

**Pontificia Universidad Católica del Ecuador**

**Facultad De Ingeniería**

**Escuela de Sistemas**



**TEMA:**

ANÁLISIS COMPARATIVO DE LAS ARQUITECTURAS SERVERLESS, QUE LOS  
DIFERENTES PROVEEDORES, OFRECEN EN LA NUBE ORIENTADO A LA CREACIÓN DE  
UNA APLICACIÓN WEB.

**AUTOR:**

JOSÉ NICOLÁS MAYANQUER ROSERO

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS Y  
COMPUTACIÓN

**QUITO, 24 de noviembre del 2023**

## DEDICATORIA

---

Este gran logro de mi vida se lo quiero dedicar a mis padres los cuales a parte de la ayuda económica que me han dado, han sido mi mayor fuente de apoyo y mi mayor orgullo, aparte de ser un ejemplo de perseverancia y fuerza que me ha servido para poder culminar cada cosa importante que me lo proponga. A mis abuelitos Tabita y Papijo que han sido como mis segundos padres para mi durante toda mi vida, han sido mi fuente de motivación más grande para nunca rendirme y dar todo de mí, aunque el cansancio sea más. Valoro mucho cada sacrificio que hicieron para yo poder culminar esta gran etapa de mi vida, la cual me va a abrir las puertas para poder ser un futuro profesional, gracias a los valores que me han dado cada uno de ustedes.

Gracias por ser siempre sacar la mejor versión de mí mismo, les dedico mi tesis con amor y gratitud eterna.

## AGRADECIMIENTO

---

Quiero expresar mi agradecimiento a mi tutor de tesis que siempre estuvo pendiente de mí y a pesar del corto tiempo que a veces yo tenía, siempre sacaba un tiempo para ayudarme y nunca abandonarme. Agradezco de la manera más profunda a mis compañeros de carrera han sido mi segunda familia durante esta etapa de mi vida y de las mejores personas que he podido conocer juntos a ellos he podido conseguir muchos logros, muchas risas, triunfos y desafíos. Agradecer a los profesores de la universidad que siempre han sacado la mejor versión de mí, para poder seguir aprendiendo y nunca quedarme en la mediocridad.

Gracias por ser mi fuente de inspiración y por hacer de esta experiencia educativa un viaje memorable.

## RESUMEN

---

El presente trabajo de titulación pretende dar a conocer un poco más esta arquitectura que muchas empresas ya están optando por utilizarla y muchas otras aún no conocen del gran potencial que tiene trabajar bajo la arquitectura serverless y los grandes beneficios que conlleva. Este trabajo de titulación se divide en [cuatro](#) partes fundamentales, la primera donde se describe la problemática y que se intenta alcanzar con el presente proyecto, la segunda parte es la introducción a que es una arquitectura serverless, que es el cloud computing, que son proveedores [de servicio en](#) la nube, que es una aplicación web y los componentes que vamos a utilizar para la creación de nuestra aplicación bajo la estructura serverless, también se describe brevemente que es un contenido estático y dinámico dentro de esta infraestructura. La tercera parte se muestra la estructura de nuestra aplicación web y los servicios que se usan para que sea serverless y se describe cada una de esta, para nuestro análisis comparativo usaremos las [tres](#) plataformas en la nube más [famosas utilizadas](#) al día de hoy, las cuales son AWS, Azure Cloud Platform y Google Cloud Platform, la última parte del [presente este](#) proyecto presenta el modelo de costos que [se](#) tiene [para](#) trabajar con cada una de estas tres plataformas seguido de un análisis de cuál de las plataformas es mejor utilizar en función de las características de sus servicios, disponibilidad y costo.

## ÍNDICE

---

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS .....	5
ÍNDICE DE FIGURAS .....	5
ÍNDICE DE TABLAS.....	6
1.    Marco de referencias .....	8
1.1.    Justificación .....	8
1.2.    Planteamiento del problema.....	8
1.3.    Objetivo General.....	9
1.4.    Objetivos Específicos .....	9
1.5.    Antecedentes.....	9
1.6.    Alcance .....	10
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA .....	11
2.    Marco Teórico .....	11
2.1.    Cloud Computing.....	11
2.1.1.    Infraestructura como servicio (IaaS).....	11
2.1.2.    Plataforma como servicio (PaaS). .....	12
2.1.3.    Software como servicio (SaaS).....	12
2.1.4.    Función como servicio (FaaS) .....	13
2.2.    Infraestructura subyacente.....	13
2.3.    Proveedor de servicios en la nube.....	13

2.4. Arquitectura Serverless .....	14
2.5. Aplicación Web .....	15
2.6. Contenido Estático (Frontend) .....	15
2.7. Contenido dinámico (Backend) .....	16
CAPÍTULO III: METODOLOGÍA .....	20
3.1. Metodología de desarrollo de tesis .....	20
CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN.....	22
4.1. Introducción .....	22
4.2. Amazon Web Services (AWS) .....	24
4.2.1. AWS Lambda .....	24
4.3. Azure Cloud Platform .....	24
4.3.1. Azure Function .....	25
4.4. Google Cloud Platform (GCP) .....	25
4.4.1. Cloud Functions .....	25
4.5. Diseño de la aplicación web .....	25
4.5.1. Arquitectura de la Aplicación Web.....	26
4.5.1.1. Serverless Computing en AWS.....	26
4.5.1.2. Serverless Computing en Azure Cloud Platform .....	30
4.5.1.3. Serverless Computing en Google Cloud Platform .....	33
CAPÍTULO V: MODELO DE COSTOS .....	<del>37</del> 36
5.1. Calculadora de servicios AWS, ACP y GCP .....	<del>37</del> 36

5.2. Amazon Web Services.....	37
5.2.1. Amazon S3.....	38
5.1.2. Amazon Cognito.....	<del>39</del> 38
5.1.3. Amazon API Gateway.....	<del>40</del> 39
5.1.4. AWS Lambda.....	40
5.1.5. Amazon DynamoDB .....	<del>42</del> 41
5.1.6. Amazon Cloudfront .....	42
5.2. Microsoft Azure .....	44
5.2.1. Azure Blob Storage .....	44
5.2.2. Azure Active Directory B2C .....	45
5.2.3. Azure API Management.....	45
5.2.4. Azure Functions .....	46
5.2.5. Azure Cosmos DB .....	47
5.2.6. Azure Content Delivery Network .....	48
5.3. Google Cloud Platform.....	49
5.3.1. Cloud Storage .....	49
5.3.2. Identity Platform.....	50
5.3.3. API Gateway .....	50
5.3.4. Cloud Functions .....	51
5.3.5. Datastore.....	52
5.3.6. Cloud CDN.....	53

5.4. Resultados.....	54
5.4.1. Costos finales por mes .....	55
5.4.2. Análisis Comparativo .....	57
CONCLUSIONES Y RECOMENDACIONES .....	61
BIBLIOGRFÍA.....	<del>6463</del>
GLOSARIO DE TÉRMINOS .....	<del>6766</del>
ANEXOS.....	<del>6968</del>

## ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

---

### ÍNDICE DE FIGURAS

Figura 1 Backend vs Frontend. Fuente: (Basílio, 2022).....	19
Figura 2. Cloud Storage Market. Fuente: (Mordor Intelligence, 2023) .....	<del>2224</del>
Figura 3 Estructura la aplicación Web en AWS con arquitectura serverless .....	<del>2726</del>
Figura 4 Estructura de la aplicación Web en ACP con arquitectura serverless .....	<del>3130</del>
Figura 5 Estructura la aplicación Web en GCP con arquitectura serverless.....	<del>3433</del>
Figura 6 Calculo de solicitudes S3 .....	<del>3937</del>
Figura 7. Costos MAU Fuente (AWS, 2023) .....	<del>3938</del>
Figura 8 Costo AWS Cloudfront por región Fuente (AWS, 2023) .....	<del>4342</del>
Figura 9 Tabla de Precios Azure CDN Fuente (Azure,2023).....	<del>4847</del>
Figura 10 Costos de MAU en Identity Platform .....	<del>5049</del>
Figura 11 Costo por cada millón de llamadas API Fuente (Google Cloud, 2023).....	<del>5150</del>
Figura 12 Tabla de costo Cloud CDN Fuente (Google Cloud, 2023) .....	<del>5352</del>

## ÍNDICE DE TABLAS

Tabla 1 Proveedores Cloud y Plataformas Serverless .....	23
Tabla 2 Servicios de Amazon Web Services .....	26
Tabla 3 Servicios de Azure Cloud Platform .....	30
Tabla 4 Servicios de Google Cloud Platform .....	33
Tabla 5 Calculadora de servicios AWS, ACP y GCP .....	37
Tabla 6 Costo mensual de Amazon S3 .....	38
Tabla 7 Ejemplo mensual Cognito .....	39
Tabla 8 Costo mensual API Gateway .....	40
Tabla 9 Costo mensual de AWS Lambda .....	41
Tabla 10 Costo mensual Amazon DynamoDB .....	42
Tabla 11 Costo mensual Amazon Cloudfront .....	43
Tabla 12 Costo mensual Cloud Storage .....	45
Tabla 13 Costo mensual de Azure Active Directory B2C .....	45
Tabla 14 Costo mensual API Gateway .....	46
Tabla 15 Costo mensual de Azure Functions .....	47
Tabla 16 Costo mensual Azure Cosmos DB .....	48
Tabla 17 Costo mensual Azure CDN .....	49
Tabla 18 Costo mensual de Cloud Storage .....	49
Tabla 19 Costo mensual de Identity Platform .....	50
Tabla 20 Costo mensual API Gateway .....	51
Tabla 21 Costo mensual Google Cloud Functions .....	52
Tabla 22 Costo mensual DataStore .....	53
Tabla 23 Costo mensual CDN .....	54
Tabla 24 Costo por mes Cloud Storage de AWS, ACP y GCP .....	55
Tabla 25 Costo por mes Authenticate de AWS, ACP y GCP .....	55

Tabla 26 Costo por mes API Gateway de AWS, ACP Y GCP .....	56
Tabla 27 Costo por mes Functions de AWS, ACP y GCP .....	56
Tabla 28 Costo por mes de DB de AWS, ACP y GCP.....	56
Tabla 29 Costo por mes CDN de AWS, ACP y GCP.....	57
Tabla 30 Costo de todos los servicios por mes.....	57

## CAPÍTULO I: INTRODUCCIÓN

---

### 1. Marco de referencias

#### 1.1. Justificación

Hoy en día las empresas no cuentan con un buen presupuesto para las áreas de TI y de acuerdo con estudios realizados por instituciones los costos se abaratan al usar aplicaciones serverless que utilizan la nube. Por eso, "Serverless, es el siguiente paso a la evolución de los sistemas computacionales en la nube. Estas tecnologías se basan en funciones atómicas y similitudes con respecto lo que es la computación distribuida y esto desconocen las empresas que trabajan sin tener conocimiento de esta arquitectura la cual solo traerá beneficios a la empresa." (Izquierdo, D. (2022)). Se cuentan con varias de las arquitecturas de software para aplicaciones serverless, pero que lamentablemente se desconoce la funcionalidad y no se cuenta con suficiente información de sus beneficios que permitan analizar su implementación en las empresas.

A través del presente trabajo de titulación se pretende analizar las ventajas y desventajas de las aplicaciones serverless, frente a la creación de aplicaciones web. También se va a analizar, como con esta arquitectura serverless, una empresa va a tener una gran cantidad de servicios, previo al análisis de costos vs beneficio para la empresa y sus usuarios.

#### 1.2. Planteamiento del problema

En la actualidad se dispone de proveedores de servicios de internet en la nube que ofrecen servicios como los denominados SaaS (Software como servicio), PaaS (Plataforma como servicio), IaaS (Infraestructura como servicio), FaaS (Función como servicio), entre otros, que son los modelos de servicio más conocidos, pero lamentablemente las empresas conocen poco el funcionamiento de la nube, como funcionan la arquitectura serverless y como esta beneficia a su giro de negocio.

Adicionalmente, es conocido que las empresas se dedican al giro de su negocio, desconociendo las ventajas y desventajas de contar con servicios de la nube. Por este motivo se pretende mostrar cuáles son las mejores opciones de aplicaciones serverless para poder manejar la creación de aplicaciones web y ejecutar aplicaciones con rapidez y a menor costo total de propiedad, por lo que, no es necesario aprovisionar y administrar infraestructura.

### **1.3. Objetivo General**

Realizar un análisis comparativo de las aplicaciones serverless, que los diferentes proveedores ofrecen en la nube para la creación de aplicaciones web.

### **1.4. Objetivos Específicos**

1. Examinar los diferentes proveedores de aplicaciones serverless.
2. Analizar las principales aplicaciones serverless disponibles en la nube.
3. Explicar los componentes que debe tener una aplicación web desarrollada y explicar los servicios necesarios para su implementación.
4. Desarrollar un análisis comparativo de los costos ofrecidos por los proveedores de aplicaciones serverless.

### **1.5. Antecedentes**

Hoy en día se desconoce sobre la informática en la nube, y más aún el concepto de computación sin servidor, conocido como "serverless," ha surgido como una innovación disruptiva en el desarrollo de aplicaciones web. Este enfoque se caracteriza por prescindir de la necesidad de servidores físicos y permite la ejecución de funciones de manera reactiva ante eventos específicos. En consecuencia, este paradigma ha transformado radicalmente la forma en que se conciben construye y escalan las aplicaciones web. El presente trabajo de investigación profundiza en el mundo de las arquitecturas serverless, sumergiéndose en la historia y los antecedentes que dieron origen a esta tendencia. Se explora detenidamente cómo esta tecnología se relaciona con la creación de aplicaciones web y cómo ha dejado una huella

significativa en la industria tecnológica. Se va a realizar la comparación de 3 proveedores los cuales hoy en día son los más importantes, el primero es AWS el cual es uno de los favoritos a nivel mundial, sin embargo tienes varios factores los cuales lo hacen menos preferidos frente a su competencia, estos pueden ser por costos, complejidad, soporte y localización geográfica de sus servidores, por otro lado tenemos a Azure, este es uno de los favoritos para las empresas ya que la ubicación geográfica de sus servidores lo hace más factible, sin embargo, tiene algunas desventajas la cuales podrían variar entre costos, complejidad, ecosistema de desarrollo, flexibilidad y neutralidad en la nube; para finalizar tenemos a Google Cloud Function, de las dos ya mencionadas, esta es la menos favorita para las empresas, esto se debe a su menor presencia regional, complejidad de la oferta de servicios, soporte y servicios profesionales y por último y más importante para algunas empresas son los requisitos de cumplimiento y regulaciones.

#### **1.6. Alcance**

El presente proyecto de titulación culminará con la entrega de un documento final el cual contendrá:

- Análisis de las aplicaciones serverless en la nube.
- Análisis de los diferentes proveedores de aplicaciones serverless en la nube.
- Análisis comparativo sobre las principales aplicaciones serverless que ofrecen en la nube de diferentes proveedores.
  - Tales como: AWS, Azure y Google Cloud Functions.
- Desarrollo de un análisis de costos de las aplicaciones serverless para la creación de aplicaciones web.

## CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

---

### 2. Marco Teórico

#### 2.1. Cloud Computing

Cloud Computing como un término simple se refiere a la entrega de servicios de computación vía internet. Esto quiere decir que en vez de almacenar las aplicaciones o datos en servidores locales o en hardware físico, los usuarios pueden acceder a recursos informáticos, tales como servidores, bases de datos, redes, software y más, a través de la nube, la cual es una red de servidores remotos interconectados. Tiene muchas ventajas la computación en la nube las cuales son la escalabilidad, la flexibilidad, el ahorro de costos, la accesibilidad desde cualquier lugar con conexión a internet y la capacidad de externalizar la gestión de la infraestructura informática. Permite a las empresas y usuarios acceder a recursos informáticos sin la necesidad de inversiones significativas en hardware y mantenimiento de esta, los principales modelos de servicio en la nube son infraestructura como servicio (IaaS), plataforma como servicio (PaaS), software como servicio (SaaS) y función como servicio (FaaS) los cuales se detallarán a continuación.

##### 2.1.1. Infraestructura como servicio (IaaS)

Los proveedores de servicios en la nube de este tipo de modelo ofrecen recursos informáticos virtuales, tales como servidores virtuales, almacenamiento y redes, a los usuarios. Una de las ventajas en este tipo de modelo es que los usuarios pueden configurar y gestionar estos recursos según sus necesidades o las necesidades de la empresa. Un claro ejemplo, de este servicio es Amazon Web Services, ya que este tiene servidores virtuales como EC2, almacenamiento escalable, redes y base de datos. Al trabajar con IaaS, se tiene varias ventajas como son la escalabilidad, costo-efectividad, flexibilidad y rapidez de implementación, y como desventajas podríamos tener la dependencia que tiene con el proveedor, y a su vez la seguridad

y privacidad, ya que los datos y aplicaciones se vuelven parte de la responsabilidad compartida con el proveedor.

### **2.1.2. Plataforma como servicio (PaaS).**

Los proveedores de servicios en la nube de este modelo ofrecen una plataforma de desarrollo la cual incluye herramientas y servicios para facilitar la creación, implementación y gestión de aplicaciones. En este modelo los desarrolladores se pueden enfocar en el desarrollo de software sin la preocupación por la infraestructura inferior. Un ejemplo de este servicio es Google App Engine, este pertenece a Google Cloud Platform y permite a los desarrolladores crear y albergar aplicaciones móviles y web de forma más sencilla que otros métodos. Cuando hablamos de ventajas de este servicio tenemos la escalabilidad automática, ahorro de tiempo y recursos, se preocupa menos sobre la infraestructura y se facilita el desarrollo de aplicaciones, por otro lado, también tenemos desventajas, pero la que más afecta al sector empresarial, es que sus costos aumentan cuando sus servicios son usados a largo plazo, y su producto presenta un crecimiento significativo.

### **2.1.3. Software como servicio (SaaS)**

En este modelo, los usuarios o empresas que lo utilicen pueden acceder a aplicaciones software a través de internet sin tener que preocuparse por la gestión de servidores o la infraestructura. Uno de los ejemplos de SaaS que se utiliza en su mayoría es la de correo electrónico basadas en la nube como Gmail y suites de productividad como Microsoft 365. Posee ventajas bastante significativas como es el ahorro de costos, facilidad de acceso, actualizaciones automáticas y una implementación rápida ya que no es necesario instalar ni mantener software en servicios locales, una desventaja de este servicio vendría a ser su dependencia netamente con el proveedor.

#### **2.1.4. Función como servicio (FaaS)**

Es un modelo de computación en la nube el cual se centra en la ejecución de funciones individuales de software o "functions" en respuesta a eventos o solicitudes específicas. En vez de ejecutar una aplicación continua o un servidor, las funciones de FaaS son unidades de código que se activan en respuesta a eventos las cuales proceden a detenerse una vez a finalizado su tarea. Estas funciones pueden ser desarrolladas en lenguajes de programación como Python, JavaScript, Java, entre otros. Un ejemplo de este servicio es el AWS Lambda.

Una de las ventajas de FaaS es que es útil para trabajos que pueden ser altamente escalables y rápidos, como es la manipulación de datos, el procesamiento de eventos en tiempo real, la automatización de tareas, la generación de informes, etc. Los servicios de FaaS más conocidos incluyen AWS Lambda (Amazon Web Services), Azure Functions (Microsoft Azure), y Google Cloud Functions (Google Cloud), los cuales van a ser utilizados para realizar este trabajo de titulación.

#### **2.2. Infraestructura subyacente**

En el contexto de tecnología y de informática, esta infraestructura se refiere a los componentes tanto de hardware como de software que respalda y mantienen una plataforma de tecnología o un servicio en la nube.

#### **2.3. Proveedor de servicios en la nube**

Es una entidad u organización la cual ofrece servicios de computación en la nube a usuarios o a empresas las cuales requieran este servicio. Su manera de trabajar es mediante centros de datos, y con esto ofrecen una amplia gama de servicios a través del internet, esto permite al cliente gestionar datos, aplicaciones y servicios dentro de la nube, sin necesidad de tener una infraestructura de TI.

#### 2.4. Arquitectura Serverless

La arquitectura serverless, o "sin servidor", es una orientación de diseño de aplicaciones en la informática la cual está en la nube y se caracteriza por eliminar la necesidad de gestionar servidores de manera explícita por parte de los desarrolladores. Aunque el nombre puede ser un tanto engañoso, no significa que no haya servidores en absoluto, sino que los detalles de la infraestructura subyacente son administrados por el proveedor de servicios en la nube, permitiendo a los desarrolladores centrarse en el código de la aplicación y en la lógica del negocio.

Las características más importantes de esta arquitectura son:

- **Escalabilidad:** Los proveedores de servicios en la nube, administran de manera automática la escalabilidad de la infraestructura, lo que hace que la aplicación pueda manejar de forma automática aumentos de carga, sin necesidad de intervención por parte de los usuarios.
- **Pago por uso:** Los costos de utilizar este tipo de arquitectura están directamente relacionados con el consumo real de recursos, como el tiempo de CPU y la memoria utilizada.
- **Tiempo de ejecución:** Las funciones serverless se ejecutan en contenedores o buckets que son más rápidos estos se inician cuando se necesita y se apagan cuando termina la ejecución, lo que ahorra recursos y permite una mayor flexibilidad.
- **Alta disponibilidad y tolerancia a fallas:** Sin necesidad de hacer algún proceso para replicar datos en diferentes zonas de disponibilidad (lugares geográficamente ubicados para garantizar disponibilidad y tolerancia a fallos en la nube), los servicios sin servidor ofrecen alta disponibilidad y tolerancia a fallas de forma predeterminada.

## **2.5. Aplicación Web**

Una aplicación web es un tipo de software el cual se ejecuta en un navegador web, como Google Chrome, Mozilla Firefox, Microsoft Edge, Opera, entre otros, en lugar de ser instalado directamente en una computadora. A diferencia de las aplicaciones de escritorio comunes que ya todos conocemos, las aplicaciones web residen en servidores remotos y se acceden a través de internet.

Ahora hablando dentro de lo que es una arquitectura serverless, una aplicación web es un tipo de software en línea el cual aprovecha un enfoque de cómputo serverless para encargarse de su infraestructura del servidor, este necesita servicios y componentes web que faciliten su ejecución dentro del marco del desarrollo. Cuando hablamos del término "serverless" este no implica la ausencia de servidores, esto hace alusión a que los desarrolladores no tienen que lidiar con la gestión ni el mantenimiento de servidores físicos o virtuales de manera directa. En su lugar, este trabajo recae en el proveedor de servicios en la nube como por ejemplo (Amazon, Google, Azure, entre otros), quien se encarga de administrar la infraestructura, escalabilidad y los recursos.

## **2.6. Contenido Estático (Frontend)**

Cuando hablamos de contenido estático este es la información en una plataforma digital la cual no cambia automáticamente ni se adapta a las acciones del usuario, este permanece constante y predefinido. Algunos ejemplos de contenido estático son imágenes, archivos de texto, videos, hojas de estilo y elementos de diseño en una página web, por este motivo a este contenido también se lo conoce como Frontend.

Contenido Estático en Arquitectura Serverless:

- En una arquitectura serverless se incluye elementos como HTML, CSS, imágenes y archivos JavaScript que no cambian con frecuencia con respecto a otras arquitecturas.

- Estos recursos estáticos se pueden albergar en servicios de almacenamiento de objetos, como Amazon S3, Google Cloud Storage o Azure Blob Storage. Estos servicios ofrecen alta disponibilidad y escalabilidad.
- Los archivos estáticos con los que se trabaja se pueden servir directamente desde el almacenamiento de objetos sin necesidad de servidores dedicados, esto es un gran ahorro de costos y reduce la complejidad.

Para poder trabajar con el frontend desde la arquitectura serverless es importante identificar los servicios que permitirán almacenar el frontend o contenido estático, estos servicios son:

➤ **Content Delivery Network (CDN):**

Parte del frontend, la cual sirve para que el usuario pueda interactuar mejor con la aplicación web, esto gracias a la administración del tráfico que procesa las solicitudes del usuario de manera más rápida. En resumen, este servicio esto permite entregar contenido web estático de una manera más global, confiable gracias a sus proveedores y eficiente, esto gracias a una red geográfica que alberga ubicaciones de caché.

➤ **Cloud Storage:**

Este servicio como dice su nombre es el almacenamiento en la nube, este es importante para guardar cualquier cantidad de datos, memoria, entre otros. Tiene funciones para que se pueda optimizar, organizar y configurar el acceso a los datos, como una ventaja de este servicio es que se puede recuperar los datos cuando el usuario necesite.

## **2.7. Contenido dinámico (Backend)**

El contenido dinámico o backend es la información de una plataforma digital la cual cambia en respuesta a la interacción del usuario o a eventos específicos como días en los que se realiza una transferencia o un registro de asistencia en una empresa. Este tipo de contenido se lo genera en tiempo real y este se adapta según las acciones del usuario. Algunos ejemplos de

contenido dinámico son las noticias en tiempo real, actualizaciones de redes sociales, entre otras. Para crear el backend, se utilizan scripts y bases de datos, y la información se actualiza en el servidor antes de entregarse al usuario por medio del frontend.

#### **Contenido Dinámico en Arquitectura Serverless:**

- Se forma por medio que se lo solicita y este puede variar según la entrada del usuario o datos externos.
- Las funciones serverless son ideales para manejar contenido dinámico esto gracias a que pueden responder a eventos específicos, procesar solicitudes HTTP e interactuar con bases de datos.
- Las funciones serverless se utilizan en función de cuando se activan sus eventos y se ejecutan de manera más rápida, esto permite una escalabilidad automática y un pago basado en la demanda evitan un mal uso de sus recursos.
- AWS Lambda, Azure Functions o Google Cloud Functions, son servicios serverless los cuales están utilizados para crear y ejecutar estas funciones dinámicas o de backend.

Para poder trabajar con el backend desde la arquitectura serverless es importante identificar los servicios que permitirán implementar la lógica del código y el usuario pueda interactuar con la aplicación web, se utilizan los siguientes servicios fundamentales.

##### ➤ **Authenticate**

Esta parte del backend es fundamental ya que en este servicio se realiza el control de acceso a través de la autenticación y autorización que pida la aplicación para poder ingresar al usuario que quiera utilizar la aplicación web y si este tiene los permisos necesarios o no.

##### ➤ **API Gateway**

Este servicio toma todas las llamadas API (Application Programming Interface) de los usuarios y realiza el enrutamiento de las solicitudes emitidas, esto mediante la composición y traducción de protocolos. Este servicio admite cargas de trabajo en aplicaciones serverless lo cual es importante al momento de trabajar en la creación de una aplicación web. Para empezar a trabajar, lo primero es invocar funciones sincrónicamente con una solicitud HTTP. Con esto la API Gateway espera una respuesta de la función y comunica el resultado al usuario al cual este llamando.

Hoy en día las aplicaciones web son sistemas que necesitan que estén trabajando con backend y frontend. Las interfaces de usuario son donde se muestra todo con lo que se pueda interactuar, el backend archiva esta información y la entrega. Ahora, para poder gestionar este enlace es donde entra una API.

#### ➤ **Functions (Faas)**

Este servicio es el motor de la Arquitectura Serverless, esto se debe a que se puede crear funciones autónomas para cualquier tipo de actividad que por lo general se liberan por eventos y se activan servicios de mensajes, recordatorios, seguridad, autenticación o procesos en los que se trabaja con otro tipo de herramientas de los proveedores de la nube.

#### ➤ **Database**

Las bases de datos se conocen que son almacenes de datos de todo tipo, ahora para trabajar con la arquitectura serverless se necesitan base de datos sin servidor las cuales están diseñados para cargas de trabajo más rápidas e impredecibles.

##### - **Database NoSQL**

Estas están diseñadas para aplicaciones modernas por su esquema flexible, estas bases de datos son ampliamente reconocidas por su funcionalidad, su facilidad para desarrollar y su rendimiento a escala.

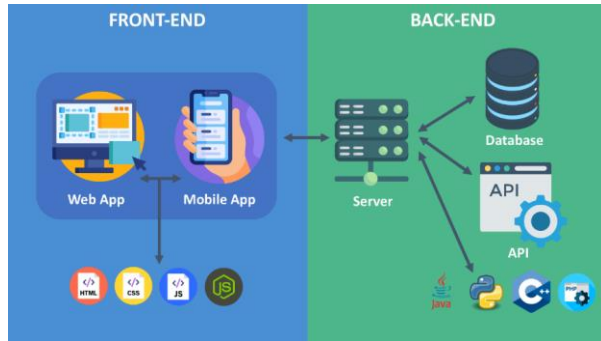


Figura 114 Backend vs Frontend. Fuente: (Basílio, 2022)

## CAPÍTULO III: METODOLOGÍA

---

### 3.1. Metodología de desarrollo de tesis

El presente proyecto de investigación se envuelve en un enfoque analítico y documental que permite el análisis e interpretación de datos provenientes de múltiples fuentes. Cuando se habla de un enfoque analítico quiero decir que se va a descomponer un problema y partes más pequeñas para poder comprender su estructura y funcionamiento, con esto vamos a trabajar en cómo se crea la estructura de la aplicación web y por qué el uso de esos servicios para que sea una arquitectura serverless. Por otro lado, está el enfoque documental gracias a este enfoque vamos a recopilar y presentar información de manera sistemática, con este enfoque podemos analizar y revisar documentos ya existentes que en este caso son los costos que nos presenta cada proveedor web, y a continuación crear nuestro propio documento con los datos ya recopilados y presentarlos.

El objetivo de esta investigación abarca tres partes claves los cuales están explicados a continuación. En la primera parte, se va a llevar a cabo un proceso de análisis y recolección de información de varios proveedores de servicios en la nube que ofrecen una experiencia de arquitecturas [Serverless](#), culminando en la selección de tres plataformas específicas las cuales son [AWS](#), [AZP](#) y [GCP](#). En la segunda sección, se presenta un esquema detallado para el desarrollo de una aplicación web utilizando servicios los cuales permiten ser a la aplicación web parte de la arquitectura de Serverless Computing. Por último, la tercera sección se dedica a la exposición de los costos asociados con la implementación de una aplicación web en cada una de las plataformas elegidas y de la región escogida que en este caso será Norteamérica ya que la mayoría de empresas de América trabajan bajo esta región.

**Comentado [XV1]:** Verificar que esto conste en el glosario

El objetivo principal de esta investigación radica en identificar una amplia gama de servicios proporcionados por los actuales proveedores de servicios en la nube, con el propósito de facilitar eficazmente el desarrollo de aplicaciones web basadas en la arquitectura [sServerless](#).

## CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN

---

### 4.1. Introducción

En la actualidad son cada vez más empresas las cuales empiezan a trabajar con la nube, aproximadamente este mercado está en crecimiento exactamente se calcula que para el 2028 el crecimiento de este mercado sea del 24%, como se puede ver en la figura 2. Gracias a esto las empresas han optado por empezar a trabajar con arquitectura serverless y la mayoría de los proveedores de nube pública ofrecen este servicio. En la tabla 1.1 se presenta 3 proveedores los cuales van a ser utilizados para este caso de estudio, cada uno cuenta con su respectiva plataforma cloud y su función serverless dentro de cada uno.

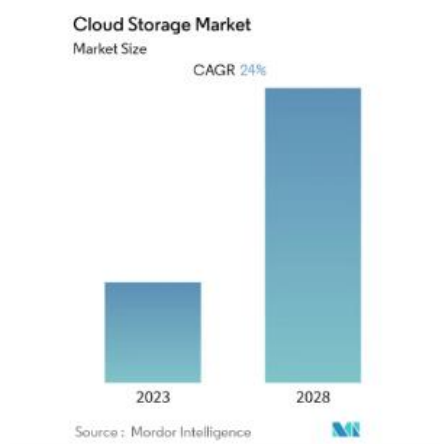


Figura 222. Cloud Storage Market. Fuente: (Mordor Intelligence, 2023)

La arquitectura serverless empezó a ser famoso en la industria desde el año 2010, sin embargo, en 2014 con el lanzamiento de AWS Lambda de Amazon Web Services (AWS) se popularizó esta tendencia. Más tarde, en el año 2016 Google e IBM con su Google Cloud Function e IBM Cloud Function se unieron al mercado de la arquitectura sin servidor, poco después en el año 2017 Microsoft lanzó Functions de Azure.

Esta arquitectura sin servidor o serverless es más rentable que asignar una cantidad exacta de servidores para un trabajo en específico la cual puede implicar tiempo de inactividad la cual es un costo que se lo está consumiendo de manera inadecuada. El trabajo en serverless es de la siguiente manera se necesita escribir código o funciones las cuales debe tener una tarea en específico, esto a su vez va a seleccionar la cantidad de recursos adecuados para que se ejecute el código y eso se enviará al servicio serverless en la nube. Este servicio garantiza que el código tenga la cantidad requerida de memoria y no más de esta, también calcula los ciclos del CPU necesario para que se pueda ejecutar.

Cuando se habla de números según una de las firmas de investigación llamada Gartner, el mercado de infraestructura en la nube abarca ciento de millones de dólares en la industria, con AWS como su mayor exponente con 33%, seguido por Azure con 18% y en tercer lugar Google Cloud Platform con 9 %. Cuando se habla de la tecnología serverless es todo un nuevo mundo, el cual aún falta por explorar, por ende, en este proyecto de titulación se pretende identificar las características de cada una para la implementación de una aplicación web y lo que las diferencie del otro, ya que cada uno tiene, diferentes precios para la utilización de sus recursos, características, carga del código y sus servicios gratuitos.

Tabla 114 Proveedores Cloud y Plataformas Serverless

Proveedor Cloud	Plataforma Cloud	Serverless
Amazon	Amazon Web Services (AWS)	AWS Lambda
Microsoft	Azure Cloud Platform	Google Cloud Functions
Google	Google Cloud Platform (GCP)	Azure Functions

## **4.2. Amazon Web Services (AWS)**

Esta plataforma cloud es la más completa en todo el mundo, por esta razón es la más adoptada por las empresas a nivel global, esto se debe a que ofrece más de 200 servicios integrales de centros de datos por todo el mundo. Este servicio tiene millones de clientes, incluso organismos gubernamentales usan esta plataforma, esto para reducir sus costos de infraestructura y aumentar la agilidad con la se procesan los datos, también las organizaciones adoptan este servicio para poder innovar sus productos.

### **4.2.1. AWS Lambda**

Su lanzamiento fue en el 2014, este es un servicio de cómputo que ejecuta códigos en respuesta a los eventos en los que se lo esté utilizando. Cuando se desarrolla una aplicación web el código se escribe como funciones, cada función de Lambda puede ser utilizada bajo demanda o se lo utiliza dinámicamente en función de ciertos tipos de eventos los cuales estén permitidos. Algunos eventos son activados cuando en el código realizan algún cambio con los objetos que están trabajando con funciones o con servicios complementarios que se esté utilizando en la aplicación web.

El servicio de Lambda se encarga de administrar y provisionar recursos de la infraestructura subyacente, así se asegura que el código se implemente de manera correcta y sin ningún tipo de inconveniente. Lambda también se ocupa de la alta disponibilidad en las diferentes regiones y de la escalabilidad.

## **4.3. Azure Cloud Platform**

Esta plataforma cloud brinda una amplia gama de servicios esenciales en el campo de la informática en la nube, abarcando desde la creación y gestión de aplicaciones hasta la optimización de los costos operativos, la mejora de la agilidad empresarial y la elaboración de estrategias más eficaces. También proporciona acceso a un almacenamiento prácticamente ilimitado, al mismo tiempo que asume la responsabilidad de las actualizaciones de hardware,

aliviando así a los usuarios de esta carga. A pesar de su relativa novedad en el mercado, esta plataforma ha avanzado rápidamente, ampliando sus capacidades y opciones disponibles de manera significativa. Su constante evolución la convierte en una solución cada vez más completa y poderosa para satisfacer las necesidades de la computación en la nube.

#### **4.3.1. Azure Function**

Es un servicio bajo demanda el cual proporciona todos los recursos y la infraestructura actualizados que son necesarios ejecutar las aplicaciones. Tiene la misma lógica que las otras plataformas, se diseña el código y en este caso el Functions se encargará del resto. Azure Functions se lo utiliza para procesar flujo de datos, administrar mensajes, crear API web, entre otras.

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

Es un repertorio de la infraestructura interna de Google, en el cual se encuentran sus productos de software y hardware, esta colección posee servidores, máquinas virtuales, almacenamientos, los cuales son servicios muy comunes dentro de los proveedores de la nube. Esta empresa opera a escala mundial tendiendo así beneficios económicos y esto se refleja en el precio de sus productos, ya que posee una inmensa infraestructura física la cual permite a parte de costos más bajos un rendimiento mejorado a otros proveedores.

##### **4.4.1. Cloud Functions**

Se lanzó en febrero del 2016, ofrece muchas utilidades para la solución de problemas, su principal caso de uso es la creación de servicios que son establecidos por el usuario, pero son administrados por otras personas.

#### **4.5. Diseño de la aplicación web**

Para este presente proyecto de titulación se decidió trazar la aplicación web de tal manera que se pueda llevar a cabo un análisis comparativo más exacto utilizando servicios similares

ofrecidos por los proveedores de nube seleccionados. Se optó por un enfoque, en el cual los usuarios acceden a través de un navegador web para interactuar con el frontend. Posteriormente, se emplea un servicio de autenticación para gestionar a los usuarios, esto por mayor seguridad para los usuarios que estén utilizando la aplicación web, por otra parte, el núcleo de la aplicación también utiliza este tipo de servicio. El backend, se construye utilizando la tecnología de cómputo sin servidor (Serverless Computing) proporcionada por cada proveedor. Esta elección de Serverless Computing no solo permite organizar la lógica y el código de la aplicación de manera eficaz, sino que también facilita la unificación con otros servicios, lo que resulta en un rendimiento óptimo de la aplicación para compensar las necesidades de los usuarios.

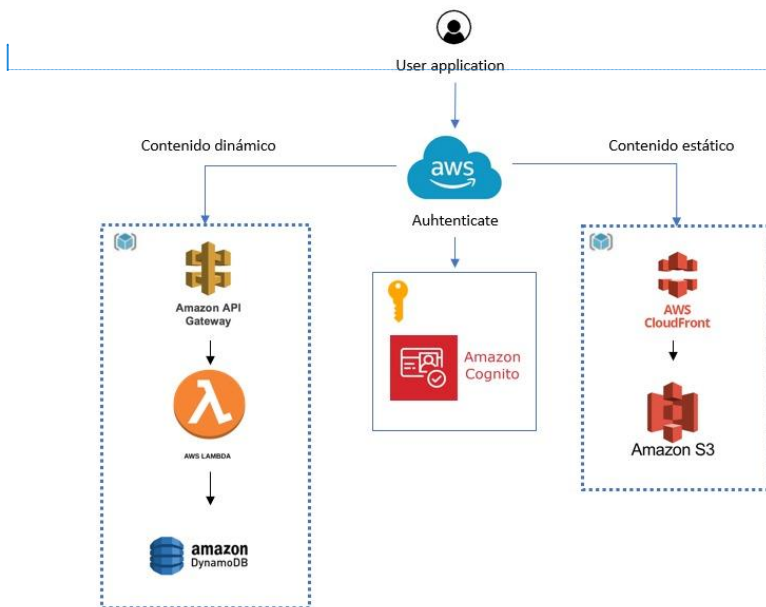
#### 4.5.1. Arquitectura de la Aplicación Web

##### 4.5.1.1. Serverless Computing en AWS

Tabla 222 Servicios de Amazon Web Services

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

Como se puede observar en la tabla 2, se detalla cada servicio al que pertenece cada componente que se usa en esta estructura de la aplicación web, la cual fue mencionada en el marco teórico en el apartado de Contenido dinámico y estático, con los servicios que se iba trabajar. A continuación, en la figura 3 se podrá ver de manera grafica la estructura de la aplicación web en este caso es Amazon Web Services y luego se procederá a describir la función de cada uno de sus servicios.



Comentado [XV2]: Indicar la fuente

Figura 333 Estructura la aplicación Web en AWS con arquitectura serverless

### Amazon Cloudfront

Amazon CloudFront es un servicio online diseñado para aligerar la entrega de contenido web estático, incluyendo archivos como .html, .css, .js y archivos de imágenes, a los usuarios de las páginas web. Este servicio trabaja mediante una extensa red de centros de datos que están distribuidos globalmente a lo largo de Norteamérica, Sudamérica Europa y Asia, se los conoce como ubicaciones de borde. Cuando un usuario solicita contenido que se encuentra distribuido a través de CloudFront, la solicitud se redirige automáticamente a la ubicación de borde que ofrezca la menor latencia o retraso en el tiempo, asegurando así que el contenido se entregue de manera eficiente y con un rendimiento óptimo.

En este caso pueden acontecer dos sucesos dentro del CloudFront, en primer lugar, si el contenido ya se encuentra dentro de la ubicación de borde con menor latencia, este se lo entregará de manera inmediata. Por otra parte, si el contenido no se encuentra en la ubicación

de borde, lo que hará el CloudFront es recuperarlo del origen que ya esté definido como la última versión del contenido, como por ejemplo puede ser desde un servidor HTTP o un bucket S3 de Amazon.

### **Amazon S3**

Las funciones Lambda de Amazon S3 tales como la carga de archivos, control de acceso, eliminación de archivos y eventos de transición de almacenamiento, se activan de manera automática cuando se crea, actualiza o elimina un objeto que se encuentre en el servicio. Amazon S3 se utiliza para alojar los activos estáticos de aplicaciones web y, de forma segura y se los puede distribuir a través de CloudFront. A este servicio se lo conoce, por su facilidad y flexibilidad en el almacenamiento de datos, y es una de las ofertas más populares y llamativas de AWS. Además, es confiable y ofrece un rendimiento excepcional a gran escala. Un aspecto clave de Amazon S3 es su modelo de precios, ya que solo paga por el almacenamiento que utiliza, eliminando la necesidad de proveer recursos con anticipación.

Amazon S3 es una plataforma destinada a almacenar archivos estáticos de sitios web, como archivos HTML, CSS, JavaScript e imágenes. Para utilizarlo, simplemente se debe crear un depósito, el cual funcione como un directorio para almacenar todos sus archivos estáticos de manera ordenada. Esto facilita la gestión y entrega de contenido estático de forma eficiente.

### **Amazon Cognito**

Es un servicio de autenticación y autorización proporcionado por Amazon Web Services (AWS) se lo utiliza por lo general en aplicaciones móviles y web para la gestión de la identidad de los usuarios. Lo que hace este servicio es recopilar los atributos del perfil del usuario en diferentes directorios los cuales son utilizados para la aplicación web, esto se configura de acuerdo con el acceso limitado de los recursos de AWS. Amazon Cognito cumple con las siguientes características:

- ✓ Registro e inicio de sesión de usuarios
- ✓ Control de acceso y autorización
- ✓ Integración con servicios de AWS
- ✓ Sincronización de datos

Amazon Cognito simplifica la gestión de la identidad de usuarios, gracias a esto permite a los desarrolladores centrarse en la lógica de la aplicación mientras garantiza la seguridad y la escalabilidad de la autenticación y autorización de los clientes.

### **Amazon API Gateway**

Este servicio tiene como función proteger, monitorear, crear y publicar APIs en cualquier tipo de escala. Se encarga de todas las actividades necesarias para aceptar y procesar cientos de miles de llamadas de API, y el tema de costos es de acuerdo con el uso que se lo da.

Algunos beneficios de API Gateway es su integración con diferentes tipos de backends, como lo es Lambda y en este ejemplo se lo utiliza. Es seguro ya que trabaja con Amazon Cognito y tiene una gran escalabilidad.

### **Amazon Lambda**

Es un servicio serverless, se basa en eventos, es la más popular dentro de lo que es AWS. El servicio de Lambda admite varios lenguajes de programación tales como: Python, C++, Node.js, Go y Java.

Entre las características más destacadas de AWS Lambda, se encuentran que proporciona una API para ejecutar la función, tiene un modelo de precios que se basa en funcional uso, se integra de forma perfecta con el entorno de AWS, el código tiene una memoria asignada, entre otros.

### **Amazon DynamoDB**

DynamoDB es una base de datos NoSQL la cual se destaca como una de las opciones favoritas para aplicaciones que operan con cantidades masivas de datos y demandan una latencia precisa, incluso en conjuntos de datos que superan los 100 TB. Gracias a esto, lo convierte en un recurso confiable para aplicaciones las cuales requieren un acceso veloz a grandes volúmenes de información.

Además, este servicio de base a datos, se integra de manera fluida con aplicaciones sin servidor que aprovechan por medio de AWS Lambda. AWS Lambda se combina de forma óptima con DynamoDB para construir aplicaciones serverless altamente eficientes. Gracias a su capacidad de acceder a DynamoDB por medio de una API HTTP y gestionar la autenticación y autorización mediante roles de IAM, esta se convierte en la elección perfecta para desarrollar aplicaciones sin servidor de forma segura y rápida. Además, DynamoDB resulta perfecto para conjuntos de datos con patrones de acceso simples y bien definidos, los cuales se los utiliza para generar y ofrecer recomendaciones a los usuarios, esto gracias a su estructura de clave-valor.

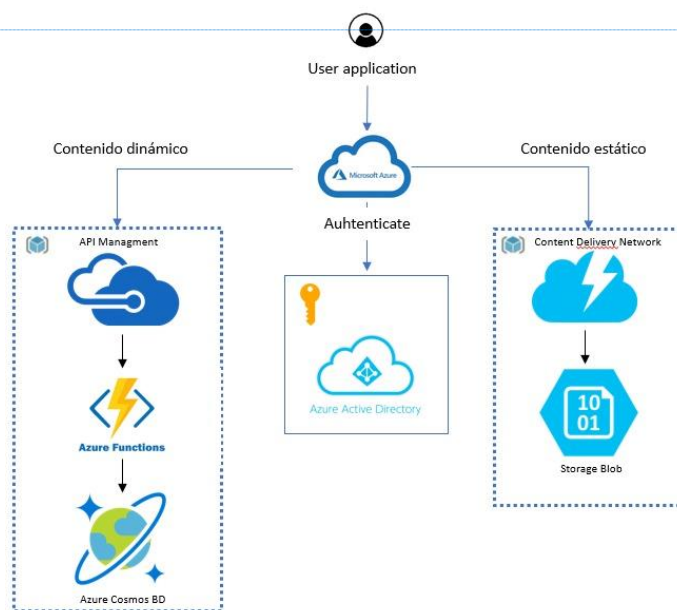
Las funciones de AWS Lambda se activan cuando se realizan actualizaciones en las tablas de datos de DynamoDB, como, por ejemplo, eliminación, editar una fila, inserción de filas, entre otras. DynamoDB cuenta con una API pequeña la cual permite el acceso simple de patrones de consulta de datos, más avanzados.

#### 4.5.1.2. Serverless Computing en Azure Cloud Platform

Tabla 333 Servicios de Azure Cloud Platform

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

Como se puede observar en la tabla 3, se detalla cada servicio al que pertenece cada componente que se usa en esta estructura de la aplicación web, esto ya se especificó anteriormente con cuales se iba a trabajar. A continuación, en la figura 4 se podrá ver de manera grafica la estructura de la aplicación web en este caso es Azure Cloud Platform y luego se procederá a describir la función de cada uno de sus servicios.



Comentado [XV3]: Indicar la fuente

Figura 444 Estructura de la aplicación Web en ACP con arquitectura serverless

**Azure Blob Storage**

Es un servicio de almacenamiento de objetos de Microsoft Azure, el cual es escalable y duradero. Este se lo utiliza para almacenar datos no estructurados como archivos de imagen, binarios, documentos, de texto, entre otros tipos de datos. Este servicio optimiza los costos con almacenamiento para datos los cuales son a largo plazo y escale los recursos con flexibilidad y verticalmente para cargas de trabajo de aprendizaje automático y de alto rendimiento.

Ofrece tres tipos de cuentas de almacenamiento:

- ✓ Blob Storage: Perfecto para almacenar datos no estructurados, como archivos, multimedia, copias de seguridad y documentos.
- ✓ Data Lake Storage: Creado para procesamiento y análisis de Big Data.
- ✓ Managed Disk: Se lo utiliza para almacenar discos virtuales de máquinas virtuales de Microsoft Azure.

### **Azure Active Directory B2C**

Este servicio de Microsoft Azure brinda soluciones de identidad y acceso para aplicación y servicios alojados en la nube. Las iniciales "B2C" significan Business To Consumer, el cual hace alusión a que este servicio está creado para habilitar la administración de identidades y autenticación de usuarios externos, como consumidores, socios y clientes reduciendo así el tiempo de administración de contraseñas. Este servicio proporciona un inicio de sesión único acceso condicional para protegerse contra el 99.9% de los ciber ataques y una autenticación multifactor.

### **Azure API Management**

Un servicio de Microsoft Azure el cual permite a las empresas administrar, publicar, proteger, y analizar APIs de maneras más centralizada. Tiene mayor flexibilidad y control en el acceso a los servicios y datos de una entidad. Cuando se importan ciertas APIs o se importan recursos de Azure, lo que hace Azure API Management es configurar el backend de la API de manera automática.

### **Azure Functions**

Es un servicio informático en la nube desarrollado por Microsoft, que permite a los desarrolladores ejecutar, crear y desplegar funciones de manera serverless. Gracias a este servicio se puede escribir pequeñas partes del código que se ejecutan en respuesta a solicitudes HTTP, mensajes en una cola, cambios en una base de datos, entre otros.

Azure Functions, tiene la capacidad de escalar según sea necesario, esto quiere decir que, si la carga es alta, pues este servicio se adaptará y escalará al mismo nivel de la carga, que en este caso es alta. Entre sus características más importantes tenemos dos la primera es que se puede probar el código directamente desde la interfaz sin necesidad de utilizar un entorno de desarrollo integrado (IDE), en segundo lugar, es que su modelo de costos va a la par del tiempo que se ejecuta el código. Entre los lenguajes de programación que admite Azure Functions, están: JavaScript, Java, Python, TypeScript, Batch, Bash y C#.

### Azure Cosmos DB

Es una base de datos NoSQL en la nube de Microsoft Azure, fue diseñado explícitamente para abordar los desafíos de latencia y escalabilidad que se presentan en aplicaciones modernas. Esta base de datos está respaldada por su seguridad a nivel empresarial y la disponibilidad respaldada por SLA. Hablando del desarrollo de aplicaciones este es más eficaz, productivo y rápido esto gracias a la distribución de datos en múltiples regiones a nivel global, aparte de tener un SDK para idiomas populares y un API de código abierto.

Este servicio de Azure trabaja muy bien con aplicaciones serverless, ya que responde con las necesidades de las aplicaciones a la par que la demanda, haciéndolo de forma automática y completamente segura.

#### 4.5.1.3. Serverless Computing en Google Cloud Platform

Tabla 444 Servicios de Google Cloud Platform

Proveedor	Servicio	Amazon Web Service
GCP	Content Delivery Network	Cloud CDN
GCP	Cloud Storage	Cloud Storage
GCP	Authenticate	Identity Platform
GCP	API Gateway	API Gateway
GCP	Function (FaaS)	Gloud Functions
GCP	Database	Datastore

Como se puede visualizar en la tabla 4, se detalla cada servicio al que pertenece cada componente que se usa en esta estructura de la aplicación web, esto ya se especificó anteriormente con cuales se iba a trabajar. A continuación, en la figura 5 se podrá ver de manera grafica la estructura de la aplicación web en este caso es Google Cloud Platform y luego se procederá a describir la función de cada uno de sus servicios.

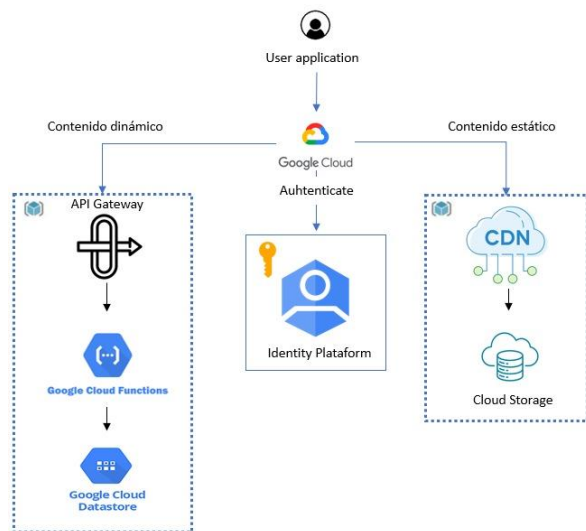


Figura 555 Estructura la aplicación Web en GCP con arquitectura

## Cloud CDN

El servicio de CDN se encarga de almacenar el contenido de sitios web en servidores ubicados de manera estratégica a nivel global. Este conjunto de servidores trabaja con un servicio de equilibrio de carga HTTP(s) para que así el contenido a los usuarios sea de manera más eficiente.

Cuando se realiza una solicitud en la plataforma de Google, este se dirige a un servidor en una ubicación que se encuentre geográficamente cerca al usuario que realizó la solicitud. Cloud CDN almacena copias en la memoria caché de ese contenido en estas ubicaciones, lo que representa que, si el usuario solicita nuevamente ese contenido, se le proporcionará más

rápido, por el motivo que esta información se encuentra en un servidor cercano en lugar de tener que viajar a través de distancias largas con grandes volúmenes de datos.

### **Cloud Storage**

Es un servicio de Google Cloud el cual se encarga del almacenamiento en la nube, este hace alusión a la práctica de almacenar datos digitalmente en servidores remotos a los que se tiene acceso a través de Internet. Estos en vez guardar información en dispositivos de almacenamiento físicos, los usuarios pueden almacenar y acceder a sus datos a través de servicios en la nube.

### **Identity Platform**

Este servicio brinda seguridad y autenticación para el registro y el inicio de sesión de los usuarios de manera rápida y personalizada. Con Identity Platform se vuelve una tarea fácil el agregar servicios de autenticación personalizada a aplicaciones móviles y web y así concentrarse en crear las funciones para la aplicación y utilizar otros servicios adicionales los cuales pueden estar ligados a nuestro proyecto.

### **API Gateway**

Este servicio de Google Cloud facilita el trabajo de poder administrar todas las llamadas API al backend sin necesidad de un servidor, Como un gran dato para este servicio, que gracias a estas llamadas API, no solo se ha visto una mejoría en la resolución de problemas, sino que también, según Google se ha visto una mejora de 15 veces la administración de las API, mediante el uso de plataformas de la nube únicas.

### **Cloud Functions**

Cloud Functions es un servicio de GPC de informática serverless la cual permite a los desarrolladores elaborar funciones en la nube en contestación a eventos específicos sin la

necesidad de encargarse la infraestructura subyacente. Este está diseñado para facilitar la creación y ejecución de pequeñas partes del código en respuesta a eventos en GCP y servicios de terceros. Una de sus características más importantes resalta el pago por uso y su integración con otros servicios de Google Cloud. Este servicio, admite múltiples lenguajes de programación como: Node.js, Java, Go, .NET, Ruby y Python.

### **Cloud Datastore**

Cloud Datastore un servicio de Google de base de datos NoSQL. Este servicio tiene un almacenamiento de datos altamente escalable y serverless para aplicaciones web y móviles. Al ser un base de datos NoSQL, significa que no utiliza un modelo relacional tradicional, sino que, constituye los datos de manera que sea altamente flexible y escalable, admitiendo también grandes cantidades de volúmenes de datos.

## CAPÍTULO V: MODELO DE COSTOS

---

### 5.1. Calculadora de servicios AWS, ACP y GCP

AWS, Azure Cloud Platform y Google Cloud Platform, cuentan con una calculadora para que se pueda tener una idea de los costos de los servicios y estos se pueden configurar a las exigencias de la empresa, gracias a esta herramienta es más accesible poder obtener un precio final más detallado de acuerdo con la necesidad de los diferentes servicios que requiera el cliente. En la tabla 5 se encuentran los vínculos de acceso a las calculadoras de cada uno de los proveedores de la nube. Hay que tener en cuenta que se debe conocer los servicios que se brinda y se necesitan, para no realizar el cálculo incorrecto.

Tabla 555 Calculadora de servicios AWS, ACP y GCP

Plataforma en la nube	Calculadora
AWS	<a href="https://calculator.aws/#/">https://calculator.aws/#/</a>
Azure Cloud Platform	<a href="https://cloud.google.com/products/calculator/">https://cloud.google.com/products/calculator/</a>
Google Cloud Platform	<a href="https://azure.microsoft.com/es-mx/pricing/calculator/">https://azure.microsoft.com/es-mx/pricing/calculator/</a>

### 5.2. Amazon Web Services

Amazon Web Services se ha ajustado con el tiempo a las nuevas reglas y estándares que los consumidores van exigiendo, entre estos los tipos de licencias que ofrecen, soporte, confidencialidad, condiciones de uso, entre otras. Gracias a la infraestructura serverless y la nube que usa este proveedor permite consumir los servicios con características más convenientes y aumenta la escalabilidad, elasticidad flexibilidad y disponibilidad, ya que, con acceso a internet, desde cualquier parte del mundo se puede acceder. Posee una interfaz muy vistosa y fácil de consumir para el cliente.

#### Parte gratuita

Amazon AWS, ofrece 3 modelos gratuitos, a continuación, se detallará como es cada uno.

- ✓ **Free Trials o Prueba Gratuita:** Esta oferta de corto plazo se inicia a partir del primer día en el que se activa cualquier servicio.
- ✓ **12 months free o 12 meses de uso gratuito:** Esta oferta dura 12 meses después del primer día que se realizó el registro inicial en AWS.
- ✓ **Always Free o Gratis para siempre:** Esta oferta no caduca y está disponible para todos los usuarios registrados en AWS.

A continuación, se detalla las ofertas y los precios para cada uno de los servicios que se van a utilizar para la aplicación web.

### 5.2.1. Amazon S3

Oferta de 12 meses gratis en este servicio incluye:

- 100 GB de transferencia de datos por cada mes.
- 2000 solicitudes PUT, POST y COPY.
- 20.000 solicitudes GET.
- 5 GB de almacenamiento

Con formato: Sangría: Izquierda: 0.45", Sangría francesa: 0.25"

A continuación, en la tabla 6 se realizó un pequeño ejemplo de un análisis de costos una vez se hayan finalizado los 12 meses gratis, que Amazon tiene.

Tabla 666 Costo mensual de Amazon S3

Datos de Ejemplo	Costo de Amazon	Costo total calculado
8 GB	\$0.023 por cada GB al mes	\$0.18
50.000 solicitudes POST	\$0.005 por cada 1.000 solicitudes	\$0.25
50.000 solicitudes GET	\$0.004 por cada 1.000 solicitudes	\$0.02
	<b>Costo por mes</b>	<b>\$0.45</b>

Para poder realizar el cálculo final de las solicitudes se realiza la siguiente fórmula:

$$\text{Costo total} = \left( \frac{\text{Número de solicitudes}}{1,000} \right) \times \text{Costo por 1,000 solicitudes}$$

**Comentado [XV5]:** De donde salió la fórmula. Indicar fuente

Figura 666 Calculo de solicitudes S3

### 5.1.2. Amazon Cognito

Para este servicio se utiliza la oferta Gratis para siempre ya que no se vence de manera automática al finalizar la oferta de 12 meses gratis, esta incluye:

- 1'000.000 de operaciones sincronizadas por mes.
- 50.000 MAU (Monthly Active Users) al mes.
- 10 GB de almacenamiento en la nube.

Pricing Tier (MAUs)	Price per MAU
50,001-100,000 (after the 50,000 free tier)	\$0.0055
Next 900,000	\$0.0046
Next 9,000,000	\$0.00325
Greater than 10,000,000	\$0.0025

Figura 777. Costos MAU Fuente (AWS, 2023)

En la figura 7 se muestran los cotos del MAU, para cada región el precio es diferente en este caso se utiliza la región de US East (Ohio), que es la que se utiliza la mayoría de las empresas ubicadas América al usar este servicio.

En la tabla 7 se puede apreciar un pequeño ejemplo con una carga de trabajo no tan pesada para que pueda escalar 58000 usuarios.

Tabla 777 Ejemplo mensual Cognito

Datos	Costo	Costo total calculado
52.000 MAU	\$0.0055 por cada MAU luego de los 50.000 MAU gratis	\$11.00
	<b>Costo por mes</b>	<b>\$11</b>

### 5.1.3. Amazon API Gateway

Oferta de 12 meses gratis en este servicio incluye:

- 1 M mensajes y 750.000 minutos online.
- 1 M de llamadas API recibidas por cada mes.

Cada solicitud a la API posee un tamaño de 512 KB de datos, si este valor es sobrepasado se tiene que realizar el cálculo respectivo para determinar el número de solicitudes que se tendrá como respuesta.

Para poder presentar un ejemplo de costo, se realizará el siguiente análisis, se va a utilizar una solicitud de 1512 KB, que aproximadamente son 3 solicitudes a la API, en la tabla 8 se presentara el costo mensual de la API Gateway.

Para realizar este ejemplo se debe tener en cuenta que el precio por millón de AWS es de \$1.00 esto quiere decir que el precio por solicitud es de \$0.000001, ahora a nuestras 3 solicitudes se les multiplica por 1'000.000 multiplicador de unidades, dándonos como resultado: 3'000.000 de solicitudes facturables.

Tabla 888 Costo mensual API Gateway

Datos	Costo	Costo total calculado
3'000.000	\$0.000001 por solicitud	\$3.00
	<b>Costo por mes</b>	<b>\$3</b>

### 5.1.4. AWS Lambda

En AWS Lambda se utiliza la Oferta de Gratis por siempre porque al ser un servicio el cual proporciona ejecución de funciones serverless, contiene costos agrupados con el tiempo de ejecución y el gasto de recursos, en este servicio, este incluye:

- 3,2 millones segundos de tiempo computacional al mes
- 1'000.000 solicitudes gratuitas al mes.

Se debe tener en cuenta que cuando el número de solicitudes sobre pasa el límite de la oferta gratis el costo va a variar dependiendo de la duración de cada solicitud y la memoria asignada. Cuando esto sucede se realiza un nuevo cobro de \$0.20 por 1'000.000 de solicitudes, teniendo en cuenta el costo de la duración el cual es \$0.0000166667/GB-seg.

Se realizará un cálculo el cual se considere que se tiene 20'000.000 solicitudes al mes, memoria de 5 GB y la duración de 30 ms.

Primero se realiza la conversión a segundos, el cual se multiplica las solicitudes por la duración y por el factor de conversión que es 0.001.

$$20'000.000 \times 30 \text{ ms} \times 0.001 \text{ ms} = 600.000 \text{ s}$$

Para sacar el valor del cómputo total el cual es la cantidad total de los recursos de cómputo utilizados ya por un servicio, por un tiempo en específicos, es se lo calcula multiplicando la cantidad de memoria por los segundos.

$$5 \text{ GB} \times 600.000 \text{ s} = 3'000.000 \text{ GB-seg}$$

Se le resta lo del nivel gratuito.

$$3'000.000 \text{ GB-seg} - 1'400.000 \text{ GB-seg} = 1'600.000 \text{ GB-seg}$$

Para sacar las solicitudes total facturables se resta las solicitudes menos las solicitudes que viene ya en el nivel gratuito.

$$20'000.000 - 1'000.000 = 19'000.000 \text{ solicitudes facturables}$$

En la tabla 9 se presentará el cálculo total mensual de AWS Lambda.

Tabla 999 Costo mensual de AWS Lambda

Datos	Costo	Costo total calculado
1'600.000 GB-seg	\$0.0000166667 por cada GB-seg	\$26.66
19'000.000 solicitudes	\$0.0000002 por solicitud	\$3.80
	<b>Costo por mes</b>	<b>\$30.46</b>

### 5.1.5. Amazon DynamoDB

En este servicio se utiliza la oferta Gratis para siempre ya que, para trabajar con bases de datos, se requiere guardar los datos de manera permanente y no por un periodo de tiempo, este incluye:

- 25 WCU (Write Capacity Units)
- 25 RCU (Read Capacity Units)
- 25 GB de almacenamiento

El costo por GB una vez se utilice la capa gratis es de \$0.25.

Suponiendo que se requiere 30 GB adicionales, el cálculo respectivo se mostrará en la Tabla 10.

Tabla 101010 Costo mensual Amazon DynamoDB

Datos	Costo	Costo total calculado
30 GB de almacenamiento	\$0.25 por cada GB	\$7.50
	<b>Costo por mes</b>	<b>\$7.50</b>

### 5.1.6. Amazon Cloudfront

Oferta de 12 meses gratis en este servicio incluye:

- 2'000.000 invocaciones de funciones.
- 10'000.000 solicitudes HTTPS o HTTP.
- 1 TB de transferencia de datos.

Si la aplicación usa menos cantidad de transferencia de datos de lo que ofrece la capa gratuita que es 1 TB y menos de 20'000.000 solicitudes HTTPS en total, esta oferta cubrirá solicitudes DTO (Data Transfer Object) y HTTPS de forma gratuita.

En la figura 8 se detalla el costo por mes de la transferencia de datos por regiones.

Regional Data Transfer Out to Internet (per GB)

Per Month	United States, Mexico, and Canada	Europe and Israel	South Africa, Kenya, and Middle East	South America	Japan	Australia and New Zealand	Hong Kong, Indonesia, Philippines, Singapore, South Korea, Taiwan, Thailand, Malaysia, and Vietnam	India
First 10TB	\$0.085	\$0.085	\$0.110	\$0.110	\$0.114	\$0.114	\$0.120	\$0.109
Next 40TB	\$0.080	\$0.080	\$0.105	\$0.105	\$0.089	\$0.098	\$0.100	\$0.085
Next 100TB	\$0.060	\$0.060	\$0.090	\$0.090	\$0.086	\$0.094	\$0.095	\$0.082
Next 350TB	\$0.040	\$0.040	\$0.080	\$0.080	\$0.084	\$0.092	\$0.090	\$0.080
Next 524TB	\$0.030	\$0.030	\$0.060	\$0.060	\$0.080	\$0.090	\$0.080	\$0.078
Next 4PB	\$0.025	\$0.025	\$0.050	\$0.050	\$0.070	\$0.085	\$0.070	\$0.075
Over 5PB	\$0.020	\$0.020	\$0.040	\$0.040	\$0.060	\$0.080	\$0.060	\$0.072

Customers willing to make minimum traffic commits of typically 10 TB/month or higher are eligible for discounted pricing. [Contact us](#)

Figura 888 Costo AWS Cloudfront por región Fuente (AWS, 2023)

Se va a realizar un pequeño análisis, con una pequeña carga de producción suponiendo que se tiene 150 GB de datos con salida al internet desde un caché de Cloudfront cada mes y se va a realizar 1'500.000 de solicitudes HTTPS. En este caso se trabaja con funciones de servicio por solicitud de 10'000.000 las cuales nos permiten remitir las mismas hacia Lambda, se debe tener en cuenta que las primeras 2'000.000 no tienen costo adicional luego se cobrará \$0.1 por cada 1M de solicitudes, entonces la cantidad total de funciones por solicitud es de 8'000.000.

En la tabla 11, se presentará el cálculo mensual de Amazon CloudFront.

Tabla 11111 Costo mensual Amazon Cloudfront

Datos	Costo	Costo total calculado
150 GB transferencia de datos	1 TB gratis	\$0.00
1'500.000 solicitudes HTTPS	10'000.000 gratis	\$0.00
Transferencia datos a internet	\$0.22 costo	\$0.22
Transferencia datos a origen	\$0.25 costo	\$0.25
8'000.000 funciones por solicitud	8'000.000 x \$0.0000001 por cada solicitud	\$0.08
	<b>Costo por mes</b>	<b>\$0.55</b>

Tabla con formato

## 5.2. Microsoft Azure

Microsoft Azure ha avanzado exponencialmente a nivel tecnológico esto ha hecho que cree su propia plataforma en la nube llamado AzureCloud Platform gracias a eso tiene nuevos tipos de licencias, soporte, confidencialidad, condiciones de uso que le hacen competir dentro del mercador de la arquitectura serverless. Gracias a esta nueva arquitectura y su tecnología en la nube sus servicios se han vuelto más convenientes para el usuario y aumenta la escalabilidad, elasticidad flexibilidad y disponibilidad.

El protocolo que sigue Azure Cloud Platform es que una vez estes registrado en una cuenta se puede acceder a un crédito de \$200 para utilizarlo en 30 días, durante este periodo se puede hacer uso de más de 40 servicios gratuitos, no existen cargos cuando finaliza la prueba gratuita.

### Parte gratuita

- **12 meses gratis:** Ofrece algunos servicios durante 12 meses.
- **Gratis:** Más de 55 servicios adicionales que se pueden utilizar.

#### 5.2.1. Azure Blob Storage

La oferta de 12 meses gratis solo incluye:

- 5 GB de almacenamiento

En el informe detallado no precisa que se tengan solicitudes GET y PUT de manera gratuita, sin embargo, se conoce que por cada 10000 solicitudes GET el costo es de \$0.065 y por cada 10000 solicitudes PUT el costo es de \$0.005 por cada 10000.

Para el siguiente ejemplo se van a usar los mismos datos que se utilizó anteriormente, con el fin de luego desarrollar un análisis comparativo. Esto quiere decir que se van a utilizar 8 GB,

50.000 solicitudes GET y 50.000 solicitudes PUT. En la tabla 12 se mostrará el cálculo mensual.

Tabla ~~1212+2~~ Costo mensual Cloud Storage

Datos	Costo	Costo total calculado
8 GB	\$0.026 por cada GB al mes	\$0.20
50.000 solicitudes POST	\$0.065 por cada 10000 solicitudes	\$0.32
50.000 solicitudes GET	\$0.005 por cada 10000 solicitudes	\$0.03
	<b>Costo por mes</b>	<b>\$0.55</b>

### 5.2.2. Azure Active Directory B2C

Oferta de 12 meses incluye:

- 50.000 MAU gratis por mes

Para el siguiente ejemplo se van a usar los mismos datos que se utilizó anteriormente, con el fin de luego desarrollar un análisis comparativo. Esto quiere decir que se van a utilizar 52000 MAU.

En la tabla 13 se presenta el cálculo por mes para el ejemplo planteado.

Tabla ~~1313+3~~ Costo mensual de Azure Active Directory B2C

Datos	Costo	Costo total calculado
52.000 MAU	\$0.00325 x MAU	\$6.50
	<b>Costo por mes</b>	<b>\$6.50</b>

### 5.2.3. Azure API Management

No existe en este caso oferta gratis, sin embargo, se factura por ejecución, en este está permitido de 0 a 1'000.000 millón de solicitudes por suscripción. Se le incluye el costo de \$0.042 por 10.000 solicitudes.

Para el siguiente ejemplo se van a usar los mismos datos que se utilizó anteriormente, con el fin de luego desarrollar un análisis comparativo. Esto quiere decir que se van a utilizar 52000 MAU.

En la tabla 14 se presenta el cálculo por mes para el ejemplo planteado.

Tabla ~~14144~~ Costo mensual API Gateway

Datos	Costo	Costo total calculado
3'000.000	\$0.0000042 por solicitud	\$12.60
	<b>Costo por mes</b>	<b>\$12.60</b>

#### 5.2.4. Azure Functions

Oferta Siempre gratis incluye:

- 400.000 GB-seg por un mes suscrito.
- 1'000.000 de solicitudes

Una vez supere los 400.000 el costo pasa a ser \$0.000016 GB-seg cada uno, esta cantidad permite el 1'000.000 de ejecuciones y si este valor se sobrepasa, por el millón de ejecuciones el costo sería \$0.20.

En este servicio solo se paga por los recursos computaciones en ejecución.

Se realizará el mismo cálculo y situación ya planteado anterior mente, el cual se considera que se tiene 20'000.000 solicitudes al mes, memoria de 5 GB y la duración de 30 ms.

Primero se realiza la conversión a segundos, el cual se multiplica las solicitudes por la duración y por el factor de conversión que es 0.001.

$$20'000.000 \times 30 \text{ ms} \times 0.001 \text{ ms} = 400.000 \text{ s}$$

Para sacar el valor del cómputo total el cual es la cantidad total de los recursos de cómputo utilizados ya por un servicio, por un tiempo en específicos, este se lo calcula multiplicando la cantidad de memoria por los segundos.

$$5 \text{ GB} \times 400.000 \text{ s} = 2'000.000 \text{ GB-seg}$$

Se le resta lo del nivel gratuito.

$$2'000.000 \text{ GB-seg} - 400.000 \text{ GB-seg} = 1'600.000 \text{ GB-seg}$$

Para sacar las solicitudes total facturables se resta las solicitudes menos las solicitudes que viene ya en el nivel gratuito.

$$10'000.000 - 1'000.000 = 19'000.000 \text{ solicitudes facturables}$$

En la tabla 15 se presentará el cálculo total mensual de Azure Functions.

Tabla 15 Costo mensual de Azure Functions

Datos	Costo	Costo total calculado
1'600.000 GB-seg	\$0.0000166667 por cada GB-seg	\$26.66
19'000.000 solicitudes	\$0.0000002 por solicitud	\$3.80
	<b>Costo por mes</b>	<b>\$30.46</b>

### 5.2.5. Azure Cosmos DB

En la oferta Siempre Gratis, incluye:

- 25 GB de almacenamiento
- 1000 unidades de solicitud por segundo

Para el siguiente ejemplo se van a usar los mismos datos que se utilizó anteriormente, con el fin de luego desarrollar un análisis comparativo. Esto quiere decir que se van a utilizar 30 GB de almacenamiento. En la tabla 16 se presenta el cálculo mensual de Azure Cosmos DB.

Tabla 1616+6 Costo mensual Azure Cosmos DB

Datos	Costo	Costo total calculado
30 GB de almacenamiento	\$0.15 por cada GB	\$4.50
	<b>Costo por mes</b>	<b>\$4.50</b>

### 5.2.6. Azure Content Delivery Network

No existe oferta gratis en este servicio, sin embargo, permite optimizar el acceso a recursos del frontend, esto gracias a su red de entrega de contenido. Existen 5 zonas de distribución para la transferencia de datos esto se dividen en:

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

En la figura 6 se detalla el costo de la transferencia de datos.

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
Siguientes 500 TB (500-1,000 TB)/mes	\$0,028 por GB	\$0,075 por GB	\$0,13 por GB	\$0,088 por GB	<a href="#">Contacto</a>
Siguientes 4,000 TB (1,000-5,000 TB)/mes	\$0,023 por GB	\$0,065 por GB	\$0,121 por GB	\$0,084 por GB	<a href="#">Contacto</a>
Más de 5,000 TB/mes	<a href="#">Contacto</a>	<a href="#">Contacto</a>	<a href="#">Contacto</a>	<a href="#">Contacto</a>	<a href="#">Contacto</a>

Figura 999 Tabla de Precios Azure CDN Fuente (Azure,2023)

Se realizará un pequeño análisis con una pequeña carga de 150 GB de datos y 1'500.000 solicitudes HTTPS, en la tabla 17 se realiza el cálculo mensual de Azure CDN.

Tabla ~~171747~~ Costo mensual Azure CDN

Datos	Costo	Costo total calculado
150 GB transferencia de datos	\$0.081 por GB en la Zona 1	\$12.15
1'500.000 solicitudes HTTPS	\$0.60 por 1'000.000	\$0.90
	<b>Costo por mes</b>	<b>\$13.05</b>

### 5.3. Google Cloud Platform

Google Cloud Platform es uno de los líderes en la actualidad en este ámbito cuenta con nuevas licencias, soporte a nivel mundial, confidencialidad en sus datos y tiene nuevas condiciones de uso que le hacen competir incluso liderar dentro del mercador de la arquitectura serverless.

Una vez se tenga una cuenta creada, se tiene acceso a todos lo servicios y producto, a su vez se da un crédito de \$300 con una duración de 90 días y se puede hacer uso de mas de 20 servicios. Una vez este periodo finalice, no existen cargos automáticos.

#### 5.3.1. Cloud Storage

Al no poseer ofertas gratis, este servicio no cuenta con ofertas, si no, ya directamente tiene una tarifa fija para el valor de solicitudes y GB.

Para el siguiente ejemplo se van a usar los mismos datos que se utilizó anteriormente, con el fin de luego desarrollar un análisis comparativo. Esto quiere decir que se van a utilizar 8 GB, 50.000 solicitudes GET y 50.000 solicitudes PUT. En la tabla 18 se mostrará el cálculo mensual.

Tabla ~~181848~~ Costo mensual de Cloud Storage

Datos	Costo	Costo total calculado
8 GB	\$0.026 por cada GB al mes	\$0.20
50.000 solicitudes POST	\$0.000005 por cada solicitud	\$0.25
50.000 solicitudes GET	\$0.0000004 por cada solicitud	\$0.02

	<b>Costo por mes</b>	<b>\$0.47</b>
--	----------------------	---------------

### 5.3.2. Identity Platform

Los costos que se verán en la figura 7 se dividen en diferentes niveles gracias al método de autenticación utilizado.

UAM	Precio por UAM (USD)
Entre 0 y 49.999	0
Entre 50.000 y 99.999	0,0055
Entre 100.000 y 999.999	0,0046
Entre 1.000.000 y 9.999.999	0,0032
Más de 10.000.000	0,0025

Figura 101049 Costos de MAU en Identity Platform

Para el siguiente ejemplo se van a usar los mismos datos que se utilizó anteriormente, con el fin de luego desarrollar un análisis comparativo. Esto quiere decir que se van a utilizar 52000 MAU. En la tabla 19 se mostrará el cálculo mensual.

Tabla 191949 Costo mensual de Identity Platform

Datos	Costo	Costo total calculado
52.000 MAU	\$0.0055 x MAU	\$11.00
	<b>Costo por mes</b>	<b>\$11.00</b>

### 5.3.3. API Gateway

En la figura 3, se puede observar las llamadas API por mes y su cuenta de facturación, incluso se puede ver, que, hasta dos millones de llamadas, son gratuitas.

Llamadas a la API por mes y cuenta de facturación	Coste por cada millón de llamadas a la API
Hasta dos millones	0,00 USD
De dos millones a mil millones	3,00 USD
Más de mil millones	1,50 USD

Figura 111144 Costo por cada millón de llamadas API Fuente (Google Cloud, 2023)

Para el siguiente ejemplo se van a usar los mismos datos que se utilizó anteriormente, con el fin de luego desarrollar un análisis comparativo. Esto quiere decir que se van a utilizar 3'000.000 de solicitudes. En la tabla 20 se mostrará el cálculo mensual.

Tabla 202020 Costo mensual API Gateway

Datos	Costo	Costo total calculado
3'000.000	\$3.00	\$3.00
	<b>Costo por mes</b>	<b>\$3.00</b>

#### 5.3.4. Cloud Functions

Este servicio si tiene una parte gratuita por mes, la cual incluye:

- 2'000.000 de invocaciones.
- 400.000 GB-seg
- 200.000 GHz-seg
- 5 redes

En Google Cloud Functions se trabaja con invocaciones, estas se refieren al proceso de activación y ejecución de una función en respuesta a cualquier evento, viene a ser lo mismo que solicitudes en otras plataformas. Ahora, este servicio tiene un costo de \$0.0000004 por invocación, externo a la oferta gratuita que este posee. El costo por invocación luego de la oferta gratis pasa a ser \$0.0000004 por cada invocación y por último luego de utilizar los 400.000 Gb-seg pasa a tener un costo de 0.0000025.

Se realizará el siguiente ejemplo de cómo operaría con una carga pequeña de trabajo si se tiene 20'000.000 de invocaciones y 2'000.000 de GB-seg. Este proceso también se le debe tener en cuenta la ejecución de eventos, para así tener información más precisa acerca de los GB-seg por invocación utilizados y de igual forma los GHz utilizados. En la tabla 21 se mostrará el cálculo por mes.

Tabla 2124 Costo mensual Google Cloud Functions

Datos	Costo	Costo total calculado
1'600.000 GB-seg	\$0.0000025 por cada GB-seg	\$4.00
19'000.000 invocaciones	\$0.0000004 por invocación	\$7.60
<b>Costo por mes</b>		<b>\$11.60</b>

Los valores de la tabla 21, ya están restados con sus partes gratuitas.

### 5.3.5. Datastore

Este servicio posee una capa gratuita la cual incluye:

- 1 GB de almacén de datos
- 50000 datos de lectura
- 20000 datos de escritura
- 20000 datos de eliminación
- 50000 operaciones pequeñas

Una vez se hayan excedido de la oferta gratuita, los precios serían los siguientes:

- Por cada GB al mes \$0.18.
- Por cada 100000 de lectura el precio sería \$0.06
- Por cada 100000 de escritura \$0.18
- Por cada 100000 de eliminación \$0.02
- Operaciones pequeñas se mantiene en estado gratuito

Para el siguiente ejemplo se van a usar los mismos datos que se utilizó anteriormente, con el fin de luego desarrollar un análisis comparativo. Esto quiere decir que se van a utilizar 30 GB de almacenamiento. En la tabla 22 se presenta el cálculo mensual de Azure Cosmos DB.

Tabla 222222 Costo mensual DataStore

Datos	Costo	Costo total calculado
30 GB de almacenamiento	\$0.18 por cada GB	\$5.40
	<b>Costo por mes</b>	<b>\$5.40</b>

### 5.3.6. Cloud CDN

En la figura 9 se observa el costo por salida del caché.

## Salida de caché

Los cargos por salida de caché se corresponden con las respuestas que se han servido desde cachés de Cloud CDN y varían según el destino y el uso mensual. El uso mensual se calcula por proyecto y por destino. El destino es una zona geográfica que depende de la dirección IP del cliente.

Destino	Precio (por GiB) por uso mensual			
	< 10 TiB	10-150 TiB	150-500 TiB	> 500 TiB
Asia-Pacífico (Hong Kong incluido)	0,09 USD	0,06 USD	0,05 USD	<a href="#">Contactar con nosotros</a>
China <sup>1</sup>	0,20 USD	0,17 USD	0,16 USD	<a href="#">Contactar con nosotros</a>
Europa	0,08 USD	0,055 USD	0,03 USD	<a href="#">Contactar con nosotros</a>
Norteamérica (Hawái incluido)	0,08 USD	0,055 USD	0,03 USD	<a href="#">Contactar con nosotros</a>
Oceania <sup>2</sup>	0,11 USD	0,09 USD	0,08 USD	<a href="#">Contactar con nosotros</a>
Sudamérica	0,09 USD	0,06 USD	0,05 USD	<a href="#">Contactar con nosotros</a>
Todos los demás destinos (incluidos México, Centroamérica y Oriente Medio)	0,09 USD	0,06 USD	0,05 USD	<a href="#">Contactar con nosotros</a>

Figura 121242 Tabla de costo Cloud CDN Fuente (Google Cloud, 2023)

Para el siguiente ejemplo se van a usar los mismos datos que se utilizó anteriormente, con el fin de luego desarrollar un análisis comparativo. Esto quiere decir que se van a realizar 1'500.000 solicitudes HTTPS y 150 GB de datos de salida a internet desde el caché de CDN. En la tabla 20 se mostrará el cálculo mensual.

Tabla 232323 Costo mensual CDN

Datos	Costo	Costo total calculado
150 GB transferencia de datos	\$0.08 por GB en Norteamérica	\$12.00
1'500.000 solicitudes HTTPS	\$0.75 por 1'000.000	\$1.13
	<b>Costo por mes</b>	<b>\$13.13</b>

#### 5.4. Resultados

A continuación, se van a presentar los resultados obtenidos de la investigación realizada anteriormente, esta parte va a mostrar los costos que genera el crear una aplicación web bajo la arquitectura serverless con los servicios ya mencionados anteriormente, utilizando los proveedores de nube Amazon, Microsoft y Google. Se presentará también un pequeño análisis de los datos obtenidos gracias a las tablas y cálculo mensual que se realizó en cada servicio. Como bien se sabe este trabajo de titulación tiene un enfoque de la creación de una aplicación web por medio de la tecnología serverless o sin servidor, esto quiere decir que gracias a esta tecnología se puede ejecutar pequeñas partes del código llamadas funciones por un periodo de tiempo determinado, estas son el núcleo de esta arquitectura ya que por medio de eventos pueden ser integrados en otros servicios de esta misma infraestructura.

Se debe tener en cuenta que los servicios que se usó para el análisis son especialmente para la creación de una aplicación web por medio de la arquitectura serverless, a continuación, se va a detallar el servicio que se utilizó para cada uno de los componentes de la aplicación de las diferentes plataformas en la nube.

- ✓ **Cloud Storage:** Amazon S3, Azure Blob Storage y Cloud Storage.
- ✓ **Authenticate:** Amazon Cognito, Azure Active Directory B2C y Identity Platform.
- ✓ **API Gateway:** Amazon API Gateway, Azure API Management y API Gateway.
- ✓ **FaaS:** AWS Lambda, Azure Functions y Cloud Functions.
- ✓ **Database:** Amazon DynamoDB, Azure Cosmos DB y DataStore.
- ✓ **Content Delivery Network:** Amazon Cloudfront, Azure CDN y Cloud CDN.

#### 5.4.1. Costos finales por mes

Los costos que se tomaron en cuenta para este análisis comparativo fueron tomados en cuenta para condiciones y situaciones similares, por un periodo mensual y por una pequeña carga laboral que se enfocaría la aplicación web. Cabe recalcar que estos costos, no tienen gastos en servidores e infraestructura, ya que todo está puesto en el internet.

El costo por mes total que se presenta en la tabla 24 son los resultados ya obtenidos del cálculo del costo ya calculado por mes de las tablas 6,12 y 18, este es de Cloud Storage osea es almacenamiento del código albergado en el frontend de las tres plataformas.

Tabla 242424 Costo por mes Cloud Storage de AWS, ACP y GCP

Plataforma	Servicio	Costo por mes
<b>Amazon Web Service</b>	Amazon S3	\$0.45
<b>Azure Cloud Platform</b>	Azure Blob Storage	\$0.55
<b>Google Cloud Platform</b>	Cloud Storage	\$0.47

En la tabla 25 se presentará el costo mensual total el cual ya fue calculado en las tablas 7,13 y 19 esta tabla es acerca de la Autenticación de los usuarios que ingresan a la pagina web, y trabajan con backend o frontend, de las tres plataformas.

Tabla 252525 Costo por mes Authenticate de AWS, ACP y GCP

Plataforma	Servicio	Costo por mes
<b>Amazon Web Service</b>	Amazon Cognito	\$11.00
<b>Azure Cloud Platform</b>	Azure Active Directory B2C	\$6.50

<b>Google Cloud Platform</b>	Identity Platform	\$11.00
------------------------------	-------------------	---------

El costo por mes que tiene el uso de API Gateway para el trabajo con el backend de cada una de las aplicaciones se presenta en la tabla 26 los datos que se presentan son los resultados ya obtenidos del cálculo del costo ya calculado por mes de las tablas 8,14 y 20.

Tabla ~~262626~~ Costo por mes API Gateway de AWS, ACP Y GCP

Plataforma	Servicio	Costo por mes
<b>Amazon Web Service</b>	Amazon API Gateway	\$3.00
<b>Azure Cloud Platform</b>	Azure API Management	\$12.60
<b>Google Cloud Platform</b>	API Gateway	\$3.00

El costo por mes de la solución serverless para el backend de la aplicación de las tres plataformas, se muestra en la tabla 27 los datos que se presentan en la tabla son los resultados ya obtenidos del cálculo del costo ya calculado por mes de las tablas 9,15 y 21.

Tabla ~~272727~~ Costo por mes Functions de AWS, ACP y GCP

Plataforma	Servicio	Costo por mes
<b>Amazon Web Service</b>	Amazon Lambda	\$30.46
<b>Azure Cloud Platform</b>	Azure Functions	\$30.46
<b>Google Cloud Platform</b>	Cloud Functions	\$11.60

El costo por mes del uso de la Base de Datos para almacenamiento de las plataformas se encuentra en la tabla 28 los datos que se presentan en la tabla son los resultados ya obtenidos del cálculo del costo ya calculado por mes de las tablas 10,16 y 22.

Tabla ~~282828~~ Costo por mes de DB de AWS, ACP y GCP

Plataforma	Servicio	Costo por mes
<b>Amazon Web Service</b>	Amazon DynamoDB	\$7.50
<b>Azure Cloud Platform</b>	Azure CosmosDB	\$4.50
<b>Google Cloud Platform</b>	DataStore	\$5.40

El costo por mes por el uso de CDN para el frontend de la aplicación para cada una de las plataformas se presentan en la tabla 29 los datos que se presentan en la tabla son los resultados ya obtenidos del cálculo del costo ya calculado por mes de las tablas 11,17 y 23.

Tabla ~~292929~~ Costo por mes CDN de AWS, ACP y GCP

Plataforma	Servicio	Costo por mes
Amazon Web Service	Amazon S3	\$0.55
Azure Cloud Platform	Azure Blob Storage	\$13.05
Google Cloud Platform	Cloud Storage	\$13.13

#### 5.4.2. Análisis Comparativo

Como se observa en la tabla 30, tenemos la suma del costo de todos los servicios que se necesitan para generar una aplicación web bajo la tecnología serverless de las distintas plataformas en la nube.

Tabla ~~303030~~ Costo de todos los servicios por mes

Plataforma	Cloud Storage	Autenticación	API Gateway	Funci ons	Base de Datos	CDN	Total por mes
Amazon Web Service	\$0.45	\$11	\$3	\$30.46	\$7.50	\$0.55	\$52.96
Azure Cloud Platform	\$0.55	\$6.50	\$12.60	\$30.46	\$4.50	\$13.05	\$67.66
Google Cloud Platform	\$0.47	\$11	\$3	\$11.60	\$5.40	\$13.13	\$44.60

Como se observa en la tabla 30, el costo total entre cada una de las plataformas no existe una gran diferencia, y la favorita por el precio es Google Cloud Functions, seguido de Amazon AWS y las más cara Azure Cloud Platform, sin embargo, hay que tener en cuenta, los servicios y ofertas gratuitas que tanto como AWS, GCP y ACP ofrecen a sus usuarios, esto es necesario de tomar en cuenta ya que, al momento de realizar el cálculo para determinar el

costo final, es necesario restar la oferta gratis de cada uno de los servicios para tener el valor exacto y realista de cuánto costaría.

Para este proyecto presente proyecto de titulación basándonos netamente en costos la opción mas viable para la creación de una aplicación web bajo la arquitectura serverless sería Google Cloud Platform, ya que tiene el valor mensual mas bajo a pagar con \$44.60, esta plataforma va teniendo mayor reconocimiento a nivel global, lo que le hace mejorar sus servicios y servicio al cliente cada vez más. La diferencia de costos con AWS no es mucha, se puede decir que igual es una opción viable utilizar esta plataforma para la creación de una página web, a parte de que AWS tiene mucha más información en internet y lo vuelve más accesible y fácil de entender para cualquier persona, también cuenta con un servicio al cliente mucho mejor que GCP y ACP. Hablando del costo de Azure, podemos ver que aquí si existe una gran diferencia que AWS y GCP, esto se debe a la tecnología que usa ya que es de Microsoft y esta es más conocida y entendida a nivel global, sin embargo, el soporte que tiene esta pagina es un poco lenta y sin respuestas exactas.

Para la creación de la estructura de la pagina web se utilizaron servicios que son muy importantes dentro de la arquitectura serverless, entre estos están API Gateway que es el conector entre el backend y frontend gracias que invoca funciones a través de eventos por todos los servicios; por otra parte, tenemos el Cloud Storage que es el almacenamiento del frontend y su precio no varía entre las plataformas. Estos dos servicios son esenciales para la tecnología serverless y las tres plataformas tienen buenas ofertas y excelentes propuestas, aparte de que su costo no varía en entre los tres a excepción de Azure en API Gateway que su costo es 4 veces más a diferencia de las otras dos plataformas, estos los hace a las tres plataformas sean excelentes opciones, para la creación de una aplicación web. Se realizará un análisis comparativo más profundo de los cuatro servicios faltantes porque aquí sus

características, ofertas y precio si varían, los servicios faltantes serían: Autenticación, CDN, Functions y Base de Datos.

### **Autenticación**

Este servicio conformado por Amazon Cognito de AWS, Azure Active Directory B2C de Azure y Identity Platform de Google. Si comparamos el costo por mes de cada uno, como se puede ver en la tabla 30 el costo de Azure es menor al de las otras dos plataformas, ganando por este ámbito Azure, cabe recalcar que el costo de los tres se lo calcula en función a los MAU. Por otro lado, los tres servicios tienen control de los usuarios, pero el de Azure es mucho mejor ya que lo hacen como característica indispensable dentro del servicio a diferencia de las otras plataformas, en este sentido, Azure Active Directory B2C es la mejor opción cuando se trata de autenticación de usuarios.

### **Functions**

Este servicio conformado por AWS Lambda, Azure Functions y Cloud Functions. Functions es el servicio el cual es el núcleo de la aplicación web, entonces es el más importante al momento de hablar de arquitectura serverless, por este lado cuando hablamos de costos como se puede visualizar en la tabla 30 Cloud Functions tiene el menor costo frente a las otras dos aplicaciones y tiene 1 millón más de solicitudes mensuales gratuitas que AWS y ACP, esto lo hace la favorita por tema de precio a GCP. Sin embargo, no se debe tener en cuenta solo el costo para tomar una decisión final, ya que también se tiene que tomar en cuenta los lenguajes de programación que soportan los tres servicios, en este caso Azure admite mayores lenguajes, pero AWS admite lenguajes que hoy en día están siendo mayormente utilizados para la creación de páginas web. Teniendo en cuenta todos los detalles como precio, soporte, oferta, costo por cada millón de solicitudes y lenguajes de programación que admiten la mejor opción sería AWS.

## Base de datos

Este servicio conformado por Amazon DynamoDB, Azure CosmosDB y Datastore. Uno de los servicios mas importantes ya que en este se almacena todos los datos que son ingresados a la página web, para la arquitectura serverless se trabaja con base [de-en](#) datos NoSQL, ya que tiene un escalado automático, mejor rendimiento, lectura de mayor número de datos y una estructura flexible. Cuando hablamos de costos, como se puede observar en la tabla 30 CosmosDB es la más barata y DynamoDB es la más cara, por ende, cuando hablamos de precio por almacenamiento Amazon no se vuelve la mejor solución. Sin embargo, Amazon DynamoDB es la más conocida a nivel global, por ende, cuenta con mayor soporte, portal de ayuda e interfaz amigable con el usuario. Por la ubicación geográfica de sus servidores, Amazon DynamoDB se vuelve la favorita frente a las otras opciones, aunque su costo sea mayor que las otras dos plataformas.

## Content Delivery Network

Este servicio conformado por Amazon Cloudfront, Azure Content Delivery Network y Cloud CDN. En este servicio, como se puede observar en la tabla 30, Amazon Cloudfront es un claro ganador, frente a las otras dos plataformas, este costo es gracias a que Cloudfront posee una oferta gratuita la cual permite que los costos se abaraten al momento de realizar el calculo mensual, por otra parte, Cloud CDN y Azure Content Delivery Network, no poseen ofertas gratuitas. Por otra parte, Amazon posee un servicio al cliente especialmente para este servicio, y las otras dos plataformas no lo tiene. Este servicio es importante en la infraestructura [s](#)Serverless ya que mejora el acceso de forma rápido a los recursos, por ende, la mejor opción es Amazon Cloudfront.

## CONCLUSIONES Y RECOMENDACIONES

---

### Conclusiones

- Amazon Web Services (AWS) es la plataforma de computación en la nube más popular, gracias a su larga trayectoria, sus beneficios, ofertas y servicios gratuitos, esto lo hace la favorita a mi criterio frente a las otras dos plataformas. Muchas empresas han logrado posicionarse gracias a la arquitectura Serverless Computing, un ejemplo de esto es Netflix por su gran cantidad de datos que tiene. Aunque la idea de implementar esta arquitectura es muy atractiva, para las empresas, se debe tener en cuenta que los costos son en función del consumo, en caso de exceder los recursos gratuitos, los costos no difieren mucho entre las tres principales plataformas de computación en la nube. Por lo antes mencionado, es necesario realizar un estudio a fondo de todas las dependencias necesarias para crear aplicaciones, el plazo de tiempo que permitirá obtener un costo reducido y analizar el peor de los escenarios.
- Las plataformas tres plataformas ofrecen una amplia gama de servicios para crear aplicaciones web y otros servicios los cuales no fueron mencionados en este trabajo de titulación, pero serian de gran ayuda para otro tipo de proyectos como automatización de tareas, chatbots, entre otros. Sin embargo, los modelos de suscripción, el crédito inicial, los beneficios y los costes mensuales de cada plataforma siempre van a estar diferidos. Por lo tanto, es importante analizar los servicios y recursos necesarios antes de elegir una plataforma.

**Comentado [XV6]:** Las conclusiones deben tener concordancia con los objetivos planteados, es decir tu trabajo debe dar respuesta a esos objetivos. La conclusión es el resultado del objetivo. Si tienes 4 objetivos de conclusiones a cada uno de ellos. Si tienes 4 entonces 4 conclusiones o puedes poner más por cada objetivo

- Los Serverless Computing son un cambio de enfoque que libera a los desarrolladores de la necesidad de administrar servidores. En cambio, los desarrolladores solo deben preocuparse por el pago de los recursos que utilizan. Esto puede simplificar el desarrollo de aplicaciones web, pero también requiere que los desarrolladores conozcan un poco más acerca de esta arquitectura serverless, que comprendan las plataformas y servicios que ofrecen los proveedores y sepan cómo utilizarlos para la creación de este tipo de proyectos.

### Recomendaciones

- Cuando uno conoce los servicios que se quieren utilizar, es recomendable usar las calculadoras de precios que ofrecen las plataformas [Serverless](#) ya que son de suma utilidad para estimar el coste de los servicios. Algunos servicios no están integrados en estas calculadoras y hay que tener en cuenta que los cálculos pueden no tener en cuenta los recursos gratuitos. Por lo tanto, es importante consultar la información de precios de las páginas oficiales de las plataformas junto con las calculadoras. Esto ayudará a establecer límites para aprovechar las ofertas de cada servicio.
- Se recomienda utilizar términos en inglés para redes de servidores, servicios, protocolos, lenguajes de programación y tecnología en general, ya que la mayoría de la información relevante está disponible en este idioma. Esto gracias a que Serverless Computing es una tendencia tecnológica relativamente nueva y forma parte de una nueva generación de productos y servicios, que van a ser utilizados en grandes empresas de desarrollo, software, bankin, streaming, entre otras.
- Para este trabajo de titulación se **desarrolló una aplicación web pequeña** bajo la arquitectura serverless. Se recomienda crear un modelo de aplicación web basado en los mismos servicios para empresas u organizaciones mundiales, teniendo en

**Comentado [XV7]:** En donde está? Cuál es? Si no la haces referencia dentro del trabajo es mejor que no la pongas esta ecomendación o la modifiques.

cuenta las consideraciones presentadas en este proyecto. Es importante analizar cuál es la mejor solución para cada caso.

## BIBLIOGRFÍA

---

AWS. (2023). Obtenido de AWS Lambda Pricing:

[https://aws.amazon.com/lambda/pricing/?nc2=type\\_a](https://aws.amazon.com/lambda/pricing/?nc2=type_a)

AWS. (2023). *Amazon Cloud Front*. Obtenido de

[https://aws.amazon.com/cloudfront/?nc2=type\\_a](https://aws.amazon.com/cloudfront/?nc2=type_a)

AWS. (2023). *Amazon Cloudfront Pricing*. Obtenido de

[https://aws.amazon.com/cloudfront/pricing/?loc=ft#AWS\\_Free\\_Usage\\_Tier](https://aws.amazon.com/cloudfront/pricing/?loc=ft#AWS_Free_Usage_Tier)

AWS. (2023). *Amazon Web Service* . Obtenido de [https://aws.amazon.com/free/?all-free-](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=categor)

[tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=\\*all&awsf.Free%20Tier%20Categories=categor](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=categor)  
[ies%23serverless&aws.page-all-free-tier=1](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=categor)

AWS. (2023). *AWS Pricing*. Obtenido de [https://aws.amazon.com/pricing/?aws-products-](https://aws.amazon.com/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all)

[pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=\\*all&awsf.tech-category=\\*all](https://aws.amazon.com/pricing/?aws-products-pricing.sort-by=item.additionalFields.productNameLowercase&aws-products-pricing.sort-order=asc&awsf.Free%20Tier%20Type=*all&awsf.tech-category=*all)

Azure. (2023). *Azure Content Delivery Network*. Obtenido de [https://azure.microsoft.com/en-](https://azure.microsoft.com/en-us/products/cdn/)

[us/products/cdn/](https://azure.microsoft.com/en-us/products/cdn/)

Azure. (2023). *CDN Pricing*. Obtenido de <https://azure.microsoft.com/es-es/pricing/details/cdn/>

Azure. (2023). *Precios de Azure*. Obtenido de [https://azure.microsoft.com/es-](https://azure.microsoft.com/es-mx/pricing/?&ef_id=_k_Cj0KCQiA6vaqBhCbARIsACF9M6miyAU-)

[mx/pricing/?&ef\\_id=\\_k\\_Cj0KCQiA6vaqBhCbARIsACF9M6miyAU-](https://azure.microsoft.com/es-mx/pricing/?&ef_id=_k_Cj0KCQiA6vaqBhCbARIsACF9M6miyAU-)  
[xcT0\\_YHJob9CIngDNhlpuRkfG7zoXCeo5hbgJURtkTG8OwaAmVoEALw\\_wcB\\_k\\_&OCID=AIDcmmvcssag76\\_SEM\\_\\_k\\_Cj0KCQiA6vaqBhCbARIsACF9M6miyAU-](https://azure.microsoft.com/es-mx/pricing/?&ef_id=_k_Cj0KCQiA6vaqBhCbARIsACF9M6miyAU-)  
[xcT0\\_YHJob9CIngDNhlpuRkfG7zoXCeo5hbgJURtkTG8Ow](https://azure.microsoft.com/es-mx/pricing/?&ef_id=_k_Cj0KCQiA6vaqBhCbARIsACF9M6miyAU-)

Bautista, I. (30 de 3 de 2021). *Backend y Frontend, ¿Qué es y cómo funcionan en la programación?* Obtenido de <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>

Carrero, L. (1 de febrero de 2023). *Stackscale*. Obtenido de <https://www.stackscale.com/es/blog/modelos-de-servicio-cloud/>

Flores, F. (11 de octubre de 2021). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-serverless-ventajas-y-servicios/>

Google Cloud. (2023). *¿Qué es IaaS?* Obtenido de <https://cloud.google.com/learn/what-is-iaas?hl=es>

Google Cloud. (2023). *¿Qué es un proveedor de servicios en la nube?* Obtenido de <https://cloud.google.com/learn/what-is-a-cloud-service-provider?hl=es>

Google Cloud. (2023). *API Gateway*. Obtenido de <https://cloud.google.com/api-gateway?hl=es>

Google Cloud. (2023). *Cloud CDN*. Obtenido de <https://cloud.google.com/cdn/pricing?hl=es>

Google Cloud. (2023). *Cloud CDN Pricing*. Obtenido de <https://cloud.google.com/cdn?hl=es#pricing-module>

Google Cloud. (2023). *Cloud Storage Pricing*. Obtenido de <https://cloud.google.com/storage?hl=es#pricing-module>

Google Cloud. (2023). *Identity Platform*. Obtenido de <https://cloud.google.com/identity-platform?hl=es>

Google Cloud. (s.f). *¿Qué es cloud computing?* Obtenido de <https://cloud.google.com/learn/what-is-cloud-computing?hl=es>

Izquierdo, D. P. (2021). *Universidad complutense madrid*. Obtenido de

<https://docta.ucm.es/rest/api/core/bitstreams/f9c2f7d3-7f87-45d8-9e4f-5609bf61cc79/content>

Kudryashov, R. (19 de julio de 2021). *DZone*. Obtenido de [https://dzone.com/articles/not-only-](https://dzone.com/articles/not-only-spring-boot-a-review-of-alternatives?edition=582294&utm_source=Zone%20Newsletter&utm_medium=email&utm_campaign=microservices%202020-02-26)

[spring-boot-a-review-of-](https://dzone.com/articles/not-only-spring-boot-a-review-of-alternatives?edition=582294&utm_source=Zone%20Newsletter&utm_medium=email&utm_campaign=microservices%202020-02-26)

[alternatives?edition=582294&utm\\_source=Zone%20Newsletter&utm\\_medium=email&utm\\_campaign=microservices%202020-02-26](https://dzone.com/articles/not-only-spring-boot-a-review-of-alternatives?edition=582294&utm_source=Zone%20Newsletter&utm_medium=email&utm_campaign=microservices%202020-02-26)

Mordor Intelligence. (2023). *ANÁLISIS DEL TAMAÑO Y LA PARTICIPACIÓN DEL MERCADO*

*DE ALMACENAMIENTO EN LA NUBE TENDENCIAS Y PRONÓSTICOS DE CRECIMIENTO (2023 - 2028)*. Obtenido de

<https://www.mordorintelligence.com/es/industry-reports/cloud-storage-market>

Red Hat. (04 de 07 de 2023). *¿Qué es la infraestructura de TI?* Obtenido de

<https://www.redhat.com/es/topics/cloud-computing/what-is-it-infrastructure>

## GLOSARIO DE TÉRMINOS

---

**AWS:** Amazon Web Services.

**GCP:** Google Cloud Platform.

**AZP:** Azure Cloud Platform.

**Solicitudes PUT:** Solicitud que envía datos al servidor para actualizar o crear.

**Solicitudes POST:** Solicitud que envía datos al servidor para crear.

**Solicitudes GET:** Solicitud que se utiliza para recuperar información.

**Solicitudes COPY:** Solicitud que se utiliza para la edición.

**TI:** Tecnología de la información.

**HTTP:** Hypertext Transfer Protocol.

**CSS:** Cascading Style Sheets.

**HTML:** Hypertext Markup Language.

**JS:** JavaScript.

**IAM:** Identity and Access Management.

**SLA:** Service Level Agreement.

**SDK:** Software Development Kit.

**CDN:** Content Delivery Network

**GB:** Gigabyte.

**TB:** Terabyte.

**GHZ:** Gigahertz.

**DB:** Database.

**Caché:** Es un tipo de memoria de alta velocidad y accesos que es utilizada para almacenar datos temporalmente.

Con formato: Español (Ecuador)

**ANEXOS**

---