



Planeación, modernización y migración de la infraestructura transaccional para la pasarela de pagos Placetopay

Valeria Granada Rodas

Informe presentado como requisito para optar al título de Ingeniero de sistemas

Modalidad

Práctica Empresarial

Asesor Interno

Oscar Camilo Lopera Lopera, Ingeniero de Sistemas

Asesor externo

Mario David Gonzalez Portillo, Ingeniero de Sistemas

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería de sistemas

Medellín

2025

- [1] V. Granada Rodas, "Planeación, modernización y migración de la infraestructura transaccional para la pasarela de pagos Placetopay", Ingeniería de sistemas UdeA, Universidad de Antioquia, Medellín, 2025.



Asesor interno UdeA: Oscar Camilo Lopera Lopera

Asesor externo Evertec inc: Mario David Gonzalez Portillo



Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes.

Decano/Director: Julio César Saldarriaga.

Jefe departamento: Danny Alejandro Múnera Ramírez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

A Dios, por ser mi fortaleza y guía. A mi familia, por ser el pilar fundamental en cada paso de mi vida, por su amor incondicional, paciencia y apoyo constante. A mis padres, por creer en mí y brindarme las herramientas necesarias para alcanzar mis sueños. A mis amigos, quienes con su compañía y aliento hicieron más llevadero este camino, y a quienes siempre estuvieron ahí para darme una palabra de ánimo o compartir una sonrisa en los momentos de dificultad.

Agradecimientos

Quiero expresar mi agradecimiento a Evertec Inc por brindarme la oportunidad de realizar mis prácticas académicas en un ambiente de constante aprendizaje y crecimiento profesional. A mi equipo de trabajo, gracias por su orientación, confianza y por permitirme formar parte de proyectos que contribuyen significativamente a mi desarrollo como ingeniera de sistemas. A mi líder técnico Mario Gonzalez Gracias por su guía a lo largo de este proceso, por impulsarme a mejorar constantemente y por su compromiso con mi crecimiento profesional. Su apoyo y disposición fueron clave no solo durante esta etapa, sino también a lo largo de mi carrera, y por ello les estaré siempre agradecida.

A mi asesor de prácticas, Oscar Camilo Lopera Lopera, le debo un profundo agradecimiento por su constante seguimiento y apoyo. Sus consejos, su disposición para resolver dudas y sus aportes fueron fundamentales para asegurar que este trabajo se realizara de manera correcta y con calidad.

También extiendo mi gratitud a la Universidad de Antioquia, por proporcionarme las herramientas necesarias y una formación integral que me permitió alcanzar esta meta, y a todas aquellas personas que, de una u otra manera, contribuyeron al éxito de este trabajo.

TABLA DE CONTENIDO

RESUMEN	6
ABSTRACT	7
I. INTRODUCCIÓN	8
II. OBJETIVOS	9
A. Objetivo general	9
B. Objetivos específicos	9
III. MARCO TEÓRICO	10
Limitaciones de los Servidores Tradicionales	10
Tipos de Servidores Tradicionales	11
Ventajas de la Modernización de Sistemas	12
Buenas Prácticas para la Modernización de Sistemas	13
Serverless	14
IV. METODOLOGÍA	14
Eventos Scrum:	14
Revisiones de Código:	15
Cronograma de actividades:	15
V. RESULTADOS	16
Arquitectura propuesta	20
Adopción de una infraestructura serverless	22
Comparación entre Proveedores de Nube	23
Criterios de comparación entre AWS, Azure y GCP	23
Arquitectura definida:	26
Implementación y Ejecución	29
Creación de la estructura serverless	30
Uso de Terraform para la gestión de infraestructura	30
Construcción de imágenes Docker	31
Balanceo de carga y autoscaling	31
Ejecución de comandos	32
Proceso de despliegue	32
Planificación y Ejecución de la Migración	33
Despliegue en un Ambiente de Testing Controlado	33
Verificación y Despliegue en Producción	34
Documentación y Capacitación	34
VI. ANÁLISIS	35
VII. CONCLUSIONES	36
REFERENCIAS	37
ANEXOS	39

LISTA DE TABLAS

TABLA I	
COMPARACIÓN DE SERVICIOS ON-PREMISE VS. SERVICIOS EN LA NUBE	17
TABLA II	
COMPONENTES DE LA NUEVA INFRAESTRUCTURA	21
TABLA III	
COMPARACIÓN ENTRE PROVEEDORES DE NUBE	23
TABLA IV	
COMPONENTES INFRAESTRUCTURA FINAL EN AWS	27

LISTA DE FIGURAS

Fig. 1. Cronograma de actividades	15
Fig. 2. Infraestructura antigua del aplicativo Placetopay	16
Fig. 3: Arquitectura en la nube propuesta	20
Fig. 4: Arquitectura propuesta en AWS	26
Fig. 5: Arquitectura propuesta en AWS detallada	27
Fig. 6: Diagrama de despliegue	32

SIGLAS, ACRÓNIMOS Y ABREVIATURAS

AWS	Amazon Web Services
ECS	Amazon Elastic Container Service
GPC	Google Cloud Platform
IaaS	Infrastructure as a Service
UdeA	Universidad de Antioquia
VPC	Virtual Private Cloud

RESUMEN

Evertec es una empresa cuya línea de negocio principal es a nivel transaccional y financiero. Dentro de mi práctica académica se trabajó en el producto Placetopay que es la pasarela de pagos de esta compañía financiera; se abordó la modernización de la infraestructura del core transaccional de Placetopay, actualmente alojado en un servidor de Legacy de un proveedor externo, cuyo objetivo principal fue migrar hacia una infraestructura más moderna en la nube para mejorar la seguridad, disponibilidad, consistencia y funcionalidad de los servicios dispuestos para nuestros clientes y empresas.

Palabras clave — Modernización, Nube, Seguridad, Consistencia, Funcionalidad, Infraestructura, AWS, Serverless, Servicios legacy.

ABSTRACT

Evertec is a company whose main business line is in the transactional and financial sector. During my academic internship, I worked on the Placetopay product, which is the payment gateway of this financial company. The focus was on modernizing the infrastructure of the Placetopay transactional core, which is currently hosted on a Legacy server from an external provider. The main objective was to migrate to a more modern cloud infrastructure to improve the security, availability, consistency, and functionality of the services provided to our clients and businesses.

Keywords — Modernization, Cloud, Security, Consistency, Functionality, Infrastructure, AWS, Serverless, Legacy services.

I. INTRODUCCIÓN

El presente documento tiene como propósito abordar el proyecto de modernización de la infraestructura del core transaccional de Placetopay, una empresa líder en soluciones de pagos electrónicos en América Latina. Este trabajo se enmarca dentro de la necesidad de migrar los servicios críticos de la plataforma, actualmente alojados en un servidor on-premise, hacia una infraestructura más moderna en la nube. El objetivo es mejorar la seguridad, disponibilidad, consistencia y funcionalidad de los servicios transaccionales que Placetopay ofrece a sus clientes y empresas.

El contexto de este proyecto surge a partir de las limitaciones inherentes a la infraestructura tradicional, la cual ha demostrado ser insuficiente para manejar de manera eficiente el crecimiento y las demandas actuales del negocio. Estas limitaciones incluyen dificultades en escalabilidad, altos costos operativos, dependencias en hardware especializado, y una flexibilidad reducida frente a los cambios y nuevas tecnologías.

La metodología para este proyecto se desarrolló bajo el marco de la modernización tecnológica, comenzando con un análisis detallado de la infraestructura actual. A partir de ese diagnóstico, se propuso un diseño arquitectónico basado en servicios cloud, seguido de pruebas rigurosas y un despliegue controlado en la nueva infraestructura, manteniendo un enfoque continuo en la seguridad y el rendimiento.

Los resultados obtenidos incluyen la implementación de una arquitectura escalable y flexible en la nube, la mejora de la capacidad operativa de Placetopay, y la capacitación del equipo en el uso y mantenimiento de la nueva infraestructura. Estos cambios aseguran una operación eficiente y segura, adecuada para los volúmenes transaccionales que maneja la plataforma y con la capacidad de adaptarse a las necesidades futuras del negocio.

II. OBJETIVOS

A. Objetivo general

Modernizar la infraestructura del core transaccional de Placetopay para garantizar una plataforma más segura, disponible, consistente y funcional, mejorando el rendimiento y la gestión eficiente de los recursos de la compañía para sus clientes.

B. Objetivos específicos

- Realizar un análisis del entorno actual y los requisitos funcionales y no funcionales para la modernización
- Investigar y determinar las mejores tecnologías, teniendo en cuenta aspectos como escalabilidad, costo, rendimiento y compatibilidad con los sistemas existentes.
- Establecer y verificar políticas y prácticas de seguridad robustas para garantizar la protección de la infraestructura modernizada.
- Implementar las tecnologías elegidas, asegurando que se adapten a los objetivos y requisitos del proyecto de modernización.
- Planificar y ejecutar la migración de los servicios de manera ordenada, asegurando una transición transparente y limpia para el cliente.
- Verificar y asegurar el correcto despliegue y funcionamiento de la nueva infraestructura y apagado de la infraestructura anterior.
- Documentar y capacitar al equipo sobre los procesos, servicios y nuevas características implementadas en la infraestructura propuesta.

III. MARCO TEÓRICO

El marco teórico de este trabajo se basa en conceptos clave relacionados con la modernización de infraestructuras tecnológicas y la migración a la nube, con un enfoque en los beneficios y desafíos que implica para una plataforma de pagos electrónicos como Placetopay. Para comprender el contexto de esta modernización, es esencial analizar las limitaciones de los servidores tradicionales y las ventajas que ofrece la infraestructura en la nube, especialmente en términos de escalabilidad, seguridad y eficiencia operativa.

Limitaciones de los Servidores Tradicionales

Los servidores tradicionales, conocidos también como infraestructura on-premise, se encuentran alojados en las instalaciones de la organización, proporcionando control total sobre el hardware y el software. Sin embargo, esta configuración presenta varios inconvenientes:

- **Altos costos operativos:** La adquisición, instalación y mantenimiento de servidores físicos requieren inversiones significativas en hardware, software, energía y personal especializado [4].
- **Escalabilidad limitada:** La capacidad de estos servidores es finita. Si la demanda de recursos aumenta, es necesario adquirir más servidores, lo cual implica tiempo, espacio y costos adicionales [1].
- **Obsolescencia tecnológica:** Los servidores tradicionales suelen quedar obsoletos rápidamente, lo que genera la necesidad de actualizaciones constantes o de reemplazos completos, afectando la continuidad del servicio [8].
- **Falta de flexibilidad:** Adaptarse a nuevas tecnologías o a cambios en las necesidades del negocio es un desafío, debido a que los servidores tradicionales no permiten configuraciones ágiles [6].

-
- **Seguridad:** Si bien los servidores tradicionales pueden ofrecer cierto grado de control sobre la seguridad, también son más vulnerables a ataques externos e internos. La organización debe invertir en medidas de seguridad adicionales, como firewalls, software antivirus y personal especializado, para proteger sus datos y sistemas [9].

Tipos de Servidores Tradicionales

Los servidores tradicionales, pueden clasificarse en varias categorías según su función, configuración y capacidades. Cada tipo de servidor tiene aplicaciones específicas y ofrece diferentes niveles de control, rendimiento y flexibilidad. A continuación, se describen algunos de los tipos más comunes de servidores tradicionales:

- **Servidores físicos:** Estos servidores están compuestos por hardware dedicado que se encuentra físicamente en las instalaciones de la organización. Son conocidos por ofrecer el máximo control y rendimiento, pero también son costosos en términos de adquisición, instalación y mantenimiento. Estos servidores suelen requerir grandes inversiones iniciales y continuos gastos operativos para asegurar su correcto funcionamiento [2].
- **Servidores virtuales:** Utilizan la tecnología de virtualización para crear múltiples servidores dentro de una misma máquina física. Esto permite un uso más eficiente de los recursos, ya que varios servidores virtuales pueden ejecutarse en paralelo en el mismo hardware físico. Aunque ofrecen una mejor optimización de recursos y reducción de costos en comparación con los servidores físicos, los servidores virtuales requieren un software de virtualización robusto y potente para gestionar adecuadamente la carga [3].
- **Servidores en rack:** Son servidores físicos diseñados para montarse en racks estándar en centros de datos. Este tipo de servidores optimiza el espacio físico, ya que permite alojar múltiples servidores en un solo rack. Son comunes en entornos empresariales con alta demanda de procesamiento y almacenamiento, y facilitan la organización y escalabilidad en centros de datos de gran tamaño [4].

-
- **Servidores blade:** Se caracterizan por ser compactos y modulares, ya que están diseñados para insertarse en chasis especializados que alojan varios servidores en un espacio reducido. Los servidores blade ofrecen una alta densidad y escalabilidad, lo que los convierte en una excelente opción para entornos que requieren alta disponibilidad y grandes volúmenes de procesamiento [5].
 - **Servidores de almacenamiento:** Están dedicados a almacenar grandes volúmenes de datos. Estos servidores ofrecen alta capacidad, confiabilidad y rendimiento, siendo fundamentales para organizaciones que manejan grandes bases de datos o archivos. Son esenciales en entornos empresariales donde la cantidad de datos almacenados y procesados crece de manera exponencial [6].
 - **Servidores de red:** Su función principal es gestionar el tráfico de red y proporcionar servicios esenciales como DHCP (Dynamic Host Configuration Protocol), DNS (Domain Name System) y firewall. Estos servidores son esenciales para garantizar la conectividad y seguridad de la red empresarial, ya que gestionan la distribución de direcciones IP y otros servicios de red [8].

Ventajas de la Modernización de Sistemas

La adopción de una infraestructura en la nube es una alternativa moderna que resuelve muchos de los problemas asociados con los servidores tradicionales. Entre las principales ventajas destacan:

- **Mayor eficiencia y productividad:** Los sistemas modernos son más eficientes, rápidos y confiables, lo que permite a los empleados trabajar de manera más productiva y dedicar menos tiempo a tareas manuales o repetitivas [2].
- **Reducción de costos:** La modernización puede reducir significativamente los costos de TI, tanto en licencias de software como en hardware, mantenimiento y personal. Además, puede optimizar el uso de recursos y reducir el consumo de energía [3].
- **Mejora de la seguridad:** Los sistemas modernos incorporan las últimas tecnologías de seguridad, lo que ayuda a proteger los datos y sistemas de la organización contra amenazas cibernéticas [8].

-
- **Mayor agilidad e innovación:** La modernización facilita la adaptación a nuevos cambios y la implementación de nuevas tecnologías, lo que permite a la organización ser más ágil e innovadora [6].
 - **Mejora de la experiencia del usuario:** Los sistemas modernos ofrecen una mejor experiencia de usuario, con interfaces más intuitivas y fáciles de usar. Esto aumenta la satisfacción de los empleados y clientes [3].

Buenas Prácticas para la Modernización de Sistemas

Para llevar a cabo una modernización exitosa de sistemas, es importante seguir algunas buenas prácticas:

1. **Establecer objetivos claros:** Definir claramente los objetivos que se persiguen con la modernización, como mejorar la eficiencia, reducir costos o aumentar la seguridad.
2. **Evaluar el estado actual:** Realizar una evaluación exhaustiva del estado actual de los sistemas, incluyendo hardware, software, procesos y datos.
3. **Identificar las necesidades:** Identificar las necesidades específicas de la organización en cuanto a funcionalidad, rendimiento, seguridad y escalabilidad [10].
4. **Elegir la estrategia adecuada:** Seleccionar la estrategia de modernización que mejor se adapte a las necesidades y objetivos de la organización. Existen diversas opciones disponibles, como la migración a la nube, la reingeniería de software o la refactorización [5].
5. **Planificar cuidadosamente:** Desarrollar un plan detallado que incluya el alcance del proyecto, el cronograma, los recursos necesarios y las estrategias de gestión de riesgos.
6. **Comunicarse efectivamente:** Mantener una comunicación abierta y transparente con todas las partes interesadas durante todo el proceso de modernización.
7. **Probar y validar:** Probar rigurosamente los sistemas modernizados antes de su implementación en producción [10].
8. **Monitorear y optimizar:** Monitorear continuamente el rendimiento y la seguridad de los sistemas modernizados, y realizar ajustes según sea necesario.

Serverless

El enfoque serverless se ha consolidado como una solución eficiente dentro de la infraestructura en la nube. Este modelo permite a las empresas ejecutar aplicaciones sin necesidad de gestionar los servidores, enfocándose en el desarrollo y la implementación de código en lugar de en la administración de la infraestructura. En el caso de Placetopay, la adopción de serverless permite manejar picos de tráfico de manera eficiente, asegurando una continuidad operativa en una plataforma transaccional de alto volumen [1].

IV. METODOLOGÍA

La metodología del proyecto se estructura en varias fases que combinan enfoques cualitativos y cuantitativos para asegurar una modernización exitosa y sin contratiempos. En compañía del equipo de desarrollo del core transaccional de Placetopay, cuyo objetivo como equipo es entregar soluciones de software rápidas, seguras y de calidad, se adoptará la metodología Scrum para gestionar y ejecutar el proyecto. Esta metodología incluye eventos clave como reuniones diarias (daily), planificación de sprints (sprint planning), revisiones de código mediante pull requests en pares, y revisiones de sprint (sprint reviews).

Eventos Scrum:

- **Dailys:** Reuniones diarias de 15 minutos donde los miembros del equipo discuten lo realizado el día anterior, lo que planean hacer hoy y cualquier obstáculo que estén enfrentando. Estas reuniones fomentan la comunicación constante y la rápida resolución de problemas.
- **Sprint Planning:** Reuniones quincenales para planificar el trabajo del próximo sprint. Durante estas reuniones, el equipo define las tareas y objetivos que se deben cumplir en las próximas dos semanas. Esto asegura que todos los miembros del equipo estén alineados y trabajen hacia los mismos objetivos.

- **Sprint Review:** Reuniones al final de cada sprint para revisar el trabajo completado, demostrar las nuevas características y obtener feedback. Estas revisiones permiten al equipo reflejar su progreso y ajustar el plan según sea necesario.
- **Sprint Retrospective:** Reuniones al final de cada sprint para discutir lo que funcionó bien, lo que no y cómo se pueden hacer mejoras en el próximo sprint. Esta retrospectiva es crucial para la mejora continua del equipo.

Revisiones de Código:

Todo el código desarrollado será revisado por al menos un compañero del equipo antes de ser integrado en la rama principal. Este proceso asegura una alta calidad del código y reduce la introducción de errores, promoviendo la colaboración y el aprendizaje entre los miembros del equipo. Para este proceso, se hará a través de la revisión de Pull Request usando herramientas como Bitbucket para el alojamiento de código y colaboración basada en Git.

Cronograma de actividades:

A continuación se presenta el cronograma detallado de las actividades realizadas durante la práctica académica. Este cronograma refleja las fases clave del proyecto, comenzando con la evaluación inicial del entorno on-premise hasta el despliegue en productivo.



Fig. 1. Cronograma de actividades

V. RESULTADOS

El análisis de la infraestructura de Placetopay reveló las limitaciones del modelo on-premise, basado en los servicios de un proveedor externo. Esta configuración dependía de servidores dedicados, lo que restringía significativamente la escalabilidad y flexibilidad del sistema. El tráfico y las transacciones eran gestionados a través de un core transaccional centralizado, que conectaba los servicios web y las bases de datos a través de una red interna. La falta de distribución global en esta infraestructura resultaba en alta latencia para los usuarios ubicados fuera de la región principal, afectando su experiencia. Además, esta infraestructura alojaba tanto el core transaccional como otras aplicaciones críticas de la empresa, como se observa en la **Fig. 1**.

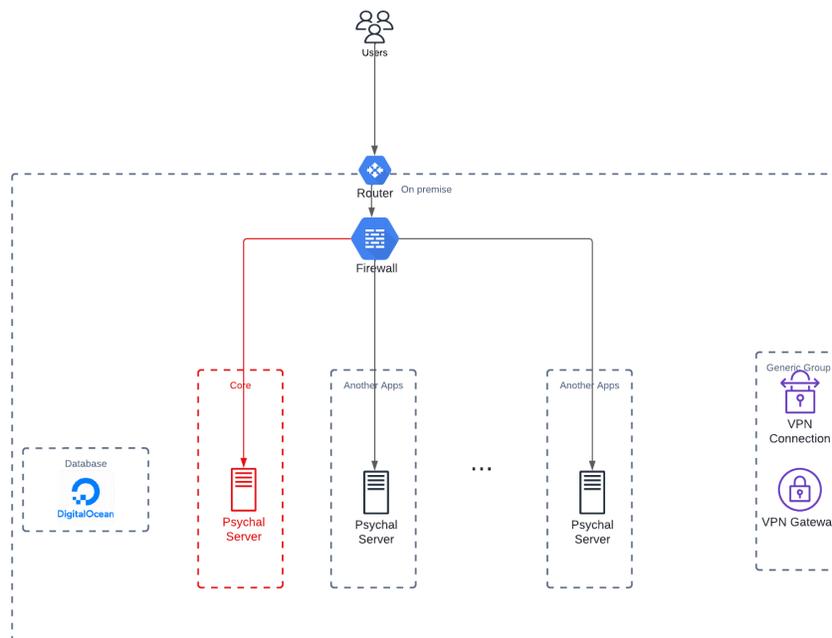


Fig. 2. Infraestructura antigua del aplicativo Placetopay

Aunque este enfoque ofrecía control total sobre la infraestructura, también implicaba elevados costos de mantenimiento y actualización de hardware, además de dificultades para manejar picos de tráfico y el crecimiento en la demanda. La adquisición de nuevo hardware era costosa y lenta, lo que limitaba la capacidad de la empresa para reaccionar rápidamente ante fluctuaciones en el tráfico.

Para resolver estos desafíos, se llevó a cabo una exhaustiva comparación entre los servicios on-premise y las soluciones en la nube, evaluando diferentes plataformas como AWS, Azure y GCP. Esta comparación se basó en criterios clave como costos, escalabilidad, seguridad y facilidad de adopción de nuevas tecnologías. Los resultados de esta evaluación se presentan en la **Tabla I**, destacando las ventajas que ofrece la nube frente al modelo on-premise.

TABLA I
COMPARACIÓN DE SERVICIOS ON-PREMISE VS. SERVICIOS EN LA NUBE

Criterio	On-Premise	Nube (AWS, Azure, GCP)	Comentarios
Costos	Muy Altos: Inversión significativa en hardware especializado, redundancia, disponibilidad, seguridad física, mantenimiento continuo, y personal especializado.	Variables: Pago por uso, opciones de escalabilidad automática, controlados opciones de instancias reservadas y spot. Sin costos iniciales en infraestructura física.	Pago por de La nube ofrece un modelo de costos más flexible y predecible, crucial para gestionar los picos de tráfico en una pasarela de pagos.

Cobertura	<p>Limitada: Alta latencia si no se implementa infraestructura distribuida. Cobertura limitada a la ubicación física de los centros de datos.</p>	<p>Global: Disponibilidad en múltiples regiones y zonas de disponibilidad, permitiendo baja latencia y alta disponibilidad a nivel global.</p>	<p>La nube permite despliegues globales con replicación de datos y baja latencia, esenciales para una pasarela de pagos que opera en múltiples regiones.</p>
Innovación y Agilidad	<p>La adopción de nuevas tecnologías suele ser más lenta, ya que requiere la compra e integración de hardware adicional, lo que puede limitar la capacidad de la empresa para innovar rápidamente</p>	<p>La nube permite a las empresas adoptar rápidamente nuevas tecnologías y servicios innovadores (como inteligencia artificial, big data, y análisis en tiempo real) sin la necesidad de implementar y gestionar la infraestructura subyacente.</p>	<p>La nube facilita la experimentación y adaptación en un mercado en constante evolución.</p>
Escalabilidad	<p>Limitada: Aumentar la capacidad requiere adquisición de nuevo hardware, lo cual es lento y costoso.</p>	<p>Alta: Escalabilidad automática basada en la demanda. Capacidad para manejar incrementos repentinos en el tráfico, crucial para mantener la experiencia del usuario y evitar caídas del sistema.</p>	<p>La nube ofrece escalabilidad bajo demanda, evitando problemas de capacidad en eventos de alto tráfico, como días festivos o promociones.</p>

Facilidad con el Lenguaje	Mediana: Requiere configuración manual, y administración de actualización de servidores, y manejo de entornos complejos.	Alta: Herramientas y servicios que facilitan el despliegue y gestión de aplicaciones PHP/Laravel, cómo Elastic Beanstalk en AWS o App Service en Azure, con menos sobrecarga administrativa.	La nube simplifica la gestión del código y el despliegue, permitiendo que el equipo se concentre en optimizar el flujo transaccional y no en la infraestructura.
Seguridad y Cumplimiento	Alta: Control total sobre la infraestructura, pero requiere inversiones importantes en seguridad física y cibernética.	Alta: Los principales proveedores de nube cumplen con estándares globales de seguridad y regulación (PCI DSS, GDPR, etc.). Implementaciones de seguridad avanzadas como cifrado de datos, IAM, y monitoreo de actividades.	La nube ofrece un entorno de seguridad robusto y certificado, crucial para operaciones financieras.

Esta comparación destaca las claras ventajas de la infraestructura en la nube, especialmente en términos de flexibilidad y capacidad de escalar automáticamente en función de la demanda, algo crucial para una pasarela de pagos que experimenta picos de tráfico en eventos como promociones o temporadas festivas.

Arquitectura propuesta

La nueva arquitectura propuesta para Placetopay se basa en una solución en la nube distribuida a través de múltiples regiones geográficas, utilizando servicios de AWS para lograr una mayor flexibilidad, escalabilidad y eficiencia operativa. Esta infraestructura adopta un enfoque serverless, donde los servicios web y las bases de datos se gestionan a través de balanceadores de carga y subredes privadas, optimizando la disponibilidad del sistema y mejorando la capacidad de respuesta.

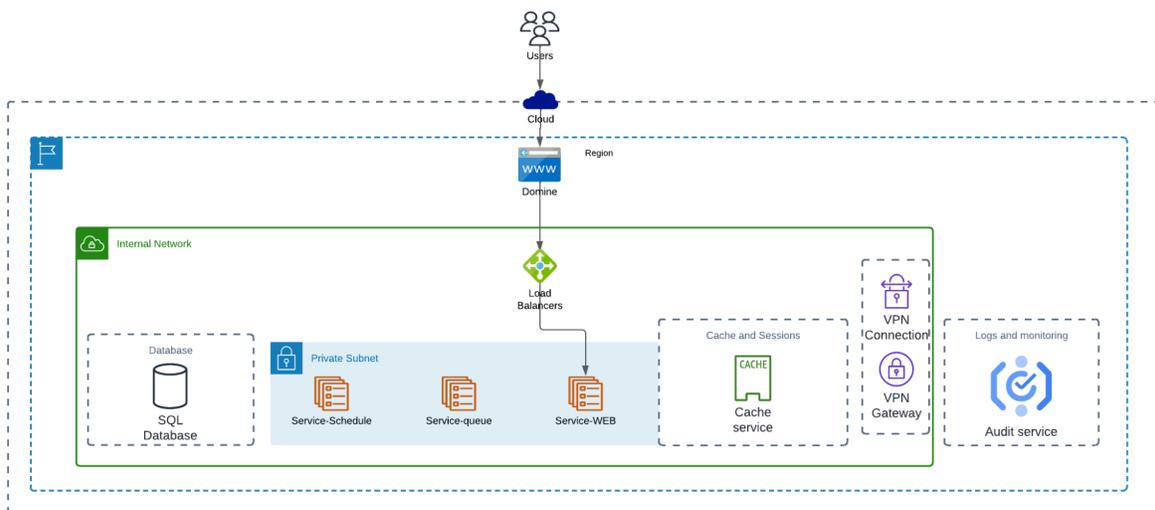


Fig. 3: Arquitectura en la nube propuesta

La implementación de esta arquitectura permite a Placetopay ajustar automáticamente su capacidad según la demanda en tiempo real, lo que es esencial para gestionar grandes volúmenes de transacciones de manera eficiente. Además, la alta disponibilidad está garantizada por la distribución geográfica de los recursos y el uso de mecanismos avanzados de seguridad proporcionados por Infrastructure as a Service (AWS, Azure, GCP), lo que asegura la protección de los datos y el rendimiento del sistema.

El diseño de la arquitectura propuesta incluye una serie de componentes clave que trabajan en conjunto para ofrecer un sistema eficiente, seguro y altamente escalable:

TABLA II
COMPONENTES DE LA NUEVA INFRAESTRUCTURA

Componente	Definiciones
Usuarios	Los usuarios acceden al sistema a través de un navegador o cliente. Todas las peticiones realizadas por los usuarios son enviadas a través de la red hacia la infraestructura en la nube, donde se gestionan los recursos necesarios para procesarlas y devolver las respuestas correspondientes.
Cloud y Región	La infraestructura está desplegada en la nube, distribuyendo los recursos en una o varias regiones geográficas. Este enfoque garantiza alta disponibilidad y redundancia geográfica, lo que permite que el sistema siga funcionando incluso en caso de fallos en una región específica.
Dominio	El dominio público es el punto de entrada para los usuarios, permitiendo que accedan a los servicios web a través de un nombre de dominio amigable. Este dominio apunta a los recursos en la nube, facilitando el acceso a los servicios distribuidos.
Balancedores de Carga	Los balanceadores de carga son responsables de distribuir el tráfico entrante entre múltiples instancias de servicios web, evitando que una instancia individual se sobrecargue. Esto mejora tanto la disponibilidad como la escalabilidad del sistema, garantizando una respuesta rápida y consistente incluso durante picos de tráfico.
Internal Network (Red Interna)	La red interna proporciona un entorno seguro y aislado dentro de la nube, donde se despliegan los servicios principales, como bases de datos y servicios backend. Esto asegura que la comunicación interna del sistema esté protegida y optimizada para el rendimiento.
Private Subnet (Subred Privada)	Dentro de la red interna, los servicios más críticos y sensibles se alojan en una subred privada. Estos servicios no están expuestos directamente a Internet, lo que añade una capa adicional de seguridad y protección frente a amenazas externas.
Service-Schedule (Servicio de Schedule)	Este servicio es responsable de la planificación y ejecución de tareas programadas, como cron jobs y comandos. Al gestionar las tareas recurrentes de forma independiente del servicio web, se evita la sobrecarga y se asegura que los procesos críticos se ejecuten de manera oportuna y confiable.
Service-Queue (Servicio de Colas)	El servicio de colas gestiona las tareas asíncronas, desacoplando la producción y el consumo de tareas. Esto permite que las tareas sean procesadas de manera eficiente y escalable, mejorando el rendimiento general del sistema.

Service-WEB (Servicio Web)	Este componente maneja las solicitudes web que provienen de los usuarios, sirviendo contenido dinámico o estático según sea necesario. Al estar detrás del balanceador de carga y dentro de la subred privada, se garantiza que las solicitudes se gestionen de manera segura y eficiente.
SQL Database	Este componente maneja las solicitudes web que provienen de los usuarios, sirviendo contenido dinámico o estático según sea necesario. Al estar detrás del balanceador de carga y dentro de la subred privada, se garantiza que las solicitudes se gestionen de manera segura y eficiente.
Cache Service (Servicio de Caché)	Este servicio gestiona el almacenamiento en caché de datos y sesiones, lo que mejora significativamente el rendimiento del sistema. Al reducir la carga en la base de datos, acelera la entrega de respuestas a los usuarios, optimizando la experiencia general.
Audit Service (Servicio de Auditoría)	Para garantizar la seguridad y el cumplimiento de normativas, se sugiere la implementación de un servicio de auditoría dedicado. Este componente recopila logs, realiza el monitoreo continuo del sistema y ayuda a identificar y solucionar problemas, lo que es crucial para mantener altos estándares de seguridad y rendimiento.

Adopción de una infraestructura serverless

La infraestructura serverless representa un enfoque moderno en la computación en la nube, donde los desarrolladores pueden ejecutar aplicaciones sin preocuparse por la administración de la infraestructura subyacente. En lugar de aprovisionar, escalar y gestionar servidores de manera manual, el proveedor de servicios en la nube se encarga automáticamente de estos aspectos. Esto permite que los equipos de desarrollo se centren exclusivamente en la creación de código y la implementación de nuevas funcionalidades, optimizando el proceso de desarrollo. En una arquitectura serverless, las aplicaciones se dividen en pequeñas funciones que se ejecutan en respuesta a eventos específicos y escalan automáticamente según la demanda.

La adopción de una arquitectura serverless en Placetopay ha sido clave para mejorar la eficiencia operativa y la escalabilidad del sistema, permitiendo a la empresa enfocarse en el desarrollo y la innovación sin las complicaciones de gestionar infraestructura física. La escalabilidad automática, el modelo de pago por uso y la capacidad de responder rápidamente a picos de tráfico hacen que serverless sea una opción ideal para una pasarela de pagos de alto flujo transaccional.

Comparación entre Proveedores de Nube

Una vez tomada la decisión de migrar a la nube, fue necesario seleccionar el proveedor adecuado para implementar esta nueva arquitectura. Para ello, se construyó una tabla comparativa que evaluó los principales proveedores de servicios en la nube: AWS, Azure, y Google Cloud Platform (GCP). La comparación se centró en aspectos críticos como costos, cobertura, latencia, escalabilidad y soporte para serverless, todos elementos clave para asegurar el éxito de la plataforma en su transición hacia la nube.

Criterios de comparación entre AWS, Azure y GCP

En la tabla III se analizan criterios como costos, cobertura, latencia, escalabilidad, y soporte para serverless, todos aspectos críticos para la pasarela de pagos que maneja un alto volumen transaccional y que requiere una infraestructura flexible y altamente disponible. La comparación proporciona una visión clara de cómo cada proveedor puede cumplir con los requisitos del proyecto, facilitando la selección de la opción más adecuada para implementar la solución en la nube.

TABLA III
COMPARACIÓN ENTRE PROVEEDORES DE NUBE

Criterio	AWS	AZURE	GCP	Comentarios
-----------------	------------	--------------	------------	--------------------

Costos	Competitivo, con modelos de ahorro a largo plazo y opciones como instancias Spot, que permiten optimizar costos.	Similar a AWS, con buena integración a entornos Microsoft, ya existe dependencia en tecnologías como .NET o SQL Server.	Generalmente competitivo, con herramientas especializadas en machine learning y análisis de datos.	AWS ofrece una amplia gama de opciones de precios, con el mayor número de servicios optimizados para aplicaciones empresariales críticas.
Cobertura y Latencia	Amplia cobertura global con la mayor cantidad de zonas de disponibilidad. Servicios como Global Accelerator mejoran la latencia.	Cobertura global amplia, con una red sólida que soporta grandes volúmenes de transacciones.	Buena cobertura, aunque menos regiones que AWS y Azure. Servicios como Cloud Load Balancing ayudan a reducir la latencia.	AWS es líder en cobertura y opciones para reducir la latencia, crucial para la pasarela de pagos ya que tiene usuarios globales.
Escalabilidad	Escalabilidad automática a través de servicios como AWS Auto Scaling, ECS, y Lambda para manejo de altos volúmenes transaccionales.	Escalabilidad automática con Azure Autoscale y Azure Kubernetes Service (AKS).	Escalabilidad eficiente con Google Kubernetes Engine (GKE) y Cloud Functions.	AWS ofrece la más amplia gama de herramientas de escalabilidad, ideales para flujos transaccionales de alto volumen.

Seguridad y Cumplimiento	Cumplimiento con estándares globales, incluyendo PCI DSS, herramientas como AWS Shield y GuardDuty para protección adicional.	Cumplimiento de estándares globales con servicios avanzados de seguridad como Azure Security Center.	Fuerte enfoque en seguridad y cumplimiento con herramientas como Google Cloud Armor y VPC Service Controls.	AWS proporciona la oferta más amplia y madura de herramientas de seguridad, con una vasta experiencia en cumplimiento de normativas financieras.
Soporte para Serverless	AWS Fargate permite ejecutar contenedores sin necesidad de gestionar servidores, soportando aplicaciones más complejas que no se ajustan a un modelo puramente serverless como Lambda.	Azure Container Instances ofrece una solución similar, permitiendo el despliegue rápido de contenedores sin gestión de infraestructura.	Google Cloud Run ofrece capacidades similares para ejecutar contenedores en un entorno serverless.	AWS Fargate es ideal para aplicaciones que requieren mayor control y funcionalidad avanzada, manteniendo un enfoque serverless sin las limitaciones de AWS Lambda.

Después de analizar las capacidades de cada proveedor, AWS se destacó como la opción más adecuada para Placetopay, principalmente por su flexibilidad en costos, cobertura global y la solidez de su oferta serverless. Su capacidad para ofrecer escalabilidad bajo demanda y un conjunto integral de herramientas para la implementación de soluciones serverless hace que AWS sea el mejor aliado para manejar el alto volumen transaccional que requiere Placetopay. Además, la capacidad de AWS para adaptarse a las necesidades crecientes del negocio sin comprometer la seguridad o la eficiencia operativa ha sido determinante para la selección final.

Con la infraestructura serverless de AWS, Placetopay no solo asegura su capacidad para gestionar grandes volúmenes de transacciones de manera eficiente, sino que también optimiza costos y acelera el ciclo de innovación, lo que le permite mantenerse competitiva en un entorno dinámico y en constante evolución.

Arquitectura definida:

La arquitectura final seleccionada para la implementación en la nube de Placetopay, utilizando AWS, se ilustra en **Fig. 4** y **Fig. 5**. Esta arquitectura muestra cómo los distintos componentes del sistema se organizan e interconectan dentro de la infraestructura, aprovechando servicios clave como Amazon ECS con Fargate para la gestión de contenedores, Amazon Route 53 para la distribución eficiente del tráfico, y Amazon ElastiCache para la gestión de sesiones y el almacenamiento en caché. El diseño está optimizado para ofrecer alta disponibilidad y escalabilidad automática, asegurando que el sistema pueda adaptarse a las necesidades crecientes de Placetopay, especialmente en momentos de alto flujo transaccional. Además, se incorporan herramientas robustas para el despliegue y monitoreo, como AWS CloudFormation y Amazon CloudWatch, que permiten una gestión y supervisión efectiva del sistema.

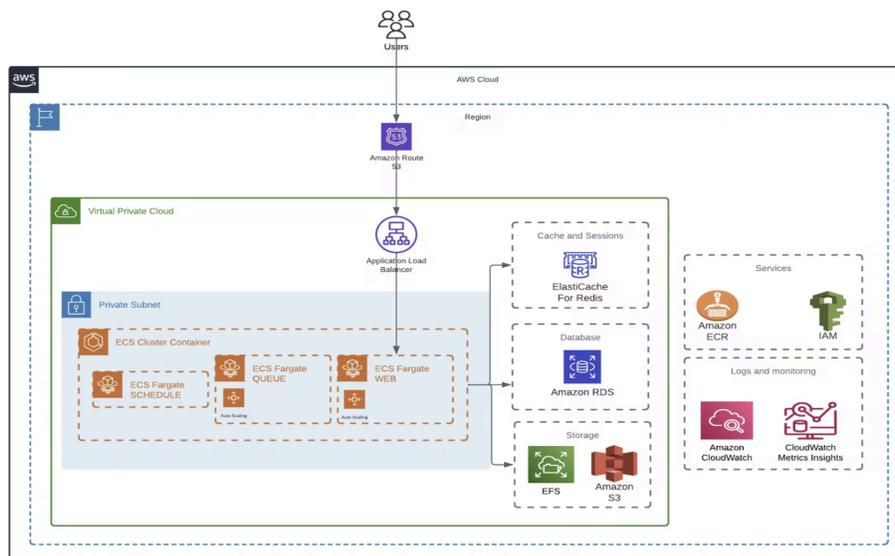


Fig. 4: Arquitectura propuesta en AWS

Fargate	<p>La elección de Fargate se basó en varios factores técnicos. En primer lugar, Fargate es un servicio sin servidor para la ejecución de contenedores, lo que elimina la necesidad de administrar la infraestructura subyacente. Esto permite un enfoque más centrado en el desarrollo y la implementación de la aplicación, en lugar de preocuparse por la gestión de servidores.</p> <p>Además, Fargate ofrece una alta escalabilidad y elasticidad, lo que garantiza que el aplicativo pueda manejar aumentos en la demanda de manera eficiente, sin comprometer el rendimiento ni la disponibilidad. Fargate también ofrece un aislamiento seguro entre contenedores, lo que mejora la seguridad del aplicativo y protege los datos sensibles.</p> <p>Otro beneficio importante de Fargate es su integración directa con otros servicios de AWS, como Amazon Elastic Container Registry (ECR) para el almacenamiento de imágenes de contenedores, Amazon Elastic Load Balancer (ELB) para la distribución de carga y Amazon CloudWatch para el monitoreo y la observación del sistema. Esta integración facilita la implementación y la gestión del aplicativo en la nube, aprovechando la infraestructura escalable y altamente disponible de AWS.</p>
Task Definition	<p>Una definición de tarea (Task Definition) en ECS es una descripción de cómo se ejecutará una tarea en un clúster. Incluye información sobre los contenedores que se utilizarán, los recursos asignados, las variables de entorno y las configuraciones de red, entre otros detalles. Una vez que se crea una definición de tarea, se puede utilizar para ejecutar múltiples instancias de esa tarea.</p>
Task	<p>Una tarea (Task) en ECS representa una unidad de trabajo que se ejecuta en un contenedor. Puedes definir una tarea para ejecutar uno o más contenedores simultáneamente, lo que permite desplegar aplicaciones con múltiples componentes. Las tareas de ECS se pueden ejecutar en clústeres de EC2 o en Fargate.</p>
Service	<p>Un servicio en ECS es una especificación que define cómo se deben ejecutar y mantener las tareas en un clúster de contenedores. Un servicio garantiza que el número deseado de tareas se esté ejecutando y proporciona escalabilidad automática, balanceo de carga y recuperación ante fallas.</p>

CloudFormation	AWS CloudFormation es un servicio que permite aprovisionar y administrar recursos de AWS de manera declarativa. Te permite crear plantillas que describen la infraestructura y los recursos necesarios para tu aplicación. Con CloudFormation, puedes crear y actualizar fácilmente el conjunto completo de recursos de tu sistema legacy, incluidos los servicios de ECS, Fargate, EKS, etc.
Amazon CloudWatch	Amazon CloudWatch es un servicio de supervisión y observación que recopila y rastrea métricas, registros y eventos operativos en AWS. Con CloudWatch, puedes monitorear el rendimiento de tus aplicaciones y recursos, obtener alarmas en caso de problemas y acceder a datos detallados de registro para análisis y depuración.
AWS Lambda	AWS Lambda es un servicio de computación sin servidor que te permite ejecutar código sin provisionar ni administrar servidores. Puedes utilizar Lambda para ejecutar código en respuesta a eventos, como cambios en el estado de tus servicios de ECS. Lambda puede integrarse con otros servicios de AWS para crear flujos de trabajo automatizados y escalar automáticamente tu sistema legacy.

En conjunto, estos servicios de AWS ofrecen una infraestructura escalable y flexible para ejecutar, orquestar y monitorear aplicaciones en contenedores, así como automatizar tareas y operaciones en el aplicativo. Proporcionan una mayor eficiencia, seguridad y facilidad de administración, permitiéndote modernizar y optimizar el entorno de manera ágil y sin preocuparnos por la gestión de la infraestructura subyacente.

Implementación y Ejecución

Durante la modernización de la infraestructura de Placetopay, se optó por implementar una solución serverless utilizando el **Serverless Framework**, que se ajusta perfectamente a las necesidades del aplicativo Placetopay. Este framework resultó ser la herramienta ideal para integrar Laravel con una arquitectura serverless, permitiendo automatizar la creación y despliegue de servicios de forma eficiente. La adopción de Serverless facilitó la administración de recursos en la nube sin necesidad de gestionar servidores directamente, lo que liberó al equipo de desarrollo para concentrarse en la funcionalidad del software.

Creación de la estructura serverless

La estructura creada para Placetopay incluyó componentes clave que permiten un despliegue escalable y seguro en la nube de AWS. La configuración estableció la base para que los servicios web, las colas de procesamiento de tareas y las tareas programadas operaran de manera óptima en un entorno serverless. Se integraron servicios como AWS Fargate para la gestión de contenedores, Amazon Route 53 para la distribución de tráfico y la creación de un sistema de balanceo de carga, y Amazon ElastiCache para optimizar el rendimiento de las consultas y gestionar sesiones de manera eficiente.

El **Serverless Framework** permitió definir las políticas de escalabilidad automática para los servicios de ECS, y la infraestructura necesaria para asegurar la alta disponibilidad y el correcto desempeño del sistema.

Uso de Terraform para la gestión de infraestructura

Aunque **Serverless Framework** permite definir además las políticas de seguridad y acceso, además de gestionar la creación y configuración de recursos como roles de **IAM**, Por temas de manejo interno de la compañía en colaboración con el equipo de infraestructura, se utilizó Terraform para gestionar la creación de estos recursos clave en AWS. El equipo no solo creó los usuarios IAM y otros componentes de seguridad, si no que también realizó la migración la base de datos existente de Placetopay a la infraestructura en la nube.

Se configuraron bases de datos relacionales en Amazon RDS y se implementó un sistema de caché en Redis para mejorar el rendimiento de los datos temporales. Para optimizar la gestión de las sesiones del sistema y mejorar la experiencia del usuario, también se implementó Amazon DynamoDB como una base de datos NoSQL que permite almacenar de forma eficiente las sesiones del sistema con un alto rendimiento y baja latencia.

Construcción de imágenes Docker

Uno de los elementos esenciales del despliegue fue la construcción de imágenes Docker personalizadas para los diferentes servicios de Placetopay. Se crearon imágenes específicas para el servicio web, las colas de procesamiento y las tareas programadas, garantizando que cada servicio fuera independiente, pero a la vez integrado dentro del ecosistema de contenedores gestionados por AWS Fargate.

El uso de Docker permitió encapsular todas las dependencias y configuraciones necesarias para cada servicio, lo que facilitó su despliegue y escalabilidad. Al utilizar estas imágenes, el sistema puede ajustarse automáticamente en función de la demanda, iniciando nuevos contenedores cuando sea necesario sin comprometer el rendimiento. Además, las imágenes Docker aseguran la portabilidad entre entornos de desarrollo, prueba y producción, garantizando que el sistema funcione de manera consistente en todas las etapas del ciclo de vida del software.

Balanceo de carga y autoscaling

Para mejorar aún más la eficiencia y disponibilidad del sistema, se configuraron balanceadores de carga mediante Amazon Elastic Load Balancer (ELB). Estos balanceadores distribuyen automáticamente el tráfico entre las distintas instancias del servicio, garantizando que no haya sobrecarga en ninguna de ellas y que el sistema mantenga un alto rendimiento en todo momento. También se establecieron políticas de autoscaling que permiten que los servicios ajusten su capacidad de procesamiento según la demanda, lo que es crucial para una pasarela de pagos como Placetopay, especialmente durante eventos de alto tráfico.

Ejecución de comandos

Dentro de esta solución se contempló la creación de una **función Lambda**, que permitirá la ejecución de comandos dentro del aplicativo, proporcionando una forma eficiente y escalable de realizar tareas y operaciones específicas. Esta función ofrecerá la flexibilidad necesaria para realizar procesos automatizados y manejar eventos dentro del sistema sin comprometer el rendimiento.

Proceso de despliegue

Para llevar a cabo el despliegue del aplicativo en la infraestructura propuesta, se implementó una pipeline de despliegue continuo en Bitbucket inicialmente para los entornos de Desarrollo, con la ayuda del ya mencionado **Serverless Framework**.

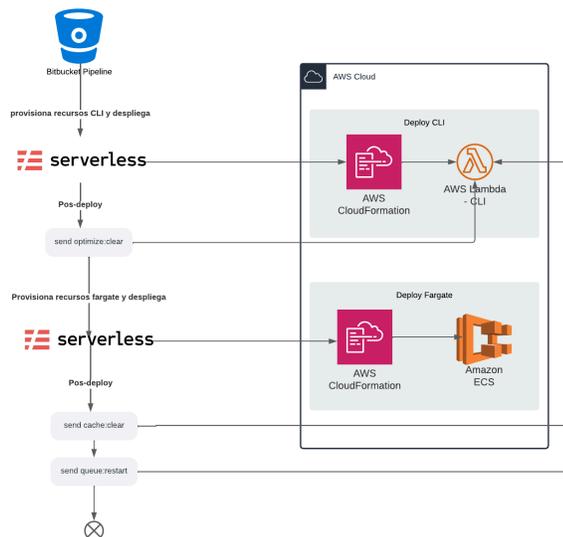


Fig. 6: Diagrama de despliegue

El despliegue de la nueva infraestructura para el aplicativo "Placetopay" se realizó de manera planificada y en fases, cumpliendo con los objetivos de garantizar una transición ordenada y sin interrupciones en los servicios. El proceso fue diseñado para asegurar una migración transparente para el cliente, verificar el correcto funcionamiento de la nueva infraestructura, y proporcionar la documentación y capacitación necesarias para el equipo de desarrollo.

Planificación y Ejecución de la Migración

Inicialmente, se decidió llevar a cabo el despliegue en un ambiente de desarrollo. En esta etapa, se desplegaron los servicios necesarios y se realizaron pruebas exhaustivas de todos los servicios de la API. Aunque al principio el proceso fue tedioso debido a fallos en la creación de algunos componentes y configuraciones iniciales, el equipo de desarrollo logró superar estos inconvenientes. Los problemas se debían principalmente al desconocimiento de la infraestructura.

Se llevaron a cabo pruebas de escalabilidad para asegurarse de que el sistema pudiera manejar picos de tráfico, lo cual es fundamental en una pasarela de pagos como Placetopay. Las pruebas iniciales revelaron algunas limitaciones en la configuración, pero tras realizar ajustes y optimizaciones en la infraestructura, las pruebas resultaron exitosas. Esto permitió verificar que los servicios funcionaban correctamente bajo carga y que la infraestructura podría escalar automáticamente en respuesta a la demanda.

Despliegue en un Ambiente de Testing Controlado

Una vez se validó el correcto funcionamiento en el ambiente de desarrollo, el siguiente paso fue desplegar la infraestructura en un ambiente de Testing Controlado. Esta fase permitió simular escenarios reales de uso y validar el comportamiento de los servicios bajo condiciones similares a las de producción.

Las pruebas en este entorno incluyeron:

- Validación del comportamiento de los servicios bajo diferentes cargas de trabajo.
- Pruebas de escalabilidad y autoajuste de recursos.

-
- Verificación de la integración entre los servicios, asegurando que la comunicación entre los componentes funcionaba sin problemas.

El despliegue en el ambiente de Testing Controlado fue clave para identificar y solucionar problemas potenciales antes de migrar al entorno de producción, lo que redujo significativamente el riesgo de fallos en el proceso de transición. Tras realizar los ajustes finales, los resultados de las pruebas fueron satisfactorios, confirmando que la infraestructura estaba lista para su implementación en producción.

Verificación y Despliegue en Producción

Con los resultados positivos obtenidos en el ambiente de testing, se procedió a la migración final hacia el entorno de producción. Este paso se planificó cuidadosamente para asegurar una transición limpia y transparente para el cliente.

Se verificó que todos los servicios estuvieran operativos y que la infraestructura pudiera manejar el tráfico real de manera eficiente. Además, se implementaron mecanismos de monitoreo para garantizar que el sistema continuará funcionando correctamente después del despliegue, permitiendo reaccionar de manera proactiva a cualquier eventualidad.

Documentación y Capacitación

Para asegurar una operación continua y sostenible, se documentaron detalladamente todos los procesos implementados, incluyendo:

- La arquitectura de la nueva infraestructura.
- Los servicios desplegados y sus configuraciones.
- Los procedimientos para el escalado automático y la gestión de cargas de trabajo.

Se organizó una capacitación para el equipo de desarrollo e infraestructura, asegurando que todos los miembros del equipo estuvieran familiarizados con las nuevas características y procesos. Esta capacitación incluyó sesiones prácticas sobre cómo gestionar la nueva infraestructura, ajustar las configuraciones en caso de necesidad y cómo realizar monitoreos continuos para asegurar la estabilidad del sistema.

VI. ANÁLISIS

La modernización de la infraestructura transaccional de Placetopay fue un proceso complejo que implicó la transición de un entorno on-premise basado en una solución en la nube, buscando mayor escalabilidad, flexibilidad y optimización de costos. Durante esta fase, se identificaron limitaciones clave del modelo on-premise, como los elevados costos de mantenimiento y la rigidez para adaptarse a variaciones en la demanda.

La implementación de una infraestructura en la nube, utilizando servicios serverless como AWS Fargate y Lambda, resolvió estos problemas y mejoró la capacidad operativa del sistema. Tras evaluar proveedores como AWS, Azure y GCP, se optó por AWS debido a su sólido soporte para soluciones serverless, su alcance global y su flexibilidad en costos. Herramientas como Docker, Terraform y el Serverless Framework facilitaron la automatización y creación de una infraestructura ajustada a las necesidades del negocio, sin comprometer la seguridad ni la eficiencia.

El despliegue se realizó de manera gradual, comenzando en entornos de desarrollo y siguiendo con pruebas exhaustivas en un ambiente controlado de testing. Aunque hubo desafíos técnicos iniciales, la planificación y pruebas escalonadas permitieron una implementación exitosa en producción, sin interrupciones de servicio y garantizando una transición fluida para los usuarios.

VII. CONCLUSIONES

La modernización de la infraestructura de Placetopay hacia la nube fue una experiencia que no solo trajo avances técnicos, sino que también enriqueció profundamente el conocimiento del equipo en cuanto a tecnologías cloud. Durante la práctica, se tuvo la oportunidad de familiarizarse de manera directa con los principios y ventajas de la computación en la nube.

Uno de los principales logros fue la implementación de una infraestructura serverless, utilizando AWS Fargate y Amazon Lambda, que permitió gestionar automáticamente los recursos y optimizar el uso de la infraestructura de manera eficiente, asegurando que la plataforma pudiera manejar picos de tráfico sin interrupciones. Este modelo también ha reducido los costos operativos al eliminar la necesidad de adquirir y mantener hardware físico, permitiendo un modelo de pago por uso más flexible y predecible.

Comprender la importancia de la escalabilidad automática y la optimización de recursos fue un elemento fundamental para llevar a cabo este trabajo. Estos factores contrastan con los enfoques más rígidos de sistemas on-premise. La capacidad de ajustar dinámicamente los recursos según la demanda proporcionó una visión más clara sobre cómo la nube permite mejorar tanto el rendimiento como la eficiencia operativa, y cómo los modelos de pago por uso ayudan a optimizar costos.

A lo largo del proyecto, surgieron desafíos que permitieron profundizar en la gestión de infraestructura a través de herramientas como Docker, Terraform y el Serverless Framework, lo cual no solo fue clave para el éxito de la migración, sino que también sirvió como un proceso formativo que amplió las competencias del equipo en términos de automatización, seguridad y monitoreo en entornos cloud.

En cuanto a estimaciones, los retrasos en la integración de la infraestructura propuesta demostraron la necesidad de ajustar las expectativas iniciales. Estos aprendizajes ayudaron a refinar la planificación y a adoptar un enfoque más ágil en la gestión de tiempos y recursos, lo

cual es esencial en entornos de infraestructura cloud, donde los cambios y la adaptación son constantes.

Finalmente, la práctica no sólo permitió modernizar la plataforma de Placetopay, sino que también sirvió como un vehículo para desarrollar un entendimiento más profundo de las tecnologías cloud, enriqueciendo tanto el conocimiento técnico como las habilidades prácticas del equipo. Estos aprendizajes y experiencias no solo preparan para futuros proyectos en la nube, sino que también ofrecen una sólida base para enfrentar desafíos complejos de la ingeniería de sistemas.

REFERENCIAS

- [1] Ambit Software, "Amazon Web Services," [Online]. Available: <https://www.ambitsoftware.com/accelerate/alliances-platforms/amazon-web-services/>
- [2] Forrester, "Introducing the Forrester Wave: Applications Modernization and Migration Services, Q1 2024," 2024. [Online]. Available: <https://www.forrester.com/blogs/introducing-the-forrester-wave-applications-modernization-and-migration-services-q1-2024/>
- [3] Henson Group, "Why Cloud With Us," [Online]. Available: <https://www.hensongroup.com/why-cloud-with-us>
- [4] iCorps, "Pros and cons of Microsoft Azure cloud service for businesses," [Online]. Available: <https://blog.icorps.com/pros-and-cons-microsoft-azure>
- [5] McKinsey & Company, "Modernizing IT for a Digital Era," [Online]. Available: <https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/modernizing-it-for-a-digital-era>
- [6] Oracle, "Oracle Cloud for ISVs," [Online]. Available: <https://www.oracle.com/cloud/oracle-at-oracle/isv/>
- [7] Serverless, "How it works," [Online]. Available: <https://www.serverless.com/#How-It-works>
- [8] Tivit, "Ventajas de los servicios en la nube," [Online]. Available: <https://latam.tivit.com/blog/servicios-en-la-nube>

-
- [9] Ventajas y Desventajas Top, "Alibaba Cloud: Ventajas y desventajas," [Online]. Available: <https://ventajasydesventajastop.com/alibaba-cloud-ventajas-y-desventajas>
- [10] VMware, "3 Steps to Modernize Infrastructure for Cloud," [Online]. Available: <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/hyper-converged-infrastructure/vmw-3-steps-to-modernize-infrastructure-for-cloud-infographic.pdf>

ANEXOS

Anexo A. Poster

Departamento de Ingeniería de Sistemas

Planeación, modernización y migración de la infraestructura transaccional para la pasarela de pagos Placetopay



UNIVERSIDAD DE ANTIOQUIA
Facultad de Ingeniería

PRACTICANTE: Valeria Granada Rodas

PROGRAMA: Ingeniería de sistemas

ASESORES: Oscar Lopera (Interno), Mario González Portillo (Evertec Inc.)

Semestre de la práctica: 2024-1

Evertec Inc. es una empresa líder en soluciones tecnológicas y servicios transaccionales en América Latina y el Caribe, especializada en la gestión de pagos electrónicos y el desarrollo de plataformas financieras. Entre sus principales productos se encuentra Placetopay, una pasarela de pagos que permite a negocios y consumidores realizar transacciones financieras de manera segura y eficiente. Placetopay es un referente en el sector financiero, proporcionando herramientas para la gestión de pagos en línea, optimizando la experiencia del usuario, y asegurando altos estándares de seguridad en todas sus operaciones. Este producto es esencial para el comercio electrónico y las operaciones digitales de múltiples industrias.



Introducción

El proyecto de práctica académica se enfocó en la modernización del núcleo transaccional de Placetopay, que anteriormente operaba en una infraestructura on-premise basada en servidores físicos. Dado el crecimiento del negocio y las demandas de escalabilidad, se identificó la necesidad de migrar hacia una infraestructura en la nube más moderna. Este cambio tenía como objetivo mejorar la seguridad, la disponibilidad, y la capacidad de la plataforma para adaptarse a las fluctuaciones del mercado. La práctica permitió implementar tecnologías avanzadas, como soluciones serverless en AWS, y contribuyó significativamente a la mejora operativa de Placetopay, marcando un hito en su evolución tecnológica.



Objetivos

Analizar la infraestructura actual y definir requisitos funcionales y no funcionales.

Evaluar tecnologías y proveedores cloud considerando costo, escalabilidad y compatibilidad.

Diseñar e implementar una infraestructura en la nube con políticas robustas de seguridad.

Planificar y ejecutar la migración de forma transparente para los usuarios.

Capacitar al equipo en el manejo de la nueva infraestructura y documentar los procesos clave.



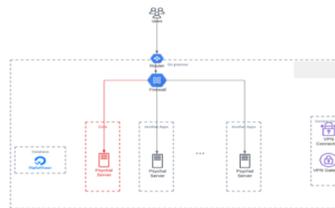
Metodología

Para llevar a cabo el proyecto, se inició con un análisis detallado de la infraestructura actual, identificando sus limitaciones y definiendo los requisitos funcionales y no funcionales para la modernización. Posteriormente, se diseñó una solución basada en la nube, evaluando tecnologías y herramientas que mejor se adaptaran a los objetivos del proyecto. Se realizó la implementación de la arquitectura propuesta. El enfoque adoptado incluyó la planificación y ejecución de pruebas en entornos controlados para garantizar la estabilidad y escalabilidad de la nueva infraestructura antes de su despliegue final.

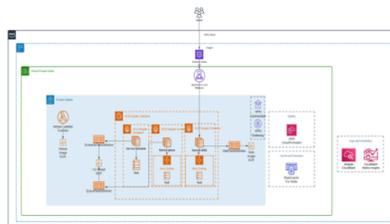


Resultados

- Migración exitosa del núcleo transaccional de Placetopay hacia una arquitectura serverless en AWS.
- Introducción de capacidades para automatizar procesos de despliegue, simplificando configuraciones y actualizaciones, lo que permite una gestión más eficiente para el equipo de desarrollo.
- Beneficios de la infraestructura en la nube
- La nueva infraestructura garantiza un mejor rendimiento del sistema y una experiencia mejorada para los usuarios finales, posicionando a Placetopay como una solución tecnológica avanzada y preparada para el futuro.



Infraestructura antigua del aplicativo Placetopay



Arquitectura propuesta en AWS detallada

Conclusiones

La migración a una infraestructura en la nube marcó un hito en la evolución de Placetopay, mejorando su capacidad para manejar grandes volúmenes de transacciones de manera eficiente y segura.

La arquitectura serverless implementada asegura escalabilidad y flexibilidad, elementos clave para responder a las demandas actuales y futuras del mercado.

La automatización y el manejo de infraestructura como código facilitaron la gestión y optimización del sistema, reduciendo errores y tiempos de implementación.

Este proyecto no solo fortaleció la operación tecnológica de Placetopay, sino que también brindó al equipo técnico conocimientos avanzados en soluciones cloud, posicionándose para afrontar nuevos desafíos tecnológicos.

DATOS DE CONTACTO DEL AUTOR:



+57 3128150756



valeria.granada1@udea.edu.co



@valeriagranadarodas



<https://co.linkedin.com/in/vale0722>