



1859

UNL

Universidad
Nacional
de Loja

Universidad Nacional de Loja

**Facultad de la Energía, las Industrias y los de Recursos Naturales
No Renovables**

Maestría en Ingeniería en Software

**Análisis de plataformas de computación en la nube para el
desarrollo de aplicaciones web bajo una arquitectura cliente-
servidor**

**Trabajo de Titulación previo a la
obtención del título de Magíster en
Ingeniería en Software**

AUTOR:

Pedro Fernando Aponte Rueda

DIRECTOR:

Ing. Cristian Ramiro Narváez Guillén, Mg. Sc.

Loja - Ecuador
2023

Certificación

Loja, 30 de abril de 2023

Ing. Cristian Ramiro Narváez Guillén, Mg.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo proceso de la elaboración del Trabajo de Titulación denominado: **Análisis de plataformas de computación en la nube para el desarrollo de aplicaciones web bajo una arquitectura cliente-servidor**, previo a la obtención del título de **Magíster en Ingeniería en Software**, de autoría del estudiante **Pedro Fernando Aponte Rueda**, con cédula de identidad Nro. **1103983498**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja para el efecto, autorizo la presentación para la respectiva sustentación y defensa.

Ing. Cristian Ramiro Narváez Guillén, Mg.Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Pedro Fernando Aponte Rueda**, declaro ser autor del Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación del Trabajo de Titulación en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

Cédula de Identidad: 1103983498

Fecha: 08/05/2023

Correo electrónico: pedro.aponte@unl.edu.ec

Teléfono: 593991419277

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica de texto completo, del Trabajo de Titulación

Yo, **Pedro Fernando Aponte Rueda**, declaro ser autor del Trabajo de Titulación denominado: **Análisis de plataformas de computación en la nube para el desarrollo de aplicaciones web bajo una arquitectura cliente-servidor** como requisito para optar el título de **Magíster en Ingeniería en Software**, autorizo al sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos muestre la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los ocho días del mes de mayo de dos mil veintitrés.

Firma:

Autor: Pedro Fernando Aponte Rueda

Cédula de identidad: 1103983498

Dirección: Avda. Chuquiribamba y Nazareth

Correo electrónico: pedro.aponte@unl.edu.ec

Teléfono: 593991419277

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Cristian Ramiro Narváez Guillén Mg. Sc.

Dedicatoria

Este Trabajo de Titulación está dedicado con amor y gratitud a mi esposa y a mi hijo. A pesar de los días en los que no he podido estar a su lado, su apoyo incondicional me ha permitido alcanzar esta nueva meta en beneficio de nuestra familia.

Pedro Fernando Aponte Rueda

Agradecimiento

Expreso mi agradecimiento a la Universidad Nacional de Loja y a su destacado cuerpo docente, quienes, gracias a sus vastos conocimientos, me han proporcionado valiosas herramientas que han enriquecido mi perfil profesional e intelectual.

Pedro Fernando Aponte Rueda

Índice de contenidos

Portada.....	i
Certificación	ii
Autoría	iii
Carta de autorización	iv
Dedicatoria.....	v
Agradecimiento	vi
Índice de contenidos.....	vii
Índice de tablas:	ix
Índice de figuras:.....	x
Índice de anexos:	xi
1. Título	1
2. Resumen.....	2
2.1. Abstract.....	3
3. Introducción	4
4. Marco teórico	5
4.1. Computación en la nube.....	5
4.1.1. Tipos de implementación en la nube.....	5
4.1.2. Modelos de implementación en la nube.....	6
4.1.3. Amazon Web Services (AWS)	8
4.1.4. Microsoft Azure	9
4.1.5. Google Cloud Platform	10
4.1.6. Despliegue de aplicaciones web en la nube pública	11
4.1.7. Herramientas de despliegue	13
4.1.8. ISO/IEC 25010.....	16
5. Metodología	18
5.1. Área de estudio	18
5.2. Procedimiento	18
5.2.1. Población y muestra	18
5.2.2. Método de investigación	19
5.2.3. Variables de investigación	19
5.2.4. Técnicas de investigación	22
6. Resultados.....	23
6.1. Detección de plataformas líderes	23

6.2. Aplicación KEA.....	25
6.3. Despliegue de aplicación KEA	27
6.3.1. Despliegue de aplicación en AWS Elastic Beanstalk	27
6.3.2. Despliegue de aplicación en Azure App Service	27
6.3.3. Despliegue de aplicación en GCP App Engine.....	27
6.4. Comparación de resultados de despliegue.	28
6.4.1. Resultados en AWS Elastic Beanstalk.....	28
6.4.2. Resultados en Azure App Service.....	30
6.4.3. Resultados en GCP App Engine	32
7. Discusión	36
8. Conclusiones	37
9. Recomendaciones	38
10. Bibliografía	39
11. Anexos	42

Índice de tablas:

TABLA I. VARIABLES DE INVESTIGACIÓN	20
TABLA II. COSTO INVERSIÓN INICIAL MODELO ON-PREMISE	26
TABLA III. COSTO MENSUAL MODELO ON-PREMISE.....	26
TABLA IV. SERVICIOS DESPLEGADOS EN ELASTIC BEANSTALK DE AWS.....	27
TABLA V. SERVICIOS DESPLEGADOS EN APP SERVICE DE AZURE.....	27
TABLA VI. SERVICIOS DESPLEGADOS EN APP ENGINE DE GCP.....	27
TABLA VII. RESULTADO Y CALIFICACIÓN RENDIMIENTO DE ELASTIC BEANSTALK ..	28
TABLA VIII. INDICADOR DE RENDIMIENTO DE ELASTIC BEANSTALK	28
TABLA IX. RESULTADO Y CALIFICACIÓN DE SEGURIDAD DE ELASTIC BEANSTALK ..	29
TABLA X. INDICADOR DE SEGURIDAD EN ELASTIC BEANSTALK.....	29
TABLA XI. INDICADOR DE COMPATIBILIDAD DE ELASTIC BEANSTALK.....	29
TABLA XII. RESULTADO Y CALIFICACIÓN DE RENDIMIENTO DE APP SERVICE.....	30
TABLA XIII. INDICADOR DE RENDIMIENTO DE APP SERVICE.....	30
TABLA XIV. RESULTADO Y CALIFICACIÓN DE SEGURIDAD DE APP SERVICE	31
TABLA XV. INDICADOR DE SEGURIDAD DE APP SERVICE	31
TABLA XVI. INDICADOR DE COMPATIBILIDAD DE APP SERVICE	31
TABLA XVII. RESULTADO Y CALIFICACIÓN DE RENDIMIENTO DE APP ENGINE.....	32
TABLA XVIII. INDICADOR DE RENDIMIENTO DE APP ENGINE.....	32
TABLA XIX. RESULTADO Y CALIFICACIÓN DE SEGURIDAD DE APP ENGINE	33
TABLA XX. INDICADOR SEGURIDAD APP ENGINE	33
TABLA XXI. INDICADOR COMPATIBILIDAD APP ENGINE.....	33
TABLA XXII. CUADRO COMPARATIVO DE INDICADORES.....	34
TABLA XXIII. EFICIENCIA RENDIMIENTO.....	42
TABLA XXIV. RANGO VALORACIÓN DE EFICIENCIA RENDIMIENTO	42
TABLA XXV. CARACTERÍSTICAS DE EVALUACIÓN DE LA SEGURIDAD	43
TABLA XXVI. RANGO DE VALORACIÓN DE SEGURIDAD	43
TABLA XXVII. CARACTERÍSTICAS DE COMPATIBILIDAD	43
TABLA XXVIII. TRABAJOS RELACIONADOS CON EL TT	44
TABLA XXIX. PLATAFORMAS ESTUDIADAS EN ARTÍCULOS RELACIONADOS.....	47

Índice de figuras:

Fig. 1. Consola de Administración Amazon Web Service	8
Fig. 2. Consola de Administración de Microsoft Azure	9
Fig. 3. Consola de Administración de Google Cloud Plataform	10
Fig. 4. Características calidad de software	16
Fig. 5. Cuadrante de Gartner IaaS y PaaS.....	24
Fig. 6. Arquitectura actual de la aplicación KEA	25
Fig. 7. Inicio de sesión en la aplicación KEA.....	26
Fig. 8. Consulta de Facturas en la aplicación KEA	26
Fig. 9. Creación ambiente en Elastic Beanstalk.....	48
Fig. 10. Configuración de puerto backend en AWS	48
Fig. 11. Configuración RDS en AWS.....	49
Fig. 12. Creación del entorno en AWS para backend.....	49
Fig. 13. Carga del backend en AWS.....	50
Fig. 14. Configuración de properties para backend en AWS	50
Fig. 15. Desbloqueo del acceso público en Bucket de AWS.....	50
Fig. 16. Subida de archivos y permisos en Bucket S3 de AWS	51
Fig. 17. Creación de recurso de base de datos MySQL en App Service	52
Fig. 18. Creación App Service Web	52
Fig. 19. Centro de implementación de App Web Service Azure.....	53
Fig. 20. Configuración de GitHub con CI/CD en App Web Service de Azure	53
Fig. 21. Creación de App Web Service para frontend	54
Fig. 22. Ejecución de pipeline de App Service de Azure en Github Actions	54
Fig. 23. Consola de App Engine de GCP	55
Fig. 24. Configuración MySQL GCP	55
Fig. 25. Crear base de datos kea en instancia GCP.....	56
Fig. 26. Instalación SDK Google Cloud	56
Fig. 27. Configuración archivo properties con información GCP	56
Fig. 28. Configuración gcloudignore para deploy GCP.....	57
Fig. 29. Configuración app.yaml para deploy GCP.....	57
Fig. 30. Despliegue listo de backend en GCP.....	57
Fig. 31. Configuración archivo app.yaml para deploy frontend GCP	58
Fig. 32. Deploy frontend en GCP	58

Índice de anexos:

Anexo 1. Documento de seguimiento para servicios de computación en la nube.	42
Anexo 2. Trabajos relacionados.....	44
Anexo 3. Manual de despliegue aplicación cliente-servidor en AWS Elastic Beanstalk	48
Anexo 4. Manual de despliegue de aplicación cliente-servidor en App Service de Azure	52
Anexo 5. Manual de despliegue de aplicación cliente-servidor en Cloud Engine de GCP	55
Anexo 6. Certificado de traducción al idioma inglés.....	59

1. Título

Análisis de plataformas de computación en la nube para el desarrollo de aplicaciones web bajo una arquitectura cliente-servidor.

2. Resumen

La popularidad de los servicios en la nube ha impulsado a las organizaciones a trasladar su estructura tecnológica hacia la nube. El propósito de este Trabajo de Titulación (TT) es analizar las plataformas de los proveedores principales de servicios en la nube, identificando sus ventajas y desventajas, para proporcionar información valiosa a los desarrolladores desde un enfoque práctico, permitiéndoles tomar decisiones informadas al desplegar sus aplicaciones web con una arquitectura cliente-servidor.

El presente TT se desarrolló en tres fases, en la primera fase a través de la revisión de la literatura, se identificaron los principales proveedores de servicios en la nube. Estas plataformas se escogieron en base a los artículos investigados y en especial del artículo publicado por Gartner Inc. denominado "Magic Quadrant for Cloud Infrastructure and Platform Services" emitido el 19 de octubre del 2022, que presenta una lista de los principales proveedores de servicios en la nube. Amazon Web Services lidera la lista, seguido de Microsoft Azure y finalmente Google Cloud Platform.

Posteriormente, en la segunda fase se desplegó en los tres proveedores de servicios en la nube mencionados una aplicación web de facturación electrónica, desarrollada en Java con Spring Boot y Angular, integrada a una base de datos MySQL. Luego, se compararon sus servicios de PaaS, basándose en el rendimiento, seguridad y compatibilidad, según la normativa ISO/IEC 25010.

Finalmente, en la tercera fase, se realizó la comparativa de las plataformas sujetas a análisis, para determinar las ventajas y desventajas. Elastic BeanStalk destaca en rendimiento, escalado y costos proporcionales, pero carece de conexión directa a repositorios de código y no posee protocolo HTTPS por defecto. App Service resalta en comportamiento temporal, opciones de suscripción y seguridad, con conexión directa a repositorios de código, pero presenta ancho de banda limitado en instancias básicas. App Engine tiene destacada seguridad y firewall sin costos adicionales, pero limitaciones en tamaño de solicitudes y falta de conexión a repositorios.

***Palabras claves:** Plataforma en la nube, PaaS, Elastic Beanstalk, App Service, App Engine.*

2.1. Abstract

The popularity of cloud services has driven organizations to move their technological infrastructure to the cloud. The purpose of this Graduation Project (GP) is to analyze the platforms of the leading cloud service providers, identifying their advantages and disadvantages, to provide valuable information to developers from a practical perspective, allowing them to make informed decisions when deploying their web applications with a client-server architecture.

This GP was developed in three phases. In the first phase, through a literature review, the main cloud service providers were identified. These platforms were chosen based on the researched articles and especially the article published by Gartner Inc. called "Magic Quadrant for Cloud Infrastructure and Platform Services" issued on October 19, 2022, which presents a list of the leading cloud service providers. Amazon Web Services leads the list, followed by Microsoft Azure and finally Google Cloud Platform.

Subsequently, in the second phase, an electronic billing web application, developed in Java with Spring Boot and Angular, integrated into a MySQL database, was deployed on the three mentioned cloud service providers. Then, their PaaS services were compared based on performance, security, and compatibility, according to the ISO/IEC 25010 standard.

Finally, in the third phase, a comparison of the platforms under analysis was made to determine their advantages and disadvantages. Elastic Beanstalk stands out in performance, scaling, and proportional costs, but lacks direct connection to code repositories and does not have HTTPS protocol by default. App Service excels in temporal behavior, subscription options, and security, with direct connection to code repositories, but has limited bandwidth in basic instances. App Engine has outstanding security and firewall without additional costs, but limitations in request size and lack of connection to repositories.

Keywords: *Cloud platform, PaaS, Elastic Beanstalk, App Service, App Engine.*

3. Introducción

La demanda de soluciones de alojamiento en línea se ha incrementado en los últimos años, especialmente por parte de empresas y organizaciones. La razón principal de esto es la necesidad de contar con acceso ininterrumpido y confiable a aplicaciones y servicios, lo cual no siempre es posible en una red local. Actualmente, hay muchas opciones disponibles para la computación en la nube, que incluyen proveedores de plataformas, infraestructuras y servicios en línea. Esta diversidad puede hacer que la elección de la plataforma adecuada sea complicada, ya que no todas las opciones se adaptan a cada necesidad empresarial. Sin embargo, el despliegue de plataformas de nube para el alojamiento de servicios ha aumentado en popularidad, ya que permite el acceso desde cualquier dispositivo con conexión a internet, lo que mejora la accesibilidad y disminuye los costos de infraestructura. Existen varios tipos de nubes, como las públicas, privadas, híbridas y comunitarias, que ofrecen diferentes niveles de servicio y seguridad. Para los desarrolladores y empresas de desarrollo, la elección de la plataforma adecuada para crear aplicaciones web puede ser un desafío, ya que deben tener en cuenta factores como la escalabilidad, la seguridad y la disponibilidad del servicio.

La aplicación de facturación electrónica KEA experimenta interrupciones en su servicio debido a diversas causas, como fallas en el suministro eléctrico y proveedores de servicios de Internet, ya que está alojada en un servidor local que carece de un entorno de desarrollo y pruebas adecuado. Es esencial encontrar una solución en la nube que permita desplegar la aplicación en un entorno de producción mientras se utiliza el servidor local como entorno de desarrollo y pruebas. Por lo tanto, el TT se centra en analizar las ventajas y desventajas de las tres principales plataformas de la nube, para implementar aplicaciones web basadas en la arquitectura cliente-servidor.

La investigación se centra en la exigencia de trasladar las aplicaciones web con arquitectura cliente-servidor a la nube, lo que permite aprovechar la infraestructura ofrecida por los proveedores de computación en la nube. El resultado de esta investigación proporcionará información valiosa a los desarrolladores de software, que podrán utilizarla como referencia para evaluar las ventajas y desventajas de las diferentes plataformas de servicios en la nube disponibles. Actualmente, las empresas que tienen aplicaciones alojadas en servidores locales están migrando a plataformas en la nube debido a la alta disponibilidad de servicios y los bajos costos que ofrecen.

4. Marco teórico

4.1. Computación en la nube

Es un modelo de prestación de servicios de TI que permite a los usuarios acceder a recursos informáticos compartidos y escalables por medio de Internet, en lugar de tener que mantener y administrar infraestructuras de TI localmente [1]. Este modelo de servicios facilita el acceso a una variedad de recursos, como almacenamiento de datos, potencia de procesamiento y aplicaciones, en función de las necesidades del usuario y permite a las organizaciones reducir costos y aumentar la eficiencia.

Las tecnologías en la nube están en constante evolución, mejorando sus servicios y generando nuevas herramientas para que las personas o empresas puedan adoptarlas de manera fácil y a costos accesibles.

4.1.1. Tipos de implementación en la nube

Puede incluir el uso de nubes públicas, privadas o híbridas, dependiendo de la propiedad de la aplicación informática y de los requisitos específicos de la empresa.

Nube privada

Una nube privada es una infraestructura de nube diseñada exclusivamente para ser utilizada por una única organización. A diferencia de las nubes públicas, las nubes privadas ofrecen un mayor control y seguridad, ya que los recursos y servicios se mantienen dentro de la organización [2].

Nube pública

En una nube pública, los servicios y recursos de TI son proporcionados por un proveedor externo y están disponibles para cualquier cliente a través de Internet. Estas nubes se basan en la compartición de recursos y ofrecen escalabilidad y elasticidad [2].

La nube pública permite a las organizaciones acceder a recursos de computación virtualizados a través de Internet, proporcionados por proveedores de servicios en la nube como Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP). Este modelo de despliegue ofrece ventajas significativas, como escalabilidad, flexibilidad y costos reducidos, ya que los usuarios solo pagan por los recursos que

consumen y no necesitan invertir en infraestructura física. Sin embargo, también se plantean preocupaciones de seguridad y privacidad, ya que los datos y aplicaciones de los usuarios se almacenan y procesan en infraestructuras compartidas por múltiples inquilinos [3].

Nube híbrida

La nube híbrida combina características de las nubes públicas y privadas. Este tipo de nube permite a las organizaciones aprovechar las ventajas de ambas opciones, utilizando la nube pública para recursos y servicios que no requieren una alta seguridad y la nube privada para aquellos recursos y servicios que necesitan un mayor nivel de control y seguridad [2].

Nube comunitaria

La nube comunitaria es un modelo en el que varios usuarios u organizaciones con intereses comunes comparten recursos y servicios de la nube, ofreciendo una solución intermedia entre la nube pública y la nube privada. Este modelo permite a las organizaciones participantes aprovechar los beneficios de la nube, como la escalabilidad y la reducción de costos, al mismo tiempo que se garantiza un mayor control y personalización de los servicios. La nube comunitaria también puede abordar las preocupaciones específicas de seguridad, privacidad y cumplimiento normativo que enfrentan las organizaciones con intereses comunes, proporcionando una infraestructura compartida diseñada para satisfacer sus necesidades particulares [4].

4.1.2. Modelos de implementación en la nube

Los proveedores de la nube están capacitados para manejar múltiples clientes y un aumento repentino en la demanda de recursos. Los diferentes modelos de servicio de la nube incluyen infraestructura, plataforma, software y backend como servicio, cada uno con sus propias características y ventajas para la gestión de recursos informáticos [5].

Serverless

En este modelo no se tiene la necesidad de administrar infraestructura de servidor. En lugar de preocuparse por la administración de servidores, los desarrolladores se enfocan en escribir código para crear una lógica de negocio. Los proveedores de serverless

ofrecen una plataforma de ejecución que puede escalar automáticamente para manejar la carga de trabajo. Esto permite una mayor eficiencia en el uso de recursos y una reducción en los costos operativos. Los principales desafíos de la tecnología serverless incluyen la gestión de datos y el aseguramiento de la calidad del software [6].

En el contexto de la computación en la nube, las arquitecturas sin servidor (serverless) han surgido como una evolución que permite a los desarrolladores centrarse en la funcionalidad de sus aplicaciones sin preocuparse por la administración de la infraestructura subyacente. Las ventajas de los modelos sin servidor incluyen la escalabilidad automática, la reducción de costos operativos debido al pago por uso y la menor responsabilidad en la gestión y mantenimiento de la infraestructura. No obstante, también se deben considerar algunas desventajas asociadas con este enfoque. Entre ellas, la complejidad en la monitorización y depuración de aplicaciones puede aumentar, el tiempo de arranque en frío de las funciones puede ser mayor y existe el riesgo de bloqueo del proveedor debido a la dependencia de servicios específicos de la nube [7].

Software como servicio (SaaS)

SaaS se refiere a la provisión de aplicaciones a través de Internet, en lugar de que los usuarios instalen y administren software localmente. Los proveedores de SaaS se encargan de mantener y actualizar las aplicaciones, y los usuarios acceden a ellas mediante navegadores web [8]. Sin embargo, existe la opción de que el cliente administre ciertos ajustes de configuración específicos de la aplicación [9].

Infraestructura como servicio (IaaS)

Proporciona recursos de computación virtualizados en la nube, incluyendo servidores, almacenamiento y redes. Los usuarios pueden aprovisionar y escalar recursos según sus necesidades, lo que permite mayor flexibilidad y control sobre la infraestructura [8].

Plataforma como servicio (PaaS)

Son una categoría de servicios en la nube que permiten a los desarrolladores crear, probar, implementar y administrar aplicaciones sin tener que preocuparse por la infraestructura subyacente. Al analizar las plataformas AWS, Azure y GCP, se puede ver que cada proveedor ofrece una amplia gama de servicios PaaS, que incluyen bases de datos,

herramientas de desarrollo, análisis y mucho más. AWS cuenta con servicios de PaaS como AWS Elastic Beanstalk, AWS Amplify y AWS App Runner, mientras que Azure tiene servicios como Azure App Service, Azure Functions y Azure Logic Apps, y GCP tiene servicios como Google App Engine, Cloud Functions y Cloud Run. Cada plataforma PaaS tiene sus propias características y ventajas, por lo que es importante que los desarrolladores elijan la que mejor se adapte a sus necesidades., según una investigación realizada por K. Lee [10].

4.1.3. Amazon Web Services (AWS)

Amazon Web Services (AWS) es una plataforma de servicios en la nube que ofrece una amplia variedad de soluciones informáticas, incluyendo almacenamiento, bases de datos, inteligencia artificial y análisis de datos. AWS permite a las organizaciones implementar aplicaciones y servicios rápidamente, reduciendo la carga de la gestión de la infraestructura de TI y permitiendo la escalabilidad y flexibilidad de los recursos en función de las demandas de la empresa.

Uno de los principales beneficios de AWS es su conjunto de servicios integrados que permiten a los desarrolladores construir, desplegar y mantener aplicaciones en la nube de manera eficiente y segura. Además, AWS ofrece una serie de características para facilitar la adopción de prácticas de DevOps, incluyendo la automatización de procesos y la integración y entrega continuas. En la Fig. 1 se puede observar la consola de administración con los servicios más utilizados [11].

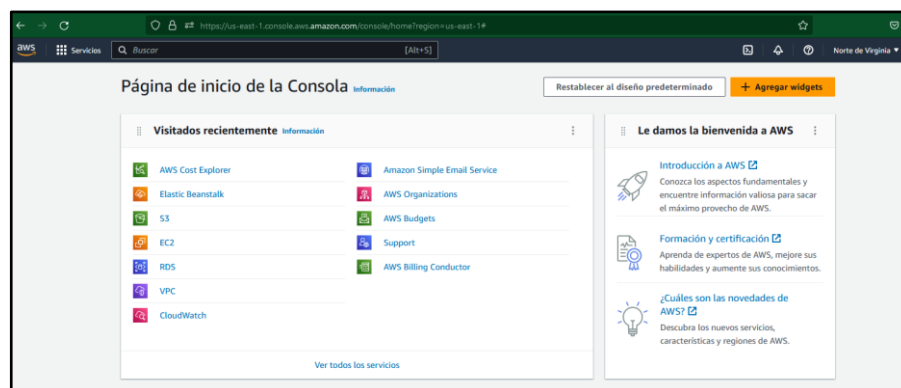


Fig. 1. Consola de Administración Amazon Web Service

La consola de AWS es una interfaz gráfica de usuario que permite a los usuarios administrar y monitorear una amplia gama de servicios y recursos en la nube de Amazon

Web Services. A través de la consola de AWS, los usuarios pueden crear, configurar y gestionar instancias de Amazon EC2, almacenamiento en Amazon S3, bases de datos en Amazon RDS, funciones sin servidor en AWS Lambda, y muchos otros servicios como seguridad, análisis de datos, aprendizaje automático e IoT. Además, la consola de AWS ofrece herramientas para la visualización de métricas de rendimiento, configuración de alarmas y notificaciones, y la administración de usuarios y políticas de acceso a través de AWS Identity and Access Management (IAM).

4.1.4. Microsoft Azure

Microsoft Azure es una plataforma de servicios en la nube líder que ofrece una amplia gama de servicios y herramientas de infraestructura, plataforma y software como servicio (IaaS, PaaS y SaaS). Los servicios de Azure son altamente escalables y se pueden personalizar para satisfacer las necesidades específicas de los usuarios, lo que la hace atractiva para empresas de todos los tamaños. Además, Azure tiene una gran cantidad de características de seguridad integradas para ayudar a proteger los datos y aplicaciones de los clientes. La plataforma también cuenta con herramientas de análisis y aprendizaje automático. Azure es compatible con una amplia gama de lenguajes de programación y marcos, lo que permite a los desarrolladores trabajar en el entorno que mejor se adapte a sus necesidades. La Fig. 2 muestra la consola de administración de la plataforma [12].

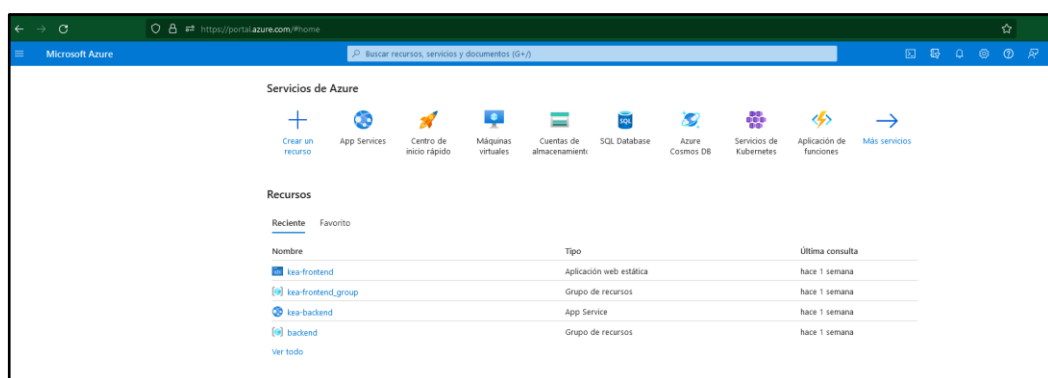


Fig. 2. Consola de Administración de Microsoft Azure

La consola de Azure, también conocida como Azure Portal, es una interfaz gráfica de usuario que permite a los usuarios administrar y monitorear una amplia gama de servicios y recursos en Microsoft Azure. Azure Portal proporciona herramientas para visualizar métricas de rendimiento, configurar alarmas y notificaciones, así como administrar usuarios, roles y políticas de acceso mediante Azure Active Directory.

4.1.5. Google Cloud Platform

GCP es altamente escalable y flexible, lo que la hace ideal para empresas de todos los tamaños. La plataforma es compatible con una gran cantidad de lenguajes de programación y marcos, lo que permite a los desarrolladores trabajar en el entorno que mejor se adapte a sus necesidades. Además, GCP cuenta con herramientas de análisis y aprendizaje automático avanzadas, lo que permite a los clientes procesar grandes cantidades de datos y obtener información valiosa. La plataforma también tiene un alto nivel de seguridad integrado, lo que permite a los usuarios proteger sus datos y aplicaciones. La Fig. 3 muestra la consola principal de administración de GCP [13].

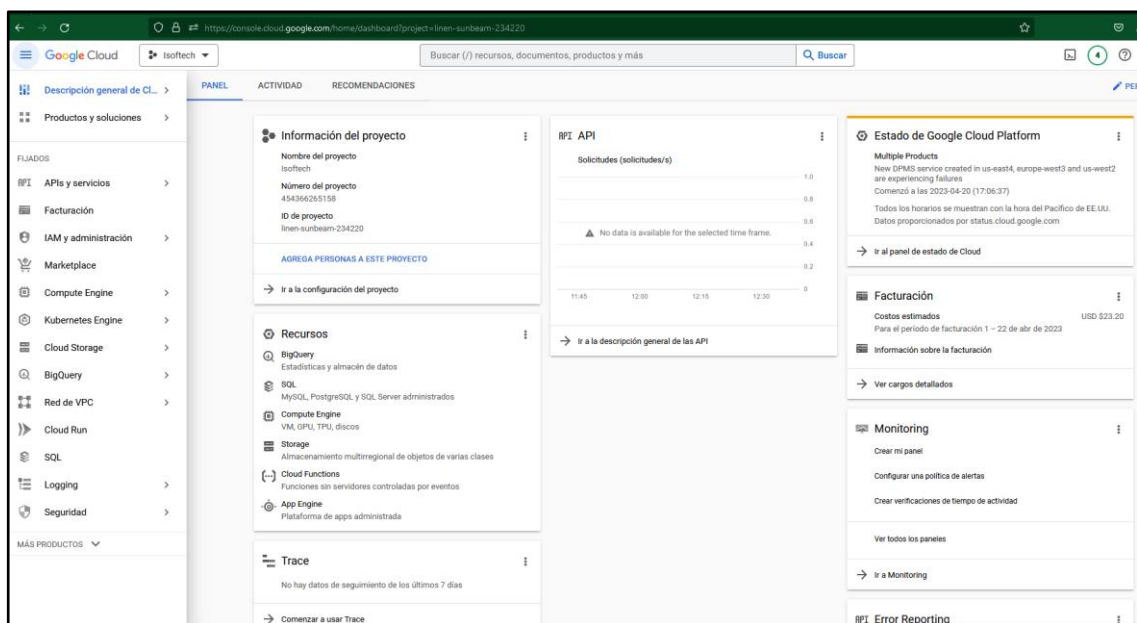


Fig. 3. Consola de Administración de Google Cloud Plataforma

La consola de GCP, conocida como Google Cloud Console, es una interfaz gráfica de usuario que permite a los usuarios administrar y monitorear una amplia gama de servicios y recursos en la nube de Google Cloud Platform. A través de la Google Cloud Console, los usuarios pueden crear, configurar y gestionar máquinas virtuales en Compute Engine, servicios de aplicaciones en App Engine, bases de datos en Cloud SQL y Cloud Spanner, almacenamiento en Cloud Storage, redes y muchos otros servicios, como análisis de datos, aprendizaje automático, IoT y seguridad. Además, la Google Cloud Console proporciona herramientas para visualizar métricas de rendimiento, configurar alertas y notificaciones, así como administrar usuarios, roles y políticas de acceso mediante Cloud Identity and Access Management (IAM).

4.1.6. Despliegue de aplicaciones web en la nube pública

La nube pública proporciona una plataforma eficiente y accesible para desarrolladores y empresas que buscan implementar sus aplicaciones en entornos altamente disponibles y con recursos fácilmente ajustables según las necesidades del negocio. Además, la nube pública permite a las organizaciones ahorrar en costos de infraestructura y mantenimiento, ya que los proveedores de servicios en la nube se encargan de la administración y actualización de los recursos necesarios para mantener las aplicaciones en funcionamiento [14].

Al mismo tiempo, el despliegue de aplicaciones web en la nube pública presenta desafíos en términos de seguridad y privacidad, ya que los datos y aplicaciones están alojados en servidores compartidos por múltiples usuarios. Sin embargo, los principales proveedores de servicios en la nube han implementado medidas de seguridad sólidas y cumplen con estándares internacionales para proteger la información y garantizar la privacidad de los datos [15].

El despliegue de aplicaciones web en la nube pública también puede facilitar la colaboración y la integración entre equipos de desarrollo, operaciones y administración. Los proveedores de servicios en la nube ofrecen herramientas y servicios integrados que simplifican la implementación de aplicaciones, la gestión de infraestructura y la monitorización del rendimiento. Estas soluciones permiten a los equipos de desarrollo centrarse en la creación de aplicaciones de alta calidad y funcionales, mientras que los aspectos relacionados con la infraestructura y la escalabilidad son manejados automáticamente por la plataforma en la nube. Además, la adopción de enfoques de desarrollo modernos, como DevOps, se ve facilitada por las capacidades y servicios que proporcionan las plataformas de nube pública [16].

En este tipo de nube los proveedores más destacados son Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), Alibaba Cloud, Oracle Cloud e IBM Cloud.

Arquitectura

La arquitectura define cómo se estructura una aplicación web, incluyendo la división en componentes y la interacción entre ellos. Una arquitectura adecuada para el desarrollo de aplicaciones web debe ser escalable, flexible y segura. Algunas de las arquitecturas más populares para el desarrollo de aplicaciones web son cliente-servidor, microservicios y sin servidor.

La arquitectura cliente-servidor divide a la aplicación en dos partes: el cliente y el servidor. El cliente es la parte que se ejecuta en el navegador del usuario y se encarga de la interfaz de usuario, mientras que el servidor es la parte que se ejecuta en el servidor y se encarga del procesamiento de datos y la lógica de negocio. La arquitectura cliente-servidor es escalable y puede manejar grandes volúmenes de usuarios.

La arquitectura sin servidor es una arquitectura emergente que se está convirtiendo en una opción popular para desarrollar aplicaciones. Esta arquitectura se enfoca en la creación de funciones independientes que se ejecutan en los servicios de la nube, en lugar de en servidores dedicados. La arquitectura sin servidor ofrece una gran flexibilidad y escalabilidad, ya que las funciones pueden ajustarse automáticamente a la demanda de los usuarios. Además, la arquitectura sin servidor permite a los desarrolladores enfocarse en la lógica de negocio en lugar de preocuparse por la infraestructura, lo que reduce los costos operativos [17].

Arquitectura sin servidor

La arquitectura sin servidor permite que las aplicaciones se escalen automáticamente a medida que aumenta la demanda de los usuarios, lo que la hace ideal para aplicaciones con picos de uso. Además, la arquitectura sin servidor puede reducir los costos operativos al eliminar la necesidad de mantener servidores dedicados. A medida que la demanda de aplicaciones escalables y flexibles aumenta, se espera que la arquitectura sin servidor se convierta en una opción aún más popular para el desarrollo de aplicaciones en la nube [18].

4.1.7. Herramientas de despliegue

AWS Elastic Beanstalk

Elastic Beanstalk proporciona una plataforma automatizada para la implementación, administración y escalabilidad de aplicaciones, abstrayendo la infraestructura subyacente y permitiendo a los desarrolladores centrarse en el código de la aplicación en lugar de la administración de la infraestructura. Este servicio es compatible con una amplia gama de lenguajes de programación y marcos de desarrollo, lo que permite a los desarrolladores elegir las tecnologías que mejor se adapten a sus necesidades [19].

Amazon Elastic Compute Cloud (EC2)

Es un servicio de infraestructura en la nube que proporciona a los usuarios capacidad de cómputo escalable y a pedido, clave para el funcionamiento eficiente de aplicaciones y servicios en línea. EC2 permite a organizaciones y desarrolladores implementar entornos virtuales, conocidos como instancias EC2, con una amplia selección de sistemas operativos y configuraciones de hardware, lo que posibilita la adaptación de la infraestructura de tecnología de la información a requisitos específicos. Además, EC2 ofrece características avanzadas como balanceo de carga, Elastic Block Store y administración de sistemas, lo cual facilita el desarrollo y escalabilidad de aplicaciones y servicios en la nube [20].

Amazon Simple Storage Service (S3)

Es un servicio de almacenamiento en la nube altamente escalable y duradero que facilita a los usuarios el almacenamiento y recuperación de grandes cantidades de datos. S3 ofrece una infraestructura segura y de bajo costo, permitiendo a organizaciones y desarrolladores almacenar, proteger y gestionar información de manera eficiente. Además, S3 proporciona características avanzadas como control de acceso granular, replicación entre regiones y versionado de objetos, lo cual mejora la gestión y recuperación de datos en caso de pérdida o corrupción accidental [21].

Amazon Relational Database Service (RDS)

Es un servicio de base de datos relacional en la nube que simplifica la configuración, operación y escalabilidad de bases de datos para los usuarios. RDS permite a

organizaciones y desarrolladores implementar bases de datos relacionales en los servicios de la nube de manera rápida y eficiente, reduciendo la complejidad asociada a la administración de dichas bases de datos. Además, RDS ofrece características avanzadas como copias de seguridad automáticas, monitoreo de rendimiento y replicación de datos para mejorar la confiabilidad y disponibilidad de la información [22].

Azure App Service

Azure App Service es un servicio de plataforma como servicio (PaaS) en la nube que permite a los desarrolladores y organizaciones crear, implementar y gestionar aplicaciones web y móviles de manera rápida y eficiente. Esta plataforma proporciona herramientas y servicios integrados para simplificar el proceso de desarrollo, permitiendo a los usuarios enfocarse en la creación de soluciones de alta calidad sin preocuparse por la infraestructura subyacente. Además, Azure App Service ofrece características avanzadas como escalabilidad automática, integración con repositorios de código y compatibilidad con múltiples lenguajes y frameworks, lo que facilita la creación y expansión de aplicaciones en la nube [23].

Azure Cloud Services

Por otro lado, Azure Cloud Services es otro servicio PaaS que ofrece Microsoft Azure, diseñado para admitir aplicaciones escalables y de alta disponibilidad. A diferencia de App Service, Cloud Services proporciona más control sobre la infraestructura y un mayor nivel de personalización, lo que permite a los desarrolladores configurar y administrar aplicaciones en función de sus requisitos específicos. Esto lo hace más adecuado para aplicaciones que requieren una mayor flexibilidad y control sobre la infraestructura de la nube [24].

Azure Database for MySQL

Azure Database for MySQL es un servicio de base de datos en la nube totalmente administrado que permite a desarrolladores y organizaciones implementar y gestionar bases de datos MySQL de manera eficiente y escalable. Esta solución brinda un entorno de alto rendimiento y confiabilidad, optimizado para la construcción de aplicaciones modernas y basado en el popular sistema de gestión de bases de datos de código abierto, MySQL. Además, Azure Database for MySQL ofrece características avanzadas como

copias de seguridad automáticas, monitoreo en tiempo real y cifrado de datos en reposo y en tránsito, garantizando la seguridad y la disponibilidad de la información [25].

Google App Engine

Google App Engine es compatible con lenguajes de programación, como Java, Python, Go, PHP y Node.js, y se integra perfectamente con otros servicios de Google Cloud para proporcionar capacidades avanzadas a las aplicaciones.

Una de las principales ventajas de utilizar Google App Engine es su capacidad de escalabilidad automática. Según la demanda de los usuarios, el servicio ajusta automáticamente la cantidad de recursos asignados a una aplicación, lo que garantiza un rendimiento óptimo y reduce los costos innecesarios por recursos no utilizados.

Además, Google App Engine ofrece características de monitoreo y administración de aplicaciones, así como la capacidad de implementar versiones de aplicaciones y configuraciones específicas del entorno. Todo esto permite a los desarrolladores centrarse en la lógica y funcionalidad de sus aplicaciones, mientras que Google Cloud se encarga de la infraestructura y la escalabilidad [26].

Google Cloud SQL

Google Cloud SQL es un servicio en la nube completamente administrado que facilita la implementación, el mantenimiento y la gestión de bases de datos relacionales como MySQL, PostgreSQL y SQL Server. Este servicio permite a desarrolladores y organizaciones centrarse en la construcción de aplicaciones escalables y de alto rendimiento, al tiempo que reduce la carga de la administración y el mantenimiento de la infraestructura de la base de datos. Además, Google Cloud SQL ofrece características avanzadas, como la replicación de datos, respaldos automáticos y monitoreo en tiempo real [27].

4.1.8. ISO/IEC 25010

La norma ISO/IEC 25010 [28] es una parte de la serie de estándares ISO/IEC 25000, también conocida como "SQuaRE" (System and Software Quality Requirements and Evaluation). La norma ISO/IEC 25010 se centra en la definición de un modelo de calidad del producto de software y sus características, proporcionando un marco para evaluar y mejorar la calidad del software [29]. Este modelo de calidad consta de dos dimensiones principales: calidad en uso y calidad del producto. La calidad en uso se refiere a cómo el software satisface las necesidades de los usuarios en un contexto específico, mientras que la calidad del producto se centra en las propiedades intrínsecas del software, independientemente del contexto de uso.

El modelo de calidad del producto de software ISO/IEC 25010 incluye ocho características: funcionalidad, rendimiento, compatibilidad, usabilidad, fiabilidad, seguridad, mantenibilidad y portabilidad. Estas características se dividen en subcaracterísticas que permiten una evaluación más detallada de la calidad del software. La aplicación de este estándar ayuda a las organizaciones a identificar áreas de mejora y a desarrollar estrategias para abordar las deficiencias en la calidad del software. En la Fig. 4 se pueden observar dichas características [29].

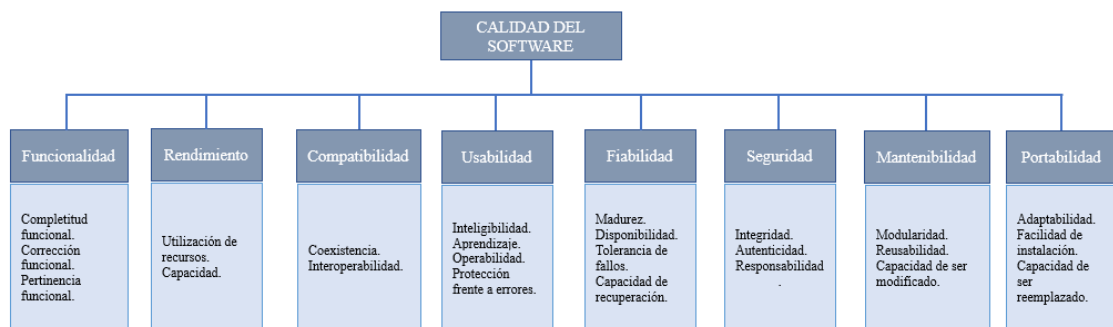


Fig. 4. Características calidad de software

- La funcionalidad se refiere a la capacidad del software para proporcionar funciones que satisfagan las necesidades y requisitos declarados e implícitos de los usuarios. Incluye aspectos como la adecuación funcional, la exactitud y la seguridad funcional.

- El rendimiento de un software se refiere a su capacidad para funcionar de manera óptima y eficiente, en base a los recursos que utiliza, la capacidad y la escalabilidad.
- La compatibilidad se refiere a la capacidad del software para interactuar adecuadamente con otros sistemas o componentes de software, y para operar en diferentes entornos sin conflictos. Incluye la coexistencia y la interoperabilidad.
- La usabilidad hace referencia a la facilidad con la que los usuarios pueden aprender, operar y controlar el software para lograr sus objetivos. La usabilidad incluye aspectos como la comprensibilidad, la facilidad de aprendizaje y la operabilidad.
- La fiabilidad incluye la madurez, la tolerancia a fallos y la recuperabilidad.
- La seguridad se refiere a la capacidad del software para proteger la información y los datos de accesos no autorizados, garantizando la confidencialidad, integridad y disponibilidad. La seguridad incluye aspectos como la confidencialidad, integridad, autenticación y no repudio.
- La mantenibilidad incluye la modularidad, la reusabilidad, la analizabilidad, la modificabilidad y la facilidad para probar.
- La portabilidad incluye la adaptabilidad, la capacidad para ser instalado y la capacidad para ser reemplazado.

5. Metodología

5.1. Área de estudio

La aplicación web fue diseñada y desarrollada para los diferentes tipos de contribuyentes que requieren de facturación electrónica en línea, actualmente cuenta con 39 clientes activos.

Para desplegar la aplicación se utilizó la plataforma AWS Elastic Beanstalk con los servicios Elastic Compute Cloud (EC2), Simple Storage Service (S3) y Relational Database Service (RDS). También se escogió la plataforma Azure App Service con los servicios Cloud Service y Database for MySQL. Y también se utilizó la plataforma GCP App Engine con los servicios Cloud SQL. En todas las plataformas se seleccionaron recursos alojados en Estados Unidos.

El estudio se llevó a cabo a lo largo del primer trimestre de 2023, inició con la recopilación y análisis de la información encontrada en los artículos estudiados. Después de esto, se procedió a desplegar la aplicación web en AWS, Azure y GCP, esto se completó a finales de marzo de 2023. El TT tuvo una duración de cuatro meses para su ejecución práctica.

5.2. Procedimiento

El presente TT se considera exploratorio, ya que su objetivo principal fue evaluar las tres principales plataformas en la nube para el despliegue de aplicaciones web cliente-servidor, utilizando tanto métodos cualitativos como cuantitativos para recopilar los datos e información necesarios. Se obtuvieron resultados en términos de rapidez, escalabilidad y almacenamiento, lo que permitió concluir las ventajas y desventajas de cada plataforma.

5.2.1. Población y muestra

Se consideraron varias empresas, incluyendo a Cloud Platform de Google, Web Services de Amazon y Azure de Microsoft, todas ellas con una posición de líderes del mercado. Se realizaron investigaciones exhaustivas sobre el rendimiento y las funcionalidades ofrecidas por estas plataformas, y se compararon con otras opciones. Este estudio permitió identificar las ventajas y desventajas de cada plataforma analizada, en base a los

requerimientos de la aplicación. Los principales beneficiarios de este proyecto de investigación son los programadores, quienes podrán utilizar los resultados obtenidos como referencia para sus proyectos relacionados. La aplicación se ha desplegado específicamente para satisfacer las necesidades de los contribuyentes que requieren el servicio de facturación electrónica, brindando un servicio continuo y sin interrupciones a los usuarios. De esta manera, se busca garantizar la eficiencia en el proceso de facturación y evitar posibles inconvenientes que puedan afectar el normal desarrollo de las actividades comerciales.

5.2.2. Método de investigación

Se empleó la técnica inductiva de investigación para determinar las ventajas y desventajas de cada plataforma analizada, basándose en los resultados de rendimiento, compatibilidad y seguridad.

También se utilizó la técnica descriptiva para examinar las características clave de las plataformas de computación en la nube, con el fin de emplear el ambiente de cada plataforma en consideración.

Además, se aplicó la técnica experimental para obtener datos de almacenamiento, velocidad, escalabilidad y compatibilidad con herramientas externas como repositorios de código.

5.2.3. Variables de investigación

Dentro de la metodología propuesta, se pueden identificar varias variables clave relacionadas con el rendimiento, compatibilidad y seguridad de las plataformas AWS, Azure y GCP según la norma ISO/IEC 25010. A continuación, se describen estas variables en la Tabla I y cómo fueron abordadas en el TT. La Tabla I presenta cada variable junto con sus indicadores y tipo de variable (cuantitativa continua o cualitativa politómica). Este enfoque estructurado y detallado permite una evaluación exhaustiva y objetiva de las plataformas de analizadas, simplificando el proceso de decisión para la selección óptima de la plataforma adecuada para la implementación de aplicaciones bajo una arquitectura cliente-servidor.

TABLA I
VARIABLES DE INVESTIGACIÓN

Variable	Indicador	Tipo de Variable
Rendimiento	Tiempo de respuesta	Cuantitativa continua
	Ancho de banda	Cuantitativa continua
	Escalabilidad	Cuantitativa continua
Compatibilidad	Interoperabilidad	Cualitativa politómica
	Coexistencia	Cualitativa politómica
Seguridad	Confidencialidad	Cualitativa politómica
	Integridad	Cualitativa politómica
	Autenticidad	Cualitativa politómica

Rendimiento

- **Tiempo de respuesta:** Es relevante ya que un tiempo de respuesta más corto mejora la experiencia del usuario y la eficiencia de las aplicaciones cliente-servidor. La selección de esta unidad se justifica porque permite comparar directamente la velocidad con la que cada plataforma procesa las solicitudes. La unidad de medida en milisegundos (ms) o segundos (s), dependiendo de la naturaleza de la tarea o solicitud.
- **Ancho de banda:** Es importante para garantizar la transferencia de datos eficiente entre el servidor y los clientes. La selección de Mbps o Gbps permite una comparación objetiva de la capacidad de transferencia de datos de cada plataforma. La unidad de medida es en Megabits por segundo (Mbps) o Gigabits por segundo (Gbps), según la capacidad de transferencia de datos.
- **Escalabilidad:** Es crucial para adaptarse a cambios en la demanda y asegurar el desempeño adecuado de la aplicación. La selección de unidades de recursos o porcentaje de crecimiento permite evaluar cómo cada plataforma maneja la variación en la carga de trabajo. La unidad de medida es en unidades de recursos (por ejemplo, número de instancias de máquinas virtuales) o porcentaje de crecimiento en la carga de trabajo.

Compatibilidad

- **Interoperabilidad:** es relevante porque las aplicaciones cliente-servidor a menudo necesitan interactuar con otros sistemas, aplicaciones o servicios. La selección del porcentaje de éxito en la integración proporciona una medida objetiva y comparable para evaluar esta característica.
- **Coexistencia:** es una característica importante de calidad del software que se refiere a la capacidad de un software para funcionar correctamente junto con otros sistemas o aplicaciones, sin generar conflictos o incompatibilidades.

Seguridad

- **Confidencialidad:** es relevante porque proteger la información y datos sensibles es fundamental en aplicaciones cliente-servidor. La selección del número de incidentes de violaciones de datos permite comparar la efectividad de las medidas de seguridad en cada plataforma.
- **Integridad:** es importante para garantizar que los datos no sean alterados o corrompidos. El porcentaje de éxito en la detección y prevención de alteraciones proporciona una medida objetiva y comparable de esta característica en las distintas plataformas.
- **Autenticidad:** es relevante verificar que la información proviene de una fuente confiable y que no ha sido alterada durante su transmisión o almacenamiento. La autenticidad puede asegurarse mediante técnicas de cifrado y firma digital, que permiten verificar la integridad y origen de la información.

5.2.4. Técnicas de investigación

Se emplearon dos métodos de investigación para recolectar datos sobre las variables examinadas en el estudio: observación científica y experimentación.

La observación científica se utilizó en las plataformas evaluadas, para examinar los servicios integrados, bases de datos, compatibilidad con repositorios de código y lenguajes de programación. La inspección se llevó a cabo entre febrero y marzo de 2023, utilizando una hoja de observación detallada en el anexo 1. Lo más importante de este estudio fueron las características evaluadas, que fueron seleccionadas según la norma internacional ISO/IEC 25010 de calidad del software.

Asimismo, se empleó un enfoque experimental para presentar resultados en términos de escalabilidad, capacidad y tiempos de respuesta, facilitando la identificación de ventajas y desventajas en las plataformas examinadas. Para ello, se utilizó la herramienta de diseño experimental detallada en el anexo 1. Entre febrero y abril de 2023, se evaluaron características de rendimiento, seguridad y compatibilidad plataformas analizadas y se concluyeron las ventajas y desventajas en función de los parámetros analizados.

Además, se empleó la metodología de estadística descriptiva para examinar los datos cualitativos y cuantitativos recolectados a lo largo del proceso investigativo. Este método permitió la consolidación de los datos de desempeño recolectados en las diversas plataformas analizadas.

6. Resultados

6.1. Detección de plataformas líderes

El artículo "Analysis of PaaS offerings from AWS, Azure, and GCP" presentado en la Conferencia Internacional de Computación en la Nube del IEEE en 2021, realiza un análisis comparativo de las plataformas como servicio (PaaS) ofrecidas por Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP). El estudio considera varios factores como características, rendimiento, funcionalidades, precios y soporte al cliente. Los resultados obtenidos permitieron identificar las ventajas y desventajas de cada plataforma y proporcionar una comparativa detallada para ayudar a las empresas a tomar una decisión informada al elegir una plataforma PaaS para sus proyectos de aplicación en la nube [10].

En el artículo titulado "Comparison of Cloud Platforms for Migrating On-Premises Web Application" de M. Malik, N. B. E. Ahmed, A. M. M. Al-Qaydi y M. A. Alghamdi, presentado en la 3ª Conferencia Internacional sobre Tecnologías de Computación, Matemáticas e Ingeniería (iCoMET) en 2019, los autores abordan la comparación de diferentes plataformas de computación en la nube para migrar aplicaciones web desde un entorno local. El objetivo de la investigación es identificar las ventajas y desventajas de cada plataforma y determinar cuál es la más adecuada para trasladar aplicaciones web on-premise a la nube. Los autores comparan y evalúan las principales plataformas de computación en la nube, incluidas Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP), en función de factores como rendimiento, escalabilidad, facilidad de uso y costos asociados. Al final, el estudio proporciona una visión detallada de las diferencias entre las plataformas de nube analizadas y ofrece recomendaciones valiosas para los profesionales y las organizaciones que buscan migrar sus aplicaciones web a un entorno en la nube.

En el artículo "A Comparative Study of Cloud PaaS Services for Web Application Deployment", presentado en la Conferencia Internacional de Sistemas Inteligentes del IEEE en 2021, se realiza un estudio comparativo de Elastic Beanstalk, App Service y App Engine, ya que son las plataformas principales para el despliegue de aplicaciones web debido a su facilidad de uso, escalabilidad y flexibilidad. El análisis incluyó factores como la calidad del soporte, la velocidad de despliegue y la variedad de opciones de lenguaje de programación y frameworks soportados. Los autores también encontraron

que Elastic Beanstalk es la mejor opción para aplicaciones basadas en contenedores, mientras que App Service es ideal para aplicaciones .NET y App Engine es adecuada para aplicaciones basadas en Python y Java [30].

Después de revisar varios artículos, se ha concluido que la mayoría de las propuestas utilizan plataformas en la nube como AWS, GCP y Microsoft Azure. Esta literatura ha sido respaldada por el artículo "Magic Quadrant for Cloud Infrastructure and Platform Services" de Gartner Inc., publicado el 19 de octubre de 2022, destaca a AWS, Azure y GCP como líderes en el mercado de computación en la nube. AWS sobresale por su innovación constante, escalabilidad y desempeño, mientras que Azure se fortalece gracias a su integración con productos de Microsoft y su enfoque híbrido de nube. Por otro lado, GCP se distingue por sus capacidades de inteligencia artificial, aprendizaje automático y sostenibilidad. Estas plataformas ofrecen soluciones robustas, innovadoras y versátiles que satisfacen las necesidades de diversas organizaciones y aplicaciones, compitiendo en un mercado en constante crecimiento y evolución, lo que resulta en una amplia variedad de opciones y servicios para los clientes. La Fig. 5 ilustra los diferentes proveedores de servicios en la nube en cada eje correspondiente.



Fig. 5. Cuadrante de Gartner IaaS y PaaS

6.2. Aplicación KEA

La aplicación web empleada en las diversas pruebas se denomina KEA, dirigida a contribuyentes que necesitan un software de facturación electrónica en línea. Esta aplicación se desarrolló utilizando la arquitectura cliente-servidor.

En el backend, se incorporó una API Rest construida con Java versión 17 y el framework Spring Boot versión 2.7.4. Por otro lado, en el frontend se incluyeron varios módulos agrupados en una única página web mediante el uso del framework Angular versión 15.1.4. La Fig. 6 presenta la arquitectura actual de la aplicación KEA con un modelo on-premise.

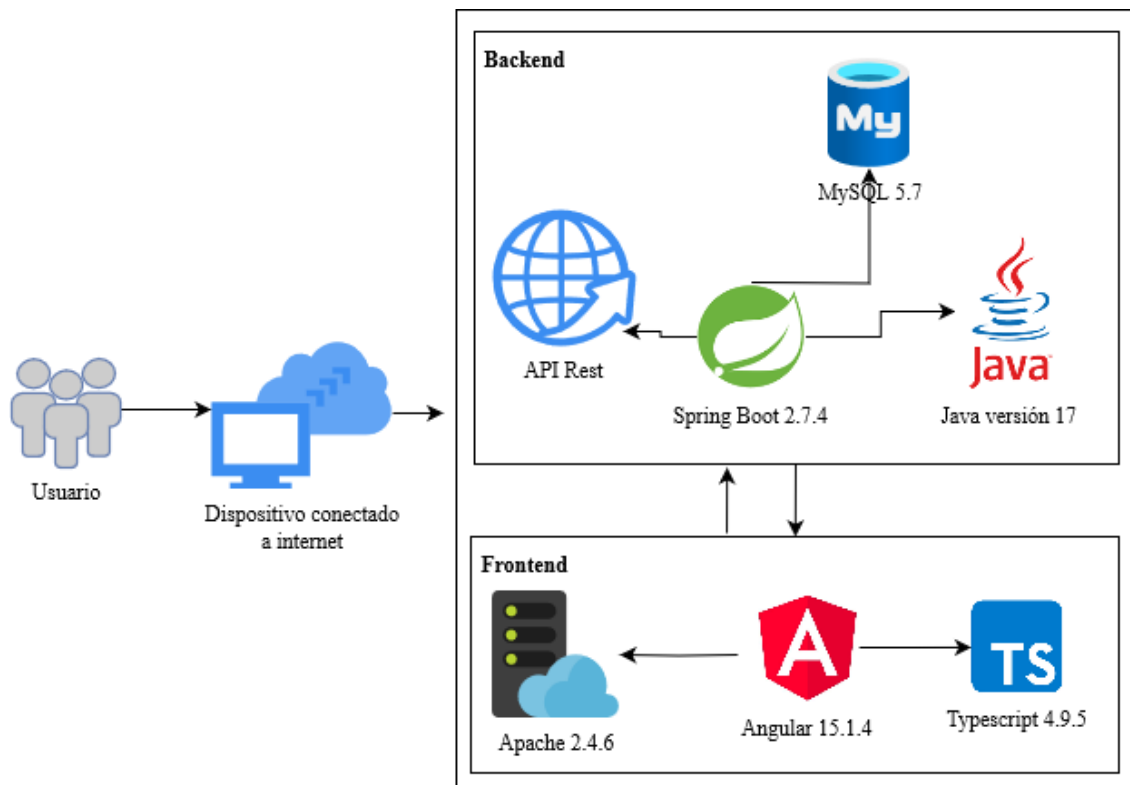


Fig. 6. Arquitectura actual de la aplicación KEA

En la Fig. 7 se puede ver la pantalla de ingreso y en la Fig. 8 un formulario de la aplicación web.

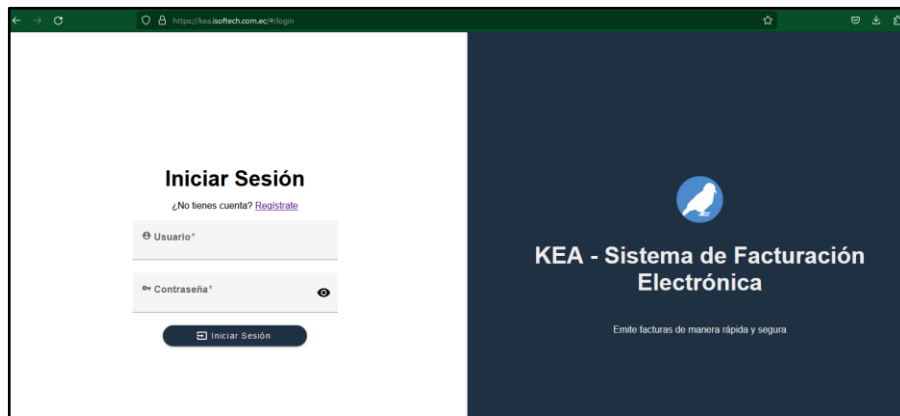


Fig. 7. Inicio de sesión en la aplicación KEA

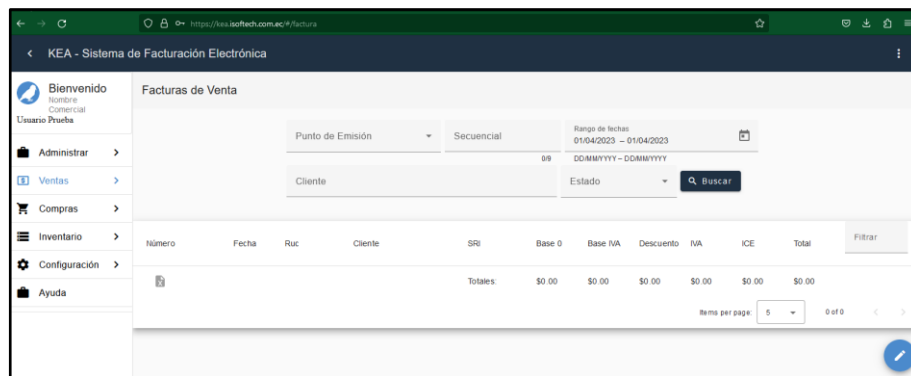


Fig. 8. Consulta de Facturas en la aplicación KEA

En la Tabla II y Tabla III se presenta los costos de inversión inicial y de mantenimiento mensual de la aplicación KEA en modelo on-premise.

TABLA II
COSTO INVERSIÓN INICIAL MODELO ON-PREMISE

Rubro	Costo
Servidor HP Proliant	\$980.00
UPS	\$180.00
Instalación	\$100.00
	\$1260.00

TABLA III
COSTO MENSUAL MODELO ON-PREMISE

Rubro	Costo
Internet 70MB Corporativo	\$55.00
Energía Eléctrica	\$7.00
Mantenimiento	\$40.00
	\$102.00

6.3. Despliegue de aplicación KEA

6.3.1. Despliegue de aplicación en AWS Elastic Beanstalk

Al desplegar la aplicación web de arquitectura cliente-servidor, se emplearon tres servicios de Elastic Beanstalk de AWS: EC2, S3 y RDS. En la Tabla IV se presenta los servicios desplegados con la respectiva URL.

TABLA IV
SERVICIOS DESPLEGADOS EN ELASTIC BEANSTALK DE AWS

Servicio	Estatus	URL
Base de datos MySQL	OK	awseb-e-83sxudddku-stack-awsebrdsdatabase-lwiexb2lrgeq
Backend Spring Boot	OK	http://kea-back.eba-g4c2ygaf.us-east-1.elasticbeanstalk.com
Frontend Angular	OK	http://kea-front.s3-website-us-east-1.amazonaws.com

6.3.2. Despliegue de aplicación en Azure App Service

La aplicación se desplegó en los servicios App Service y Database Azure MySQL. En la Tabla V se presenta los servicios desplegados con la respectiva URL.

TABLA V
SERVICIOS DESPLEGADOS EN APP SERVICE DE AZURE

Servicio	Estatus	URL
Base de datos MySQL	OK	kea-mysql.mysql.database.azure.com
Backend Spring Boot	OK	https://kea-backend.azurewebsites.net/
Frontend Angular	OK	https://proud-smoke-01525b710.3.azurestaticapps.net/

6.3.3. Despliegue de aplicación en GCP App Engine

Los servicios utilizados en GCP fueron App Engine y Cloud SQL. En la Tabla VI se presenta los servicios desplegados con la respectivas URL.

TABLA VI
SERVICIOS DESPLEGADOS EN APP ENGINE DE GCP

Servicio	Estatus	URL
Base de datos MySQL	OK	34.27.200.58
Backend Spring Boot	OK	https://kea-backend.uc.r.appspot.com
Frontend Angular	OK	https://kea-frontend.uc.r.appspot.com

6.4. Comparación de resultados de despliegue.

6.4.1. Resultados en AWS Elastic Beanstalk

Elastic Beanstalk ofrece buenas características de rendimiento, en la Tabla VII se muestra los resultados y el puntaje obtenido en cada indicador a evaluar.

TABLA VII
RESULTADO Y CALIFICACIÓN RENDIMIENTO DE ELASTIC BEANSTALK

Indicador	Resultado	Calificación
Tiempo de respuesta	Con la instancia básica se obtuvo los siguientes resultados: POST con un paquete de 643 bytes logró realizarla en 135 milisegundos. GET con un paquete de 710 bytes la realizó en 116 milisegundos.	9
Ancho de banda	Instancias t2.micro y t2.small, se permite un ancho de banda máximo de 1 Gbps, mientras que para las instancias c5d.9xlarge y m5a.24xlarge, el límite de ancho de banda es de 25 Gbps.	9
Escalabilidad	La instancia t3.micro cuenta con 2 vCPU, 1 GB de memoria y 8 GB de almacenamiento, mientras que la instancia c5d.9xlarge cuenta con 36 vCPU, 72 GB de memoria y 900 GB de almacenamiento.	9

La Tabla VIII señala las ponderaciones de cada uno de los indicadores proporcionados en el apartado de rendimiento.

TABLA VIII
INDICADOR DE RENDIMIENTO DE ELASTIC BEANSTALK

Señale con visto	Destacada	Moderada	Mínima
Dinámica temporal	✓		
Capacidad	✓		
Escalabilidad	✓		

Elastic Beanstalk ofrece configurar la seguridad según las necesidades del cliente o del servicio específico. A continuación, en la Tabla IX se presenta los resultados y calificaciones obtenidas en cada uno de los indicadores de seguridad.

TABLA IX
RESULTADO Y CALIFICACIÓN DE SEGURIDAD DE ELASTIC BEANSTALK

Indicador	Resultado	Calificación
Confidencialidad	Ofrece configurar la seguridad según las necesidades del cliente o del servicio específico	9
Integridad	Los datos son precisos y consistentes, esto se pudo evidenciar al realizar una petición GET y POST.	9
Autenticidad	Se verifica que las direcciones otorgadas tanto en backend como en frontend siempre son las mismas en cualquier petición realizada. El único contra que se encontró es que no se configura el protocolo HTTPS en la URL por defecto, ya que esto proporcionaría una capa adicional de seguridad y privacidad en la transferencia de datos.	8

La Tabla X señala la ponderación de cada una de las configuraciones proporcionadas en el apartado de seguridad.

TABLA X
INDICADOR DE SEGURIDAD EN ELASTIC BEANSTALK

Señale con visto	Destacada	Moderada	Mínima
Confidencialidad	✓		
Integridad	✓		
Autenticidad		✓	

Elastic Beanstalk no cuenta con una conexión directa a los repositorios de código desde su consola, lo que impide la automatización de los despliegues desde la propia plataforma. Sin embargo, ofrece pipelines que los clientes pueden configurar en sus repositorios de código, aunque también es necesario habilitar permisos adicionales para lograr este objetivo. Pero por otro lado se puede operar con varios lenguajes de programación, por ejemplo, Java y Python. La Tabla XI presenta los parámetros empleados para evaluar la compatibilidad del servicio con repositorios de código.

TABLA XI
INDICADOR DE COMPATIBILIDAD DE ELASTIC BEANSTALK

Señale con visto	Existe	No existe
Coexistencia con repositorios de código		✓
Interoperabilidad con lenguajes de programación	✓	

6.4.2. Resultados en Azure App Service

App Service ofrece buenas características de rendimiento en instancias premium, pero en instancias básicas tiene un rendimiento moderado. En la Tabla XII se ilustra los resultados y el puntaje obtenido en cada indicador a evaluar.

TABLA XII
RESULTADO Y CALIFICACIÓN DE RENDIMIENTO DE APP SERVICE

Indicador	Resultado	Calificación
Tiempo de respuesta	Con la instancia básica se obtuvo los siguientes resultados: POST de un paquete de 671 bytes fue de 229 milisegundos. GET de un paquete de 768 bytes fue de 164 milisegundos.	9
Ancho de banda	Plan "Gratis": El ancho de banda de entrada y salida está limitado a 165 MB por hora. Plan "Compartido": El ancho de banda de entrada y salida está limitado a 165 MB por hora. Plan "Básico": El ancho de banda de entrada y salida está limitado a 5 GB por día. Plan "Estándar": El ancho de banda de entrada y salida está limitado a 50 GB por día. Plan "Premium": El ancho de banda de entrada y salida está limitado a 200 GB por día.	8
Escalabilidad	ofrece varias opciones para el escalado automático, según el plan de precios que se esté utilizando. El plan "Basic" de App Service cuenta con una capacidad de procesamiento compartida, 1.75 GB de memoria y 10 GB de almacenamiento, mientras que el plan "Premium" cuenta con una capacidad de procesamiento dedicada, 14 GB de memoria y hasta 1 TB de almacenamiento.	8

La Tabla XIII presenta la ponderación realizada sobre la eficacia del rendimiento en App Service.

TABLA XIII
INDICADOR DE RENDIMIENTO DE APP SERVICE

Señale con visto	Destacada	Moderada	Mínima
Dinámica temporal	✓		
Capacidad		✓	
Escalabilidad		✓	

A continuación, en la Tabla XIV se presenta los resultados y calificaciones obtenidas en cada uno de los indicadores de seguridad.

TABLA XIV
RESULTADO Y CALIFICACIÓN DE SEGURIDAD DE APP SERVICE

Indicador	Resultado	Calificación
Confidencialidad	Proporciona opciones de configuración para proteger los datos y garantizar el funcionamiento ininterrumpido de la aplicación. Además, ofrece recursos adicionales de firewall que pueden vincularse al servicio contratado.	9
Integridad	Los datos son precisos y consistentes, esto se pudo evidenciar al realizar una petición GET y POST.	9
Autenticidad	Se verifica que las direcciones otorgadas tanto en backend como en frontend siempre son las mismas en cualquier petición realizada. Además, se ofrece el protocolo HTTPS en la URL por defecto.	9

La Tabla XV presenta los aspectos esenciales para garantizar la seguridad de los datos en el servicio.

TABLA XV
INDICADOR DE SEGURIDAD DE APP SERVICE

Señale con visto	Destacada	Moderada	Mínima
Confidencialidad	✓		
Integridad	✓		
Autenticidad	✓		

App Service brinda conexión directa a los principales repositorios de código desde la plataforma, entre estos se encuentra Github. Así mismo realiza la configuración del despliegue y entrega continua de forma automática. Por otro lado, también permite operar con varios lenguajes de programación como Java. En la Tabla XVI se observan los indicadores de la compatibilidad de los repositorios de código con el servicio de Azure.

TABLA XVI
INDICADOR DE COMPATIBILIDAD DE APP SERVICE

Señale con visto	Existe	No existe
Coexistencia con repositorios de código	✓	
Interoperabilidad con lenguajes de programación	✓	

6.4.3. Resultados en GCP App Engine

App Engine ofrece características moderadas en las instancias básicas, pero en instancias donde se realizó un escalado adecuado y en base a la demanda puede mejorar considerablemente. En la Tabla XVII se muestra los resultados y el puntaje obtenido en cada indicador a evaluar.

TABLA XVII
RESULTADO Y CALIFICACIÓN DE RENDIMIENTO DE APP ENGINE

Indicador	Resultado	Calificación
Tiempo de respuesta	Con la instancia básica se obtuvo los siguientes resultados: GET de un paquete de 803 bytes tardó 252 milisegundos. POST de un paquete de 739 bytes tardó 387 milisegundos	9
Ancho de banda	En lugar de tener un límite de ancho de banda específico, App Engine utiliza un modelo de precios basado en el consumo de recursos, como el tiempo de CPU, el almacenamiento y el ancho de banda de la red. Los recursos de red que se consumen se miden en función de los bytes enviados y recibidos por la aplicación. La capacidad máxima de operación resultó ser media, ya que el límite de una solicitud HTTP es de 32 MB.	8
Escalabilidad	Las instancias estándar de App Engine pueden tener hasta 8 vCPU, 30 GB de memoria y 2 TB de almacenamiento por instancia.	9

La Tabla XVIII señala las ponderaciones de cada uno de los indicadores proporcionados en el apartado de rendimiento.

TABLA XVIII
INDICADOR DE RENDIMIENTO DE APP ENGINE

Señale con visto	Destacada	Moderada	Mínima
Dinámica temporal	✓		
Capacidad		✓	
Escalabilidad	✓		

App Engine implementa políticas que proporcionan seguridad, asegurando los datos de los clientes. App Engine cuenta con un firewall capaz de regular el acceso a la aplicación mediante reglas de acceso. Las características evaluadas de seguridad en App Engine se describen en la Tabla XIX.

TABLA XIX
RESULTADO Y CALIFICACIÓN DE SEGURIDAD DE APP ENGINE

Indicador	Resultado	Calificación
Confidencialidad	Implementa políticas de seguridad y cuenta con un firewall capaz de regular el acceso a la aplicación mediante reglas de acceso.	9
Integridad	Los datos son precisos y consistentes, esto se pudo evidenciar al realizar una petición GET y POST.	9
Autenticidad	Se verifica que las direcciones otorgadas tanto en backend como en frontend siempre son las mismas en cualquier petición realizada. Además, se ofrece el protocolo HTTPS en la URL por defecto.	9

La Tabla XX presenta los aspectos esenciales para garantizar la seguridad de los datos en el servicio.

TABLA XX
INDICADOR SEGURIDAD APP ENGINE

Señale con visto	Destacada	Moderada	Mínima
Confidencialidad	✓		
Integridad	✓		
Autenticidad	✓		

App Engine no cuenta con una conexión directa a los repositorios de código desde su consola, lo cual impide la automatización de los despliegues desde la propia plataforma. Sin embargo, ofrece pipelines que los clientes pueden configurar en sus repositorios de código, aunque también es necesario habilitar permisos adicionales para lograr este objetivo. Además, ofrece soporte para operar con varios lenguajes de programación como Java. La Tabla XXI presenta los indicadores empleados para evaluar la compatibilidad de App Engine.

TABLA XXI
INDICADOR COMPATIBILIDAD APP ENGINE

Señale con visto	Existe	No existe
Coexistencia con repositorios de código		✓
Interoperabilidad con lenguajes de programación	✓	

6.4.4. Cuadro comparativo de las plataformas analizadas

TABLA XXII
CUADRO COMPARATIVO DE INDICADORES

Indicadores		Elastic Beanstalk	App Services	App Engine	
Eficiencia en el rendimiento	Dinámica Temporal	Destacada	✓	✓	✓
		Moderada			
		Mínima			
	Capacidad	Destacada	✓		
		Moderada		✓	✓
		Mínima			
	Escalabilidad	Destacada	✓		✓
		Moderada		✓	
		Mínima			
Seguridad	Confidencialidad	Destacada	✓	✓	✓
		Moderada			
		Mínima			
	Integridad	Destacada	✓	✓	✓
		Moderada			
		Mínima			
	Autenticidad	Destacada		✓	✓
		Moderada	✓		
		Mínima			
Compatibilidad	Coexistencia	Existe		✓	
		No existe	✓		✓
	Interoperabilidad	Existe	✓	✓	✓
		No existe			

La Tabla XXII muestra los resultados del análisis comparativo de las plataformas líderes en la nube, basándose en los atributos de rendimiento, seguridad y compatibilidad, de calidad del software de la norma internacional ISO/IEC 25010. A partir de estos resultados, se establecieron rangos de ponderación que indican que las ventajas de Elastic Beanstalk de AWS son una eficiencia de rendimiento favorable, provisión de recursos y capacidad máxima elevada debido al escalado horizontal automático, costos proporcionales a los recursos utilizados y opciones de configuración de seguridad

personalizables. No obstante, presenta desventajas como la falta de conexión directa a repositorios de código, lo cual dificulta la configuración de automatización de despliegues en su plataforma, y otra desventaja encontrada es el tiempo de respuesta moderado al implementar instancias y aplicaciones web.

Por otro lado App Service de Microsoft Azure presenta ventajas como un destacable comportamiento temporal, rápida creación de instancias, implementación dentro de los límites previstos, variedad de opciones de suscripción, adecuados límites de recursos y opciones de seguridad configurables, incluyendo firewall y configuración por roles de usuario. Además, ofrece conexión directa a repositorios de código y configuración automática de despliegue y entrega continua. Sin embargo, tiene desventajas como la posibilidad de cargos adicionales por transferencia de datos al usar ciertas capacidades.

La tercera plataforma estudiada que es App Engine de Google Cloud presenta ventajas como una dinámica temporal promedio, uso mínimo de recursos, seguridad mediante políticas y un firewall que regula el acceso a la aplicación sin generar costos adicionales en la facturación del servicio. Sin embargo, cuenta con desventajas como capacidad máxima de operación media, límite de 32 MB en cada solicitud HTTP y falta de conexión directa a repositorios de código al interior de la plataforma.

7. Discusión

El artículo "Magic Quadrant for Cloud Infrastructure and Platform Services" [31] permite comprender por qué AWS, Azure y GCP son considerados los principales proveedores de servicios de plataforma como servicio (PaaS). La posición de liderazgo de AWS, Azure y GCP en el mercado de PaaS se debe a su capacidad para satisfacer las necesidades empresariales actuales y futuras en términos de innovación, escalabilidad, soporte, seguridad y reputación. Estos factores, combinados con una oferta de servicios diversificada y en constante evolución, hacen que estas plataformas sean la elección preferida para las organizaciones que buscan aprovechar los beneficios de la nube.

Malik, Ahmed, Al-Qaydi y Alghamdi [32], analizan la migración de aplicaciones web on-premise a la nube y realizan una comparativa entre Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP). Presentan un caso de estudio en el que se migra una aplicación web específica a las tres plataformas en la nube. El objetivo principal del estudio es comparar el rendimiento, la escalabilidad, la disponibilidad y los costos asociados con cada plataforma. Los resultados del análisis proporcionaron información útil para los desarrolladores y las organizaciones que planean migrar sus aplicaciones web a un entorno de servicios en la nube. Esta información se pudo corroborar al implementar en las plataformas líderes, la aplicación web KEA que actualmente funciona en un modelo On-Premise, con lo cual se analizaron varias características en base al rendimiento, compatibilidad y seguridad.

Patil y Yardi [33], analizan las plataformas líderes en la nube, destacando algunas ventajas y desventajas de cada una de ellas. En el caso de AWS Elastic Beanstalk, se menciona la facilidad de uso, el escalado automático y el rendimiento eficiente como ventajas clave. Sin embargo, la falta de conexión directa a repositorios de código y la latencia en el despliegue se consideran desventajas. En cuanto a Microsoft Azure App Service, se destacan la rápida creación de instancias, el despliegue ágil y las opciones de suscripción flexibles. Aunque, se mencionan los posibles cargos adicionales por transferencia de datos como una desventaja. Por último, Google Cloud App Engine es elogiado por su seguridad y bajo consumo de recursos, pero se critica por sus limitaciones en el tamaño de las solicitudes HTTP y la falta de conexión directa a repositorios de código. En el presente TT se puede comprobar a través de las ventajas y desventajas obtenidas en el análisis comparativo.

8. Conclusiones

En conclusión, se ha establecido que Elastic Beanstalk de AWS, App Service de Microsoft Azure y App Engine de GCP son las plataformas principales en el ámbito de servicios en la nube para el desarrollo de aplicaciones web. Estas plataformas no solo lideran en el área de PaaS, sino también en IaaS, debido a la calidad del software evaluado. Además, es fundamental resaltar la importancia de integrar aplicaciones web en la nube para lograr una mayor escalabilidad y ofrecer un servicio confiable y continuo a los usuarios. La aplicación web KEA se desplegó exitosamente en las plataformas mencionadas, demostrando la utilidad de los servicios de PaaS en facilitar el despliegue, gracias a su amplia documentación y comunidad de apoyo.

En relación con la comparación de las plataformas según las facilidades de despliegue, Elastic Beanstalk de AWS sobresale en términos de eficiencia de rendimiento, tiempos de respuesta adecuados, y capacidad máxima elevada debido al escalado horizontal automático, costos proporcionales a los recursos utilizados y opciones de configuración de seguridad personalizables. Por otro lado, App Service de Microsoft Azure destaca por su rápida creación de instancias, conexión directa a repositorios de código, opciones de seguridad configurables y firewall, aunque presenta limitaciones respecto al ancho de banda y escalado en instancias básicas. Finalmente, App Engine de Google Cloud proporciona una dinámica temporal notable basada en los tiempos de respuesta obtenidos, seguridad reforzada mediante políticas y firewall, y protocolo HTTPS por defecto, aunque carece de conexión directa a los repositorios de código.

Es importante enfatizar que las tres plataformas presentaron buenos resultados en términos de tiempo de respuesta para peticiones GET y POST. Aunque los tiempos registrados en la aplicación web fueron ligeramente menores que en la nube, el margen de diferencia es estrecho y se podría disminuir al configurar instancias con más recursos. Finalmente, se sugiere que las organizaciones evalúen sus necesidades y objetivos específicos, considerando tanto las ventajas como las desventajas de cada plataforma, para tomar una decisión informada con relación a la selección de la plataforma de computación en la nube más adecuada para sus requisitos.

9. Recomendaciones

Es importante tomar en cuenta las plataformas en la nube para el desarrollo y despliegue de aplicaciones web, debido a que proporcionan mayor escalabilidad y confiabilidad en comparación con un modelo On-Premise. Es fundamental evaluar las necesidades y objetivos de la organización antes de seleccionar una plataforma en la nube, ponderando las ventajas y desventajas de cada proveedor y plataforma según los requisitos específicos.

En el caso de Elastic Beanstalk de AWS, es aconsejable emplear instancias con más recursos para disminuir la diferencia en los tiempos de respuesta respecto al modelo On-Premise. Además, se recomienda configurar la automatización de despliegues mediante herramientas externas como CodePipeline, que faciliten la integración con repositorios de código.

Para Microsoft Azure App Service, es crucial analizar el presupuesto disponible y considerar la inversión en instancias premium para obtener mayor escalabilidad y capacidad de almacenamiento. También es necesario tener presente el límite de ancho de banda y memoria en las instancias básicas.

Respecto a Google Cloud Platform App Engine, es conveniente examinar detenidamente las necesidades de almacenamiento y procesamiento de la aplicación, ya que el límite de una solicitud HTTP es de 32 MB. Asimismo, se aconseja contemplar la implementación de integración continua mediante herramientas externas que posibiliten la conexión directa a repositorios de código, esto se puede lograr activando otro servicio denominado Cloud Source Repositories.

En términos generales, es primordial considerar la seguridad de la aplicación y evaluar las opciones de configuración de seguridad que ofrecen cada una de las plataformas analizadas, incluyendo la disponibilidad del protocolo HTTPS por defecto y opciones de firewall y configuración por roles de usuario.

10. Bibliografía

- [1] P. Mell and T. Grance, “The NIST definition of cloud computing,” *NIST Special Publication*, vol. 800, no. 145, p. 7, 2011.
- [2] L. M. Vaquero, L. Roderó-Merino, J. Caceres, and M. Lindner, “A break in the clouds: towards a cloud definition,” *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2009.
- [3] K. Gai, M. Qiu, X. Sun, and L. Yang, “A Survey on FinTech,” *Journal of Network and Computer Applications*, vol. 103, pp. 262–273, 2019.
- [4] A. Sharma, A. K. Singh, and M. M. Tripathi, “A framework for intercloud interoperability in community cloud using open virtualization format,” *Future Generation Computer Systems*, vol. 83, pp. 339–353, 2018.
- [5] A. García, “Cloud computing as a managed service for resource management in organizations,” *Proc. 2019 IEEE International Conference on Cloud Computing (ICCC), San Diego, USA*, pp. 123–128, 2019.
- [6] S. González, “Serverless computing: An overview of architecture and applications,” *Proc. 2020 IEEE International Conference on Cloud Computing (ICCC), Beijing, China*, pp. 183–190, 2020.
- [7] P. Sbarski and S. Kroonenburg, “Serverless Architectures: The Evolution of Cloud Computing,” *2019 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 379–383, 2019.
- [8] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.
- [9] P. Singh, G. B. Singh, and K. Jyoti, “A study on resource provisioning of multi-tier web applications in cloud computing,” in *2015 International Conference on Computing for Sustainable Global Development*, 2015, pp. 799–802.
- [10] K. Lee, “An analysis of PaaS offerings from AWS, Azure, and GCP,” *Proc. 2021 IEEE International Conference on Cloud Computing (ICCC), Honolulu, USA*, pp. 123–130, 2021.
- [11] S. S. Gill, R. Buyya, and S. K. Garg, “An Evolutionary Algorithm for Scheduling Internet of Things Applications in Cloud Computing Environments,” *J Parallel Distrib Comput*, vol. 122, pp. 87–98, 2018.
- [12] J. Kim, “An overview of Microsoft Azure: Services, features, and capabilities,” *Proc. 2022 IEEE International Conference on Cloud Computing (ICCC), San Francisco, USA*, pp. 12–19, 2022.
- [13] D. Kim, “An overview of Google Cloud Platform: Services, features, and capabilities,” *Proc. 2023 IEEE International Conference on Cloud Computing (ICCC), Chicago, USA*, pp. 45–52, 2023.
- [14] K. Sivalingam and P. S. Kumar, “A Comparative Study of Cloud Service Providers for Web Application Deployment,” *2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India*, pp. 419–423, 2019.
- [15] M. A. AlZain and E. Pardede, “Handbook of Big Data and IoT Security,” *Cloud Computing Security: From Single to Multi-Clouds*, pp. 271–291, 2020.
- [16] D. Taibi, V. Lenarduzzi, and P. Diebold, “How Developers Perceive the Continuous Integration and Continuous Delivery (CI/CD) Activities in Cloud Platforms,” *EEE/ACM 5th International Workshop on Conducting Empirical Studies in Industry (CESI)*, pp. 49–52, 2018.

- [17] J. Lee, "Architectures for web application development: A review and comparison," *Proc. 2021 IEEE International Conference on Cloud Computing (ICCC), Honolulu, USA*, pp. 78–85, 2021.
- [18] S. Kim, "Serverless architecture for cloud applications: A review and outlook," *Proc. 2022 IEEE International Conference on Cloud Computing (ICCC), San Francisco, USA*, pp. 76–83, 2022.
- [19] Yadav, P., and Singh, "A comprehensive study of AWS and its services," *International Journal of Engineering Science and Computing*, vol. 9, 2019.
- [20] J. Pérez and M. Sánchez, "Inspección y valoración de la eficiencia de Amazon EC2 en el contexto de aplicaciones corporativas," *Revista de Tecnología de la Información*, vol. 14, no. 4, pp. 78–85, 2021.
- [21] R. López and L. García, "Evaluación y comparativa del servicio de almacenamiento Amazon S3 en entornos empresariales," *Revista de Tecnología de la Información*, vol. 14, no. 2, pp. 46–53, 2021.
- [22] P. Martínez and F. Rodríguez, "Análisis y optimización del rendimiento en bases de datos utilizando Amazon RDS," *Revista de Tecnología de la Información*, vol. 14, no. 5, pp. 109–117, 2021.
- [23] N. Sánchez and D. Torres, "Evaluación de la plataforma Azure App Service en el desarrollo y despliegue de aplicaciones web y móviles," *Revista de Tecnología de la Información*, vol. 13, no. 3, pp. 67–74, 2020.
- [24] Balazín, "Comparison of Platform as a Service Solutions for IoT System Development," *Journal of Information and Organizational Sciences*, vol. 1, pp. 59–73, 2020.
- [25] J. Gómez and A. Herrera, "Desarrollo y despliegue de aplicaciones utilizando Azure Database for MySQL: un enfoque práctico," *Revista de Investigación en Tecnologías de la Información*, vol. 12, no. 4, pp. 89–97, 2020.
- [26] Bhardwaj, Chawla, and Syal, "A Comprehensive Study on Google App Engine and Its Services," *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1284–1289, 2019.
- [27] C. Navarro and E. Ortega, "Exploración y evaluación de Google Cloud SQL en el desarrollo y despliegue de aplicaciones en la nube," *Revista de Investigación en Tecnologías de la Información*, vol. 15, no. 6, pp. 120–128, 2021.
- [28] ISO/IEC JTC 1/SC 7 Software and systems engineering, *ISO/IEC 25010*, ISO/IEC 25010. 2015.
- [29] E. Almeida, F. Dantas, C. Oliveira, and L. Morais, "An ISO-based approach to deriving quality indicators for assessing software product quality," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion, QRS-C*, Ed., 2018, pp. 400–407.
- [30] Zhang, Gao, and Qiu, "A Comparative Study of Cloud PaaS Services for Web Application Deployment," *International Conference on Cybernetics and Intelligent Systems and IEEE Conference on Robotics, Automation and Mechatronics*, pp. 550–555, 2021.
- [31] B. Analyst, R. Bala, D. Smith, K. Ji, D. Wright, and M. Angel Borrega, "Magic Quadrant for Cloud Infrastructure and Platform Services," 2022.
- [32] M. Malik, N. B. E. Ahmed, A. M. M. Al-Qaydi, and M. A. Alghamdi, "Comparison of Cloud Platforms for Migrating On-Premises Web Application," *3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan*, pp. 1–7, 2019.
- [33] J. R. Patil and A. R. Yardi, "Comparative study of cloud computing platforms: Amazon Web Services (AWS), Google Cloud Platform (GCP), IBM Cloud and Microsoft Azure,"

- International Conference on Communication, Information and Computing Technology (ICCICT), Mumbai, India*, pp. 1–6, 2018.
- [34] R. Mathew, R. Ramachandran, and B. Radhakrishnan, “Comparative analysis of Kubernetes based container orchestration platforms: Red Hat OpenShift, Google Kubernetes Engine and Azure Kubernetes Service,” *International Conference on Artificial Intelligence and Signal Processing (AISP), Jalgaon, India*, pp. 1–6, 2020.
- [35] Smith and J. K., “An analysis of web application deployment options in the cloud,” *Journal of Cloud Computing*, pp. 127–134, 2021.
- [36] D. Bhowmik and A. Bhattacharya, “A comparative study of Heroku, AWS and GCP for web application hosting,” *Calcutta Conference (CALCON), Kolkata, India*, pp. 370–375, 2019.
- [37] Kim S. and Park J., “A comparative study of cloud platforms for web application deployment,” *Proceedings of the 2021 IEEE International Conference on Cloud Computing*, pp. 234–241, 2021.
- [38] S. Garg, A. K. Singh, and M. A. Rodrigues, “Comparative Analysis of Cloud Computing Services: Alibaba Cloud, Amazon Web Services and Microsoft Azure,” *4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India*, pp. 364–369, 2019.

11. Anexos

Anexo 1. Documento de seguimiento para servicios de computación en la nube.

La Tabla XXIII muestra las características que afectan al rendimiento de la plataforma de computación en la nube. Estas características incluyen la dinámica temporal, la capacidad y la escalabilidad. La dinámica temporal mide el tiempo de respuesta de peticiones simples como GET y POST. La capacidad se mide de acuerdo con el ancho de banda proporcionado por cada plataforma. Finalmente, la escalabilidad se evalúa para determinar la capacidad de ajustar los recursos necesarios automáticamente para satisfacer la demanda de usuarios.

TABLA XXIII
EFICIENCIA RENDIMIENTO

Señale con visto	Destacada	Moderada	Mínima
Dinámica temporal			
Capacidad			
Escalabilidad			

En la Tabla XXIV, se presentan intervalos de ponderación que indican el nivel de importancia de cada característica evaluada. El rango 1-4 se interpreta como un rendimiento mínimo, el rango 5-8 indica una eficiencia de rendimiento moderado y el rango 9-10 refleja un destacado nivel de importancia.

TABLA XXIV
RANGO VALORACIÓN DE EFICIENCIA RENDIMIENTO

Rango	Interpretación	Nivel de importancia
9-10	Rendimiento destacado	Destacado
5-8	Rendimiento moderado	Moderado
1-4	Rendimiento mínimo	Mínimo

La Tabla XXV enumera los factores críticos que deben ser considerados para evaluar la seguridad de una plataforma.

La confidencialidad se determina en base a los accesos no autorizados o incidentes de violaciones de datos. La integridad se determina en base a la garantía de que los datos no sean alterados o corrompidos. Y la autenticidad se determina en base a que la información proviene de una fuente confiable y que no ha sido alterada durante su transmisión.

TABLA XXV
CARACTERÍSTICAS DE EVALUACIÓN DE LA SEGURIDAD

Señale con visto	Destacada	Moderada	Mínima
Confidencialidad			
Integridad			
Autenticidad			

La Tabla XXVI se presenta para evaluar el aspecto de seguridad, en la cual se detallan los rangos correspondientes a los parámetros descritos en la Tabla XXV. Así, un rango de 1 a 4 indica un nivel de seguridad mínimo, de 5 a 7 representa un nivel de seguridad moderado y, finalmente, un rango de 8 a 10 denota un nivel de seguridad destacado, que es el ideal para asegurar la protección.

TABLA XXVI
RANGO DE VALORACIÓN DE SEGURIDAD

Rango	Interpretación	Nivel de importancia
8-10	Seguridad destacada	Destacada
4-7	Seguridad moderada	Moderada
1-3	Seguridad mínima	Mínima

Para confirmar si la plataforma es compatible, se ha seleccionado dos elementos importantes que son la coexistencia con repositorios de código y la integración con varios lenguajes de programación. Se muestra la Tabla XXVII, donde se destacan las características esenciales para respaldar la evaluación de la característica de compatibilidad.

TABLA XXVII
CARACTERÍSTICAS DE COMPATIBILIDAD

Señale con visto	Existe	No existe
Coexistencia con repositorios de código		
Interoperabilidad con lenguajes de programación		

Anexo 2. Trabajos relacionados

En la Tabla XXVIII se describe algunos trabajos relacionados al trabajo de titulación.

TABLA XXVIII
TRABAJOS RELACIONADOS CON EL TT

Nº	Título	Hallazgo	Cita	Lugar
1	An analysis of PaaS offerings from AWS, Azure, and GCP	Este artículo realiza un análisis comparativo de las plataformas como servicio (PaaS) ofrecidas por Amazon Web Services (AWS), Microsoft Azure y Google Cloud Platform (GCP). El estudio considera varios factores como características, rendimiento, funcionalidades, precios y soporte al cliente. Los resultados obtenidos permitieron identificar las ventajas y desventajas de cada plataforma y proporcionar una comparativa detallada para ayudar a las empresas a tomar una decisión informada al elegir una plataforma PaaS para sus proyectos de aplicación en la nube.	[10]	Honolulu, USA 2021
2	Comparative analysis of Kubernetes based container orchestration platforms: Red Hat OpenShift, Google Kubernetes Engine and Azure Kubernetes Service	En este estudio se compararon tres plataformas en la nube basadas en Kubernetes: Red Hat OpenShift, Google Kubernetes Engine (GKE) y Microsoft Azure Kubernetes Service (AKS). Los autores analizaron factores como rendimiento, escalabilidad, seguridad, costos y facilidad de uso. Red Hat OpenShift fue destacado por su enfoque en contenedores y orquestación de Kubernetes, así como por su solidez en la administración de clústeres y aplicaciones empresariales. GKE y AKS también fueron evaluados en base a sus características y capacidades en el ecosistema de Kubernetes. De acuerdo con los autores, al seleccionar la plataforma en la nube adecuada, es crucial que las organizaciones evalúen sus propias necesidades y objetivos, y consideren tanto las ventajas como las desventajas de cada plataforma en función de sus requisitos específicos.	[34]	Jalgaon, India 2020

3	An analysis of web application deployment options in the cloud. Journal of Cloud Computing	Este estudio comparativo de las diferentes opciones de despliegue de aplicaciones web en la nube encontró que App Service de Microsoft Azure es una excelente opción para los desarrolladores. La integración de herramientas de terceros y la facilidad de uso de la plataforma fueron aspectos muy valorados por los usuarios. En general, se concluyó que App Service de Azure es una excelente opción para el despliegue de aplicaciones web, especialmente para empresas que ya utilizan otros servicios en la nube de Microsoft.	[35]	Berlín, Alemania 2021
4	A comparative study of Heroku, AWS and GCP for web application hosting	En este estudio se compararon tres plataformas en la nube: Heroku, Amazon Web Services (AWS) y Google Cloud Platform (GCP), centrándose en el desarrollo y despliegue de aplicaciones web. Los autores analizaron factores como rendimiento, escalabilidad, seguridad, costos y facilidad de uso. Heroku fue destacado por su simplicidad y facilidad de uso, especialmente para desarrolladores que buscan implementar aplicaciones rápidamente. AWS y GCP fueron evaluados en base a su amplia gama de servicios y flexibilidad. Los autores concluyeron que la elección de la plataforma en la nube depende de las necesidades específicas y los objetivos de cada desarrollador.	[36]	Kolkata, India 2019
5	A comparative study of cloud platforms for web application deployment	En este estudio se compararon las plataformas líderes en términos de rendimiento, escalabilidad, facilidad de uso y flexibilidad. Se observó que estas tres plataformas ofrecen una amplia variedad de características y opciones de personalización para satisfacer las necesidades específicas de cada proyecto. Además, la integración con otras herramientas y servicios de terceros fue considerada una ventaja para estas plataformas.	[37]	San Francisco, USA 2021

6	Comparative Analysis of Cloud Computing Services: Alibaba Cloud, Amazon Web Services and Microsoft Azure	Los autores realizaron un estudio comparativo de las plataformas en la nube Alibaba Cloud, Amazon Web Services (AWS) y Microsoft Azure. Los autores analizaron factores como rendimiento, escalabilidad, seguridad, costos y facilidad de uso. Alibaba Cloud fue destacado por su desempeño en el mercado asiático y por su creciente presencia global. AWS y Microsoft Azure fueron evaluados en base a su amplia gama de servicios y su dominio en el mercado global.	[38]	Mathura, India 2019
7	A Comparative Study of Cloud PaaS Services for Web Application Deployment	Según este estudio, Elastic Beanstalk, App Service y App Engine son las plataformas líderes para el despliegue de aplicaciones web debido a su facilidad de uso, escalabilidad y flexibilidad. Los autores encontraron que Elastic Beanstalk es la mejor opción para aplicaciones basadas en contenedores, mientras que App Service es ideal para aplicaciones .NET y App Engine es adecuada para aplicaciones basadas en Python y Java.	[30]	Nanjing, China 2021
8	Comparative study of cloud computing platforms: Amazon Web Services (AWS), Google Cloud Platform (GCP), IBM Cloud and Microsoft Azure	En este estudio además de comparar las plataformas en la nube IBM Cloud, AWS, GCP y Azure en términos de rendimiento, escalabilidad, seguridad, costos y facilidad de uso, los autores también concluyeron que cada plataforma tiene características y servicios específicos que la hacen única y adecuada para ciertos casos de uso.	[33]	Mumbai, India 2018
9	Comparison of Cloud Platforms for Migrating On-Premises Web Application	En este estudio se analiza la migración de aplicaciones web On-Premise a la nube y realizan una comparativa entre AWS, Microsoft Azure y GCP. Presentan un caso de estudio en el que se migra una aplicación web específica a las tres plataformas en la nube. Se comparó el rendimiento, la escalabilidad, la disponibilidad y los costos asociados con cada plataforma.	[32]	Sukkur, Pakistan 2019

Las principales plataformas que se estudiaron en los artículos mencionados en la tabla anterior se describen a continuación en la Tabla XXIX.

TABLA XXIX
PLATAFORMAS ESTUDIADAS EN ARTÍCULOS RELACIONADOS

Nº	Título	Plataformas estudiadas
1	An analysis of PaaS offerings from AWS, Azure, and GCP	AWS, Microsoft Azure y GCP
2	Comparative analysis of Kubernetes based container orchestration platforms	Red Hat OpenShift, Google Kubernetes Engine and Azure Kubernetes Service
3	An analysis of web application deployment options in the cloud. <i>Journal of Cloud Computing</i>	Microsoft Azure
4	A comparative study of Heroku, AWS and GCP for web application hosting	Heroku, AWS y GCP
5	A comparative study of cloud platforms for web application deployment	AWS, Microsoft Azure y GCP
6	Comparative Analysis of Cloud Computing Services	Alibaba Cloud, AWS y Microsoft Azure
7	A Comparative Study of Cloud PaaS Services for Web Application Deployment	AWS, Azure y GCP
8	Comparative study of cloud computing platforms	IBM Cloud, AWS, GCP y Microsoft Azure
9	Comparison of Cloud Platforms for Migrating On-Premises Web Application	AWS, Microsoft Azure y GCP

Anexo 3. Manual de despliegue aplicación cliente-servidor en AWS Elastic Beanstalk

En este paso se debe elegir la plataforma adecuada, para esta aplicación se eligió Java, como se ilustra en la Fig. 9.

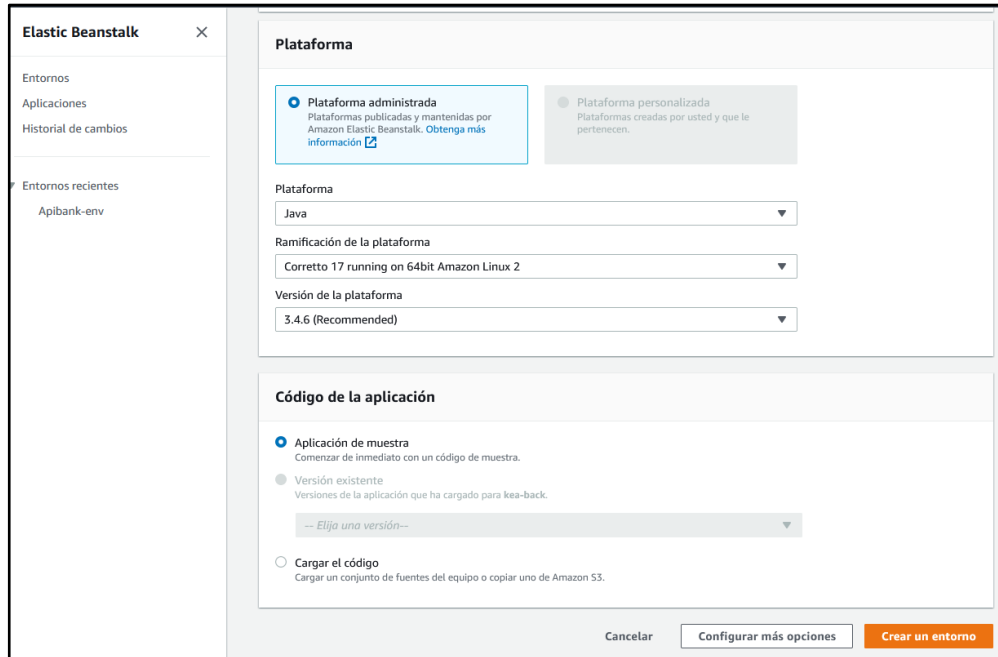


Fig. 9. Creación ambiente en Elastic Beanstalk

Una vez creado el entorno, el siguiente paso fue configurar el puerto en "Más Opciones" > "Software" > "Editar". De esta manera, especificamos el puerto que utilizará la aplicación en el lado del servidor, es decir, el backend, tal como se puede apreciar en la Fig. 10.

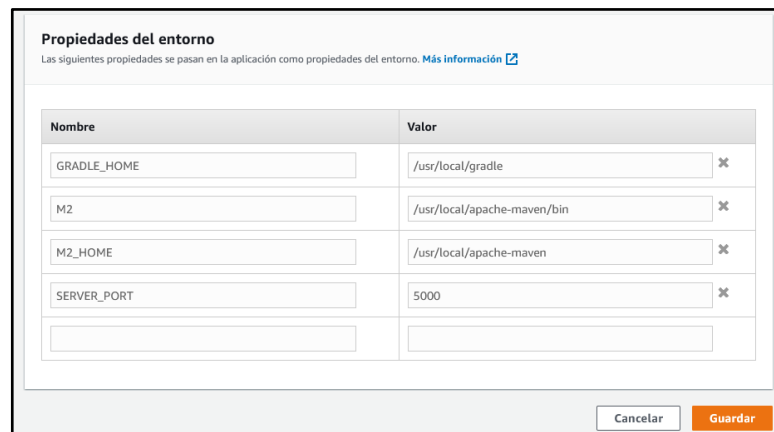


Fig. 10. Configuración de puerto backend en AWS

Después, en la opción "Base de datos", se estableció la conexión correspondiente a MySQL, tal como se ilustra en la Fig. 11.

Configuración de la base de datos

Seleccione un motor y tipo de instancia para la base de datos de su entorno.

Motor
mysql

Versión del motor
5.7.41

Clase de instancia
db.t2.micro

Almacenamiento
Elija un número entre 5 GB y 1024 GB.
5

Nombre de usuario
root

Contraseña
[Oculto]

Fig. 11. Configuración RDS en AWS

Una vez creado el entorno, es necesario esperar varios minutos para que se complete el proceso, como se puede apreciar en la Fig. 12.

Elastic Beanstalk > Entornos > kea-back

Creando kea-back
Este proceso tardará unos minutos. ..

```
11:21pm Using elasticbeanstalk-us-east-1-874485790217 as Amazon S3 storage bucket for environment data.  
11:21pm createEnvironment is starting.
```

Fig. 12. Creación del entorno en AWS para backend

Posteriormente, se cargó el archivo JAR correspondiente mediante la opción de carga disponible en el entorno recién creado, como se ilustra en la Fig. 13.

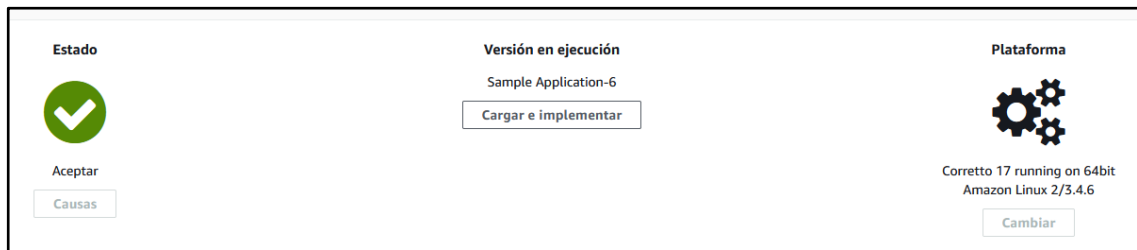


Fig. 13. Carga del backend en AWS

No obstante, es necesario realizar la configuración de los properties en el backend, de acuerdo con lo ilustrado en la Fig.14.

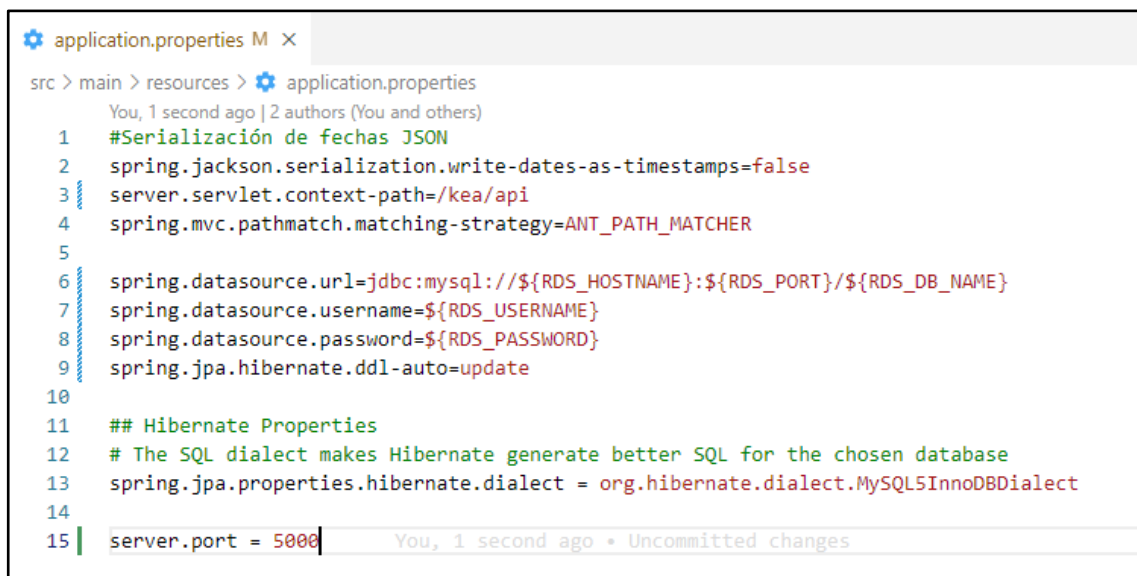


Fig. 14. Configuración de properties para backend en AWS

Una vez desplegado el backend, se procedió a crear un bucket en S3 para el frontend. En este paso, es importante habilitar el acceso público completo, tal como se ilustra en la Fig. 15.

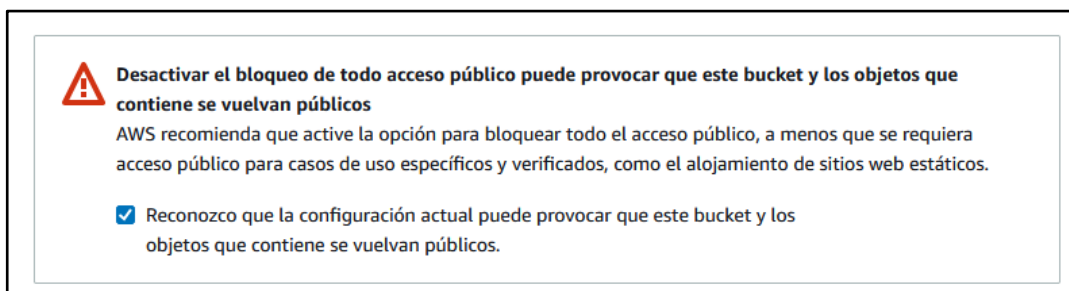


Fig. 15. Desbloqueo del acceso público en Bucket de AWS

Posteriormente, la siguiente configuración se llevó a cabo de manera más simple, ya que solo fue necesario subir los archivos compilados de Angular y hacerlos públicos, como se muestra en la Fig. 16, obviamente debe configurarse previamente el ambiente de producción con la URL del backend que se genera en EC2. Al completar este proceso, la aplicación se encontraba completamente funcional en la nube de AWS.

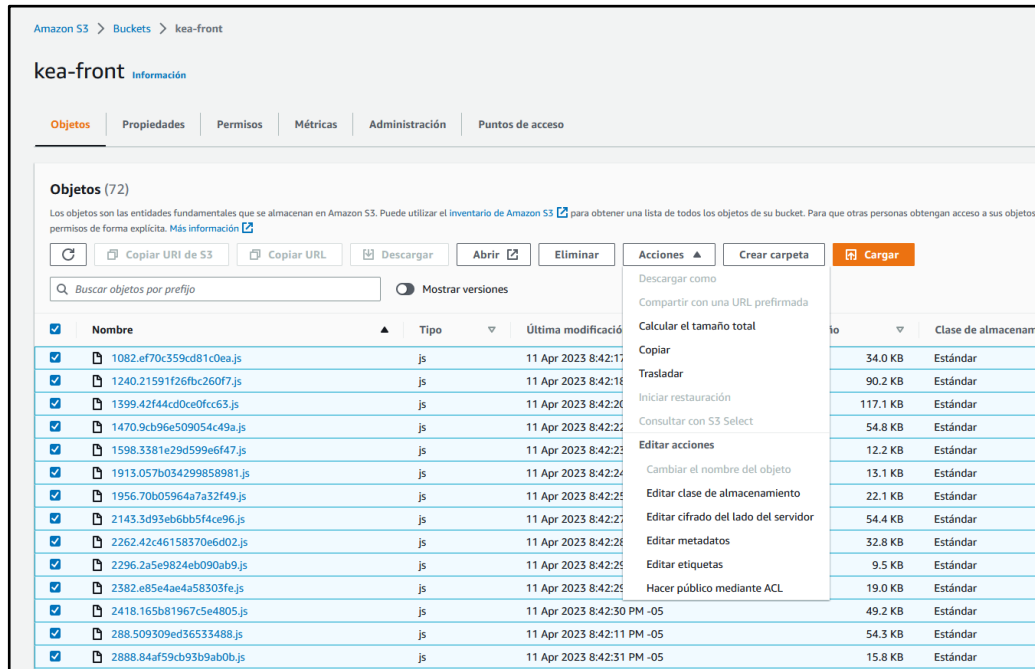


Fig. 16. Subida de archivos y permisos en Bucket S3 de AWS

Anexo 4. Manual de despliegue de aplicación cliente-servidor en App Service de Azure

Para el despliegue del backend en Azure, se creó un recurso de base de datos MySQL versión 5.7, esto se ilustra en la Fig. 17.

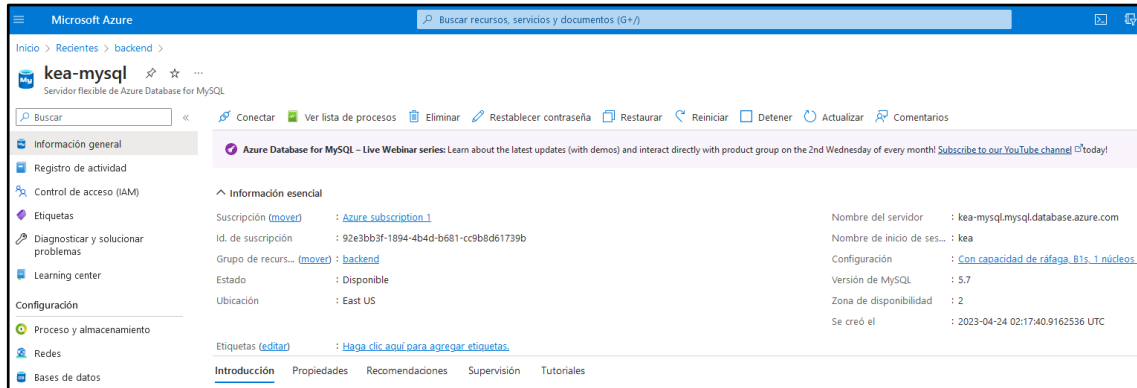
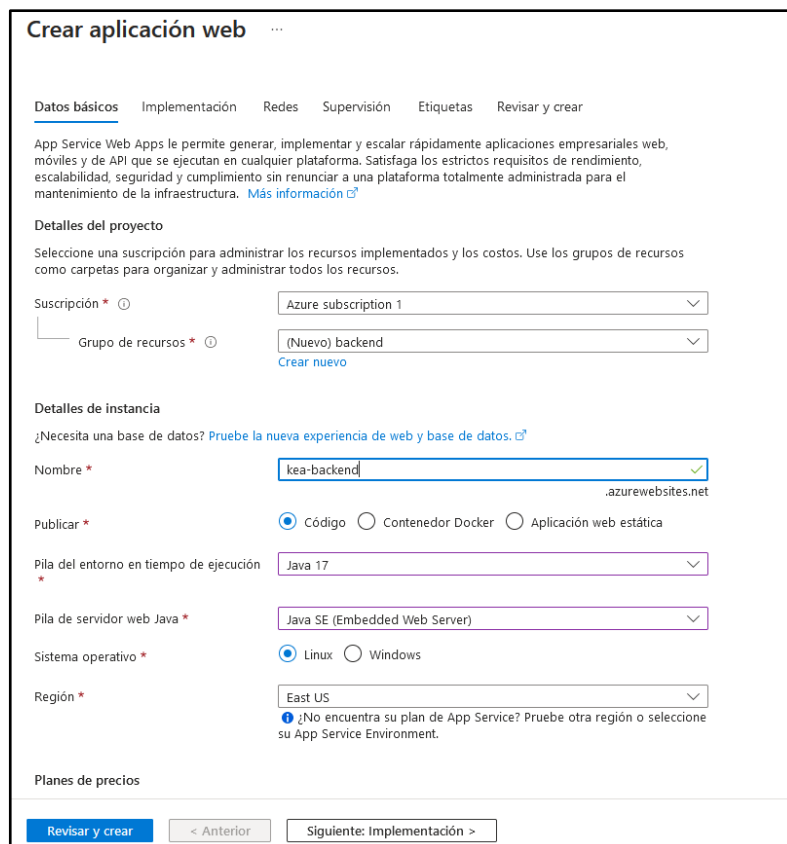


Fig. 17. Creación de recurso de base de datos MySQL en App Service

App Service Web, en el cual se detalló la versión de java, como se ilustra en la Fig. 18.



The screenshot shows the 'Crear aplicación web' wizard in the Azure portal. The 'Datos básicos' tab is active. The wizard provides an overview of App Service Web Apps and asks for project details. The 'Suscripción' is set to 'Azure subscription 1' and the 'Grupo de recursos' is '(Nuevo) backend'. Under 'Detalles de instancia', the 'Nombre' is 'kea-backend', 'Publicar' is set to 'Código', the 'Pila del entorno en tiempo de ejecución' is 'Java 17', the 'Pila de servidor web Java' is 'Java SE (Embedded Web Server)', the 'Sistema operativo' is 'Linux', and the 'Región' is 'East US'. At the bottom, there are buttons for 'Revisar y crear', '< Anterior', and 'Siguiente: Implementación >'.

Fig. 18. Creación App Service Web

Luego que en un par de minutos se termina de implementar el App Web Service, en la opción Centro de Implementación enlazamos directamente el repositorio de código para realizar el despliegue directamente y dejando lista la configuración de entrega continua, tal como se observa en la Fig. 19.

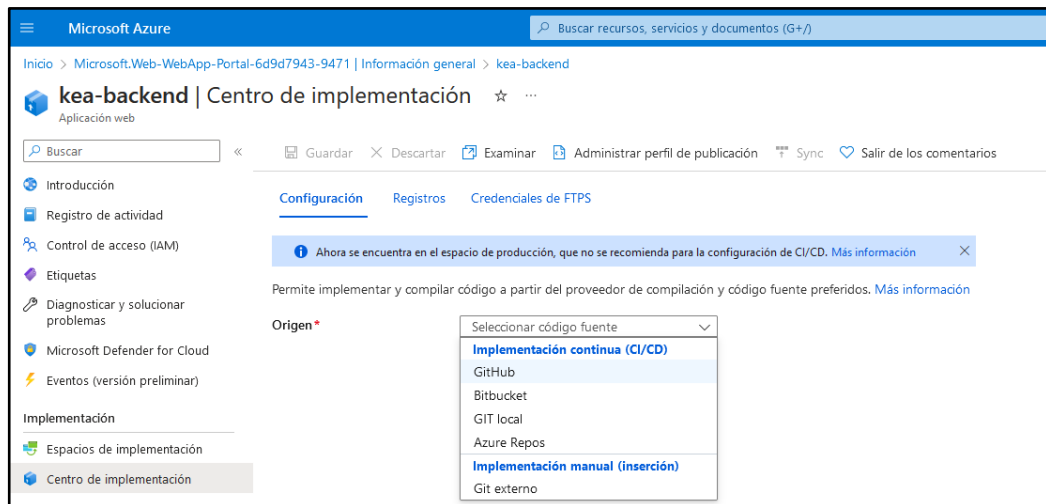


Fig. 19. Centro de implementación de App Web Service Azure

En la Fig. 20 se muestra la forma de conectar la cuenta de github. Luego de eso se seleccionó el repositorio y la rama principal, con lo cual luego de guardar, Azure automáticamente crea un archivo yml en el repositorio para desplegar la aplicación cada vez que se confirme un cambio en la rama seleccionada. De esa forma el backend queda listo y configurado para desplegar cualquier cambio que se pase a la rama de producción.

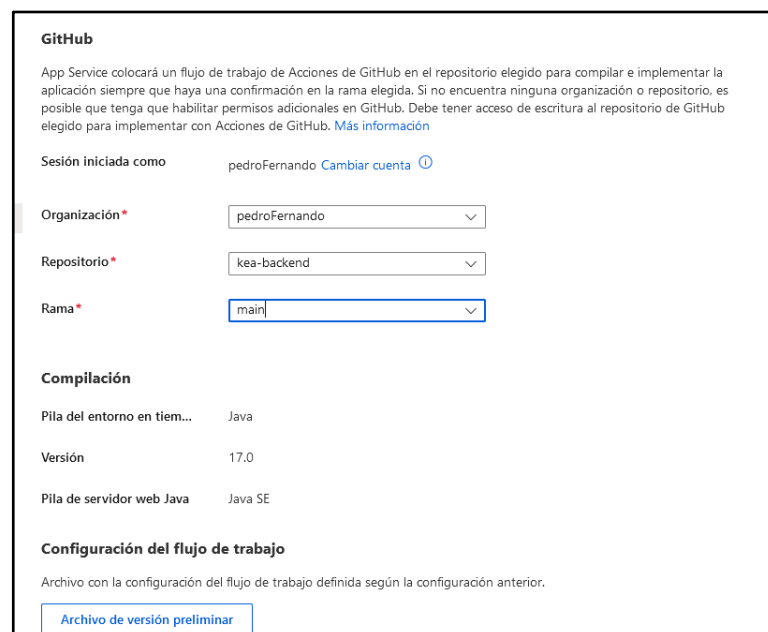


Fig. 20. Configuración de GitHub con CI/CD en App Web Service de Azure

Para el despliegue del frontend, se configuró previamente el ambiente de producción con la URL proporcionada en el despliegue del backend y luego se siguió el mismo procedimiento de este, pero en la opción Publicar se escoge Aplicación web estática, así como se muestra en la Fig. 21.

The screenshot shows the 'Crear una aplicación web estática' (Create a static web application) configuration page in the Azure portal. The page is titled 'Inicio > Crear una aplicación web estática ...'. It includes sections for 'Plan de hospedaje' (Hosting plan), 'Detalles de Azure Functions y detalles de la ubicación provisional' (Azure Functions details and provisional location), 'Detalles de la implementación' (Deployment details), and 'Detalles de la compilación' (Build details). The 'Tipo de plan' (Plan type) is set to 'Gratis' (Free). The 'Región para la API y los entornos de ensayo de Azure Functions' (Region for API and Azure Functions test environments) is set to 'Central US'. The 'Origen' (Origin) is set to 'GitHub'. The 'Cuenta de GitHub' (GitHub account) is 'pedrofernando'. The 'Organización' (Organization) is 'pedrofernando', the 'Repositorio' (Repository) is 'kea-frontend', and the 'Rama' (Branch) is 'main'. The 'Valores preestablecidos de compilación' (Default build values) are set to 'Estos campos reflejarán la estructura de proyecto predeterminada del' (These fields will reflect the default project structure). The page has a 'Revisar y crear' (Review and create) button and navigation links for '< Anterior' (Previous) and 'Siguiente: Etiquetas >' (Next: Tags).

Fig. 21. Creación de App Web Service para frontend

En la Fig. 22 finalmente se puede observar en Github que el pipeline que App Service de Azure se ejecuta correctamente, cabe destacar que cada vez que se suba un cambio a la rama seleccionada, en este caso main, se actualizará la aplicación en App Service.

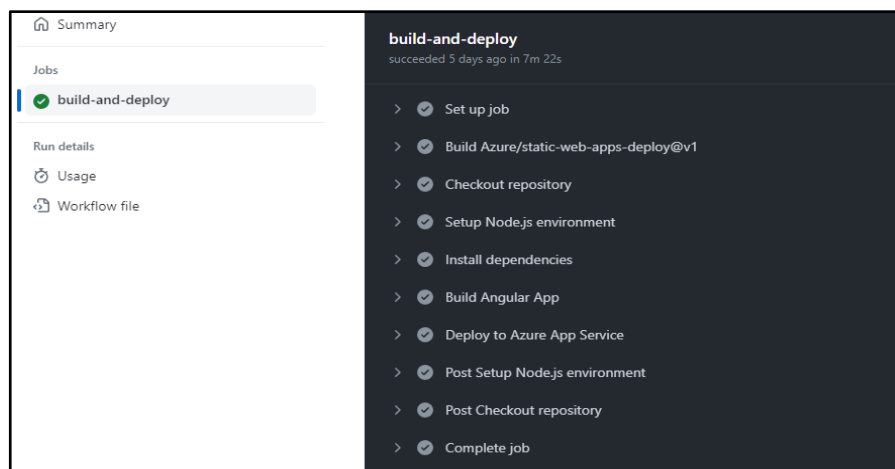


Fig. 22. Ejecución de pipeline de App Service de Azure en Github Actions

Anexo 5. Manual de despliegue de aplicación cliente-servidor en Cloud Engine de GCP

Lo primero que se realizó es crear dos proyectos en Google Cloud, uno para backend y otro para frontend, en los cuales se deben habilitar el servicio App Engine, tal como se puede apreciar en la Fig. 23.

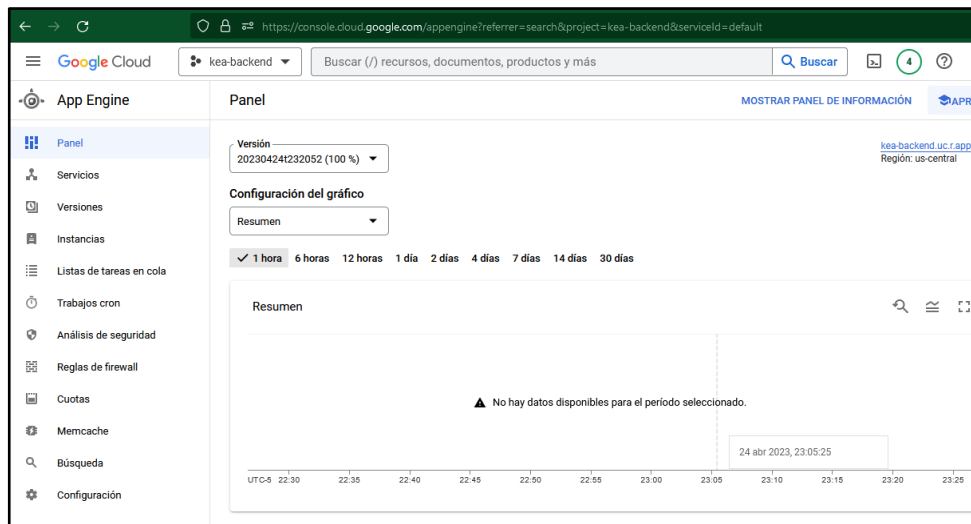


Fig. 23. Consola de App Engine de GCP

Para el despliegue del backend en GCP se configuró una instancia de MySQL versión 5.7, tal como se ilustra en la Fig. 24.

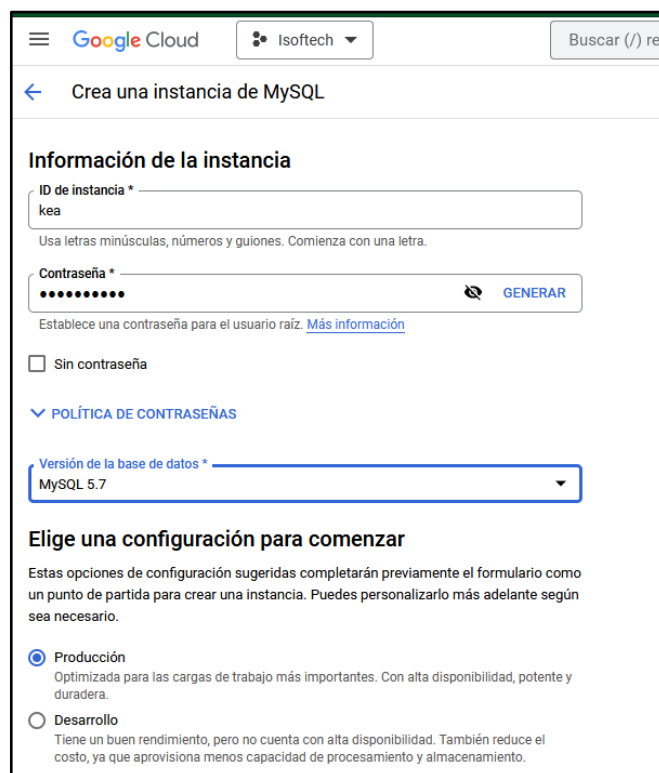


Fig. 24. Configuración MySQL GCP

Luego en la opción Base de datos se creó la base kea con el grupo de caracteres requerido en la aplicación, tal como se ilustra en la Fig. 25.

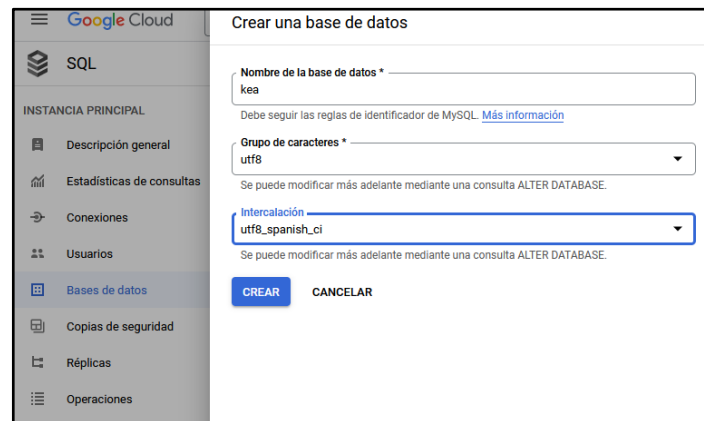


Fig. 25. Crear base de datos kea en instancia GCP

En la Fig. 26 se presenta la instalación del SDK de Google Cloud, para desplegar la aplicación Java directamente desde la terminal.

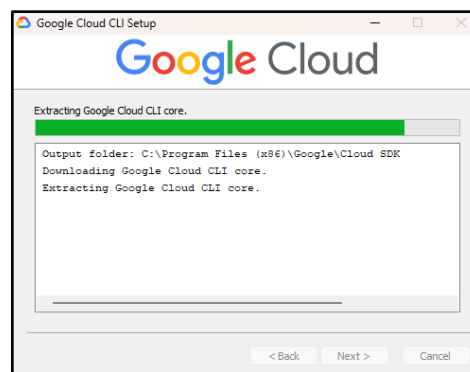


Fig. 26. Instalación SDK Google Cloud

Al final de la instalación se realizó la selección del proyecto que previamente debe ser configurado en la consola de Google Cloud, para luego configurar la información de conexión de la base de datos en el archivo properties del backend, como se puede ver en la Fig. 27.

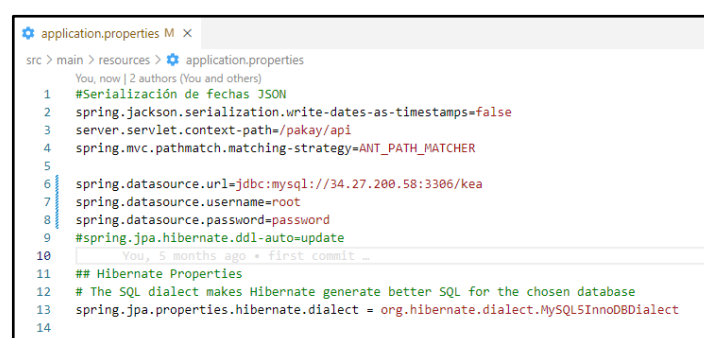
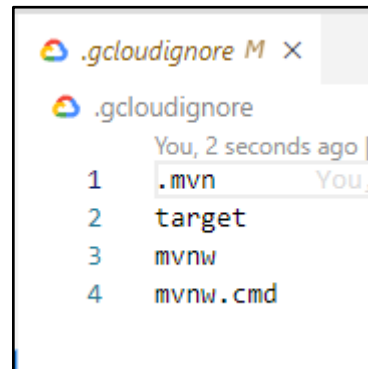


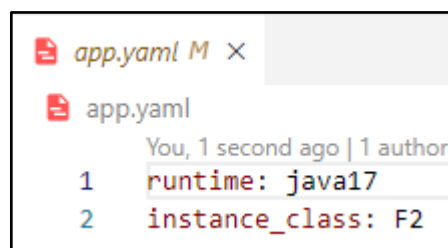
Fig. 27. Configuración archivo properties con información GCP

Luego se configuró los archivos `gcloudignore` y `app.yaml`, como se muestra en la Fig. 28 y Fig. 29.



```
.gcloudignore M X
.gcloudignore
You, 2 seconds ago |
1 .mvn
2 target
3 mvnw
4 mvnw.cmd
```

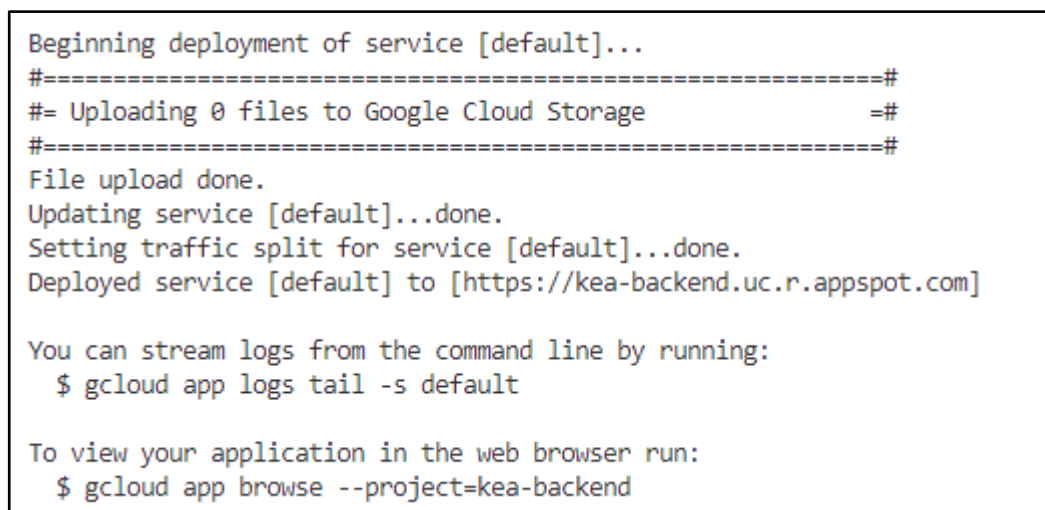
Fig. 28. Configuración `gcloudignore` para deploy GCP



```
app.yaml M X
app.yaml
You, 1 second ago | 1 author
1 runtime: java17
2 instance_class: F2
```

Fig. 29. Configuración `app.yaml` para deploy GCP

Después con el comando `gcloud app deploy -q --project kea-backend` se desplegó la aplicación, seleccionando la localización del servidor, con lo que la API Rest del backend queda lista para ser consumida, tal como se puede observar en la Fig. 30.



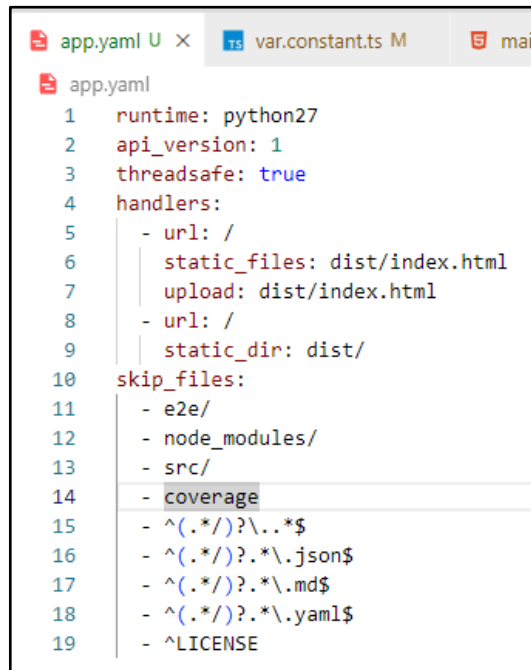
```
Beginning deployment of service [default]...
#=====#
#=- Uploading 0 files to Google Cloud Storage      =#
#=====#
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://kea-backend.uc.r.appspot.com]

You can stream logs from the command line by running:
  $ gcloud app logs tail -s default

To view your application in the web browser run:
  $ gcloud app browse --project=kea-backend
```

Fig. 30. Despliegue listo de backend en GCP

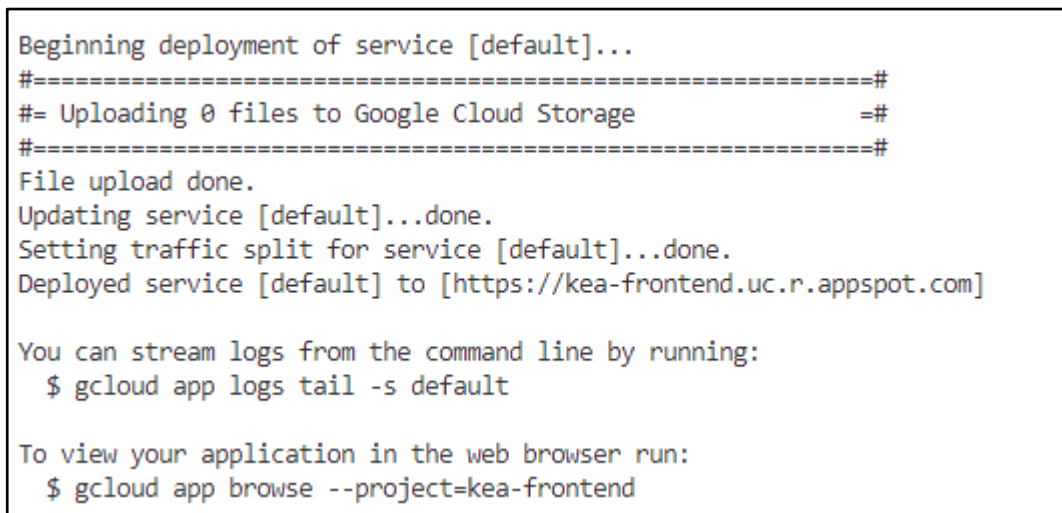
Para el caso del frontend se realizó la configuración del archivo `app.yaml`, tal como se puede observar en la Fig. 31.



```
app.yaml
1 runtime: python27
2 api_version: 1
3 threadsafe: true
4 handlers:
5   - url: /
6     static_files: dist/index.html
7     upload: dist/index.html
8   - url: /
9     static_dir: dist/
10 skip_files:
11   - e2e/
12   - node_modules/
13   - src/
14   - coverage
15   - ^(\.*/)?\..*$
16   - ^(\.*/)?.*\.json$
17   - ^(\.*/)?.*\.md$
18   - ^(\.*/)?.*\.yaml$
19   - ^LICENSE
```

Fig. 31. Configuración archivo `app.yaml` para deploy frontend GCP

Finalmente, con el comando `gcloud app deploy -q --project kea-frontend` se despliega el frontend quedando lista la aplicación en GCP, tal como se ilustra en la Fig. 32.



```
Beginning deployment of service [default]...
#=====#
#= Uploading 0 files to Google Cloud Storage      =#
#=====#
File upload done.
Updating service [default]...done.
Setting traffic split for service [default]...done.
Deployed service [default] to [https://kea-frontend.uc.r.appspot.com]

You can stream logs from the command line by running:
$ gcloud app logs tail -s default

To view your application in the web browser run:
$ gcloud app browse --project=kea-frontend
```

Fig. 32. Deploy frontend en GCP

Anexo 6. Certificado de traducción al idioma inglés

Certificación

Ciudad, 30 de abril del 2023

Yo, traductora pública María Florencia Presta, portadora de DNI argentino 32.947.135, certifico que la traducción al inglés del resumen del Trabajo de Titulación denominado: **Análisis de plataformas de computación en la nube para el desarrollo de aplicaciones web bajo una arquitectura cliente-servidor**, perteneciente al Sr. **Pedro Fernando Aponte Rueda**, corresponde al texto original en español.

Atentamente,
María Florencia Presta
Traductora Pública de Inglés
Mat. XXI F°061-C.T.P.C.B.A. – Nro. 8173

PRESTA
Maria
Florencia



Firmado digitalmente por
PRESTA Maria
Florencia
Fecha: 2023.05.02
15:50:36 -03'00'