



**FACULTAD DE INGENIERÍA Y ARQUITECTURA**

**ACTIVIDAD:**

AVANCE PROYECTO.

**CÁTEDRA:**

GESTIÓN DE SERVIDORES WEB.

**CATEDRÁTICO:**

ING. CARLOS ISAIAS RIOS LUNA.

**ESTUDIANTE:**

ALEX FRANCISCO LOVOS ARGUETA. (U20241471)

**CARRERA:**

INGENIERÍA EN DESARROLLO DE SOFTWARE.

**FECHA:**

MIÉRCOLES 10 DE DICIEMBRE DE 2025.

SAN MIGUEL, EL SALVADOR, CENTROAMÉRICA.

## Indice

<b>1. INTRODUCCION.....</b>	<b>3</b>
<b>2. OBJETIVOS.....</b>	<b>4</b>
2.1.    OBJETIVO GENERAL .....	4
2.2.    OBJETIVOS ESPECIFICOS .....	4
<b>3. ENUNCIADO DEL PROBLEMA.....</b>	<b>5</b>
<b>4. PROBLEMÁTICA DE LA EMPRESA.....</b>	<b>6</b>
<b>5. MARCO TEORICO: SERVICIOS WEB EN LA NUBE: MICROSOFT AZURE VS. AMAZON WEB SERVICES (AWS) .....</b>	<b>7</b>
5.1.    HISTORIA Y EVOLUCIÓN DE MICROSOFT AZURE Y AWS .....	7
5.2.    MODELOS DE SERVICIO EN LA NUBE: IAAS, PAAS, SAAS .....	8
5.3.    SERVICIOS WEB PRINCIPALES PARA DESPLIEGUE DE APLICACIONES, SITIOS WEB Y API .....	10
5.4.    COMPARACIÓN ENTRE AZURE Y AWS: ARQUITECTURA, DISPONIBILIDAD GLOBAL, FACILIDAD DE USO, PRECIOS, ESCALABILIDAD Y SEGURIDAD .....	14
5.4.1.    Arquitectura y disponibilidad global .....	14
5.4.2.    Facilidad de uso y curva de aprendizaje.....	15
5.4.3.    Precios y modelos de costo .....	16
5.4.4.    Escala bilitad y rendimiento.....	17
5.4.5.    Seguridad, cumplimiento y gestión de identidades.....	19
<b>6. REQUERIMIENTOS PARA LA IMPLEMENTACIÓN DEL PROYECTO.....</b>	<b>21</b>
6.1.    REQUIREMENTOS TÉCNICOS.....	21
6.1.1.    Servicios en la nube de Microsoft Azure.....	21
6.1.2.    Infraestructura y software necesarios (del lado del cliente).....	21
6.2.    REQUERIMIENTOS OPERATIVOS Y HUMANOS .....	22
6.3.    REQUISITOS GENERALES DEL PROYECTO .....	22
<b>7. COSTOS REALES DE IMPLEMENTACIÓN DE LA SOLUCIÓN EN LA NUBE POR SERVICIO .....</b>	<b>22</b>
<b>8. ESQUEMA DE SOLUCION .....</b>	<b>24</b>
<b>9. CONCLUSION .....</b>	<b>25</b>
<b>10. BIBLIOGRAFIA .....</b>	<b>26</b>

# 1.Introduccion

En la actualidad, las organizaciones dependen cada vez más de las soluciones tecnológicas que les permitan operar de manera eficiente, segura y accesible desde cualquier lugar. Las infraestructuras físicas tradicionales presentan muchas limitaciones muy importantes, ya sea los altos costos de mantenimiento, los riesgos de pérdida de información y la poca capacidad para poder escalar ante el crecimiento de la empresa. Frente a este escenario, la migración hacia las plataformas en la nube se ha convertido en una alternativa clave y estratégica para optimizar recursos y mejorar la continuidad del negocio.

La empresa TierraByte, dedicada a la comercialización de productos tecnológicos, enfrenta dicho desafío el cual esta relacionado con la administración de sus sistemas locales, por los cuales dificultan su expansión y reducen la disponibilidad de sus servicios internos. Por ello, este proyecto propone la evaluación e implementación de una arquitectura basada en servicios web en la nube de Microsoft Azure, con el fin de poder modernizar su infraestructura, fortalecer la seguridad de sus datos y facilitar el acceso a sus sistemas desde cualquier ubicación.

A lo largo del documento se presentará la problemática actual de la empresa, los fundamentos teóricos de las soluciones en la nube, los requerimientos técnicos, los costos estimados y el diseño de la propuesta de migración. Finalmente, se desarrollará un sitio web de prueba alojado en Azure para demostrar la viabilidad del proyecto y evidenciar los beneficios de adoptar una plataforma de servicios en la nube.

## **2. Objetivos**

### **2.1. Objetivo General**

Proponer una solución de migración en la nube para la empresa TierraByte a través de los servicios de Microsoft Azure, con el fin de optimizar su infraestructura tecnológica y mejorar la disponibilidad, seguridad y escalabilidad de sus sistemas.

### **2.2. Objetivos Especificos**

- Analizar la infraestructura actual de TierraByte para poder identificar las limitaciones tecnológicas y los servicios que requieren modernización.
- Describir los servicios de Microsoft Azure aplicables a la migración de los sistemas de información de la empresa.
- Determinar los requerimientos técnicos y costos reales necesarios para implementar la solución propuesta en la nube.
- Diseñar y demostrar una implementación práctica mediante la creación y despliegue de un sitio web de prueba alojado en Azure.

### 3. Enunciado del problema

En un entorno empresarial ideal, las organizaciones modernas deberían de contar con las infraestructuras tecnológicas capaces de poder garantizar una alta disponibilidad, la seguridad de la información, la escalabilidad y el acceso remoto para sus colaboradores. Las soluciones basadas en la nube permiten cumplir dichos estándares al poder ofrecer servicios confiables, automatizados y sobre todo con menores costos operativos en comparación con la infraestructura física tradicional.

Sin embargo, la realidad demuestra que muchas pequeñas y medianas empresas aún operan con servidores locales lo cual dichas empresas presentan limitaciones técnicas significativas. A nivel global, según datos de Gartner (2024), más del 35% de las empresas que dependen únicamente de infraestructura local experimentan caídas de servicio cada año, afectando la productividad y ocasionando muchas pérdidas económicas. En Latinoamérica, Microsoft (2023) reporta que más del 45% de las PYMES carecen de estrategias de respaldo de datos, lo que hace que incremente el riesgo de pérdida de información crítica. En El Salvador, la Superintendencia de Electricidad y Comunicaciones (SIGET) señala que los apagones y variaciones eléctricas llegan a afectar de manera recurrente a pequeñas empresas, poniendo así en riesgo sus sistemas locales.

En este contexto, la empresa TierraByte, ubicada en el departamento de San Miguel, enfrenta problemas derivados de su infraestructura física: su servidor local presenta riesgos de fallos eléctricos, no cuenta con sistemas automáticos de respaldo, no permite el acceso remoto para empleados y limita la expansión tecnológica. Esto afecta directamente la gestión de inventarios, ventas y atención al cliente, reduciendo así la eficiencia operativa y aumentando los costos de mantenimiento.

Dada esta situación, surge lo que sería la necesidad de analizar alternativas que permitan modernizar dicha infraestructura tecnológica de la empresa mediante el uso de servicios en la nube. Entre las opciones disponibles, Microsoft Azure destaca por su flexibilidad, seguridad y escalabilidad, características que podrían resolver las dificultades actuales de la organización.

En función de lo anterior, se plantea la siguiente pregunta de investigación:

**¿Cómo puede la migración de la infraestructura tecnológica de TierraByte hacia servicios en la nube de Microsoft Azure mejorar la disponibilidad, seguridad y eficiencia operativa de sus sistemas de información?**

Abordar esta problemática nos ayudara a identificar una solución real, tecnológica y viable que contribuya al crecimiento de la empresa, optimice recursos, mejore la

continuidad del negocio y brinde una mayor estabilidad operativa en comparación con la infraestructura local actual.

## **4. Problemática de la empresa**

La empresa TierraByte, dedicada a la comercialización de productos tecnológicos en el departamento de San Miguel, depende mucho de su sistema de inventario y ventas para poder operar diariamente. Sin embargo, actualmente enfrenta una serie de limitaciones la cual se derivan de su infraestructura tecnológica basada en un servidor local. Dicha infraestructura, aunque sea funcional, se ha vuelto insuficiente ante el crecimiento exponencial de la empresa y las exigencias que surgen en este entorno digital moderno.

En primer lugar, el servidor local no cuenta con mecanismos de alta disponibilidad, lo que significa que cualquier falla eléctrica que haya o problema de hardware puede llegar a detener completamente las operaciones. Esta vulnerabilidad afecta de manera directa la atención al cliente, ya que la empresa no puede procesar ventas ni consultar inventarios durante los dichos fallos. Esto se agrava debido a que TierraByte opera en una zona donde las variaciones eléctricas son bastantes frecuentes, lo cual incrementa el riesgo de pérdida o corrupción de datos.

Además, la empresa carece de un sistema de respaldos automáticos, por lo que la información crítica como los registros de ventas, datos de los clientes y movimientos de inventario se expone a pérdidas irreversibles. Aunque se realizan respaldos manuales de manera ocasional, estos dependen completamente de la disponibilidad del personal, lo que lo vuelve un proceso poco confiable.

Por otra parte, el sistema actual no permite acceso remoto, limitando así las capacidades operativas del personal en los horarios extendidos o situaciones en las que se requiere trabajar fuera de la tienda. Todo esto dificulta la toma de decisiones, la supervisión de inventarios en tiempo real y la atención a clientes que realizan consultas fuera del horario habitual.

Otro aspecto relevante es la falta de escalabilidad. A medida que TierraByte se ha ido ampliando el catálogo de productos y su volumen de clientes, el servidor local muestra que hay signos de saturación, disminuyendo así el rendimiento del sistema. Esto se traduce en tiempos de espera prolongados, procesos más lentos y una experiencia deficiente tanto para los empleados como para los clientes.

Finalmente, los costos de mantenimiento del servidor local representan una carga económica constante. La empresa necesita invertir en técnicos, reemplazo de componentes, protección eléctrica y mejoras periódicas. Sin embargo, todas estas inversiones no llegan a garantizar la estabilidad a largo plazo ni preparan a la empresa

para posibles expansiones, como lo que sería la apertura de nuevas sucursales o la implementación de ventas en línea.

En conjunto, estas limitaciones evidencian la gran necesidad de que TierraByte modernice su infraestructura tecnológica y migre hacia una alternativa más segura, escalable y sobre todo eficiente. La adopción de servicios en la nube, como los que ofrece Microsoft Azure, representa una gran oportunidad estratégica para poder llegar a superar estos desafíos y lograr asegurar la continuidad operativa de la empresa en un entorno digital que cada vez es más competitivo.

## **5.Marco Teorico: Servicios web en la nube: Microsoft Azure vs. Amazon Web Services (AWS)**

### **5.1. Historia y evolución de Microsoft Azure y AWS**

Amazon Web Services (AWS): AWS fue lanzado oficialmente por Amazon en 2006, marcando un hito en la computación en la nube a gran escala. Inicialmente ofreció servicios básicos como Amazon S3 (almacenamiento de objetos) y Amazon EC2 (computación elástica), reutilizando la infraestructura de Amazon para que otras empresas pudieran usarla. A partir de allí, AWS creció rápidamente ampliando su catálogo a más de 200 servicios en la actualidad, incluyendo bases de datos, análisis de datos, inteligencia artificial, Internet de las Cosas (IoT) y más. AWS ha sido pionero en áreas como la computación sin servidor con el lanzamiento de AWS Lambda en 2014, y servicios de aprendizaje automático como Amazon SageMaker. Hoy en día, AWS lidera el mercado global de infraestructura en la nube con aproximadamente un 33% de cuota de mercado (datos de 2022), atendiendo a millones de clientes desde startups hasta grandes corporaciones y agencias gubernamentales.

Microsoft Azure: Microsoft Azure fue lanzado en febrero de 2010 bajo el nombre de Windows Azure, cambiando a su nombre actual en 2014 para reflejar que abarcaba mucho más que solo el ecosistema Windows. Desde su lanzamiento inicial (cuando contaba con pocas ofertas competitivas), Azure ha crecido de forma significativa. La plataforma se integró profundamente con otros productos de Microsoft como Office 365 y Dynamics 365, y expandió sus servicios para cubrir computación, almacenamiento y capacidades de inteligencia artificial. Un momento clave fue en 2014, cuando Satya Nadella (CEO de Microsoft) impulsó una estrategia de “cloud-first”, haciendo de Azure una prioridad corporativa. Azure se destaca por su perfecta integración con el entorno

empresarial de Microsoft, aprovechando el legado de la compañía en software corporativo y sistemas operativos. Desde sus inicios, Azure ha evolucionado continuamente incorporando soluciones de vanguardia en inteligencia artificial, IoT y analítica de datos. Actualmente, Azure es el segundo proveedor de nube a nivel mundial con alrededor del 21% del mercado, recortando distancias con AWS desde 2017. Azure sobresale especialmente en materia de seguridad y en integración con herramientas y servicios de Microsoft lo que lo hace atractivo para organizaciones ya inmersas en dicho ecosistema. En resumen, la evolución de ambos refleja una carrera constante de innovación: AWS aportando rapidez en la introducción de nuevos servicios y amplia experiencia operativa, y Azure enfocándose en la interoperabilidad, la oferta híbrida y la mejora continua para empresas tradicionales.

## **5.2. Modelos de servicio en la nube: IaaS, PaaS, SaaS**

En computación en la nube existen tres modelos principales de servicios: Infraestructura como Servicio (IaaS), Plataforma como Servicio (PaaS) y Software como Servicio (SaaS). Estos modelos definen el nivel de control y responsabilidad que mantiene el usuario versus lo que gestiona el proveedor de nube. A continuación se explican cada uno, junto con ejemplos específicos de AWS y Azure:

- **Infraestructura como Servicio (IaaS):** En IaaS, el proveedor ofrece infraestructura IT virtualizada a demanda: recursos básicos de cómputo, almacenamiento y redes accesibles por internet. El usuario administra el sistema operativo, las aplicaciones y los datos, mientras que el proveedor gestiona el hardware físico subyacente, virtualización y otros componentes básicos. Esto ofrece gran flexibilidad para aprovisionar y escalar solo los recursos necesarios, pagando típicamente solo por uso. *Ejemplos:* Las máquinas virtuales son la forma clásica de IaaS. AWS ofrece Amazon EC2 (Elastic Compute Cloud) como servicio IaaS emblemático, que permite lanzar instancias de servidor virtual en la nube. De forma similar, Azure proporciona Azure Virtual Machines, que cumple el mismo rol en la plataforma de Microsoft. Ambos servicios permiten al usuario elegir sistemas operativos, configurar entornos y tener control casi total del servidor en la nube, pero sin preocuparse por el mantenimiento del hardware físico.
- **Plataforma como Servicio (PaaS):** En PaaS, el proveedor entrega una plataforma completa de desarrollo y despliegue de aplicaciones, gestionando por el cliente la infraestructura subyacente (servidores, almacenamiento, middleware, runtime, etc.). Esto permite a los desarrolladores centrarse solo en el código y la funcionalidad de sus aplicaciones, sin tener que ocuparse de la configuración de servidores o sistemas operativos. El entorno de ejecución, escalado automático, balanceo de carga y actualizaciones de software son manejados por el proveedor. *Ejemplos:* AWS Elastic Beanstalk es un servicio



PaaS de AWS que simplifica el despliegue y escalado de aplicaciones web desarrolladas en lenguajes como Java, .NET, PHP, Node.js, Python, Ruby, Go, etc., sobre servidores conocidos (Apache, Nginx, IIS, etc.). El usuario simplemente carga su código y Elastic Beanstalk automáticamente aprovisiona la infraestructura necesaria, realiza la configuración, despliega la aplicación y escala los recursos según demanda, encargándose de tareas operativas como balanceo de carga, monitoreo y mantenimiento. Por parte de Azure, un ejemplo destacado es Azure App Service, que es la oferta PaaS de Microsoft para alojar aplicaciones web, servicios API REST y back-ends móviles de manera totalmente gestionada. Con Azure App Service, los desarrolladores pueden ejecutar aplicaciones en diversos lenguajes y entornos (.NET, Java, Node.js, Python, PHP, contenedores Docker, etc.) sin administrar la infraestructura de servidores, beneficiándose de escalado automático, implementaciones continuas (CI/CD) e integración con herramientas de desarrollo de Microsoft. En resumen, PaaS ofrece un entorno listo para usar donde desarrollar y desplegar aplicaciones de forma ágil.

- **Software como Servicio (SaaS):** Es el modelo en el que el proveedor ofrece aplicaciones de software completas, que el usuario consume normalmente a través de un navegador web o cliente ligero. Toda la infraestructura, el middleware y la propia aplicación son gestionados por el proveedor; el usuario simplemente utiliza la funcionalidad del software, usualmente bajo un modelo de suscripción. Esto elimina la necesidad de instalar, mantener o actualizar software por parte del usuario. *Ejemplos:* Muchos servicios cotidianos son SaaS. En el ecosistema de Microsoft, Office 365 (Microsoft 365) es un claro ejemplo de SaaS: la suite ofimática completa (Word, Excel, Outlook, Teams, etc.) se ofrece como servicio en la nube, accesible vía web, con Microsoft gestionando la plataforma y actualizaciones. Ese servicio SaaS se apoya en la infraestructura de Azure en la nube para su entrega. Del lado de AWS, si bien Amazon se ha centrado más en IaaS/PaaS, también ofrece algunos SaaS empresariales. Un ejemplo es Amazon WorkMail, un servicio de correo electrónico y calendarios empresariales hospedado en AWS, comparable a soluciones como Microsoft Exchange Online. Otro es Amazon WorkDocs, para almacenamiento y colaboración en documentos. Estos servicios (WorkMail, WorkDocs, Amazon Chime para comunicaciones, etc.) son ofrecidos por AWS listos para usar, sin que el cliente administre servidores de correo o de archivos. En general, también son ejemplos de SaaS aplicaciones como Salesforce (CRM en la nube) o Google Workspace (Gmail, Drive, Docs), en cuyos casos el proveedor (Salesforce, Google) opera todo el software en la nube y el cliente solo lo utiliza vía internet.

Cabe mencionar que, además de IaaS, PaaS y SaaS, existen otros modelos especializados como *Función como Servicio (FaaS)* o *Serverless Computing* (por ejemplo, AWS Lambda, Azure Functions), *Base de Datos como Servicio (DBaaS)*, *Contenedores como Servicio (CaaS)*, entre otros. Sin embargo, IaaS, PaaS y SaaS son los tres pilares fundamentales para clasificar los servicios de nube y entender la diferencia en cuánto gestiona el proveedor vs. el usuario.

### 5.3. Servicios web principales para despliegue de aplicaciones, sitios web y API

Tanto AWS como Azure ofrecen numerosos **servicios web** destinados a facilitar el despliegue de aplicaciones empresariales, sitios web y servicios API, cubriendo distintas arquitecturas (servidores virtuales, plataformas totalmente gestionadas, contenedores, funciones sin servidor, etc.). A continuación se describen los principales servicios equivalentes en ambas plataformas para estos fines:

- **Máquinas Virtuales (IaaS):** La forma más básica de desplegar una aplicación en la nube es mediante máquinas virtuales. AWS ofrece **Amazon EC2** (Elastic Compute Cloud) como su servicio de VM escalable, mientras que Azure cuenta con **Azure Virtual Machines**. Con ambos, los usuarios pueden aprovisionar servidores virtuales con el sistema operativo de su elección, instalar sus aplicaciones y configurar el entorno a medida. Este enfoque proporciona máximo control y flexibilidad (similar a un servidor on-premise), aunque requiere gestionar actualizaciones de SO, parches, escalado manual (o a través de autoescalado configurado) y demás tareas de administración del servidor. Las VM suelen usarse para aplicaciones legadas o casos donde se necesita control total del entorno. Por ejemplo, una empresa puede migrar una aplicación empresarial existente desplegándola en instancias EC2 o Azure VMs, configurando grupos de autoescalado en AWS o conjuntos de escalado en Azure para manejar variaciones de carga.
- **Plataformas de aplicaciones web (PaaS) – Azure App Service vs. AWS Elastic Beanstalk:** Para simplificar el despliegue de **sitios web y aplicaciones web/API** sin administrar servidores, Azure y AWS proveen servicios PaaS robustos:
  - **Azure App Service:** Es la plataforma como servicio de Azure para hospedar aplicaciones web, APIs RESTful y aplicaciones móviles de manera totalmente gestionada. Permite implementar aplicaciones desarrolladas en .NET, Java, Node.js, Python, PHP y otros lenguajes, ejecutándolas en Windows o Linux, e incluso desplegar contenedores Docker personalizados. App Service abstrae al desarrollador de la infraestructura subyacente: **“permite ejecutar aplicaciones web, back-ends móviles y APIs RESTful sin preocuparse**

- por administrar la infraestructura subyacente**". Ofrece características integradas como escalado automático según la demanda, balanceo de carga, certificados SSL, autenticación con redes sociales o Azure AD, copias de seguridad, implementación continua desde repositorios de código (GitHub, Azure DevOps), entre otras. Un uso típico es desplegar un sitio web corporativo o una API en cuestión de minutos: se sube el código (o se conecta el repositorio) en Azure App Service, se configura el plan (por ejemplo, nivel Gratis, Básico, Standard o Premium según recursos necesarios) y Azure se encarga de aprovisionar los servidores, configurar el entorno de ejecución, aplicar actualizaciones de sistema, escalar instancias según el tráfico, etc.
- **AWS Elastic Beanstalk:** Es el servicio PaaS equivalente en AWS. Elastic Beanstalk facilita la puesta en producción de aplicaciones web y servicios **sin que el desarrollador tenga que aprovisionar ni gestionar manualmente la infraestructura de EC2, balanceadores o autoescalado**. Según la documentación de AWS, *"Elastic Beanstalk es un servicio fácil de usar para desplegar y escalar aplicaciones web y servicios desarrollados con Java, .NET, PHP, Node.js, Python, Ruby, Go o Docker en servidores familiares (Apache, Nginx, IIS, etc.). Es una solución completa de gestión de aplicaciones que se encarga de todas las tareas de infraestructura y plataforma en tu nombre"*. En la práctica, el desarrollador solo debe subir el código de su aplicación (por ejemplo un paquete WAR, archivo ZIP del proyecto Node.js, etc.) y **Elastic Beanstalk automáticamente aprovisiona** los recursos necesarios: instancia(s) EC2, contenedores de aplicación, balanceador de carga ELB, grupos de Auto Scaling, y conecta otros servicios como bases de datos si se requieren. También realiza despliegues de nuevas versiones sin tiempo de inactividad apreciable y monitorea la salud de la aplicación. Todo esto *reduce la complejidad operativa* y acelera el proceso de publicación de sitios web y APIs en AWS. Un caso de uso sería desplegar rápidamente una aplicación web corporativa escrita en .NET: en lugar de crear manualmente un servidor IIS en EC2, configurar autoescalado, etc., se utiliza Elastic Beanstalk que hace todo eso automáticamente en unos minutos, permitiendo al desarrollador enfocarse en el código.
  - **Computación sin servidor (FaaS) – Azure Functions vs. AWS Lambda:** Otro método muy popular para desplegar aplicaciones y especialmente *servicios API* es el modelo **serverless** o de funciones en la nube. En este modelo, el código se ejecuta en **funciones efímeras bajo demanda**, sin que el usuario gestione servidores ni instancias continuamente activas; el proveedor escala automáticamente las funciones según las invocaciones.

- **AWS Lambda:** Introducido en 2014, Lambda fue pionero en el concepto FaaS. Permite ejecutar código en respuesta a eventos (por ejemplo, peticiones HTTP, mensajes en colas, cargas de archivos, etc.) sin aprovisionar servidores. El desarrollador escribe funciones (en Node.js, Python, Java, C#, etc.), define los triggers/eventos que las invocan, y AWS se encarga de asignarles recursos en milisegundos cuando se disparan, cobrando solo por el tiempo de ejecución consumido en cada invocación. Esto resulta ideal para **APIs REST** (usando Amazon API Gateway para exponer las funciones Lambda como endpoints HTTP) y para tareas de backend event-driven. Por ejemplo, una empresa puede implementar un servicio API donde cada endpoint HTTP está respaldado por una función Lambda que consulta una base de datos y devuelve un resultado, sin mantener servidores web 24/7.
- **Azure Functions:** Es la contraparte en Azure, con funcionalidad similar. Azure Functions soporta múltiples lenguajes (.NET, JavaScript, Python, Java, etc.) y se integra con Azure API Management o Azure App Service (modo Functions) para exponerse vía HTTP. Azure Functions también permite un modelo de *consumo dinámico*, escalando automáticamente y cobrando por ejecución. Una ventaja en Azure es la posibilidad de ejecutar funciones tanto en la nube pública como en entornos on-premises/híbridos (Azure Functions Runtime) facilitando escenarios de integración. Tanto Lambda como Functions han ganado popularidad porque **simplifican el despliegue de microservicios y APIs**: el desarrollador solo sube código modular y el proveedor se ocupa de la infraestructura elástica.
- **Servicios de contenedores – Azure Kubernetes Service vs. AWS ECS/EKS:** Con la adopción de contenedores (Docker, etc.) en desarrollo de aplicaciones, ambos proveedores ofrecen servicios para desplegar y orquestar contenedores en la nube:
  - **Azure Kubernetes Service (AKS):** es el servicio administrado de Azure para **orquestación de contenedores con Kubernetes**. Permite crear clústeres de Kubernetes sin tener que instalar o mantener el plano de control de Kubernetes; Azure gestiona el master node, actualizaciones y escalado automático de nodos. AKS se integra con Azure CLI, portal y herramientas como Azure DevOps para facilitar el despliegue continuo de aplicaciones contenedorizadas. Azure también ofrece **Azure Container Instances (ACI)**, un servicio más simple para ejecutar contenedores individuales on-demand sin necesidad de orquestador, útil para trabajos batch o microservicios aislados.
  - **AWS Elastic Container Service (ECS) y Elastic Kubernetes Service (EKS):** AWS proporciona ECS como su propio orquestador nativo

(propietario) para contenedores Docker, y EKS como servicio administrado de Kubernetes. Con **Amazon EKS**, se puede ejecutar Kubernetes en AWS con configuración simplificada (AWS administra los componentes maestros, etc.), mientras que **Amazon ECS** ofrece una experiencia de orquestación más integrada con otros servicios AWS, a veces con menor complejidad si se opera enteramente en entorno AWS. Además, AWS introdujo **AWS Fargate**, que permite ejecutar contenedores sin gestionar servidores (serverless containers), funcionando tanto con ECS como con EKS. En términos prácticos, una empresa puede desplegar sus aplicaciones **microservicios** en contenedores Docker: en Azure usar AKS para gestionar los pods y escalado; en AWS optar por EKS (si ya usa Kubernetes) o ECS/Fargate para una solución manejada. Según análisis recientes, *“ambos proveedores ofrecen plataformas de contenedores maduras y flexibles con cobertura comparable”*.

- **Servicios para APIs y desarrollo web frontend:** Para completar el panorama de despliegue web, cabe mencionar:
  - **Servicios de gestión de APIs:** Azure ofrece **Azure API Management**, un servicio PaaS que permite publicar, proteger, monitorear y versionar APIs de forma centralizada. AWS ofrece **Amazon API Gateway** con un objetivo similar: facilitar la publicación de APIs seguras a escala, con control de versiones, cuotas, claves, etc., usualmente en conjunto con Lambda u otros backend. Estos servicios manejan el enrutamiento de solicitudes, limitación de tasa, autenticación, y generación de portales para desarrolladores, útiles cuando se exponen APIs a terceros.
  - **Alojamiento de sitios estáticos y contenido:** AWS S3 puede servir sitios web estáticos directamente desde un bucket (con Amazon CloudFront para CDN), mientras Azure Blob Storage ofrece **Static Website Hosting** para contenido HTML/JS estático, también integrable con Azure CDN. Adicionalmente, Azure tiene **Azure Static Web Apps** (servicio reciente) que combina hosting estático con funciones serverless (integrando Azure Functions) ideal para aplicaciones de una sola página (SPA) con frontend estático y backend sin servidor.
  - **Herramientas de desarrollo y DevOps integradas:** Ambos proveedores fomentan servicios para facilitar el ciclo de vida de las aplicaciones. Azure se integra fuertemente con **Azure DevOps** y **GitHub Actions** para CI/CD hacia App Service, Functions, AKS, etc. AWS cuenta con **AWS CodeStar**, **CodePipeline**, **CodeDeploy** y otros en su suite de DevOps para compilar, testear y desplegar aplicaciones en sus servicios.

En resumen, tanto Azure como AWS cubren *todas las necesidades* para desplegar aplicaciones en la nube. Azure suele destacar en escenarios **híbridos o centrados en tecnologías Microsoft**, ofreciendo caminos más sencillos para quienes vienen de IIS,

SQL Server, .NET, etc. AWS por su parte brinda una enorme variedad de servicios especializados y flexibilidad, lo cual beneficia a empresas que buscan opciones muy específicas de implementación. En la práctica, muchas arquitecturas empresariales en la nube usan una combinación de VM, servicios PaaS, contenedores y funciones serverless para conformar un ecosistema óptimo de alojamiento de aplicaciones, apoyándose en servicios gestionados (bases de datos, mensajería, monitoreo) provistos por el mismo cloud.

## 5.4. Comparación entre Azure y AWS: arquitectura, disponibilidad global, facilidad de uso, precios, escalabilidad y seguridad

A pesar de que Azure y AWS ofrecen **conjuntos de servicios muy similares** en la nube, existen diferencias sutiles en su enfoque, arquitectura y características. A continuación se comparan varios aspectos clave:

### 5.4.1. Arquitectura y disponibilidad global

Ambos proveedores operan una red global de centros de datos organizada en **regiones geográficas** (cada región contiene múltiples centros de datos). AWS introdujo el concepto de **Regiones y Zonas de Disponibilidad (AZ)**: cada región de AWS típicamente posee varias AZ (cada AZ es un centro de datos físicamente aislado, con energía y red independientes, dentro de la misma región) para tolerancia a fallos locales. Microsoft Azure también utiliza regiones, y en años recientes incorporó **Zonas de Disponibilidad** en muchas de ellas (aunque no en todas), así como el concepto único de **regiones emparejadas** (paired regions) para recuperación ante desastres regionales.

En cuanto a alcance global, Azure suele resaltar que tiene **más regiones** que cualquier otro proveedor. De hecho, Azure cuenta con **60+ regiones geográficas** alrededor del mundo donde se pueden desplegar recursos. AWS por su parte opera en **aproximadamente 40 regiones** actualmente, con más de **100 Zonas de Disponibilidad** en total. Aunque Azure tiene mayor número de regiones (al cubrir algunos países con nubes gubernamentales y comerciales separadas, por ejemplo), AWS suele tener **más zonas** por región y una infraestructura de red perimetral muy extensa: más de 400 ubicaciones de borde (Edge Locations) de su CDN CloudFront y múltiples *Local Zones* para acercar recursos a ciudades específicas. Esto significa que **AWS** ha invertido en lograr bajísimas latencias a nivel global, útil para aplicaciones que requieren respuesta casi en tiempo real en cualquier continente. **Azure**, por su parte, también expande continuamente su red y ha alcanzado regiones en geografías donde AWS tiene menor presencia (por ejemplo, mayor cantidad de regiones en África y Medio Oriente). En la práctica, ambos garantizan amplia **cobertura mundial**, con presencia en América, Europa, Asia-Pacífico y otras áreas. Para la mayoría de

empresas, la disponibilidad de un centro de datos cercano estará cubierta por ambos; sin embargo, en ciertos países o para requerimientos de **residencia de datos**, la oferta regional de uno u otro podría ser determinante. Vale añadir que ambas plataformas permiten diseñar aplicaciones altamente disponibles distribuyendo cargas entre múltiples zonas y regiones: AWS con uso de AZs y balanceadores globales (Route 53, etc.), y Azure con sus zonas de disponibilidad y las *paired regions* que aseguran redundancia geográfica separada por >300 millas para resiliencia.

En términos de **arquitectura híbrida**, Azure ha enfatizado más en la integración con entornos on-premise desde sus inicios. Azure ofrece soluciones como **Azure Stack** (para ejecutar una instancia local de Azure), **Azure Arc** (para administrar recursos locales y de múltiples nubes desde Azure) y una integración nativa de Azure Active Directory con Active Directory tradicional, etc. AWS históricamente fue más *cloud-only*, pero ha introducido equivalentes como **AWS Outposts** (hardware AWS para centros de datos del cliente) y mejorado soporte híbrido. Aun así, se reconoce que *“Azure exhibe un fuerte enfoque en nube híbrida... mientras AWS se centra en infraestructura cloud nativa”*. Por ende, organizaciones con grandes entornos legados Windows/AD suelen encontrar en Azure una continuidad más sencilla.

#### 5.4.2. Facilidad de uso y curva de aprendizaje

La **facilidad de uso** puede ser subjetiva y a menudo depende de la familiaridad del usuario con determinadas herramientas. En general, **Azure** es considerado ligeramente más intuitivo para principiantes o para entornos Microsoft, gracias a una interfaz gráfica (*Azure Portal*) unificada y orientada a guiar al usuario con asistentes claros. La experiencia de Azure Portal suele ser visualmente amigable y coherente con otras herramientas Microsoft, lo que beneficia a administradores acostumbrados a entornos Windows. Por ejemplo, Azure integra en su portal muchas vistas de monitoreo, configuraciones rápidas y tutoriales integrados, lo que *“ofrece una navegación clara y una interfaz simplificada, siendo una opción atractiva para una amplia gama de usuarios”*.

AWS, por otro lado, tiene una consola web muy completa pero que puede resultar abrumadora al inicio debido al enorme número de servicios listados. Dado que AWS fue el primero en el mercado, su consola y nomenclatura de servicios fueron creciendo orgánicamente, a veces con una **experiencia menos guiada**. Algunos usuarios opinan que AWS exige una comprensión más profunda desde el comienzo para moverse entre sus servicios. No obstante, AWS ha mejorado su UI con los años y provee un *dashboard* personalizable, documentación contextual, etc. En cuanto a **documentación y recursos de aprendizaje**, AWS ofrece guías técnicas muy exhaustivas, whitepapers y ejemplos de arquitectura, mientras Azure proporciona tutoriales paso a paso y módulos interactivos en Microsoft Learn que muchos

encuentran más fáciles de seguir. De hecho, *“AWS posee documentación técnica más exhaustiva, mientras Azure ofrece tutoriales más accesibles y una incorporación guiada”*. En resumen, ambas plataformas tienen una curva de aprendizaje notable (reflejada también en la amplia oferta de certificaciones que cada una tiene para formar profesionales), pero **Azure puede resultar más familiar para equipos habituados a entornos Microsoft**, y AWS tiende a ser la referencia para gran parte de la comunidad DevOps debido a su madurez y la gran base de usuarios que comparte soluciones y buenas prácticas.

### 5.4.3. Precios y modelos de costo

Comparar precios entre Azure y AWS es complejo, ya que ambos manejan miles de SKU de servicios con distintos modelos de tarificación. En términos generales, **ambos utilizan un modelo de pago por uso (pay-as-you-go)** sin compromisos a largo plazo por defecto. Esto significa que el cliente paga únicamente por los recursos consumidos (por hora o incluso por segundos de cómputo, por GB almacenado al mes, etc.). También ambos ofrecen **capas gratuitas iniciales**: AWS con su *free tier* de 12 meses para muchos servicios y algunas cantidades gratis perpetuas; Azure con un crédito inicial y ciertos servicios gratuitos 12 meses, además de niveles gratuitos continuos (por ejemplo, 1 millón de invocaciones Functions al mes gratis).

No obstante, hay diferencias en **estructura de precios y descuentos**:

- AWS tradicionalmente facturaba instancias EC2 por hora, mientras Azure facturaba por minuto; actualmente ambos ofrecen facturación por segundos en muchos casos (particularmente Linux en EC2 desde 2017). Azure ha sido percibido como más granular en algunos aspectos de facturación (p.ej., prorrateo de VM por minutos exactos).
- Azure tiende a ofrecer **descuentos a clientes existentes de Microsoft**. Por ejemplo, con el beneficio **Azure Hybrid**, las empresas con licencias Windows Server o SQL Server con Software Assurance pueden usarlas en Azure (evitando pagar de nuevo por la licencia en la VM). También hay descuentos si se tiene suscripción Enterprise Agreement de Microsoft. *“El modelo de precios de Azure favorece a usuarios de su ecosistema, ofreciendo descuentos significativos a clientes de Microsoft”*.
- AWS por su parte popularizó las **instancias reservadas** y **Savings Plans**, donde un cliente se compromete a un uso continuo (1 o 3 años, ya sea de una instancia específica o un monto de gasto) a cambio de tarifas mucho más bajas (p. ej., 30-60% de descuento sobre el precio on-demand). Azure también tiene instancias reservadas similares y planes de ahorro propios, pero AWS tiene quizás más opciones refinadas en este terreno para optimizar costos en computación, bases de datos, etc.



- Ambos proveedores ofrecen **descuentos por volumen**. AWS, por ejemplo, reduce automáticamente el costo por GB de almacenamiento S3 o por TB de transferencia de datos a medida que se usa más. Azure también maneja escalas de precios decrecientes en servicios como Azure Storage, pero un diferenciador de AWS son sus múltiples niveles de almacenamiento (S3 Standard, IA, One Zone, Glacier, etc.) muy granulares que permiten ahorros si se ajusta bien la política de ciclo de vida de datos.
- En **complejidad de precios**, algunos analistas señalan que AWS puede ser más **intrincado** debido a la multitud de opciones. *“Uno de los desafíos en AWS es su modelo de precios complejo; la multitud de servicios y opciones puede dificultar estimar costos con precisión”*. Esto obliga a planificar cuidadosamente, utilizando herramientas como calculadoras de precio o AWS Cost Explorer para monitorear gastos. Azure, por comparación, *“es conocido por un modelo de precios más sencillo, que facilita a los usuarios entender y estimar costos”*. Microsoft ofrece una calculadora visual muy detallada y suele tener estructuras de precios más homogéneas en algunos servicios. Aun así, Azure tampoco está libre de complejidad, especialmente a medida que su portafolio se acerca en tamaño al de AWS.

En términos de **precio absoluto**, no existe un ganador claro: dependiendo del servicio y la carga, uno u otro podría resultar más económico. Por ejemplo, máquinas Linux de cierta categoría pueden ser más baratas en AWS, mientras bases de datos SQL gestionadas pueden tener mejor precio en Azure con sus descuentos a licencias. Los costos de salida de datos (egreso) de Azure a internet a veces son ligeramente menores en regiones comunes, pero AWS ofrece tráfico entrante gratuito universalmente y ciertas promociones. En cualquier caso, **ambos proveedores compiten agresivamente en precios** y suelen bajar costos o mejorar prestaciones periódicamente. Para una empresa, la recomendación es aprovechar las herramientas de cada plataforma para optimizar gastos: p. ej., utilizar instancias reservadas/Savings Plan en AWS o Azure según corresponda, apagar recursos fuera de horario, escalar automáticamente, etc.

#### 5.4.4. Escalabilidad y rendimiento

Tanto AWS como Azure fueron diseñados con la **escalabilidad masiva** como objetivo central. Los dos ofrecen la capacidad de iniciar desde pequeñas pruebas de concepto hasta crecer a aplicaciones globales con millones de usuarios, de forma elástica. Algunos puntos a considerar:

- **Escalado automático:** AWS introdujo desde temprano (circa 2009) su servicio **Auto Scaling** junto con EC2, permitiendo definir políticas para agregar o quitar instancias EC2 basadas en métricas (CPU, tráfico, etc.). Azure ofrece

funcionalidad análoga de **Autoscale** para sus VM Scale Sets y para servicios PaaS (por ejemplo, Azure App Service puede aumentar instancias de forma automática según CPU o cola de peticiones). En general, AWS y Azure permiten escalar **verticalmente** (aumentar el tamaño de una instancia/servicio) y **horizontalmente** (añadir más instancias en paralelo) según la demanda. Una aplicación correctamente diseñada en cualquiera de las nubes puede manejar cargas variables eficientemente usando estas capacidades.

- **Variedad de tamaños y rendimiento de instancias:** AWS posee quizás la gama más amplia de tipos de instancia de cómputo (EC2) optimizadas para diferentes casos (cálculo intensivo, memoria, almacenamiento I/O, GPUs, FPGA, etc.), incluyendo familias como las *Graviton* (basadas en ARM de alto rendimiento y menor costo). Azure tiene también numerosas series de VM (A, D, E, F, H para HPC, NV/ND para GPU, etc.) que ofrecen opciones comparables a las de AWS. En almacenamiento, ambos tienen opciones de alta escalabilidad: Amazon S3 y Azure Blob Storage son capaces de manejar virtualmente **volúmenes ilimitados de datos** con rendimiento ajustable mediante escalado horizontal. Para *throughput* de red, AWS y Azure han desarrollado redes backbone globales capaces de soportar aplicaciones de baja latencia y alto ancho de banda (por ejemplo, ambos soportan hasta 100 Gbps de throughput en instancias especializadas).
- **Alcance global de baja latencia:** AWS, al tener una red de edge muy densa (CloudFront, Local Zones), puede servir contenidos o servicios con milisegundos de respuesta a usuarios en muchas ciudades. Azure también invirtió en mejorar su red; un dato destacable es que Azure a veces **opera con más proximidad geográfica a ciertos mercados** gracias a su mayor número de regiones (por ejemplo, tiene más de una región en algunos países donde AWS solo tiene una). En definitiva, en cuanto a rendimiento, ninguna de las plataformas supone un cuello de botella significativo – ambas pueden soportar aplicaciones de misión crítica, alta concurrencia y big data sin problema, siempre que la arquitectura esté bien dimensionada.

Un punto diferenciador es que **AWS tiene más trayectoria en cargas de trabajo extremadamente especializadas de alto rendimiento (HPC)**. Por ejemplo, AWS dispone de instancias con chips custom (Inferentia, Trainium para machine learning) y ha mostrado casos de uso de supercómputo en la nube (simulaciones científicas, genómica, etc.) apoyados por su amplia gama de instancias HPC y interconexiones de baja latencia. Azure también soporta HPC (tiene la serie H de VMs, por ejemplo, y ha incorporado InfiniBand en la nube), pero AWS suele citarse más en este espacio por madurez. No obstante, Azure ha innovado con integración de **FPGA en su red (Proyecto Catapult)** para acelerar cargas de IA y ofrece servicios avanzados como Azure ML con aceleración, por lo que la brecha se ha cerrado.

En resumen, **en escalabilidad ambos están parejos**: permiten escalar de forma prácticamente ilimitada tanto infraestructuras web, móviles, IoT, big data, etc. Los límites predeterminados (p. ej., número de VMs por suscripción) se pueden aumentar previa solicitud en ambos. La capacidad de soportar picos repentinos o crecimiento global sostenido es intrínseca al diseño de Azure y AWS, al punto que muchas de las compañías de internet más grandes del mundo corren sobre uno u otro proveedor (o ambos).

#### 5.4.5. Seguridad, cumplimiento y gestión de identidades

La seguridad en la nube se maneja bajo un modelo de **responsabilidad compartida**: el proveedor asegura la *seguridad de la nube* (infraestructura física, hipervisores, red troncal), mientras el cliente es responsable de la *seguridad en la nube* (configuración segura de sus recursos, protección de datos, gestión de accesos). Tanto AWS como Azure invierten fuertemente en seguridad y ofrecen numerosas herramientas para ayudar al cliente a proteger sus entornos.

**Gestión de identidades y control de acceso:** Aquí se nota una diferencia de enfoque. AWS cuenta con **AWS Identity and Access Management (IAM)**, que permite definir usuarios, roles y políticas muy granulares para acceder a los recursos AWS. IAM se integra servicio por servicio y utiliza políticas JSON para permisos. Azure, en cambio, utiliza **Azure Active Directory (Azure AD)** como núcleo de identidad para todos sus servicios en la nube. Azure AD proporciona autenticación federada, inicio de sesión único (SSO) y una integración profunda con identidades corporativas de Windows (los directorios Active Directory locales pueden sincronizarse o federarse con Azure AD). De hecho, Azure AD es la base no solo para Azure sino también para Office 365 y otros servicios SaaS de Microsoft, ofreciendo una experiencia unificada de identidad híbrida. *“Azure utiliza Azure Active Directory... integrado con Windows y Microsoft 365; en contraste, AWS emplea IAM con control de acceso de alta granularidad por políticas”*. Para los administradores, esto significa que si la empresa ya tiene un directorio de empleados, Azure facilita extender ese mismo esquema de usuarios y grupos a la nube. AWS IAM por su parte es independiente pero muy robusto en control detallado de cada acción que un servicio puede realizar o no. Ambos soportan características como **MFA (autenticación multifactor)**, permisos temporales delegados, integración con proveedores SAML/OAuth, etc., aunque por ecosistema, Azure AD suele brillar en entornos Windows y AWS IAM en entornos cloud-native Linux.

**Cifrado y gestión de claves:** Los dos proveedores ofrecen cifrado de datos en tránsito y en reposo en prácticamente todos sus servicios (muchas veces activado por defecto). Para gestionar las claves de cifrado, Azure proporciona **Azure Key Vault**, que permite almacenar claves, secretos y certificados de forma segura (incluyendo opciones de módulos HSM). AWS ofrece un servicio similar llamado **AWS Key Management**

**Service (KMS)**, así como AWS CloudHSM para hardware dedicado. Ambos soportan esquemas de *BYOK (Bring Your Own Key)* por los cuales el cliente puede proveer sus propias claves maestras. Una diferencia es que Azure Key Vault se integra muy de cerca con servicios de Azure y Active Directory, facilitando políticas de acceso basadas en identidad corporativa, mientras AWS KMS está muy integrado a IAM y políticas de recurso AWS. En la práctica, ambos permiten lograr altos estándares de cifrado (AES-256, RSA 2048+, etc.) y cumplir requisitos de seguridad para datos sensibles.

**Detección de amenazas y monitoreo de seguridad:** Azure y AWS proveen servicios inteligentes para ayudar a vigilar posibles amenazas:

- Azure incluye **Microsoft Defender for Cloud** (antes Azure Security Center) que evalúa configuraciones, recomienda mejoras de seguridad y detecta actividades sospechosas en recursos de Azure, además del SIEM/SOAR en la nube **Microsoft Sentinel** para correlacionar eventos de seguridad a gran escala, incluso en entornos híbridos/multinube.
- AWS ofrece **Amazon GuardDuty**, un servicio administrado que analiza logs (de CloudTrail, VPC Flow Logs, DNS logs) en busca de patrones anómalos o maliciosos, alertando de posibles compromisos. También servicios como AWS Security Hub para centralizar hallazgos de seguridad de varios servicios (GuardDuty, Inspector, Macie, etc.).

Ambos proveedores mantienen una amplia certificación en **cumplimiento normativo**: cumplen con estándares internacionales y sectoriales como ISO 27001, SOC 1-2-3, PCI-DSS para procesamiento de pagos, normas de protección de datos de salud (HIPAA en EE.UU.), estándares gubernamentales (FedRAMP, ENS en España, etc.) y muchos otros. AWS y Azure publican documentación de cumplimiento y ofrecen acuerdos de procesador de datos (DPA) para ayudar a clientes a cumplir GDPR y otras leyes. Azure a veces es preferido en sectores muy regulados (gobierno, finanzas, salud) porque **destaca en seguridad y cumplimiento, siendo ideal para sectores regulados**, en parte debido a las opciones de nube gubernamental separada que Microsoft ofrece y su largo historial con clientes enterprise. No obstante, AWS igualmente sirve a numerosas entidades altamente reguladas (por ejemplo, es proveedor de muchos organismos gubernamentales en EEUU y otros países) y ofrece servicios especiales como **AWS Artifact** (portal para acceder a documentación de compliance) y regiones aisladas como AWS GovCloud.

En cuanto a **seguridad física y de infraestructura**, ambos tienen similares medidas: centros de datos con control de acceso estricto, vigilancia, redundancias de energía y clima, y equipos dedicados a ciberseguridad 24/7. También ambos cuentan con programas de respuesta a incidentes y comparten públicamente información de estado y disponibilidad de sus servicios (dashboards de estatus).

Entonces Azure y AWS proveen un entorno muy seguro si se utilizan correctamente las herramientas disponibles. AWS tiene una ligera reputación de **controles granulares** (gracias a IAM y su cultura de *security-first*), mientras Azure aprovecha la familiaridad de los administradores corporativos en torno a Active Directory y ofrece facilidad en configuraciones seguras por defecto en entornos Windows. La **seguridad no es un diferenciador decisivo** entre los dos para la mayoría de casos, ya que ambos cumplen altos estándares; más bien, la diferencia estará en qué plataforma le resulta más cómoda al equipo de seguridad de la empresa para integrar con sus procesos existentes.

## 6.Requerimientos para la implementación del proyecto

### 6.1. Requerimientos técnicos

#### 6.1.1. Servicios en la nube de Microsoft Azure

Para llevar a cabo la migración y modernización de la infraestructura tecnológica de la empresa TierraByte, se utilizarán los siguientes servicios de Azure:

SERVICIO	FUNCIÓN ESPECÍFICA
AZURE APP SERVICE	Hospedaje de la aplicación web principal (sitio institucional, panel de administración, etc.).
AZURE SQL DATABASE	Almacenamiento de información estructurada: clientes, productos, facturas e inventario.
AZURE STORAGE (BLOB)	Almacenamiento de archivos no estructurados, como imágenes de productos, documentos y respaldos.
AZURE VIRTUAL MACHINES (OPCIONAL)	Para virtualizar el servidor local si se requiere migrar una aplicación legada.
AZURE BACKUP	Implementación de copias de seguridad automáticas.
AZURE ACTIVE DIRECTORY (AZURE AD)	Control de acceso y autenticación para empleados y administradores.
AZURE MONITOR + APPLICATION INSIGHTS	Supervisión del rendimiento, tráfico y errores del sistema.

#### 6.1.2. Infraestructura y software necesarios (del lado del cliente)

- **Conexión a Internet estable** (mínimo 10 Mbps) para acceso constante a la nube.
- **Navegador web moderno** compatible con Azure Portal (Edge, Chrome, Firefox).
- **Licencia de Windows o Microsoft 365** (opcional) si se desea integrar funciones de productividad con Azure.
- **Editor de código o IDE** (Visual Studio Code, Visual Studio, etc.) para desarrollo y despliegue.
- **Cuenta en Microsoft Azure** con plan gratuito o suscripción inicial para pruebas.
- **Repositorio GitHub o Azure Repos** para despliegue continuo (CI/CD), opcional pero recomendado.

## 6.2. Requerimientos operativos y humanos

RECURSO HUMANO	ROL
ADMINISTRADOR DE SISTEMAS O DEVOPS	Encargado de configurar recursos en Azure, gestionar servicios y seguridad.
DESARROLLADOR WEB	Adaptar o crear la aplicación para ser desplegada en Azure App Service.
ENCARGADO DE TI / SOPORTE TÉCNICO	Migrar datos de los sistemas antiguos y validar su integridad.
CAPACITACIÓN BÁSICA PARA PERSONAL DE TIERRABYTE	Uso del nuevo sistema web y acceso remoto.

## 6.3. Requisitos generales del proyecto

- **Disponibilidad de tiempo estimada:** 2 a 3 semanas para pruebas y despliegue inicial.
- **Presupuesto estimado:** Se detallará en la sección de costos, pero se comenzará con el uso del plan gratuito y escalamiento mínimo.
- **Documentación técnica:** Manual de configuración, bitácora de cambios, guías de acceso y respaldo.
- **Mecanismos de respaldo:** Implementar backup diario automático con retención mínima de 7 días.

# 7. Costos reales de implementación de la solución en la nube por servicio

Para una solución web empresarial básica desplegada en Microsoft Azure con los servicios listados, se presentan a continuación los costos aproximados mensuales en dólares estadounidenses. La tabla incluye cada servicio, su función, el plan/tier

sugerido para un escenario de pequeña a mediana escala, el precio estimado por mes y la fuente oficial del dato de precio:

NOMBRE DEL SERVICIO	Descripción	Plan/Nivel sugerido	Precio estimado (USD/mes)	Fuente del precio
<b>AZURE APP SERVICE</b>	Plataforma PaaS para desplegar y escalar aplicaciones web.	Plan Básico B1 (1 vCPU, 1.75 GB RAM, 10 GB storage) – adecuado para app pequeña/mediana.	~US\$54.75/mes (B1 Windows)	Azure Q&A (precio B1)
<b>AZURE SQL DATABASE</b>	Base de datos SQL en la nube totalmente administrada (PaaS).	Tier Standard, instancia S0 (10 DTU, hasta 250 GB) – soporta ~5–10 GB de datos.	~US\$15/mes (S0, 5 DTU)	Azure Pricing (SQL S0)
<b>AZURE STORAGE (BLOB)</b>	Almacenamiento de blobs para archivos (imágenes, documentos).	Cuenta con redundancia local (LRS), acceso <b>Hot</b> (frecuente).	~US\$0.018 por GB/mes (≈US\$0.18 por 10 GB)	Azure Pricing (Blob Hot LRS)
<b>AZURE BACKUP</b>	Servicio de respaldo automatizado de datos/VMs (retención 7 días).	Instancia protegida ≤50 GB + bóveda LRS (Storage).	~US\$5/mes por instancia + almacenamiento (ejemplo: +~US\$1 por ~10–20 GB)	Azure Pricing (Backup)
<b>AZURE VIRTUAL MACHINE(OPSIONAL)</b>	Máquina virtual en Azure para ejecutar un servicio legado si es necesario.	Tamaño pequeño B1s (1 vCPU, 1 GB RAM, serie <b>Bs</b> burstable).	~US\$8.76/mes (Linux OS)	Azure VM Pricing (B1s)
<b>AZURE ACTIVE DIRECTORY</b>	Directorio de identidades en la nube para autenticación y SSO.	Edición <b>Free</b> (básica para usuarios/grupos).	US\$0/mes (incluido sin costo con Azure)	F1 Group (Azure AD Free)
<b>AZURE MONITOR (APP INSIGHTS)</b>	Monitoreo de aplicaciones (métricas, registros, trazas) y rendimiento.	Plan <b>Pay-as-you-go</b> (telemetría bajo demanda, 5 GB gratis/mes incluidos).	US\$0/mes (≤5 GB de datos; luego ~US\$2.76/GB)	OpsMan (Azure Monitor Logs)

## 8. Esquema de solución

ASPECTO	ANTES (INFRAESTRUCTURA LOCAL)	DESPUÉS (SOLUCIÓN EN LA NUBE CON AZURE)
UBICACIÓN DEL SISTEMA	Servidor físico ubicado dentro de la tienda TierraByte.	Infraestructura alojada en la nube de Microsoft Azure.
ACCESO AL SISTEMA	Solo accesible desde equipos conectados a la red local.	Accesible desde cualquier lugar con conexión a Internet.
RESPALDO DE DATOS	Realizado de forma manual y no periódica. Riesgo alto de pérdida.	Automatizado mediante Azure Backup con políticas de retención definidas.
ESCALABILIDAD	Limitada por capacidad del servidor físico.	Escalabilidad flexible según la demanda de uso.
DISPONIBILIDAD WEB	No disponible en línea. Sin acceso externo al sistema.	Sitio web y panel de gestión accesibles desde navegadores web.
SEGURIDAD	Sin controles de acceso avanzados. Sin cifrado.	Autenticación mediante Azure Active Directory. Datos cifrados.
COSTOS	Inversión inicial alta y gastos frecuentes en mantenimiento.	Modelo de pago por uso mensual. Reducción de costos de infraestructura.



## 9. Conclusion

La transformación digital de las pequeñas y medianas empresas se ha convertido en una necesidad estratégica para garantizar su competitividad, continuidad operativa y capacidad de adaptación al entorno actual. En el presente proyecto se analizó la situación tecnológica de la empresa TierraByte, identificando sus principales limitaciones en cuanto a infraestructura física, seguridad de la información, accesibilidad y escalabilidad.

A partir de ese diagnóstico, se propuso una solución basada en servicios web en la nube de Microsoft Azure, seleccionando componentes específicos como Azure App Service, Azure SQL Database, Azure Backup y Azure Active Directory para cubrir los requerimientos de operación de la empresa. Se estimaron los costos reales de implementación, se comparó el estado actual y el futuro esperado mediante un esquema “antes y después”, y se detallaron los beneficios operativos, técnicos y económicos de migrar a la nube.

La implementación de esta solución no solo representa una modernización tecnológica, sino una oportunidad para optimizar procesos, reducir costos y brindar mayor disponibilidad de los sistemas empresariales. Se concluye que el uso de servicios web en la nube mediante plataformas como Azure es una alternativa viable, escalable y segura para empresas como TierraByte que buscan crecer y digitalizarse de forma sostenible.

## 10. Bibliografía

**Gartner. (2024).** *Annual Report on IT Infrastructure Reliability and System Outages*. Gartner Research.

**Microsoft Corporation. (2023).** *Estado de la Transformación Digital en Pequeñas y Medianas Empresas de Latinoamérica*. Microsoft Latin America Research.

**Superintendencia General de Electricidad y Telecomunicaciones (SIGET). (2023).** *Informe anual sobre calidad del servicio eléctrico en El Salvador*. SIGET.

**Avsar, N. (2023, 6 de noviembre).** *Understanding Cloud Deployment and Service Models in Cloud Computing — Part 1*. Medium. Recuperado de <https://medium.com/@neslihannavsar/understanding-cloud-deployment-and-service-models-in-cloud-computing-1-accacfa1ee9>

**Domínguez, S. (2024).** *AWS vs Azure: Comparativa de gigantes en la nube*. OpenWebinars (blog). Publicado 6/11/2019, actualizado 19/01/2024. Recuperado de <https://openwebinars.net/blog/aws-vs-azure/>

**Duong, E. (2024, 26 de febrero).** *AWS, Azure, and GCP: 6 Key differences - Cloud Computing*. Medium. Recuperado de <https://medium.com/@ethan.duong1120/aws-azure-and-gcp-6-key-differences-cloud-computing-5a073a956a38>

**Fivetran (2025).** *Microsoft Azure vs AWS: A comprehensive comparison guide*. Fivetran Learn. Recuperado de <https://www.fivetran.com/learn/microsoft-azure-vs-aws>

**Microsoft (2023).** *App Service overview*. Microsoft Learn – Azure App Service. Recuperado de <https://learn.microsoft.com/en-us/azure/app-service/overview>

**Microsoft (2023).** *Azure API Management – Overview*. Microsoft Learn – Azure API Management. Recuperado de <https://learn.microsoft.com/en-us/azure/api-management/api-management-key-concepts>

**Red Hat (2022, 16 de agosto).** *IaaS vs. PaaS vs. SaaS*. Red Hat Official Blog. Recuperado de <https://www.redhat.com/en/topics/cloud-computing/iaas-vs-paas-vs-saas>

**Zerviz (2024, 5 de septiembre).** *AWS vs. Azure: comparación detallada de los gigantes de la nube*. Zerviz Blog. Recuperado de <https://www.zervizgroup.com/aws-vs-azure/>

**AWS Whitepaper (s.f.).** *Overview of Deployment Options on AWS – AWS Elastic Beanstalk*. Amazon Web Services. Recuperado de <https://docs.aws.amazon.com/whitepapers/latest/overview-deployment-options/aws-elastic-beanstalk.html>

**AWS Documentation** (s.f.). *Deploying applications to AWS Elastic Beanstalk*. Amazon Web Services. Recuperado de <https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/Welcome.html>

**Microsoft Tech Community** (2018, 7 de septiembre). *History of Microsoft Azure*. Microsoft Azure Tech Community Blog. Recuperado de <https://techcommunity.microsoft.com/t5/educator-developer-blog/history-of-microsoft-azure/bc-p/257473> (Historia del lanzamiento de Azure en 2008-2010).

**Datademia** (2024). *La historia de AWS, Microsoft Azure y GCP*. Datademia Blog. Recuperado de <https://datademia.es/blog/historia-aws-microsoft-azure-gcp>