

**PROPUESTA DE ARQUITECTURA BASADA EN MICROSERVICIOS PARA
BILLETERAS DIGITALES.**

LUIS DANIEL SIERRA PINEDA

Monografía para optar al título de Ingeniero de Telecomunicaciones.

Director.

Rafael Orlando Cubillos Sanchez.

UNIVERSIDAD SANTO TOMÁS.

DEPARTAMENTO DE INGENIERÍAS.

INGENIERÍA DE TELECOMUNICACIONES.

BOGOTÁ D.C.

2022.

TABLA DE CONTENIDO

INTRODUCCIÓN.

CAPÍTULO 1. ASPECTOS GENERALES.

1.1 PLANTEAMIENTO DEL PROBLEMA.

1.2 JUSTIFICACIÓN.

1.3 OBJETIVOS.

1.3.1 OBJETIVO GENERAL.

1.3.2 OBJETIVOS ESPECÍFICOS.

1.4 METODOLOGÍA PROPUESTA.

1.5 ESTRUCTURA DE DOCUMENTO.

CAPÍTULO 2. MARCO TEÓRICO

2.1 CLOUD COMPUTING.

2.2 APLICACIONES BANCARIAS

2.3 PROVEEDORES CLOUD

2.4 VIRTUALIZACIÓN

2.4.1 CONTENEDORES (DOCKER)

2.4.2 MICROSERVICIOS

2.5 SERVICIOS ADMINISTRADOS PARA CONTENEDORES

2.5.1 DOCKER SWARM.

2.5.2 EKS

2.5.3 ECS

2.5.4 AWS FARGATE

2.5.5 EC2

CAPÍTULO 3. INFRAESTRUCTURA PROPUESTA.

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

BIBLIOGRAFÍA.

LISTA DE TABLAS.

Tabla 1. Comparación entre contenedores y máquinas virtuales.

LISTA DE FIGURAS.

Figura 1. Arquitectura monolítica en servicios financieros.

Figura 2. Ventajas Cloud Computing.

Figura 3. Banca tradicional.

Figura 4. Tendencias de usuarios para soluciones financieras.

Figura 5. Cuadrante mágico de Gartner - Proveedores cloud.

Figura 6. Virtualización.

Figura 7. Tecnología de contenedores.

Figura 8. Soluciones empresariales tradicionales.

Figura 9. Arquitectura de microservicios.

Figura 10. Amazon EKS.

Figura 11. Amazon ECS.

Figura 12. Amazon EC2.

Figura 13. Arquitectura bancaria propuesta basada en microservicios.

INTRODUCCIÓN.

Las tecnologías de computación en la nube se basan en la entrega y consumo de servicios de forma escalable y elásticas de forma ágil e inmediata, a través de un proveedor que presta el servicio frente a la necesidad de un consumidor que busca adquirir recursos que una solución requiera o le hagan falta para sus actividades. La industria se ha beneficiado con estos servicios accesibles a través de internet, ya que el acceso directo, sin intermediarios, ajustándose a la demanda del usuario, pagando por uso, facilita a las organizaciones adaptarse, organizándose a las necesidades de la misma y de la industria, ahorrando costos, evitando desperdicio de recursos TI, con las características que trae implementar infraestructura en la nube, bajo el modelo de infraestructura como servicio (IaaS), muchas organizaciones están atravesando cambios, aplicando el uso de modelos y tecnologías emergentes, que facilitan la escalabilidad de aplicaciones, aumentando la productividad, con una mejor experiencia de cara al cliente.

Los equipos de desarrollo y operaciones en el desarrollo de una aplicación consideran las mejores tecnologías para la implementación de infraestructura. Existen varios modelos para la implementación de soluciones en la nube, se explicará cada uno, haciendo un enfoque en las características que trae la creación de servicios independientes para soluciones financieras a través del uso de tecnologías de virtualización implementadas en plataformas cloud, planteando una infraestructura que tenga las características necesarias para un correcto funcionamiento.

CAPÍTULO 1.

ASPECTOS GENERALES.

1.1 PLANTEAMIENTO DEL PROBLEMA.

En el desarrollo de aplicaciones bancarias la adopción de prácticas, procesos y tecnologías novedosas enfocadas en la agilidad de implementación, permite alcanzar los objetivos del negocio, reduciendo tiempos de comercialización, creando aplicaciones confiables y estables, con tiempos de recuperación bajos ante desastres, manteniendo una calidad alta en la entrega al usuario. Los servicios cloud traen varias propiedades necesarias a examinar para la implementación de tecnologías y modelos de infraestructura, por la importancia que trae para cualquier industria, tal como la bancaria, impactada por la innovación que los servicios cloud otorgan.

En los inicios de un proyecto, muchos equipos implementan un modelo de arquitectura tradicional, que unifica todos los componentes en un bloque, llamada arquitectura monolítica, en la mayoría de los casos porque los equipos están más familiarizados este tipo de modelo tradicional, los cambios constantes en el código y crecimiento de la solución usando esta construcción trae inconvenientes en el funcionamiento, el desconocimiento técnico forman parte de las razones del porque muchas organizaciones no dan el paso a la implementación de

una arquitectura basada en servicios independientes, causando muchos problemas en la disponibilidad de la solución, siendo común una indisponibilidades totales si hay fallas en alguna funcionalidad, para los equipos de operaciones se convierte en un problema para el aprovisionamiento de servicios, además de afectar a los equipos de desarrollo por la complejidad innecesaria que adquiere el código, siendo en muchos casos incomprensible para las personas nuevas que se sumen al equipo, terminando en muchos casos en el abandono del proyecto, múltiples efectos y causas, presentados en el árbol de problemas de la figura 1.

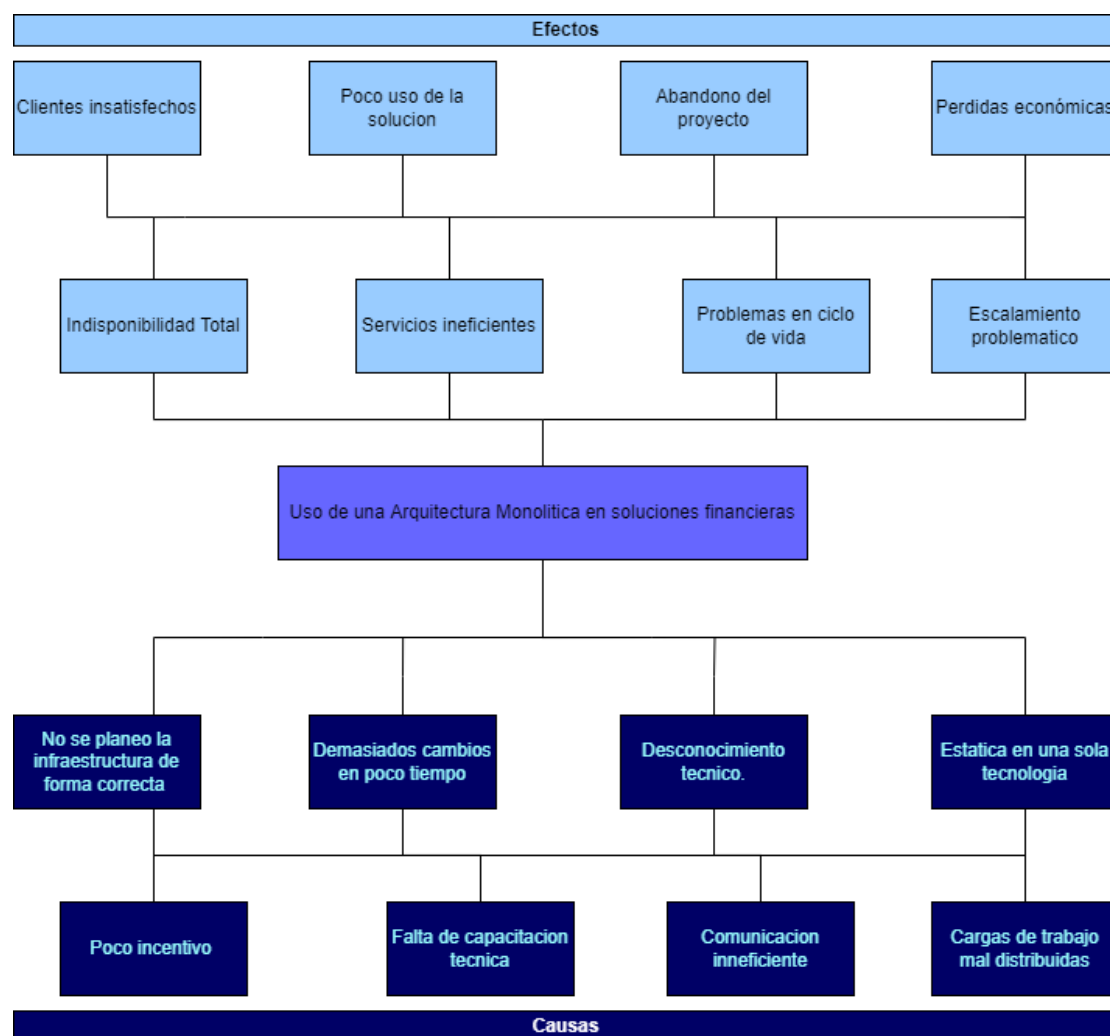


Figura 1. Arquitectura monolítica en servicios financieros.
Creación Propia

La Figura 1 siendo el árbol de problemas trae como problema central el uso de la arquitectura monolítica en soluciones financieras, las causas directas de uso se pueden atribuir a la incorrecta planeación de la arquitectura de la solución, el equipo de desarrollo al tener poco tiempo para realizar entregas, deciden ir por la solución más tradicional optando por establecer todo el código contenga absolutamente toda la funcionalidad de la aplicación, como lo expone la importante desarrolladora de Software Atlassian en la comparación que realizan entre las arquitecturas monolíticas y las arquitecturas basada en servicios [1] . Muchos equipos aún no están familiarizados con la arquitectura orientada a servicios, determinado el modelo a usar en el poco tiempo que se debe disponer para el desarrollo, sin posibilidad de invertir tiempos en el estudio e implementación de nuevas ideas en ambientes de laboratorio, quedando estáticos en una sola tecnología.

1.2 JUSTIFICACIÓN.

La innovación y modernización son características que muchas organizaciones están asociado en sus soluciones empresariales, los consumidores requieren versatilidad, usabilidad y rapidez al momento de usar la solución, siendo necesario que las empresas se armen con equipos ágiles reducidos, enfocados en el desarrollo de software de forma ágil, en menos tiempo y con rápida publicación para el consumo de la solución por parte de los usuarios finales, por lo que una infraestructura tecnológica más simple con sistemas de escalado auto gestionados presenta muchas ventajas en aplicaciones bancarias, la informática en la nube brinda las capacidades de innovación por su agilidad, volviendo un proceso sencillo la implementación de servicios informáticos, también presenta una elasticidad, aprovisionando los servicios en la medida que se necesite, evitando el desperdicio de recursos, ahorrando costos, pagando en base a la demanda de lo que se utiliza, con reducción de costos significativos que para muchas empresas era una limitante al momento de implementar soluciones que satisficiera las necesidades de los consumidores.

Muchas Organizaciones están migrando su infraestructura a servicios cloud, siendo una de las principales prioridades que se fijan las empresas, el 70% de las organizaciones priman por el uso de servicios cloud en sus operaciones comerciales [2]. El sector financiero se ve muy beneficiado con plataformas nativas en la nube, dando muchas ventajas de frente al usuario, mejorando la velocidad operativa en el procesamiento de datos y acceso a la información, mejorando la experiencia del usuario, con una alta disponibilidad, ofreciendo servicios

bancarios en cualquier momento y lugar, además de la seguridad que los proveedores cloud dan a sus servicios de infraestructura, buscando y mitigando cualquier vulnerabilidad en los escaneos que se realizan de forma constante.

Con tantas ventajas de implementación de aplicaciones nativas en la nube, vienen grandes retos para los equipos TI en el desarrollo de soluciones acertadas, que beneficien y aprovechen todo el potencial que los servicios informáticos en la nube trae consigo, siendo importante analizar qué características tiene implementar un modelo basado en microservicios o su modelo de comparación, la arquitectura monolítica, mostrando que particularidades de implementación tienen cada una, teniendo en mira que los equipos TI administren de la mejor forma la infraestructura cloud, optimizando procesos que muchos servicios de orquestación maneja.

1.3 OBJETIVOS.

1.3.1. GENERAL.

Proponer una infraestructura basada en microservicios para soluciones financieras, basada en las mejores prácticas para contenedores, en la plataforma cloud que presente las mejores características de implementación.

1.3.2 ESPECÍFICOS.

- Mostrar las propiedades de la tecnología cloud.
- Enunciar las tendencias de las aplicaciones con servicios financieros.
- Exponer los diferentes proveedores de servicios cloud a través de investigación de mercado.
- Nombrar las características de las tecnologías de virtualización, describiendo cada una, mostrando la tecnología de contenedorización.
- Explicar los servicios cloud disponibles para la implementación de microservicios.
- Proponer una infraestructura en la solución cloud que presente las capacidades de administración necesarias para una solución financiera.

1.4 METODOLOGÍA PROPUESTA.

Para el desarrollo de la investigación se definirán los conceptos básicos de los servicios Cloud, bajo el modelo de pago bajo demanda, estableciendo las características y modelos que tiene esta tecnología, presentando la importancia para los equipos TI en el desarrollo de software.

Las plataformas bancarias actualmente necesitan modernizar su infraestructura financiera para prestar un mejor servicio, se documentaran las tendencias y necesidades actuales para las organizaciones que prestan servicios financieros, donde los servicios cloud innovan para el desarrollo escalable de soluciones modernas, siendo necesario analizar cuál plataforma de nube pública presenta mejores prestaciones en base a publicaciones que presentan los líderes de las plataformas, que mejor entienden las nuevas tecnologías y los panoramas en las plataformas IaaS y PaaS.

El uso de tecnologías de virtualización de software, usando contenedores para la implementación de microservicios, presenta varias ventajas frente a tecnologías más tradicionales, se documentara las características de los contenedores aprovechables para la industria financiera usándolos para la implementación de microservicios.

Crear infraestructura basada en servicios independientes requiere ciertas características informáticas para que el funcionamiento sea óptimo, independientemente de la escala, la carga o la complejidad, El líder del sector debe ofrecer servicios para la creación y administración, enumerando cada uno de ellos, con las propiedades diferenciales en cada ecosistema, proponiendo una infraestructura óptima para servicios financieros en el ecosistema más conveniente para los equipos TI.

CAPÍTULO 2.

MARCO TEÓRICO.

2.1 CLOUD COMPUTING.

La computación en la nube provee nuevos modos de uso y oferta de recursos informáticos, siendo proporcionados en base a la demanda de utilización, convirtiéndose en el paradigma “paga sobre la marcha” o “pay as you go” cobrando por consumo, además del aprovisionamiento de hardware, software y redes para cualquier persona con acceso a Internet. Las soluciones de computación en la nube son instrumentos poderosos diseñados para hacer realidad muchas ideas de negocio que por limitantes económicas antes no se podían realizar, por el costo elevado que conlleva adquirir una infraestructura dedicada. De acuerdo al Instituto Nacional de Estándares y Tecnología de los Estados Unidos (NIST) la computación en la nube es definida como:

“Un modelo que permite el acceso bajo demanda y a través de la red a un conjunto de recursos compartidos y configurables (como redes, servidores capacidad de almacenamiento, aplicaciones y servicios) que pueden ser rápidamente asignados y liberados con una mínima gestión por parte del proveedor del servicio” [3]

En la recomendación “Special Publication 800-145” [3] también se definen las principales características que el modelo debe tener.

- Autoservicio bajo demanda: El consumidor tiene acceso a capacidades en la nube de forma automática en la medida que se requiera, de forma que autoescala servicios sin

la necesidad de interacción humana.

- **Amplio acceso a la red:** Los servicios que se proveen deben ser accesibles a través de internet, que promuevan el acceso desde una gama amplia de dispositivos de usuarios, desde smartphones hasta estaciones de trabajo.
- **Recursos compartidos:** Los recursos informáticos del proveedor se agrupan para servir a múltiples consumidores, con diferentes recursos físicos y virtuales, de forma que van asignando recursos dinámicamente en función de la demanda de los usuarios, existiendo una percepción de independencia porque los usuarios desconocen la ubicación exacta de los recursos proporcionados, solamente pueden tener conocimiento del centro de datos y de la zona de disponibilidad.
- **Elasticidad:** Las capacidades informáticas pueden aprovisionarse y liberarse rápidamente, muchas veces de forma automática, de forma que los usuarios tienen los recursos necesarios, siempre disponibles.
- **Servicio Medido:** Los proveedores deben dar la medición de los servicios efectivamente entregados de los servicios, para la supervisión, control y notificación, dando transparencia al usuario de lo consumido en cada uno de los servicios administrados.

Los modelos de informática en la nube ofrecen niveles de control, flexibilidad y administración con diferentes características de implementación, estos modelos son:

- Software como Servicio (SaaS): Proporciona un producto completo que el proveedor del servicio ejecuta y administra, son las aplicaciones que van dirigidas al usuario final, un ejemplo claro es el correo electrónico, no se debe pensar en tener en cuenta la administración y componentes subyacentes, sólo en el uso que se va a dar a la solución.
- Plataforma como Servicio (PaaS): Elimina la necesidad de administrar la infraestructura de una solución, como redes y hardware, solo concentrándose en el desarrollo consumiendo solo los servicios creados usando lenguajes de programación, bibliotecas y herramientas que suministre el proveedor cloud, solo siendo responsable por las aplicaciones alojadas y algunas configuraciones de alojamiento de la solución.
- Infraestructura como servicio (IaaS): Permite la creación de recursos TI esenciales, redes, procesamiento, almacenamiento, donde el consumidor ejecuta cualquier tipo de software, dando la posibilidad de implementar cualquier tipo de sistema operativo requerido y cualquier aplicación.

La Nist define modelos de implementación en la nube, precisando que la ubicación de los servicios implementados y quien los administra, siendo estos:

- Nube Privada: Es toda infraestructura que se aprovisiona para el uso exclusivo de una sola organización, que aun siendo para múltiples consumidores, puede ser propiedad de la organización, administrada por un tercero o una combinación de ellos, estableciéndose en la mayoría de los casos, donde la organización la requiera físicamente.

- Nube Pública: No se posee ningún tipo de hardware, todos los recursos los proporciona un proveedor de servicios en la nube, accesible para el público en general, sólo requiere tener acceso a internet y una cuenta de la plataforma de computación en la nube, tal como Amazon Web Services (AWS), Microsoft Azure, IBM Cloud o Google Cloud Platform (GCP).

Este modelo al estar disponible para cualquier tipo de consumidor u organización, los recursos son compartidos, establecidos en las instalaciones del proveedor de la nube.

Los proveedores cloud ofrecen muchos servicios de servidores, además de almacenamiento y servicios de virtualización simples.

- Nube Híbrida: Es la combinación de dos o más clouds individuales, pudiendo ser propias, compartidas o en la nube pública, compartiendo recursos y datos, muchas organizaciones cuentan con hardware propio, conectándose con la nube pública, funcionando como un solo sistema, útil en procesos de migración completa a nube pública.

La computación en la nube reúne múltiples tecnologías, creando un ecosistema técnico que trae varias características únicas que usando los avances en procesadores, tecnologías de virtualización, almacenamiento con unidades de estado sólido, conexión a internet de banda ancha, con conexión a servidores rápidos y baratos se convirtió en un solución indispensable para cualquier organización.

Con todas las características que ofrece la computación en la nube, el uso de esta tecnología ofrece una ventajas claves que la hace muy competitiva:

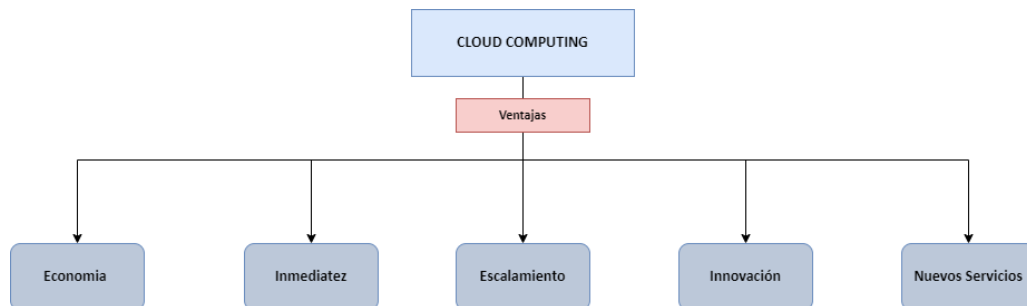


Figura 2. Ventajas Cloud Computing.
Creación Propia

La informática en la nube reduce de forma dramática los costos en implementación de infraestructura [4], que ayuda a la economía de muchas empresas, ayudando al crecimiento de empresas pequeñas que no tienen acceso a sistemas dedicados que solo tienen disponibles las grandes empresas, la implementación de infraestructura en la nube pública tiene un modelo que reduce costos, maximizando ganancias en base a la demanda de recursos, aprovisionando de forma dinámica los servicios, descartando de forma sencilla y rápida lo que no sea necesario para la operación.

La inmediatez de acceso a los servicios, de acuerdo a la documentación de Amazon [4] expone que la tecnologías cloud permiten acelerar los tiempos de comercialización y que los equipos TI reduzcan tiempos para implementación de servicios, por la facilidad de uso que permite en unos sencillos pasos aprovisionamiento y permite la customización necesaria para crear aplicaciones más robustas.

El escalamiento de los servicios implementados permite agregar o quitar servicios de cómputo, almacenamiento y redes de forma dinámica satisfaciendo la demanda de usuarios e

infraestructura en base a las cargas activas de trabajo, permitiendo equilibrar cargas sobre la marcha, ajustando los recursos de cómputo en base al uso de los usuarios y solicitudes de forma automática, de acuerdo a la documentación de la NIST.

Para muchas organizaciones la transformación e innovación digital es importante en las empresas por las mejoras que supone la digitalización de la información, dando versatilidad a los procesos misionales, además, las tecnologías cloud aceleran el cambio a nuevos modelos de negocios, eliminando las barreras del acceso a nuevas tecnologías, dando paso a la prestación de nuevos servicios que antes no eran posibles, que respondan en tiempo real a la demanda de los usuarios, a través de sistemas automáticos.

2.2 APLICACIONES BANCARIAS.

La banca digital ha tenido un gran auge a nivel global, entre el primer trimestre del año 2019 y el primer trimestre del año 2021 las instalaciones de aplicaciones financieras llegaron a los 4.7B [5]. Entre los mercados con mayor potencial en aumento de descargas se encuentra latinoamérica, este mercado tiene ciertas características haciendo que estas aplicaciones tengan una demanda creciente, teniendo un aumento de descargas de apps bancarias del 80% entre el año 2020 y 2021. El crecimiento en la demanda se debe en gran medida a la población no bancarizada, personas que no tienen participación financiera, existiendo varios factores para eso, como bajos ingresos, ser inmigrantes o municipios con poca presencia de entidades bancarias [6].

La inclusión financiera a través de plataformas digitales son unos los principales beneficios de las soluciones bancarias a través de dispositivos móviles, sin limitarse a un espacio físico, eliminando las barreras físicas y solo con acceso a internet, se da una mayor cobertura financiera, impulsando el autoservicio para un conjunto de operaciones financieras, ofreciendo el mayor número de modos de pago, siempre dando prioridad a las transacciones más comunes, la que los usuarios usen más, dando un dinamismo a la economía, además proporcionando al cliente una nueva independencia.

Las plataformas bancarias, funcionando como un mercado digital que brinda servicios bancarios y no bancarios de múltiples alianzas y de independientes, para que los usuarios cambien remesas, hagan pagos de facturas y tarjetas, transferencias inmediatas, transacciones a otros bancos, pagos en plataformas digital, entre otros servicios financieros, funcionando a través de una API, la cual es la interfaz que define la comunicación entre sistemas mediante solicitudes y respuestas del sistema [7].

De esta forma los usuarios ya no buscan a la banca tradicional que sólo les pocas opciones en sus operaciones financieras, solo con los servicios que la entidad ofrezca.



Figura 3. Banca tradicional.
Creación propia

La tendencia de los usuarios es a escoger soluciones donde tengan la posibilidad de elegir los servicios ofrecidos por diferentes instituciones y terceros, ya no siendo una relación uno a uno, transformándose en una relación uno a muchos, habilitadas por el banco líder de preferencia.

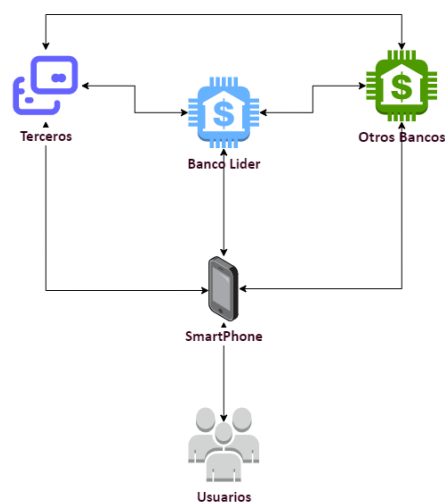


Figura 4. Tendencias de usuarios para soluciones financieras.
Elaboración propia.

Con esta tendencia de uso es necesario una plataforma robusta que tenga todas estas características, debe haber una coherencia Holística, donde los sistemas deben funcionar como un todo, priorizando la escalabilidad, disponibilidad y rendimiento del software, donde los equipos TI puedan implementar con facilidad los servicios de terceros.

La escalabilidad siendo fundamente por el grado atención a usuarios, prestando servicios a los usuarios mientras el número de clientes y transaccionalidad va en crecimiento, ante una mayor demanda en los recursos, se deben asignar más recursos para evitar el colapso del sistema, dando seguridad a los usuarios, incentivando el uso continuo de los servicios.

Una solución con altos porcentajes de disponibilidad para las entidades financieras es esencial para brindar servicios sin demora, buscando la modularidad de los servicios transaccionales para que frente a cualquier falla los usuarios puedan interactuar con sus productos, evitando indisponibilidades totales [8].

Para las entidades financieras es importante que nuevas funcionalidades no impliquen un proceso tardado, complejo volviendo a implementar la plataforma de software nuevamente, pudiendo implementar de forma independiente se ahorran costos y tiempo que los equipos TI pueden ocupar mejorando los servicios, no resolviendo problemas.

2.3 PROVEEDORES CLOUD.

Las soluciones financieras requieren un ecosistema que les brinde las mejores características en la implementación de servicios cloud, existiendo múltiples proveedores de servicios cloud es necesario, identificar los líderes del mercado, que brinden ofertas estandarizadas, con servicios de virtualización y automatización, con servicios adecuados para la adopción de estrategias innovadoras, teniendo los servicios necesarios para una alta gama de casos de uso, sobresaliendo en muchos de ellos. Se tiene como referencia el cuadrante mágico de Gartner, siendo Gartner una de las consultoras más importantes a nivel global en la investigación y análisis de las tendencias del mercado tecnológico. Analizar el mercado de servicios de infraestructura, con sus puntos fuertes y precauciones, ayuda a los equipos TI a escoger en base a sus necesidades.

El cuadrante mágico de Gartner a culminación de la investigación del mercado, mostrando como principales líderes y visionarios de plataformas de nube pública, teniendo como líder principal a Amazon con su colección de servicios Amazon Web Services, Microsoft con Azure y Google con los recursos de computación Google Cloud Platform [9].



Figura 5. Cuadrante mágico de Gartner - Proveedores cloud.

Fuente: Gartner Inc, 2021, Magic Quadrant, Grafica, Recuperado de: https://www.gartner.com/technology/media-products/reprints/AWS/images/736363_ESXM.jpg

Gartner define a Amazon Web Services (AWS) como el líder del mercado por su diversidad de servicios TI provisionales, dando posibilidad a cualquier persona de tener recursos informáticos solo creando una cuenta en la plataforma, impulsando la creación de aplicaciones nativas a empresas emergentes. Teniendo presencia en múltiples regiones y varias zonas de disponibilidad de forma aislada, garantizan alta disponibilidad pudiendo dividir las aplicaciones en diferentes zonas.

AWS tiene compromisos financieros importantes con las empresas que acuerden el uso de los servicios informáticos en un periodo de tiempo de uno a tres años, manteniendo precios flexibles, con precios bajos y planes de ahorro, ofreciendo hasta un 72% de ahorro en los servicios bajo demanda.

La innovación es muy importante para AWS, vital para el desarrollo económico de las empresas, transformando la cadena de suministros con el fin de ofrecer nuevos servicios, basados en las nuevas formas de desarrollo y tendencias de uso, impulsando los sistemas automatizados y virtualización.

Microsoft es líder del cuadrante mágico en el estudio de mercado [9], fuerte en muchos casos de uso, con una nube pública muy amplia en su plataforma Azure, enfocada en las empresas que se centran en servicios Microsoft, teniendo varios puntos fuertes en su operación, prestando múltiples servicios empresariales, teniendo un gran atractivo por sus herramientas para desarrolladores como Visual Studio y Github.

La solidez que mostró Microsoft da una gran confianza a los usuarios para usar los servicios administrados, además de las inversiones en bases de datos y soluciones para machine learning, muestra la gran capacidad de la plataforma, sin embargo hay varias preocupaciones para la plataforma de Azure, la capacidad de recuperación de servicios críticos es muy limitada,

mostrando que algunas zonas de disponibilidad no están equipadas con procedimientos de recuperación ante desastres.

Azure aún cuenta con una alta complejidad técnica con gestión de cuentas compleja que impide el crecimiento empresarial, los costos impiden economizar los gastos totales, sin planes de ahorros los usuarios ven desigualdad en facturaciones totales en el consumo de servicios, alejando a nuevas posibles organizaciones.

El tercer líder del cuadrante que señaló Gartner [9] es google con la plataforma Google Cloud Platform (GCP) es un proveedor fuerte con las capacidades IaaS y PaaS que puede ofrecer para los equipos TI que buscan agilidad en producción de aplicaciones comerciales, muchas de ellas, empresas con tendencias de crecimiento, Las operaciones tienen una disponibilidad geográfica amplia, aumentando su presencia en el mercado, viendo significativamente el aumento de adopción de los servicios de google.

GCP es una plataforma aún en etapa de maduración, por lo que muchas empresas evidencian experiencias deficientes en soporte de soluciones dedicadas de la plataforma.

Financieramente la plataforma opera con pérdidas monetarias al ser una unidad menor de la empresa matriz, impactando en las ayudas financieras que reciben los clientes al usar la plataforma, viendo disminuidos los incentivos y ahorro para las empresas, además del aumento de costos asociado con el aumento de clientes, limitando el crecimiento de las empresas que quieran usar los servicios de Google.

2.4 VIRTUALIZACIÓN.

Una gran tecnología que agiliza procesos en los equipos TI es la virtualización, el cual a través de software simula uno o varios entornos informáticos (procesadores, memoria, almacenamiento y más) en un host físico, dando la capacidad de crear varias máquinas con diferentes sistemas operativos y diferentes aplicaciones, teniendo independencia de los demás sistemas virtualizados, otorgando la capacidad de administración avanzada al ser máquinas definidas por software, creando políticas y flujos de trabajo de forma más ágil. Para una administración simplificada, el hypervisor o el monitor de máquina virtual es el que coordina los sistemas invitados [10].

Las propiedades de los sistemas virtualizados son [11]:

- Particionamiento:
 - Permite ejecutar múltiples sistemas operativos en una sola máquina física o servidor.
 - Necesidad de adquirir equipos físicos reducidos.
- Aislamiento:
 - Las máquinas virtuales pueden acceder a los recursos físicos compartiendo recursos del servidor o equipo, siendo completamente independientes uno del otro.
 - Separa los entornos, aislando la seguridad, con facilidad de recuperación ante desastres, si una máquina falla no cambia el estado de las demás máquinas que estén corriendo en paralelo.

- Encapsulación:
 - Condensa una gran cantidad de recursos virtuales, incluyendo aplicaciones.
 - Captura el estado de la máquina en un archivo, haciéndola portable y de fácil administración.
 - Permite compartir o mover fácilmente todo el sistema capturado de un sistema a otro, como cualquier otro archivo, archivo o documento.
- Independencia:
 - Las máquinas virtuales son configuradas con componentes virtuales, como CPU, memoria, tarjeta de red, completamente independientes del hardware físico.
 - Al no depender de un hardware, permite aprovisionarse o migrar a cualquier servidor host.

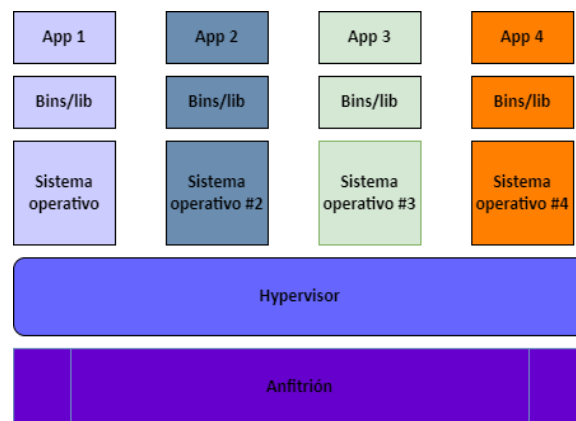


Figura 6. Virtualización.
Creación Propia.

2.4.1 CONTENEDORES.

La tecnología de contenedores es una forma de virtualización, de acuerdo a Docker, Inc un contenedor se define como:

“Un contenedor es una unidad estándar de software que empaqueta el código y todas sus dependencias para que la aplicación se ejecute de manera rápida y confiable de un entorno informático a otro. Una imagen de contenedor de Docker es un paquete de software ligero, independiente y ejecutable que incluye todo lo necesario para ejecutar una aplicación: código, tiempo de ejecución, herramientas del sistema, bibliotecas del sistema y configuraciones.” [12]

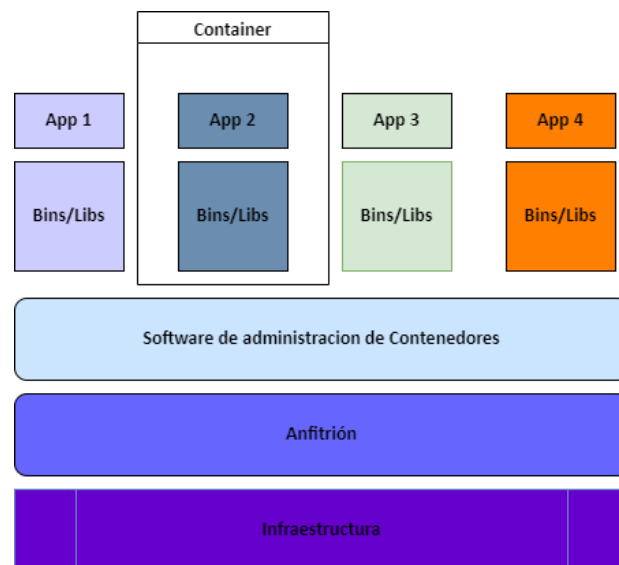


Figura 7. Tecnología de contenedores.
Creación Propia

Es un instrumento para almacenar un ítem, conteniendo aplicaciones, empaquetando a las aplicaciones un entorno que solo ejecuta las dependencias necesarias, aislada de programas y procesos relevantes que tiene un sistema operativo, proporcionando eficiencia operativa a las

aplicaciones contenidas en contenedores, dando la posibilidad de dividir toda una aplicación compleja en múltiples módulos, creando diferentes contenedores para cada uno de los módulos. Surgiendo en base a las necesidades de los equipos de desarrollo, donde los tiempos de migración presentaban problemas entre los entornos de pruebas y de producción que no son iguales en la mayoría de los casos, los contenedores solo encapsulan las librerías, archivos, archivos binarios, y dependencias indispensables para la ejecución de la aplicación.

El uso de contenedores es una forma ágil, simple al momento de crear, probar, implementar y migrar aplicaciones, al tener un tamaño reducido una gran variedad de ambientes son compatibles con esta tecnología, aportando agilidad y facilidad en la gestión del ciclo de vida de cualquier aplicación.

Se detallarán algunas diferencias entre la virtualización y contenerización [13].

Característica	Contenedores	Máquinas Virtuales
SO Anfitrión	Todos los invitados comparten el mismo sistema operativo y kernel. La imagen de kernel se carga en la memoria física.	Cada Máquina se ejecuta en un Hardware virtual y el kernel es cargado en su propia región de memoria
Comunicación	Mecanismos estándar de comunicación a través de los espacios de memoria compartidos, sockets.	Conectividad a través de dispositivos Ethernet
Seguridad	A través del Control de acceso	Depende del hypervisor

	obligatorio se puede asignar un nivel de seguridad a cada objeto.	
Desempeño	Rendimiento optimizado al requerir menos espacio que una máquina virtual.	Las máquinas se ven a menudo sobrecargadas por las instrucciones de la máquina que se traducen del sistema operativo huésped al anfitrión.
Aislamiento	Los subdirectorios pueden compartirse entre contenedores.	Compartir archivos, librerías y datos entre máquinas no es posible.
Tiempos de arranque	Pueden ser creados e iniciados en pocos segundos.	Los huéspedes necesitan de varios minutos para iniciar.
Almacenamiento	Son extremadamente ligeros, ocupan menos espacio ya que el sistema operativo base es compartido.	Ocupan mucho más espacio de almacenamiento, teniendo que instalar y ejecutar todo proceso del sistema operativo, con sus programas.

Tabla 1. Comparación entre contenedores y máquinas virtuales.

Los Casos de uso más habituales en las que los equipos TI usan los contenedores son:

- **Arquitectura de microservicios:** La arquitectura de microservicios descompone grandes aplicaciones en unidades funcionales más pequeñas, delimitadas, implementados por servicios autónomos, acoplados y desplegados de forma independiente.

- Multinube: El uso de contenedores facilita la implementación de aplicaciones en diferentes entornos de nube, por la naturaleza portátil pueden ser migrados entre nubes públicas o de forma local a la nube, y viceversa, empaquetar la aplicación en contenedores facilita los procesos de migración entre tecnologías.
- Multitenencia: Los contenedores facilitan alojar servicios multiusuarios, siendo una tecnología liviana y rápida, los servicios iniciarán en pocos segundos.
- DevOps: Los marcos de trabajo en desarrollo de aplicaciones se ven beneficiadas por el uso de contenedores al ser prácticos, los desarrolladores no pierden tiempo en configuraciones, permitiendo trasladar los esfuerzos al desarrollo, para las diferentes pruebas, se pueden establecer entornos de prueba y laboratorio rápidamente.

2.4.2 MICROSERVICIOS.

De acuerdo a [14] que citó a [15] la arquitectura de microservicios es un cambio hacia la transformación de servicios/componentes en microservicios: servicios más detallados y autónomos que aíslan funcionalidades comerciales detalladas por límites e interactúan a través de interfaces estandarizadas.

La adopción de microservicios ha aumentado rápidamente, muchos sistemas distribuidos basados en la nube han transformado sus arquitecturas tradicionales a nuevas arquitecturas con marcos de trabajo mejorados y agilizando el lanzamiento de las soluciones.

Las aplicaciones empresariales están usualmente construidas bajo un modelo de tres niveles, la interfaz de usuarios que se ejecuta en la máquina del usuario, una aplicación que gestiona las solicitudes HTTP que ejecuta la lógica de la solución, la base de datos, que recupera y modifica los datos en las tablas. La aplicación del lado del servidor es construida de tal forma que alberga todos los servicios de la solución en una sola unidad, funcionando como un todo, presentando problemas cuando la solución cambia de tamaño y se requiere modificar el código fuente de la aplicación, añadiendo complejidad en cada actualización, dando como resultado que la solución se comporte de manera inesperada, dificultando el análisis de errores, adicionalmente dado el caso de necesitar detener e iniciar procesos o se presenta algún incidente, toda la funcionalidad y los usuarios se verán afectados.

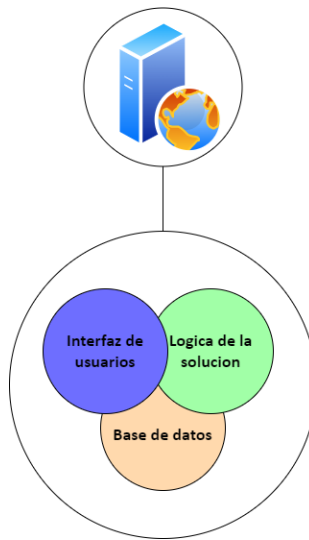


Figura 8. Soluciones empresariales tradicionales.
Creación Propia

Los microservicios presentan un enfoque diferente en la arquitectura de las aplicaciones modernas, intentando abordar los retos y oportunidades nuevos que tienen los equipos en el desarrollo de software.

En las arquitecturas orientadas a servicios, las funcionalidades son autónomas y autosuficientes, invocados a través de la red, produciendo el mismo resultado frente a peticiones realizadas de forma repetida, con una arquitectura de microservicios, una aplicación es separada en componentes independientes, que ejecuta cada funcionalidad de la aplicación como un servicio, a través del sistema enrutador hacia un grupo de servicios o interfaces de terceros (API) , cada servicio realiza una función específica y al funcionar de forma independiente, los aprovisionamientos, actualizaciones, despliegues y escalamiento de los microservicios puede hacerse de forma sencilla, sin necesidad de tiempos de inactividad, ni de reimplementación de todo el sistema.

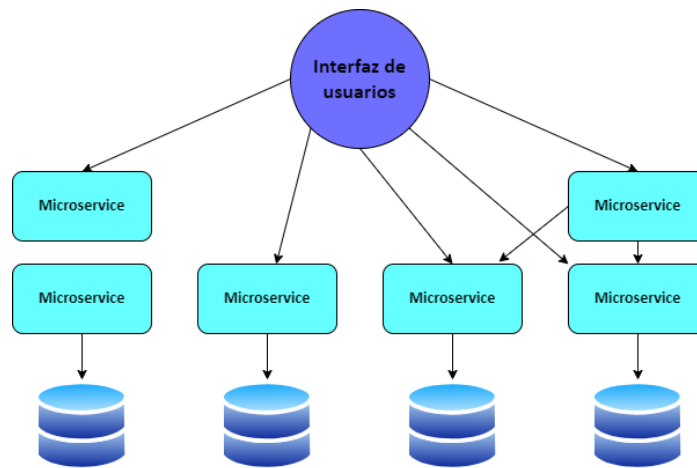


Figura 9. Arquitectura de microservicios.
Creación propia.

Características de los microservicios:

- Autonomía: Cada servicio funciona de forma independiente, totalmente responsable de la ejecución de un proceso de negocio determinado, sin afectar otros procesos, en muchos de los casos no comparten código, pudiendo ser implementados en múltiples lenguajes, los servicios no son dependientes de otros servicios, la comunicación entre componentes se realiza a través de una API definida.
- Especializados: Cada servicio está diseñado con ciertas capacidades de tal forma que cumple con las necesidades específicas de la solución, entre más código se vaya agregando a lo largo del tiempo, la complejidad aumenta, los microservicios dan solución, creando servicios más pequeños, pudiendo dividir en unidades más pequeñas los servicios con facilidad, si es necesario.

Las ventajas de los microservicios son:

- Tolerancia a fallos: a través de redundancia física, teniendo múltiples servidores con el servicio se mantiene consistencia en la solución, si se presenta un error en toda la solución, la continuidad del negocio se mantiene degradando la funcionalidad sin causar indisponibilidades totales a los usuarios, hasta encontrar el error y solventarlo.
- Agilidad: Los microservicios ayudan a los equipos de desarrollo pequeños, centrándose en partes pequeñas que son entregadas en menos tiempo, aumentando el rendimiento de la organización, ya que los requisitos y soluciones evolucionan con el tiempo, a través de una implementación sencilla, permitiendo probar ideas y revertirlas si no funciona correctamente.
- Libertad tecnológica: La arquitectura de microservicios no sigue un diseño único, da la posibilidad de unir todo tipo herramientas y enfoques, funcionando correctamente.

2.5 SERVICIOS ADMINISTRADOS PARA MICROSERVICIOS.

Para la implementación de Microservicios a través de contenedores existen diferentes servicios y orquestadores para la ejecución, gestión y monitorización completa de estos, que contengan dependencias, herramientas, archivos de soporte, ajustable a la necesidad de desarrollo, con características de automatización, los principales siendo:

2.5.1. DOCKER SWARM.

Es la agrupación de contenedores ejecutadas en la aplicación de Docker, permite gestionar contenedores desplegadas en múltiples anfitriones, uniéndolas en un Cluster, otorga alto un alto nivel de disponibilidad al disponer de un gestor que controla los recursos de los contenedores de forma que trabajen de forma eficiente[16].

2.5.2 AMAZON ELASTIC KUBERNETES SERVICE (EKS).

Es el servicio de orquestación de contenedores, que facilita la ejecución de Kubernetes administrados para automatizar la implementación, escalado y administración de contenedores lista para usar en la nube pública de Amazon, ofrece una gestión dinámica de recursos, creciendo en recursos cuando sea necesario y reduciendo en igual medida, cuando ya no se necesite, evitando el desperdicio, ahorrando costos de utilización.

Amazon EKS administra de forma automática la disponibilidad y el escalamiento de las máquinas a través del controlador, el cual es responsable de tareas clave como la gestión de los

contenedores, administración de disponibilidad de las aplicaciones contenidas y control del almacenamiento de datos del clúster[17].

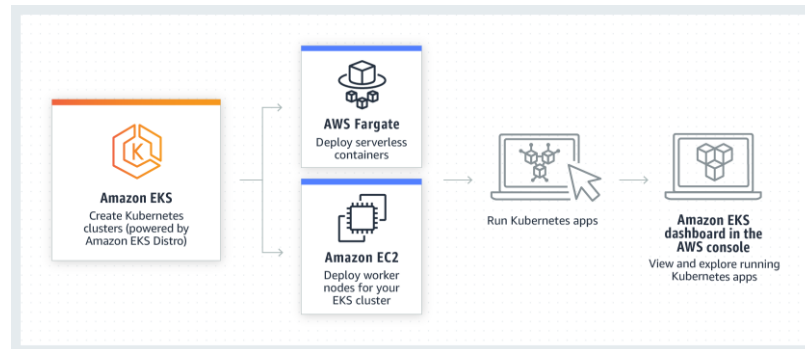


Figura 10. Amazon EKS.

Fuente: Amazon Web Services, Diagrama Amazon EKS, Ilustración, Recuperado de: https://d1.awsstatic.com/product-page-diagram_Amazon-EKS%402x.ddc48a43756bff3baead68406d3cac88b4151a7e.ddc48a43756bff3baead68406d3cac88b4151a7e.png

Las características de los clusters administrados por kubernetes descritos en la documentación de AWS son [17]:

- Plano de control de Kubernetes:

Es el conjunto de componentes y procesos, administra como kubernetes se comunica con el cluster, manteniendo todos los registros e instrucciones que se están ejecutando en las máquinas, EKS garantiza una alta disponibilidad, ejecutando el plano en tres zonas de disponibilidad, detectando y reemplazando de forma automática las máquinas que presenten errores.

- Integración de servicios:

A través de la herramienta AWS Controllers for Kubernetes permite administrar los servicios de AWS desde kubernetes, simplifica la creación de aplicaciones escalables,

con alta disponibilidad en los servicios de AWS.

- Consola de Kubernetes alojada:

EKS proporciona una consola unificada para desarrolladores y administradores, siendo el lugar ideal para organizar, visualizar y solucionar problemas de las aplicaciones alojadas.

2.5.3 ELASTIC CONTAINER SERVICE (ECS).

Es el servicio de orquestación de contenedores compatible con Docker, a través de una consola simplificada que permite administrar, escalar e implementar cargas de trabajo de forma dinámica, integrando herramientas de automatización.

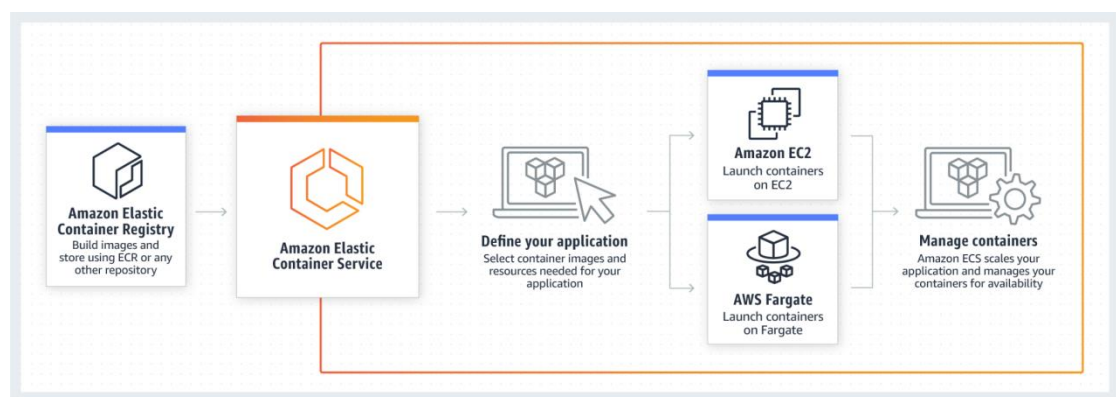


Figura 11. Amazon ECS.

Fuente: Amazon Web Services, Diagrama Amazon ECS, Ilustración, Recuperado de: https://d1.awsstatic.com/product-page-diagram_Amazon-ECS%402x.0d872eb6fb782ddc733a27d2bb9db795fed71185.png

ECS posee ciertas características en la administración de contenedores que hace a la plataforma implementar servicios lo más simple posible [18].

- Gestión de clústeres:

El servicio instala, opera y escala los servicios informáticos de forma automática, solo requiriendo lanzar el grupo de instancias de contenedores, especificando las tareas que van a realizar. Puede cargar una versión nueva de la definición de tareas de su aplicación y el programador de Amazon ECS iniciará contenedores nuevos automáticamente a partir de la imagen actualizada, y detendrá los contenedores que ejecuten una versión anterior, actualizando a la versión más reciente.

- Docker:

ECS ofrece soporte nativo para Docker, otorgando características de administración de contenedores Docker en un cluster de instancias EC2, para que pueda definir y ejecutar aplicaciones con varios contenedores. Las aplicaciones que se empaquetan localmente como un contenedor se implementarán y ejecutarán en Amazon ECS sin la necesidad de realizar ningún cambio de configuración.

- Monitoreo:

ECS proporciona funciones de monitoreo para cada uno de los contenedores y clusters a través de Amazon Cloudwatch, monitoreando usos promedios y consumos de CPU y memoria durante la ejecución de tareas, avisando cuando es necesario escalar vertical o horizontalmente, de forma adicional a través de los registros de los contenedores, se permite realizar un diagnóstico rápido y óptimo.

2.5.4 AWS FARGATE.

Es un entorno Serverless, sin servidor, compatible con EKS y ECS, centrado en la creación de aplicaciones sin preocuparse por el aprovisionamiento de servidores, completamente administrado por AWS. Al ejecutar las tareas y los servicios de Amazon ECS con el tipo de lanzamiento de Fargate o un proveedor de capacidad de Fargate, la aplicación se empaqueta en contenedores, se especifican los requisitos de sistema operativo, CPU y de memoria, se definen las políticas de IAM y las redes, y se lanza la aplicación. Cada tarea de Fargate tiene su propio límite de aislamiento y no comparte el kernel subyacente, los recursos de CPU, los recursos de memoria ni la interfaz de red elástica con otra tarea.[19]

2.5.5. ELASTIC COMPUTE CLOUD.

Amazon EC2 es la parte central de los servicios de la nube pública de Amazon, otorgando las capacidades informáticas necesarias y redimensionables para el desarrollo de aplicaciones modernas. Bajo el modelo de pago bajo demanda, optimiza el costo, facturando por segundo usando, permitiendo centrarse en mejorar los servicios y no maximizando el uso para ciertas horas.



Figura 12. Amazon EC2

Fuente: Amazon Web Services, Diagrama Amazon EC2, Ilustración, Recuperado de: https://d1.awsstatic.com/products/EC2/Spot/product-page-diagram_EC2-Spot-Instances.6c3c51f4c6a28cd71d8fef8231510b5619e84eea.png

A través de Amazon EC2 Auto Scaling se mejora la disponibilidad de las aplicaciones, permitiendo detectar errores, finalizar y lanzar una instancia de reemplazo; Garantiza que siempre se disponga de los recursos informáticos adecuados y añade instancias sólo cuando la infraestructura la necesita para su funcionamiento normal.

EC2 permite aumentar o disminuir la capacidad en minutos. Puede crear las instancias necesarias, una o miles de forma simultánea, controlado a través de la API.

Amazon EC2 trabaja en conjunto con Amazon VPC para brindar seguridad y una sólida funcionalidad de red para sus recursos informáticos. Sus instancias informáticas están ubicadas en una nube privada virtual (VPC) con un rango de IP que usted especifica. los equipos deciden qué instancias están expuestas a Internet y cuáles permanecen privadas.

- Los grupos de seguridad y las ACL de redes permiten controlar el acceso a la red entrante y saliente hacia y desde sus instancias.
- Puede aprovisionar sus recursos de EC2 como instancias dedicadas. Las instancias dedicadas son instancias de Amazon EC2 que se ejecutan en hardware dedicado a un solo cliente para un aislamiento adicional.
- Si no tiene una VPC predeterminada, debe crear una VPC y lanzar instancias en esa VPC para aprovechar funciones de red avanzadas, como subredes privadas, filtrado de grupos de seguridad de salida, ACL de red e instancias dedicadas. [20]

CAPÍTULO 3.

INFRAESTRUCTURA PROPUESTA.

Amazon Web Services es el proveedor más relevante, otorga una gran variedad de servicios informáticos potentes para el desarrollo de aplicaciones nativas en la nube, ofrece accesibilidad para que cualquier persona o nuevas organizaciones empiecen a desarrollar aplicaciones sin límites de accesibilidad a nuevas tecnologías, AWS facilita múltiples servicios de contenedores totalmente administrados, Siendo los servicios ECS y EKS los que admiten una amplia gama de servicios informáticos relacionados.

Las soluciones bancarias han experimentado una evolución acelerada hacia la digitalización, acercando los servicios bancarios a todo tipo de usuarios, facilitando el acceso a todo tipo de servicios, dando como resultado que la población de bancarizada disminuya, con un enfoque más abierto, siguiendo las tendencias del mercado con el fin de ofrecer y distribuir servicios que los usuarios están solicitando.

Es probable que esta nueva frontera se abra traspasada por la plataforma bancaria, una fusión habilitada por la tecnología de la banca tradicional y digital, fintech y terceros. La banca de plataforma cambia el modelo bancario tradicional por uno centrado en el cliente que ofrece un mercado de productos y servicios financieros provenientes de instituciones múltiples e independientes. [8]

De acuerdo a la hoja de ruta [8] para la mayoría de los bancos, la adopción exitosa de los estándares bancarios de plataforma requerirá nuevas estrategias y tecnologías, con el fin de reemplazar la arquitectura e infraestructura de la aplicación bancaria central actual. También requerirá una transición en toda la empresa hacia una arquitectura basada en microservicios, que es un habilitador crítico que permite una integración eficiente y acelerada con terceros,

que puede convertirse en el principal diferenciador competitivo en el ecosistema bancario de plataforma.

Para la implementación de microservicios es necesario decidir cuál es el servicio de contenedores que de las mejores prestaciones para la implementación de aplicaciones nativas en la nube pública.

Con un número mayor de contenedores y creación de estos en base a cómo va creciendo la solución, es requerido herramientas para la asignación de recursos, manteniendo normalidad en la transaccionalidad.

Con el fin de analizar el potencial de los orquestadores se crearon dos cluster, uno en el servicio de Kubernetes y uno en ECS, cada uno con dos instancias t2.micro de la capa gratuita, con un contenedor docker cada uno, para ver la facilidad de creación e implementación de microservicios con los conocimientos con los que una empresa emergente puede contar.

Se deben las instrucciones en la documentación de AWS para la implementación de los cluster, detectando la practicidad de implantación de servicios en la nube, para enfocarse en el desarrollo de aplicaciones que el principal atractivo que ofrece la computación en la nube.

ECS presenta mayor simplicidad para la implementación de contenedores a escala, lo que reduce los tiempos de creación e implementación de contenedores, facilita la toma de decisiones que los equipos deben tomar frente a las configuraciones informáticas, de red y de seguridad, sin verse perjudicado por el crecimiento vertical o horizontal, aun siendo una plataforma simplificada ofrece la potencia necesaria para que no se vea afectado el alcance de las soluciones, contando con un conjunto de servicios diseñados para administrar sistemas y

operaciones que le permitirán controlar los recursos de la infraestructura aplicando las directrices de gobernanza y cumplimiento apropiadas.

EKS ofrece toda la flexibilidad con la que kubernetes ya cuenta gracias a la comunidad que alimenta las herramientas y complementos de forma constante, haciéndola una plataforma potente para la implementación de microservicios, ofreciendo una plataforma administrada segura, confiable, escalable, con resistencia a cargas de trabajo pesado, fácilmente asignable a otros servicios del catálogo disponible por AWS, como las bases de datos RDS, esenciales en toda solución.

Sin embargo EKS requiere un nivel más alto de conocimiento para la implementación, por ciertos complementos que se requieren para habilitar servicios, como IAM, de igual forma para los equipos que necesitan implementar rápidamente servicios añade una capa de complejidad para los equipos que no están familiarizados con Kubernetes, aumentando los tiempos y costos. Usando ECS no se incurren en costos adicionales, con la que EKS sí cuenta, pagando 0,10 USD por hora, que puede ser una limitante para las empresas emergentes.

ECS proporciona una plataforma versátil para la implementación de microservicios, siendo más sencillo escalar y acelerar los procesos de desarrollo de cualquier tipo de aplicación, para servicios financieros, como una billetera digital, una arquitectura de microservicios funcional a través de ECS para servicios financieros es:

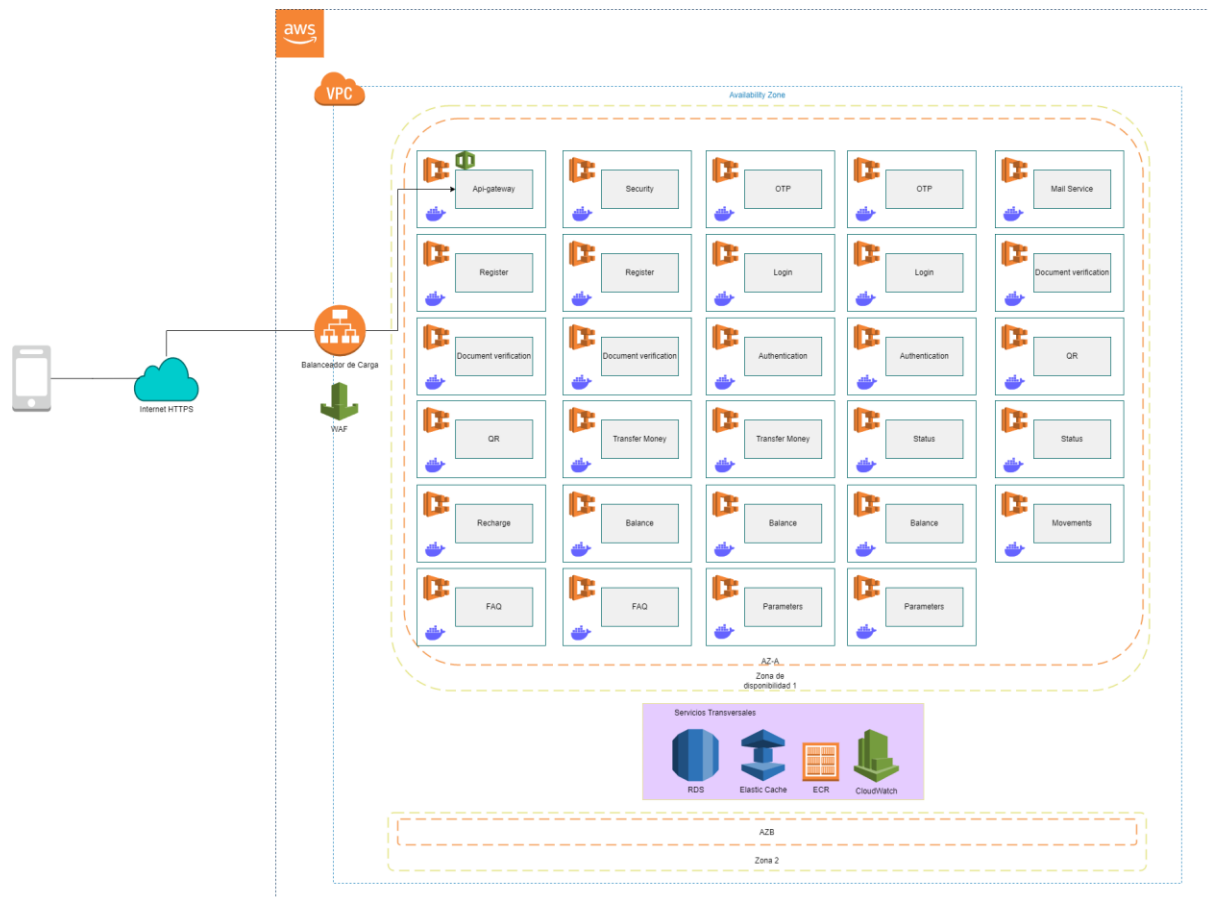


Figura 13. Arquitectura bancaria propuesta basada en microservicios.
Creación Propia.

En el diagrama anterior se pueden evidenciar múltiples microservicios con la misma funcionalidad, como “OTP” o “Login” se plantea con uno o más contenedores similares, por los beneficios que trae el uso de microservicios, la independencia y aislamiento entre servicios permite crear múltiples instancias con la misma funcionalidad en los servicios más críticos usando la escalabilidad que otorga EC2. Los servicios pueden verse con un consumo de CPU elevado que cree indisponibilidad en los servicios, la API basada en Websocket direcciona a una instancia con un menor consumo para mantener la continuidad de negocio de cara a cliente, no necesitando un orden establecido para el correcto funcionamiento de los microservicios empleando la funcionalidad de VPCLink que se puede crear para la conexión de entre cada una de las instancias, sin emplear una puerta de enlace de internet, un dispositivo de traducción de

direcciones de red, una dirección IP pública, para comunicarse con el servicio, el tráfico no sale de la red de Amazon. [21]

En la infraestructura propuesta las solicitudes HTTPS entrantes son comunicadas por la tecnología WebSocket hacia el balanceador de carga que se comunica con el Api-gateway que contiene la tecnología Api de WebSocket, construido con las plantillas que entrega CloudFormation, desempeñando como puerta de enlace a las tareas definidas que realizan los microservicios del clúster [22].

Descripción de los microservicios propuestos que son necesarios para un correcto funcionamiento de la solución:

El flujo de cliente empieza realizando cualquier ingreso a la solución, realizando una solicitud HTTPS, la que a través de una comunicación persistente a través de websocket el balanceador de carga de aplicaciones recibe la solicitud HTTPS actualizando la conexión a una WebSocket, redistribuyendo las solicitudes en base a solicitudes de tareas a una instancia designada para el api-gateway.

El uso del protocolo WebSocket es importante en el balanceo de la carga de solicitudes de la API, admitiendo la comunicación bidireccional entre las aplicaciones cliente y el backend de la billetera digital, abriendo una sesión entre usuario y servidor basado en eventos, sin tener que consultar al servidor para recibir una respuesta.

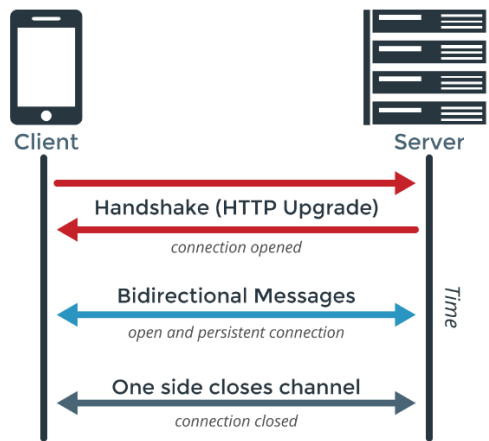


Figura 14. Protocolo WebSocket

Fuente: <https://www.dotcom-monitor.com/blog/es/2020/05/25/websocket-application-monitoring/>

Los clientes pueden enviar mensajes a un servicio y los servicios pueden enviar mensajes a los clientes de forma independiente, WebSocket es una conexión full-duplex, como se ve en el diagrama anterior, el cliente también puede enviar datos al servidor, lo que permite interacciones entre clientes y servicios que son aprovechadas para las arquitecturas de microservicios y las billeteras digitales sin requerir que los clientes realicen una solicitud explícita, los usuarios llevan a cabo varias transacciones en periodos cortos de tiempo, siendo necesario una conexión persistente.

Posterior a la creación de la conexión hacia el balanceador, la carga es distribuida de forma eficiente hacia el api gateway basado en WebSocket, usando la VPC Link creada se define a donde se envía el tráfico, en el caso de la propuesta, la API enviar las tareas al clúster creado en el servicio ECS de AWS, el cual contiene cada uno de los microservicios necesarios, iniciando el flujo en la instancia EC2 con funciones API, llamada Api-gateway siendo una integración privada, requiriendo lo anterior descrito. La integración sencilla de la API creada permite enlazar los destinos a través de la ip de la instancia.

Cada instancia EC2 usara imágenes AMI (imágenes de máquinas de Amazon), las cuales son plantillas configuradas con librerías y un sistema operativo preconfigurado basadas en Linux, para crear una aplicación basada en microservicios, los componentes de la arquitectura se

construirán bajo el framework spring boot, por la oportunidad que da para escalar aplicaciones simples, con capacidades reactivas, fáciles de aprender e implementar.

Con la comunicación establecida entre cliente y el backend de cada servicio, el usuario tiene la posibilidad de realizar múltiples transacciones, realizar un login simple a la solución, sería posible gracias al microservicio llamado “Login”, recibe las credenciales número, pudiendo ser un numero celular o identificación, que posteriormente se asignara en el XML como identificador, para realizar tareas en el contenedor de “Security” que se encarga de realizar de crear y validar las llaves, todo usuario realiza consultas del saldo disponible en la wallet, con una gran probabilidad de saturación de CPU o memoria, por la consultas masivas que se dan en ciertas horas y transaccionalidades que se pueden realizar, mostrando la importancia de crear varios contenedores con el mismo propósito, visualizando el saldo, los usuarios ven la disponibilidad para realizar pagos, el microservicio de Payment trae todas las urls a las que el servicio debe apuntar para consumir los servicios tanto los servicios internos o de aliados, para el consumo de bienes y servicios por parte de los clientes.

Una parte fundamental es el registro de nuevos usuarios para la fidelización de los usuarios, una aplicación con problemas de registro en el primer intento causa rechazo al momento de intentar nuevamente, el registro empieza en el servicio de “register”, ingresando los datos de la persona, validando que el usuario y número no exista en el servicio de “status”, si los condicionales que se establezcan en el servicio de “status” muestran que el usuario ya existe o presenta algún bloqueante como que el numero celular ya existe con otro usuario, el flujo muere en el servicio de register, mostrando un código de error asociado. Si el usuario en el registro no presenta bloqueantes el flujo puede continua en la verificación del email registrado, recibido y validado por el servicio de “Mail” el cual envía un correo al email guardado, posterior se debe validar el número ingresado, el servicio de “OTP” envía un SMS con una contraseña de un solo uso al celular registrado, con la validación del número sucede la verificación el

documento en el microservicio “Document verification” usando el SDK de AWS que trae las facilidades para usar los kits de desarrollos creados para el escaneo de documentos de identificación, terminando el flujo de registro en ese punto.

Descripción corta de los microservicios propuestos que son necesarios para un correcto funcionamiento de la solución:

- Security: Se definirán tareas en la generación e intercambio de llaves criptográficas para el establecimiento de las comunicaciones con los servicios.
- OTP: Cumple con las tareas de creación de códigos de autenticación de un solo uso.
- Mail Service: Validación de correo electrónico registrado.
- Authentication: Es el encargado de validar las credenciales de usuario.
- Register: cumple con las funciones de realizar el registro de usuarios
- Login: Recibe las credenciales de seguridad
- Document verification: Tiene la función de verificar el documento de identificación.
- QR: Generación de QR para pagos en datafono.
- Transfer Money: servicio encargado de las transferencias bancarias entre usuarios en la misma solución o terceros aliados.
- Status: Tiene la tarea almacenar y validar el estado de registro del usuario, que permite el acceso a la solución.
- Recharge: Permite la recarga por el método de preferencia del usuario.
- Balance: Acceso al saldo actual de usuario, permite realizar múltiples transacciones disponibles.
- Movements: Permite ver al usuario que movimientos financieros ha realizado.
- FAQ: Servicio de preguntas frecuentes.
- Parameter: Cuenta con los parámetros iniciales de la solución.

- Payments: Designación de tareas para el pago de facturas y otros recibos de terceros.

Existen múltiples Frameworks o marcos de trabajo para la implementación de microservicios tal como Spring boot o NodeJS a tener en cuenta para la creación de aplicaciones, la propuesta se va a limitar en la arquitectura teórica en servicios cloud, proponiendo funcionalidades generales de los microservicios.

CAPÍTULO 4.

CONCLUSIONES Y RECOMENDACIONES.

El enfoque innovador en los servicios de Amazon web services permite implementar aplicaciones nativas por los incentivos económicos y el soporte técnico que ofrece, sin embargo frente a las nuevas estrategias cloud, es necesario plantear la implementación de servicios en varios proveedores de nube pública, multi nube, que presenta varias ventajas, contribuyendo a una menor dependencia de un solo proveedor, teniendo alternativas frente a bloqueos del proveedor principal, el uso de contenedores beneficia estas implementaciones al ser una solución ligera y portable, siendo compatible con múltiples plataformas, facilitando el desacoplamiento en partes pequeñas, granulando la arquitectura en múltiples plataformas.

EKS presenta múltiples funciones importantes para la implementación de microservicios en contenedores por la comunidad que siempre esta actualizando kubernetes, para hacer de sus servicios lo más accesible, sin embargo, actualmente la sencillez de ECS, da más ventajas para la implementación de soluciones financieras, con una mayor compatibilidad de servicios de AWS.

Las plataformas bancarias tienen un camino desafiante para la implementación de plataformas bancarias, sin embargo AWS y la implementación de contenedores permiten concentrarse en la mejora de los servicios, permitiendo un crecimiento constante, con un papel activo en el mundo bancario, resultando en un experiencia de cliente excelente y eficiencia operativa mejorada.

Crear la infraestructura en otras zonas de disponibilidad permite que los niveles de disponibilidad mejoren, funcionando como contingencia frente a indisponibilidades masivas que pueda tener el proveedor.

ECS presenta mejor balance costo beneficio, el uso de un mes de las instancias t2.micro presento un costo de \$0 US, frente al servicio EKS que en el mismo rango de tiempo dio un costo elevado de \$70 US, solo tendiendo un cluster con dos instancias de la capa gratuita, corriendo un contenedor Docker cada una, mientras que ECS sólo registró un costo de \$11 US en el mes transcurrido, entre más crezca la solución, los costes se van a ver aumentados, lo que puede traer costos que una compañía emergente tendría problemas para costear, o se podría utilizar en mejoras en el equipo TI.

De acuerdo a la propuesta planteada existen muchos modos de implementar una arquitectura de microservicios, esta propuesta es solo un énfasis en las practicas mas relevantes para la implementación en un servicio de billetera digital, la infraestructura puede ser modificada en base a las necesidades y alcance de la solución, siendo flexible en los modos de implementación, siendo posible definir las comunicaciones en base a solicitudes HTTP y no en base a eventos, directamente se puede realizar la usando el balanceador de carga usando el servicio de AWS Api Gateway con REST Api.

Actualmente las implementaciones de funciones lambda son un gran atractivo para las organizaciones en la implementación de soluciones basada en microservicios, solo cargando el código en lambda y el servicio se encarga de ejecutar las funciones independientes al ser un entorno sin servidor, son llamadas por el servicio Api Gateway, solo enfocándose en el código sin aprovisionar infraestructura siendo un ahorro en costos para las organizaciones.

Para una arquitectura de microservicios la intervención de una API, funciona como integrador de los módulos separados, como las REST API, utilizando proxy o sin el para respuestas del backend del servicio.

BIBLIOGRAFÍA.

[1] Comparación entre la arquitectura monolítica y la arquitectura de microservicios. (n.d.).

Retrieved September 8, 2022, from <https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith>

[2] LATAM, tendencias que marcarán la migración a la nube en 2022 - Davinci

Technologies. (n.d.). Retrieved September 8, 2022, from <https://davinci.tech/blog/tendencias-migracion-nube-latam-2022/>

[3] Mell, P. M., & Grance, T. (2011). The NIST definition of cloud computing.

<https://doi.org/10.6028/NIST.SP.800-145>.

[4] AWS | Informática en la nube. Ventajas y Beneficios. (n.d.). Retrieved August 4, 2022,

from <https://aws.amazon.com/es/what-is-cloud-computing/>

[5] What Technologies are Driving the Digital Wallet Trend? | FinTech Magazine. (n.d.).

Retrieved August 5, 2022, from <https://fintechmagazine.com/digital-payments/what-is-driving-the-digital-wallet-revolution>

[6] Descarga de aplicaciones móviles de los bancos aumentó un 232% en el 2020 | El

Financiero. (n.d.). Retrieved August 5, 2022, from <https://www.elfinancierocr.com/tecnologia/descarga-de-aplicaciones-moviles-de-los-bancos/KV5JVCXO4NAXZDBB5JAKXXX7GE/story/>

[7] Bucchiarone, A., Kessler, B., Dragoni, N., Dustdar, S., Wien, T. U., Larsen, S. T., & Bank, D. (n.d.). *From Monolithic to Microservices An Experience Report from the Banking Domain*.

[8] *Opening banking through architecture re-engineering A microservices-based roadmap*. (n.d.).

[9] Bala, R., & Gill, B. (n.d.). *Market Definition/Description Magic Quadrant Licensed for Distribution Magic Quadrant for Cloud Infrastructure and Platform Services*.
<https://www.gartner.com/doc/reprints?id=1-271OE4VR&ct=210802&st=sb>

[10] *¿Qué es la virtualización? - Colombia | IBM*. (n.d.). Retrieved August 5, 2022, from
<https://www.ibm.com/co-es/cloud/learn/virtualization-a-complete-guide>

[11] Dua, R., Raja, A. R., & Kakadia, D. (2014). *Virtualization vs containerization to support PaaS*. *Proceedings - 2014 IEEE International Conference on Cloud Engineering, IC2E 2014*, 610–614. <https://doi.org/10.1109/IC2E.2014.41>

[12] *¿Qué es un Contenedor? - ventana acoplable*. (n.d.). Retrieved July 2, 2022, from
<https://www.docker.com/resources/what-container/>

[13] *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*. (n.d.). IEEE.

[14] Hassan, S., Ali, N., & Bahsoon, R. (2017). *Microservice Ambients: An Architectural Meta-Modelling Approach for Microservice Granularity*. *Proceedings - 2017 IEEE International Conference on Software Architecture, ICSA 2017*, 1–10. <https://doi.org/10.1109/ICSA.2017.32>

[15] S.Hassan and R.Bahsoon, “Microservices and their design trade-offs: A self-adaptive roadmap,” in *13th IEEE International Conference on Services Computing (SCC)*, San Francisco, USA, jun 2016.

[16] What is Docker Swarm? | Sumo Logic. (n.d.). Retrieved August 5, 2022, from <https://www.sumologic.com/glossary/docker-swarm/>

[17] Servicio administrado de Kubernetes - Características de Amazon EKS - Amazon Web Services. (n.d.). Retrieved August 5, 2022, from <https://aws.amazon.com/es/eks/features/>

[18] Orquestación de contenedores completamente administrada - Características de Amazon Elastic Container Service (Amazon ECS) - Amazon Web Services. (n.d.). Retrieved August 5, 2022, from <https://aws.amazon.com/es/ecs/features/>

[19] ¿Qué es AWS Fargate? - Amazon ECS. (n.d.). Retrieved July 10, 2022, from https://docs.aws.amazon.com/es_es/AmazonECS/latest/userguide/what-is-fargate.html

[20] Amazon EC2 | Amazon Elastic Compute Cloud | Amazon Web Services. (n.d.). Retrieved July 10, 2022, from <https://www.amazonaws.cn/en/ec2/>.

[21] Tutorial: Crear una API REST con la integración privada de API Gateway - Amazon API Gateway. (n.d.). Retrieved August 23, 2022, from https://docs.aws.amazon.com/es_es/apigateway/latest/developerguide/getting-started-with-private-integration.html.

[22] Automatice la creación de API de WebSocket en Amazon API Gateway con AWS CloudFormation. (n.d.). Retrieved August 23, 2022, from <https://aws.amazon.com/es/about-aws/whats-new/2019/02/automate-websocket-api-creation-in-api-gateway-with-cloudformation/>