

# **ESCUELA POLITÉCNICA NACIONAL**

**FACULTAD DE INGENIERÍA ELÉCTRICA Y  
ELECTRÓNICA**

**ESTUDIO COMPARATIVO DE ARQUITECTURAS SERVERLESS  
COMPUTING DE DIFERENTES PROVEEDORES DE NUBE  
ORIENTADO A LA CREACIÓN DE UNA APLICACIÓN WEB**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERA EN  
TECNOLOGÍAS DE LA INFORMACIÓN**

**KATHERINE LIZBETH VALDIVIESO GAVILANEZ**

**[katherine.valdivieso@epn.edu.ec](mailto:katherine.valdivieso@epn.edu.ec)**

**DIRECTOR: WILLIAM FERNANDO FLORES CIFUENTES**

**[fernando.flores@epn.edu.ec](mailto:fernando.flores@epn.edu.ec)**

**DMQ, febrero 2022**

## **CERTIFICACIONES**

Yo, KATHERINE LIZBETH VALDIVIESO GAVILANEZ declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

\_\_\_\_\_  
**NOMBRE\_ESTUDIANTE**

Certifico que el presente trabajo de integración curricular fue desarrollado por KATHERINE LIZBETH VALDIVIESO GAVILANEZ, bajo mi supervisión.

\_\_\_\_\_  
**NOMBRE\_DIRECTOR**  
**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

KATHERINE LIZBETH VALDIVIESO GAVILANEZ

WILLIAM FERNANDO FLORES CIFUENTES

## **DEDICATORIA**

A Dios y a mi madre que significan todo en mi vida.

## **AGRADECIMIENTO**

En primer lugar, a Dios por ser la guía día a día en mi vida.

A mis abuelitos, Papinito y Maminita a quienes considero mis padres por darme tanto apoyo y creer en mí.

A mi madre y hermano, Alexi y Kevin por ser las personas que amo y me han acompañado en los momentos buenos y malos en el transcurso de la vida.

A mi padre, Hernán por tener un cariño especial hacía mí y darme la fortaleza que actualmente poseo.

A mis tíos y primos, por su apoyo en cada paso que he dado en mi vida.

A mis amigos de la universidad, por su apoyo incondicional y ser quiénes siempre me brindaron su ayuda para cumplir metas personales y profesionales.

A mi tutor, Ing. Fernando Flores gracias por toda su confianza y guía en el desarrollo de este proyecto.

## ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VII
ABSTRACT .....	VII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO .....	1
1.1. OBJETIVO GENERAL .....	2
1.2. OBJETIVOS ESPECÍFICOS .....	2
1.3. ALCANCE .....	2
1.4. MARCO TEÓRICO.....	3
1.4.1. INTRODUCCIÓN.....	3
1.4.2. CLOUD COMPUTING .....	4
1.4.3. PROVEEDORES DE NUBE .....	4
1.4.4. SERVERLESS COMPUTING .....	4
1.4.5. FASS (FUNCTION AS A SERVICE).....	6
1.4.6. APLICACIONES WEB .....	7
1.4.7. FRONTEND O CONTENIDO ESTÁTICO.....	8
1.4.8. BACKEND O CONTENIDO DINÁMICO .....	8
2 METODOLOGÍA.....	10
2.1. PROCESAMIENTO DE LA INFORMACIÓN .....	11
2.1.1. LEVANTAMIENTO DE LA INFORMACIÓN.....	11
2.2. DISEÑO DE LA APLICACIÓN.....	13
2.2.1. ARQUITECTURA DE LA APLICACIÓN WEB SERVERLESS COMPUTING.....	14
2.3. COSTO DE LA APLICACIÓN WEB .....	23
2.3.1. MODELOS COMERCIALES DE PROVEEDORES DE NUBE .....	23
2.3.2. AMAZON WEB SERVICES .....	24
2.3.3. GOOGLE CLOUD PLATFORM .....	30
2.3.4. MICROSOFT AZURE .....	34
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	38

3.1.	COSTOS FINALES DEL MODELO DE APLICACIÓN WEB CREADO BAJO LA ARQUITECTURA SERVERLESS COMPUTING.....	39
3.2.	ANÁLISIS COMPARATIVO ARQUITECTURAS SERVERLESS COMPUTING .....	41
3.3.	CONCLUSIONES.....	45
3.4.	RECOMENDACIONES .....	47
4	REFERENCIAS BIBLIOGRÁFICAS.....	48

## RESUMEN

En la actualidad existe un número incalculable de tecnologías nuevas, cada avance tecnológico surge a partir de la necesidad. Empresas o proveedores presentan soluciones y servicios alojados en su nube, llamados también servicios de Cloud Computing.

El objetivo de la investigación es realizar un análisis comparativo de tres plataformas que facilitan un entorno de desarrollo e implementación de la arquitectura Serverless Computing, creadas por Amazon, Google y Microsoft, proveedores que hoy en día dominan el mercado de los servicios en la nube.

Serverless Computing, es una arquitectura que proporciona una autogestión de servidores, siendo el proveedor quién administra y el usuario o desarrollador únicamente quién se encarga del desarrollo del software correspondiente.

El presente documento se divide en cuatro secciones:

En la primera sección se presenta la descripción del componente, los objetivos, alcance y el marco conceptual de la base en la que se sustenta la investigación.

En la segunda sección se describe los requerimientos, el diseño y presentación de costos para la creación de una aplicación web alojada en las plataformas Amazon Web Services, Azure Cloud Platform, Google Cloud Platform, bajo la arquitectura Serverless Computing.

En la tercera sección se presentan los resultados del análisis de costos de implementación de la aplicación web, conclusiones y recomendaciones del proyecto de investigación.

En la cuarta sección se adjunta referencias bibliográficas de las fuentes que permitieron el desarrollo del documento.

**PALABRAS CLAVE:** Cloud Computing, Serverless Computing, proveedores, arquitectura, aplicación web, costos.

## **ABSTRACT**

Today there are an incalculable number of new technologies, each technological advance arises from necessity. Companies or providers present solutions and services hosted in their cloud, also called Cloud Computing services.

The objective of the research is to carry out a comparative analysis of three platforms that facilitate an environment for the development and implementation of the Serverless Computing architecture, created by Amazon, Google and Microsoft, providers that today dominate the cloud services market.

Serverless Computing is an architecture that provides self-management of servers, with the provider managing and the user or developer solely in charge of developing the corresponding software.

This document is divided into four sections:

The first section presents the description of the components, the objectives, the scope and the conceptual framework of the basis on which the research is based.

The second section describes the requirements, the design and presentation of costs for the creation of a web application hosted on the Amazon Web Services, Azure Cloud Platform, Google Cloud Platform platforms, under the Serverless Computing architecture.

The third section presents the results of the cost analysis of the implementation of the web application, conclusions and recommendations of the research project.

In the fourth section, bibliographic references of the sources that allowed the development of the document are attached.

**KEYWORDS:** Cloud Computing, Serverless Computing, providers, architecture, web application, costs.

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Tradicionalmente el desarrollo de aplicaciones ha generado altos costos de implementación considerando la gestión y mantenimiento que conlleva un servidor, actualmente existen proveedores de nube que ofrecen soluciones Serverless.

Serverless Computing es una arquitectura que facilita el desarrollo de una aplicación, considerando costos reducidos al no preocuparse por servidores, sistemas operativos, hardware que revisar o actualizar ya que es el proveedor el encargado de suministrarlos, administrarlos y gestionarlos, de esta manera la creación y ejecución de aplicaciones se reduce al desarrollo de código únicamente.

La arquitectura, aunque se ha concebido con el término “sin servidor” cuenta con servidores físicos y virtuales, que permiten la ejecución de funciones a través de respuesta a eventos, generando sin número de beneficios como escalamiento, suministro ágil de recursos, los cobros se realizan en función de lo utilizado y se maneja gran tolerancia a errores debido a la infraestructura flexible.

Hoy en día se conoce que los proveedores de nube ofrecen infinidad de servicios entre ellos bases de datos, control de usuarios, y demás componentes para el desarrollo de una aplicación, sin embargo los modelos de suscripción y uso de cada uno de ellos están en función de la capacidad de almacenamiento y funciones que se requiera utilizar, por esto es fundamental identificar y elegir un proveedor que se acoja al desarrollo de una aplicación bajo la arquitectura conocida como Serverless de acuerdo a los requerimientos del usuario y la lógica de la aplicación ya que una de las partes importantes es la dependencia que genera el proveedor al emplear esta arquitectura.

De lo anterior parte la idea de establecer un análisis comparativo de costos de implementación de la arquitectura Serverless Computing en tres diferentes plataformas de proveedores de nube.

Serverless Computing permite el desarrollo de sin número de aplicaciones, sin embargo, se ha enfocado el estudio al modelo de una aplicación web utilizando los servicios propios de cada una de las plataformas.

El componente está compuesto por cuatro secciones.

En la primera sección se conceptualiza todo término referido al desarrollo de una aplicación web bajo una arquitectura Serverless.

En la segunda sección se identifica los requerimientos para la creación de una aplicación web, el diseño de la misma y los costos que genera su implementación en cada una de las plataformas.

En la tercera sección de resultados se presenta la información relevante obtenida en la investigación, estableciendo un análisis comparativo de la implementación de una aplicación web bajo la arquitectura Serverless Computing considerando ventajas y desventajas entre los diferentes servicios de proveedores de nube. Además, se establecen conclusiones y algunas recomendaciones sobre la investigación realizada y desarrollos futuros, que darán continuidad a esta línea de investigación.

Finalmente, en la cuarta sección se adjunta referencias bibliográficas de libros, sitios web y documentación que permitieron la investigación y desarrollo del proyecto.

## **1.1. Objetivo general**

Elaborar un estudio comparativo de arquitecturas Serverless Computing de diferentes proveedores de nube orientado a la creación de una aplicación web.

## **1.2. Objetivos específicos**

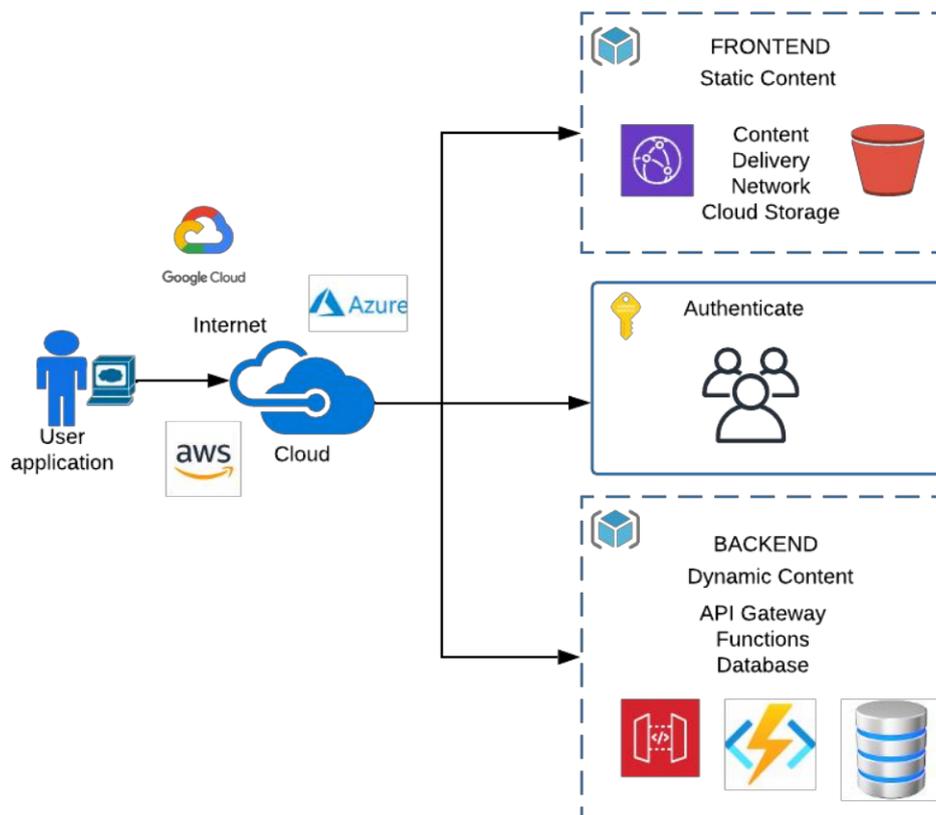
2. Definir la arquitectura Serverless Computing.
3. Estudiar proveedores de nube que proporcionen la arquitectura.
4. Diseñar los componentes que tendrá la aplicación web y describir los servicios que ofrecen diferentes plataformas para su implementación.
5. Desarrollar un análisis comparativo de las arquitecturas seleccionadas y establecer la mejor solución en función de costos.

## **1.3. Alcance**

En este proyecto se plantea estudiar y comprender el concepto y casos de uso de la arquitectura Serverless Computing. A su vez determinar los beneficios que ofrece para el despliegue de una aplicación web.

Para llevar a cabo este proyecto, se realizará un estudio de tres proveedores de nube que ofrecen soluciones Serverless Computing y el conjunto de servicios necesarios para la correcta implementación de una aplicación web.

Para el diseño de la aplicación web se hará uso de los servicios que ofrecen cada una de las plataformas con la finalidad de facilitar la creación en la nube de forma rápida y rentable, como se observa en la Figura 1.1.



**Figura 1. 1.** Modelo de Aplicación Web Serverless Computing

Se realizará una presentación de los costos que genera la implementación de una aplicación web en cada una de las plataformas.

Finalmente se presentará un análisis comparativo de los servicios que ofrecen diferentes proveedores de nube para el desarrollo de una aplicación web de las diferentes soluciones considerando mejores costos y beneficios.

## 1.4. Marco teórico

### 1.4.1. Introducción

La ingeniería y la arquitectura son campos relativamente incipientes en el software, ya que las cosas están cambiando todo el tiempo. En los años 1990s, las organizaciones solían adquirir y mantener su propio hardware y crear topologías de red locales. Mirar hacia el futuro implica escalar, diseñar un espacio en el que se va a construir software aun cuando quede atrás patrones de diseño de software tradicionales debido a que estas estrategias se vuelven complejas para cumplir los amplios requisitos de empresas que desean romper barreras geográficas a través de transformaciones digitales [1].

### **1.4.2. Cloud Computing**

Cloud computing o computación en la nube, es un modelo de computación que permite al proveedor tecnológico ofrecer servicios a través de internet. La prestación de servicios permite al cliente el acceso bajo demanda y a través de la red a un conjunto de recursos compartidos y configurables (redes, servidores, almacenamiento, aplicaciones y servicios) que pueden ser rápidamente asignados y liberados con una mínima gestión por parte del proveedor. En resumen, permite acceder a los servicios y recursos contratados proporcionando flexibilidad de dimensionamiento y acceso [2].

La computación en la nube proporciona soluciones y servicios confiables, escalables, seguros, rentables y sofisticados que ayudan a las empresas a adaptarse, actualizarse y modernizarse rápidamente; resolviendo los problemas típicos de hardware desde la adquisición hasta el mantenimiento [1].

### **1.4.3. Proveedores de nube**

Los proveedores de servicios en la nube son compañías externas que ofrecen servicios de plataforma, infraestructura, aplicaciones o almacenamiento basados en la nube. Las características a analizar para la selección de un proveedor de nube son:

Analizar solidez financiera del negocio, debe ser una organización de confianza y contar con conocimiento empresarial y técnico.

Identificar soporte administrativo, evaluar contrato a nivel de servicio, revisar informes de rendimiento, facturación, contabilidad, precios y modelos de suscripción.

Verificar funcionalidad y procesos técnicos como la facilidad de implementación, interfaces, administración de cambios y eventos integrados en su sistema.

Debe poseer prácticas y directivas de seguridad en su infraestructura, administración de identidades, copias de seguridad y retención de los datos además de seguridad física [3].

### **1.4.4. Serverless Computing**

#### **1.4.4.1. Introducción**

La computación sin servidor es uno de los últimos modelos de computación en la nube que ha cobrado impulso en los últimos años, ayudando a los desarrolladores a centrarse en implementar una lógica de aplicación mediante código sin preocuparse por detalles de gestión y administración de servidores [1].

#### **1.4.4.2. Características principales**

En este apartado se explica brevemente las características principales de la arquitectura Serverless.

- **Sin administración de servidores**

La administración de los servidores o los clústeres no es necesario, por lo tanto, no existe aprovisionamiento, gestión y mantenimiento de servidores. La infraestructura no es un tema del cual se debe preocupar, ya que está delegado al proveedor [4].

- **Escalabilidad**

La escalabilidad en arquitecturas Serverless Computing representa una de las principales ventajas frente al desarrollo de aplicaciones de forma tradicional, ya que la capacidad incrementará basado en el uso, ajustándose de esta manera continuamente al volumen de datos que contenga [4].

- **Disponibilidad**

Las respuestas son automáticas, las solicitudes pueden ser una docena o millones y la capacidad de respuesta seguirá siendo inmediata.

- **Bajo demanda y pago por uso**

Funcionamiento basado en el uso, la facturación se la realiza de acuerdo al rendimiento uniforme o duración de ejecución en lugar de hacerlo por unidad de servidor. De esta manera sólo se paga por los recursos que se consumen. Serverless Computing permite un uso real de los recursos por los que se paga [4].

#### **1.4.4.3. Casos de Uso**

Las aplicaciones sin servidor generalmente se crean utilizando servicios completamente administrados, las soluciones sin servidor están al centro de la mayoría de las aplicaciones, respaldados por servicios de bases de datos, autenticación, almacenamiento, entre otros [4].

Serverless Computing dada su particularidad tiene sin número de escenarios en los que la tecnología se puede implementar, algunos ejemplos y principales casos de uso se mencionan a continuación:

- Aplicaciones Web
- Aplicaciones móviles

- IOT (Internet of things)
- Procesamiento de datos (macrodatos)
- Juegos de realidad aumentada
- Mensajería [5]

#### **1.4.4.4. Definición**

Los entornos Cloud facilitan la creación de aplicaciones bajo la arquitectura “Serverless” porque se da mayor importancia al código que a la infraestructura. La computación sin servidor es un modelo de ejecución de código donde la lógica del lado del servidor se ejecuta en contenedores informáticos, activados por eventos que son completamente administrados por un tercero [6].

Estas arquitecturas eliminan la necesidad de sistemas en ejecución continuos por dirigidos a eventos, generando aplicaciones totalmente funcionales y escalables. La viabilidad de crear aplicaciones dentro de esta arquitectura es enorme, simplemente escribir código, cargarlo y ejecutarlo, esto no significa que no requiera servidores, pero serán los proveedores de nube los encargados de administrarlos [7].

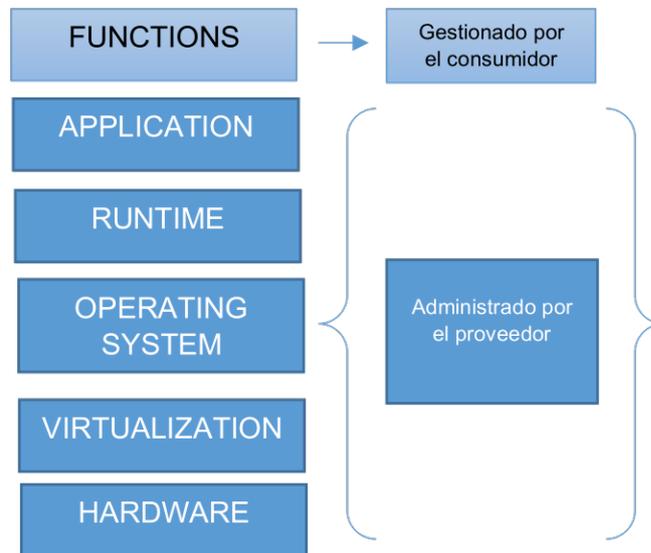
#### **1.4.5. FasS (Function as a service).**

FasS permite desarrollar únicamente la funcionalidad de una aplicación mediante la creación de funciones, sin considerar infraestructura subyacente o configuración de servidores, es muy útil para crear aplicaciones sin servidor [8].

Las funciones como servicio son consideradas parte de la arquitectura informática basada en eventos siendo el núcleo de lo que se denomina arquitectura sin servidor, dado que las funciones desencadenan por un evento específico como por ejemplo solicitudes HTTP. [9]

FasS es una categoría de servicios de computación en la nube que actualmente es posible efectuar en el desarrollo de aplicaciones y sistemas sin servidor [7].

Para alojar la aplicación en la nube, se necesita algún tipo de infraestructura esto implica la gestión del hardware, servidores, virtualización, sistemas operativos, etc. Las funciones como servicio o FasS permite la gestión de todos los componentes por el proveedor y no por el consumidor. El consumidor únicamente se encarga de la gestión y desarrollo de las funciones para construir la aplicación como se puede observar en la figura 1.2.



**Figura 1. 2.** FasS (Function as a Service)

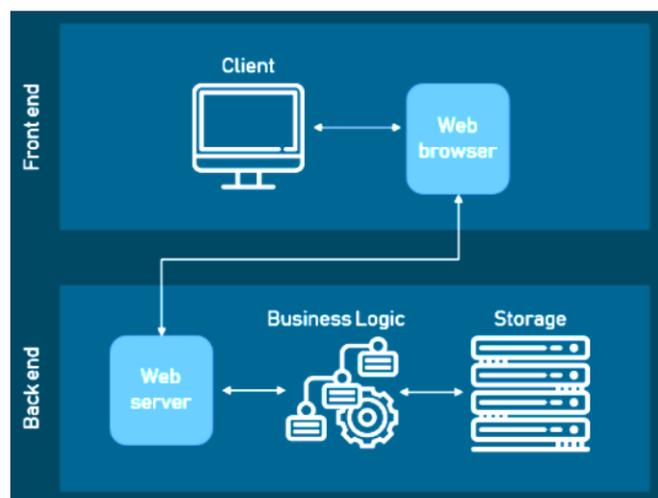
### 1.4.6. Aplicaciones Web

Una aplicación web bajo la arquitectura Serverless Computing necesita servicios que faciliten su ejecución, componentes web que participen en el funcionamiento correcto de la propia aplicación web.

#### 1.4.6.1. Contenido Estático y Dinámico

Uno de los aspectos más importantes a considerar del proveedor de servicios a elegir es el nivel de servicios y soporte que ofrece para servir contenido estático y dinámico como parte de la arquitectura Serverless siendo uno de los puntos más críticos en su elección.

El contenido estático básicamente es el frontend y el contenido dinámico el backend de una aplicación, tal como se muestra en la figura 1.3.



**Figura 1. 3.** Frontend y Backend

### 1.4.7. Frontend o Contenido Estático

En una aplicación web el frontend es aquello que el usuario ve y con lo que puede interactuar usando un navegador. La interfaz gráfica integra los lenguajes de maquetación HTML (HyperText markup Language) para la estructura y contenido básico de la página, CSS (Cascading Style Sheets) para diseño, y JavaScript que permite la lógica de la aplicación, al recibir solicitudes de usuarios y enviarlos al backend.

En resumen, el contenido y el diseño de las aplicaciones pertenecen al frontend, para llevar a cabo la creación del frontend utilizando la arquitectura serverless es necesario identificar servicios que permitirán alojar contenido estático y mejorar el rendimiento de la aplicación como se puede observar en la tabla 1.1.

**Tabla 1. 1.** Servicios de Frontend (Contenido estático)

<b>Servicio</b>	<b>Función</b>
Content Delivery Network	Distribución de contenido estático
Cloud Storage	Alojar contenido estático

#### 1.4.7.1. Content Delivery Network o Red de entrega de contenido (CDN)

Este servicio forma parte del frontend para interactuar con el usuario al permitir una mejor administración del tráfico al procesar las solicitudes de los usuarios más rápido, resultando una experiencia general más satisfactoria.

Permite entregar contenido web estático de manera global, eficiente y confiable, a través de una red global de ubicaciones de caché. Las solicitudes de contenido se dirigen automáticamente a la ubicación más cercana, para que el contenido se entregue con el mejor rendimiento posible [12].

#### 1.4.7.2. Cloud Storage

El almacenamiento en la nube es necesario para guardar cualquier cantidad de datos y recuperarlos con la frecuencia que se desee, además proporciona funciones de gestión para que pueda optimizar, organizar y configurar el acceso a los datos para satisfacer los requisitos empresariales, organizativos y de conformidad específicos de usuarios [10].

### 1.4.8. Backend o Contenido Dinámico

En una aplicación web el backend es la parte que contiene la lógica que es transparente al usuario permite combinar la base de datos para implementar alguna lógica de código e interactuar con demás servicios y componentes de la aplicación.

La arquitectura sin servidor permite la creación de código a través de funciones, en la tabla 1.2 se presentan los servicios fundamentales para la implementación del backend de una aplicación web.

**Tabla 1. 2.** Servicios de Backend (Contenido dinámico)

Servicio	Función
Authenticate	Control de usuarios
API Gateway	Puerta de enlace
Functions (FaaS)	Solución serverless, ejecución de código
Database	Almacenamiento

#### **1.4.8.1. Autenticación y Seguridad**

El control de acceso a través de autenticación y autorización requiere confirmar dos cosas: la identidad de la persona que intenta acceder a la aplicación web, y si realmente tiene permiso para hacerlo. Ambos aspectos son fundamentales para garantizar que las organizaciones puedan mantener sus redes y recursos protegidos a salvo de los malos actores [11].

#### **1.4.8.2. Administración de API Gateway**

API Gateway toma todas las llamadas API de los clientes y realiza el enrutamiento de solicitudes mediante la composición y la traducción de protocolos. Adicional, permite desarrollar, implementar, asegurar y administrar las API con una puerta de enlace totalmente administrada y admite cargas de trabajo en aplicaciones sin servidor.

El modo de trabajo es invocar funciones sincrónicamente con un evento de una solicitud HTTP. API Gateway espera una respuesta de la función y transmite el resultado a la persona que llama. En las integraciones personalizadas, debe definir una respuesta de integración y una respuesta de método para convertir la salida de la función en una respuesta HTTP. En las integraciones de proxy, la función debe responder con una representación de la respuesta en un formato específico [12].

### **Llamadas API**

Las aplicaciones modernas suelen ser sistemas distribuidos que consisten en un frontend y un backend. La interfaz de usuario muestra iconos, botones, gráficos. El backend almacena y entrega datos. Los datos, que son administrados por los sistemas backend, deben entregarse a los sistemas frontend. Para gestionar la comunicación entre frontend y

backend, se introduce una API, abreviatura de interfaz de programación de aplicaciones. Bajo este concepto, el backend sirve a la API, el frontend consume el API.

#### **1.4.8.3. Functions (FaaS)**

FaaS es el núcleo o motor de Serverless Computing a través de este servicio es posible crear funciones autónomas para una tarea en particular que generalmente se desencadenan por tipos de eventos y activan servicios de autenticación, mensajes, etc., o procesos en los que interactúan con otras herramientas de los proveedores de nube [13].

#### **1.4.8.4. Bases de Datos**

Las bases de datos sin servidor son un requisito previo para la informática sin servidor. Está especialmente diseñado para cargas de trabajo rápidas e impredecibles. La informática sin servidor siempre requiere un servidor, y se necesita una base de datos sin servidor para ejecutar esos servidores.

#### **Bases de Datos No Relacionales**

Las bases de datos no relacionales o NoSQL están diseñadas específicamente para modelos de datos específicos y tienen esquemas flexibles para crear aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala [14] .

## **2 METODOLOGÍA**

Este proyecto de investigación es de tipo analítica y documental, al permitir analizar e interpretar información obtenida de varias fuentes.

Esta parte del proyecto abarca tres secciones. En la primera sección se realiza el procesamiento de información de los diferentes proveedores de nube que proporcionan arquitecturas Serverless y la elección de tres plataformas. En la segunda sección se tiene un esquema de aplicación web bajo la arquitectura Serverless Computing. En la tercera sección se realiza una presentación de los costos que genera la implementación de una aplicación web en cada una de las plataformas.

La investigación tiene como enfoque identificar diversos servicios que actualmente ofrecen proveedores de nube que ayudarán de manera eficaz al desarrollo de una aplicación web enfocado a una arquitectura Serverless.

## **2.1. Procesamiento de la Información**

En esta sección se realiza un análisis de proveedores de nube que ofrecen soluciones Serverless, para ello se realiza un estudio de las diferentes plataformas en el mercado y selección de tres alternativas.

### **2.1.1. Levantamiento de la información**

Actualmente, la mayoría de los proveedores de nube pública ofrecen Serverless Computing, en la tabla 2.1 se presentan un resumen de proveedores de nube con su respectivo servicio informático basado en eventos y sin servidor.

Amazon Web Services (AWS) popularizó la computación sin servidor con AWS Lambda en el año 2014. Más tarde, Google lanzó silenciosamente su Google Cloud Function e IBM lanzó IBM Cloud Functions en 2016. Microsoft lanzó functions de Azure poco después y Cloudflare también ingreso al mercado con sus Cloudflare Workers en 2017 [1].

La tecnología sin servidor generalmente se considera más rentable que la asignación de una cantidad fija de servidores, lo que generalmente implica periodos significativos de tiempo de inactividad o infrautilización.

Básicamente, esto significa que todo lo que se necesita hacer es escribir código o funciones que realizan una tarea muy específica, seleccionar la cantidad de recursos necesarios para ejecutar el código y enviarlo al servicio de computación en la nube sin servidor para que se ejecute. El servicio a su vez garantiza que el código acoja la cantidad requerida de memoria y ciclos de CPU que necesita para ejecutarse [15].

Serverless puede ser más rentable para empresas medianas y pequeñas al ser considerados grupos de escalado automático.

Todo el mercado de servicios de infraestructura en la nube tiene una industria de ciento once mil millones con AWS liderando con 33% de participación, seguido por Azure con 18% y Google Cloud Platform con 9%.

Siendo la tecnología sin servidor un paradigma nuevo, es interesante identificar que proveedor posee ciertas características para la implementación de una aplicación web como la ampliación y reducción automática de servicios, la carga de recursos solo cuando el código se ejecuta, además de los servicios gratuitos y pagos en función de los recursos utilizados por parte de los proveedores de nube.

Específicamente, en este proyecto se realizará un análisis de tres plataformas en la nube de diferentes proveedores: Amazon AWS, Microsoft Azure y Google Cloud Platform (GCP).

**Tabla 2. 1.** Proveedores de Nube y Plataformas Serverless

<b>Proveedor Cloud</b>	<b>Plataforma Cloud</b>	<b>Functions - Serverless</b>
Amazon	Amazon Web Services (AWS)	AWS Lambda
Google	Google Cloud Platform (GCP)	Google Cloud Functions
Microsoft	Azure Cloud Platform	Azure Functions

#### **2.1.1.1. Amazon AWS**

Amazon Web Services (AWS) es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 200 servicios integrales de centros de datos a nivel global. Millones de clientes, incluso las empresas emergentes que crecen más rápido, las compañías más grandes y los organismos gubernamentales líderes, están usando AWS para reducir los costos, aumentar su agilidad e innovar de forma más rápida [16].

- **AWS LAMBDA**

AWS Lambda se presentó por primera vez en el 2014, Lambda es un servicio de cómputo simple que ejecuta código en respuesta a eventos. Al desarrollar la aplicación web el código debe escribirse específicamente como funciones, cada función de Lambda se puede invocar bajo demanda o invocarse dinámicamente en función de ciertos tipos de eventos admitidos. Algunos eventos se activan cuando se realiza alguna acción con los objetos que intervienen tanto en funciones como en servicios complementarios utilizados en la aplicación web [15].

Lambda se encargará internamente de aprovisionar y administrar los recursos de la infraestructura subyacente, asegurándose de que el código se implemente correctamente. Lambda se ocupa incluso de la escalabilidad y alta disponibilidad.

#### **2.1.1.2. Google Cloud Platform (GCP)**

Google Cloud Platform (GCP) es una colección de productos de software y hardware que permite que el mundo use parte de la infraestructura interna de Google, esta colección incluye muchas cosas que son comunes a todos los proveedores de la nube como máquinas virtuales, servidores, almacenamiento y aunque GCP es similar a otros proveedores de nube opera a tal escala que tiene muchas ventajas económicas que se transmiten en forma de precios más bajos ya que posee una inmensa infraestructura física lo que permite creación de aplicaciones a precios más bajos combinado con un rendimiento mejorado [17].

- **CLOUD FUNCTIONS**

Google Cloud Functions entro en el campo de las funciones como servicio en febrero de 2016 con el lanzamiento de funciones en la nube [18]. La plataforma ofrece una gran cantidad de utilidades para la solución de problemas, uno de los principales casos de uso es la creación de aplicaciones con servicios establecidos por usuarios, pero administrados por terceros. Cloud Functions incluye un paquete de opciones, dependencias con los demás servicios internos de GCP.

### **2.1.1.3. Azure Cloud Platform**

Azure Cloud Platform incluye muchos servicios, esta plataforma informática cubre una amplia gama de servicios fundamentales de la computación en la nube. Esto incluye servicios que se puede usar para ayudar a construir y operar aplicaciones considerando reducción de costos operativos, el aumento de agilidad, el desarrollo de mejores estrategias, el acceso al almacenamiento ilimitado y responsabilidad externa de las actualizaciones de hardware. Aunque es un servicio relativamente nuevo ha avanzado a pasos agigantados en términos de capacidades y ofertas [19].

- **AZURE FUNCTIONS**

Azure Functions es un servicio en la nube disponible bajo demanda que proporciona toda la infraestructura y los recursos continuamente actualizados necesarios para ejecutar aplicaciones [20]. El enfoque será desarrollar el código y Functions se encarga del resto. Functions proporciona computación sin servidor para Azure. Puede usar Funciones para crear API web, responder a cambios en la base de datos, procesar flujos de datos, administrar colas de mensajes y más.

## **2.2. Diseño de la Aplicación**

Para realizar el diseño de la aplicación web se consideró implementar servicios similares de cada plataforma de los proveedores de nube seleccionados con la finalidad de permitir realizar un análisis comparativo con mayor precisión, el usuario accede a través de un navegador a internet e interactúa con el frontend de la aplicación, posterior a ello se utiliza un servicio de autenticación para el control de usuarios y el backend que permite estructurar la lógica y el código se lo realiza a través de funciones utilizando la tecnología Serverless Computing de cada proveedor que a su vez permitirá la interacción con los demás servicios y un óptimo rendimiento de la aplicación para satisfacción del usuario.

## 2.2.1. Arquitectura de la Aplicación Web Serverless Computing

### 2.2.1.1. Serverless Computing en AWS

En la figura 2.1 se puede visualizar la estructura de la aplicación web en AWS y en la tabla 2.2 se detalla el servicio al que pertenece cada componente que forma parte de la aplicación.

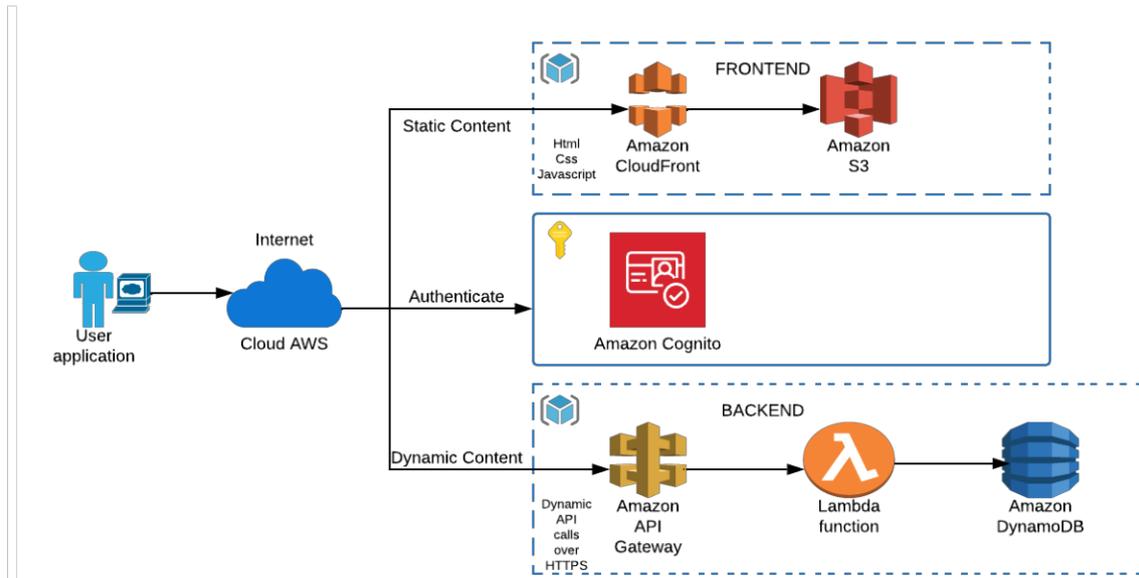


Figura 2. 1. Diagrama de aplicación web en AWS

Tabla 2. 2. Servicios de AWS

Servicio	Amazon Web Service
Content Delivery Network	Amazon Cloudfront
Cloud Storage	Amazon S3
Authenticate	Amazon Cognito
API Gateway	Amazon API Gateway
Functions (FaaS)	AWS Lambda
Database	Amazon DynamoDB

### Amazon Cloudfront

Amazon CloudFront es un servicio web que agiliza la distribución de contenido web estático como archivos .html, .css, .js y archivos de imágenes a los usuarios. CloudFront entrega el contenido a través de una red mundial de centros de datos que reciben el nombre de

ubicaciones de borde. Cuando un usuario solicita contenido que se distribuye con CloudFront, la solicitud se redirige a la ubicación de borde que ofrece la mínima latencia (retraso de tiempo), de modo que el contenido se entregue con el mejor rendimiento posible [20].

- Si el contenido ya se encuentra en la ubicación de borde con menor latencia, CloudFront lo entrega inmediatamente.
- Si el contenido no se encuentra en dicha ubicación de borde, CloudFront lo recupera de un origen que haya definido como, por ejemplo, un bucket de Amazon S3, un canal de MediaPackage o un servidor HTTP (por ejemplo, un servidor web) que se haya definido como origen de la versión definitiva del contenido.

### **Amazon S3**

Las funciones Lambda de Amazon S3 se pueden activar cuando se crea, actualiza o elimina un objeto. Amazon S3 aloja activos estáticos de aplicación Web y la entrega de manera segura a través de CloudFront.

S3 es un servicio de almacenamiento simple diseñado para almacenar datos y es una de las ofertas de AWS más populares con precios flexibles. Es muy fiable y ofrece un alto rendimiento a gran escala. Uno de los aspectos más importantes de S3 es que sólo paga por el almacenamiento utilizado, no por el provisionado.

S3 es donde se almacenará los archivos del sitio estático: html, css, js, imágenes, etc. S3 requiere que cree un depósito, que actúa como un directorio para almacenar sus archivos estáticos [21].

### **Amazon Cognito**

Amazon Cognito recopila los atributos del perfil de un usuario en directorios denominados grupos de usuarios que utiliza una aplicación web para configurar el acceso limitado a los recursos de AWS.

Es posible configurar Amazon Cognito para aceptar flujos de eventos a medida que se actualizan y sincronizan los datos. De esta manera Cognito permite:

- Agregar fácilmente registro de usuario, inicio de sesión y control de acceso a sus aplicaciones con su interfaz de usuario integrada (UI ) y fácil configuración
- Identidades federadas de proveedores de identidades sociales
- Sincronizar datos en múltiples dispositivos y aplicaciones

- Proporcione acceso seguro a otros servicios de AWS desde su aplicación definiendo roles y asignando usuarios a diferentes roles

Cada usuario puede guardar un máximo de 20 MB de datos, y cada conjunto de datos individual contiene hasta 1 MB.

### **Amazon API Gateway**

Se trata de un servicio que permite crear, publicar, proteger y monitorear APIs en cualquier escala. El mismo se encarga de todas las tareas necesarias para aceptar y procesar cientos de miles de llamadas concurrentes de API, solamente pagando de acuerdo a su uso (llamadas de API + tráfico saliente) [22].

Algunos de los beneficios de API Gateway:

- Integración: permite crear APIs integradas con diferentes tipos de back-ends, como funciones AWS Lambda, otros servicios AWS.
- Seguridad: es posible implementar autenticación y control de acceso a través de funcionalidades existentes de AWS, como Amazon Cognito.
- Escalabilidad: además de escalar automáticamente, también es posible definir reglas de throttling para limitar la cantidad de llamadas que sus APIs van a aceptar, y así evitar sobrecargar sus servidores de backend.
- Visibilidad: API Gateway provee métricas sobre la cantidad de llamadas, latencia, tasa de errores sobre sus APIs, así como también logs para facilitar las tareas de debugging.

### **AWS Lambda**

Servicio informático sin servidor y basado en eventos es la más popular y tiene algunas características relevantes. Lambda admite varios lenguajes de programación como:

- Node.js
- Go
- Java
- Python
- C+

A continuación, se mencionan algunas de las características relevantes de AWS Lambda:

- Cantidad de memoria asignada al código.
- Un evento le dice a la función cuando se debe ejecutar.
- Proporciona una API para ejecutar la función.
- Proporciona capacidades de registro, monitoreo. Identificar el tiempo en milisegundos que demora en ejecutar.
- Abstracción
- Modelo de seguridad para configurar acceso y autenticación.
- Modelo de precios, pago en función del uso.
- La integración con el ecosistema de servicios de AWS.

## **Amazon DynamoDB**

Las funciones Lambda se activan cuando se realizan actualizaciones en una tabla particular de DynamoDB, como la inserción de filas, eliminación, etc.

DynamoDB es una base de datos NoSQL alojada que ofrece:

- Rendimiento fiable incluso a medida que escala
- Una experiencia administrada, por lo que no tendrá que usar SSH en los servidores para actualizar las bibliotecas criptográficas
- Una API pequeña y simple que permite un acceso simple de clave-valor, así como patrones de consulta más avanzados.

DynamoDB es particularmente adecuado para los siguientes casos de uso:

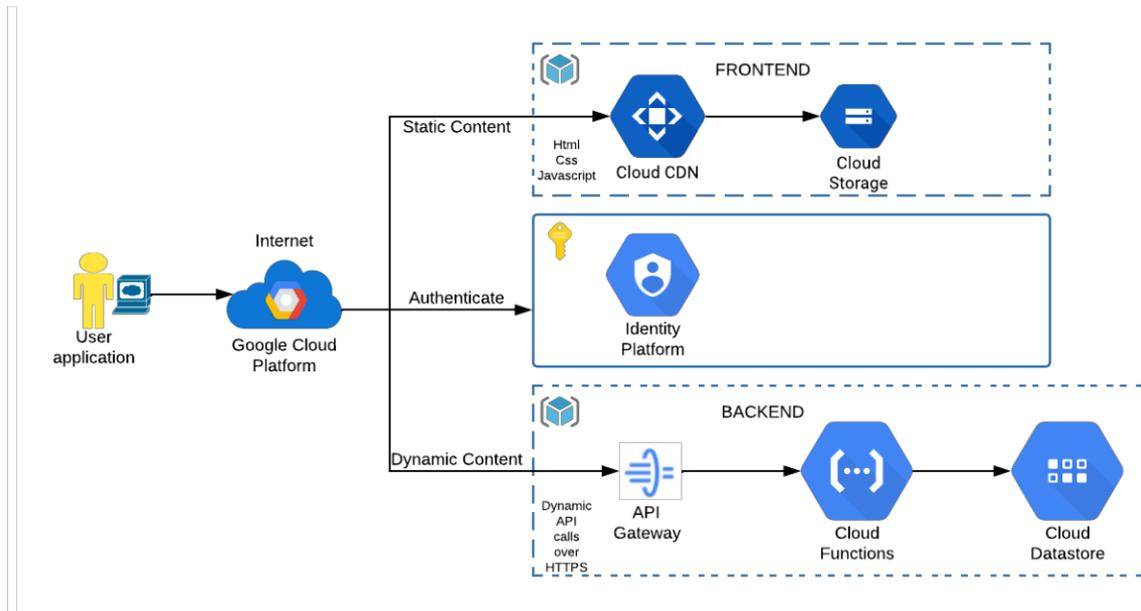
Aplicaciones con grandes cantidades de datos y estrictos requisitos de latencia. Con DynamoDB, sus consultas tienen una latencia predecible de hasta cualquier tamaño, incluidos más de 100 TB.

Aplicaciones sin servidor que utilizan AWS Lambda. AWS Lambda proporciona computación efímera, sin estado y de escalado automático en respuesta a activadores de eventos. Se puede acceder a DynamoDB a través de una API HTTP y realiza la autenticación y la autorización a través de roles de IAM, lo que lo convierte en la opción perfecta para crear aplicaciones sin servidor.

Conjuntos de datos con patrones de acceso simple y conocido. Si está generando recomendaciones y sirviéndolas a los usuarios, los sencillos patrones de acceso de clave-valor de DynamoDB lo convierten en una opción rápida y confiable.

### 2.2.1.2. Serverless Computing en GCP

En la figura 2.2 se puede visualizar la estructura de la aplicación web en GCP y en la tabla 2.3 se detalla el servicio al que pertenece cada componente que forma parte de la aplicación.



**Figura 2. 2.** Diagrama de aplicación web en GCP

**Tabla 2. 3.** Servicios de GCP

Servicio	Google Cloud Platform
Content Delivery Network	Cloud CDN
Cloud Storage	Cloud Storage
Authenticate	Identity Platform
API Gateway	API Gateway
Functions (FaaS)	Cloud Functions
Database	Datastore

## **Cloud CDN**

La red de entrega de contenido en la nube (CDN) almacena contenido localmente en ciertos bordes y funciona con el servicio de equilibrio de carga HTTP(s) para entregar este contenido a los usuarios [23]. Google hace que esto suceda con sus puntos de presencia globales que están ubicados geográficamente, lo que permite que el contenido se venda más cerca de los usuarios.

Cuando un usuario solicita contenido de un sitio alojado en GCP, la solicitud llega a una ubicación de borde de la red de Google que, por lo general, está más cerca del usuario. Cloud CDN se almacena en caché en estas ubicaciones, lo que mejora los tiempos de respuesta para el usuario en caso de que solicite el mismo contenido nuevamente.

## **Cloud Storage**

Cloud Functions utiliza funciones basadas en eventos para manejar eventos de su infraestructura en la nube. Por ejemplo, Cloud Functions puede responder a las notificaciones de cambios que surgen de Google Cloud Storage [24]. Estas notificaciones se pueden configurar para que se activen en respuesta a varios eventos dentro de un depósito: creación, eliminación, archivo y actualizaciones de metadatos de objetos.

Los eventos de Cloud Storage que usa Cloud Functions se basan en las notificaciones de Cloud Pub/Sub para Google Cloud Storage y se proporcionan en el formato de la API de Cloud Storage.

## **Identity Platform**

Identity Platform proporciona un servicio de autenticación personalizado y directo para el registro y el inicio de sesión de los usuarios, generando una plataforma de administración de acceso e identidad ayudando a agregar funciones de administración de acceso e identidad de aplicaciones web, y así proteger las cuentas de los usuarios y escalar con confianza en Google Cloud Platform [25].

Con Identity Platform, resulta sencillo agregar un servicio de autenticación personalizado adoptado a aplicaciones web y así concentrarse en crear la aplicación e implementar los servicios adicionales.

Identity Platform protege a los usuarios evitando la apropiación de cuentas ofreciendo autenticación con la inteligencia de Google para proteger las cuentas.

## **API Gateway**

API Gateway facilita el trabajo de administrar todas las llamadas API al backend sin servidor. El escalado y la consolidación de diferentes tipos de llamadas API antes de implementar API Gateway; no solo han resuelto problemas, sino que se ha experimentado una mejora de 10 veces la administración de API mediante el uso de una plataforma única a la que se puede acceder fácilmente a través de la consola en la nube y que Google Cloud administra por completo [26].

## **Cloud Functions**

No existen servidores para aprovisionar, administrar o actualizar, a través de funciones escalables de pago por uso como servicio (FaaS) es factible la ejecución del código sin administración de servidor [27].

A continuación, se menciona características relevantes:

- Escalado automático en función de la carga
- Capacidad integrada de supervisión, registro y depuración
- Seguridad incorporada a nivel de rol y por función basada en el principio de privilegio mínimo
- Capacidades de red clave para escenarios híbridos y de múltiples nubes

Cloud Functions admite varios lenguajes de programación como:

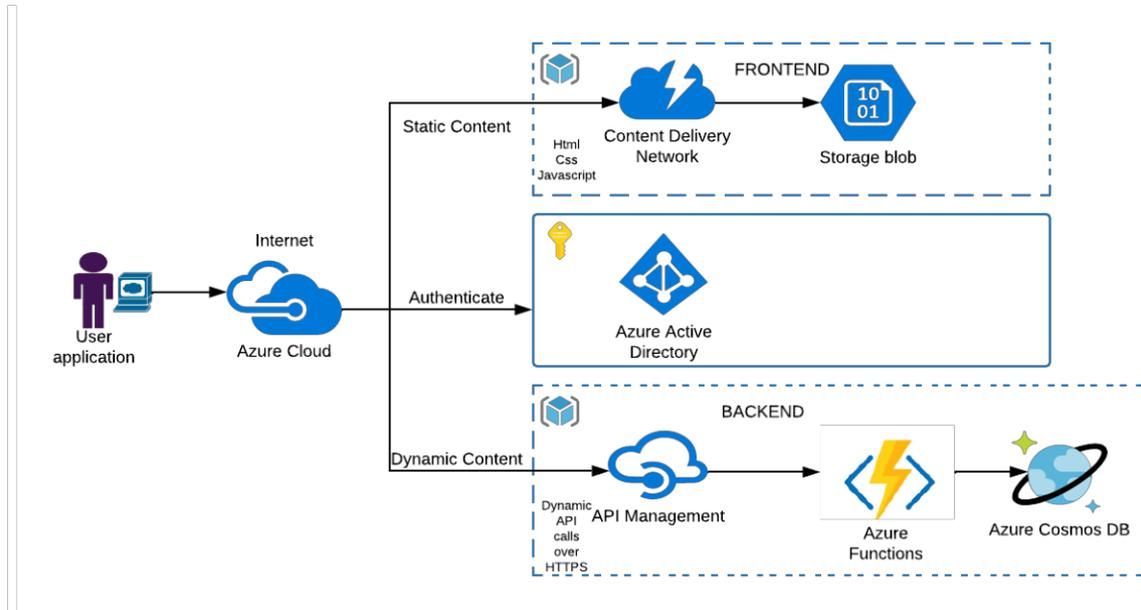
- Node.js
- Python

## **Datastore**

Datastore es una base de datos NoSQL altamente escalable para sus aplicaciones. Datastore maneja automáticamente la fragmentación y la replicación, lo que le brinda una base de datos duradera y de alta disponibilidad que se escala automáticamente para manejar la carga de sus aplicaciones. Datastore proporciona una gran variedad de capacidades, como transacciones ACID, consultas similares a SQL, índices y mucho más.

### 2.2.1.3. Serverless Computing en Azure Cloud Platform

En la figura 2.3 se puede visualizar la estructura de la aplicación web en GCP y en la tabla 2.4 se detalla el servicio al que pertenece cada componente que forma parte de la aplicación.



**Figura 2. 3.** Diagrama de aplicación web en Azure

**Tabla 2. 4.** Servicios de Azure Cloud Platform

Servicio	Azure Cloud Platform
Content Delivery Network	Azure Content Delivery Network
Cloud Storage	Azure Blob Storage
Authenticate	Azure Active Directory B2C
API Gateway	Azure API Management
Functions (FaaS)	Azure Functions Serverless Compute
Database	Azure Cosmos DB

### Azure Content Delivery Network

Ofrece una solución global para entregar contenido rápidamente. Ahorra ancho de banda y mejore la capacidad de respuesta al codificar o distribuir software, actualizaciones de firmware y puntos finales. Reduce los tiempos de carga de los sitios web, las aplicaciones y los medios de transmisión para aumentar la satisfacción del usuario a nivel mundial.

## **Azure Blob Storage**

Facilita la creación de lagos de datos para satisfacer sus necesidades de análisis y proporciona almacenamiento para crear aplicaciones móviles y nativas de nube con un gran potencial [29]. Optimice los costos con almacenamiento en capas para los datos a largo plazo y escale los recursos verticalmente con flexibilidad para las cargas de trabajo de informática de alto rendimiento y de aprendizaje automático.

## **Azure Active Directory B2C**

El servicio de identidad empresarial de Azure Active Directory (Azure AD) proporciona inicio de sesión único, autenticación multifactor y acceso condicional para protegerse contra el 99,9 % de los ataques de ciberseguridad.

Proporciona una experiencia de inicio de sesión fácil y rápida para mantener la productividad de los usuarios, reduce el tiempo de administración de contraseñas y minimizar la fricción.

## **Azure API Management**

Al importar ciertas API, API Management configura el backend de la API automáticamente. Por ejemplo, API Management configura el backend al importar recursos de Azure, como una aplicación de funciones (Serverless) o una aplicación lógica de Azure activada por HTTP.

## **Azure Functions**

Azure Functions proporciona la capacidad de escalar según sea necesario; por lo tanto, si la carga es alta se puede esperar que Azure Functions escale y se adapte a la carga alta.

Algunas características de Azure Functions son:

- Interfaz basada en el navegador, se puede escribir y probar código directamente en la interfaz sin usar ningún entorno de desarrollo integrado (IDE).
- Con Azure Functions solo se paga por el tiempo que se ejecuta el código por lo que, si la carga en la aplicación es baja, paga menos.
- Azure Functions se integra perfectamente con aplicaciones de terceros.

Los lenguajes de programación que admite Azure Functions son:

- C#
- JavaScript

- F#
- Java
- Python
- TypeScript
- PHP
- Batch
- Bash
- PowerShell

## **Azure Cosmos DB**

Azure Cosmos DB es una base de datos NoSQL completamente administrada para el desarrollo de aplicaciones modernas. Los tiempos de respuesta de milisegundos de un solo dígito y la escalabilidad automática e instantánea garantizan la velocidad a cualquier escala [30]. La continuidad del negocio está asegurada con disponibilidad respaldada por SLA y seguridad de nivel empresarial. El desarrollo de aplicaciones es más rápido y productivo gracias a la distribución de datos llave en mano en múltiples regiones en cualquier parte del mundo, API de código abierto y SDK para idiomas populares. Como un servicio completamente administrado, Azure Cosmos DB le quita la administración de la base de datos con la administración automática, las actualizaciones y la aplicación de parches. También gestiona la gestión de la capacidad con opciones rentables de escalado automático y sin servidor que responden a las necesidades de las aplicaciones para hacer coincidir la capacidad con la demanda.

### **2.3. Costo de la Aplicación Web**

#### **2.3.1. Modelos comerciales de proveedores de nube**

Como modelo comercial, la muestra gratuita funciona bien para los productos de software. Con el tiempo, los estándares y las reglas se han ajustado a los mejores intereses de ambas partes: consumidores y proveedores de productos de software. Hay muchas cosas a considerar cuando se trata de marketing de software: acuerdos de confidencialidad; términos y condiciones de uso; tipos de licencias; tipos de cuenta; y distribuciones con funcionalidad y soporte.

La nube ha permitido utilizar aplicaciones de Internet para realizar tareas, en lugar de ejecutarlas con el hardware existente. Estas aplicaciones van desde el almacenamiento

hasta la infraestructura completa. Esta tecnología permite consumir sus servicios con características más convenientes. En este modelo el pago es por consumo y además introduce nuevos conceptos para definir requerimientos: escalabilidad, elasticidad, flexibilidad y disponibilidad. Todo esto está disponible desde cualquier lugar con acceso a internet, a través de un portal autoalimentado.

Al evaluar los beneficios y las opciones que ofrecen los diferentes proveedores de servicios en la nube, vale la pena señalar que las opciones son prácticamente ilimitadas cuando se trata de lo que una empresa podría necesitar de un proveedor. Las opciones de nivel gratuito del modelo de negocio en la nube de soluciones serverless son infinitas.

Las páginas oficiales de AWS, GCP y Azure Cloud Platform permiten acceder a una calculadora de precios para estimar los costos de los servicios y configurarlos de acuerdo a los requerimientos, con esta herramienta es posible optimizar gastos que permita acogerse a un presupuesto y obtener un precio final detallado.

Una consideración importante al momento de utilizar las calculadoras es tener conocimiento de los recursos que cada servicio brinda en pruebas o periodos gratuitos para no realizar un cálculo incorrecto, es decir es fundamental indagar cada uno de los servicios de cada una de las plataformas a profundidad antes de agregarlos a la aplicación.

La tabla 2.5 contiene los enlaces a las calculadoras de cada una de las plataformas.

**Tabla 2. 5.** Calculadoras de Servicios de AWS, GCP y Azure Cloud Platform

<b>Plataforma</b>	<b>Calculadora</b>
Amazon Web Service	<a href="https://calculator.aws/#/">https://calculator.aws/#/</a>
Google Cloud Platform	<a href="https://cloud.google.com/products/calculator/">https://cloud.google.com/products/calculator/</a>
Azure Cloud Platform	<a href="https://azure.microsoft.com/es-mx/pricing/calculator/">https://azure.microsoft.com/es-mx/pricing/calculator/</a>

Existen variaciones en los precios de acuerdo a la ubicación física en todo el mundo donde los proveedores de nube agrupan los centros de datos. En esta investigación los cálculos estimados se los obtendrá considerando la localidad Estados Unidos.

### **2.3.2. AMAZON WEB SERVICES**

AWS permite obtener experiencia práctica y gratuita con la plataforma, los productos y los servicios de capa gratuita.

#### **Capa gratuita**

Existen tres tipos de ofertas de capa gratuita:

- **Prueba Gratis:** Inicia a partir de la fecha en que activa un servicio en particular
- **12 meses Gratis:** Inicia a partir de la fecha de registro inicial en AWS
- **Gratis:** Esta oferta no vence y está disponible para todos los clientes de AWS [28].

## Amazon Cloudfront

- Oferta: 12 meses gratis
- Incluye: 1 TB de transferencia de datos  
10 000 000 de solicitudes HTTP o HTTPS  
2 000 000 invocaciones de funciones de CloudFront [29]

Suponiendo que la aplicación tiene menos de 1 TB de transferencia de datos a Internet y menos de 20 000 000 de solicitudes HTTPS en total, la capa gratuita de AWS cubrirá las solicitudes DTO y HTTPS, sin costo alguno.

El siguiente análisis se realizará considerando el servicio se utilice en una pequeña carga de trabajo de producción.

Si tiene 100 GB de datos que salen a Internet desde un caché de CloudFront por mes y realiza 1 000 000 de solicitudes HTTPS al obtener contenido de CloudFront y entregarlo a su espectador.

También se utiliza CloudFront Functions para el procesamiento ligero de solicitudes web, como la manipulación de claves de caché o la reescritura de URL.

La transferencia de datos tiene un costo de acuerdo a la localidad, una vez se consuma 1 TB que ofrece la prueba gratuita los valores deben ser multiplicados por los que se observan en la figura 2.4.

Por mes	Estados Unidos, México y Canadá	Europa e Israel	Sudáfrica, Kenia y Medio Oriente	Sudamerica	Japón	Australia y Nueva Zelanda	Hong Kong, Indonesia, Filipinas, Singapur, Corea del Sur, Taiwán y Tailandia	India
Primeros 10 TB	\$0.085	\$0.085	\$0.110	\$0.110	\$0.114	\$0.114	\$0.120	\$0.109
Siguientes 40 TB	\$0.080	\$0.080	\$0.105	\$0.105	\$0.089	\$0.098	\$0.100	\$0.085
Siguientes 100 TB	\$0.060	\$0.060	\$0.090	\$0.090	\$0.086	\$0.094	\$0.095	\$0.082
Siguientes 350 TB	\$0.040	\$0.040	\$0.080	\$0.080	\$0.084	\$0.092	\$0.090	\$0.080
Siguiente 524 TB	\$0.030	\$0.030	\$0.060	\$0.060	\$0.080	\$0.090	\$0.080	\$0.078
Siguiente 4PB	\$0.025	\$0.025	\$0.050	\$0.050	\$0.070	\$0.085	\$0.070	\$0.075
Más de 5 PB	\$0.020	\$0.020	\$0.040	\$0.040	\$0.060	\$0.080	\$0.060	\$0.072

**Figura 2. 4.** Tabla de Precios Amazon Cloudfront [32]

CloudFront utiliza una solicitud de espectador y una función de respuesta de espectador en cada solicitud. Esto invocaría dos funciones por solicitud, las primeras 2 000 000 de solicitudes no generarán ningún cargo, luego se le cobrará \$0,1 por cada millón de solicitudes. En la tabla 2.6 se observa a detalle los cálculos realizados.

**Tabla 2. 6.** Costo mensual de Amazon CloudFront

	<b>Cálculo de costo</b>	<b>Coste total</b>
Transferencia de datos de 100 GB	100	0 USD
1000000 de solicitudes HTTPS	1 000 000 x \$0,00 por los primeros 10,000,000	0 USD
Funciones de visor de 12000000	(12 000 000 – 2 000 000) x 0,01 USD por 1 000 000 solicitudes	0,10 USD
Transferencia de datos a Internet	0,22 USD	0,22 USD
Transferencia de datos a Origen	0,25 USD	0,25 USD
	<b>Costo mensual</b>	<b>0,57 USD</b>

### 2.3.2.1. Amazon S3

- Oferta: 12 meses gratis
- Incluye: 5 GB de almacenamiento estándar  
20 000 solicitudes Get  
2000 solicitudes Put  
100 GB de transferencia de datos cada mes.

El siguiente análisis se realizará considerando el servicio se utilice en una pequeña carga de trabajo de producción y una vez finalice los 12 meses gratis.

Precio para escalar a: 10 GB

$$10 \text{ GB} \times 0,0230000000 \text{ USD} = 0,23 \text{ USD}$$

Costo total del nivel = 0,23 USD (costo de almacenamiento estándar de S3)

100 000 solicitudes PUT para S3 Storage x 0,000005 USD por solicitud = 0,50 USD (costo de solicitudes PUT estándar de S3)

100 000 solicitudes GET en un mes x 0,0000004 USD por solicitud = 0,004 USD (costo de solicitudes GET estándar de S3)

0,23 USD + 0,004 USD + 0,50 USD = 0,73 USD (Almacenamiento estándar total de S3, solicitudes de datos), en la tabla 2.7 se observa a detalle los cálculos realizados.

**Tabla 2. 7.** Costo mensual de Amazon S3

	<b>Costo</b>	<b>Coste total</b>
10 GB de almacenamiento	0,023 USD por GB	0,23 USD
100 000 solicitudes PUT	0,000005 USD por solicitud	0,50 USD
100 000 solicitudes GET	0,0000004 USD por solicitud	0,04 USD
	<b>Costo mensual</b>	<b>0,77 USD</b>

### 2.3.2.2. Amazon Cognito

- Oferta: Gratuito
- Incluye: Nivel gratuito de 50 000 MAU (Usuarios activos mensuales).  
10 GB de almacenamiento de sincronización en la nube  
1 000 000 de operaciones de sincronización por mes

La capa gratuita no vence automáticamente al final de su plazo de 12 meses de la capa gratuita de AWS y está disponible para los clientes nuevos y existentes de AWS de forma indefinida [30].

Nivel de precios (MAU)      50 001-100 000 = \$0,0055 x MAU

Próximos 9 000000 = \$0,0046 x MAU

Próximos 9 000000 = \$0,00325 x MAU

Superior a 10 000 000 = \$0,0025 x MAU

El siguiente análisis se ha realizado considerando el servicio se utilice en una pequeña carga de trabajo de producción siendo necesario escalar a 51000 usuarios. En la tabla 2.8 se observa a detalle del costo mensual.

**Tabla 2. 8.** Costo mensual de Cognito

	<b>Costo</b>	<b>Coste total</b>
51000 MAU	\$0,0055 x MAU	5,50 USD
	<b>Costo mensual</b>	5,50 USD

### 2.3.2.3. Amazon API Gateway

- Oferta: 12 meses gratis
- Incluye: 1 millón de llamadas de API recibidas por mes  
1M mensajes y 750.000 minutos de conexión

Las solicitudes tienen un tamaño de 512 KB de datos, si supera este valor se realiza el cálculo respectivo para determinar al número de solicitudes que se obtendrán [31].

El siguiente análisis se ha realizado considerando el servicio se utilice en una pequeña carga de trabajo de producción y se estima una solicitud de tamaño 1000 KB, por lo tanto, la solicitud será identificada como 2 solicitudes, el costo mensual se detalla en la tabla 2.9.

2 solicitud por mes x 1 000 000 multiplicador de unidades x 2 solicitudes facturables = 4 000 000 solicitudes facturables totales

Precio escalonado para: 2 000 000 solicitudes

**Tabla 2. 9.** Costo mensual de API Gateway

	<b>Costo</b>	<b>Coste total</b>
2 000 000 solicitudes	0,000001 USD/solicitud	4,00 USD
	<b>Costo mensual</b>	4,00 USD

### 2.3.2.4. AWS Lambda

- Oferta: Gratuito
- Incluye: 1 000 000 solicitudes gratuitas por mes  
3,2 millones de segundos de tiempo de informática por mes [32].

Si el número de solicitudes supera el valor mencionado en el apartado anterior el costo varía de acuerdo a:

- La cantidad de memoria asignada
- La duración de cada solicitud, la que se calcula desde el momento que empieza a ejecutarse hasta que finaliza
- La cantidad de memoria asignada

Después del 1 millón de solicitudes gratis se realizará un cobro de \$0.20/1 millón de solicitudes, más \$0.00001667/GB-seg.

El siguiente cálculo se realizará considerando se tenga 10000000 de solicitudes al mes, la duración de solicitud de 20 ms y la cantidad de memoria 5 GB.

10000000 solicitudes x 20 ms x 0,001 ms a factor de conversión de segundos = 200000 (segundos)

5 GB x 200 000 segundos = 1 000 000 de cómputo total (GB-s)

1 000 000,00 GB-s - 400 000 GB-s de nivel gratuito = 600 000,00 GB-s

600.000,00 GB-s x 0,0000166667 USD = 10,00 USD (cargos de cómputo mensuales)

10000000 solicitudes - 1000000 solicitudes de nivel gratuito = 9 000 000 solicitudes facturables mensuales

900000000 solicitudes facturables mensuales totales x 0,0000002 USD = 1,80 USD.

El costo mensual se detalla en la tabla 2.10.

**Tabla 2. 10. Costo mensual de AWS Lambda**

	<b>Costo</b>	<b>Coste total</b>
600 00000 GB-s	\$0.00001667/GB-seg	10,00 USD
9 000 000 solicitudes	0,0000002 USD	1,80 USD
	<b>Costo mensual con nivel gratuito</b>	11,80 USD

### 2.3.2.5. Amazon DynamoDB

- Oferta: Gratuito
- Incluye: 25 GB de almacenamiento

25 unidades de capacidad de escritura (WCU) aprovisionada

25 unidades de capacidad de lectura (RCU) aprovisionada

Una vez termine el espacio de almacenamiento gratuito, el costo será de \$0.25 por GB-mes a partir de entonces [33].

El siguiente cálculo se realizará considerando se requiere 25 GB adicionales, el detalle del costo mensual se observa en la tabla 2.11.

**Tabla 2. 11.** Costo mensual de DynamoDB

	<b>Costo</b>	<b>Coste total</b>
25 GB de almacenamiento	\$0.25 por GB	6,25 USD
	<b>Costo mensual</b>	6,25 USD

### **2.3.3. GOOGLE CLOUD PLATFORM**

GCP permite acceder a todos los productos y servicios una vez se haya creado una cuenta, proporciona un crédito de \$300 con la limitación de utilizarlo en 90 días, periodo en el cual es posible hacer uso de más de 20 servicios gratuitos. No existen cargos automáticos cuando finaliza la prueba gratuita. Una vez termine el plazo de prueba los precios varían según el producto y el uso [34].

De acuerdo al esquema de aplicación que se ha diseñado, en la siguiente sección se mencionan los costos de cada una de los servicios a utilizar, las consideraciones se las realiza en función.

#### **2.3.3.2. Cloud CDN**

Los cargos por salida de caché representan respuestas almacenadas en caché servidas desde los cachés de Cloud CDN y varían según el destino y su uso mensual como se observa en la figura 2.5.

	Primeros 10 TB	Siguientes 140 TB	Siguientes 850 TB	> 1000TB
Asia Pacífico (incluido Hong Kong)	\$0.09	\$0.06	\$0.05	\$0.04
chino <sup>1</sup>	\$0.20	\$0.17	\$0.16	\$0.145
Europa	\$0.08	\$0.055	\$0.03	\$0.02
Norteamérica (incluido Hawái)	\$0.08	\$0.055	\$0.03	\$0.02
Oceanía <sup>2</sup>	\$0.11	\$0.09	\$0.08	\$0.065
Sudamerica	\$0.09	\$0.06	\$0.05	\$0.04
Todos los demás destinos	\$0.09	\$0.06	\$0.05	\$0.04

**Figura 2. 5.** Tabla de Precios Cloud CDN [24]

El siguiente análisis se ha realizado considerando se tiene 100 GB de datos que salen a Internet desde un caché de CDN por mes y realiza 1 000 000 de solicitudes HTTPS al obtener contenido de la CDN y entregarlo al usuario, los detalles de costo mensual se observan en la tabla 2.12.

**Tabla 2. 12.** Costo mensual de Cloud CDN

	Costo	Coste total
100 GB	Considerando una distribución a cada una de las zonas	11,85 USD
1000000 de solicitudes de búsqueda de caché HTTPS	0,75 USD por millón	0,75
	<b>Costo mensual</b>	<b>9,75 USD</b>

### 2.3.3.3. Cloud Storage

El siguiente análisis se ha realizado considerando se utilice en una pequeña carga de trabajo y una vez finalice los 12 meses gratis, si se requiere escalar a 10 GB, el costo mensual se observa en la tabla 2.13.

**Tabla 2. 13.** Costo mensual de Cloud Storage

	Costo	Coste total
10 GB	\$0.026 por GB	0,26 USD

100000 peticiones PUT	\$0.05 por 10000	0,50 USD
100000 peticiones GET	\$0.004 por 10000	0,04 USD
	<b>Costo mensual</b>	0,80 USD

#### 2.3.3.4. Identity Platform

Los precios de Identity Platform se dividen en diferentes niveles según el método de autenticación utilizado, en usuarios activos mensuales (MAU) de proveedores sociales, anónimos, email más contraseña y autenticación telefónica. Sin cargo adicional por múltiples autenticaciones por mes para cada MAU, el nivel de precios es:

50,001-100,000 = \$0.0055 x MAU

Próximos 9, 000,000 = \$0.0046 x MAU

Próximos 9, 000,000 = \$0.0032 x MAU

Superior a 10, 000,000 = \$0.0025 x MAU

El siguiente análisis se ha realizado considerando el servicio se utilice en una pequeña carga de trabajo de producción siendo necesario escalar a 51000 usuarios. En la tabla 2.14 se observa a detalle del costo mensual.

**Tabla 2. 14.** Costo mensual de Identity Platform

	<b>Costo</b>	<b>Coste total</b>
51000 usuarios	\$0.0055 x MAU	5,50 USD
	<b>Costo mensual</b>	5,50 USD

#### 2.3.3.5. API Gateway

API Gateway cobra por sus llamadas a Service Control. Cada llamada de API procesada por API Gateway se informa como una operación rastreada por la API de Control de servicios y se incluye como un elemento de línea para Control de servicios en su factura.

El precio de API Gateway depende de la cantidad de llamadas a su API, como se describe a continuación [36]:

Llamadas a la API por mes por cuenta de facturación y coste por millón

0-2M= \$0.00

2M-1B= \$3.00

1B+= \$1.50

El siguiente análisis se ha realizado en el caso de tener un número de solicitudes de 2000000, el costo mensual se detalla en la tabla 2.15.

**Tabla 2. 15.** Costo mensual de API Gateway

	<b>Costo</b>	<b>Coste total</b>
2000000 solicitudes	2M-1B= \$3.00	3,00 USD
	<b>Costo mensual</b>	3,00 USD

### 2.3.3.6. Cloud Functions

Las invocaciones se cobran a una tarifa por unidad de \$0,0000004 por invocación (o \$0,40 por millón), sin incluir las primeras 2 millones de invocaciones gratuitas por mes. Existen cargos independientemente del resultado de la función o su duración; al igual que cargos por implementar sus funciones, incluso si la cantidad de invocaciones se encuentra dentro del nivel gratuito.

El tiempo de cómputo se mide desde el momento en que su función recibe una solicitud hasta el momento en que se completa, ya sea a través de su señalización de finalización, o mediante un tiempo de espera, otra falla o cualquier otra terminación. El tiempo de cálculo se mide en incrementos de 100 ms, redondeado al incremento más cercano. Por ejemplo, una función que se ejecuta durante 260 ms se facturaría como 300 ms [37].

Después del 1 millón de solicitudes gratis se realizará un cobro de \$0.40/1 millón de invocaciones, más \$0.0000025/GB-seg, es posible visualizar esta información de forma detallada en la tabla 2.16.

El siguiente análisis en el caso de existir 10000000 solicitudes, basada en eventos con 128 MB de memoria y una CPU de 200 MHz ejecutándose durante 300 ms, en la tabla 2.14 se observa el resumen de los cálculos realizados y el costo mensual total.

Tiempo de cómputo:

$$(128 \text{ MB}/1024 \text{ MB}/\text{GB}) \times 0,3 \text{ s} = 0,0375 \text{ GB-segundos por invocación}$$

$$(200 \text{ MHz} / 1000 \text{ MHz}/\text{GHz}) \times 0,3 \text{ s} = 0,0600 \text{ GHz-segundos por invocación}$$

$$10\ 000\ 000 \text{ invocaciones} \times 0,0375 \text{ GB-segundos} = 375\ 000 \text{ GB-segundos al mes}$$

$$10\ 000\ 000 \text{ invocaciones} \times 0,0600 \text{ GHz-segundos} = 600\ 000 \text{ GHz-segundos al mes}$$

**Tabla 2. 16.** Costo mensual de Google Cloud Functions

Métrico	Valor	Nivel gratuito	Valor neto	Precio unitario	Precio total
invocaciones	10,000,000	2,000,000	8,000,000	\$0.0000004	\$3.20
GB-segundos	375.000	400.000	< 0	\$0.0000025	\$0.00
GHz-segundos	600.000	200,000	400.000	\$0.0000100	\$4.00
Redes	0	5	0	\$0.12	\$0.00
				<b>Costo mensual</b>	<b>\$7.20</b>

### 2.3.3.7. Datastore

Datastore es una base de datos NoSQL altamente escalable en la creación de aplicaciones web [38].

Existe un servicio gratuito limitado por día en la que se permite tamaño de datos almacenados de 1GB, lecturas 50000, escrituras 20000, eliminaciones 20000, operaciones pequeñas 50000.

En caso de superar el límite el costo de almacenamiento será de \$0,18 GB/mes, \$0,06 por 100000 lecturas, \$0,18 por 100000 escrituras, \$0.02 por 100000 eliminaciones, y operaciones pequeñas gratis.

El siguiente cálculo se realizará considerando se requiere 25 GB adicionales, el detalle del costo mensual se observa en la tabla 2.17.

**Tabla 2. 17.** Costo mensual de DataStore

	Costo	Coste total
25 GB de almacenamiento	\$0,18 por GB	4,32 USD
	<b>Costo mensual</b>	<b>4,32 USD</b>

### 2.3.4. MICROSOFT AZURE

Azure Cloud Platform permite acceder a todos los productos y servicios una vez se haya creado una cuenta y proporciona un crédito de \$200 con la limitación de utilizarlo en 30 días, periodo en el cual es posible hacer uso de más de 40 servicios gratuitos, no existen cargos automáticos cuando finaliza la prueba gratuita. Después del crédito para seguir

recibiendo servicios populares y más de 25 servicios más, solo paga si usa más de las cantidades mensuales gratuitas. [39].

Existen dos tipos de ofertas de capa gratuita:

- 12 meses gratis
- Siempre gratis

### 2.3.4.2. Azure Content Delivery Network

La red de entrega de contenido estático permite optimizar el acceso a los recursos del frontend de la aplicación y mejora la experiencia del usuario.

Para la transferencia de datos los costos varían de acuerdo a la zona como se observa en la figura 2.6.

Transferencias de datos de salida <sup>1</sup>	Zona 1 <sup>2</sup>	Zona 2 <sup>2</sup>	Zona 3 <sup>2</sup>	Zona 4 <sup>2</sup>	Zona 5 <sup>2</sup>
Primeros 10 TB / mes	\$0,081 por GB	\$0,129 por GB	\$0,233 por GB	\$0,13 por GB	\$0,158 por GB
Siguientes 40 TB (10-50 TB)/mes	\$0,075 por GB	\$0,121 por GB	\$0,186 por GB	\$0,126 por GB	\$0,121 por GB
Siguientes 100 TB (50-150 TB)/mes	\$0,056 por GB	\$0,112 por GB	\$0,168 por GB	\$0,112 por GB	\$0,102 por GB
Siguientes 350 TB (150-500 TB)/mes	\$0,037 por GB	\$0,093 por GB	\$0,149 por GB	\$0,093 por GB	\$0,093 por GB

**Figura 2. 6.** Tabla de Precios Azure CDN [41].

Existen 5 zonas: Zona 1: América del Norte – Europa, Zona 2: Asia Pacífico, Japón, Zona 3: Brasil, Zona 4: Australia, Zona 5: India

Las solicitudes procesadas por millón tienen un costo de \$0,60

El siguiente análisis se realizará considerando el servicio se utilice en una pequeña carga de trabajo de producción.

Si requiere 100 GB de datos y realiza 1 000 000 de solicitudes HTTPS al obtener contenido de CloudFront y entregarlo a su espectador se obtendrán los costos señalados en la tabla 2.18.

**Tabla 2. 18.** Costo mensual de DataStore

	Cálculo de costo	Coste total
Transferencia de datos de 100 GB	Considerando una distribución a cada una de las zonas	14,36 USD

1 000 000 de solicitudes	\$0,60 por millón	0,60 USD
	<b>Costo mensual</b>	14,96 USD

### 2.3.4.3. Azure Blob Storage

El siguiente análisis se ha realizado considerando se utilice el servicio en una pequeña carga de trabajo, si se requiere escalar a 10 GB, el costo mensual se observa en la tabla 2.19.

**Tabla 2. 19.** Costo mensual de Cloud Storage

	<b>Costo</b>	<b>Coste total</b>
10 GB	\$0.026 por GB	0,26 USD
100000 peticiones GET	\$0.065 por 10000	0,65 USD
100000 peticiones PUT	\$0.005 por 10000	0,05 USD
	<b>Costo mensual</b>	0,96 USD

### 2.3.4.4. Azure Active Directory Business-to-Customer (B2C)

El precio de Azure AD se basa en los usuarios activos mensuales (MAU), lo que le ayuda a reducir costos y realizar previsiones con confianza. Los primeros 50.000 MAU por mes son gratuitos, más de 50.000 MAU= \$0.00325 x MAU. Se factura una tarifa fija de \$0.03 por cada intento de autenticación

El siguiente análisis se ha realizado considerando el servicio se utilice en una pequeña carga de trabajo de producción siendo necesario escalar a 51000 usuarios. En la tabla 2.20 se observa a detalle del costo mensual.

**Tabla 2. 20.** Costo mensual de Identity Platform

	<b>Costo</b>	<b>Coste total</b>
51000 usuarios	\$0.00325 x MAU	3,25 USD
	<b>Costo mensual</b>	3,25 USD

### 2.3.4.5. Azure API Management

Azure Api Managment sin servidor del servicio API Management, facturado por ejecución permite 0-1 millón de llamadas por suscripción. Incluido más de un millón de llamadas con un costo de \$0.042 por 10,000 llamadas [42].

El siguiente análisis se ha realizado en el caso de tener un número de solicitudes de 2000000 solicitudes, el costo mensual se detalla en la tabla 2.21

**Tabla 2. 21.** Costo mensual de API Gateway

	<b>Costo</b>	<b>Coste total</b>
2000000 solicitudes	0.042 por 10,000 solicitudes	4,20 USD
	<b>Costo mensual</b>	4,20 USD

#### 2.3.4.6. Azure Functions

La Solución Serverless de Azure facilita escalar automáticamente y pagar solo por los recursos informáticos cuando las funciones se estén ejecutando. En el plan de consumo, las instancias del host de funciones se agregan y eliminan dinámicamente según la cantidad de eventos entrantes [40].

El plan de consumo de Azure Functions se factura en función del consumo de recursos por segundo y las ejecuciones.

El Plan gratuito mensual incluye:

- 1 millón de solicitudes
- Tiempo de ejecución: 400000 GB-s de consumo de recursos por mes en suscripción en precios de pago por uso en todas las aplicaciones de función en esa suscripción, si supera este valor el costo es \$0.000016 /GB-s permitiendo 1 millón de ejecuciones y si sobrepasa genera un valor de \$0.20 por millón de ejecuciones.

El siguiente cálculo se realizará considerando se tenga 10000000 de solicitudes al mes, la duración de solicitud de 20 ms y la cantidad de memoria 5 GB.

$5 \text{ GB} \times 200 \text{ 000 segundos} = 1 \text{ 000 000 de cómputo total (GB-s)}$

$1 \text{ 000 000,00 GB-s} - 400 \text{ 000 GB-s de nivel gratuito} = 600 \text{ 000,00 GB-s}$

$600.000,00 \text{ GB-s} \times 0,0000166667 \text{ USD} = 10,00 \text{ USD (cargos de cómputo mensuales)}$

$10000000 \text{ solicitudes} - 1000000 \text{ solicitudes de nivel gratuito} = 9 \text{ 000 000 solicitudes facturables mensuales}$

$900000000 \text{ solicitudes facturables mensuales totales} \times 0,0000002 \text{ USD} = 1,80 \text{ USD.}$

El costo mensual se detalla en la tabla 2.22.

**Tabla 2. 22.** Costo mensual de AWS Lambda

	<b>Costo</b>	<b>Coste total</b>
600 00000 GB-s	\$0.00001667/GB-seg	10,00 USD
9 000 000 solicitudes	0,0000002 USD	1,80 USD
	<b>Costo mensual con nivel gratuito</b>	11,80 USD

#### **2.3.4.7. Azure Cosmos DB**

Cosmos DB ofrece 1000 unidades de solicitud por segundo de procesamiento provisionado con 25 GB de almacenamiento mensualmente siempre gratis, el pago es de acuerdo al consumo y se lo realiza una vez supere las cantidades gratuitas establecidas.

El siguiente cálculo se realizará considerando se requiere 25 GB adicionales, el detalle del costo mensual se observa en la tabla 2.23.

**Tabla 2. 23.** Costo mensual de DataStore

	<b>Costo</b>	<b>Coste total</b>
25 GB de almacenamiento	\$0,15 por GB	3,75 USD
	<b>Costo mensual</b>	3,75 USD

### **3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES**

En esta sección se muestran los resultados obtenidos de la investigación refiriéndose a los costos que genera la creación de una aplicación web, utilizando las plataformas Amazon Web Service (AWS), Google Cloud Platform (GCP) y Azure Cloud Platform de los proveedores de nube Google, Amazon y Microsoft respectivamente.

Además, se presentan las conclusiones y recomendaciones del estudio realizado, fundamentados en las secciones anteriores.

El proyecto estuvo enfocado a la creación de una aplicación web bajo la arquitectura Serverless Computing, con la finalidad de identificar servicios que faciliten el modelo de trabajo de esta arquitectura.

La investigación ha permitido abordar algunos desafíos como lo es definir la arquitectura Serverless Computing, al comprender que Serverless Computing permite ejecutar durante un periodo de tiempo determinadas porciones de código denominadas "funciones", siendo las funciones el núcleo de la arquitectura, considerando aquello el estudio se focaliza en

AWS Lambda, Google Cloud Functions y Azure Functions. Las soluciones Serverless deben integrarse con otros servicios que invoquen a las llamadas “funciones” a través de eventos.

Es así que los tres proveedores permiten cubrir la necesidad de crear una aplicación web, los servicios que integran cada plataforma trabajan en conjunto con funciones, y de esta manera fue posible evidenciar la existencia de servicios con operatividad similar en cada uno de los proveedores de nube. En la tabla 3.1, se puede apreciar el tipo y nombre de los servicios utilizados.

**Tabla 3. 1.** Servicios de AWS, Google y Azure para la creación de una aplicación web

<b>Service</b>	<b>Amazon Web Service</b>	<b>Google Cloud Platform</b>	<b>Azure Cloud Platform</b>
<b>Content Delivery Network</b>	Amazon Cloudfront	Cloud CDN	Azure Content Delivery Network
<b>Cloud Storage</b>	Amazon S3	Cloud Storage	Azure Blob Storage
<b>Authenticate</b>	Amazon Cognito	Identity Platform	Azure Active Directory B2C
<b>API Gateway</b>	Amazon API Gateway	API Gateway	Azure API Management
<b>Functions (FaaS)</b>	AWS Lambda	Cloud Functions	Azure Functions
<b>Database</b>	Amazon DynamoDB	Datastore	Azure Cosmos DB

### **3.1. Costos finales del modelo de aplicación web creado bajo la arquitectura Serverless Computing**

Los costos presentados en la metodología se estimaron tomando en cuenta condiciones similares y periodos mensuales para las tres plataformas enfocadas a la creación de una aplicación web para quienes habitualmente manejan pequeña carga de trabajo o requieren empezar creando aplicaciones modernas sin tener que realizar gastos de infraestructura y servidores.

A continuación, se detallan los costos de cada uno de los servicios utilizados en las plataformas AWS, GCP y Azure Cloud Platform.

El costo mensual de Content Delivery Network o red de distribución de contenido para el frontend de la aplicación en cada una de las plataformas se mencionan en la tabla 3.2.

**Tabla 3. 2.** Costo mensual CDN

<b>Plataforma</b>	<b>Servicio</b>	<b>Costo mensual</b>
<b>Amazon Web Service</b>	Amazon Cloudfront	0,57 USD
<b>Google Cloud Platform</b>	Cloud CDN	9,75 USD
<b>Azure Cloud Platform</b>	Azure CDN	14,36 USD

El costo mensual de Cloud Storage para el almacenamiento de código de frontend en cada una de las plataformas se menciona en la tabla 3.3.

**Tabla 3. 3.** Costo mensual Cloud Storage

<b>Plataforma</b>	<b>Servicio</b>	<b>Costo mensual</b>
<b>Amazon Web Service</b>	Amazon S3	0,77 USD
<b>Google Cloud Platform</b>	Cloud Storage	0,80 USD
<b>Azure Cloud Platform</b>	Azure Blob Storage	0,96 USD

El costo mensual de Autenticación para el control de usuarios que opera en conjunto con el frontend y backend de la aplicación en cada una de las plataformas se menciona en la tabla 3.4.

**Tabla 3. 4.** Costo mensual Autenticación

<b>Plataforma</b>	<b>Servicio</b>	<b>Costo mensual</b>
<b>Amazon Web Service</b>	Amazon Cognito	5,50 USD
<b>Google Cloud Platform</b>	Identity Platform	5,50 USD
<b>Azure Cloud Platform</b>	Azure Active Directory B2C	3,25 USD

El costo mensual de API Gateway para el backend de la aplicación en cada una de las plataformas se menciona en la tabla 3.5.

**Tabla 3. 5.** Costo mensual API Gateway

<b>Plataforma</b>	<b>Servicio</b>	<b>Costo mensual</b>
<b>Amazon Web Service</b>	Amazon API Gateway	4,00 USD
<b>Google Cloud Platform</b>	API Gateway	3,00 USD

<b>Azure Cloud Platform</b>	Azure API Management	4,20 USD
-----------------------------	----------------------	----------

El costo mensual de Functions conocida también solución serverless para el backend de la aplicación en cada una de las plataformas se menciona en la tabla 3.6.

**Tabla 3. 6.** Costo mensual Functions

<b>Plataforma</b>	<b>Servicio</b>	<b>Costo mensual</b>
<b>Amazon Web Service</b>	AWS Lambda	11,80 USD
<b>Google Cloud Platform</b>	Cloud Functions	7.20 USD
<b>Azure Cloud Platform</b>	Azure Functions	11,80 USD

El costo mensual de Database o base de datos NoSQL para almacenamiento en cada una de las plataformas se menciona en la tabla 3.7.

**Tabla 3. 7.** Costo mensual Database

<b>Plataforma</b>	<b>Servicio</b>	<b>Costo mensual</b>
<b>Amazon Web Service</b>	Amazon DynamoDB	6,25 USD
<b>Google Cloud Platform</b>	Datastore	4,32 USD
<b>Azure Cloud Platform</b>	Azure CosmosDB	3,75 USD

### **3.2. Análisis comparativo arquitecturas Serverless Computing**

En función de la sección anterior es posible determinar los costos mensuales finales que genera la creación de un modelo de aplicación web bajo la Arquitectura Serverless Computing considerando se inicie el proyecto ya sea en Amazon Web Services, Google Cloud Platform o Azure Cloud Platform, tal cual se detalla en la tabla 3.8.

**Tabla 3. 8.** Costo Aplicación Web en AWS, GCP y Azure Cloud Functions

<b>Plataforma</b>	<b>Servicio</b>	<b>Costo mensual</b>
<b>Amazon Web Service</b>	Amazon DynamoDB	28,89 USD
<b>Google Cloud Platform</b>	Datastore	30,57 USD
<b>Azure Cloud Platform</b>	Azure CosmosDB	38,32 USD

En el proyecto se determinaron costos mensuales con la finalidad de realizar un análisis comparativo y en función de la información que proveen las páginas oficiales. Tal y como se evidencia en la tabla 3.8 los costos mensuales de la aplicación web tienen una diferencia mínima, pero es importante hacer hincapié que no existe la misma correspondencia con los servicios incluidos en la aplicación.

Aunque los costos mensuales que genera Serverless Computing son similares, la manera de cobro de cada plataforma es diferente de acuerdo a ello se puede apreciar lo siguiente:

- En Amazon Web Services existen tres tipos de ofertas prueba gratis, 12 meses gratis y gratuito siempre con más de 200 servicios disponibles.
- En Google Cloud Platform al suscribirse la plataforma ofrece un crédito inicial de \$300 por 90 días y más de 20 servicios gratuitos.
- En Azure Cloud Platform al suscribirse la plataforma ofrece \$200 por 30 días y más de 40 servicios gratuitos.

Basado en las premisas anteriores es posible concluir que las ofertas de cada plataforma permitirán realizar un análisis de acuerdo a la aplicación que se desee implementar, en este caso la mejor solución para la creación de una aplicación web bajo la arquitectura Serverless Computing es AWS puesto que el valor mensual a pagar es de 28,89 USD; siendo el costo más bajo de las tres plataformas.

El costo en GCP no varía en gran proporción con respecto a AWS, podría ser considerado tomando en cuenta que en los tres meses de crédito que ofrece la plataforma es posible realizar pruebas, modificar servicios y analizar el costo mensual que genera la aplicación.

Por otra parte, Azure Cloud Platform aunque genera un costo mayor emplea la tecnología de Microsoft, pues resulta ser la más conocida y entendida por la mayoría de usuarios por su uso cotidiano en la mayoría de sectores.

Los servicios utilizados en la elaboración del proyecto tienen gran importancia puesto que un modelo de aplicación web sin servidor necesita esos componentes, de tal manera se permitirá al usuario interactuar y realizar una o varias tareas.

Existen servicios de la aplicación web que generan un costo similar en cada una de las plataformas, los servicios se mencionan a continuación:

- El API Gateway que interviene al invocar funciones como respuesta al evento de una solicitud resulta factible utilizarlo en las tres plataformas.

- Así mismo Cloud Storage para el almacenamiento de archivos del frontend tiene un valor final que no difiere en gran proporción en las tres plataformas.

API Gateway y Cloud Storage son servicios importantes en una arquitectura Serverless Computing, tomando en cuenta esta consideración GCP, AWS y Azure Cloud Platform serán una excelente opción.

Por otra parte, a continuación se presenta un análisis comparativo de los servicios con mayor diferencia en cuánto a costos de las plataformas seleccionadas.

### **Amazon Cloudfront (AWS), Cloud CDN (GCP) y Azure Content Delivery Network**

Como se puede apreciar en la tabla 3.2 del servicio de Red de Distribución de contenido o CDN, existe una gran diferencia entre Amazon Cloudfront con Cloud CDN y Azure CDN en su costo mensual. Esto debido a que Amazon ofrece una prueba gratuita en la que incluye 1 TB de transferencia de datos, 10 000 000 de solicitudes HTTPS y 2 000 000 invocaciones de funciones de CloudFront, es decir si lo utilizado supera estos valores se genera un valor a pagar. Mientras que, Cloud CDN y Azure CDN son servicios que no incluyen recursos gratuitos.

Es así que Cloud CDN y Azure CDN generan un costo aproximado de 17 y 25 veces mayor que Cloudfront respectivamente.

El servicio es necesario incluirlo en una arquitectura Serverless Computing pues mejora la experiencia de usuario para el acceso a recursos de forma rápida, sin embargo, bajo este concepto la mejor solución es utilizar la red de entrega de contenido de AWS Cloudfront.

### **Amazon Cognito (AWS), Identity Platform y Azure Active Directory B2C**

Los tres servicios permiten el control de usuarios, el costo se lo calcula en función de usuarios activos mensuales también llamados MAU. Amazon Cognito, Identity Platform y Azure Active Directory B2C admiten 50000 usuarios. En caso de sobrepasar el límite se realiza un cobro por usuario. En el caso de Cognito e Identity Platform el costo es el mismo, mientras que en Azure el costo es menor.

Acorde a lo anterior es factible considerar que al crear una aplicación web bajo una arquitectura Serverless Computing en la que sea indispensable el control de usuarios la mejor solución es Azure Active Directory.

### **AWS Lambda, Cloud Functions y Azure Functions**

Los servicios de FasS de cada una de las plataformas representan el núcleo del proyecto, lo relevante de estos tres servicios es la gratuidad que ofrecen y el costo se lo realiza una vez se supere los límites establecidos en la tabla 3.9

**Tabla 3. 9.** Costos AWS, GCP y Azure

<b>Servicio</b>	<b>Duración mensual gratuita (GB-segundos)</b>	<b>Solicitudes mensuales gratuitas</b>	<b>Costo por millón de solicitudes adicionales</b>	<b>Costo de GB por segundo adicional</b>
AWS Lambda	400000	1 millón	\$0.20	\$0.000016
Cloud Functions	400000	2 millones	\$0.40	\$0.0000125
Azure Functions	400000	1 millón	\$0.20	\$0.000016

De acuerdo a los datos obtenidos Cloud Functions admite un mayor número de solicitudes mensuales gratuitas, en caso de requerir crear una aplicación web bajo la arquitectura Serverless Computing es una buena opción.

Además, en la tabla 3.9 es posible identificar AWS Lambda y Azure Functions cumplen con la misma oferta, mientras que Cloud Functions en GB por segundo adicional genera un costo menor.

En conclusión, aunque el precio es menor en GCP la investigación permitió identificar no basta considerar únicamente el costo de la solución Serverless que ofrece cada plataforma sino todos los servicios que conforman la aplicación web.

En base a la información obtenida Lambda, Cloud Functions y Azure Functions soportan gran cantidad de lenguajes de programación por lo que es posible escribir funciones que son activadas por eventos para el trabajo en conjunto con otros servicios permitiendo el uso de la aplicación web de forma óptima. El código es la parte principal al momento de crear una aplicación sin servidor, aunque Azure Functions admite mayor número de lenguajes de programación una consideración importante a la hora de elegir una plataforma es la elección del lenguaje que mayor conocimiento se tenga o se domine.

### **Amazon DynamoDB, Datastore y Azure CosmosDB**

Las bases de datos no relaciones o NoSQL, diferentes a las bases de datos tradicionales son compatibles con aplicaciones sin servidor, ya que proporcionan un rendimiento rápido para acceder a los datos con estructura flexible y ofreciendo un escalado automático.

DynamoDB y CosmosDB son las más conocidas para implementar la arquitectura Serverless Computing, aunque DynamoDB es de las primeras bases de datos de este tipo genera un costo más elevado que Datastore de GCP y CosmosDB de Azure, es decir Amazon DynamoDB no ofrece la mejor solución si de almacenamiento se trata.

Existen bases de datos que, aunque son más populares y necesarias para trabajar con functions, no siempre tendrán costos reducidos.

Tanto DynamoDB como CosmosDB ofrecen 25 GB de almacenamiento gratuito, pero en el caso de superar este valor se realiza un cobro por GB y ahí existe una diferencia en costos pues CosmosDB por GB tiene un costo más bajo.

### **3.3. Conclusiones**

La presente investigación es de gran importancia puesto que la tecnología Serverless Computing resulta ser una de las tendencias en los últimos años.

Como resultado final se obtuvo un análisis comparativo de los servicios que intervienen al crear una aplicación web bajo la arquitectura Serverless Computing de las plataformas Amazon Web Services, Google Cloud Platform y Azure Cloud Platform con sus soluciones sin servidor AWS Lambda, Google Functions y Azure Functions respectivamente.

La documentación existente en páginas oficiales de cada uno de los proveedores tiene infinidad de información referente a la creación de aplicaciones web, debido a aquello fue necesario establecer y limitar servicios que faciliten el entendimiento y funcionalidad de esta arquitectura.

- Serverless Computing es un cambio de paradigma que permite a los desarrolladores no preocuparse por el aprovisionamiento y las operaciones del servidor, la facturación se la realiza pago por solicitud de tal forma al desarrollar aplicaciones web será necesario aprovechar al máximo las plataformas y los servicios que ofrecen.
- Hoy en día Amazon Web Services es la plataforma de nube más conocida. En vista del número de servicios, como herramientas para desarrolladores y bases de datos que ofrece es la nube más adoptada y completa. Sin embargo, Azure Cloud Platform y Google Cloud Platform se han posicionado en los últimos años con un rápido crecimiento al ofrecer servicios similares a AWS, siendo evidente y se ha demostrado en esta investigación.

- Serverless Computing permite la creación de procesos mediante código a través de funciones, la ejecución de funciones genera eventos, eventos en los que intervienen otros servicios, lo cual genera una codependencia entre la oferta de servicios que ofrece cada uno de los proveedores de nube.
- Gracias al análisis de costos realizado, es posible identificar ciertas consideraciones al momento de elegir un proveedor para alojar y desplegar una aplicación web. El punto de partida como se pudo demostrar será identificar los servicios que se desee implementar según los requerimientos de la empresa u organización.
- Existe un sin número de servicios que ofrecen actualmente las plataformas AWS, GCP y Azure Cloud Platform. Sin embargo, los modelos de suscripción, crédito inicial, beneficios y costos mensuales al momento de crear una aplicación web difieren, es así que antes de elegir una plataforma en la cual crear la aplicación web siempre será necesario realizar un análisis de los servicios y recursos que se deseen acoplar.
- Las calculadoras que ofrecen las páginas oficiales de cada plataforma son de gran ayuda para obtener un valor referencial del costo de cada servicio e identificar la manera en que se realizan los cálculos, pero existen ciertos servicios que aún no están integrados. Así también, al momento de realizar los cálculos existen ciertos valores obtenidos que no consideran los recursos gratuitos, por estas razones siempre será necesario trabajar en conjunto con la información de precios que ofrecen las páginas oficiales y las calculadoras, de esta manera será posible establecer límites para aprovechar las ofertas que incluyen cada uno de los servicios.
- Los tres proveedores tienen disponibilidad regional global, por consiguiente, es necesario acogerse a los beneficios que cada una de las plataformas ofrece e identificar requerimientos de las aplicaciones con la finalidad de obtener el mejor beneficio.
- Antes de realizar el diseño de la aplicación web se estudiaron sin número de servicios que facilitan la creación de una aplicación bajo la Arquitectura Serverless Computing, en este proceso se identificaron servicios que pueden ser utilizados, pero no se requiere sobrepasar gastos por esta razón se seleccionaron servicios propios de cada plataforma que han sido creados con el enfoque de aplicaciones sin servidor.

- Los costos obtenidos por servicios y aplicación han sido de gran ayuda para identificar que beneficio obtenemos de cada uno de los proveedores, de acuerdo a ello es fundamental comprender que antes de elegir una plataforma para la creación de una aplicación siempre será indispensable un análisis previo.
- Una vez se haya creado una aplicación web sin servidor es necesario exista asesoría de una persona que tenga un alto conocimiento en aplicaciones en la nube, ya que de acuerdo a los recursos utilizados se genera un costo mensual o anual también llamado “pago por uso”, debido a ello será necesario ir optimizando gastos, realizando modificaciones en la estructura y en el código de la aplicación que generan costos más elevados.
- Actualmente Amazon y su plataforma AWS tienen mayor reconocimiento por el tiempo que está en el mercado, por sus beneficios, ofertas y servicios gratuitos. Empresas han logrado posicionarse gracias a la arquitectura Serverless Computing, aunque la idea de implementarlo resulta fabulosa en el proyecto se ha identificado que los costos son en función del consumo, pero en caso de sobrepasar recursos gratuitos los costos no difieren en gran proporción en las tres plataformas. Por esta razón, fue necesario realizar un estudio a fondo de todas las dependencias necesarias para crear aplicaciones y el plazo de tiempo que permitirá obtener un costo reducido y analizar el peor de los escenarios.

En conclusión, las tres plataformas AWS, GCP y Azure Cloud Platform ayudarán a la creación de una aplicación web utilizando la arquitectura Serverless Computing los servicios que ofertan son similares y permiten realizar un análisis comparativo de los costos que implica, en el modelo de aplicación web establecido en el proyecto AWS y GCP resultan la mejor solución considerando generan los costos más bajos.

Con este proyecto se permite comprender mejor la tecnología y asumir no siempre el posicionamiento en el mercado de las diferentes plataformas o beneficios que ofrecen hacen que sea la mejor opción pues cada componente que conforma la aplicación tiene costos no entendidos a primera vista razón por la cual se realizó un estudio a profundidad.

### **3.4. Recomendaciones**

El estudio comparativo de las tres plataformas basado en la arquitectura Serverless Computing fue de gran ayuda para desarrollar un esquema de costos y beneficios que ofrecen cada uno de los proveedores de nube y la ventaja de crear aplicaciones modernas gracias a Cloud Computing.

El proyecto permite identificar claramente pautas a considerar una vez se desee crear una aplicación web sin servidor, sin embargo, existen posibles estudios futuros que se mencionan a continuación:

- La aceleración del crecimiento y avance tecnológico cada vez es sorprendente, debido a ello este estudio abre puertas a identificar soluciones Serverless futuras que faciliten aún más crear aplicaciones web.
- La competencia que existe entre los tres proveedores de nube Amazon, Google y Microsoft genera cambios en los modelos de suscripción, ofertas y servicios gratuitos que ofrecen, siendo un tema importante a considerar al momento de crear una aplicación pues este proyecto permitirá realizar estudios a futuro de los cambios que se realicen en cada una de las plataformas.

Para mejorar el estudio de aplicaciones web en una arquitectura Serverless Computing será recomendable tomar en cuenta las siguientes consideraciones:

- El análisis se ha realizado con cálculos teóricos de acuerdo a los costos que indican páginas referenciales en un escenario en el cual se supere los límites gratuitos, se recomienda crear una aplicación real e ir evaluando los costos reales con los teóricos presentados en esta investigación.
- En este proyecto se creó una aplicación web considerando cargas de trabajo pequeñas, se recomienda crear un modelo de aplicación web basado en estos servicios para empresas u organizaciones mundiales bajo las consideraciones presentadas en este proyecto y analizar cuál es la mejor solución.
- Se recomienda manejar términos en inglés referente a redes de servidores, servicios, protocolos, lenguajes de programación y tecnología en general dado que la mayoría de la información relevante se encuentra en documentos con este idioma debido a que Serverless Computing es una tendencia tecnológica que surgió en años recientes y forma parte de una nueva generación de productos y servicios.

#### **4 REFERENCIAS BIBLIOGRÁFICAS**

- [1] R. Vemula, Integrating Serverless Architecture Using Azure Functions, Cosmos DB, and SignalR Service, Visakhapatnam, Andhra Pradesh, 2019, pp. 3-5.
- [2] Instituto Nacional de Ciberseguridad, «Cloud computing. Una guía de aproximación para el empresario,» 2017. [En línea].

- [3] Azure, «What is a cloud service provider?,» [En línea]. Disponible en: <https://azure.microsoft.com/es-es/overview/what-is-a-cloud-provider/> (accedido nov. 15,2021).
- [4] AWS, «DEEP ANALYSIS CATEGORY Serverless Technology,» [En línea]. Disponible en: <https://aws.amazon.com/es/getting-started/deep-dive-serverless/> (accedido nov. 16,2021).
- [5] Microsoft, «Escenarios y casos de uso empresariales sin servidor,» 27 septiembre 2021. [En línea]. Disponible en: <https://docs.microsoft.com/es-es/dotnet/architecture/serverless/serverless-business-scenarios> (accedido nov. 16,2021).
- [6] S. Rosenbaum, de Serverless Computing in Azure with .NET, 2017.
- [7] M. Stigler, Beginning Serverless Computing, Richmond, Virginia, 2018, pp. 1-2.
- [8] M. Tuba, S. Akashe y A. Joshi, «ICT Systems and Sustainability,» vol. 1, Ahmedabad, Gujarat, 2021, pp. 376-378.
- [9] E. Brown, «Web Development with Node and Express,» Segunda ed.
- [10] J. Añel, D. Montes y J. Iglesias, Cloud and Serverless Computing for Scientists, 2020.
- [11] TeachTarget , AWS.
- [12] AWS, API Gateway.
- [13] C. Kaewkasi, Docker for Serverless Applications, 2018.
- [14] A. S. Cabezuelo, Introducción a las bases de datos NoSQL usando MongoDB, Catalunya, 2016.
- [15] Y. Wadia y U. Gupta, Mastering AWS Lambda, 2017.
- [16] AWS, «Informática en la nube con AWS,» [En línea]. Disponible en: <https://aws.amazon.com/es/what-is-aws/> (accedido nov. 22,2021).
- [17] J. J. Geewax, Google Cloud Platform in action, 2018.
- [18] T. Hunter y S. Porter, Google Cloud Platform for Developers, Julio, 2018, p. 10.
- [19] M. Copeland, J. Soh, A. Puca, M. Manning y D. Golllob, Microsoft Azure Planning, Deploying, and Managing Your Data Center in the cloud.

- [20] R. Sawhney, Beginning Azure Functions. Building Scalable and Serverless Apps.
- [21] AWS, «¿Qué es Amazon CloudFront?,» [En línea]. Disponible en: [https://docs.aws.amazon.com/es\\_es/AmazonCloudFront/latest/DeveloperGuide/Introduction.html](https://docs.aws.amazon.com/es_es/AmazonCloudFront/latest/DeveloperGuide/Introduction.html) (accedido nov. 23,2021).
- [22] Halodoc, «Implementación de aplicaciones web estáticas con AWS S3, CloudFront y sus beneficios,» 31 marzo 2020. [En línea]. Disponible en: <https://blogs.halodoc.io/deploy-static-web-application-using-s3-and-cloudfront/> (accedido nov. 23,2021).
- [23] Amazon Web Services, «Construyendo APIs usando Amazon API Gateway y AWS Lambda,» [En línea]. Disponible en: <https://pages.awscloud.com/rs/112-TZM-766/images/Construyendo%20APIs%20usando%20Amazon%20API%20Gateway%20y%20AWS%20Lambda.pdf> (accedido dic. 3,2021).
- [24] Google Cloud, «Cloud CDN,» [En línea]. Disponible en: <https://cloud.google.com/cdn> (accedido dic. 3,2021).
- [25] Google Cloud, «Google Cloud Storage Triggers,» [En línea]. Disponible en: <https://cloud.google.com/functions/docs/calling/storage>.
- [26] Google Cloud, «Identity Platform,» [En línea]. Disponible en: <https://cloud.google.com/identity-platform#:~:text=Identity%20Platform%20is%20a%20customer,with%20confidence%20on%20Google%20Cloud> (accedido dic. 10,2021).
- [27] Google Cloud, «API Gateway,» [En línea]. Disponible en: <https://cloud.google.com/api-gateway> (accedido dic. 10,2021).
- [28] Google Cloud, «Cloud Functions,» [En línea]. Disponible en: <https://cloud.google.com/functions#section-7> (accedido dic. 14,2021).
- [29] Azure, «Azure Blob Storage,» [En línea]. Disponible en: [https://azure.microsoft.com/es-mx/services/storage/blobs/?&ef\\_id=CjwKCAiA6seQBhAfEiwAvPqu1wkHtMWRO3RiS-dazlisfz1a9-Wd27Ron6vQM9cjuYkJ3o\\_Cz-AmxBoCOE0QAvD\\_BwE:G:s&OCID=AID2201052\\_SEM\\_CjwKCAiA6seQBhAfEiwAvPqu1wkHtMWRO3RiS-dazlisfz1a9-Wd27Ron6vQM9cjuYkJ3o\\_Cz-](https://azure.microsoft.com/es-mx/services/storage/blobs/?&ef_id=CjwKCAiA6seQBhAfEiwAvPqu1wkHtMWRO3RiS-dazlisfz1a9-Wd27Ron6vQM9cjuYkJ3o_Cz-AmxBoCOE0QAvD_BwE:G:s&OCID=AID2201052_SEM_CjwKCAiA6seQBhAfEiwAvPqu1wkHtMWRO3RiS-dazlisfz1a9-Wd27Ron6vQM9cjuYkJ3o_Cz-) (accedido dic. 16,2021).

- [30] Microsoft Azure, «Welcome to Azure Cosmos DB,» [En línea]. Disponible en: <https://docs.microsoft.com/en-us/azure/cosmos-db/> (accedido dic. 19,2021).
- [31] AWS, «AWS Free Tier,» [En línea]. Disponible en: [https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=\\*all&awsf.Free%20Tier%20Categories=\\*all&all-free-tier.q=cloudfront&all-free-tier.q\\_operator=AND](https://aws.amazon.com/free/?all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=*all&awsf.Free%20Tier%20Categories=*all&all-free-tier.q=cloudfront&all-free-tier.q_operator=AND).
- [32] AWS, «Cloud Front Pricing,» [En línea]. Disponible en: [https://aws.amazon.com/cloudfront/pricing/?loc=ft#AWS\\_Free\\_Usage\\_Tier](https://aws.amazon.com/cloudfront/pricing/?loc=ft#AWS_Free_Usage_Tier) (accedido ene. 2,2022).
- [33] AWS, «Amazon Cognito,» [En línea]. Disponible en: [https://aws.amazon.com/cognito/pricing/?loc=ft#Free\\_Tier](https://aws.amazon.com/cognito/pricing/?loc=ft#Free_Tier) (accedido ene. 2,2022).
- [34] AWS, «Amazon API Gateway,» [En línea]. Disponible en: <https://calculator.aws/#/createCalculator/APIGateway> (accedido ene. 2,2022).
- [35] AWS, «AWS Lambda,» [En línea]. Disponible en: [https://aws.amazon.com/lambda/?did=ft\\_card&trk=ft\\_card](https://aws.amazon.com/lambda/?did=ft_card&trk=ft_card) (accedido ene. 2,2022).
- [36] AWS, «Amazon Dynamodb Pricing,» [En línea]. Disponible en: [https://aws.amazon.com/dynamodb/pricing/?loc=ft#Free\\_tier](https://aws.amazon.com/dynamodb/pricing/?loc=ft#Free_tier) (accedido ene. 2,2022).
- [37] Google Cloud, «Precios de Google Cloud,» [En línea]. Disponible en: <https://cloud.google.com/pricing> (accedido ene. 10,2022).
- [38] Google Cloud, «Api Gateway Pricing,» [En línea]. Disponible en: <https://cloud.google.com/api-gateway/pricing> (accedido ene. 12,2022).
- [39] Google Cloud, «Functions Pricing,» [En línea]. Disponible en: <https://cloud.google.com/functions/pricing> (accedido ene. 12,2022).
- [40] Google Cloud, «Data Store,» [En línea]. Disponible en: <https://cloud.google.com/datastore> (accedido ene. 12,2022).
- [41] Azure Cloud, «Cree soluciones en la nube con una cuenta gratuita de Azure,» [En línea]. Disponible en: [https://azure.microsoft.com/es-mx/free/search/?&ef\\_id=Cj0KCQiApL2QBhC8ARIsAGMm-KHbGCH4k7wtT4K74ejla0JIJBnBopoJ\\_nP7SSG-](https://azure.microsoft.com/es-mx/free/search/?&ef_id=Cj0KCQiApL2QBhC8ARIsAGMm-KHbGCH4k7wtT4K74ejla0JIJBnBopoJ_nP7SSG-)

Dsq83OBFtkog81kaAtQaEALw\_wcB:G:s&OCID=AID2201052\_SEM\_Cj0KCQiApL2QBhC8ARIsAGMm-KHbGCH4k7wtT4K74ejla0JIJBnBopoJ\_nP7SSG-Dsq83OBFtkog81kaAtQa (accedido ene. 12,2022).

- [42] Azure Microsoft, «API Management pricing».
- [43] Azure Microsoft, «Functions Azure Pricing,» [En línea]. Disponible en: <https://azure.microsoft.com/en-us/pricing/details/functions/> (accedido ene. 14,2022).
- [44] J. Katzer, Learning Serverless Design, Develop, and Deploy with Confidence, 2020.
- [45] B. Aumaille, J2EE Desarrollo de Aplicaciones Web, Barcelona: Francesc GARCIA, p. 26.
- [46] Google Cloud, «Identity Platform Pricing,» [En línea]. Disponible en: <https://cloud.google.com/identity-platform/pricing> (accedido feb. 12 ,2022).
- [47] Stackshare, «Amazon CloudFront,» [En línea]. Disponible en: <https://stackshare.io/stackups/amazon-cloudfront-vs-google-cloud-cdn> (accedido feb. 4,2022).