implicados, verán entornos diferentes.
- No existe, entre los servidores, un sistema de sincronización de contenidos solo un balanceo de carga como gestión de usuarios y maximización del rendimiento y escalabilidad.
- De origen, la plataforma educativa elegida para este proyecto no ofrece las capacidades, para ser implementada en una arquitectura cliente / multiservidor.
- La herramienta seleccionada para las pruebas de carga, se ejecutó de manera simultánea en varios equipos cliente, para que el sistema de balanceo de carga pudiera identificar más de una solicitud de servicios.

5.7. Implementación de la infraestructura en EC2

5.7.1. Implementación

Implementar la arquitectura propuesta en ambiente de la nube requiere contar con los requerimientos mínimos para lograr el objetivo, cabe destacar que en este medio tecnológico se tiene a disposición algunas soluciones, de las que se debe tomar en cuenta la más adecuada.

Una de las soluciones consiste en la configuración del entorno sobre servicios de VPS (*Virtual Private Server*) que son espacios configurados de acuerdo a la necesidad que se presente, sin embargo, tienen la característica de ser una porción de un servidor mayor y por lo tanto capacidades relativas a lo que se desea instalar sobre ellos. La solución de servidores privados virtuales, para este caso, puede no ser la mejor alternativa para este tipo de proyectos.

La siguiente solución consiste en la contratación de servidores dedicados, a diferencia de los anteriores, estos ofrecen un equipo completo con características de mayor alcance en cuanto a disco duro, memoria RAM y procesadores, razón por la cual resulta ser la mejor opción para el proyecto.

Elegida la infraestructura y tomando en cuenta servidores dedicados, los equipos tendrían las características siguientes:

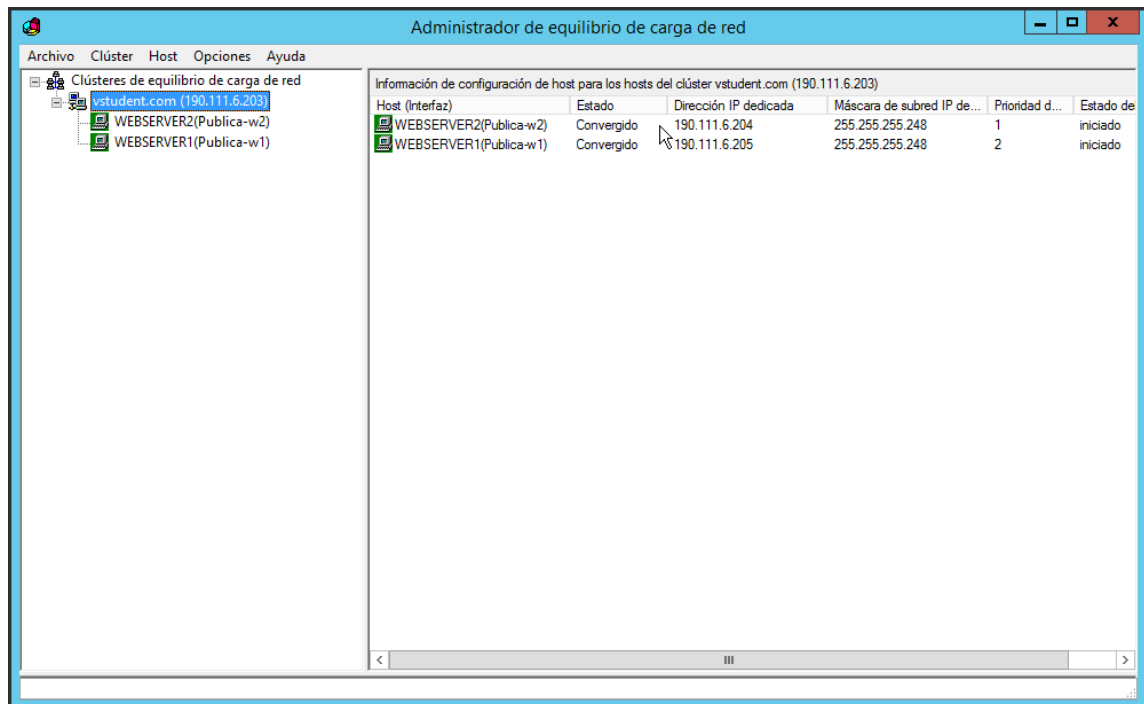
Procesador Intel Xeon E5-2600 V4 a 2.4 MHz de hasta 18 núcleos, memoria RAM de 32 Gb. Disco duro de 2 Tb. Sistema Operativo *Windows server 2016 Standard*.

El software específico para cada uno de ellos será de acuerdo a la función que desempeñan:

5.7.2. Balanceo de carga

Se debe habilitar y configurar el servicio de NLB (*Network Load Balance*), se habilitan, al menos, dos tarjetas de red, una de ellas con IP pública y tiene la función de otorgar acceso a los clientes, a los servicios que deben ser publicados mediante un dominio, la siguiente funciona dentro de la red interna como enlace de comunicación con el resto de equipos. La figura 16 muestra los nodos convergidos con IP's públicas.

Figura 16. Panel de configuración del clúster



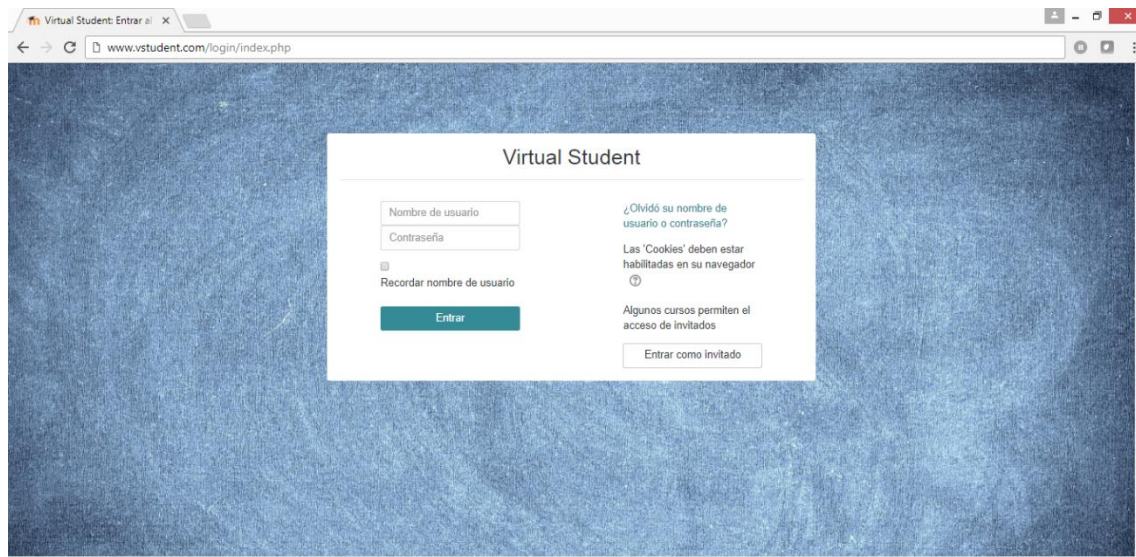
Fuente: elaboración propia.

5.7.3. Webservers

Habilitar y configurar IIS (*Internet Information Server*) para la gestión del sitio de la aplicación. Dada la naturaleza, de código abierto, de la aplicación se requiere la configuración de PHP como gestor, dentro de IIS, del sitio.

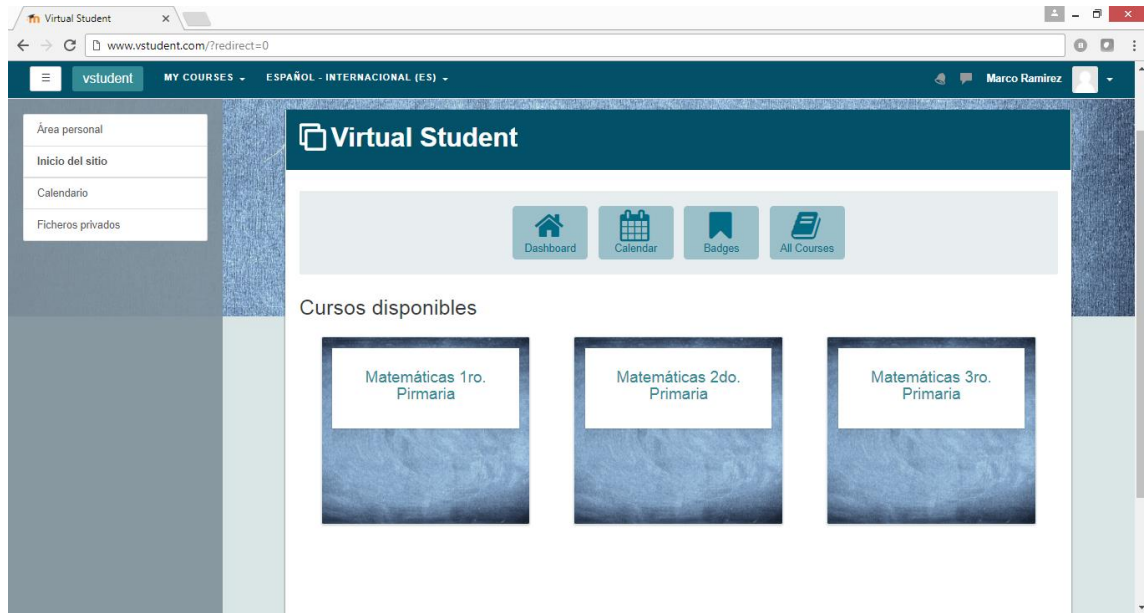
Instalación y configuración de la aplicación Moodle, para el funcionamiento en este tipo de arquitectura, la configuración adecuada de los ficheros config.ini y php.ini es importante. Posterior a la instalación y configuración de lo anterior las interfaces de acceso y vista principal serían como se pueden ver en las imágenes 17 y 18.

Figura 17. Interfase de acceso a la aplicación



Fuente: elaboración propia.

Figura 18. Vista del panel principal de la aplicación



Fuente: elaboración propia.

5.7.4. Base de datos

Gestor de base de datos *SQL server*, en este se instalará la base de datos que contiene las tablas necesarias para la colección de datos que se generan en la aplicación Moodle. Véase la figura 9, en la que se observa la instalación.

5.7.5. Estructura de costos

La proyección económica se basa en el requerimiento de las configuraciones establecidas en el apartado anterior. La tabla IV contiene la información de los aspectos que son necesarios para la implementación en un ambiente local.

Tabla IV. **Costos para implementación de servidor local**

| Descripción | Años | | | | |
|----------------------------|--------------------|--------------------|--------------------|--------------------|---------------------|
| | 1 | 2 | 3 | 4 | 5 |
| Servidores físicos | \$7,550.09 | \$7,550.09 | \$7,550.09 | \$7,550.09 | \$7,550.09 |
| Software | \$955.80 | \$955.80 | \$955.80 | \$955.80 | \$955.80 |
| Energía Eléctrica | \$5,835.61 | \$5,835.61 | \$5,835.61 | \$5,835.61 | \$5,835.61 |
| Mantenimiento | \$1,381.50 | \$1,381.50 | \$1,381.50 | \$1,381.50 | \$1,381.50 |
| Enlace de Internet | \$7,200.00 | \$7,200.00 | \$7,200.00 | \$7,200.00 | \$7,200.00 |
| Infraestructura en la nube | | | | | |
| Costos Anuales | \$22,923.00 | \$22,923.00 | \$22,923.00 | \$22,923.00 | \$22,923.00 |
| | | | Costo Total | | \$114,765.01 |

Fuente: elaboración propia.

En cuanto a la implementación que se propone, en ambiente de nube, los elementos que son necesarios y sus costos se observan en la tabla V.

Tabla V. **Costos para implementación de servidor en nube**

| Descripción | Años | | | | |
|----------------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| | 1 | 2 | 3 | 4 | 5 |
| Servidor físico | | | | | |
| Energía Eléctrica | | | | | |
| Mantenimiento | | | | | |
| Enlace de Internet | | | | | |
| Infraestructura en la nube | \$16,086.72 | \$16,086.72 | \$16,086.72 | \$16,086.72 | \$16,086.72 |
| Costos Anuales | \$16,086.72 | \$16,086.72 | \$16,086.72 | \$16,086.72 | \$16,086.72 |
| | | Costo Total | | | \$80,433.60 |

Fuente: elaboración propia.

Los costos que se presentan en las tablas anteriores se tomaron en cuenta de las cotizaciones presentadas en mayo de 2017 por proveedores de tecnología, IngeniaLA, Implementación de soluciones y TIGO, se llevó a cabo una evaluación y se determinó que las características técnicas y económicas más adecuadas son las del oferente Implementación de Soluciones.

Se debe tomar en cuenta los costos para los ambientes, local y de nube, se determina la erogación que cada usuario debería efectuar, según la arquitectura seleccionada, la tabla VI muestra los valores.

Tabla VI. **Costo por usuario de acuerdo a la arquitectura**

| Ambiente | Costo | Total Usuarios | Costo anual | Costo mensual |
|-----------------|--------------|-----------------------|--------------------|----------------------|
| Local | \$22,923.00 | 819 | \$ 27.99 | \$ 2.33 |
| Nube | \$16,086.72 | 819 | \$ 19.64 | \$ 1.64 |

Fuente: elaboración propia.

6. DISCUSIÓN DE LOS RESULTADOS

En este capítulo se trata de comprender los resultados obtenidos como resultado de la implementación del prototipo que fuera propuesto, consistente en una arquitectura multiservidor basada en la de los juegos masivos multijugador en línea, véase la figura 3, y para ello se evalúan los indicadores definidos en la metodología de investigación que consisten en una variable general denominada experiencia del usuario y dos subvariables, la primera tiempo de respuesta y la segunda rendimiento de la arquitectura, igualmente se incluye una parte dedicada a saber cómo se disminuyen los costos haciendo uso de este tipo de infraestructura.

Las arquitecturas evaluadas fueron las siguientes:

- Arquitectura cliente - servidor
- Arquitectura cliente – multiservidor

Para cada una de las arquitecturas evaluadas, se tiene un conjunto de datos que involucra los indicadores propuestos para esta investigación, de los cuales se tienen las siguientes consideraciones en los apartados que en adelante se detallan, se hizo énfasis en discutir los hallazgos que permitieron evidenciar las limitaciones de la arquitectura tradicional de un servidor para luego comparar con las evidencias del prototipo propuesto.

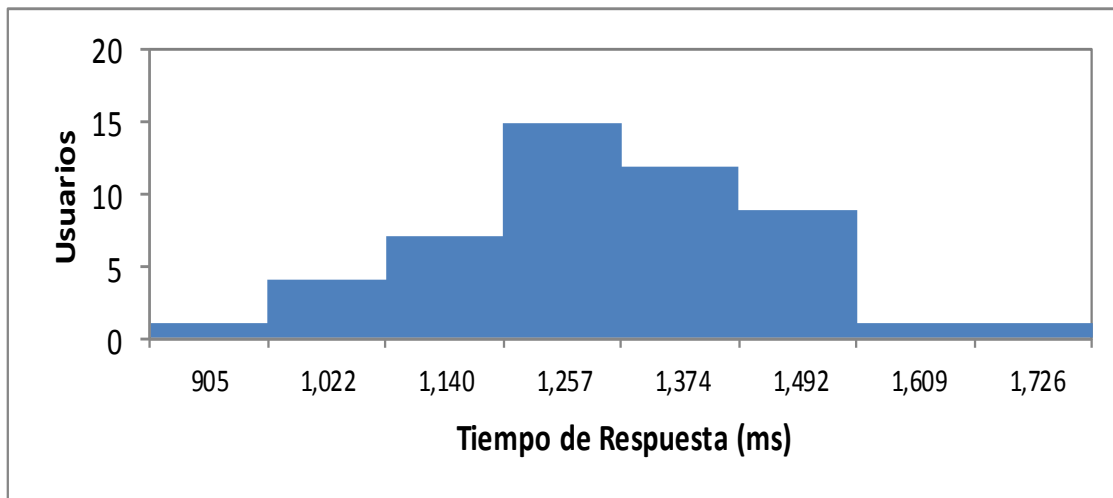
6.1. Tiempo de respuesta

6.1.1. Evaluación de la arquitectura cliente / servidor

6.1.1.1. Cantidad de milisegundos en la carga de la aplicación

Para el primero de los indicadores establecidos en esta investigación, se determinó el tiempo de respuesta que la arquitectura otorgó para la carga de la aplicación, obteniendo los datos que se pueden apreciar en la figura 19 y tabla VII, en la que se pueden identificar los recursos que fue posible medir durante la prueba.

Figura 19. **Milisegundos para carga de 50 usuarios concurrentes en cliente / servidor**



Fuente: elaboración propia.

Tabla VII. **Recursos evaluados del servidor con 50 usuarios**

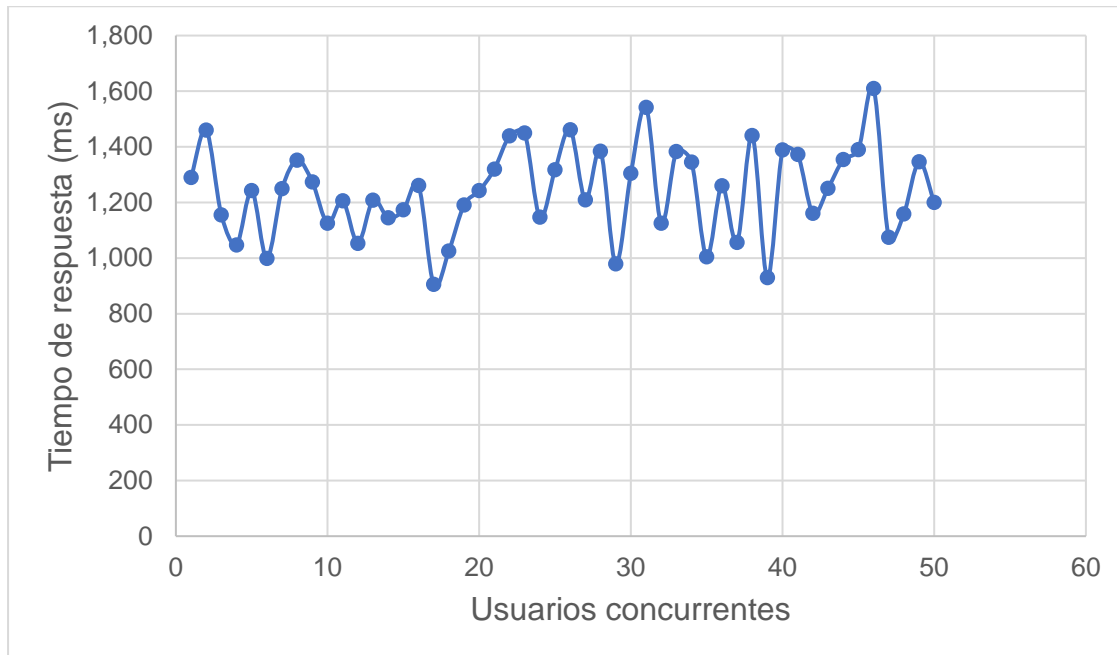
| Componente | Uso |
|-------------------|------------|
| CPU | 24 % |
| Memoria | 45 % |
| Bits | 296,914 |

Fuente: elaboración propia.

En cada uno de los casos, evaluados, el adaptador de red tuvo una capacidad de 1 Gbit por lo que la transmisión de bits, para este caso representa el 0.03%. La tabla VII muestra el porcentaje de uso de los recursos del servidor.

Los datos de la gráfica 1 muestran cómo se distribuyeron los usuarios, en los rangos respectivos y en la figura 20 se aprecia el incremento del tiempo de respuesta mientras la totalidad de usuarios permanece activa dentro de la plataforma.

Figura 20. Despliegue de 50 usuarios concurrentes

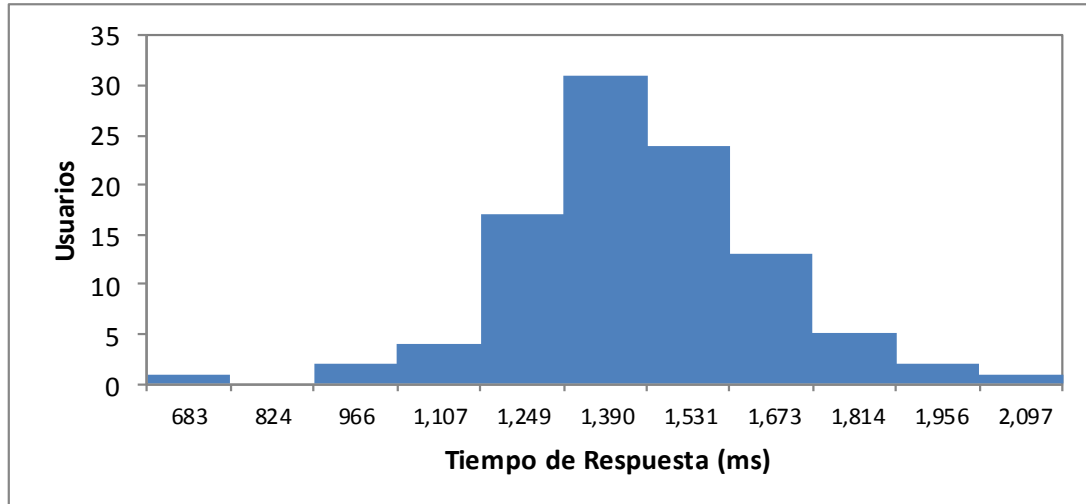


Fuente: elaboración propia.

El siguiente escenario evaluado tomó en cuenta la cantidad de 100 usuarios concurrentes, habiendo obtenido los datos que se presentan en la figura 21 y tabla VIII.

De la misma forma la figura 22 muestra como es el comportamiento de los 100 usuarios, estando activos de manera simultánea.

Figura 21. **Milisegundos para carga de 100 usuarios concurrentes en cliente / servidor**



Fuente: elaboración propia.

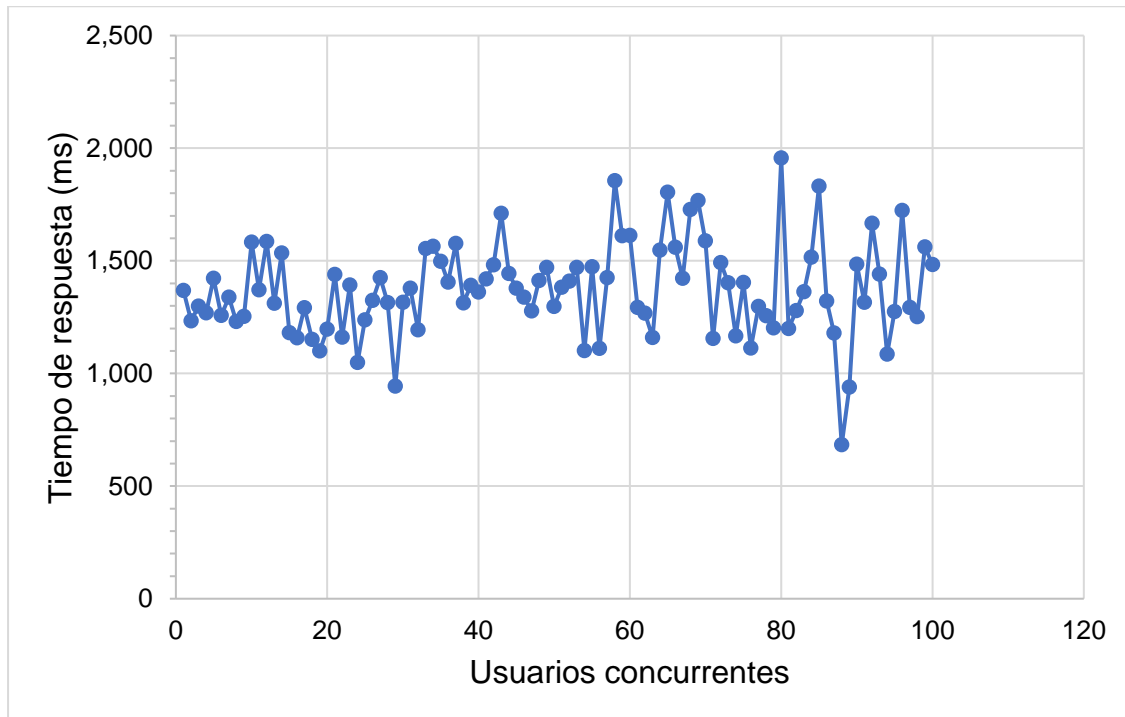
Tabla VIII. **Recursos evaluados del servidor con 100 usuarios**

| Componente | Uso |
|------------|---------|
| CPU | 41 % |
| Memoria | 49 % |
| Bits | 559,064 |

Fuente: elaboración propia.

La tabla VIII hace referencia a los recursos utilizados del servidor en pruebas para la evaluación de 100 usuarios, el dato de red asciende al 0.05% sobre la capacidad del adaptador de red.

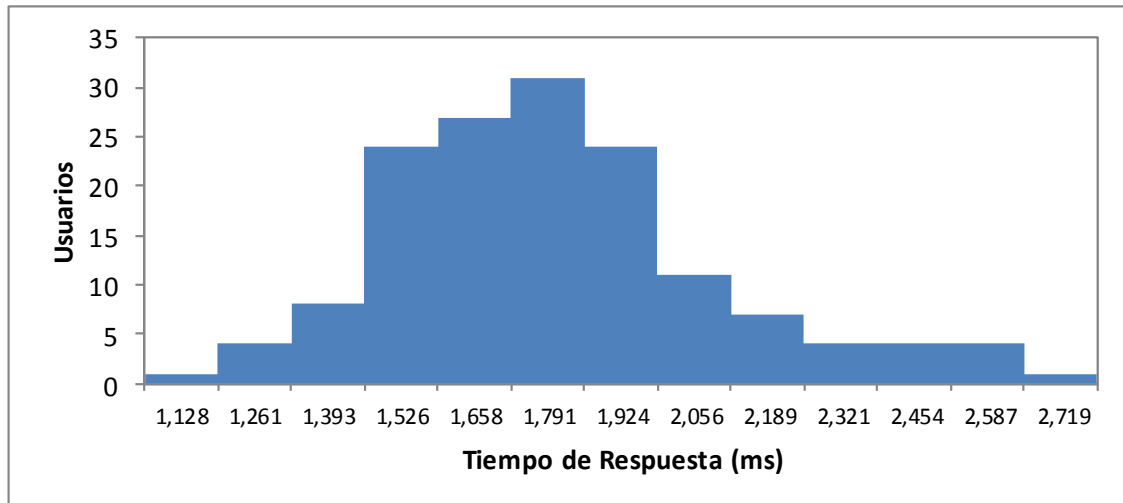
Figura 22. Despliegue de 100 usuarios concurrentes



Fuente: elaboración propia

Para el caso de los 150 usuarios los datos generados por la arquitectura muestran ya variaciones en el tiempo de respuesta, pudiéndose notar como ha incrementado el tiempo de espera, del lado del usuario, aunque esta sea poco perceptible. La figura 23 y tabla IX reflejan lo descrito, en tanto que la figura 24 muestra a esta cantidad de usuarios, en cada uno de sus tiempos de manera lineal mientras tienen actividad dentro de la plataforma.

Figura 23. **Milisegundos para carga de 150 usuarios concurrentes en cliente / servidor**



Fuente: elaboración propia.

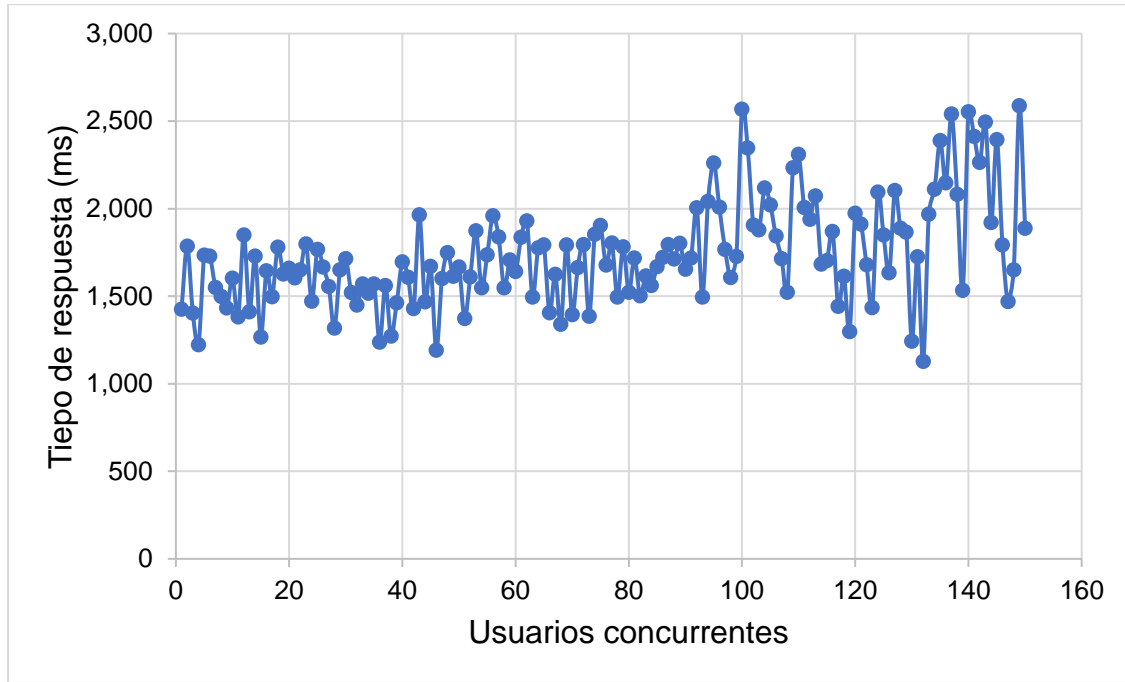
Tabla IX. **Recursos evaluados del servidor con 150 usuarios**

| Componente | Uso |
|------------|---------|
| CPU | 59 % |
| Memoria | 55 % |
| Bits | 812,312 |

Fuente: elaboración propia.

Otro dato importante, fue la transmisión de bits sobre el adaptador de red, que para este escenario se acerca al megabit y que representa un 0.08%.

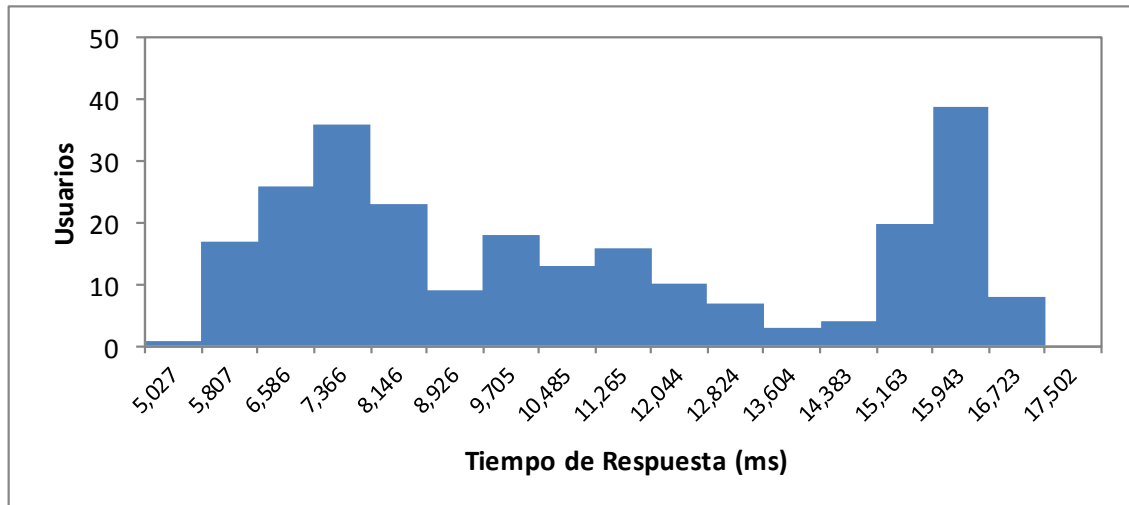
Figura 24. Despliegue de 150 usuarios concurrentes



Fuente: elaboración propia.

En el experimento final bajo el modelo de arquitectura cliente / servidor se contó con el escenario de mayor cantidad de usuarios, que corresponde a la cantidad de 250. Véase la figura 25 y tabla X para los detalles. La figura muestra como el tiempo de espera se incrementa más allá de límite establecido de 5,000 milisegundos, que es coherente con el comportamiento de los recursos de CPU y memoria RAM, que ascienden al 76% y 64% respectivamente, ver tabla X, en la figura 26 se ve la forma en que estuvieron activos los usuarios.

Figura 25. **Milisegundos para carga de 250 usuarios concurrentes en cliente / servidor**



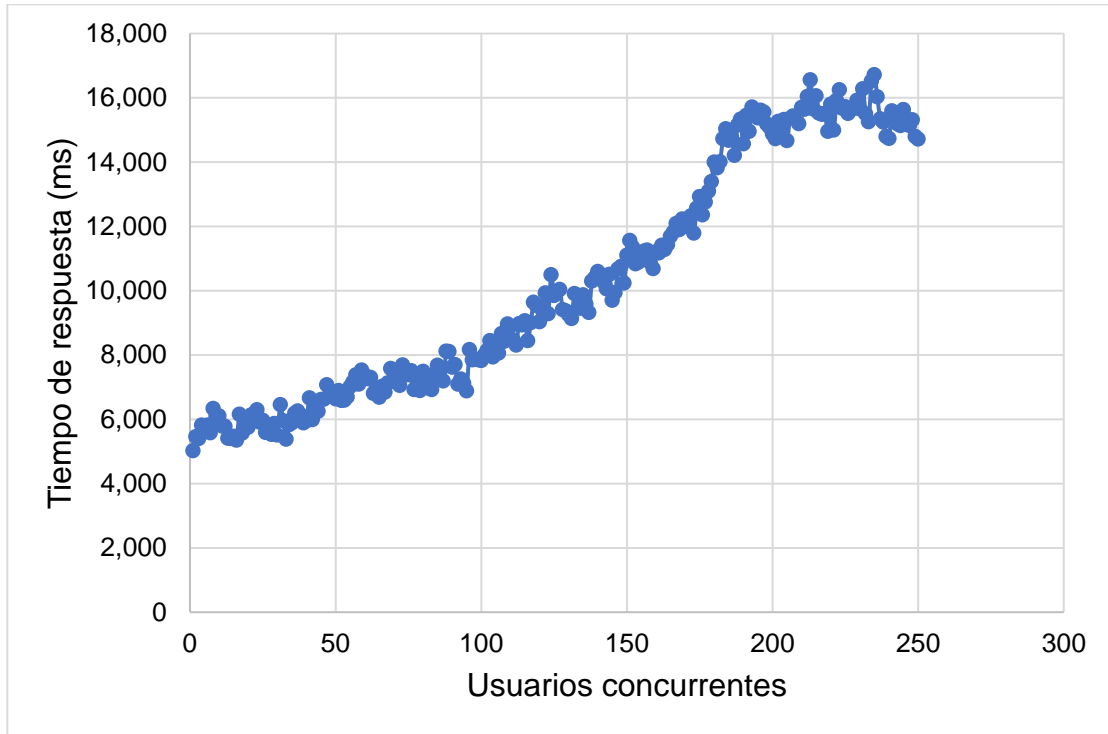
Fuente: elaboración propia.

Tabla X. **Recursos evaluados del servidor con 250 usuarios**

| Componente | Uso |
|------------|-----------|
| CPU | 76 % |
| Memoria | 64 % |
| Bits | 1,056,744 |

Fuente: elaboración propia.

Figura 26. **Despliegue de 250 usuarios concurrentes**



Fuente: elaboración propia.

6.1.1.2. Resumen del escenario cliente / servidor

El uso de las diferentes gráficas tiene el propósito de refrendar el comportamiento de la arquitectura en cada uno de los escenarios, en ese sentido, la interpretación de los datos a través de barras permite visualizar, por medio de rangos, la cantidad de usuarios que fue posible identificar con el mismo tiempo de respuesta, durante el tiempo que se ejecutó la herramienta de pruebas de estrés, en tanto que las gráficas de dispersión lineal permiten visualizar que efectivamente la cantidad de usuarios, en todas las pruebas, permanecieron activos, es decir, permanecieron concurrentes.

Como podrá verse en la tabla XI los datos que se obtuvieron en cada uno de los escenarios puestos a prueba demuestran como en cada uno de ellos se comportó cada recurso de la arquitectura.

Resalta el que los resultados del último escenario excedan la media en el tiempo de respuesta esperado de 5,000 milisegundos definidos para la prueba y junto a ello la capacidad del procesador y memoria RAM excedan el 60% de su capacidad.

De acuerdo a estos hallazgos, se concluye como la capacidad de la arquitectura se degrada en la medida que aumenta la cantidad de usuarios concurrentes y se establece que, para este modelo y con las características definidas, la mayor cantidad de usuarios soportados, dentro de los rangos de respuesta aceptables es de 150.

Tabla XI. Resumen de la actividad por escenarios en la arquitectura cliente / servidor

| Usuarios | 50 | 100 | 150 | 250 |
|-----------------|-----------|------------|------------|------------|
| Media (ms) | 1,239 | 1,371 | 1,736 | 10,262 |
| CPU (%) | 24 | 41 | 59 | 76 |
| Ram (%) | 45 | 49 | 55 | 64 |
| Bits | 296,914 | 559,064 | 812,312 | 1,056,744 |

Fuente: elaboración propia.

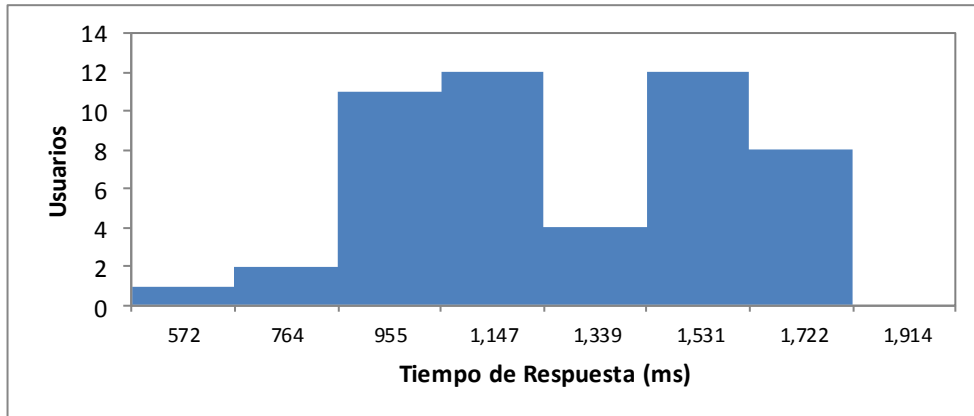
6.1.2. Evaluación del rendimiento de la arquitectura cliente / multiservidor

Para el caso de este contexto de pruebas, se tuvo la implementación de la plataforma educativa sobre una arquitectura de un servidor de balanceo de carga, dos de servicios *web* y unos más para alojamiento de base de datos, las capacidades en cuanto procesadores y memoria RAM permanecen iguales al del modelo anterior, sistema operativo Windows server 2012 R2 Estándar con 4 Gb. De memoria Ram, 2 procesadores de 2.20 GHz y disco duro de 100 Gb, en cada uno de los nodos configurados y como se verá en lo siguiente, la carga se identificó en los servidores donde se alojó la plataforma educativa, por lo que los datos obtenidos se detallan a continuación:

6.1.2.1. Cantidad de milisegundos en la carga de la aplicación

Bajo el contexto del modelo anterior, se determinaron datos de tiempo de respuesta para cada una de las series de usuarios que se detallan en las siguientes tablas y figuras, sin embargo, se llevó a cabo con una variación en la distribución de los mismos, para que fuera posible balancear las actividades en ambos servidores de aplicaciones web. La figura 27 permite ver como la cantidad de usuarios concurrentes se desempeña en un rango aceptable, en el tiempo de respuesta. La imagen muestra los datos que fueron agrupados para verlos como una sola instancia.

Figura 27. **Milisegundos para carga de 50 usuarios concurrentes en cliente / multiservidor**



Fuente: elaboración propia.

Para el logro de los resultados, de la figura anterior, la asignación de recursos de los servidores involucrados se puede ver en la tabla XII en la que se separa cada uno de ellos.

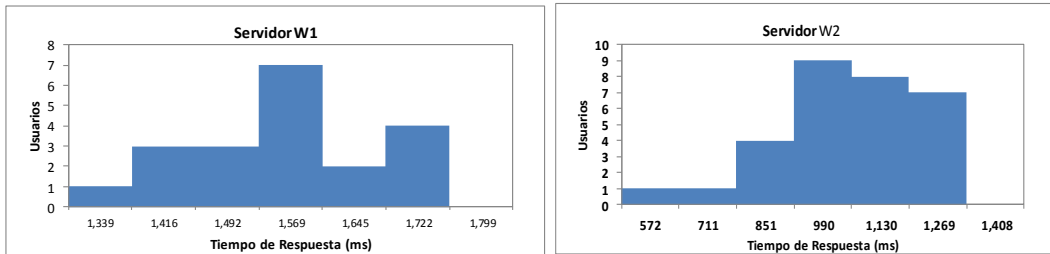
Tabla XII. **Recursos evaluados de los servidores con 50 usuarios**

| Servidor | CPU | Memoria | Red (bits) | Usuarios |
|----------|-----|---------|------------|----------|
| W1 | 13% | 23% | 138,424 | 20 |
| W2 | 11% | 24% | 191,432 | 30 |

Fuente: elaboración propia.

De los 50 usuarios concurrentes en esta evaluación, 40% estuvieron activos en el servidor w1 y 60% en el servidor w2, la frecuencia de cada uno de ellos se ve en la figura 28.

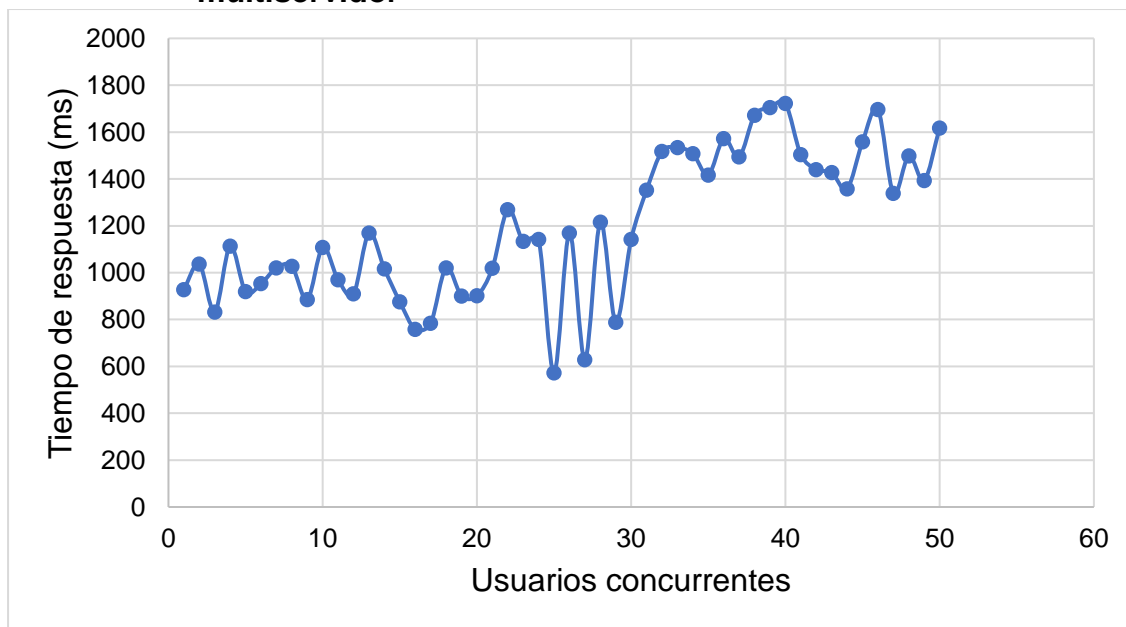
Figura 28. Milisegundos para 50 usuarios distribuidos por balanceo de carga



Fuente: elaboración propia.

Véase la figura 29 para identificar como fue el comportamiento en los tiempos de respuesta con esta cantidad de usuarios que permanecieron activos simultáneamente.

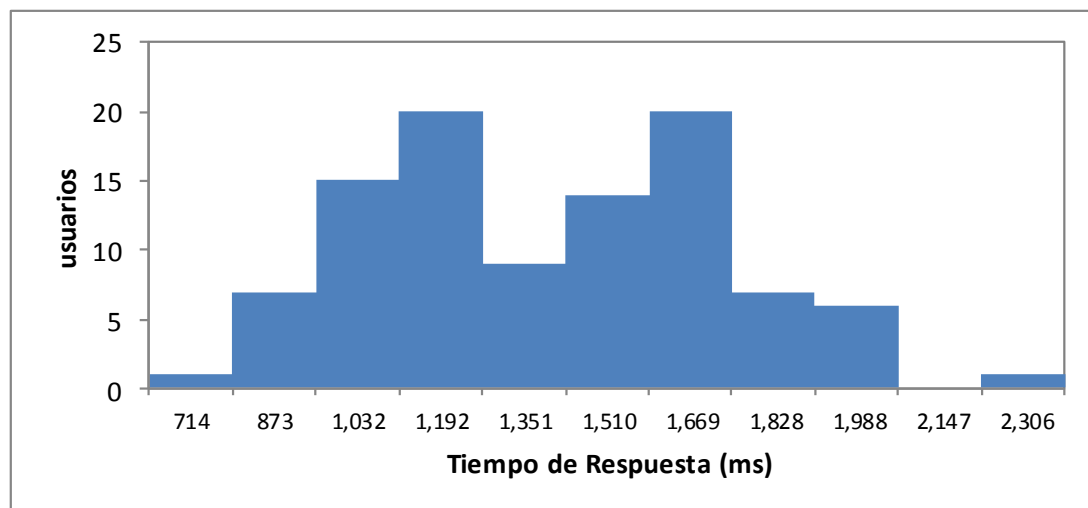
Figura 29. Despliegue de 50 usuarios concurrente en arquitectura multiservidor



Fuente: elaboración propia.

Para las pruebas realizadas con 100 usuarios se identifica un incremento, en el tiempo de respuesta, sin embargo, este no excede el límite establecido y la distribución de los usuarios es más balanceada en relación al escenario anterior, como puede verse en la figura 30, igualmente se aprecia en la figura 32 el cómo se desplegó la cantidad de usuarios durante la prueba.

Figura 30. **Milisegundos para carga de 100 usuarios concurrentes en cliente / multiservidor**



Fuente: elaboración propia.

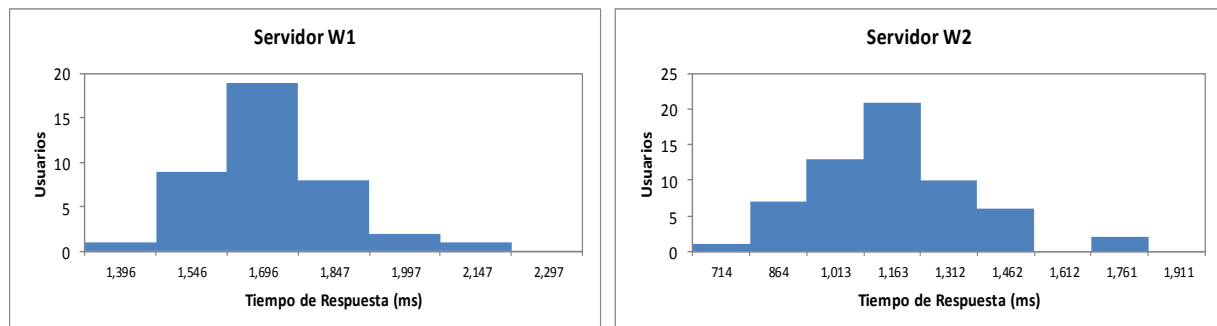
Sobre los recursos asignados la tabla XIII muestra cómo se comportó cada uno de los nodos. La carga de usuarios para cada uno de los servidores es del 40% y 60% respectivamente y su distribución por frecuencias se aprecia en la figura 31.

Tabla XIII. Recursos evaluados de los servidores con 100 usuarios

| Servidor | CPU | Memoria | Red (bits) | Usuarios |
|----------|-----|---------|------------|----------|
| W1 | 24% | 21% | 246,112 | 40 |
| W2 | 21% | 21% | 351,584 | 60 |

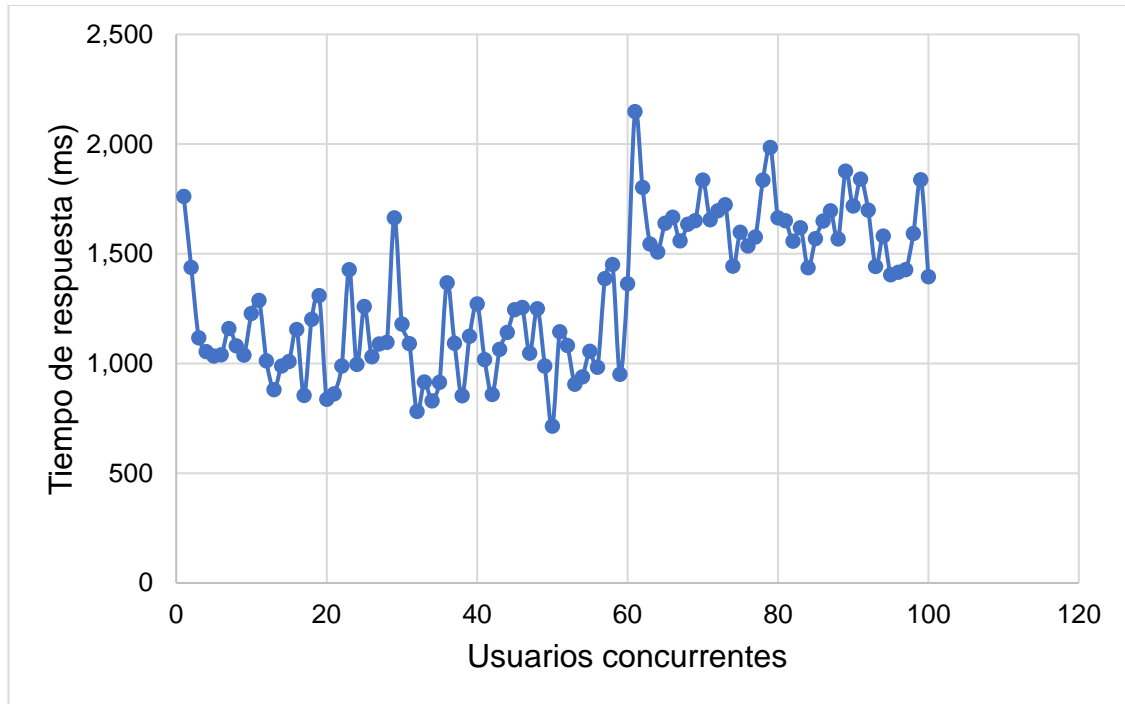
Fuente: elaboración propia.

Figura 31. Milisegundos para 100 usuarios distribuidos por balanceo de carga



Fuente: elaboración propia.

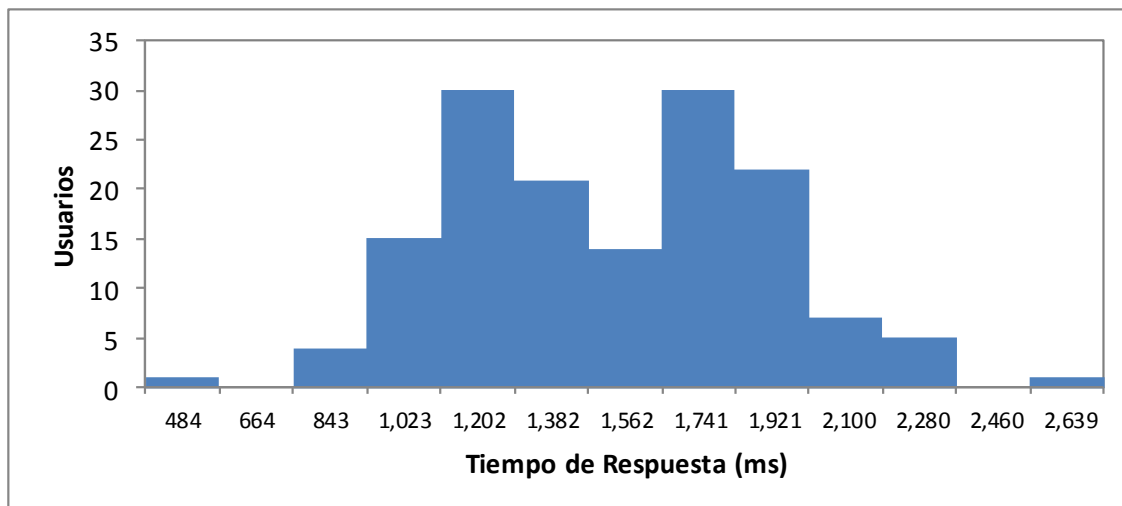
Figura 32. **Despliegue de 100 usuarios concurrente en arquitectura multiservidor**



Fuente: elaboración propia

El escenario de los 150 usuarios permanece dentro del límite establecido de 5,000 milisegundos para el tiempo de espera en la carga de la aplicación y la distribución permanece, mayormente, al centro de la gráfica, ver la figura 33.

Figura 33. **Milisegundos para carga de 150 usuarios concurrentes en cliente / multiservidor**



Fuente: elaboración propia.

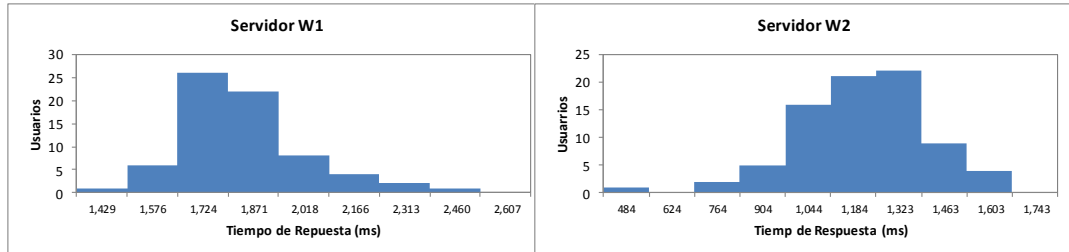
Para lo anterior, las pruebas determinaron que cada uno de los nodos tuvo una participación del 46.70% del servidor w1 y 53.30% en el servidor w2 con relación a la cantidad de usuarios concurrentes que el sistema de balanceo pudo distribuir, como se identifica en la tabla XIV y figura 34.

Tabla XIV. **Recursos evaluados de los servidores con 150 usuarios**

| Servidor | CPU | Memoria | Red (bits) | Usuarios |
|----------|-----|---------|------------|----------|
| W1 | 41% | 24% | 402,776 | 70 |
| W2 | 28% | 25% | 462,648 | 80 |

Fuente: elaboración propia.

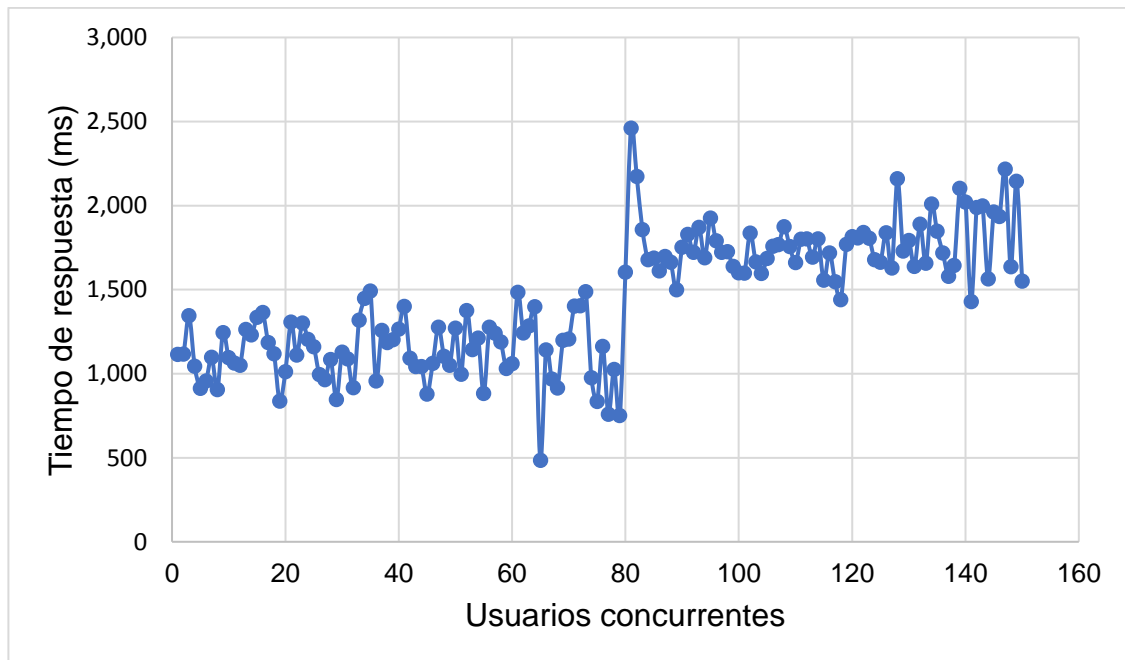
Figura 34. **Milisegundos para 150 usuarios distribuidos por balanceo de carga**



Fuente: elaboración propia.

Como se ha venido mostrando, en la figura 35 se observa la concurrencia de los usuarios dentro de la arquitectura.

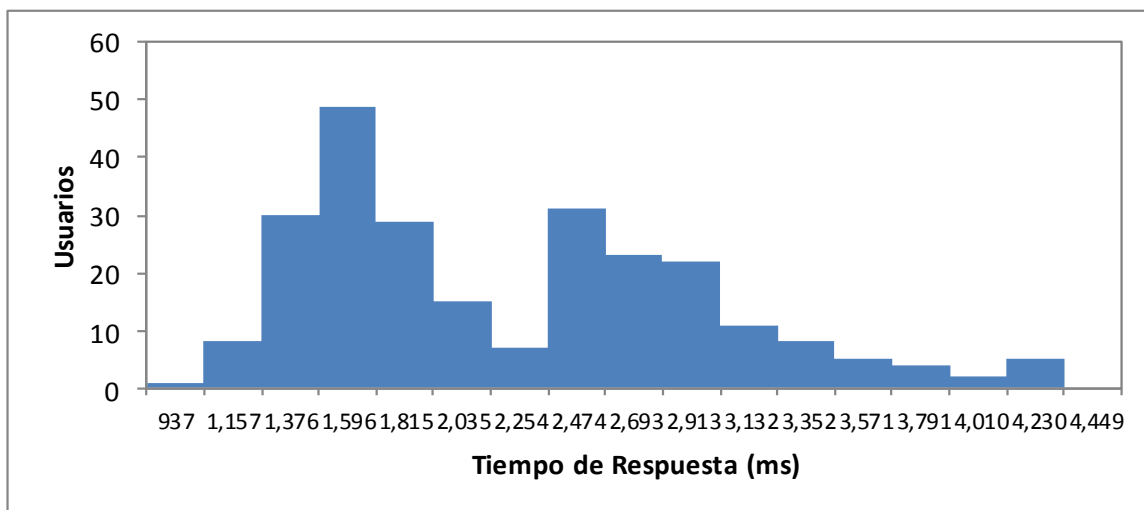
Figura 35. **Despliegue de 150 usuarios concurrente en arquitectura multiservidor**



Fuente: elaboración propia.

A diferencia de los resultados en el ambiente cliente / servidor, los datos que presenta la figura 36, demuestran como el tiempo de respuesta es menor al establecido con una carga de 250 usuarios concurrentes y con una media de 2113 milisegundos.

Figura 36. **Milisegundos para carga de 250 usuarios concurrentes en cliente / multiservidor**



Fuente: elaboración propia.

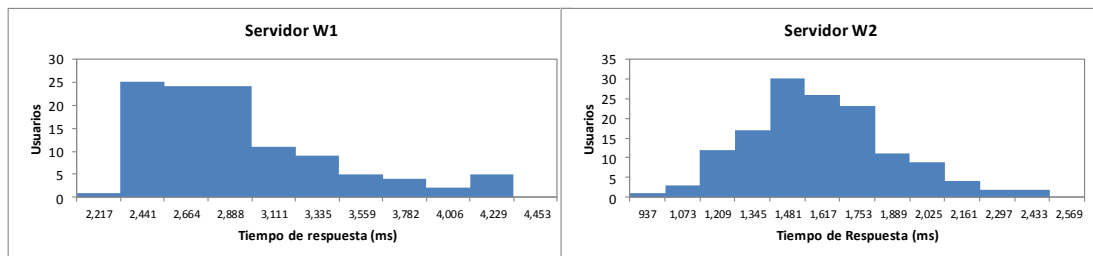
Destaca como los recursos presentan menor consumo, en cada uno de sus elementos y para este caso la distribución de usuarios es del 44% en el nodo 1 y 56% en el nodo 2, tal como se ve en la tabla XV, así mismo la distribución de los usuarios en cada uno de los servidores del clúster muestran diferencias de comportamiento, pero dentro del límite definido, como puede verse en la gráfica 19.

Tabla XV. Recursos evaluados de los servidores con 250 usuarios

| Servidor | CPU | Memoria | Red (bits) | Usuarios |
|----------|-----|---------|------------|----------|
| W1 | 60% | 28% | 602,144 | 110 |
| W2 | 48% | 31% | 771,736 | 140 |

Fuente: elaboración propia.

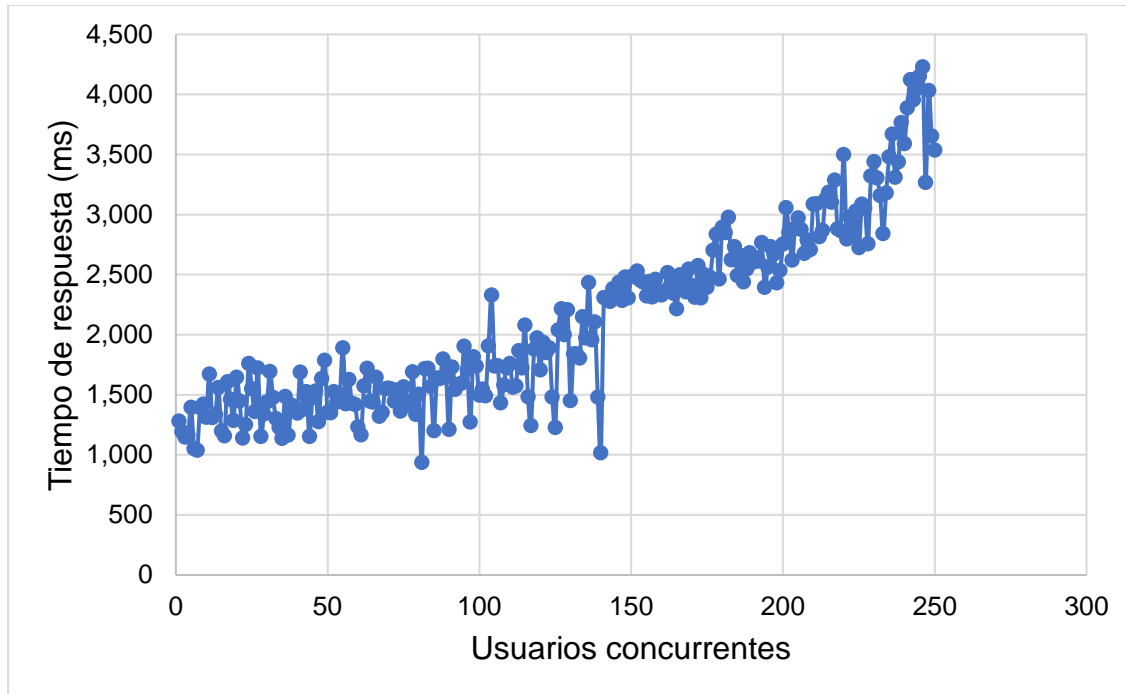
Figura 37. Milisegundos para 250 usuarios distribuidos por balanceo de carga



Fuente: elaboración propia

En la figura 38 el comportamiento, dentro de la arquitectura, queda evidenciado, en cuanto a la concurrencia de los usuarios.

Figura 38. **Despliegue de 250 usuarios concurrente en arquitectura multiservidor**



Fuente: elaboración propia.

6.1.2.2. Resumen del escenario Cliente / Multiservidor

Al igual que se expuso en el resumen del escenario cliente / servidor, las gráficas de barras conllevan la identificación de los rangos de tiempos de respuestas, con relación a los usuarios y las de dispersión, demostrar que los usuarios permanecieron activos dentro de la arquitectura.

Los resultados obtenidos en este escenario muestran como el rendimiento de los nodos, implicados en el clúster configurado, tienen un comportamiento en el que consumen menos recursos teniendo la misma carga concurrente de

usuarios, en este caso, aun con el mayor número de peticiones no se excedió el parámetro establecido de 5,000 milisegundos, en cuanto a los procesadores se distingue un uso de entre 48% y 60% y la memoria de 28% y 31%

Al tener como base los datos de esta sección se concluye que los tiempos de respuesta mejoran consistentemente por la distribución que se hace de las peticiones de usuarios en los nodos que integran el clúster de servidores, de tal manera que la cantidad de usuarios que logran interactuar de manera concurrente asciende a los 250 sin que esto afecte drásticamente el tiempo de respuesta esperado.

Los datos de la tabla XVI permiten tener a la vista el resumen de los resultados de cada una de las pruebas.

Tabla XVI. Resumen de la actividad por escenarios en la arquitectura cliente / multiservidor

| Servidor | Media (ms) | CPU | Memoria | Bits | Usuarios |
|----------|------------|-----|---------|---------|----------|
| W1 | 1,516 | 13% | 23% | 138,424 | 20 |
| W2 | 974 | 11% | 24% | 191,432 | 30 |
| | | | | | |
| Servidor | Media (ms) | CPU | Memoria | Bits | Usuarios |
| W1 | 1,642 | 24% | 21% | 246,112 | 40 |
| W2 | 1,103 | 21% | 21% | 351,584 | 60 |
| | | | | | |
| Servidor | Media (ms) | CPU | Memoria | Bits | Usuarios |
| W1 | 1,774 | 41% | 24% | 402,776 | 70 |
| W2 | 1,135 | 28% | 25% | 462,648 | 80 |
| | | | | | |
| Servidor | Media (ms) | CPU | Memoria | Bits | Usuarios |
| W1 | 2,835 | 60% | 28% | 602,144 | 110 |
| W2 | 1,546 | 48% | 31% | 771,736 | 140 |

Fuente: elaboración propia.

6.2. Rendimiento de las arquitecturas

6.2.1. Evaluación del rendimiento de la arquitectura cliente / servidor

El primero de los ejercicios se llevó a cabo en el modelo básico de la arquitectura y se ha determinado que en este ambiente el rendimiento presenta variaciones en función de la cantidad de usuarios, sin embargo, su comportamiento es aceptable hasta cierto punto, de acuerdo a la capacidad del equipo asignado. La característica de este escenario es la integración de la arquitectura con un solo servidor, al cual accedieron los usuarios para requerir los servicios de la plataforma educativa que fue implementada para la obtención de los datos.

6.2.1.1. Cantidad de usuarios concurrentes

Esta prueba fue recreada con cantidades de usuarios que permitieron determinar los datos ya expuestos en figuras y tablas anteriores, junto con ello los parámetros definidos para la experimentación son lo expresados en la tabla XVII.

Tabla XVII. **Cantidad de usuarios concurrentes para establecer rendimiento de la arquitectura cliente / servidor**

| Escenario | Usuarios | Tiempo de interacción (s) | Duración de la prueba (s) |
|------------------|-----------------|----------------------------------|----------------------------------|
| 1 | 50 | 25 | 600 |
| 2 | 100 | 25 | 600 |
| 3 | 150 | 25 | 600 |
| 4 | 250 | 25 | 600 |

Fuente: elaboración propia.

En la tabla XVII, para el caso de los segundos entre cada interacción se refiere a la cantidad de tiempo que los usuarios esperan entre la petición de una página y la siguiente que desea ver en pantalla.

6.2.1.2. Número de partidas efectuadas

El número de partidas efectuadas, para el caso del experimento, se interpreta como la división que es posible hacer de la cantidad de usuarios concurrentes que llevan a cabo la misma actividad sobre la plataforma, dentro de

las secciones que fueron creadas en el curso modelo, la tabla XVIII nos deja ver estos datos.

Tabla XVIII. **Cantidad de partidas identificadas en la actividad del curso modelo**

| Nivel | Actividad | No. Partida |
|----------------------|--|--------------------|
| 1o. Primaria | Identifica cual es el número y escribe en letras | 1 |
| | Coloca el número que sigue en la fila y completa la carga del tren. | 2 |
| | Llena los cuadros con los números que van antes y después del número que aparece en medio. | 3 |
| 2do. Primaria | Selecciona pares o impares | 4 |
| | Ordenamiento numérico | 5 |
| 3ro. Primaria | Descomposición de cifras | 6 |
| | Valor de las unidades | 7 |
| | Comparación | 8 |

Fuente: elaboración propia.

De acuerdo a lo observado en la tabla XIX, y tomando en cuenta que se tuvo procesadores de 2.20 GHz y memoria de 4 Gb. Se estima que los tiempos de respuesta, el uso de procesador y el consumo de memoria RAM, son aceptables hasta el número de 150 usuarios que interactúan dentro de la plataforma y permite que el rendimiento no se vea afectado drásticamente, sin embargo, en el último de los casos, la arquitectura se ve comprometida y debido al alto consumo de procesador y memoria, existe la posibilidad de que se generen errores que hagan inestable al equipo y se dé pauta a que los usuarios deban esperar un tiempo mayor al promedio deseado o bien se generen pérdidas de

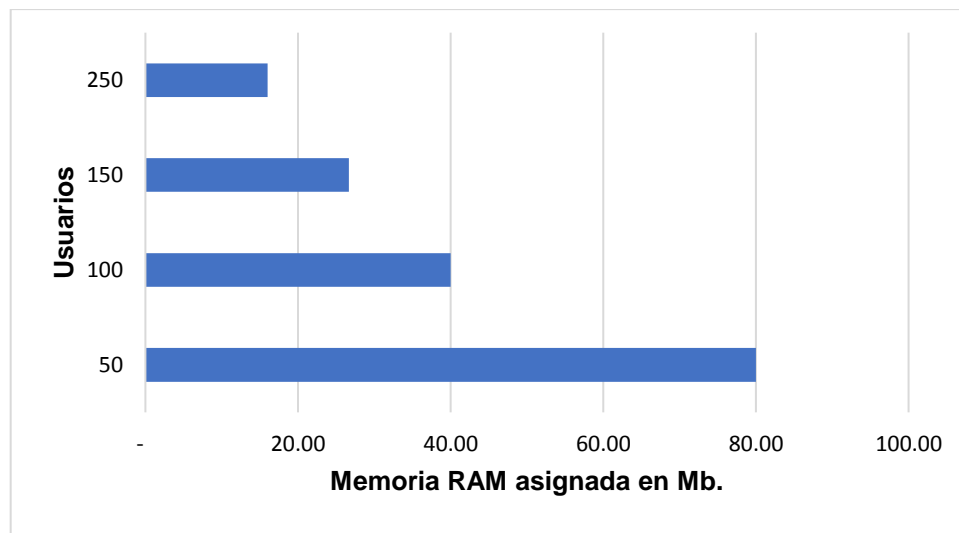
peticiones. Una imagen del comportamiento de la memoria RAM se puede ver en la figura 39.

Tabla XIX. Rendimiento de la arquitectura cliente / servidor

| Usuarios | Rango de tiempo (ms) | | Procesador | RAM | Actividad de red en bits |
|----------|----------------------|--------|------------|-----|--------------------------|
| 50 | 905 | 1,726 | 24% | 45% | 296,914 |
| 100 | 683 | 2,097 | 41% | 49% | 559,064 |
| 150 | 1,128 | 2,719 | 59% | 55% | 812,312 |
| 250 | 5,027 | 17,502 | 76% | 64% | 1,056,744 |

Fuente: elaboración propia

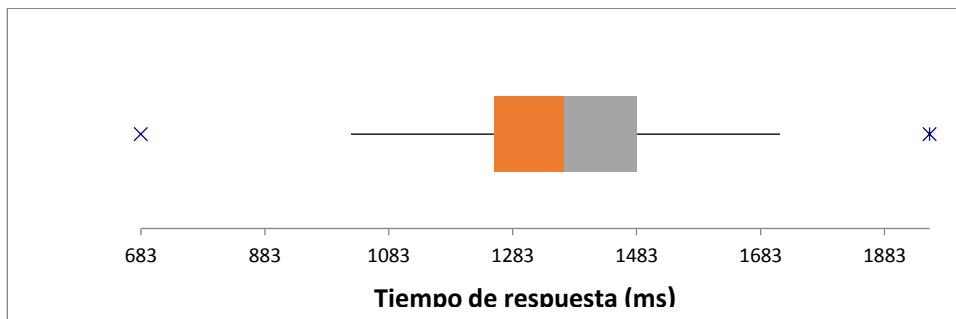
Figura 39. Asignación de memoria RAM por cantidad de usuarios



Fuente: elaboración propia.

En el mejor de los casos, se observa que la arquitectura tiene un rendimiento óptimo en el segundo de los escenarios y de manera gráfica se observa como la concurrencia de 100 usuarios muestra una distribución homogénea, de los datos obtenidos, entre los cuartiles 1ro y 3ro, tal como se aprecia en la figura 40.

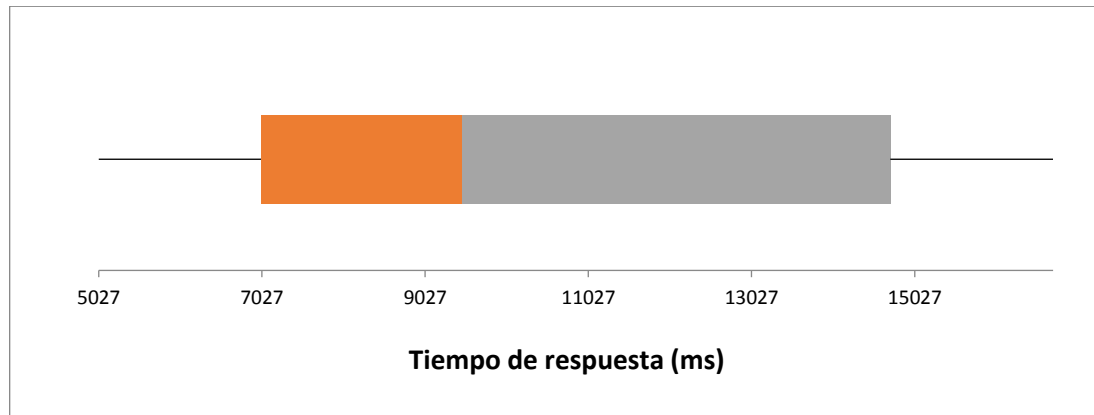
Figura 40. **Distribución de los datos en el escenario de 100 usuarios**



Fuente: elaboración propia

La limitante se observa a medida que se incrementaron las peticiones y en el último escenario, con 250 usuarios concurrentes, la distribución de los datos esta inclinada, desde la mediana hacia el límite superior, mostrándose una tendencia en el incremento del tiempo de respuesta, lo que hace que la arquitectura valla decreciendo en su capacidad y se tenga como resultado el que los usuarios deban esperar un tiempo de respuesta mayor. Ver figura 41.

Figura 41. **Distribución de los datos en el escenario de 250 usuarios**



Fuente: elaboración propia.

Los valores identificados, en cuanto a mediana corresponden a 9,476 ms, y los cuartiles 1ro. y 3ro. 7,014 ms y 14,727 ms, respectivamente.

A partir de estos resultados se distingue como el problema de escalabilidad y eficiencia se hace evidente en una arquitectura de un solo servidor, pues como lo manifestaba Vidal Rojas (s.f.) este modelo llega a representar, en ciertos niveles, un cuello de botella para los usuarios, para este caso, se identifica a partir de los 250 usuarios concurrentes.

Adicional al tema de la escalabilidad se suma el aspecto de latencia que se genera por la transmisión de muchos datos o paquetes, en un solo canal lo que se puede observar en el comportamiento del adaptador de red que fuera evaluado, en cada uno de los escenarios. Véase tabla XIX.

Una arquitectura conformada por un solo servidor, de acuerdo a esta evidencia, podría ser insuficiente para respaldar la actividad de una cantidad mayor de usuarios, específicamente en lo relacionado al uso de plataformas educativas se estima que, para lograr un mejor rendimiento, se debe incrementar las capacidades del servidor, sin embargo, es probable que por limitaciones de fabricación y temas comerciales de cada uno de los proveedores que distribuyen equipos computacionales, este llegue a un límite que finalmente no permitirá más crecimiento lo cual puede ser justificación para llevar a cabo un crecimiento horizontal.

Desde el punto de vista de la experiencia del usuario, este experimento evidencia que la variable objeto de estudio se ve afectada, principalmente, en el tiempo de respuesta que se obtiene de la arquitectura, a partir de los 250 usuarios concurrentes, y en consecuencia en el rendimiento de la arquitectura. Garret (2011) estimó en su estudio que uno de los aspectos básicos para lograr satisfacción en el usuario, es la definición de la estrategia, tomando en cuenta especificaciones funcionales, que para este caso vienen a ser limitadas para lograr el objetivo de proporcionar un modelo de arquitectura que responda a peticiones mucho mayores.

6.2.2. Evaluación del rendimiento de la arquitectura cliente / multiservidor

6.2.2.1. Cantidad de usuarios concurrentes

Las variables en cuanto a cantidad de usuarios, tiempo de duración de la prueba y la interacción entre la petición de páginas, permanecen iguales; sin embargo, se aplicó una distribución entre los servidores de manera que ambos

equipos pudieran responder a las pruebas efectuadas. La tabla XX muestra lo expuesto.

Tabla XX. **Cantidad de usuarios concurrentes para establecer rendimiento de la arquitectura cliente / multiservidor**

| Escenario | Usuarios | Servidor W1 | Servidor W2 | Tiempo entre interacción (s) | Duración de la prueba (s) |
|-----------|----------|-------------|-------------|------------------------------|---------------------------|
| 1 | 50 | 20 | 30 | 25 | 600 |
| 2 | 100 | 40 | 60 | 25 | 600 |
| 3 | 150 | 70 | 80 | 25 | 600 |
| 4 | 250 | 110 | 140 | 25 | 600 |

Fuente: elaboración propia.

6.2.2.2. Número de partidas efectuadas

De la misma manera que los usuarios se mantienen en cantidad, el número de partidas efectuadas permaneció sin cambios a pesar de la modalidad de balanceo de carga que se efectuó por lo que es posible referirse a la tabla XVIII, para observar los datos que se generaron.

Los resultados para este modelo muestran un comportamiento favorable al objetivo de contar con una arquitectura que otorgue rendimiento y escalabilidad, lo que permite que la experiencia del usuario, a través de los indicadores evaluados sea positiva, toda vez que no se exceden los límites establecidos, como puede ser el caso del tiempo de respuesta esperado en esta experimentación.

La ventaja de esta configuración resalta en la posibilidad de distribuir las peticiones de usuarios concurrentes, dentro de los nodos que componen el clúster y es así como el consumo de los recursos de los servidores disminuye casi en un 50% en cada uno de ellos, lo que permite incrementar la cantidad de usuarios que pueden participar dentro del ambiente diseñado.

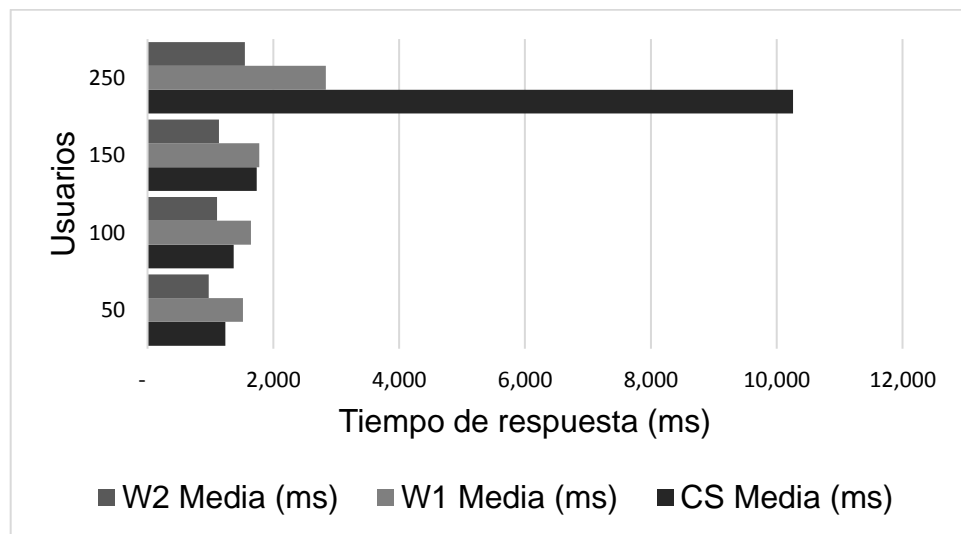
Cabe destacar que esta modalidad sigue manteniendo la configuración base de cliente / servidor, sin embargo, se adiciona un servicio de balanceo de carga que es el que ha permitido incrementar de uno a dos servidores la arquitectura, a la cual se ha llamado clúster o modelo cliente / multiservidor, de manera que confirmando lo expuesto por Diaz Henao, T.M.; Ruiz López y E.G.; Sinisterra, M.M. (2012) se ha logrado compartir la carga de trabajo y ha sido posible mantener la disponibilidad de los servicios, confirmado por el hecho de que incluso el cuarto de los escenarios de experimentación se mantuvo debajo del límite establecido de 5,000 milisegundos para el tiempo de respuesta esperado.

Es importante hacer notar, que el prototipo se diseñó para el uso de dos servidores como *back-end* de la arquitectura, sin embargo, el servicio utilizado de *Network Load Balance* de Microsoft admite hasta 32 servidores en un mismo clúster, aspecto que puede ser beneficioso en proyectos en los que se pretenda adaptar esta solución, logrando una escalabilidad robusta.

Los resultados de la sección anterior han demostrado como se ha llevado a cabo la distribución de las peticiones de los usuarios y en cada uno de los escenarios se comprueba como los recursos de servidor, tales como memoria RAM y procesador se comparten en cada uno de los nodos, la figura 42 permite

ver la comparación de los comportamientos en uno de los indicadores evaluados, tómesese en cuenta que corresponden a los ambientes cliente / servidor (cs) y el prototipo de multiservidor representado en sus nodos W1 y W2.

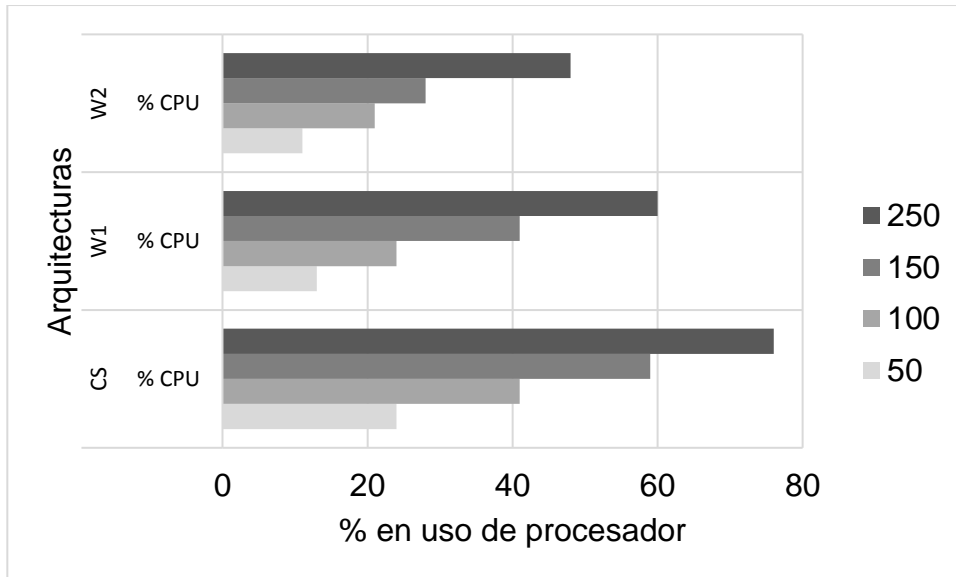
Figura 42. **Comparación del tiempo de respuesta entre las arquitecturas evaluadas**



Fuente: elaboración propia.

En cuanto a la memoria utilizada, se distingue una relación directamente proporcional, entre la cantidad de usuarios y lo disponible de este recurso, sobre todo es notable como la carga del escenario de los 250 usuarios concurrentes afecta el modelo cliente / servidor, en tanto que, en el prototipo diseñado, el consumo no excede el 35% de la memoria RAM disponible, lo que se puede verificar en la figura 43.

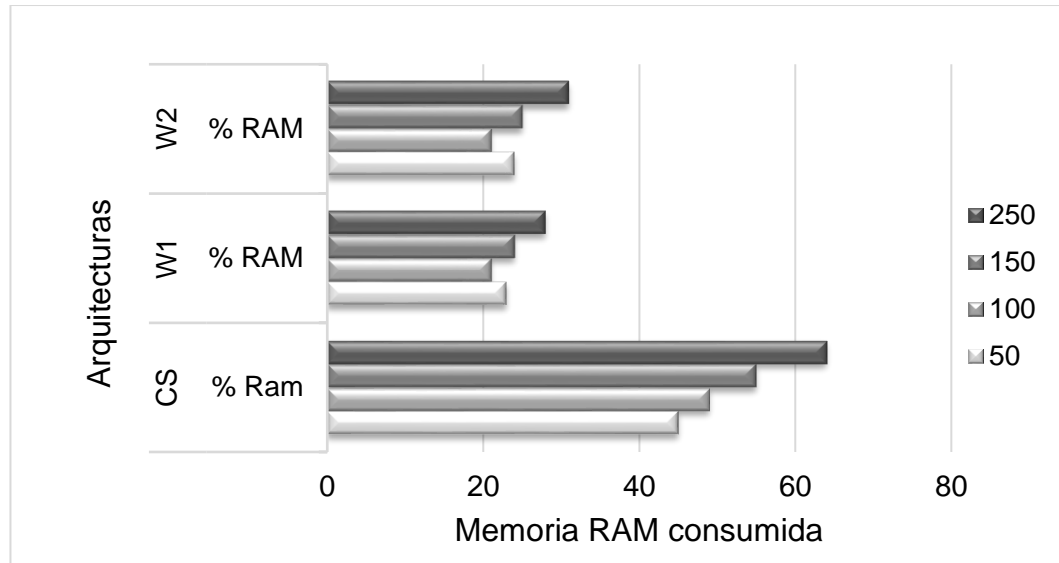
Figura 43. **Comparación del consumo de memoria RAM en las arquitecturas evaluadas**



Fuente: elaboración propia.

Sucede algo similar con el comportamiento de los procesadores, involucrados en las evaluaciones, toda vez se tiene una carga mayor de trabajo, se necesitará una respuesta que cubra las expectativas de los usuarios y ésta es distinta en cada modelo, según sea planificada la implementación de las infraestructuras necesarias, (González Mendez, 2011), para dar soporte a requerimientos mayores, para el caso que se discute, la figura 44 demuestra que existen diferencias en el tratamiento del procesamiento, al menos en el porcentaje que es utilizado de acuerdo a los escenarios de usuarios que fueron definidos para las pruebas.

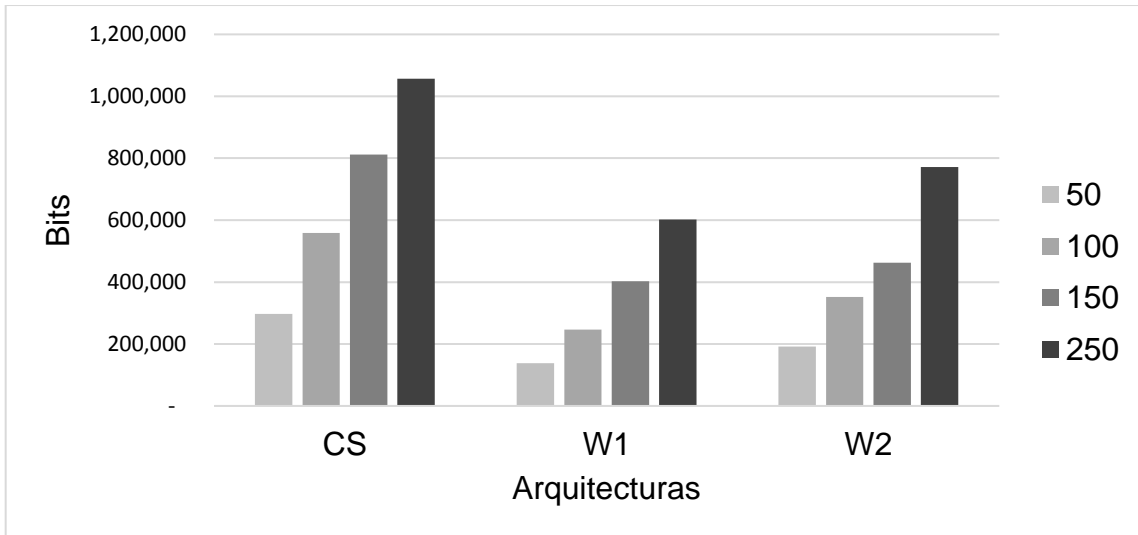
Figura 44. **Comportamiento de los procesadores en las arquitecturas evaluadas**



Fuente: elaboración propia.

Otro de los aspectos que se identificaron como puntos que aportan a la comprensión del porque el tiempo de respuesta se ve incrementado a medida que aumentan las peticiones de usuarios, es lo relacionado al adaptador de red y la transmisión de bits que esto conlleva, el hecho de particionar en dos o más adaptadores el tráfico de red, necesariamente causa una disminución en el caudal de paquetes que viajan a través de este dispositivo, por lo contrario la concentración en un solo servidor da como resultado el origen a latencias que perjudican, en última instancia, el tiempo de respuesta. Vea la figura 45 para mejor comprensión.

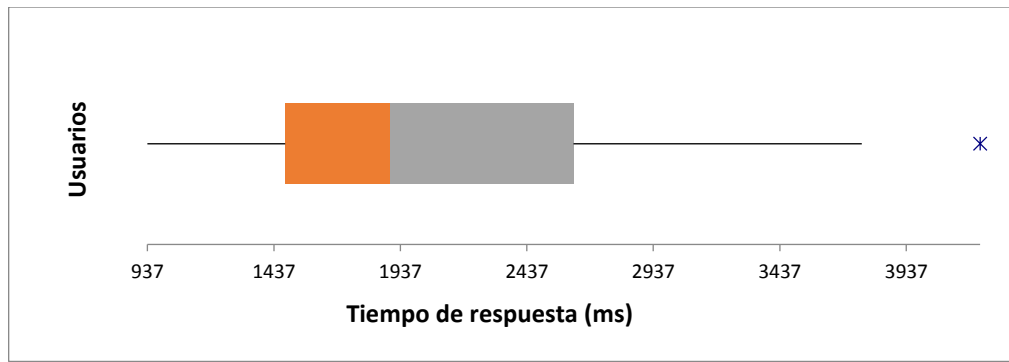
Figura 45. **Comportamiento en los adaptadores de red en las arquitecturas evaluadas.**



Fuente: elaboración propia.

En el prototipo, se evidencia que la arquitectura ofrece mejoras en su rendimiento como consecuencia de lo que ha sido previamente evaluado y para el caso de la concentración de usuarios se puede observar en las siguientes figuras como han sido distribuidos, contrastando con lo que se visualiza en la figura 46, del ambiente cliente / multiservidor.

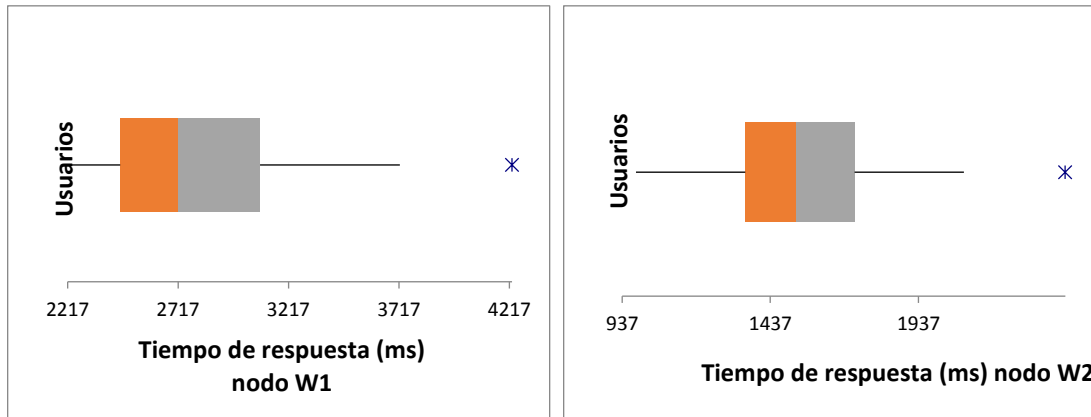
Figura 46. **Distribución de los datos de 250 usuario en arquitectura cliente / Multiservidor**



Fuente: elaboración propia.

La gráfica confirma un tiempo de respuesta mejorado, con relación al primer modelo de arquitectura evaluado, y permite confirmar que la experiencia del usuario puede ser satisfactoria, desde luego que existe concentración de usuarios en tiempos de respuesta aceptables. Lo anterior fue posible gracias al método de distribución de peticiones, en cada uno de los nodos que integran el nodo y de los cuales se puede ver su comportamiento en la figura 47.

Figura 47. **Distribución de 250 usuarios en nodos de la arquitectura cliente / multiservidor**



Fuente: elaboración propia.

6.3. Proyección de usuarios concurrentes en la implementación

La propuesta de implementación consideró tomar como base el prototipo utilizado en las pruebas, en cuanto a sus componentes los requerimientos para cada servidor se detallan en la tabla XXI y partiendo de esa configuración se determinó la posibilidad en dar cobertura a más de 26,000 usuarios dentro de esta arquitectura.

Tabla XXI. **Requerimientos para servidores de propuesta de implementación**

| Procesador | Núcleos | Memoria RAM | Ethernet |
|--|---------|--------------|----------|
| procesador Intel Xeon E5-2600 V4 a 2.4 MHz | 4 -18 | 16 – 512 Gb. | Gigabite |

Fuente: elaboración propia.

Es importante tener presente que la proyección se llevó a cabo tomando en cuenta que el consumo óptimo de memoria *RAM* por usuario, dentro de las pruebas, fue de 40 Mb y que los equipos seleccionados tendrían características propias de un servidor, como se hizo ver en espacios anteriores, el servicio de *Network Load Balance* (NLB) del sistema operativo *Windows* tiene capacidad, para controlar hasta 32 nodos, aspecto que potencializa las posibilidades de dar mayor cobertura a miles de usuarios.

En términos de proyección, la cantidad de usuarios a cubrir con el menor de los escenarios, definido en la tabla XXII, sería de 13,107 mientras que en el escenario de mayor alcance ascendería a los 419,430. Lo anterior resulta de combinar las capacidades de los nodos, en cuanto a memoria *RAM* y lo que esto representa en cantidad de usuarios y el número máximo de nodos que es posible configurar con el servicio NLB, del sistema operativo *Windows Server*.

Tabla XXII. Proyección de cobertura de usuarios

| Escenario | RAM (Gb.) | 16 | 32 | 64 | 128 | 256 | 512 |
|------------------|------------------|-----------|-----------|-----------|------------|------------|------------|
| 01 nodos | Usuarios | 410 | 819 | 1,638 | 3,277 | 6,554 | 13,107 |
| 02 nodos | Usuarios | 819 | 1,638 | 3,277 | 6,554 | 13,107 | 26,214 |
| 32 nodos | Usuarios | 13,107 | 26,214 | 52,429 | 104,858 | 209,715 | 419,430 |

Fuente: elaboración propia.

6.4. Costos incurridos

Los datos que se lograron determinar, para la ejecución del estudio efectuado, corresponden al 71 % de lo que fue planificado, destaca el hecho de haber conseguido llevar a cabo la investigación sin la participación de programador y diseñador gráfico, debido a que no fue necesario desarrollar una interfase específica para llevar a cabo las pruebas; por otra parte, el tiempo estimado de 360 horas de trabajo, se redujo a 275, lo que en conjunto representó un ahorro, en recursos, del 29 %.

6.5. Disminución de costos de implementación

En el aspecto de proyección de costos para la implementación de un proyecto de esta naturaleza, los datos contenidos en las tablas IV y V demuestran cómo es posible obtener ventajas económicas, que representan un ahorro del 29.82 % en un tiempo no mayor a 5 años, como se puede observar en la tabla XXIII.

La relación, en esta proyección, es de 1 a 1, es decir, se ha comparado el costo de implementar la arquitectura propuesta de forma local con una similar en infraestructura de nube.

Tabla XXIII. Proyección de ahorro en la implementación de servidor en la nube

| Descripción | Cantidad | Valores |
|---|-----------------|----------------|
| Costo de implementación servidor local | 4 | \$114,615.00 |
| Costo de implementación de servidor en nube | 4 | \$80,433.60 |
| Proyección de ahorro a 5 años | | \$33,225.61 |
| Relación de la proyección de ahorro | | 29.82% |

Fuente: elaboración propia.

Se parte del supuesto que el proyecto de implementación lo podría llevar a cabo un pequeño o mediano empresario o grupo de interesados que deseen invertir, con financiamiento bancario o bien, a través del patrocinio de financistas, para la comprensión de las tablas IV y V se tomaron en cuenta los siguientes elementos:

6.5.1. Implementación local

Los componentes del prototipo utilizado han servido de base para realizar la propuesta de implementación local, es decir, los 4 servidores fueron proyectados económicamente con un valor actual de USD \$. 9,210.00 y se ha considerado una tasa de interés activa en dólares del 6 %, según datos del Banco de Guatemala, para efectos de proyectar un financiamiento a 5 años plazo, de lo cual se tiene como resultado el pago mensual de USD \$.629.17 en concepto de amortización y una anualidad de USD \$.7,550.09

De igual forma se ha procedido a calcular la compra del software involucrado, que consiste en el sistema operativo y sistema de gestión de base de datos.

Por ser un elemento de un impacto considerable se tomó en cuenta el consumo eléctrico que se generaría, tanto por los 4 servidores involucrados como por una unidad de aire acondicionado, como dos de los aspectos más representativos, de estos se ha podido establecer que el consumo en *Kilowatts*, para el caso de los servidores, es de 64 por cada uno de ellos, mientras que para el sistema de enfriamiento sería de 82, el costo del *Kilowatt*/hora, de acuerdo a datos de la Empresa Eléctrica de Guatemala, asciende a USD \$.0.20, lo cual incluye gastos de transporte, cargos fijos y municipales, con lo que se tendría que erogar mensualmente USD \$.486.00 y anualmente USD \$.5,835.61.

En cuanto a los costos de mantenimiento, se estimó el 15 % sobre el valor actual de los servidores y para los servicios de conectividad una cuota mensual de USD \$.600.00 por un servicio de enlace Clear Channel de 10 Mbps, este último servicio fue cotizado con dos proveedores, Optel y TIGO.

6.5.2. Implementación en nube

Como se puede observar en la tabla V, el único desembolso que se llevaría a cabo es el pago mensual de USD \$.1,341.00 por la misma estructura tecnológica, sin embargo, con algunas ventajas pues sería posible contar con un mejor ancho de banda para la conectividad y la posibilidad de partir de una capacidad inicial de cobertura de 26,214 usuarios y llegar al total de 419,424 de manera gradual o bien, de acuerdo a la demanda reducir el consumo de recursos

si así es necesario, aspecto que en el modelo local no es posible, después de comprados los recursos para incrementar capacidad, estos permanecen en la arquitectura.

Como se puede observar, la implementación de infraestructuras como servicios en la nube representa ventajas competitivas a nivel de los costos, donde se debe incurrir, además de contar con la posibilidad de incrementar o disminuir la capacidad de cobertura a nivel de los requerimientos técnicos que viene a incidir en la menor o mayor cantidad de usuarios que se quieran soportar, para este caso evaluado, sería posible ascender a un total de 419,430.

CONCLUSIONES

1. Se llevó a cabo la implementación del prototipo de arquitectura multiservidor, en ambiente EC2, teniendo como base el modelo de los juegos masivos multijugador en línea, apoyándose en un sistema de balanceo de carga, se logró determinar diferencias de rendimiento, frente al modelo tradicional cliente / servidor. Los tiempos de respuesta en el cliente ascienden a 2,835 milisegundos, lo cual implica una reducción de 7,427 milisegundos comparado con el modelo tradicional, en cuanto al incremento gradual de la carga transaccional, el prototipo se apoya en la distribución de las peticiones en cada uno de los nodos a través del sistema NLB de *Windows server*.
2. La estructura mínima aplicada en este ejercicio consistió en la integración de servidores con funciones específicas, uno de ellos otorgando el servicio de balanceo de carga, dos con servicios de publicación *web* y uno más como repositorio de la base de datos, este esquema representa la posibilidad de dar cobertura al menos a 26,214 usuarios y si se toma en cuenta las proyecciones de las capacidades mayores el número ascendería a 419,430; en lo relacionado con la implementación, se hizo posible determinar que la propuesta garantiza una reducción de los costos, en por lo menos un 29 % haciendo la comparación de las erogaciones que representan tanto la implementación local como en ambiente de nube.
3. El desarrollo del *software* se logró llevando a cabo una integración de tecnologías propietarias como de código abierto, en tal sentido para lograr

el funcionamiento de la aplicación educativa enfocada en matemáticas para primaria inicial, se utilizó *Internet Information Server* 8.5 para la configuración, administración y publicación del sitio *web*, otra tecnología fue PHP 5.6.30 como gestor del software de la plataforma educativa Moodle 3.2.2 y finalmente sobre esta la inserción de cursos interactivos, de naturaleza libre, de Anaya. Todo lo anterior fue configurado sobre el sistema operativo *Windows server* 2012 Estándar R2 y para el caso de la gestión de base de datos *SQL server* 2008 R2.

4. Se hizo posible la implementación de un sistema de balanceo de carga a través del sistema operativo *Windows server* estándar 2012 R2 y la funcionalidad NLB, este hizo factible la distribución de las peticiones de usuario, dentro del clúster configurado, lo cual permitió demostrar el procesamiento horizontal, dentro de la arquitectura propuesta, de tal cuenta que los resultados muestran la reducción del retraso en el tiempo de respuesta en las peticiones de los clientes, así como la disminución de la carga transaccional en cada uno de los nodos y se pudo determinar que el sistema de balanceo tiene una capacidad máxima de controlar 32 nodos.

5. En función de los resultados presentados en la sección correspondiente, fue posible determinar la capacidad de la arquitectura prototipo, se obtuvo datos para cada uno de los aspectos especificados como indicadores de la variable experiencia del usuario, siendo estos: cantidad de milisegundos en la carga de la aplicación *web* configurada, cantidad de usuarios concurrentes y número de partidas efectuadas, para tal efecto se hizo uso de las herramientas *Webserver stress tool*, una herramienta *freeware* con la cual fue posible obtener datos sobre la cantidad máxima de usuarios que soportaron las plataformas, así mismo Monitor de rendimiento,

herramienta nativa del sistema Windows server, con ella fue posible determinar tiempo de respuesta de los diferentes servidores, uso de memoria RAM, comportamiento de los discos duros y procesadores.

RECOMENDACIONES

1. Debido a que la interacción de los usuarios no siempre será dentro del mismo servidor, por motivos de distribución de peticiones a cargo del sistema de balanceo de carga, es recomendable indagar sobre aplicaciones que sean capaces de sincronizar altos volúmenes de datos que sean generados por las transacciones realizadas dentro del clúster, lo cual debería aplicar tanto a los servidores a cargo de publicación de la interface *web* como de los repositorios de bases de datos, si fuera necesario.
2. Dentro del prototipo no se consideraron aspectos de seguridad perimetral, por lo que la ampliación en este aspecto resultará de interés en futuras investigaciones, sobre todo determinar que esto no afecte los tiempos de respuesta en las peticiones de los clientes.
3. Se estima conveniente la investigación acerca de *software* de plataformas educativas que permitirá una interacción del usuario con funcionalidades más avanzadas en relación a juegos por roles.
4. La implementación de algoritmos que permitan determinar el avance y logros obtenidos de los estudiantes, y que como resultado de esto pueda ofrecer reconocimientos virtuales tal el caso de la gamificación.

REFERENCIAS BIBLIOGRÁFICAS

1. Aguilar Juárez, I. y Ayala De La Vega, J. (2014). Las propiedades técnicas deseables en las plataformas educativas y herramientas de autor como productoras de contenido estandarizado. *Revista Iberoamericana para la Investigación y el Desarrollo Educativo. Universidad Autónoma De Estado De México.*
2. Alier Forment, M. (2014). Aplicaciones Educativas: Ipac. Universidad Politécnica de Catalunya (UPC), Facultad de Informática de Barcelona.
3. Álvarez García, F. (s.f.). AGRA: Sistema de distribución de Objetos para un sistema distribuido orientado a objetos soportados por una máquina abstracta. (Tesis Doctoral). Universidad de Oviedo.
4. Barri Vilardell, I. (2012). Modelado y planificación de aplicaciones de juegos masivos multijugador en red en entornos distribuidos. (Tesis Doctoral). Universitat de Lleida.
5. Bocchio, F. (2014). Modelo Cloud computing como alternativa para escalabilidad y recuperación de desastres. Escuela de Postgrado, Universidad Tecnológica Nacional. Buenos Aires.

6. Caserotto, G.A. (2012). Computación ubicua, sensibilidad al contexto y mashups. Universidad Nacional de La Plata.
7. Celorio Aguilera, C. (2010). Una arquitectura distribuida basada en agentes software para el desarrollo y el soporte de espacios de aprendizaje ubicuos. (Tesis Doctoral). Universidad Nacional de Educación a Distancia. Madrid, España.
8. Clarenc, C. A.; S. M. Castro, C. López de Lenz, M. E. Moreno y N. B. Tosco (Diciembre, 2013). Analizamos 19 plataformas de e-Learning: Investigación colaborativa sobre LMS. Grupo GEIPITE, Congreso Virtual Mundial de e-Learning.
9. Córdoba-Cely, C. (2013). La Experiencia del usuario: de la utilidad al afecto. *Revista Iconofacto Vol. 9, No. 12 pp. 56 – 70 Medellín – Colombia. Enero – Junio 2013.*
10. Díaz Henao, T.M.; Ruiz López, E.G.; Sinisterra, M.M. (2012). Clúster de balanceo de carga y alta disponibilidad para servicios *web* y *mail*. *Informador Técnico (Colombia) Edición 76, Enero - Diciembre 2012, p 93 – 102.*
11. Díaz Pineda, J. A. (2007). Análisis de la arquitectura de CDN aplicada a MMOG. Universitat Politecnica de Valencia.

12. Fernández Manjón, B. (2006). Especificaciones y estándares en *e-learning*. *Revista de Tecnologías de la información y las comunicaciones educativas*.
13. Fernández-Pampillón Cesteros, A. (2010). *Las plataformas e-learning para la enseñanza y el aprendizaje universitario en Internet*.
14. Garret, J.J. (2011). *The elements of user experience*. Berkeley, CA: New Riders. p 24.
15. González González, C.S.; Blanco Izquierdo, F. (2011). Videojuegos educativos sociales en el aula. *REVISTA ICONO 14, 2011, Año 9 Vol. 2, pp. 59-83. Madrid, España*.
16. González Méndez, M. P. (2011). Propuesta de un modelo de implementación para un sistema de monitoreo de infraestructura de IT. Universidad Simón Boliva. Venezuela.
17. Lafuente, A. (2011). *Introducción a los Sistemas Distribuidos*. Departamento de Arquitectura y Tecnología de Computadores, Universidad del País Vasco.
18. Mesa Múnera, A. (2009). Método para el manejo del balanceo de carga en sistemas de cómputo distribuido de alto desempeño. Universidad Nacional de Colombia, sede Medellín.

19. Méndez Poza, G. (2008). Una arquitectura software basada en agentes y recomendaciones metodológicas para el desarrollo de entornos virtuales de entrenamiento con tutoría inteligente. (Tesis Doctoral). Universidad Politécnica de Madrid. España.
20. Moir Sandoval, S. A. K. (2011). Criterios para desarrollar cursos de postgrado en línea. Caso: Escuela de Estudios de Postgrado, Facultad de Ciencias Económicas, Universidad de San Carlos de Guatemala.
21. Nolasco Argueta, J. A. (2012). Uso de recursos multimedia para potenciar el aprendizaje de los estudiantes del noveno grado en la asignatura de electricidad en el Centro de Investigación e Innovación Educativas de la Universidad Pedagógica Nacional Francisco Morazán.
22. Ortiz, O.; Alcover P.; Sánchez, F.; Pastor, J. A. Pastor y Herrero. R. (2014) Herramienta *m-learning* para el Aprendizaje de Programación Estructurada en los Primeros Cursos de Ingeniería. VAEP-RITA Vol. 2, Núm. 2, Jun. 2014 pp 65-71.
23. Sapiña Rubio, A. (2012). Alta disponibilidad en servidores y optimización de recursos *hardware* a bajo coste. Universidad Politécnica de Valencia.

24. Salgado García, E. (2015). La enseñanza y el aprendizaje en modalidad virtual desde la experiencia de estudiantes y profesores de postgrado de la Universidad Latinoamericana de Ciencia y Tecnología.
25. Silva-López, R.B.; Méndez-Gurrola, I.I.; Herrera Alcántara, O. (2015) Metamodelado de aprendizaje estratégico (MAE): Arquitectura de la capa de infraestructura, solución basada en *cloud computing*. *Research in Computing Science* 93 (2015) pp. 175 – 188. Universidad Autónoma Metropolitana, Distrito Federal, Mexico.
26. Suclla Ardiles, E. (2015). Diseño e implementación de un sistema de gestión de aprendizaje multimedia basado en software libre para el Instituto de Educación Superior Tecnológico Público Naval – CITEN. Pontificia Universidad Católica del Perú.
27. Vidal Rojas, J. C. (s.f.). Nuevo modelo de objetos acoplados para interfaces multiusuario. Universidad de Chile
28. Villar Monserrat. A. (2011). Soluciones distribuidas para una aplicación Cliente/Servidor de simulación. Universitat Politecnica de Valencia.