Pontificia Universidad Católica del Ecuador
Seréis mis testigos
ESMERALDAS

Ingeniería en Tecnologías de la Información

Título del proyecto: Propuesta de implementación de MOODLE de alta demanda balanceada y escalable

Previo al grado académico de Ingeniero en Tecnología de la Información

Línea de investigación: Estudio, Diseño e Implementación de Software

Autor: Anthony Aveiga Valencia

Asesor: Mgt. Marc Grob

Mayo, 2022

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

RESUMEN

MOODLE (Modular Object Oriented Dynamic Learning Environment) es una plataforma de código abierto para desarrollo de cursos virtuales ampliamente utilizado alrededor del mundo. Las instituciones educativas que cuentan con gran número de usuarios concurrentes suelen enfrentar retos relacionados al rendimiento y disponibilidad en este sistema. Las plataformas en la nube ofrecen servicios que permiten crear recursos y diseñar arquitecturas que aprovechen las capacidades propias de la computación en la nube. Entre estos beneficios se encuentra la resistencia a fallos, alta demanda, escalabilidad y reducción de costos.

En la presente investigación se identificaron las arquitecturas posibles para implementar MOODLE, las cuales se evaluaron mediante un instrumento ISO/IEC 25010 en base a parámetros de disponibilidad, tolerancia a fallos y capacidad de recuperación. La arquitectura con la calificación más alta en estos parámetros resultó ser la basada en servicios de la nube.

En base a estos resultados y a los requerimientos en la PUCESE, se procedió a proponer y diseñar una arquitectura basada en la nube de Azure, la cual se implementó usando servicios gestionados por el proveedor y de tipo serverless. Se realizaron pruebas de carga para comprobar la resistencia de la arquitectura a 100, 300 y 10000 usuarios concurrentes.

Los resultados de las pruebas de carga demostraron que la arquitectura es resistente a por lo menos 300 usuarios concurrentes, por lo que se cumple eficientemente las necesidades en la PUCESE, la cual tiene un máximo de 150 a 200 usuarios concurrentes en épocas de alta demanda.

ABSTRACT

MOODLE (Modular Object Oriented Dynamic Learning Environment) is an open source platform for the development of virtual courses widely used around the world. Educational institutions with a large number of concurrent users often face challenges related to performance and availability in this system. Cloud platforms offer services that enable the creation of resources and design architectures that take advantage of cloud computing capabilities. Among these benefits are resilience to failures, high demand, scalability and cost reduction.

In this research, possible architectures for implementing MOODLE were identified and evaluated using an ISO/IEC 25010 instrument based on availability, fault tolerance and resilience parameters. The architecture with the highest rating in these parameters turned out to be the one based on cloud services.

Based on these results and the requirements at PUCESE, we proceeded to propose and design an Azure cloud-based architecture, which was implemented using vendor-managed and serverless services. Load tests were performed to test the resilience of the architecture to 100, 300 and 10000 concurrent users.

The results of the load tests showed that the architecture is resilient to at least 300 concurrent users, thus efficiently meeting the needs at PUCESE, which has a maximum of 150 to 200 concurrent users at times of high demand.

1. INTRODUCCIÓN

1.1. DESCRIPCIÓN DEL PROBLEMA

La Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) estimó en 2019 que más de 12 millones de personas participaron en algún tipo de aprendizaje en línea. En el año 2020, las instituciones educativas implementaron rápidamente herramientas de aprendizaje en línea en respuesta a las medidas de distancia social adoptadas en todo el mundo durante la pandemia del COVID-19 [1].

Una de las herramientas más populares en el sector educativo es MOODLE, con 253 millones de usuarios de 241 países. Esta plataforma consiste en un sistema de código abierto orientado a la educación que ofrece un medio de aprendizaje virtual para profesores, estudiantes y administradores [2].

Este sistema se puede diseñar para ser escalable y adaptarse a comunidades de usuarios extremadamente grandes [3]. La disponibilidad y la escalabilidad son aspectos críticos que se deben considerar al implementar un sistema de MOODLE [4]. Las plataformas de computación en la nube ofrecen servicios como Kubernetes, que permiten desplegar este tipo de sistemas de una forma balanceada y escalable al distribuir la carga en diferentes servidores dentro de un clúster.

Este sistema debe ser lo suficientemente robusto para servir a miles de estudiantes, administradores y profesores de manera simultánea en grandes centros de estudio. Debido a la gran carga de trabajo que se puede producir en este escenario, es primordial implementar y distribuir los componentes de plataforma como el servicio web, la base de datos y el sistema de archivos en múltiples servidores para lograr alta disponibilidad y escalabilidad [3].

Los servicios en la nube permiten implementar sistemas distribuidos de alta demanda, mediante el uso de múltiples servidores y balanceadores de carga [5]. Sin embargo, estos servicios pueden producir altos costes económicos si no se diseña una arquitectura eficiente que utilice los recursos del proveedor de la nube de manera equilibrada [6].

En un escenario en donde este sistema cuente con varios miles de usuarios, es primordial garantizar que este sistema se ejecute en una arquitectura robusta que garantice la

disponibilidad del servicio, soporte la alta carga de trabajo y que permita el escalado de los recursos.

1.2. JUSTIFICACIÓN

La importancia del presente trabajo de investigación radica en que solucionará las necesidades de grandes centros de estudios, los cuales cuentan con un gran número de usuarios y el rendimiento del sitio web es afectado debido a la alta demanda del servicio.

Esta investigación será de gran utilidad ya que buscará una solución a esta problemática al diseñar una infraestructura en la nube que garantice la disponibilidad del servicio y por lo tanto mejore la experticia de uso de la plataforma MOODLE para docentes y estudiantes.

Por lo tanto, los resultados de esta investigación beneficiarán a las instituciones educativas de gran tamaño ya que se solucionarán problemas de rendimiento en el servicio web y se evitarán pérdidas de disponibilidad del sistema. Esto mejorará el proceso de aprendizaje para estudiantes y docentes el cual es afectado por estos inconvenientes.

Además, los resultados de esta investigación significaron un beneficio para los administradores de Tecnologías de la Información (TI) ya que podrán implementar una arquitectura de MOODLE que sea estable, robusta y escalable. Por lo tanto, esta investigación servirá de base para que los administradores de TI obtengan un referente científico y técnico que les facilite la implementación de este sistema en instituciones con un gran número de usuarios.

1.3. OBJETIVOS

1.3.1. Objetivo General

Proponer una arquitectura para un sistema de MOODLE mediante el uso de servidores distribuidos en la nube para obtener escalabilidad, redundancia de datos y alta disponibilidad del servicio en la PUCE Sede Esmeraldas.

1.3.2. Objetivos Específicos

1. Identificar las arquitecturas disponibles para implementar una instancia de MOODLE.
2. Comparar las diferentes formas de implementación de MOODLE mediante un instrumento de evaluación ISO.
3. Conocer los requerimientos y la infraestructura existente en la PUCE Sede Esmeraldas.
4. Elaborar una propuesta de implementación en base a los resultados obtenidos.

1.4. MARCO TEÓRICO

1.4.1. Escalabilidad

La disponibilidad y la escalabilidad son aspectos claves que se deben tener en cuenta en un sistema MOODLE que recibe un gran número de peticiones y carga de trabajo. Generalmente, la mayoría de las organizaciones alojan esta plataforma en un solo servidor, que incluye la aplicación, base de datos y sistema de archivos. Si la carga que recibe este único servidor aumenta debido al incremento de usuarios y peticiones al servicio, eventualmente será necesario aumentar los recursos del servidor para que soporte esta carga. En este sentido, surge el concepto de escalabilidad como la capacidad que tiene un sistema para ampliarse mediante la adición de hardware adicional o la actualización del hardware existente sin tener que cambiar gran parte de la aplicación.

Un sistema que escala bien es aquel que responde a los cambios en la demanda permitiendo inmediatamente sus recursos sin sufrir consecuencias negativas, como la caída del servicio o el aumento de fallas de rendimiento [7]. En un sistema de MOODLE la disponibilidad y la escalabilidad se convierten en características clave a medida que aumenta la demanda del

servicio. Este sistema puede escalar de dos formas, las cuales son descritas a continuación [8]:

Mediante el escalado vertical se puede aumentar la capacidad del hardware existente añadiendo recursos al mismo servidor. El escalado vertical mantiene la infraestructura existente, pero añade potencia de cálculo. Al aumentar la escala, se incrementa la capacidad de una sola máquina y se aumenta su rendimiento. El escalado vertical significa añadir más recursos a un único nodo y añadir más CPU, RAM y DISCO para hacer frente a una carga de trabajo creciente [9]. El escalado vertical permite que los recursos se alojen en un único nodo. Por otra parte, la escalabilidad horizontal se refiere a distribuir la carga en múltiples servidores para que funcionen como una sola unidad lógica. Es posible equilibrar la carga del sistema utilizando más de un servidor web. El escalado horizontal permite añadir más instancias de máquinas sin necesidad de implementar mejoras en las especificaciones existentes. Al escalar horizontalmente, se comparte la potencia de procesamiento y el equilibrio de carga entre varias máquinas.

El escalado horizontal significa añadir más máquinas al conjunto de recursos, en lugar de simplemente añadir recursos escalando verticalmente. El escalado horizontal es lo mismo que escalar añadiendo más máquinas a un conjunto de recursos, pero en lugar de añadir más potencia, CPU o RAM, se extiende la infraestructura existente.

1.4.2. Elasticidad

A partir de la escalabilidad surge otra característica relacionada con los servicios en la nube. Esta característica se define como la capacidad que tiene un sistema, para aumentar o disminuir sus recursos de forma automática. La provisión de estos recursos depende de la demanda del servicio, lo cual garantiza que sólo se provea la capacidad necesaria para que el sistema se mantenga operativo. De esta forma, un sistema elástico evita que se desperdicien recursos al retirarlos cuando ya no se necesiten.

La elasticidad en MOODLE se aplicaría en un escenario donde las peticiones al servicio aumenten repentinamente, por ejemplo, en época de exámenes. En este escenario, la capacidad elástica del sistema permitiría provisionar más recursos de memoria o procesamiento. Luego, cuando la demanda al servicio disminuya se retirarían estos recursos.

1.4.3. Sistema de alta disponibilidad

La alta disponibilidad de un sistema se obtiene al eliminar los puntos de fallo de una arquitectura. A medida que la demanda de un servicio aumenta, es necesario que un sistema sea lo suficientemente robusto para soportar la carga de trabajo. Los sistemas de alta disponibilidad son lo suficientemente fiables como para funcionar continuamente sin fallar [10].

1.4.4. Balanceador de carga

Para garantizar una alta disponibilidad cuando muchos usuarios acceden a un sistema, se hace necesario el equilibrio de carga. El balanceo de carga distribuye automáticamente las cargas de trabajo a los recursos del sistema, como el envío de diferentes solicitudes de datos a diferentes servicios alojados en una arquitectura de nube híbrida. El balanceador de carga decide qué recurso del sistema es más capaz de manejar eficientemente qué carga de trabajo. El uso de varios equilibradores de carga para hacer esto garantiza que ningún recurso se vea sobrecargado [11].

Los servidores de un sistema de alta disponibilidad están en clústeres y organizados en una arquitectura escalonada para responder a las peticiones de los equilibradores de carga. Si un servidor del clúster falla, un servidor replicado en otro clúster puede manejar la carga de trabajo designada para el servidor que ha fallado. Este tipo de redundancia permite la conmutación por error en la que un componente secundario asume el trabajo de un componente primario cuando el primero falla, con un impacto mínimo en el rendimiento [11].

1.4.5. Computación en la nube

La computación en nube es una nueva tecnología informática adoptada a partir de los recursos informáticos distribuidos, paralelos y en red. Según la definición del NIST, la computación en nube es un modelo de computación para el acceso a la red bajo demanda a un conjunto compartido de recursos informáticos como el servidor, la red, el almacenamiento y las aplicaciones. Estos recursos pueden estar fácilmente disponibles cuando se necesitan y liberados con el mínimo esfuerzo. Lo mejor de la computación en la nube es trabajar con todo el hardware y el software asociados de forma flexible e ininterrumpida [12]. La

computación en nube ofrece una rápida elasticidad, escalabilidad, flexibilidad, servicios bajo demanda y un modelo de pago por uso.

1.4.6. Contenedores

Los contenedores son una tecnología que permiten empacar y aislar una instancia de una aplicación. Los contenedores son intrínsecamente stateless y no conservan la información de la sesión, aunque pueden utilizarse para aplicaciones stateful. Varias instancias de una imagen de contenedor pueden ejecutarse simultáneamente, y las nuevas instancias pueden sustituir a las que fallan sin interrumpir el funcionamiento de la aplicación [13].

1.4.7. Stateless

En el caso de ser detenido o reiniciado un contenedor no mantiene los cambios realizados dentro del mismo. Esto puede resultar en perdida de información si una aplicación no está diseñada para ejecutarse en contenedores [14]. En el caso de MOODLE, la persistencia de datos en un factor crítico que debe tomarse en cuenta al distribuir la base datos, el sistema de archivos y el servicio web en base a una arquitectura de contenedores.

1.4.8. Kubernetes

Debido a la gestión de múltiples contenedores es necesario el uso de un sistema que permita controlar el estado y garantizar su disponibilidad. Kubernetes es una herramienta que permite orquestar múltiples contenedores. La alta disponibilidad se consigue cuando el sistema está disponible al menos el 99,999% del tiempo, es decir, no más de unos 5 minutos de interrupción al año. Algunas características de los contenedores, como el hecho de ser pequeños y ligeros, contribuyen naturalmente a mejorar la disponibilidad debido a la capacidad de recuperación que tiene Kubernetes [15].

1.4.9. Serverless

Serverless es un modelo de despliegue para la nube que permite ejecutar aplicaciones sin la necesidad de configurar y mantener servidores. La parte correspondiente a infraestructura y su mantenimiento le corresponde al proveedor de nube, evitando que los desarrolladores se encarguen de la administración de servidores [16]. Una arquitectura basada en serverless

permite responder en base a la demanda y escala los recursos cuando aumenta la demanda. Una vez que el servicio no reciba más peticiones se liberan los recursos, permitiendo ahorrar costos.

1.4.10. MOODLE

MOODLE es una plataforma de aprendizaje online que es ampliamente adoptada por las instituciones educativas para crear y administrar cursos en línea [17]. Esta plataforma ofrece mecanismos para compartir contenidos digitales, así como herramientas de comunicación, creación de cursos, seguimiento de las acciones de los usuarios que tienen lugar durante un curso, y puede ser enriquecida con diferentes plugins, diseñados para satisfacer las necesidades específicas de un determinado conjunto de usuarios. También es posible, a través de esta herramienta, desarrollar actividades evaluativas, como entrega de tareas, cuestionarios, entre otras acciones en actividades de aprendizaje a distancia [18].

Esta plataforma se puede diseñar para escalar y adaptarse a grandes comunidades de usuarios. Para un mayor número de estudiantes y un mayor volumen de peticiones, la plataforma debe ser lo suficientemente robusta como para servir a miles de estudiantes, administradores, creadores de contenido e instructores simultáneamente. MOODLE generalmente se implementa en un único servidor web al que tienen acceso todos los usuarios. Por lo tanto, el servidor, como piedra angular que atiende todas las solicitudes de contenido de aprendizaje, debe ser siempre accesible.

1.4.11. Arquitectura de MOODLE

El sistema de MOODLE se compone de 3 elementos principales: servicio web, base datos y sistema de archivos. Dependiendo de las necesidades de cada implementación, dichos componentes se configuran y distribuyen de diferentes formas. A continuación, se describen estos componentes:

- **Servidor Web**

Consiste en una colección de archivos PHP almacenados en un servidor web. Este componente contiene el código del núcleo, los subsistemas y los plugins. Este componente del sistema es la parte a la que acceden directamente los usuarios mediante un cliente web.

- **Base de datos**

La información del sistema está almacenada en una base de datos de más de 250 tablas. Los cursos, los usuarios, los roles, los grupos, las calificaciones y otros datos como los recursos de aprendizaje creados por los profesores, los mensajes del foro añadidos por los estudiantes y la configuración del sistema añadidos por el administrador se almacenan en su mayoría en la base de datos de la plataforma.

- **Sistema de archivos**

Corresponde a los ficheros almacenados en el directorio "moodledata" el cual contine archivos cargados al sistema por el usuario. Esto incluye documentos, presentaciones imágenes y vídeos utilizados en la plataforma.

1.5. Antecedentes de la investigación

El presente trabajo de investigación se ha basado en diferentes fuentes bibliográficas como IEEE, ACM, Scopus y Google Académico. En estas fuentes bibliográficas se ha utilizado la cadena de búsqueda ((moodle) AND (architecture OR high availability OR scalability OR redundancy)) mediante la cual se obtuvieron 6 artículos referentes a las formas de implementar una arquitectura de servicios basada en la nube, con el objetivo de identificar un modelo de arquitectura óptimo para aplicarlo en la implementación de un sistema MOODLE. Para este trabajo se han considerado estudios publicados entre el periodo 2017 – 2021, los cuales serán detallados a continuación:

Con respecto al ámbito nacional, se ha realizado un trabajo de titulación de posgrado llamado "Metodología de evaluación de riesgos MOODLE PUCE Ambato" [19] el cual está orientado a la seguridad de dicha infraestructura. Aunque también se describen las características, configuraciones y componentes de la arquitectura de alta demanda utilizada en la PUCE Sede Ambato. La plataforma de aprendizaje de esta universidad se migró a una arquitectura multicapa en AWS, que usa grupos de autoescalado para satisfacer la alta demanda. Se evaluó la infraestructura en base a la certificación ISO 27017:2015 y se obtuvo como resultado que al usar AWS la PUCE Ambato cumple con el 90% de los lineamientos establecidos por la norma ISO. Al realizar un análisis deductivo de los resultados obtenidos de aplicar esta norma ISO, el autor concluyó que la migración a los servicios de AWS permitió satisfacer la creciente demanda de los estudiantes sin tener más problemas con la disponibilidad de la aplicación y el rendimiento. Esta investigación tiene una estrecha

relación con el tema de estudio ya que analiza los beneficios de integrar los servicios de la nube para un entorno de aula virtual, de tal forma que mitigue las necesidades de disponibilidad de servicio, resistencia a la alta demanda y escalabilidad.

De igual forma, en la investigación que se titula "Improving MOODLE Architecture and Learning Features in Cloud Server Ecosystem Using Kubernetes and Gamification" [20] el objetivo era comprobar la efectividad del servicio de Kubernetes de Google Cloud Plataform con MOODLE. Mediante la simulación de escenarios, se realizaron pruebas con hasta 900 usuarios concurrentes. Los resultados indicaron que porcentaje de disponibilidad aumenta al 99% utilizado el escaldo horizontal de Kubernetes. Se concluyó que la arquitectura propuesta es lo suficientemente robusta para soportar hasta 480 usuarios concurrentes con un tiempo de respuesta 11,37 segundos. Este estudio se relaciona con la presente investigación debido al uso de Kubernetes en la nube para satisfacer la demanda del servicio del aula virtual.

Así mismo, en una investigación nacional llamada "Reserved, On Demand or Serverless: Model-based simulations for cloud budget planning" [21] se tenía como objetivo analizar los diferentes modelos cloud y proponer una herramienta que permita a los administradores diseñar, planificar y presupuestar adecuadamente sus costes de computación en la nube. En esta investigación se presentaron 4 tipos de arquitecturas con sus respectivos modelos de costes. Por medio de una metodología cuantitativa, se realizó una encuesta a diferentes empresas en la cual se obtuvo como resultado que el 36,96% de las organizaciones ecuatorianas optan por pagos mensuales fijos en lugar de modelos bajo demanda. Además, se concluyó que la herramienta propuesta "Cloudcal" permite calcular el rendimiento y los costos de diferentes modelos de precios de computación en la nube. Esta investigación se relaciona con el tema de estudio debido a que presentó diferentes modelos para el diseño y planificación de arquitecturas basadas en la nube. Estos modelos son de gran utilidad para desarrollar la propuesta de arquitectura que tiene como objetivo la presente investigación.

Por otra parte, en la investigación titulada *"Cloud infrastructure planning considering different redundancy mechanisms"* [6] el objetivo era comparar los diferentes mecanismos de redundancia y modelos de costes para infraestructuras en la nube. Esta investigación se basó en el análisis de modelos eficientes para lograr redundancia de datos y optimizar costes. En este estudio se utilizó una metodología experimental ya que se basó en la aplicación de

estos modelos en entornos reales. En este trabajo se implementó un entorno en la nube configurado con la plataforma Eucalyptus para desplegar una instancia de MOODLE usando diferentes modelos de redundancia. Se utilizó el modelo RBD (reliability block diagram) y se obtuvo como resultado una mayor disponibilidad, menor tiempo de inactividad y coste. Sin embargo, con el modelo SPN (stochastic petri net) se obtuvieron resultados similares. Luego de aplicar estos modelos, se concluyó que una combinación de los modelos RBD (reliability block diagram) y SPN (stochastic petri net) es la adecuada para optimizar aspectos de disponibilidad, tiempo de inactividad y costes de una infraestructura en la nube. Esta investigación se relaciona con el tema de estudio debido a que compara y analiza los mecanismos de redundancia que puedan proporcionar una alta disponibilidad con costes aceptables para arquitecturas basadas en la nube. También presenta un escenario de implementación de aula virtual aplicando estos mecanismos y modelos.

Además, Aleksandar Velinov [22] realizó una investigación basada en 5 meses de experiencia luego de migrar hacia la nube de Azure la plataforma MOODLE de la universidad Goce Delchev University. Su objetivo era determinar si migrar hacia la nube solucionaba los problemas de escalabilidad que afrontaba el servicio debido al creciente número de usuarios, recursos y actividades. Para poner a prueba al sistema, se observó el comportamiento de la plataforma durante la realización de un examen con 300 alumnos. Los resultados de la prueba demostraron que el uso de CPU, RAM y ancho de banda se han reducido significativamente. También se observó que el acceso a la plataforma es mucho más rápido en comparación con la instancia del servidor local. Por medio de una metodología experimental en la que se analizó el uso de recursos de hardware mediante una prueba de estrés, se concluyó que en un escenario donde el número de usuarios es grande y existen muchas peticiones al servidor, la migración a la nube garantiza una mejor escalabilidad, disponibilidad y seguridad. Este trabajo de investigación tiene relación con el tema de estudio ya que ha obtenido resultados similares a los deseados en el presente trabajo.

En una investigación llamada “Implementation of the MOODLE e-learning platform from server selection to configuration” [23] se realizó la comparación entre un aula virtual en un servidor web normal y un servidor web basado en contenedores Linux con el objetivo de escoger la mejor solución. Luego de una revisión de la literatura sobre las tecnologías

necesarias para implementar el aula virtual. Los resultados de esta investigación determinaron que una configuración óptima consistiría en un VPS (servidor virtual privado) con un sistema Linux CentOS usando docker y MariaDB como base de datos. Mediante un análisis comparativo de las características de cada tecnología en las dos arquitecturas propuestas, los investigadores concluyeron que una arquitectura de contenedores es una mejor opción ya que docker ofrece un enfoque más granular, eficiente y controlable. Este estudio se relaciona con la presente investigación en que compara diferentes enfoques para el despliegue del aula virtual.

En este mismo sentido, la investigación titulada “Fault tolerance strategy to increase MOODLE service reliability” [24] tuvo como objetivo aumentar la disponibilidad del aula virtual a pesar de posibles fallos por la alta demanda del servicio. Los autores realizaron un análisis comparativo de los tiempos de respuesta del servicio en una arquitectura de un solo servidor comparado a una arquitectura basada en un clúster de varios servidores. Los resultados arrojaron que el servicio con un solo servidor tenía un 0,11% de fallos, mientras que la arquitectura que usaba un clúster con un equilibrador de carga se podía atender el 100% de las peticiones de los usuarios. Sin embargo, el tiempo de respuesta de las peticiones aumentó al usar un clúster. Se concluyó que la implementación de un clúster tolerante a fallos mejora la fiabilidad de los servicios de la plataforma. Aunque el servidor experimentó una disminución del tiempo al responder a las peticiones de los usuarios, esto no afectó a todas las peticiones. La relación de esta investigación con el tema de estudio radica en el uso de arquitecturas basadas en clústeres para implementar instancias de MOODLE.

2. METODOLOGÍA

2.1. Delimitación

Con respecto a la delimitación espacial del estudio, la presente investigación estuvo orientada a desarrollar una propuesta de infraestructura de MOODLE para la PUCE Sede Esmeraldas. Por otra parte, esta investigación estuvo delimitada al uso de la nube como medio para desarrollar una propuesta para la implementación de esta plataforma que cumpla con los objetivos propuestos.

Por lo tanto, se abordaron cuatro formas de implementación de una instancia de MOODLE en la nube. En primer lugar, se analizaron las formas de implementar esta plataforma mediante componentes nativos de la nube, luego se comparó esta solución con la implementación de una arquitectura basada en contenedores docker la cual fue gestionada por medio de Kubernetes. Finalmente, se analizó una última propuesta la cual estuvo basada en un clúster de máquinas virtuales gestionadas por un balanceador de carga. Esta investigación se realizó durante 6 meses hasta su finalización en febrero del 2022.

2.2. Tipo de investigación

La presente investigación es de tipo aplicada debido a que se resolvió las problemáticas en la infraestructura de MOODLE de la PUCESE aplicando diferentes modelos de implementación.

La investigación es de tipo bibliográfica debido a que se basó en la recopilación de distintas fuentes científicas como artículos científicos, tesis, libros, sitios web verificados y demás recursos bibliográficos que servirán de sustento para esta investigación.

La investigación es de campo porque se analizó las condiciones actuales de la infraestructura del aula virtual dentro del departamento de TIC de la PUCESE.

De igual forma, esta investigación es de tipo experimental ya que se implementó las arquitecturas propuestas con el fin de evaluar y determinar la mejor solución.

Además, esta investigación fue de tipo cuantitativa ya que se evaluaron los resultados en cuanto al porcentaje de disponibilidad del sistema, velocidad de carga y el tiempo de respuesta del servicio en cada forma de implementación de MOODLE.

3. Métodos y técnicas

3.1.1. Métodos de la investigación

Para ejecutar esta investigación se utilizó el método deductivo, ya que se determinó de manera general las principales formas de implementación de MOODLE existentes y luego se analizaron las características específicas de cada una de estas arquitecturas.

3.1.2. Técnicas de la investigación

Para recopilar información sobre la infraestructura tecnológica existente en la PUCESE se utilizó la técnica de la entrevista. Esta entrevista estuvo dirigida al jefe del departamento de TIC de la universidad.

Por otra parte, se utilizó la técnica de la observación para identificar en qué momento y bajo qué circunstancias la plataforma empieza a experimentar problemas de rendimiento e inestabilidad.

Finalmente, mediante la técnica de recolección de documentos y registros, se analizó la información presente en los reportes de auditoría del departamento de TIC con el objetivo de identificar los problemas detectados previamente en la infraestructura.

3.2. Población y muestra

En la presente investigación se consideró como población al jefe del departamento de TIC de la PUCESE, el cual será entrevistado sobre la infraestructura tecnológica de la universidad.

Debido al tamaño de la población, en esta investigación no se estableció una muestra.

3.3. Descripción de instrumentos

Debido a que se va a ejecutar una entrevista dirigida al jefe de TIC de la PUCESE, se utilizó como instrumento un cuestionario con 10 preguntas abiertas las cuales se trataron del uso de la nube en la infraestructura de la universidad, características de los servidores físicos, herramientas de gestión de contenedores y las demás tecnologías que se utilizan en la PUCESE para el despliegue de MOODLE.

Para evaluar la calidad de cada infraestructura se utilizó la norma ISO/IEC 25010. En este instrumento se establece la eficiencia desempeño, fiabilidad como requisitos para asegurar la calidad de un producto. Para hacer esto, se utilizó una ficha de evaluación la cual consta de varios ítems en los que se evaluarán los aspectos de comportamiento temporal, utilización de recursos, capacidad, madurez, disponibilidad, tolerancia a fallos y capacidad de recuperación.

3.4. Técnicas de procesamiento y análisis de datos

Para procesar la información brindada en la entrevista se agruparon las respuestas en las siguientes categorías: uso de la nube en la infraestructura de la universidad, características de los servidores físicos y herramientas de gestión de contenedores. Luego se resumió y analizó las respuestas en base a cada uno de estos aspectos.

Por otra parte, utilizando una hoja de cálculo se tabularon los datos obtenidos en una tabla para comparar los resultados de cada una de las de infraestructuras propuestas.

De igual forma, estos resultados se interpretaron de forma escrita para valorar las características de cada infraestructura.

3.5. Normas éticas

La presente investigación se realizó en base a los lineamientos establecidos en el Reglamento de Grados de la Pontificia Universidad Católica del Ecuador Sede Esmeraldas. Este trabajo de investigación se desarrolló respetando las ideas recolectadas dentro de la literatura analizada, de esta manera se garantiza el respeto a los derechos de autor de las diferentes investigaciones y artículos abordados. De igual forma, la información recopilada es de carácter confidencial y no ha sido utilizada para otros fines ajenos a esta investigación.

4. RESULTADOS

4.1. Arquitecturas de implementación para MOODLE

En base a una revisión de la literatura disponible, se identificaron las siguientes formas de implementar MOODLE. Cada arquitectura tiene un propósito y cubre necesidades específicas en base a los requerimientos de disponibilidad y demanda del servicio.

Implementación básica: Servidor único

MOODLE es una plataforma que generalmente se despliega bajo una arquitectura monolítica por lo que sus componentes se instalan en un mismo servidor.

Una implementación básica de esta plataforma consiste en instalar el servicio web, la base datos y el sistema de archivos en un mismo servidor, sin la necesidad de distribuir estos componentes en diferentes nodos. Principalmente se instala en Linux utilizando Apache como servidor web, PostgreSQL/MySQL/MariaDB como base datos y PHP.

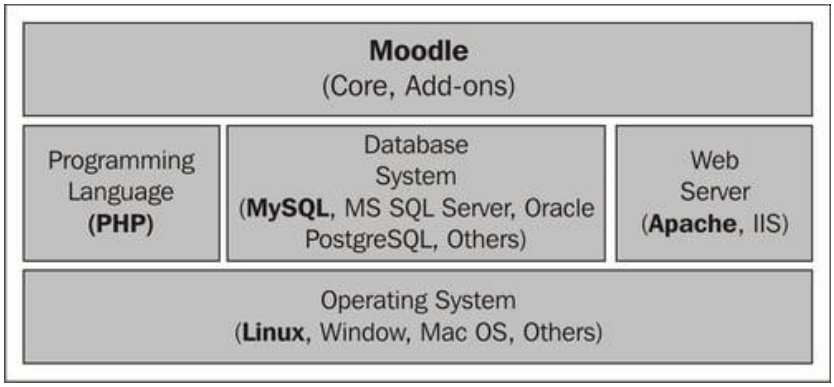


Figura 1 Arquitectura de MOODLE en un único servidor [25]

Implementación media: Clúster de servidores

Para superar el problema de que la carga del sistema tiende a saturarse en un solo servidor, se puede distribuir la carga excesivamente centralizada a varios servidores, formando un clúster. MOODLE incluye archivos del sitio web, archivos de datos y base de datos. Estos componentes se pueden separar en múltiples servidores para formar clústeres.

Esta arquitectura consiste en una estrategia de tolerancia a fallos que utiliza la tecnología de balanceo de carga. El balanceador de carga se encarga de recibir las peticiones de los clientes y de reenviarlas al servidor web correspondiente. El servidor web del servicio de MOODLE

está empaquetado en contenedores Linux que comparten las carpetas y archivos mediante un volumen distribuido como Gluster [24].

Esta solución permite que los contenedores del servidor web estén instalados de forma redundante, por lo que el balanceador de carga podría redirigir automáticamente las peticiones a otros servidores activos cuando alguno pare de funcionar.

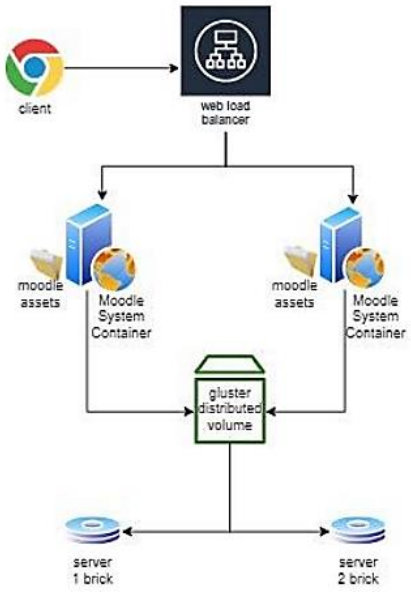


Figura 2 Arquitectura basada en servidores distribuidos [24]

Implementación avanzada: Instancias MOODLE en contenedores orquestados por Kubernetes

Un contenedor es un proceso o un conjunto de procesos aislados del resto del sistema. Todos los archivos necesarios para su ejecución son proporcionados por una imagen separada, lo que significa que los contenedores son portátiles y funcionan de la misma manera en entornos de desarrollo, prueba y producción. El sistema puede ser distribuido en múltiples contenedores, luego estas instancias se pueden gestionar mediante Kubernetes para garantizar el remplazo y escalado automático de los contenedores [26].

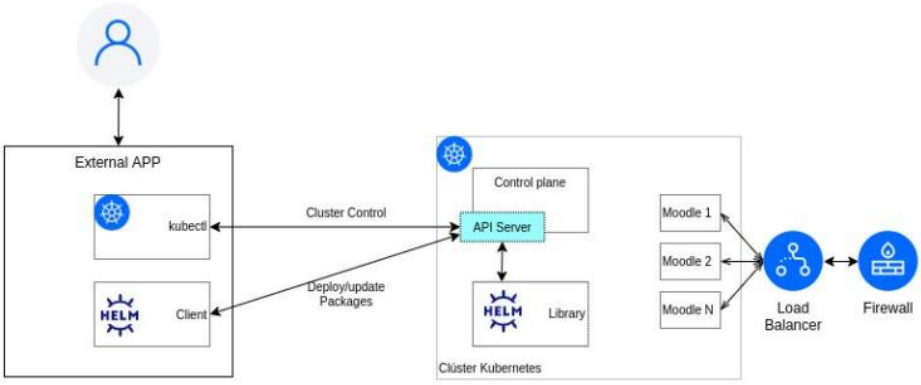


Figura 3 Arquitectura de instancias de MOODLE gestionadas por Kubernetes

Soluciones cloud: Servicios manejados por el proveedor y serverless

Las plataformas en la nube ofrecen una variedad de servicios que permiten crear arquitecturas que escalen automáticamente. Estos servicios ofrecen aumentar o reducir su capacidad automáticamente en función de las necesidades de la aplicación [27]. Además, no se necesita administrar el servidor y la configuración del sistema operativo.

En el caso de MOODLE, para automatizar el escalado del servidor web los proveedores en la nube ofrecen el servicio de "autoscaling". Este servicio responde a la carga de trabajo aumentado o disminuyendo el número de servidores web.

Con respecto al almacenamiento de datos, los proveedores en la nube ofrecen bases de datos serverless. Estos servicios serverless son compatibles con MySQL, la base de datos más usada con MOODLE. Las bases de datos serverless consisten en un sistema de almacenamiento distribuido, tolerante a errores y de recuperación automática que permite escalar verticalmente de forma automática [28].

Además, para almacenar los archivos del directorio moodledata, el cual contiene los ficheros que cargan los usuarios a la plataforma, existen diferentes soluciones serverless compatibles con el sistema de archivos NFS, el cual se usa en con esta plataforma de aprendizaje.

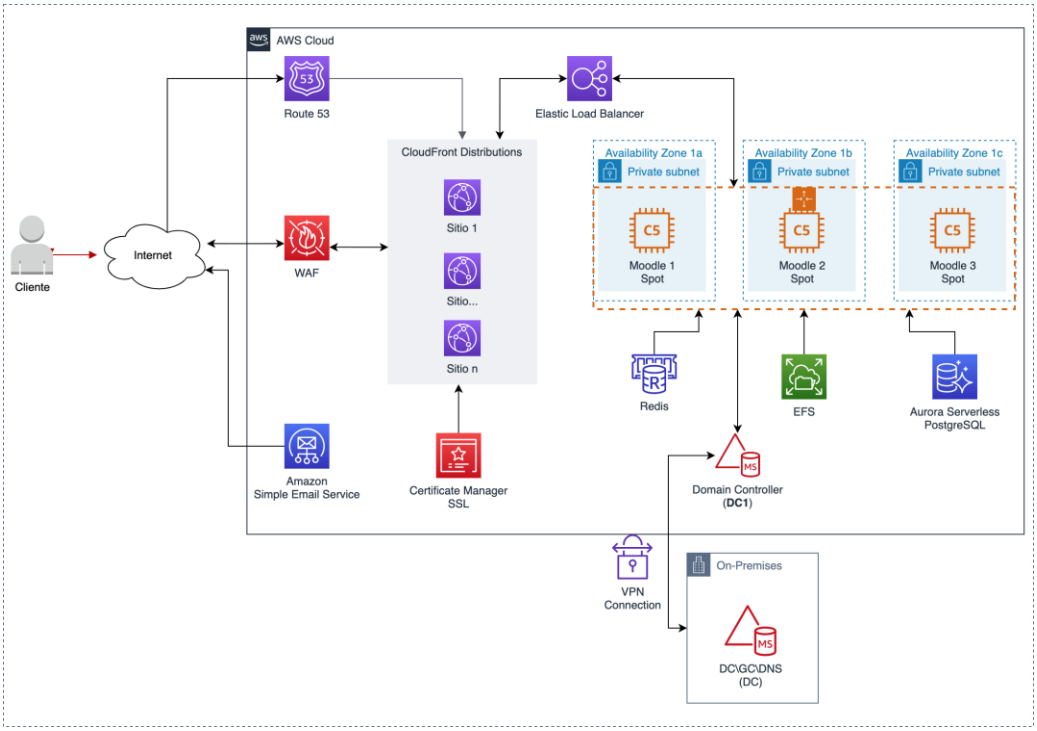


Figura 4 Arquitectura en AWS para MOODLE de alta demanda

4.2. Análisis cualitativo de las formas de implementación

La norma ISO 25010 establece 8 características principales para evaluar la calidad de un producto. Cada característica principal contiene subcaracterísticas que se usaron como parámetros para evaluar cada forma de implementación.

Se ha seleccionado la eficiencia de desempeño y fiabilidad como parámetros para analizar las arquitecturas de MOODLE antes planteadas. A continuación, se describen estas arquitecturas evaluadas en base a los parámetros establecidos por la norma ISO 25010:

Implementación básica: Servidor único

Tabla 1 Análisis de la eficiencia de desempeño arquitectura de servidor único

Eficiencia de desempeño	
Comportamiento temporal	En una arquitectura de servidor único los tiempos de respuesta del servidor y la velocidad de carga del servicio se mantienen estables mientras la cantidad de usuarios no aumente críticamente durante ciertos periodos de tiempo. Por ejemplo, durante la época de

	exámenes en donde existen una gran cantidad de usuarios utilizando el servicio a la vez.
Utilización de recursos	Al no poder distribuir la carga en otros servidores, el servicio está limitado a los recursos físicos del servidor en que esté instalado. Si existiera un incremento abrupto de la manda del servicio, los recursos de memoria y almacenamiento se pueden saturar y en consecuencia el servicio puede funcionar de forma deficiente o interrumpirse completamente. En esta arquitectura solo es posible escalar los recursos de forma vertical.
Capacidad	La capacidad de esta arquitectura para mantener el servicio operativo en base a la demanda depende exclusivamente los recursos de memoria, alucinamiento y procesamiento que ofrezca el servidor.

Tabla 2 Análisis de la fiabilidad de la arquitectura de servidor único

Fiabilidad	
Madurez	En condiciones de uso normales, esta arquitectura es lo suficientemente estable para operar sin mayores problemas de disponibilidad o rendimiento.
Disponibilidad	La disponibilidad del servicio está limitada a un único punto de redundancia, por lo cual el servicio puede dejar estar operativo bajo ciertas circunstancias en donde incremente la demanda y se agoten los recursos.
Tolerancia a fallos	Al ser una arquitectura monolítica, si uno de los componentes del servidor presenta un fallo crítico a nivel de recursos o a nivel del sistema de MOODLE, el servicio puede dejar de estar disponible.

Capacidad de recuperación	En caso de presentarse fallas en alguno de los componentes del sistema, el administrador está obligado a intervenir para solucionar el problema ya que en esta arquitectura no existe un componente que pueda restablecer el servicio automáticamente.

Implementación media: Clúster de servidores

Tabla 3 Análisis de eficiencia de desempeño en la arquitectura de clúster de servidores

Eficiencia de desempeño	
Comportamiento temporal	Bajo circunstancias de alta demanda en donde los picos de uso de recursos se elevan, este tipo de arquitectura es capaz de resistir la carga de trabajo al distribuirla en múltiples servidores. De igual forma, los tiempos de respuesta y velocidad del servicio son más eficientes.
Utilización de recursos	Este tipo de arquitectura facilita el escalado vertical al permitir añadir nuevos servidores para distribuir la carga de trabajo y los recursos.
Capacidad	Al igual que la arquitectura de un solo servidor, la capacidad de esta arquitectura está limitada a los recursos de memoria, almacenamiento y procesamiento de cada servidor en el clúster. Sin embargo, se puede aumentar su capacidad al añadir más nodos al clúster de servidores.

Tabla 4 Análisis de fiabilidad en la arquitectura de clúster de servidores

Fiabilidad	
Madurez	Bajo condiciones normales, el servicio no es afectado por problemas de disponibilidad o rendimiento. Incluso es más estable que la arquitectura basada en un servidor único.

Disponibilidad	Al soportar la alta demanda, esta arquitectura es capaz de ofrecer tiempos de disponibilidad más altos comprados con la arquitectura monolítica.
Tolerancia a fallos	Si uno de los nodos del clúster falla o alguno de sus componentes, esta arquitectura tiene mecanismos de redundancia que permiten mantener operativo el servicio a pesar de eventuales fallos.
Capacidad de recuperación	El balanceador de carga se encarga de redirigir las peticiones del cliente hacia los servidores que aún estén operativos en caso de que alguno falle. Este componente permite que se recupere la funcionalidad del servicio.

Implementación avanzada: Instancias MOODLE en contenedores orquestados por Kubernetes

Tabla 5 Análisis de eficiencia de desempeño en la arquitectura basada en Kubernetes

Eficiencia de desempeño	
Comportamiento temporal	Kubernetes se encarga de provisionar los recursos y mantener activo el servicio mediante reposición de los pods que contengan los contenedores de MOODLE. Si existe un incremento en la demanda del servicio el balanceador de carga integrado en Kubernetes distribuye las cargas de trajo a los pods correspondientes automáticamente.
Utilización de recursos	Kubernetes permite provisionar los recursos de forma vertical y horizontal en base a la demanda. Kubernetes aumenta el número de instancias de MOODLE y eventualmente disminuye el número de contenedores durante los periodos de menor actividad.

Capacidad	La capacidad de Kubernetes para aprovisionar los recursos está limitada a la capacidad de hardware que dispongan el nodo que contenga los pods y contenedores.

Tabla 6 Análisis de fiabilidad en la arquitectura basada en Kubernetes

Fiabilidad	
Madurez	Bajo condiciones normales la arquitectura es capaz de mantenerse estable. En caso de incremento en la demanda del servicio, se crean nuevos pods automáticamente.
Disponibilidad	Esta arquitectura tiene un alto nivel de disponibilidad debido a que el controlador de replicación de Kubernetes se encarga de crear nuevos pods en el caso de que un nodo falle y de replicar o escalar los pods existentes.
Tolerancia a fallos	La tolerancia a fallos de Kubernetes está gestionada por un agente del nodo llamada kubelet. Los kubelets reinician automáticamente un contenedor en caso de fallas, lo cual permite un nivel de tolerancia a fallos alto.
Capacidad de recuperación	El controlador de repliación de Kubernetes utiliza un ciclo de reconciliación que mantiene el estado deseado del clúster, lo cual garantiza que se mantenga la cantidad pods determinada por el usuario.

Soluciones cloud: Servicios manejados por el proveedor y serverless

Tabla 7 Análisis de eficiencia de desempeño en la arquitectura basada en la nube y serverless

Eficiencia de desempeño	
Comportamiento temporal	Esta arquitectura permite obtener tiempos de respuesta muy cortos debido a que distribuye los componentes de MOODLE en las zonas de locales del proveedor cloud. Estas zonas locales ubican los recursos de la nube en data centers cercanos a la petición del cliente, lo cual garantiza eficiencia en la respuesta del servicio.
Utilización de recursos	Esta arquitectura es capaz de escalar automáticamente en base demanda de recursos. Los servicios de Auto Scaling tienen la capacidad de incrementar o disminuir el número de instancias del servicio cuando sea necesario en base a la demanda.
Capacidad	Esta arquitectura tiene un nivel de capacidad alto debido a que en condiciones normales o de gran carga de trabajo es capaz de operar de forma estable.

Tabla 8 Análisis de fiabilidad en la arquitectura basada en la nube y serverless

Fiabilidad	
Madurez	Bajo condiciones normales, esta arquitectura es lo suficientemente robusta para satisfacer las necesidades de fiabilidad.
Disponibilidad	El nivel de disponibilidad para esta arquitectura es muy alto, el balanceador de carga distribuye el tráfico hacia múltiples instancias de servidores las cuales están desplegadas en múltiples zonas de disponibilidad.
Tolerancia a fallos	El nivel de tolerancia a fallos es muy alto, ya que el servicio de Auto Scaling se encarga de mantener la

	cantidad correcta de servidores disponibles para manejar la carga del sistema.
Capacidad de recuperación	En caso de presentarse fallas, el servicio Auto Scaling se encarga de restablecer las instancias que presenten problemas.

4.2.1. Resumen del análisis cualitativo de las arquitecturas

Las arquitecturas se han evaluado en base a dos características principales: Eficiencia de desempeño y fiabilidad. Cada una de estas características tiene se divide en sus respectivas subcaracterísticas. Las arquitecturas con mejores resultados en cuanto a comportamiento temporal, utilización de recursos y capacidad son aquellas que se basan en la distribución del servicio de MOODLE en diferentes nodos. Por otra parte, la arquitectura de un único servidor o monolítica no ha mostrado un óptimo funcionamiento en estos parámetros.

La arquitectura de clúster de servidores y la arquitectura de Kubernetes comparten resultados similares en cuanto a madurez, disponibilidad, tolerancia a fallos y capacidad de recuperación. La arquitectura de servicios cloud y serverless ha obtenido los mejores resultados al ser evaluada cualitativamente.

4.2.2. Resultados de la aplicación del instrumento ISO/IEC 25010

Una vez realizada la revisión bibliográfica, se procedió a valorar cada arquitectura en base a las características del instrumento ISO/IEC 25010. Esta valoración consiste en una escala que se compone de tres niveles de niveles de cumplimiento: alto, medio, bajo.

Arquitectura	**Eficiencia de desempeño**								
	Comportamiento temporal			**Utilización de recursos**			**Capacidad**		
	Alto	Medio	Bajo	Alto	Medio	Bajo	Alto	Medio	Bajo
Arquitectura de servidor único		X			X				X
Arquitectura de clúster de servidores		X		X			X		
Arquitectura de Kubernetes		X		X			X		
Arquitectura de servicios cloud y serverless	X			X			X		

Tabla 9 Comparativa de arquitecturas en base a la eficiencia de desempeño

La Tabla 9 se compone de la característica "eficiencia de desempeño" y sus respectivas subcaracterísticas. Esta característica está orientada a evaluar la cantidad de recursos utilizados bajo determinadas condiciones.

En la subcaracterística "comportamiento temporal" se evalúan los tiempos de respuesta y procesamiento de cada arquitectura, de las cuales todas tienen un nivel medio. Excepto por la arquitectura basada en la nube, cuyo nivel de "comportamiento temporal" es alto. Con respecto a la "utilización de recursos", todas las arquitecturas tienen un nivel alto a excepción de la arquitectura de servidor único cuyo nivel es medio.

En cuanto a la subcaracterística de "capacidad" se evalúan los límites máximos de cada arquitectura. Todas las arquitecturas tienen un nivel alto en este parámetro, mientras que la arquitectura de servidor único tiene un nivel bajo. La arquitectura de servicios cloud y serverless obtuvo un nivel alto en todos los parámetros.

De igual forma, la característica de “fiabilidad” está divida en sus respectivas subcaracterísticas como se observa en la Tabla 10. La fiabilidad es la capacidad que tienen las arquitecturas para funcionar bajo circunstancias y tiempos determinados.

Arquitectura	**Fiabilidad**											
	Madurez			**Disponibilidad**			**Tolerancia a fallos**			**Capacidad de recuperación**		
	Alto	Medio	Bajo	Alto	Medio	Bajo	Alto	Medio	Bajo	Alto	Medio	Bajo
Arquitectura de servidor único			X		X				X			X
Arquitectura de clúster de servidores		X			X		X				X	
Arquitectura de Kubernetes	X			X			X			X		
Arquitectura de servicios cloud y serverless	X			X			X			X		

Tabla 10 Comparativa de arquitecturas en base a la fiabilidad

La subcaracterística de “madurez” describe la capacidad de las arquitecturas para operar bajo condiciones estables. Las arquitecturas de Kubernetes y cloud tienen un nivel alto en este parámetro. Por otra parte, el parámetro “disponibilidad” define la capacidad de una arquitectura para mantenerse operativo. Las arquitecturas cloud y de Kubernetes tienen un nivel alto en este apartado.

En cuanto a la tolerancia a fallos, la arquitectura de servidor único cuenta con un nivel bajo mientras que el de las demás es alto. Con respecto a la subcaracterística de “capacidad de recuperación”, las arquitecturas cloud y Kubernetes obtuvieron un nivel alto mientras que las restantes obtuvieron un nivel medio o bajo. Según se observa en la Tabla 10, la arquitectura de servicios cloud y serverless tiene un alto nivel en todas las subcaracterísticas.

4.2.3. Resultados de la entrevista

Se realizó una entrevista al encargado de MOODLE en el Departamento de TIC de la PUCESE. El entrevistado aportó la siguiente información: De un total de 119 cursos, el aula virtual cuenta con 2400 usuarios registrados, de los cuales 50 a 200 en promedio se mantienen activos o concurrentes. El servicio recibe varias peticiones por segundo, las cuales aumentan principalmente durante exámenes, en el inicio del semestre, cuando se hacen las

sincronizaciones de los cursos y durante el registro de notas. Además, existe un aumento considerable de usuarios activos entre las 3:00 PM y 5:00 PM cada día.

Según el encargado de MOODLE, los usuarios tienen una buena percepción del rendimiento del aula virtual, sin embargo, en momentos de mayor actividad como por ejemplo durante los exámenes, mencionan cierta lentitud en el servicio. En las 8 semanas previas a la realización de la entrevista, el aula virtual tenía un porcentaje de disponibilidad del 100%. Mientras que la instancia del semestre anterior a esta tenía una disponibilidad del 99,61 %. La versión del aula se actualiza cada semestre.

En cuanto a los recursos de hardware para el funcionamiento del aula virtual, se utilizan en promedio 51,2 MB de memoria RAM por usuario. Con respecto al almacenamiento, se utilizan 120 GB de espacio en disco por semestre. En cuanto a capacidad de procesamiento, se ocupa alrededor de un CPU por cada contenedor. En la PUCESE, el aula virtual está alojado en un servidor proporcionado por CEDIA el cual contienen los múltiples contenedores que pertenecen al servicio del aula virtual. Dichos contenedores no están gestionados u orquestados. El encargado del aula virtual considera que se podría mejorar la gestión de la plataforma de aprendizaje mediante el uso de un orquestador de contenedores como Kubernetes.

En caso de algún fallo del aula virtual, el administrador debe solucionar esto de forma manual. Por lo cual debe revisar los archivos de configuración de nginx y los logs para luego reiniciar el contenedor, si es necesario. El encargado considera que una buena opción para automatizar este proceso sería utilizar un orquestador mediante un servicio en la nube. Él argumenta que esto sería beneficioso, ya que se daría continuidad al servicio de MOODLE el cual siempre debería estar disponible.

Según el jefe de TIC, debido a acuerdos con la PUCESE, una solución basada en la nube de Microsoft Azure sería conveniente.

4.3. Propuesta de implementación

Como resultado de la presente investigación se desarrolló una propuesta de arquitectura para MOODLE. Esta propuesta tiene como objetivo mejorar el funcionamiento del aula virtual en la PUCESE, en base a los siguientes aspectos:

- Escalar la nube para satisfacer la demanda y capacidad del entorno de aprendizaje virtual
- Implementación rentable
- Alta disponibilidad y escalabilidad

4.3.1. Requerimientos establecidos

En base información recogida acerca de las necesidades de la PUCESE sobre su sistema de aula virtual, se ha desarrollado una propuesta de arquitectura basada en la nube que cumpla con los siguientes requerimientos:

- Los componentes de la arquitectura deben estar distribuidos en múltiples instancias redundantes para garantizar una alta disponibilidad.
- La arquitectura debe escalar automáticamente, de tal modo que se aumenten los recursos de hardware cuando se lo requiera y se los disminuya en base a la demanda.
- La arquitectura debe ser estable y resistente a fallos. En caso de presentarse una pérdida del servicio, se debe restablecer automáticamente.
- La arquitectura debe garantizar un rendimiento óptimo en la velocidad de carga del servicio.
- La arquitectura debe utilizar los recursos de forma eficiente para reducir costos.

4.3.2. Componentes de la arquitectura

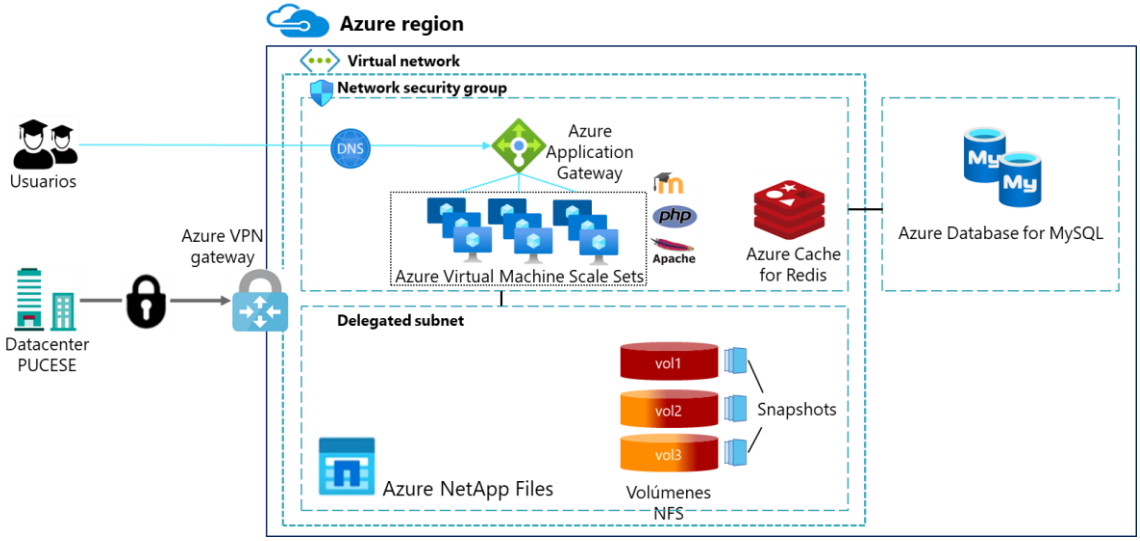


Figura 5 Diseño de la arquitectura para MOODLE en la nube de Azure

- Clúster de servidores autoescalables
- Balanceador de carga
- Base de datos serverless
- Servidor NFS
- Servidor de caché
- CDN

4.3.3. Descripción del funcionamiento de la arquitectura

A continuación, se describirá una propuesta de arquitectura para MOODLE basada en componentes genéricos lo cual permite implementar esta solución en cualquier proveedor de la nube:

Esta arquitectura consiste en un clúster de instancias que contengan el servidor web Apache con PHP. Los recursos y la cantidad de estas instancias escalan automáticamente en base a la demanda. Al frente de estas instancias un balanceador de carga distribuye las peticiones del servicio hacia las instancias en funcionamiento en caso de que alguna falle.

Por otra parte, el sistema de archivos el cual almacena los datos cargados en el directorio moodledata es gestionado por un servidor NFS. Las diferentes instancias se enlazan a una

base de datos serverless que sea compatible con MySQL. Al ser de tipo serverless, esta base de datos aumentará o reducirá su capacidad automáticamente en base a las peticiones realizadas.

MOODLE permite utilizar un servicio de caché para guardar los datos de la sesión del usuario para y mantener el estado de la aplicación. Esto permite, además, acelerar la carga de datos del lado del cliente. En esta arquitectura se propone el uso de un servidor de caché como memcached o Redis. Además, se sugiere el uso de una CDN para acelerar la carga del sitio y prevenir ataques de denegación del servicio.

4.3.4. Pruebas de carga con JMeter

La arquitectura propuesta se implementó en la nube de Microsoft Azure. Luego, se realizaron pruebas de carga para comprobar su estabilidad y rendimiento. Para este proceso se utilizó la herramienta Apache JMeter con la cual se generaron escenarios de carga con 100, 300 y 1000 usuarios concurrentes. En cada prueba se simuló el proceso de iniciar sesión y ver un curso en los que se obtuvo los siguientes resultados:

Tabla 11 Resultados de pruebas de carga con JMeter

Proceso	Medida	Usuarios concurrentes		
		100	300	1000
Frontpage logged	Average	1779 ms	817 ms	102023 ms
	Median	1706 ms	800 ms	38757 ms
	Min	1426 ms	400 ms	1679 ms
	Throughput (peticiones/minuto)	14,25	22,718	17.6
	Error %	0%	0%	38.33%
View course	Average	1364 ms	71718 ms	160975 ms
	Median	1260 ms	1261 ms	38783 ms
	Min	896 ms	885 ms	3169 ms
	Throughput	0,1705 ms	0,14739 ms	3.8/min
	Error %	0%	18,93%	35.19%

El parámetro “average” indica el tiempo promedio que se tarda una muestra en ejecutar cada proceso. El valor “Throughput” corresponde a la tasa de transferencia, la cual consiste en el número de peticiones que el servidor procesa por minuto. El parámetro de error indica la cantidad de peticiones fallidas hacia el servidor.

Se puede observar que mientras mayor sea la cantidad de usuarios concurrentes, el porcentaje de peticiones fallidas aumenta. De igual forma, la tasa de transferencia se incrementa proporcionalmente a la cantidad de usuarios simulados. En el escenario con 100 usuarios concurrentes no existe un porcentaje de peticiones fallidas. Por otra parte, en el escenario con 300 usuarios concurrentes el porcentaje de error es nulo en el proceso de iniciar sesión, mientras que en el proceso de ver un curso existe un porcentaje de error del 18%.

5. DISCUSIÓN

En la presente investigación se tuvo como objetivo proponer una arquitectura en la nube escalable y de alta demanda para MOODLE. En este sentido, se puede comparar los resultados obtenidos con estudios previos que plantean escenarios en donde se utilice esta plataforma.

En un estudio realizado en la PUCE Sede Ambato se obtuvo un 99.741% de disponibilidad del servicio del aula virtual al migrar a una arquitectura multicapa en AWS y utilizar grupos de autoescalado para satisfacer sus ciclos de demanda [19], en contraste, los resultados del presente estudio muestran un 0% de error en peticiones al servicio en base a 300 usuarios concurrentes. Dicha cantidad de usuarios es inferior al rango de 50 a 200 usuarios concurrentes actuales en la PUCESE, lo que indica que la arquitectura propuesta en esta investigación cumple con las necesidades actuales y futuras de disponibilidad del servicio.

De igual forma, en otro estudio basado en una arquitectura de MOODLE ejecutada en la nube de Google Cloud Plataform [20], se obtuvo que en una prueba de carga con 480 usuarios concurrentes el tiempo de respuesta fue 26,277 segundos. Este tiempo de respuesta es elevado comparado con los 1,676 segundos obtenidos con 1000 usuarios concurrentes en la presente investigación.

Estos resultados muestran que la arquitectura propuesta es estable y resiste a cargas de trabaja de hasta 300 usuarios concurrentes sin presentar errores al cargar iniciar sesión en el sitio. Sin embargo, se observó que existe un aumento de peticiones fallidas en el proceso de ingresar a un curso. Esto puede ser causado por un aumento repentino de consultas a la base datos. No obstante, este escenario de 300 usuarios concurrentes en un mismo curso no es común ya que actualmente el total de usuarios concurrentes en momentos de alta demanda va desde 150 hasta máximo 200 distribuidos en diferentes cursos.

6. CONCLUSIONES

Teniendo en cuenta los objetivos planteados y los resultados obtenidos en esta investigación, se puede concluir lo siguiente:

Se identificaron cuatro arquitecturas principales para implementar el servicio de MOODLE. Cada una de estas arquitecturas tienen sus características y limitaciones propias, así como su aplicación en diferentes escenarios.

Por medio la aplicación del instrumento de evaluación ISO/IEC 25010 se determinó que una arquitectura basada en la nube obtiene un alto nivel de cumplimiento en los parámetros de disponibilidad, tolerancia a fallos y capacidad de recuperación.

En base a la información recopilada en la PUCESE, se identificó la necesidad de mantener una alta disponibilidad del servicio del aula virtual, así como la capacidad de recuperarse automáticamente en caso de fallos. Para satisfacer estos requerimientos, se identificó como una solución el uso de servicios de autoescalado y serverless proporcionados por un proveedor en la nube.

Finalmente, se elaboró una arquitectura basada en la nube, cuyos componentes sean gestionados por el proveedor y la provisión de recursos aumente o disminuya en base a la demanda del servicio para satisfacer los requerimientos de alta demanda y escalabilidad pedidos por la PUCESE.

7. RECOMENDACIONES

Se recomienda que próximas investigaciones exploren otras arquitecturas para implementar MOODLE, esto puede incluir soluciones híbridas compuestas por las arquitecturas descritas en esta investigación.

De igual forma, en futuras investigaciones se podría comparar las arquitecturas en base a datos cuantitativos para un análisis desde otra perspectiva.

Con respecto a la PUCESE, se recomienda al departamento de TIC implementar la arquitectura propuesta en esta investigación. Al migrar la infraestructura de MOODLE hacia la nube se obtendrían ventajas que ofrecen los servicios gestionados por el proveedor y de tipo serverless. De esta forma, se disminuye los esfuerzos y recursos orientados a la gestión y configuración de los entornos para ejecutar la plataforma.

En lo referente a la propuesta final, se recomienda experimentar con otros servicios en la nube que remplacen a los que se proponen en esta investigación. Por ejemplo, se propone utilizar Kubernetes en lugar de servidores autoescalables.

8. REFERENCIAS

[1] UNESCO, “290 million students out of school due to COVID-19: UNESCO releases first global numbers and mobilizes response,” *290 million students out of school due to COVID-19: UNESCO releases first global numbers and mobilizes response*, 2020. https://en.unesco.org/news/290-million-students-out-school-due-covid-19-unesco-releases-first-global-numbers-and-mobilizes (accessed Jun. 01, 2021).

[2] MOODLEDocs, “Acerca de MOODLE - MOODLEDocs,” *Acerca de MOODLE*, 2020. https://docs.MOODLE.org/all/es/Acerca_de_MOODLE (accessed May 25, 2021).

[3] MOODLEDocs, “19/A favor de MOODLE - MOODLEDocs,” *A favor de MOODLE*, 2018. https://docs.MOODLE.org/all/es/19/A_favor_de_MOODLE (accessed May 25, 2021).

[4] A. Sharif, “Clustering MOODLE on Multiple Servers for High Availability and Scalability | Severalnines,” *Clustering MOODLE on Multiple Servers for High Availability and Scalability*, 2021. https://severalnines.com/database-blog/clustering-MOODLE-multiple-servers-high-availability-and-scalability (accessed May 25, 2021).

[5] M. Sadikin, R. Yusuf, and A. Rifai, “Load balancing clustering on MOODLE LMS to overcome performance issue of e-learning system,” *Jalan Meruya Selatan*, vol. 17, no. 1, 2019, doi: 10.12928/TELKOMNIKA.v17i1.10284.

[6] E. Sousa, F. Lins, E. Tavares, and P. Maciel, “Cloud infrastructure planning considering different redundancy mechanisms,” *Computing*, vol. 99, no. 9, pp. 841–864, Sep. 2017, doi: 10.1007/s00607-016-0533-6.

[7] A. Al-Said Ahmad and P. Andras, “Cloud-based software services delivery from the perspective of scalability,” *International Journal of Parallel, Emergent and Distributed Systems*, vol. 36, no. 2, pp. 53–68, 2021, doi: 10.1080/17445760.2019.1617864.

[8] F. Rossi, M. Nardelli, and V. Cardellini, “Horizontal and vertical scaling of container-based applications using reinforcement learning,” in *IEEE International Conference on Cloud Computing, CLOUD*, Jul. 2019, vol. 2019-July, pp. 329–338. doi: 10.1109/CLOUD.2019.00061.

[9] A. Gandhi, P. Dube, A. Karve, A. Kochut, and L. Zhang, "Model-driven optimal resource scaling in cloud," *Software & Systems Modeling 2017 17:2*, vol. 17, no. 2, pp. 509–526, Feb. 2017, doi: 10.1007/S10270-017-0584-Y.

[10] M. R. Mesbahi, A. M. Rahmani, and M. Hosseinzadeh, "Reliability and high availability in cloud computing environments: a reference roadmap," *Human-centric Computing and Information Sciences*, vol. 8, no. 1, pp. 1–31, Dec. 2018, doi: 10.1186/S13673-018-0143-8/TABLES/6.

[11] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A big picture," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 2, pp. 149–158, Feb. 2020, doi: 10.1016/J.JKSUCI.2018.01.003.

[12] H. Singh, S. Tyagi, and P. Kumar, "High Availability and Accessibility of Services in Cloud Environment," *Proceedings - 4th International Conference on Computing Sciences, ICCS 2018*, pp. 67–71, Jan. 2019, doi: 10.1109/ICCS.2018.00017.

[13] "What Are Containers (Container-based Virtualization or Containerization)?" https://searchitoperations.techtarget.com/definition/container-containerization-or-container-based-virtualization (accessed Nov. 15, 2021).

[14] "Stateless vs Stateful Containers: What's the Difference and Why Does It Matter? | Contino | Global Transformation Consultancy." https://www.contino.io/insights/stateless-vs-stateful-containers-whats-the-difference-and-why-does-it-matter (accessed Nov. 15, 2021).

[15] L. A. Vayghan, M. A. Saied, M. Toeroe, and F. Khendek, "Deploying Microservice Based Applications with Kubernetes: Experiments and Lessons Learned", doi: 10.1109/CLOUD.2018.00148.

[16] E. Jonas *et al.*, "Cloud Programming Simplified: A Berkeley View on Serverless Computing," Feb. 2019, Accessed: Nov. 15, 2021. [Online]. Available: https://arxiv.org/abs/1902.03383v1

[17] "About MOODLE - MOODLEDocs." https://docs.MOODLE.org/311/en/About_MOODLE (accessed Jun. 29, 2021).

[18] Z. J. Hamad and S. R. M. Zeebaree, “Recourses Utilization in a Distributed System: A Review,” *International Journal of Science and Business*, vol. 5, no. 2, pp. 42–53, 2021.

[19] K. P. M. Guamán, “METODOLOGÍA DE EVALUACIÓN DE RIESGOS MOODLE PUCE AMBATO.” Accessed: Jun. 14, 2021. [Online]. Available: https://repositorio.pucesa.edu.ec/handle/123456789/3080

[20] Y. Utomo, “Improving MOODLE Architecture and Learning Features in Cloud Server Ecosystem Using Kubernetes and Gamification,” *International Journal of Emerging Trends in Engineering Research*, vol. 8, pp. 1275–1283, Apr. 2020, doi: 10.30534/ijeter/2020/55842020.

[21] E. F. Boza, C. L. Abad, M. Villavicencio, S. Quimba, and J. A. Plaza, “Reserved, on demand or serverless: Model-based simulations for cloud budget planning,” in *2017 IEEE 2nd Ecuador Technical Chapters Meeting, ETCM 2017*, Jan. 2018, vol. 2017-January, pp. 1–6. doi: 10.1109/ETCM.2017.8247460.

[22] “Migration of MOODLE instance to the cloud - case study at Goce Delchev University - UGD Academic Repository.” https://eprints.ugd.edu.mk/28020/ (accessed Jun. 15, 2021).

[23] O. Soufiane, K. Maha, E. Mohamed, and K. Mohamed, “Implementation of the MOODLE e-learning platform from server selection to configuration,” 2021, doi: 10.30574/gscaet.2021.1.1.0023.

[24] A. Zaini, H. Santoso, and M. P. T. Sulistyanto, “Fault tolerance strategy to increase MOODLE service reliability,” in *Journal of Physics: Conference Series*, Apr. 2021, vol. 1869, no. 1, p. 12095. doi: 10.1088/1742-6596/1869/1/012095.

[25] A. G. Büchner, “MOODLE 2 administration : an administrator’s guide to configuring, securing, customizing, and extending MOODLE,” p. 396, 2011.

[26] P. Nieto Iglesias, “Diseño y desarrollo de una plataforma para el despliegue automático de microinstancias de MOODLE usando Kubernetes,” Feb. 2021, Accessed: Aug. 21, 2021. [Online]. Available: http://castor.det.uvigo.es:8080/xmlui/handle/123456789/539

[27] “Amazon Aurora Serverless | Base de datos relacional de MySQL PostgreSQL | Amazon Web Services.” https://aws.amazon.com/es/rds/aurora/serverless/ (accessed Feb. 12, 2022).

[28] “Servicio de base de datos relacional moderno – Preguntas frecuentes sobre Amazon Aurora – Amazon Web Services.” https://aws.amazon.com/es/rds/aurora/faqs/ (accessed Feb. 13, 2022).

ANEXOS

ENTREVISTA SEMIESTRUCTURADA

¿Cuántos usuarios registrados existen?

Tenemos 2400 usuarios registrados

¿Cuántos usuarios concurrentes existen?

De 78 a 80, esto varía dependiendo de, por ejemplo, cuando son exámenes esto sube un poco.

¿Cuántos cursos existen?

Tenemos 119 cursos.

¿Cuántas peticiones por segundo recibe el aula virtual?

Incluso no es por segundo; pero una por segundo, lo cual sería el mínimo. Existe un aumento cuando hay exámenes, cuando inicia el semestre, cuando se hacen las sincronizaciones de los cursos, cuando hay registro de notas.

¿En qué horarios existe un aumento de usuarios activos?

Los usuarios aumentan un poco entre las 4 y 5 de la tarde o de 3 a 5. Pero más que todo, en las ocasiones que hay registro de notas o en la semana de evaluación.

¿Cuál es la percepción de los usuarios acerca del rendimiento del aula virtual?

Cuando hay examen dicen que está un poco lenta pero generalmente hay buena percepción del funcionamiento del aula.

¿Cuál es el porcentaje de disponibilidad del servicio de MOODLE?

Según el uptime robot, el MOODLE tiene una disponibilidad del 100 % en las últimas 6 u 8 semanas desde que está funcionando. El aula anterior, que estaba con otra imagen de Docker, tenía una disponibilidad del 99,61 %

¿Con que frecuencia se actualiza la versión de MOODLE?

Cada semestre.

¿Cómo está alojado MOODLE actualmente?

Está una sobre una máquina rentada de CEDIA y está contenerizado. No están orquestados.

¿Cómo se podría mejorar el despliegue de MOODLE en la PUCESE?

Para el levantamiento algún docker-compose que permite orquestar o kubernetes.

¿Cuánta memoria RAM se necesita por usuario?

Son 20 usuarios por GB de memoria. 51,2 MB por usuario.

¿Cuánto espacio de almacenamiento se ocupa por cada semestre?

120 GB por semestre

¿Cuánta capacidad de procesamiento se necesita?

Un CPU por cada contenedor.

¿Cómo se está monitoreando MOODLE actualmente?

Para monitorearlo tenemos uptime robot, nos envía mensajes a través de telegram para tener los reportes en vivo.

¿Qué sucede en caso de un fallo?

En caso de algún fallo, tenemos que conectarnos a la maquina CEDIA, analizar que el contenedor esté levantado, si está levantado habría que revisar los archivos de configuración de nginx y se procede a, por lo menos, reiniciar el contenedor. Si no, depende de los logs tenemos que tomar distintos caminos para ver como solucionamos. Se hace de forma manual.

¿Cómo se podría mejorar esta parte?

Automatizando, orquestando, utilizando una nube; Amazon o Azure. Nos beneficiaria en que no tendríamos que estar pendientes al momento que se cae, sino, que el orquestador se encargaría de restaurar el servicio el otro contenedor y se daría continuidad al servicio de MOODLE que siempre debería estar disponible.

RESULTADOS DE PRUEBAS EN JMETER

100 usuarios concurrentes

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
Frontpage not logged	309	686	698	852	1062	1918	259	2286	0.324%	0,15866	5,19	0,02
View login page	288	516	400	653	875	1442	342	2127	0.000%	0,14808	4,27	0,03
Login	277	2927	2858	3228	3502	4322	2427	5145	0.000%	0,14289	30,68	0,12
Frontpage logged	273	1779	1706	2013	2184	3286	1426	5065	0.000%	0,1425	30,43	0,06
View course	268	1364	1260	1782	2034	3016	896	3826	0.000%	0,1705	73,3	0,04
View a page activity	268	1097	1064	1328	1471	2578	784	3059	0.000%	0,17396	42,61	0,04
View course again	268	1279	1232	1555	1767	2767	864	3801	0.000%	0,1755	75,45	0,04
View a forum activity	263	5147	2373	6817	10170	19843	1440	164806	0.760%	0,17403	287,73	0,04
View a forum discussion	245	24933	1752	132469	150120	160624	818	162632	8.571%	0,16412	127,35	0,03
Fill a form to reply a forum discusión	238	7605	1339	9480	11138	142776	905	152703	2.941%	0,16115	65,36	0,04
Send the forum discussion reply	237	1446	674	1105	2222	6988	434	125567	2.954%	0,16217	0,76	0,1
View course once more	236	3868	1176	2732	3105	125529	856	132486	0.424%	0,16315	69,89	0,04
View course participants	236	1091	1021	1468	1753	2748	747	2903	0.000%	0,16517	41,27	0,04
Logout	230	980	944	1270	1579	2289	602	2582	0.000%	0,16653	5,97	0,08
TOTAL	3636	3760	1149	2954	4493	132056	259	164806	1.073%	1,86692	679,37	0,6

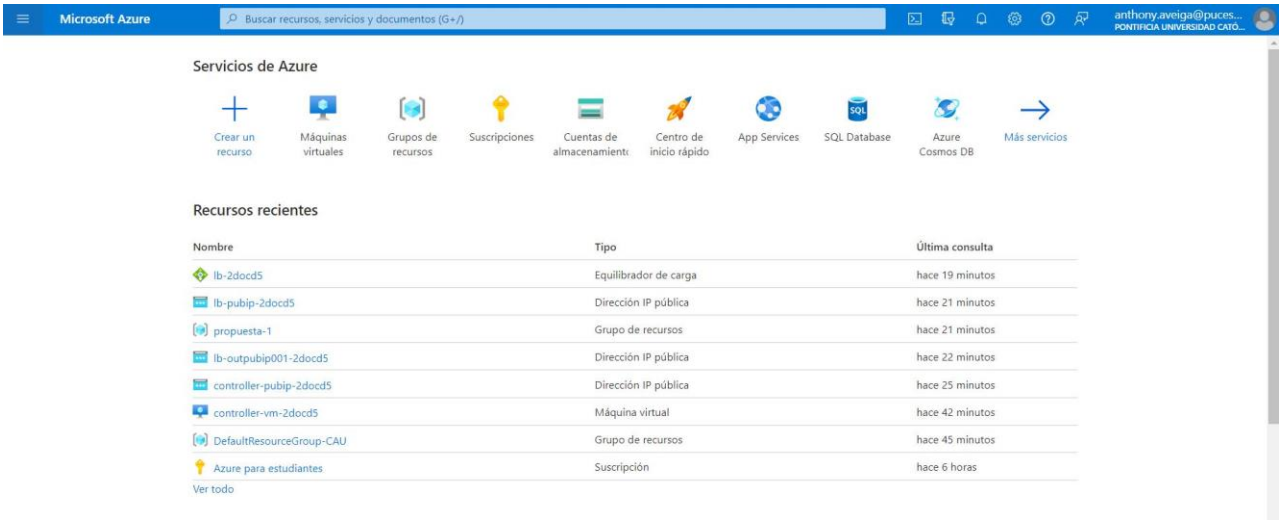
300 usuarios concurrentes

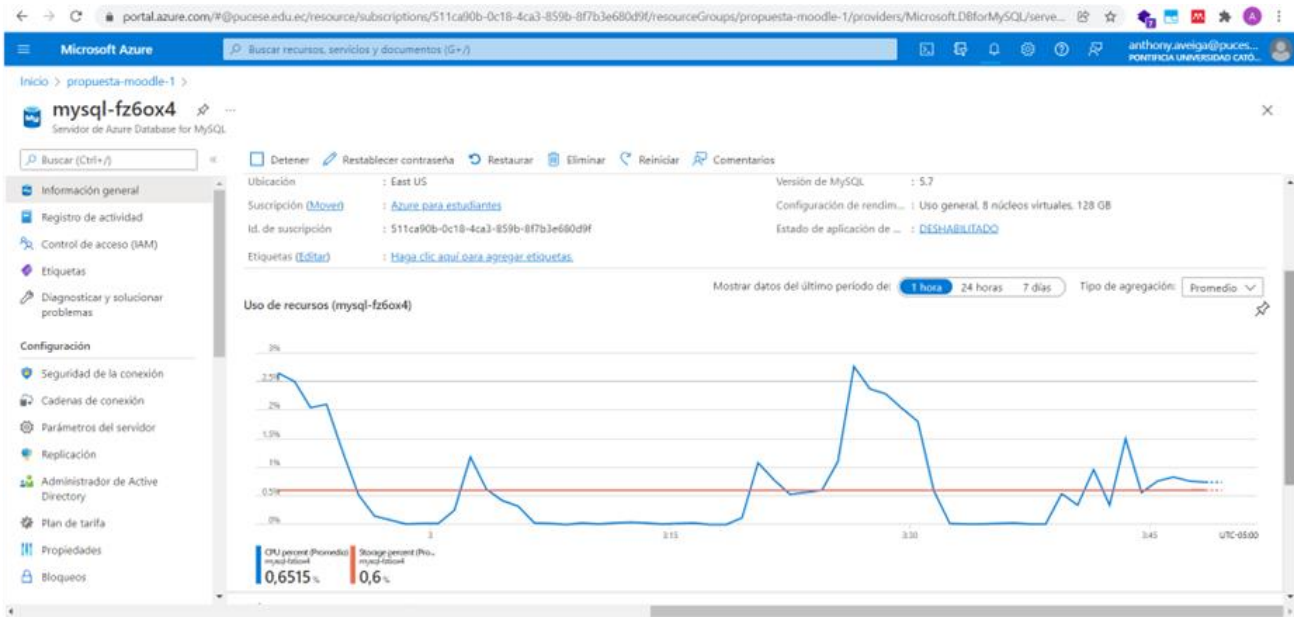
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	Received KB/sec	Sent KB/sec
Frontpage not logged	415	817	800	896	1138	3685	400	4577	0.000%	0,22718	7,45	0,03
View login page	387	616	588	874	951	1294	332	2302	0.000%	0,20992	6,1	0,05
Login	365	2323	2286	2548	2861	3803	1623	4274	0.000%	0,19869	42,08	0,17
Frontpage logged	319	1765	1720	1921	2313	3071	1161	3362	0.000%	0,17753	37,62	0,07
View course	243	71718	1261	362680	368832	458774	885	462373	18.930%	0,14739	51,45	0,03
View a page activity	172	105061	10089	362271	366352	369511	1005	454084	34.302%	0,11944	19,35	0,02
View course again	154	42122	7205	87005	353990	365303	1077	369569	13.636%	0,18634	69,26	0,04
View a forum activity	132	30356	18349	76215	79937	357157	1889	459245	7.576%	0,18752	288,78	0,04
View a forum discussion	116	9854	6455	14635	18378	66849	1064	68620	4.310%	0,43865	153,74	0,1
Fill a form to reply a forum discusión	115	5639	2233	10813	12047	70319	903	70403	0.870%	0,41439	150,54	0,1
TOTAL	2878	17482	1350	12047	76215	366352	332	462373	4.969%	1,55483	382,44	0,52

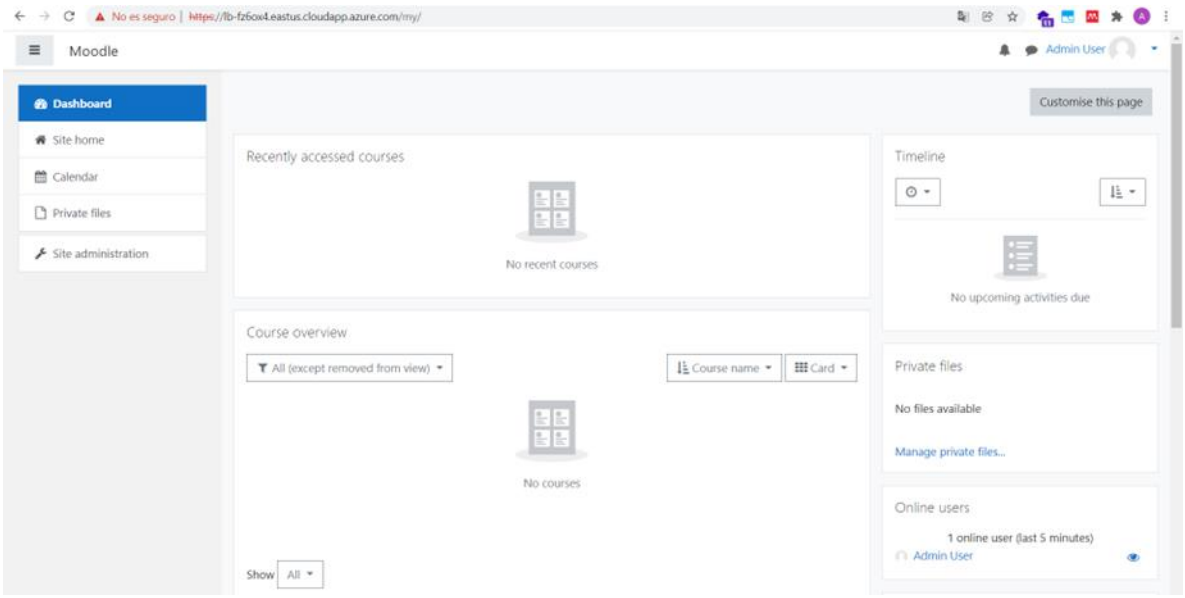
1000 usuarios concurrentes

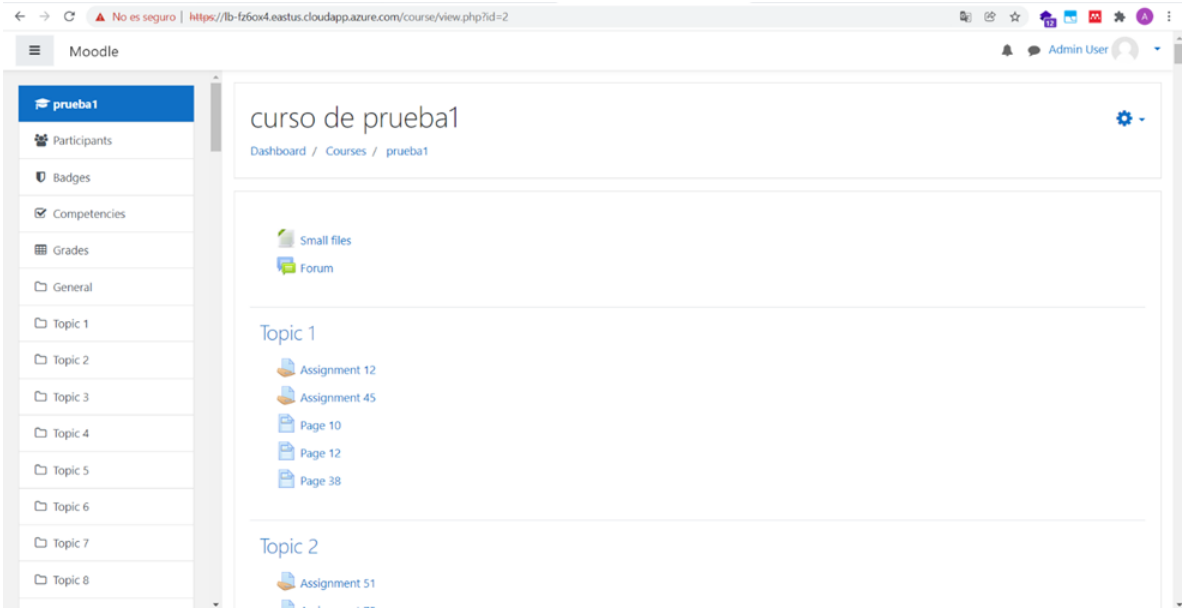
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB/sec	Sent KB/sec
Frontpage not lo...	1000	1217	936	1081	2054	5411	822	21054	0.70%	8.3/sec	269.65	1.12
View login page	799	1174	882	1246	1766	15684	426	21045	0.50%	47.0/min	23.23	0.29
Login	925	148861	114217	305514	362139	437062	929	941640	83.24%	24.8/min	16.33	0.08
Moodie Test:Vie...	201	934	877	1035	1299	2324	743	2393	0.00%	57.2/min	28.80	0.42
Moodie Test:Login	21	3059	2929	3201	4178	4571	2651	4571	0.00%	6.0/min	20.71	0.09
Frontpage logged	574	102023	38757	274402	408326	755433	1679	828588	38.33%	17.6/min	7.43	0.04
	108	160975	38783	729657	746147	758830	3169	760287	35.19%	3.8/min	5.30	0.02
TOTAL	3635	59482	1208	206939	299558	571001	426	941640	28.58%	1.2/sec	41.38	0.27

IMPLEMENTACIÓN DE ARQUITECTURA FINAL EN AZURE






No es seguro | https://lb-fz6ox4.eastus.cloudapp.azure.com/my/
Moodle
Admin User
Dashboard
Site home
Calendar
Private files
Site administration
Customise this page
Recently accessed courses
No recent courses
Timeline
No upcoming activities due
Course overview
All (except removed from view)
Course name
Card
No courses
Show All
Private files
No files available
Manage private files...
Online users
1 online user (last 5 minutes)
Admin User


No es seguro | https://lb-fz6ox4.eastus.cloudapp.azure.com/course/view.php?id=2
Moodle
Admin User
prueba1
Participants
Badges
Competencies
Grades
General
Topic 1
Topic 2
Topic 3
Topic 4
Topic 5
Topic 6
Topic 7
Topic 8
curso de prueba1
Dashboard / Courses / prueba1
Small files
Forum
Topic 1
Assignment 12
Assignment 45
Page 10
Page 12
Page 38
Topic 2
Assignment 51