



Informe final de pasantía

“Migración a microservicios en empresa de tecnologías reguladoras (RegTech)”

Fernando Mardones

Ingeniería Civil Informática

Pasantía Full-Time

3 de diciembre del 2023

índice

Resumen ejecutivo	3
Abstract	4
Introducción	5
Objetivos	11
Estado del arte	13
Solución escogida	18
Evaluación económica	20
Metodología	24
Métricas	28
Plan de implementación	33
Análisis de riesgo	36
Resultados obtenidos	40
Conclusión	41
Anexo	42
Referencias	48

Resumen ejecutivo

Fundada en 2019, Ceptinel es una empresa especializada en tecnologías reguladoras y monitoreo transaccional, con presencia consolidada en diversos países, entre ellos Chile, Colombia, Panamá y México. A principios de este año, inició un proceso de migración hacia una nueva arquitectura de software denominada microservicios, orientada a abordar desafíos de escalabilidad y mantenimiento en su plataforma de monitoreo. Este proceso no solo tiene como objetivo la modernización de la plataforma, sino también la expansión de su alcance hacia nuevos clientes mediante la oferta de un servicio diferenciado.

Con el objetivo SMART de habilitar nuevas funcionalidades mediante la migración a microservicios y ampliar el alcance a nuevos clientes para fines de 2023, se propuso un novedoso sistema de planes basado en Java. Este sistema busca segmentar las funcionalidades y los niveles de acceso otorgados a los usuarios dentro de la plataforma de monitoreo, con la finalidad de proporcionar un servicio personalizado con tarifas diferidas. A pesar de que la implementación de este sistema se vio demorada debido al aplazamiento de la migración de arquitectura para el próximo año, es crucial destacar los logros alcanzados durante este proyecto, los cuales incluyen aportes en la célula de desarrollo basados en la experiencia de introducir a un nuevo integrante, cuestionar el funcionamiento de algunos procesos para reconocer oportunidades de mejora, evidenciar la importancia de la documentación del equipo para enfrentar cualquier inconveniente futuro, y evidenciar la gran capacidad que tiene Ceptinel para recopilar datos de uso de sus clientes, los cuales entregarán la base para futuros proyectos.

En un esfuerzo paralelo, se llevó a cabo una evaluación de la construcción de los microservicios, evidenciando que el 90% las tecnologías y prácticas recomendadas por la literatura fueron implementadas con éxito. Además, se implementaron mejoras sustanciales en el uso de los recursos en la nube, lo que se reflejó en un ahorro del 4,7% de los costos en el último mes capturado. Adicionalmente, se habilitaron métricas de consumo y ejecución dentro de la plataforma, proporcionando una nueva forma de monitorear el consumo de los clientes y de levantar métricas para distintos procesos dentro de la plataforma.

Abstract

Founded in 2019, Ceptinel is a company specialized in regulatory technologies and transactional monitoring, with consolidated presence in several countries, including Chile, Colombia, Panama and Mexico. At the beginning of this year, it started a migration process towards a new software architecture called microservices, aimed at addressing scalability and maintenance challenges in its monitoring platform. This process not only aims to modernize the platform, but also to expand its reach to new clients by offering a differentiated service.

With the SMART objective of enabling new functionalities by migrating to microservices and expanding the reach to new clients by the end of 2023, an innovative Java-based plan system was developed. This system seeks to segment the functionalities and access levels granted to users within the monitoring platform, in order to provide a personalized service with deferred rates. Although the implementation of this system was delayed due to the postponement of the architecture migration to next year, it is crucial to highlight the achievements made during this project, which include contributions in the development cell based on the experience of introducing a new member, questioning the operation of some processes to recognize opportunities for improvement, evidencing the importance of team documentation to face any future inconvenience, and evidencing the great capacity that Ceptinel has to collect usage data from its customers, which will provide the basis for future projects.

In a parallel effort, an evaluation of the construction of the microservices was carried out, showing that 90% of the technologies recommended by the literature were successfully implemented. In addition, substantial improvements were implemented in the use of cloud resources, which was reflected in a 4.7% cost savings in the last month captured. In addition, consumption and execution metrics were enabled within the platform, providing a new way to monitor customer consumption and to collect metrics for different processes within the platform.

Introducción

En el presente informe, se seguirá una convención específica para las referencias y las definiciones de términos técnicos con el objetivo de mejorar la claridad y la accesibilidad dentro del documento. Las referencias serán encapsuladas de la siguiente manera: [1], [2], ..., y las definiciones de términos técnicos se proporcionarán al pie de página para facilitar la comprensión del lector.

Contexto de la empresa

Ceptinel es una empresa *RegTech*¹ fundada el 2019 que opera en Chile, Colombia, Panamá y México. Su misión es apoyar a las empresas a combatir la corrupción, prevenir el lavado de activos y detectar a tiempo las malas prácticas asociadas a fraudes financieros tanto internos como externos. Su visión es llegar a convertirse en líderes en el mercado internacional de *RegTech*.

La oficina principal se encuentra en Santiago, ubicada en la comuna de Providencia a pasos del metro Los Leones. La empresa cuenta con 16 trabajadores los cuales se dividen en distintas células de trabajo, tales como célula de propuestas comerciales, innovación de producto y desarrollo. También cuenta con trabajadores en el extranjero que brindan apoyo en el día a día.

Durante la pasantía se colaboró activamente en la célula de desarrollo, un equipo multifuncional encargado de construir software según los requisitos de los proyectos y de abordar los errores reportados por los usuarios. Esto gracias al uso de distintas tecnologías proporcionadas por empresas aliadas como Microsoft Azure (Proveedor de servicios *Cloud*²), Oracle (Proveedor del lenguaje de programación Java SE³) y VMware (Proveedor de *Spring Boot*⁴ para desarrollo de aplicaciones Enterprise basadas en Java).

Entre los servicios que ofrece la empresa se encuentra la plataforma de monitoreo transaccional, un software diseñado para la detección de fraudes, que constituye la principal fuente de ingresos, este servicio incluye un plan de implementación y un plan mensual abonado por los clientes. Además, Ceptinel brinda la posibilidad de desplegar máquinas virtuales⁵ en Azure para la privacidad de los datos de cada cliente, generando ingresos adicionales por el cobro de la administración de servicios *Cloud*.

¹ Tecnologías regulatorias, ayudan a las organizaciones a cumplir con requisitos normativos y legales.

² Servicios en la nube, ofrecen recursos de computación, almacenamiento y redes a través de Internet.

³ *Java Standard Edition*, proporciona las herramientas necesarias para el desarrollo de aplicaciones Java.

⁴ *Spring Boot* simplifica el desarrollo, configuración e implementación de aplicaciones basadas en *Spring*, un marco de trabajo que ofrece herramientas y buenas prácticas para construir aplicaciones Java flexibles y escalables.

⁵ Es un ordenador que opera completamente a través de software en lugar de hardware físico.

Contexto del problema y oportunidad

Esta plataforma fue lanzada en 2018 y ha experimentado un crecimiento continuo utilizando una arquitectura de software conocida como **monolítica**. Esta arquitectura implica que todos los procesos y funcionalidades coexisten dentro de una sola aplicación. Bajo este enfoque, se han desarrollado diversos algoritmos que posibilitan la edición, transformación y exportación de grandes volúmenes de datos.

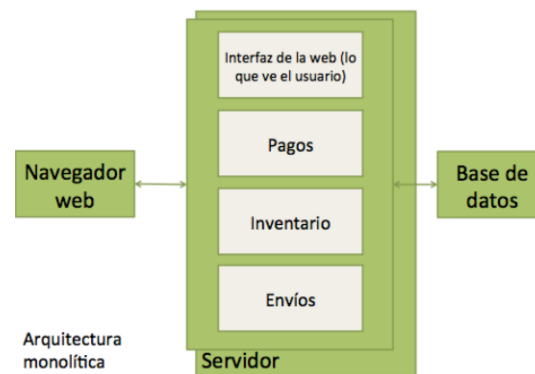


Ilustración 1 – Ejemplo de arquitectura monolítica para una tienda virtual [2]

Con el tiempo, se incorporaron diversas funcionalidades a la plataforma con el objetivo de ofrecer soluciones variadas a los clientes, esta expansión trajo consigo un aumento en el tamaño del código fuente⁷, generando una mayor interdependencia entre distintos módulos. Esto, a su vez, resultó en una mayor dificultad en la lectura del código, un esfuerzo adicional para realizar cambios de alto nivel y una complejidad general mayor de la aplicación, provocando un aumento en los tiempos necesarios para lanzar nuevas versiones.

Otro desafío que surgió fue la necesidad de actualizar el lenguaje de programación utilizado (Java) desde su versión 8 a la versión 17, con motivo de dejar atrás todas las herramientas que quedaron deprecadas⁸ con el tiempo. Se optó por la versión 17 ya que a principios de 2023 esta era la última con LTS⁹ y la que les permite acceder a mejores prestaciones y nuevas funciones. No obstante, esta actualización conlleva múltiples consideraciones: algunas características de Java pueden no ser compatibles con la nueva versión, y algunas dependencias pueden ser discontinuadas o reemplazadas por otras. Por lo tanto, cambiar de versión en una aplicación de tamaño Enterprise como Ceptinel implica una revisión exhaustiva de su código y un esfuerzo considerable para abordar posibles errores emergentes.

⁷ Contiene los algoritmos, lógica de programación y todos los elementos necesarios para que una aplicación funcione.

⁸ Que no posee soporte o que simplemente ya no funciona, coloquialmente conocido como “código muerto”.

⁹ Long Term Support, corresponden a las versiones que cuentan con soporte oficial para los años venideros.

Ceptinel también tiene como objetivo la reducción de sus costos operativos, dentro de los cuales se incluye el uso de recursos *Cloud*, dado que la empresa trabaja con datos sensibles, como transacciones bancarias con el detalle completo de cada persona, por ley se les prohíbe almacenar datos directamente. En su lugar, deben trabajar dentro de los servidores de cada cliente o implementar máquinas virtuales para garantizar la integridad de los datos y cumplir con estándares de ciberseguridad. Actualmente Ceptinel contrata servicios IaaS [Véase Anexo 1] en Azure para desplegar máquinas virtuales de alta capacidad a clientes que eligen esta opción. Sin embargo, esto conlleva altos costos mensuales en procesamiento, almacenamiento y capacidad de cada máquina, lo que resulta en gastos innecesarios si un cliente tiene bajo consumo.

Finalmente, en un mercado altamente competitivo, Ceptinel ha enfrentado desafíos al intentar atraer nuevos clientes [Véase Anexo 2]. Muchas de las grandes empresas ya cuentan con sistemas de monitoreo transaccional con contratos de al menos 5 años. Por otro lado, las empresas más pequeñas encuentran difícil asumir el costo de un sistema tan complejo y costoso como el de Ceptinel, y algunas no están dispuestas a hacerlo. Dada la competencia internacional que posee la empresa, surge la necesidad de entregar un servicio diferenciado.

Ante las problemáticas identificadas, la empresa realizó un análisis exhaustivo y reconoció la necesidad de realizar una reestructuración. Considerando los intereses de los inversionistas y las capacidades de la célula de desarrollo, se tomó la decisión de llevar a cabo un cambio disruptivo dentro de la empresa: Migrar a una nueva arquitectura de software.

A principios de este año, la empresa comenzó el proceso de migración a una arquitectura conocida como **microservicios**. Esta nueva arquitectura busca desacoplar los procesos y las funcionalidades principales de la aplicación en diferentes “microservicios”, permitiendo el desarrollo y despliegue independiente de cada uno. Esta modularidad facilita tanto el desarrollo como el escalamiento de cada componente, también posibilita la incorporación de nuevas funcionalidades gracias al proceso de migración.

Es esencial considerar que una migración no solo implica la actualización del código existente, sino que también brinda la oportunidad de implementar mejoras significativas. En algunos casos, se aprovecha la migración para realizar cambios sustanciales en determinados módulos, llegando incluso a reconstruirlos desde cero.

A continuación, se presenta una ilustración que compara ambas arquitecturas, destacando que en el caso del monolito¹⁰, toda la aplicación recibe las peticiones del usuario, las delega a las secciones correspondientes y luego se procesan en una única base de datos. En cambio, con la arquitectura de microservicios, cada una de estas secciones de código se separa en microservicios individuales. Cada microservicio puede tener su propia base de datos, permitiendo así un funcionamiento independiente entre ellos.

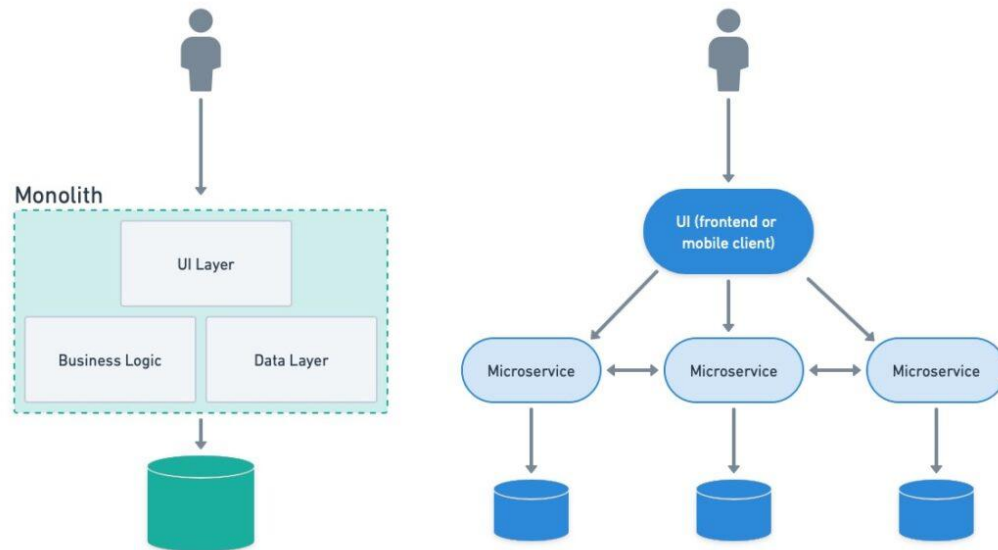


Ilustración 2 – Arquitectura monolítica vs microservicios [3]

Una ventaja significativa para la empresa es su transición desde una arquitectura monolítica, lo que facilita la identificación de los servicios que pueden desacoplarse y convertirse en microservicios. Además, existe un conocimiento claro del dominio de datos¹¹ y la lógica que cada componente debe tener.

En la actualidad, la empresa ha avanzado significativamente en la migración, encontrándose en la fase de ejecución del proceso. En el análisis de negocio inicial realizado por Ceptinel, se identificaron 31 microservicios que deben construirse [Véase Anexo 3], y se ha definido el siguiente modelo de arquitectura:

¹⁰ En este contexto, es una forma abreviada de referirse a una aplicación construida bajo una arquitectura de software monolítica (de una sola pieza)

¹¹ Consiste en que cada microservicio se encarga de una tarea específica, y, por ende, de los datos asociados a esta.

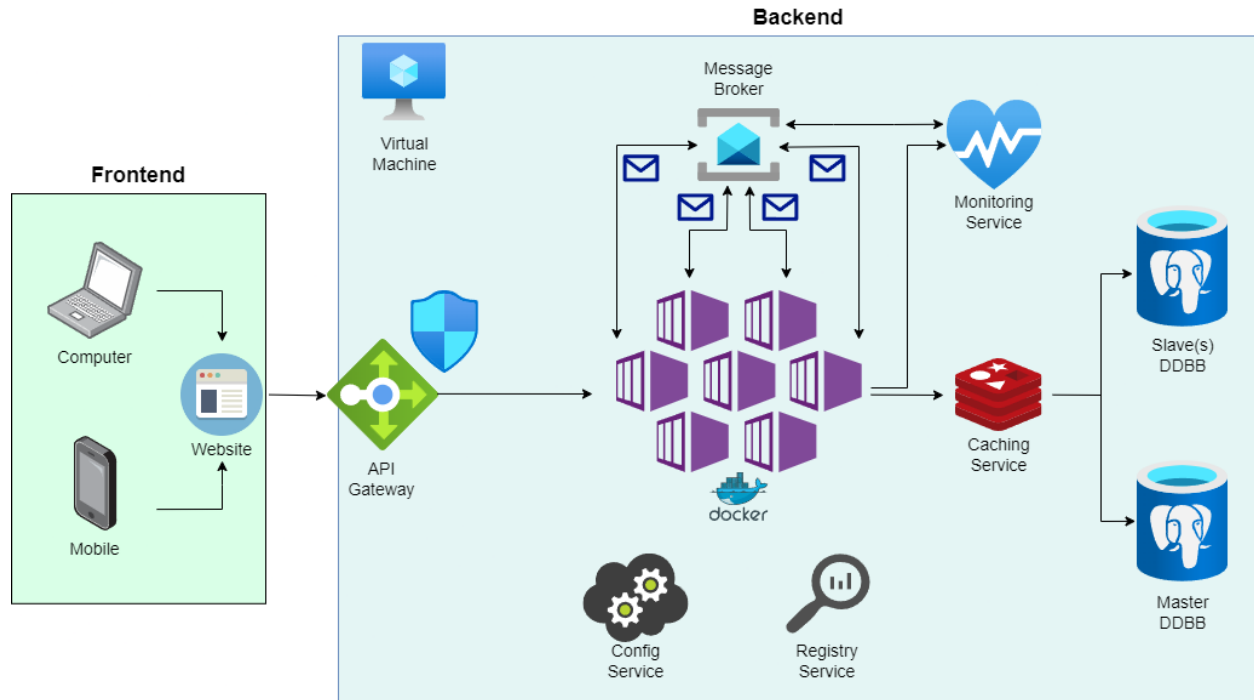


Ilustración 3 – Diagrama de arquitectura actual de microservicios en Ceptinel. Fuente: Elaboración propia

El detalle de cada uno de los componentes presentados se encuentra en el anexo [Véase Anexo 4], es importante notar que la arquitectura de microservicios está mucho más enfocada en el Backend¹² que en el Frontend¹³.

¹² Es la parte encargada de administrar la funcionalidad general de un programa o aplicación web. Cuando el usuario interactúa con el front end, la interacción envía una solicitud al back end

¹³ Es la interfaz gráfica de usuario (GUI) con la que los usuarios pueden interactuar de forma directa, como los menús de navegación, los elementos de diseño, los botones, las imágenes y los gráficos

Objetivos

Luego de conocer las razones por las cuales se produjo la migración, y tras haber determinado que esta logra atacar los principales dolores de la empresa, se decidió orientar la pasantía en aprovechar las nuevas opciones que trae microservicios y trabajar junto a la célula de desarrollo para participar en la migración desde el sistema monolítico a microservicios.

Objetivo general SMART

Habilitar nuevas funcionalidades para la plataforma de monitoreo transaccional aprovechando la migración a microservicios, con el objetivo de ofrecer un servicio diferenciado a los clientes y aumentar el alcance a estos dentro de un plazo de 3 meses.

Objetivos específicos

1. Evaluar la implementación actual de microservicios para garantizar que cumple con al menos el 80% de las buenas prácticas sugeridas por la literatura, con el fin de asegurar un despliegue estable de los microservicios y una construcción adecuada de estos.
2. Evaluar el rendimiento de los microservicios en comparación con la arquitectura monolítica anterior, con el objetivo de lograr una mejora del 15% en los tiempos de respuesta del motor de reglas, que constituye el mayor consumo dentro de la plataforma.
3. Analizar los recursos Cloud utilizados en la empresa para reajustar las máquinas virtuales asignadas a algunos clientes, identificando posibles oportunidades de ahorro y logrando una reducción del 10% en costos Cloud mensuales.

Medidas de desempeño

1. Se llevará a cabo un análisis detallado de las tecnologías utilizadas en microservicios, comparándolas con las recomendaciones encontradas en estudios y literatura especializada. El objetivo de esta evaluación es identificar la cobertura actual de las tecnologías asociadas a los módulos esenciales, tales como servicio de configuración, registro y comunicación entre microservicios. Se determinará qué tecnologías están implementadas de manera adecuada y cuáles requieren definición o implementación adicional.

$$\text{Porcentaje de cobertura} = \frac{\text{Módulos habilitados}}{\text{Total de módulos por habilitar}}$$

2. Se llevará a cabo una simulación del uso del motor de reglas en ambas arquitecturas. El objetivo principal es determinar si la migración a microservicios resulta en una mejora para los tiempos de respuesta. Se ha establecido una mejora del 15% como un objetivo alcanzable, considerando la implementación de nuevas estructuras de datos y algoritmos más eficientes durante el proceso de migración. Como propósito secundario, se busca evaluar la estabilidad de los microservicios.

$$\text{Comparación de rendimiento} = \frac{\text{Tiempo de respuesta Microservicios}}{\text{Tiempo de respuesta Monolito}}$$

3. Se llevará a cabo una revisión exhaustiva de los recursos Cloud de la empresa, incluyendo el tipo de máquinas virtuales de cada cliente, con el objetivo de identificar conceptos subutilizados o sin uso. Esta revisión tiene como meta la reducción gradual de costos, por lo que se definió alcanzar un ahorro del 10% pensando en la aparición de nuevos conceptos Cloud que podrían afectar a esta medida. Se tomarán los 6 últimos meses del estudio y se comparará la evolución de los costos mensuales hasta el final de la pasantía.

$$\text{Variación de los costos} = \frac{\text{Costo del último mes}}{\text{Costo promedio de los últimos 6 meses}}$$

Estado del arte

El análisis del estado del arte se centrará principalmente en los microservicios, ya que es crucial garantizar una arquitectura sólida antes de implementar nuevas funcionalidades. A continuación, exploraremos las múltiples consideraciones necesarias para lograr establecer un sistema distribuido como los microservicios. En este contexto, el principal desafío se encuentra en abordar la complejidad operativa propia de esta nueva arquitectura.

La ventaja de usar microservicios es la flexibilidad de su diseño. No existe un número óptimo de microservicios ni una estructura fija a seguir; todo dependerá de las necesidades específicas del negocio. En cuanto a las tecnologías a utilizar, es necesario abordar diversas consideraciones:

- ¿Cómo se gestionarán las solicitudes de los usuarios por parte de los microservicios?
- ¿Se anticipa el procesamiento de grandes cantidades de datos?
- ¿Es suficiente contar con una única base de datos para cubrir las necesidades?
- ¿Cómo se garantizará la disponibilidad y la continuidad operativa de los microservicios?

Dependiendo de las consideraciones específicas del negocio, será necesario implementar distintas soluciones, en la siguiente imagen se muestra un ejemplo de arquitectura de microservicios, evidenciando que no siempre se utilizan las mismas tecnologías a través de la industria.

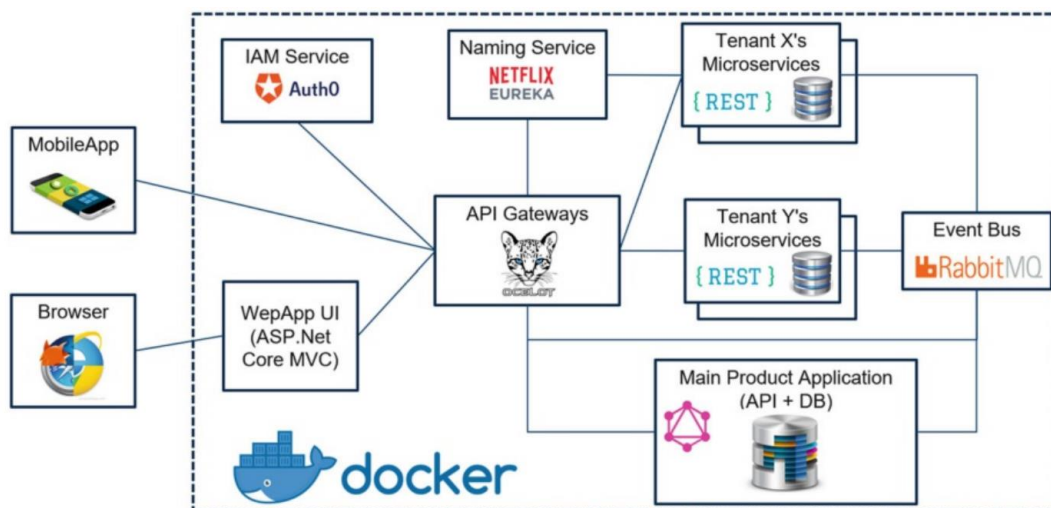
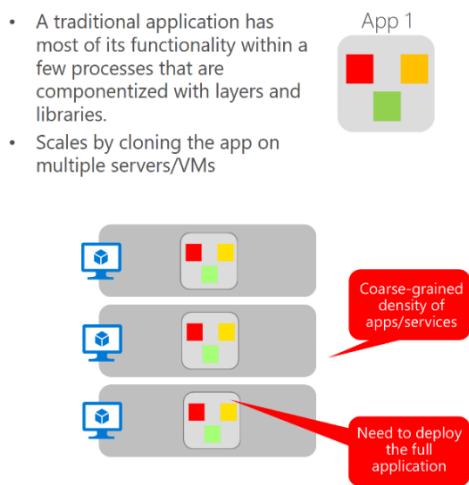


Imagen 2 - Ejemplo de un diagrama de arquitectura basada en microservicios [4]

Aunque existe la libertad de elegir libremente las tecnologías para microservicios, resulta crucial explorar las opciones más populares en el rubro. En un estudio realizado entre 36 empresas de software en Brasil [5] se llevó a cabo una encuesta para recopilar información sobre las tecnologías y prácticas más utilizadas para lograr despliegues exitosos de microservicios. Este estudio incluyó aspectos como la comunicación entre microservicios, herramientas de desarrollo, monitoreo, entre otros. [Véase Anexo 5].

Con lo anterior vimos las consideraciones para la construcción de los microservicios, ahora, veremos las consideraciones para la implementación de una aplicación basada en estos, que contrario a una aplicación monolítica, puede funcionar utilizando distintas combinaciones de microservicios en base a las funcionalidades que se quieren habilitar, como también tener múltiples instancias¹⁴ de estos para responder mejor a grandes cantidades de peticiones específicas a un solo microservicio.

Monolithic deployment approach



Microservices application approach

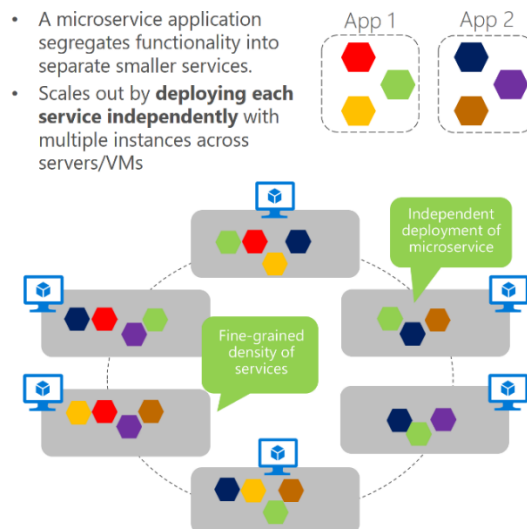


Ilustración 4 – Arquitectura monolítica vs microservicios, comparación por escalamiento [6]

Generalmente, la instalación de una aplicación monolítica implica cargarla en la nube, configurarla junto con su base de datos y realizar pruebas para verificar un despliegue exitoso, **hacer esto manualmente con 31 microservicios ya no resulta práctico**. Esto se agrava aún más si cada microservicio tiene su propia configuración y base de datos.

¹⁴ Se refiere a una ejecución específica de un microservicio, en el caso de múltiples instancias cada una trabaja independiente de las otras

Por esta razón, el equipo ha optado por adoptar contenedores *Docker*. Estos contenedores sirven como una forma de aislar cada uno de los microservicios. Operan como unidades encapsuladas que incluyen todos los elementos necesarios para que una aplicación funcione: el código de la aplicación, las bibliotecas, herramientas y otros componentes. La premisa es que los contenedores garantizan un funcionamiento coherente en cualquier entorno que tenga Docker instalado, lo que aporta significativamente a la gestión de los microservicios.

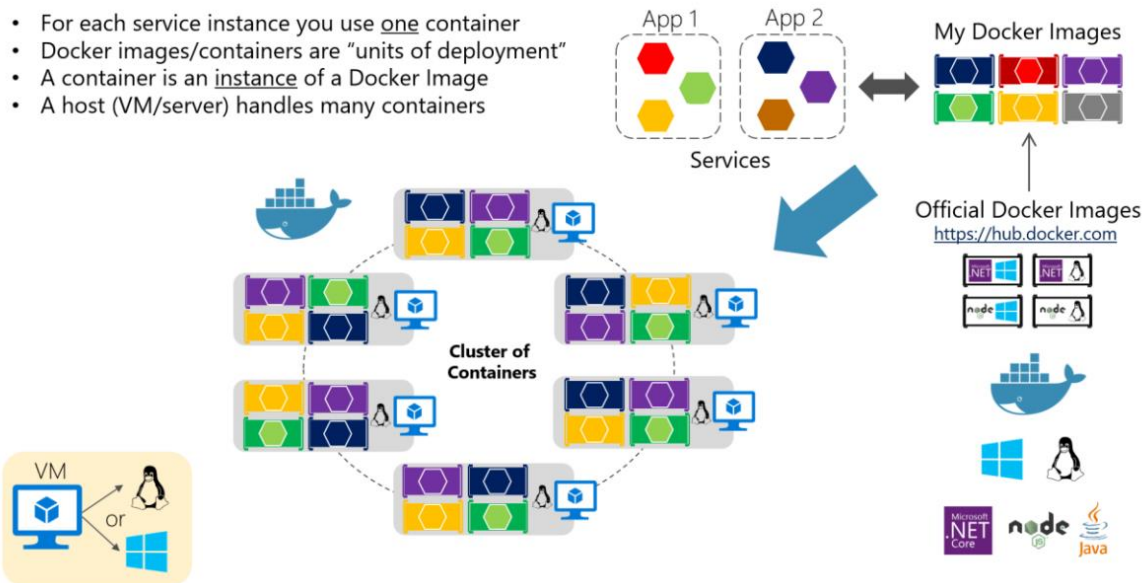


Ilustración 5 – Uso de contenedores para despliegue de microservicios [7]

En la esquina inferior derecha de la ilustración, se observa el proceso de contenerización de una aplicación, como por ejemplo una aplicación basada en Java. Este proceso implica empaquetar la aplicación, junto con sus dependencias y el sistema operativo en el que opera (puede ser Windows o Linux), en una imagen de contenedor. Esta imagen se convierte en un paquete autónomo que contiene todo lo necesario para que la aplicación funcione de manera consistente en diferentes entornos.

A partir de esta imagen de contenedor, se pueden crear instancias de contenedor, que son ejecuciones individuales de la aplicación contenida. Ahora, esto sigue teniendo el problema de tener que manejar manualmente cada uno de estos contenedores dentro de una máquina virtual, esto puede resolverse mediante el uso de *Kubernetes*, el cual corresponde a un orquestador de contenedores de alto nivel, el cual también permite controlar las máquinas virtuales en donde están instalados.

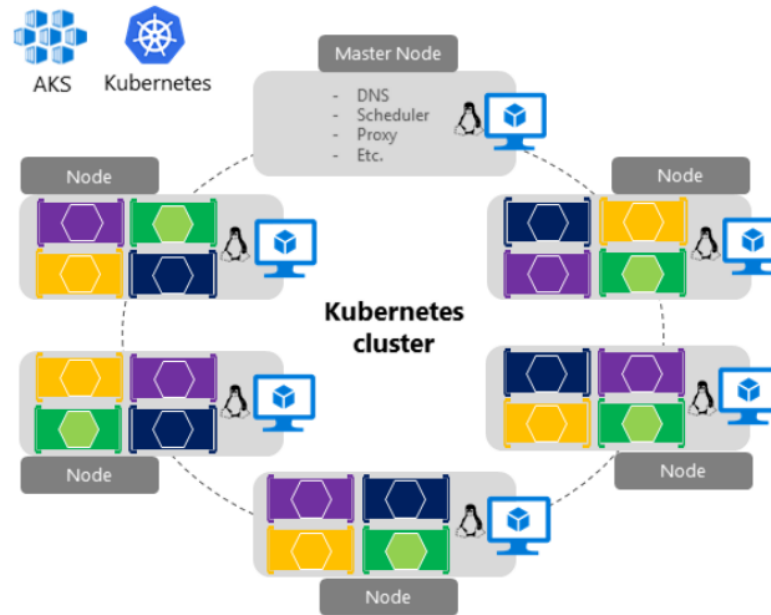


Ilustración 6 – Estructura de clusters de Kubernetes simplificada [8]

Una vez conocida la forma de construir y desplegar microservicios, debemos tener en consideración cuales son las mejores prácticas de desarrollo a considerar. A continuación, se incluyen los principales aspectos para una salida a producción exitosa con aplicaciones basadas en microservicios [9]:

- Monitoreo y chequeo de salud del servicio y la infraestructura.
- Infraestructura escalable para los servicios (Cloud y orquestados).
- Diseño de seguridad e implementación en múltiples niveles: Autenticación, autorización, manejo de secretos, comunicación segura, etc.
- Centrarse en una entrega de aplicación rápida, en el futuro se debe apuntar a dedicar un equipo de trabajo enfocado en cada microservicio.
- Finalmente, mantener buenas prácticas de DevOps¹⁵, CI/CD¹⁶ e infraestructura.

En general, la literatura favorece enormemente el uso de microservicios, pero eso no significa que sean infalibles, a principios de este año Amazon Prime Video Streaming [10] tuvo que retroceder de una arquitectura de microservicios a una monolítica debido al alto costo que trajo operar con microservicios a gran escala, dado los constantes cuellos de botella al procesar grandes volúmenes de datos.

¹⁵ Se enfoca en mejorar la colaboración entre el equipo de desarrollo de software y el de operaciones

¹⁶ Integración continua / Despliegue continuo, se enfoca en mejorar la eficiencia y la calidad del desarrollo

En definitiva, esta nueva arquitectura posee muchos beneficios, pero debe usarse correctamente bajo un contexto que efectivamente requiera de los microservicios, por lo que Ceptinel debe observar cómo se desenvuelve esta nueva arquitectura y determinar si es factible continuar con su operación en los años siguientes.

Solución escogida

Considerando las nuevas capacidades que entrega microservicios y la necesidad de alcanzar a distintos clientes, se levantan las siguientes propuestas:

- Aumentar el nivel de especialización de la plataforma de monitoreo por medio de la integración con un proyecto paralelo, basado en un motor de búsqueda de personas en listas negras internacionales y expuestas políticamente.
- Dividir el actual motor de reglas en distintos microservicios para ofrecer un servicio granular a los clientes.
- Crear un nuevo sistema de planes para dividir las funcionalidades entregadas a los clientes, manteniendo el motor de reglas completo en un solo microservicio pero limitando su acceso.

Respecto a la **primera opción**, el motor de búsqueda surgió por la importancia dentro del rubro de conocer qué personas tienen un perfil riesgoso, para así tener un mayor enfoque en sus movimientos financieros y reportarlos a las autoridades en caso de que corresponda, por lo que incluirlo dentro de la plataforma puede ser beneficioso al ser este un producto muy solicitado por los clientes.

Sin embargo, no se puede determinar el poder de ventas que esta solución potencialmente tenga al estar todavía en su fase de desarrollo, además, solo resuelve parcialmente la problemática, ya que por un lado agrega funcionalidades muy demandadas en el mercado, pero se sigue vendiendo una plataforma muy compleja para algunos potenciales clientes.

La **segunda opción** apunta a ofrecer solamente aquellos procesos y funcionalidades que el cliente requiere, por medio de la instalación selectiva de microservicios extraídos del motor de reglas, esta opción permite reducir la cantidad de componentes no utilizados y facilitar el uso de cada transformación individual según se necesite.

La dificultad de esto está al momento del despliegue, ya que fácilmente se pueden desprender 40 microservicios solamente del motor de reglas, y dependiendo de cada transformación se requerirán múltiples transacciones al resto de los microservicios, lo cual empeorará la eficiencia y aumentará mucho la complejidad del sistema. Se consideró analizar esta opción para demostrar que la excesiva granularidad de los microservicios puede ser perjudicial en algunos casos.

Dicho lo anterior, la **tercera opción** supone dividir las funcionalidades por medio de planes, contrario a la opción anterior, no se divide el motor de reglas, sino que se limitan los accesos y permisos entregados a los usuarios dependiendo del plan que tengan contratado, logrando una personalización del servicio y un mayor control de lo que los usuarios pueden usar.

Por lo tanto, se escoge la tercera opción como la solución propuesta, el motor de búsqueda seguirá desarrollándose como un producto independiente y la migración a microservicios continuará según lo planeado.

Es importante notar que no era necesario una migración de arquitectura para lograr implementar un sistema de planes, sin embargo, hubiera resultado muy difícil de lograr dado los desafíos que la empresa estaba enfrentando con su arquitectura anterior, fue gracias a la migración de arquitectura que se ha podido considerar nuevos proyectos y mejoras, por lo que se elige esta opción como la más valiosa a desarrollar.

Evaluación económica

Para la evaluación económica se consideraron dos escenarios, uno sin proyecto, donde se sigue trabajando con la actual arquitectura monolítica sin introducir nuevas funcionalidades, y uno con proyecto, que corresponde a trabajar con la arquitectura de microservicios e incorporar el nuevo sistema de planes.

En el escenario con proyecto, se incorporaron tres nuevas consideraciones. Primero, se adquieren más clientes en comparación con el escenario original. Segundo, la implementación de un sistema de planes resulta en ingresos diferidos, proporcional a la cantidad de clientes que eligen cada plan. Tercero, se logra una reducción del 20% en los costos de la nube gracias a un uso más eficiente de estos.

Los costos principales considerados son el uso de recursos en la nube y los salarios de los trabajadores. Los costos en la nube abarcan el pago por máquinas virtuales, almacenamiento, respaldos y grupos de recursos necesarios. Por otro lado, se incluyen los salarios de los trabajadores de diversas áreas esenciales para la operación de la plataforma de monitoreo transaccional.

Supuestos considerados (detalle completo incluido en el Anexo N° 6):

- El año 1 corresponde al 2024.
- Se considera solo los trabajadores, licencias y activos mínimos para que la empresa pueda operar en ambos escenarios.
- Se asume que la situación con proyecto comienza con la arquitectura microservicios ya implementada.
- Se asume que anteriormente la empresa solo vendía planes 'Premium' de la plataforma a sus clientes.
- Se considera que se existen tres planes en el nuevo sistema: Básico, Enterprise y Premium.
- Se consideró un aumento gradual del equipo y un cambio de oficina en el año 3 en ambos escenarios.
- El sueldo de un trabajador nuevo corresponde al sueldo promedio de todos los trabajadores del primer año.
- Al ser un proyecto de cambio en una empresa en marcha, no se consideran inversiones iniciales, capitales de trabajo iniciales, ni pago por préstamos, sino que el flujo está centrado principalmente en resultados no operacionales.
- No se consideran pérdidas acumuladas de periodos anteriores al ser estas externas al proyecto.
- El periodo a considerar es de 5 años, ya que la empresa planea venderse en el futuro para cerrar el ciclo de vida de lo que alguna vez fue el emprendimiento original.

Flujo de caja sin proyecto

Flujos de caja sin proyecto (Arquitectura monolítica)					
*Evaluación SOLO de mantener operando la situación actual, no considera nuevas funcionalidades al asumir que no son posibles realizar					
* Año 1 comienza desde el 2024	Año 1	Año 2	Año 3	Año 4	Año 5
Ingresos por ventas	\$ 432,000,000	\$ 669,600,000	\$ 1,134,000,000	\$ 1,674,000,000	\$ 2,187,000,000
Costos fijos	\$ -254,700,000	\$ -254,700,000	\$ -274,230,909	\$ -274,230,909	\$ -288,776,364
Costos variables	\$ -118,054,800	\$ -156,675,600	\$ -233,917,200	\$ -330,469,200	\$ -427,021,200
Depreciaciones legales			\$ -500,000	\$ -500,000	\$ -1,500,000
Resultado Operacional	\$ 59,245,200	\$ 258,224,400	\$ 625,351,891	\$ 1,068,799,891	\$ 1,469,702,436
Pérdidas del ejercicio anterior	\$ -	\$ -	\$ -	\$ -	\$ -
Resultado no Operacional	\$ 59,245,200	\$ 258,224,400	\$ 625,351,891	\$ 1,068,799,891	\$ 1,469,702,436
Utilidad antes impuestos	\$ 59,245,200	\$ 258,224,400	\$ 625,351,891	\$ 1,068,799,891	\$ 1,469,702,436
Impuesto de Primera Categoría (27%)	\$ 15,996,204	\$ 69,720,588	\$ 168,845,011	\$ 288,575,971	\$ 396,819,658
Utilidad después impuestos	\$ 43,248,996	\$ 188,503,812	\$ 456,506,880	\$ 780,223,920	\$ 1,072,882,779
Depreciaciones legales	\$ -	\$ -	\$ 500,000	\$ 500,000	\$ 1,500,000
Flujo Operacional	\$ 43,248,996	\$ 188,503,812	\$ 457,006,880	\$ 780,723,920	\$ 1,074,382,779
Valor residual de los activos	\$ -	\$ -	\$ -	\$ -	\$ 4,200,000
Capital de trabajo	\$ -	\$ -2,500,000		\$ -4,000,000	
Recuperación del capital de trabajo	\$ -	\$ -	\$ -	\$ -	\$ -
Flujo Capitales	\$ -	\$ -2,500,000	\$ -	\$ -4,000,000	\$ 4,200,000
Flujo de Caja Privado	\$ 43,248,996	\$ 186,003,812	\$ 457,006,880	\$ 776,723,920	\$ 1,078,582,779
	Tasa descuento	11.25%		VAN	\$1,661,069,543
				TIR	#N!NUM!

Tabla 1 – Flujos de caja sin proyecto (Elaboración propia)

	Proyección clientes	Diferencial
Año 1	20	0
Año 2	30	10
Año 3	50	20
Año 4	75	25
Año 5	100	25

Tabla 2 – Clientes proyectados sin proyecto

Sin proyecto							
Proyección de cliente	Año 1	Año 2	Año 3	Año 4	Año 5	VAN	\$1,661,069,543
N° clientes totales	20	30	50	75	100		
Proyección de cliente	Año 1	Año 2	Año 3	Año 4	Año 5	VAN	\$1,032,318,337
N° clientes totales	20	25	35	50	75		
Proyección de cliente	Año 1	Año 2	Año 3	Año 4	Año 5	VAN	\$ 586,096,153
N° clientes totales	20	22	27	37	50		

Tabla 3 – Diferencia de VAN variando número de clientes

Flujo de caja con proyecto

Flujos de caja con proyecto (Arquitectura microservicios)					
* Año 1 comienza desde el 2024	Año 1	Año 2	Año 3	Año 4	Año 5
Ingresos por ventas	\$343,910,000	\$617,347,500	\$ 1,056,095,000	\$ 1,657,937,500	\$ 2,228,770,000
Costos fijos	\$ -254,700,000	\$ -254,700,000	\$ -277,867,273	\$ -277,867,273	\$ -294,230,909
Costos variables	\$ -85,019,760	\$ -123,435,144	\$ -187,460,784	\$ -277,096,680	\$ -366,732,576
Depreciaciones legales			\$ -980,000	\$ -980,000	\$ -1,400,000
Resultado Operacional	\$ 4,190,240	\$ 239,212,356	\$ 589,786,943	\$ 1,101,993,547	\$ 1,566,406,515
Pérdidas del ejercicio anterior	\$ -	\$ 4,190,240	\$ -	\$ -	\$ -
Resultado no Operacional	\$ 4,190,240	\$ 243,402,596	\$ 589,786,943	\$ 1,101,993,547	\$ 1,566,406,515
Utilidad antes impuestos	\$ 4,190,240	\$ 243,402,596	\$ 589,786,943	\$ 1,101,993,547	\$ 1,566,406,515
Impuesto de Primera Categoría (27%)		\$ 65,718,701	\$ 159,242,475	\$ 297,538,258	\$ 422,929,759
Utilidad después impuestos	\$ 4,190,240	\$ 177,683,895	\$ 430,544,469	\$ 804,455,290	\$ 1,143,476,756
Depreciaciones legales	\$ -	\$ -	\$ 980,000	\$ 980,000	\$ 1,400,000
Flujo Operacional	\$ 4,190,240	\$ 177,683,895	\$ 431,524,469	\$ 805,435,290	\$ 1,144,876,756
Valor residual de los activos	\$ -	\$ -	\$ -	\$ -	\$ 7,000,000
Capital de trabajo	\$ -	\$ -4,900,000		\$ -6,300,000	
Recuperación del capital de trabajo	\$ -	\$ -	\$ -	\$ -	\$ -
Flujo Capitales	\$ -	\$ -4,900,000	\$ -	\$ -6,300,000	\$ 7,000,000
Flujo de Caja Privado	\$ 4,190,240	\$ 172,783,895	\$ 431,524,469	\$ 799,135,290	\$ 1,151,876,756
	Tasa descuento	11.25%		VAN	\$1,654,412,631
				TIR	#N!NUM!

Tabla 4 – Flujos de caja con proyecto (Elaboración propia)

	Proyección clientes	Diferencial
Año 1	20	0
Año 2	35	15
Año 3	60	25
Año 4	95	35
Año 5	130	35

	Proporción
Plan básico	20%
Plan Enterprise	10%
Plan Premium	70%

Tabla 5 – Clientes proyectados con proyecto

Proyección de client	Año 1	Año 2	Año 3	Año 4	Año 5		VAN	\$1,654,412,631
Nº clientes totales	20	35	60	95	130			
Proyección de client	Año 1	Año 2	Año 3	Año 4	Año 5		VAN	\$1,293,519,077
Nº clientes totales	20	35	60	95	80			
Proyección de client	Año 1	Año 2	Año 3	Año 4	Año 5		VAN	\$100,705,097
Nº clientes totales	20	22	24	27	30			

Tabla 6 – Diferencia de VAN variando número de clientes

Análisis de la evaluación

Comenzando con la situación sin proyecto, observamos flujos positivos que reflejan ingresos considerables al incorporar más clientes. Esto se debe a las ganancias derivadas de la gestión de los recursos Cloud de los clientes y el pago mensual por la licencia de la plataforma.

Por otro lado, analizar el escenario con proyecto fue más desafiante debido a la introducción de un nuevo sistema de planes, lo cual conlleva diversas consideraciones y tipos de ganancias. Fue necesario estimar precios realistas para cada plan, determinar qué porcentaje del total de clientes optaría por cada plan y reducir costos variables para acercarse lo más posible al escenario sin proyecto.

Aunque el escenario presentado con proyecto es muy optimista, existen varias maneras de mejorarlo. Por ejemplo, se podría considerar la implementación de soluciones *multitenant*¹⁷ para tener múltiples clientes en una sola máquina virtual y ahorrar costos. No obstante, esta opción plantea preocupaciones sobre la seguridad de los datos, especialmente cuando se trabaja con información sensible. Una alternativa sería mantener cada base de datos de cliente fuera de la máquina virtual, pero debe ser evaluado por la empresa.

En definitiva, el escenario presentado es solo uno de los muchos posibles, pero refleja la posibilidad de alcanzar una solución factible si se logran ciertas condiciones, las cuales serán buscadas lograr por la empresa.

¹⁷ Múltiples inquilinos, ocurre cuando varios usuarios comparten una misma solución u entorno virtual.

Métodología

Se asume que la arquitectura de microservicios está operativa para la construcción del sistema de planes. El nuevo sistema ofrecerá tres planes principales: Básico, Enterprise y Premium, cada uno con funcionalidades y permisos específicos. Además, se venderán algunas transformaciones y 'cajas' del motor de reglas como plugins adicionales no incluidos en los planes.

A continuación, se describen los cinco pasos recomendados para establecer un sistema de planes como un microservicio, basándose en los pasos aplicados por la empresa mediante metodología ágil Scrum. Se adjunta en cada paso las tecnologías sugeridas para su implementación.

Primer paso: Levantar historias de usuario

En este paso inicial, se crearán las historias de usuario del proyecto. Estas historias simulan los requisitos de los usuarios y las funcionalidades que el equipo de desarrollo debe construir. Cada historia de usuario tiene criterios de aceptación y, en algunos casos, reglas de negocio que deben ser respetadas. Se llevará a cabo este ejercicio con dos tareas de usuario para abordar los requerimientos más relevantes. Este proceso establece las bases para comprender las necesidades y expectativas de los usuarios, sentando así el fundamento para el desarrollo del sistema de planes como microservicio.

Historia N°1:

Cómo usuario nuevo, quiero explorar los distintos planes y funcionalidades que ofrece Ceptinel para configurar el servicio que más se adapta a mis necesidades.

Criterios de Aceptación

1. Debe existir una página o sección dedicada a mostrar todos los planes disponibles.
2. Cada plan muestra una descripción clara de lo que incluye.
3. Se pueden comparar fácilmente distintos planes, teniendo enfoque en los puntos principales de cada uno.
4. Debe existir una sección complementaria donde se puedan ver todas las funcionalidades extra que no vienen incluidas en los planes (plugins).

Historia N°2:

Como usuario registrado, quiero que la página solo me muestre las secciones y módulos que he contratado, para facilitar la navegación y centrarme en lo que me interesa usar de la plataforma.

Criterios de Aceptación

1. Al iniciar sesión en la plataforma, se muestra una página principal ajustada al plan contratado.
2. La página principal solo muestra las secciones y módulos correspondientes al plan contratado.
3. No se muestran secciones bloqueadas que intervengan con la experiencia de usuario.
4. Al abrir 'mi perfil', se pueden ver los detalles del plan contratado junto a la información de pagos.

Al analizar estas dos historias, surge la necesidad de atraer a los clientes mediante planes diseñados específicamente para ellos. Esto permitirá que los clientes personalicen su servicio contratado al agregar funcionalidades adicionales. Además, se identifican varios aspectos que deben ser gestionados por el equipo de TI, como el control de las vistas, los accesos según el plan contratado y la experiencia de navegación en la página.

Segundo paso: Mockup de las principales funcionalidades solicitadas

El siguiente paso en la metodología implica la creación de un Mockup basado en las historias de usuario. Un Mockup es un diagrama de baja resolución que proporciona una vista preliminar del diseño de un producto o artefacto. En este caso, se enfoca en cómo se comporta la plataforma para cada plan. Este paso es crucial, ya que los recursos visuales facilitan la identificación de los servicios que deben habilitarse.

Para lograr estas vistas, es necesario establecer una conexión entre el usuario y su plan contratado en la capa de datos. Esto será fundamental en el futuro para permitir que los usuarios cambien de plan o incluyan nuevos complementos en su servicio. Actualmente, estas decisiones deben definirse al momento de cotizar el servicio, ya que aún no están listas para ser soluciones Plug & Play¹⁸.

Tecnologías sugeridas: Sitios web: Figma, app.diagram.net. Programas: Paint, Adobe.

¹⁸ Sistema diseñado para ser fácilmente instalado, conectado y utilizado sin requerir una configuración muy compleja

Tercer paso: Modelar datos

En este paso, se aborda la tarea de modelar los datos relacionados con los planes con el objetivo de definir la estructura y la organización de estos dentro de una base de datos. Se ha optado por una base de datos relacional debido a la necesidad de utilizar referencias entre tablas y llevar a cabo operaciones que involucren el cruce de datos.

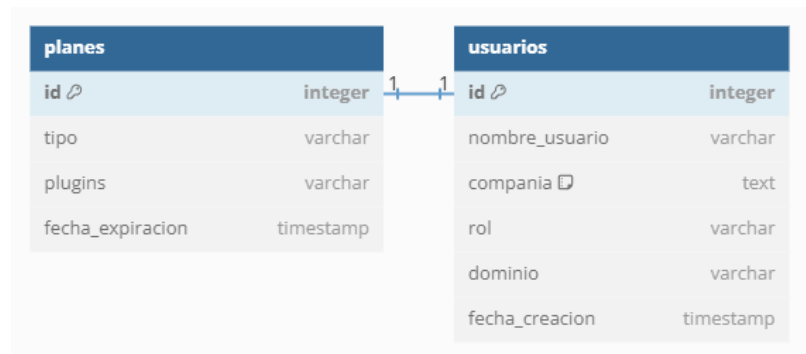


Tabla 7 - Ejemplo de tablas relacionales. Fuente: Elaboración propia

Dado que cada usuario tiene un plan asociado, se establece una relación entre ellos. La tabla de planes incorpora las funcionalidades adicionales que el cliente ha contratado, permitiendo así una representación clara y organizada de la información en la base de datos.

Tecnologías sugeridas: Diagramas: dbdiagram.io. Manejo de DDBB: DBeaver, DataGrip.

Cuarto paso: Desarrollo

El cuarto paso implica el desarrollo del microservicio, siendo fundamental ofrecer distintas funcionalidades a los clientes mediante la limitación selectiva de sus permisos. Inicialmente, se debe desarrollar una página que varíe según el tipo de plan contratado por los usuarios. En cuanto a los servicios que permitan lograr esto, se pueden abordar asignando un token único a la plataforma de cada usuario. Este token sirve para verificar si el usuario tiene los privilegios necesarios para llevar a cabo diversas operaciones dependiendo de la plataforma asociada al plan que usa. Permitiendo reducir la cantidad de modificaciones que se tengan que realizar al resto de los microservicios.

Tecnologías sugeridas: Frontend: Vue.js, Angular. Backend: Spring Boot.

Quinto paso: Testeo y despliegue

El quinto paso implica la fase de prueba y despliegue del nuevo sistema de planes en conjunto con los demás microservicios. Realizar pruebas es crucial, ya que permite verificar la funcionalidad adecuada de la arquitectura antes de su lanzamiento público. Este proceso busca prevenir posibles brechas de seguridad o vulnerabilidades del sistema que puedan ser aprovechadas por atacantes. Además, el despliegue conjunto con los demás microservicios es esencial para confirmar la compatibilidad del sistema de planes con los agentes involucrados, tales como los microservicios de usuario, el motor de reglas, entre otros.

Tecnologías sugeridas: Testing: Junit, Mockito. CI/CD: Jenkins.

Métricas

Evaluar la implementación actual de microservicios para garantizar que cumple con al menos el 80% de las buenas prácticas sugeridas por la literatura

Para realizar este análisis, compararemos las tecnologías utilizadas por Ceptinel con las más comúnmente empleadas según el estudio presentado en el estado del arte [Véase Anexo 5], el cual abarca diversas opciones para cada tipo de tecnología. Este enfoque nos permitirá evaluar la alineación de Ceptinel con las tendencias y prácticas destacadas en la literatura.

Tipo de tecnología	Más popular en la literatura	Ceptinel
Infraestructura Cloud	Amazon Web Services	Microsoft Azure
Herramientas de comunicación	REST API	Apache Kafka
Framework de desarrollo	Spring Boot	Spring Boot
Integración Continua	Jenkins	Jenkins
Test aplicados sobre microservicios	Test unitario / Integración	Test unitario
Principios de monitoreo	Orientado a contenedores	Orientado a contenedores
Herramientas de monitoreo	Kubernetes	Kubernetes

Tabla 8 – Comparación del uso de tecnologías según el estudio vs usadas por Ceptinel. Fuente: Elaboración propia

En esta comparación, observamos que Ceptinel se alinea con 5 de las 7 tecnologías más populares. Como se mencionó anteriormente, no es necesario cumplir con todas, ya que la elección debe basarse en las capacidades del equipo y las necesidades del negocio. Sin embargo, es importante considerar las opciones más populares, ya que suelen contar con un mayor soporte y tienen una mayor probabilidad de seguir operativas a lo largo del tiempo. Por lo que Ceptinel adopta un 100% de las tecnologías recomendadas.

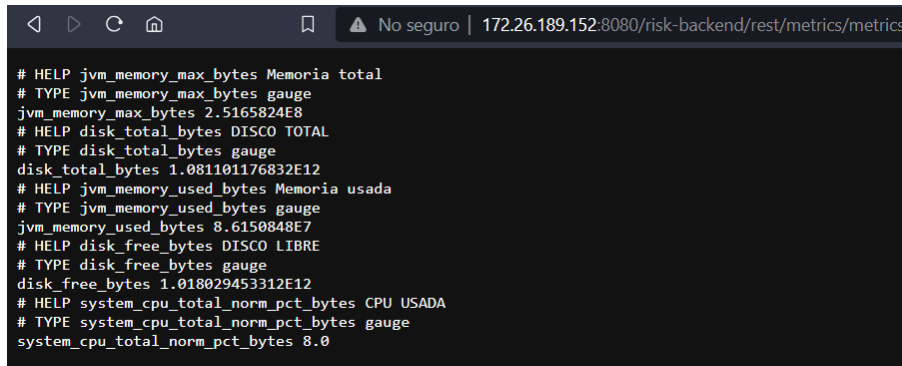
Algunos módulos, como las herramientas para el manejo de logs y la gestión de la sobrecarga de red, aún no se han implementado debido a que no se ha realizado un testeo completo de la arquitectura. Sin embargo, el equipo las está considerando para futuras implementaciones. En cuanto a la infraestructura *Cloud*, Ceptinel ha migrado de proveedor de nube con frecuencia, habiendo trabajado anteriormente con *Amazon* y luego decidido cambiar a *Microsoft Azure*. En cuanto a la herramienta de comunicación, eligieron *Apache Kafka* por su capacidad de facilitar la comunicación asíncrona entre microservicios, donde el agente que envía la petición no se detiene a esperar respuesta.

Se detallan las tecnologías del estudio en el [Anexo 5]. Es relevante destacar que la empresa implementa un 100% de las tecnologías de este estudio y un 80% de las prácticas sugeridas por la literatura [9], por lo que obtenemos que Ceptinel ha implementado un 90% de las tecnologías y practicas recomendadas.

Evaluar el rendimiento de los microservicios comparándolos a la arquitectura monolítica

Tras dialogar con la empresa, se llegó a la conclusión de que el rendimiento de los microservicios no resulta ser un objetivo principal para el desarrollo, a menos que los microservicios se vuelvan excepcionalmente lentos y deban ser intervenidos. No obstante, se considera crucial conocer los datos que resaltan durante la ejecución de la plataforma, como el consumo de memoria o uso de red.

Aunque esta información debería ser accesible a través de los tableros de Azure, los cuales registran el consumo de cada máquina virtual y recurso contratado, la empresa cuenta con un plan básico de Azure que no permite visualizar el consumo individual de cada máquina. Frente a esta limitación, se optó por encontrar un enfoque alternativo para obtener métricas, recurriendo al uso de Prometheus (software para exponer métricas de una aplicación) y Grafana (software para crear gráficos a partir de fuentes de datos).



```
# HELP jvm_memory_max_bytes Memoria total
# TYPE jvm_memory_max_bytes gauge
jvm_memory_max_bytes 2.5165824E8
# HELP disk_total_bytes DISCO TOTAL
# TYPE disk_total_bytes gauge
disk_total_bytes 1.081101176832E12
# HELP jvm_memory_used_bytes Memoria usada
# TYPE jvm_memory_used_bytes gauge
jvm_memory_used_bytes 8.6150848E7
# HELP disk_free_bytes DISCO LIBRE
# TYPE disk_free_bytes gauge
disk_free_bytes 1.018029453312E12
# HELP system_cpu_total_norm_pct_bytes CPU USADA
# TYPE system_cpu_total_norm_pct_bytes gauge
system_cpu_total_norm_pct_bytes 8.0
```

Ilustración 7 – Ejemplo de captura de métricas por medio de Prometheus [1]

Estas tecnologías se aplicaron al monolito con el propósito de realizar pruebas, y se generaron series de tiempo que abarcaron diversas métricas, como el uso de memoria, CPU y disco. Con el fin de poner a prueba estas herramientas, se diseñó una métrica personalizada para rastrear el número de ejecuciones del motor de reglas en intervalos de 30 minutos. La captura de esta métrica implica esencialmente integrar un contador que incrementa cada vez que se añaden registros de ejecución a la base de datos.

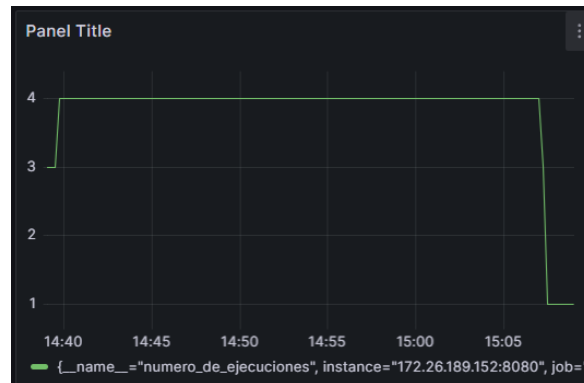


Ilustración 8 – Número de ejecuciones del motor de reglas en intervalos de 30 minutos. Fuente: Elaboración propia

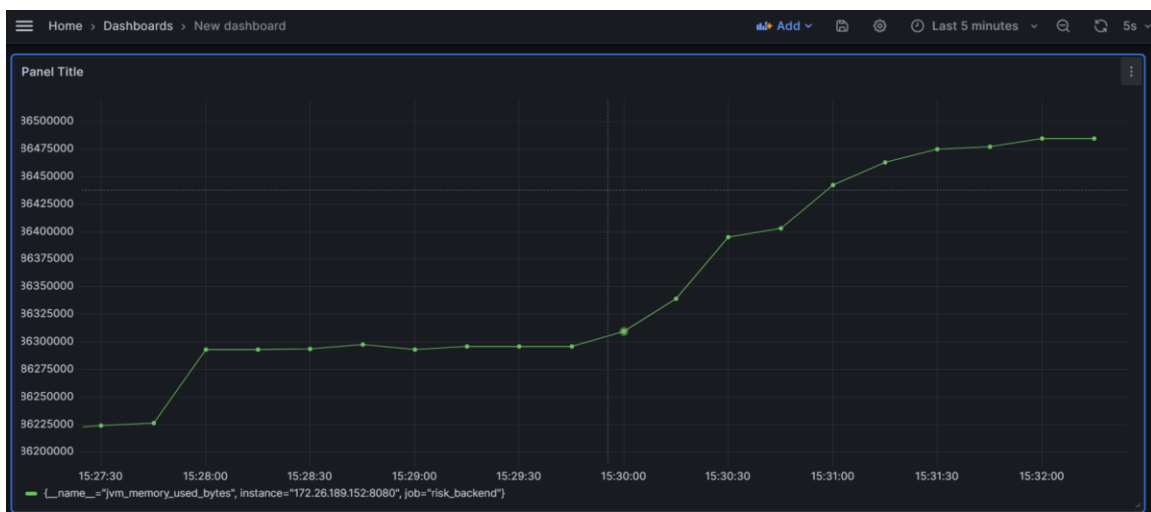


Ilustración 9 - Serie de tiempo de bytes de memoria usados por la máquina virtual de java. Fuente: Elaboración propia

La integración de estas tecnologías en el monolito fue un logro destacado, y además, se documentó el proceso de instalación de la plataforma en el sistema operativo Windows para ampliar las opciones de trabajo de la empresa, que anteriormente se limitaban al sistema operativo Linux. Este paso adicional brinda una nueva posibilidad de trabajar en distintos entornos.

A pesar de la limitación de no poder implementar las métricas en los microservicios debido a la ausencia del motor de reglas operativo, se proporcionaron pasos detallados a la empresa para instalar estas tecnologías, que son compatibles con los microservicios al seguir un patrón de diseño similar al del monolito.

En conclusión, se espera un rendimiento teóricamente similar o incluso superior al del monolito, considerando las mejoras que el equipo está realizando en la plataforma durante la migración.

Analizar los recursos *Cloud* utilizados en la empresa para reajustar las máquinas virtuales asignadas a algunos clientes

En el inicio de la pasantía surgió una inquietud respecto a los costos mensuales, originada por el análisis del arquitecto *Cloud* anterior. Este había examinado los datos de Azure y había identificado una oportunidad para reducir los costos mediante una revisión de los recursos utilizados. No obstante, la implementación de este cambio requería la aprobación tanto del equipo como de los altos mandos de la empresa, dado que implicaba modificaciones en la infraestructura, las reglas de negocio y la inversión *Cloud*.

En el mes de septiembre el arquitecto *Cloud* dejó la empresa, dejando el plan de ahorro sin concluir. Ante esta situación, el nuevo jefe de Tecnologías de la Información (TI) asumió sus responsabilidades y comenzó un proceso de capacitación para gestionar la arquitectura *Cloud* de la empresa. Aprovechando esta oportunidad, se colaboró con el jefe de TI para identificar los recursos clave y explorar formas de utilizarlos de manera más eficiente.

Aunque no se tuvo un papel central en la revisión de costos, se logró cuestionar e intervenir en diversos aspectos al participar en la célula de desarrollo. Se identificaron múltiples recursos sin uso o cuyo responsable real no estaba claro, y cada uno de estos hallazgos se destacó para su consideración, como se muestra en el anexo [Véase Anexo 7].

De forma paralela, el jefe Ti junto a otro pasante levantaron la siguiente gráfica que muestra los costos *Cloud* mensuales, siendo los conceptos principales correspondientes a máquinas virtuales, pago por almacenamiento, respaldos y uso de Firewall (barrera de seguridad virtual).

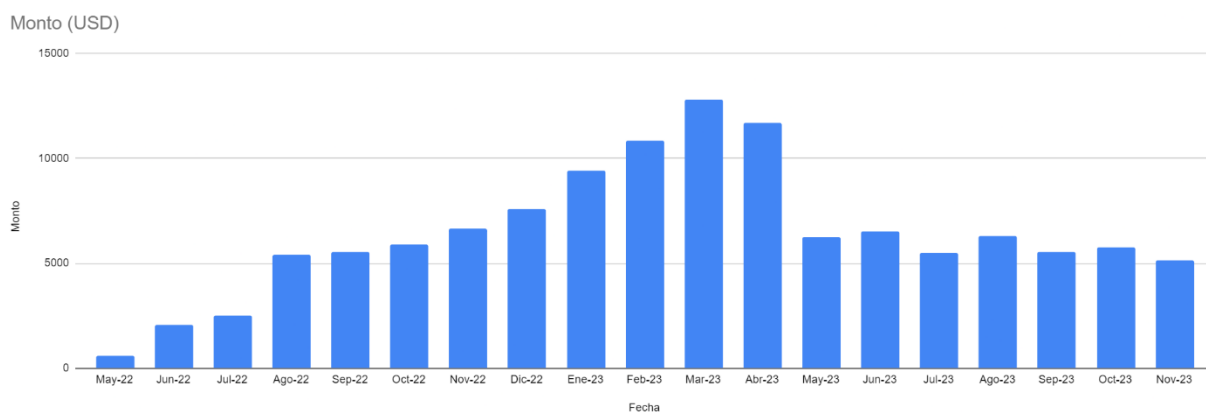


Ilustración 10 – Evolución de costos mensuales *Cloud* [1]

En el análisis de la evolución de los costos en los últimos 6 meses, se observa una reducción en el mes de noviembre, alcanzando niveles que no se registraban desde el año pasado. El estudio se basa en los seis meses precedentes a noviembre (mayo a octubre), que poseen un promedio de costos de 5987 créditos. Es importante señalar que existe un margen de error aproximado del 10%, atribuido a la incertidumbre en algunos conceptos de los costos mensuales.

Para comparar estos datos con el mes de noviembre, se realizó un ajuste considerando que los costos de este mes solo abarcan hasta el día 27. Al incorporar tres días adicionales, el total de créditos utilizados en noviembre asciende a 5703, reflejando una reducción del 4,7% en comparación con el promedio de los seis meses anteriores.

Plan de implementación

Para la construcción de servicios, se analizó el detalle de los planes definidos por la empresa [Véase Anexo 8].

Para el desarrollo de microservicios, se trabaja con el patrón MVC (Modelo-Vista-Controlador). Este patrón guía el desarrollo de una aplicación de manera que la recepción de las solicitudes de un usuario se gestiona a través de una capa de controlador. Posteriormente, estos datos se manipulan en la capa de modelo, la cual trabaja sobre ellos y realiza operaciones en la base de datos. Finalmente, los resultados se devuelven a la capa de controlador, la cual los entrega a la capa de vista. La capa de vista corresponde al sitio web o interfaz a través de la cual el usuario puede interactuar. Este enfoque de desarrollo permite una separación clara de responsabilidades, facilitando la mantenibilidad y la escalabilidad del sistema.



Ilustración 11 – Patrón MVC [11]

El siguiente paso implica la creación de un proyecto Spring Boot, que es el marco de trabajo utilizado en los microservicios. La atención se centrará en las capas de controlador y modelo, correspondientes al Backend de la aplicación. La decisión de priorizar estas capas se debe a que la parte del Frontend puede consumir mucho tiempo y solo involucra la capa de vista.

Es necesario crear un archivo de configuración (.yml) que especifique los datos de conexión a una base de datos y el puerto que se expondrá para la comunicación con el servicio de registro (llamado Eureka) y de configuración. Se requiere tener ambos servicios corriendo para que se levante un microservicio, de lo contrario no podrá ser ejecutado.

```

1  springdoc:
2    api-docs:
3      enabled: true
4      path: /api/list-service/v3/api-docs
5
6  server:
7    port: 8090
8  eureka:
9    client:
10     serviceUrl:
11       defaultZone: http://localhost:8761/eureka
12       initialInstanceInfoReplicationIntervalSeconds: 5
13       registryFetchIntervalSeconds: 5
14     instance:
15       instance-id: ${spring.application.name}:${spring.application.instance_id:${random.value}}
16       leaseRenewalIntervalInSeconds: 5
17       leaseExpirationDurationInSeconds: 5
18
19  datasource:
20    url: jdbc:postgresql://localhost:5432/my_database
21    username: default
22    password: default
23    driver-class-name: org.postgresql.Driver

```

Imagen 3 – Archivo de configuración base usado para microservicios [1]

Para la facilidad del equipo, cada uno de los microservicios se levantó como contenedor Docker en un entorno virtual de Linux, ya que en este se facilita el despliegue de los servicios.

```

montadel@B0ESKTOP-B8K8CQK:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
e98990154304   eureka-service  "java -jar /app/eure..." 2 weeks ago   Up 4 seconds  0.0.0.0:8761->8761/tcp, :::8761->8761/tcp  eureka-service
61a51edf5282   hyness/spring-cloud-config-server  "/cnb/process/web"        2 weeks ago   Up 4 minutes  0.0.0.0:8888->8888/tcp, :::8888->8888/tcp  config-service

```

Imagen 4 – Servicios de registro y configuración ejecutándose como contenedores en Linux

De esta forma, podemos desplegar nuestro proyecto de planes y verificar que sea detectado por el servicio de registro.

Instances currently registered with Eureka			
Application	AMIs	Availability Zones	Status
PLAN-MICROSERVICE	n/a (1)	(1)	UP (1) - plan-microservice-5c5695cee7ce2aa5f80397b4cf847a0d
General Info			
Name	Value		
total-avail-memory	106mb		
num-of-cpus	8		
current-memory-usage	44mb (41%)		
server-uptime	00:07		

Imagen 5 – Servicio de registro Eureka con un microservicio detectado

Con estas configuraciones en marcha, se inicia el desarrollo, siguiendo las consideraciones establecidas en la metodología. Un aspecto crítico a tener en cuenta es el papel que desempeñará el equipo de ventas en la comercialización del nuevo sistema de planes, ya que la venta de este nuevo servicio es fundamental para obtener resultados.

Análisis de riesgo

Analizando el plan de implementación y la forma en la que se espera que opere microservicios, se detectaron los siguientes riesgos junto a su probabilidad e impacto esperado:

1. Sistema de planes no logra aumentar las ventas (**Improbable - Catastrófico**).
 - Improbable: Dada la existencia de una cartera de clientes que buscan servicios personalizados.
 - Catastrófico: Considerando la inversión significativa de tiempo y recursos sin obtener un aumento en clientes.
2. Aumento en los tiempos de entrega de nuevas funcionalidades o features por parte de equipo de desarrollo (**Improbable, Moderado**).
 - Improbable: Gracias a la migración, el equipo ha adquirido conocimientos significativos sobre operar con microservicios.
 - Moderado: Implica invertir tiempo en capacitar al equipo para revisar su rendimiento.
3. Exceder la capacidad del equipo por nueva cantidad de clientes (**Probable - Moderado**).
 - Probable: Dado el aumento en el volumen de solicitudes generadas.
 - Moderado: Podría causar retrasos en respuestas a solicitudes, generando malestar en los clientes.
4. Inestabilidad de la nueva arquitectura en fase inicial (**Ocasional, catastrófico**).
 - Ocasional: Dependiendo de la calidad de los testeos realizados.
 - Catastrófico: Cualquier caída del servicio afecta negativamente la identidad de la empresa al reducir la confianza y seguridad de los clientes.
5. Altos costos por migrar a los actuales clientes a microservicios. (**Probable - Significante**)
 - Probable: Necesario migrar a los clientes debido a la decisión de discontinuar el monolito.
 - Significante: Requiere una dedicación sustancial de horas hombre para migrar datos, reglas y archivos del cliente hacia el nuevo sistema.

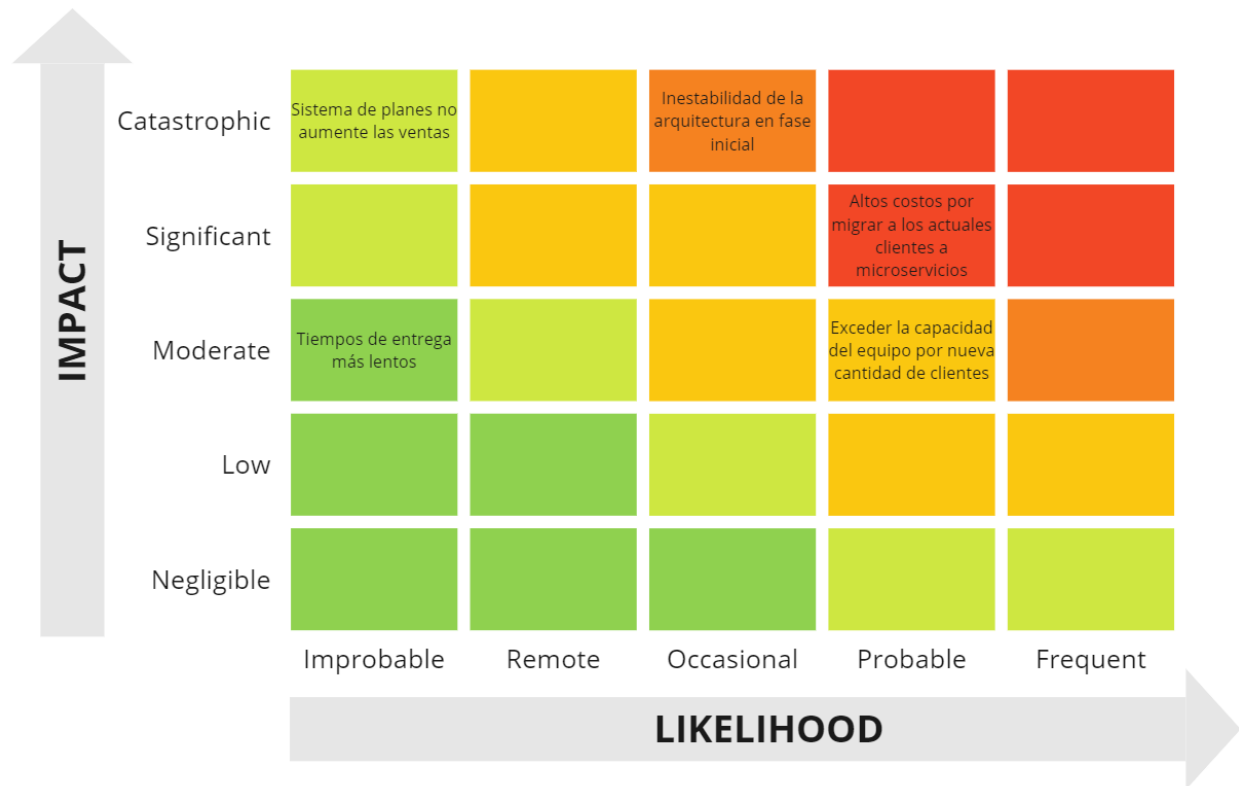


Ilustración 12 – Matriz de riesgo. Fuente: Elaboración propia

Para mitigar los tres riesgos más severos, se tomarán las siguientes medidas:

1. **Sobrecarga de capacidad:** Si se mantiene el actual tamaño del equipo, es esperable que eventualmente cuando lleguen más clientes no se podrá tener la misma capacidad de responder a todos, sin embargo, el equipo actualmente está trabajando en la automatización de procesos orientados al soporte técnico, por lo que se espera que puedan aumentar su capacidad de respuesta incluso con un alza de clientes.
2. **Inestabilidad de la nueva arquitectura:** La forma de abordar esto es realizando múltiples testeos, actualmente se utilizan test unitarios para comprobar que el software funcione a nivel de componente, pero también se deben realizar pruebas como test de carga, donde se pone a prueba el software pasándole grandes volúmenes de tráfico o múltiples usuarios concurrentes para detectar dónde falla. Otro tipo de prueba son los test de integración para comprobar que distintos componentes puedan funcionar entre sí. Las pruebas en microservicios son más complejas al tener que probar cada componente con el resto, pero es gracias a este ejercicio que se puede asegurar una entrega estable para los clientes.
3. **Altos costos por migrar a los actuales clientes a microservicios:** El costo principal viene a ser por horas hombre, ya que se deben migrar los datos y reglas del cliente, este proceso puede ser complejo ya que en microservicios se están usando distintos esquemas y tablas de bases de datos. La principal mitigación a esto es preparar un script para migrar de la base de datos anterior a la nueva, y asegurar que las reglas sean retro compatibles con la nueva versión del motor de reglas, de forma que se puedan importar y usar fácilmente.

Con estas nuevas consideraciones, se espera principalmente reducir la probabilidad de que ocurran estos riesgos, resultando en una nueva matriz de riesgos:

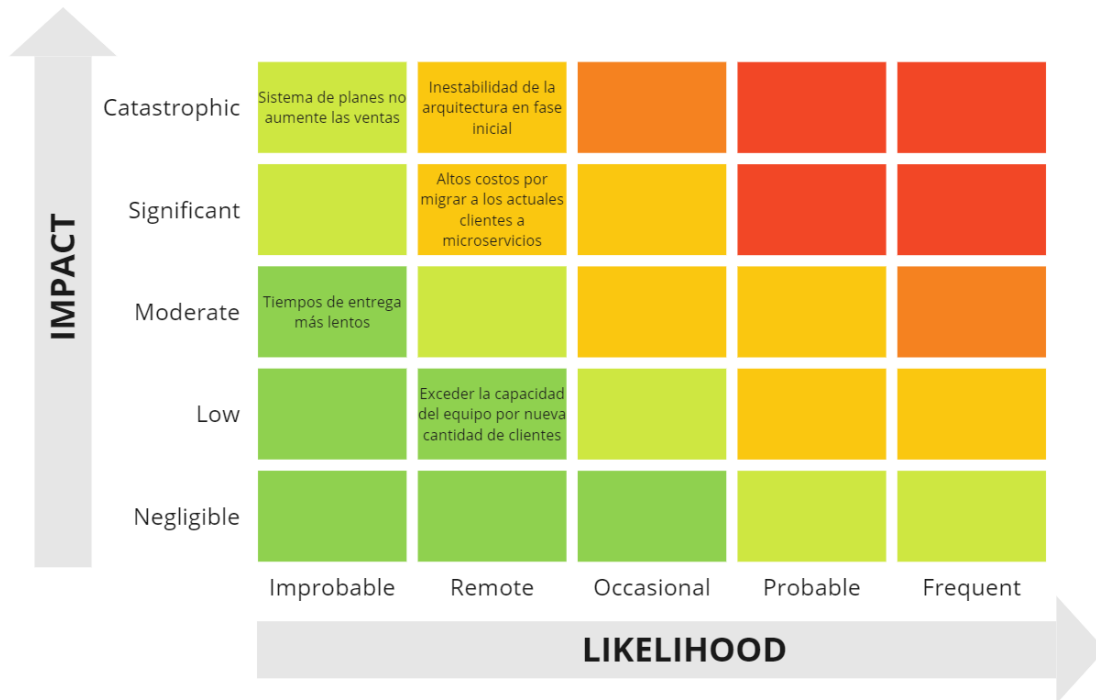


Ilustración 13 – Matriz de riesgo actualizada. Fuente: Elaboración propia

Resultados obtenidos

Debido a ciertos desafíos encontrados, la migración completa de la arquitectura no pudo llevarse a cabo en el transcurso de este año. Hasta la fecha, se han logrado levantar 18 de los 31 microservicios identificados, con 14 de ellos exitosamente sometidos a pruebas y evaluaciones. Aunque la implementación del nuevo sistema de planes no fue posible según lo inicialmente planeado debido al retraso en la migración, se obtuvieron resultados positivos en línea con los objetivos específicos.

En cuanto al porcentaje de cobertura de módulos en microservicios, se observó la implementación de un 90% de las tecnologías sugeridas por el estudio escogido [Véase Anexo 5] y las prácticas declaradas por la literatura [9]. Las restantes serán incorporadas por el equipo al momento de avanzar a la fase de pruebas del sistema completo.

En relación con las métricas de rendimiento entre arquitecturas, lamentablemente no fue posible realizar la comparación debido a la indisponibilidad del motor de búsqueda operativo en los microservicios. No obstante, se espera que el rendimiento sea similar o incluso mejore una vez que el motor esté operativo.

En cuanto al ahorro de recursos *Cloud*, estos se vieron reflejados en el último mes mediante una reducción del 4.7% en los costos en comparación con el promedio de los últimos 6 meses. Este descenso refleja los esfuerzos realizados para llevar a cabo la iniciativa de ahorro de recursos.

Aunque la implementación del sistema de planes no se logrará dentro del periodo de pasantía, la empresa se apoya en años de experiencia y en estadísticas sobre las reglas más utilizadas por sus clientes para ajustar sus planes a la medida, como se vio en la matriz de cada plan.

Principalmente, se anticipa un aumento en la base de clientes tal como se reflejó en el primer análisis de sensibilidad para microservicios. Aunque esta estimación podría parecer muy optimista, la empresa cuenta con una sólida cartera de clientes interesados en colaborar, respaldada por su prestigio internacional. Además, Ceptinel participa activamente en congresos internacionales y se esfuerza por mantener relaciones sólidas con sus contactos, lo que fortalece aún más las perspectivas de crecimiento.

Adicionalmente, la participación en el equipo permitió robustecer la documentación del trabajo de cada integrante, por un lado, porque en algunos casos estas se encontraban desactualizadas, y por otro lado porque es crucial tener la capacidad de tomar el trabajo de otra persona en caso de que alguien no se encuentre disponible, ya que el equipo de desarrollo posee pocos participantes y la ausencia de cualquiera afectaría fuertemente al resto.

Conclusión

Tras haber participado en la empresa durante el proceso de migración y el levantamiento de un nuevo proyecto, se pudo ver que Ceptinel cuenta con todas las capacidades para llevar a cabo la migración.

Para el análisis de las tecnologías utilizadas por la empresa en la construcción de microservicios y las prácticas de desarrollo empleadas, se demostró que la empresa posee un entendimiento sólido de la nueva arquitectura, un uso de tecnologías que conversan con los objetivos de negocio y una clara proyección de los pasos futuros de la migración.

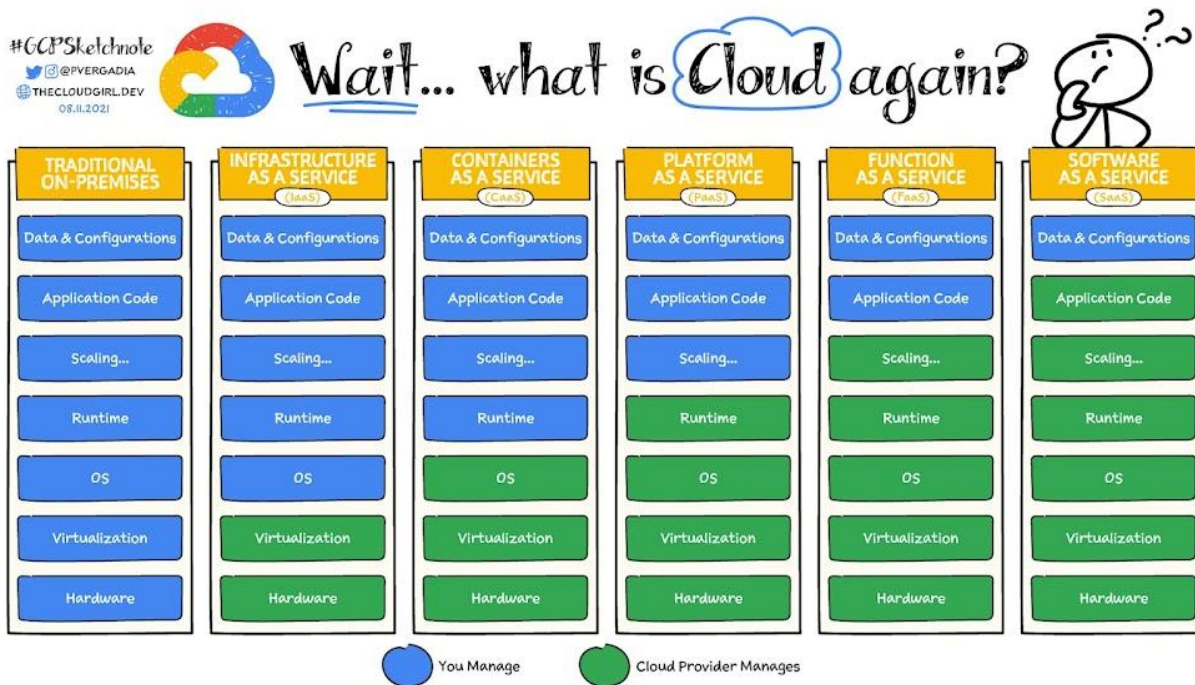
Por otro lado, las métricas recopiladas demuestran la capacidad de la empresa para capturar diversos datos basándose en el uso que los clientes hacen de la plataforma. A través de esta información, se logra identificar las principales necesidades de los clientes, lo que facilita el levantamiento y análisis de futuros proyectos.

Además de la reducción de costos en la nube durante el último mes, se fomentó el conocimiento sobre servicios *Cloud* en el equipo, revelando su importancia para los altos cargos de la empresa y destacando la relevancia de su gestión y monitoreo.


Otro aspecto crucial fue la participación en el levantamiento de un producto externo al de la pasantía, lo que permitió adquirir conocimientos fundamentales sobre cómo definir, modelar y construir un proyecto desde cero. También se desempeñó un rol en el proceso de migración, estando a cargo de la creación de un nuevo microservicio encargado del dominio de las contraseñas de los usuarios. Esta experiencia facilitó la mejora de aspectos valiosos para el equipo, como la documentación, los estándares de programación y las convenciones de nombres, entre otros.

Finalmente, se espera que el nuevo sistema de planes no solo represente una nueva fuente de ingresos para la empresa, sino también una solución accesible para diversas organizaciones en toda Latinoamérica que buscan abordar problemas como la corrupción interna, el lavado de activos, el narcotráfico y los fraudes financieros.

Anexo



Anexo 1 - ¿Cuál es la diferencia entre Paas, Iaas, y SaaS? [12]

KPI's comerciales 2023 – Monitoreo Transaccional 

KPI	Meta	Frec	Mayo	Junio	Julio	Agosto	Septiembre
Reuniones	15	mensual	20 (133%)	16 (107%)	14 (93%)	20 (133%)	12 (80%)
Propuestas comerciales	7	mensual	8 (114%)	10 (143%)	8 (114%)	9 (129%)	6 (86%)
Propuestas adjudicadas	3	mensual	0 (0%)	1 (33%)	1 (33%)	1 (33%)	0 (0%)

Anexo 2 – KPI comerciales de reuniones iniciales, propuestas comerciales y propuestas finales adjudicadas a clientes [1]

#	Funcionalidad	Descripción
1	Motor de reglas	Ejecución de reglas
2	Issues	Se utiliza para la creación de casos
3	File storage	Almacenamiento de Ceptinel, el cliente puede crear carpetas, subir archivos, imágenes, etc.
4	Portal ético	Recursos de Portal ético (Gestión de denuncias, subir archivos, etc.)
5	ACL	
6	Searchers	Para realizar búsqueda por índices - Se utiliza Apache Lucene
7	Chart	Gráficas del dashboard
8	Schema	CRUD de esquemas de las listas, issues
9	Audit	
10	Auth	
11	Gateway	Puerta de entrada y reglas de los microservicios
12	Usuarios	Gestión de usuarios
13	Data management	CRUD que reemplaza las listas
14	Comments	Se utiliza para agregar comentarios
15	Companies	Se utiliza para crear compañías (Filiales)
16	Config	Se utiliza para el módulo de configuración global del sistema
17	Custom-report	Se utiliza para la ficha de clientes (Datos del cliente)
18	Dashboard	Se utiliza para el dashboard de la pantalla inicial de la aplicación
19	Filter	Tiene que ver con las reglas
20	Notifications	CRUD de notificaciones (Alertas Issues)
21	Report	En revisión
22	Report-PowerBI	Reportes de métricas - Se conecta con Azure
23	Config Service	Servicio de configuración centralizada
24	Eureka Server	Servicio de registro
25	Zipkin	Trazabilidad de los microservicios
26	Monitor Services	Monitor de status de los microservicios
27	Swagger	Documentación de los endpoint del Backend
28	SonarQube	Calidad del código
29	Group	CRUD de grupos asociados a los usuarios
30	Indicators	Para configurar la paleta de colores
31	References	Campos referenciados de Issues

Anexo 3 – Microservicios identificados por Ceptinel [1]

Anexo 4 – Descripción de los componentes de microservicios

Cada microservicio puede funcionar por sí solo, pero necesitamos más herramientas para que funcionen de manera distribuida, a continuación, describiré cómo la empresa construyó su arquitectura.

La primera herramienta es un **servicio de configuración**, este permite centralizar la configuración de cada microservicio para distintos perfiles (uso local, desarrollo y productivo) con el fin de facilitar el manejo de cada configuración y evitar tener que hacerlo manualmente para cada servicio (especialmente para múltiples instancias de un mismo microservicio).

La segunda herramienta es un **servicio de registro**, este permite que los microservicios se conecten a un mismo punto común donde revelan su ubicación y credenciales al resto de los microservicios para facilitar la comunicación. También nos sirve para monitorear cuales microservicios se encuentran arriba.

La tercera herramienta es una **Api Gateway**, esta funciona como principal barrera entre el cliente y el Backend, dado que recibe directamente las peticiones y las envía al microservicio correspondiente, funciona también punto de autenticación y autorización, permitiendo quitarle esa carga al resto de los servicios. También funciona como balanceador de carga, el cual se encarga de redirigir las peticiones a los microservicios con mayor disponibilidad.

La cuarta herramienta es un **Message Broker**, el cual permite la comunicación asíncrona entre microservicios por medio de la subscripción a canales con un tópico específico, similar a como nos conectamos a WhatsApp para conversar con otra persona sin que otros puedan ver la conversación.

La quinta herramienta es una **base de datos** de tipo relacional, las bases de datos relacionales son como grandes hojas de cálculo organizadas que nos ayudan a guardar y relacionar información, facilitando la búsqueda y gestión de datos, que una base sea relacional permite conectar distintas tablas entre sí para realizar operaciones de cruce de datos.

La sexta herramienta es un **servicio de caching**, o de memoria persistente, permite ser instalado como intermediario entre los microservicios y la base de datos, de manera que se capturan las peticiones más frecuentes y se guardan dentro del servicio de *caching*, resultando en una comunicación más eficiente al obtener información directamente del *caching* en vez de consultarla a la base de datos.

Finalmente, la séptima herramienta corresponde a un **servicio de monitoreo**, que sirve para rastrear las peticiones y los mensajes enviados por microservicios para rastrear fácilmente su punto de falla en caso de que fallen.

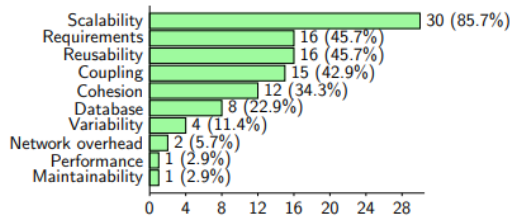


Figure 2: Criteria used for the microservice extraction.

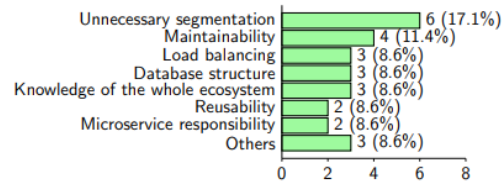


Figure 3: Concerns on the microservice granularity.

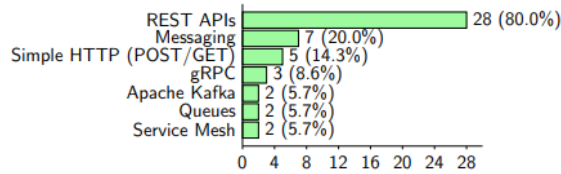


Figure 5: Communication tools used.

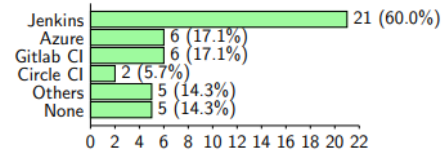


Figure 7: Continuous integration tools used.

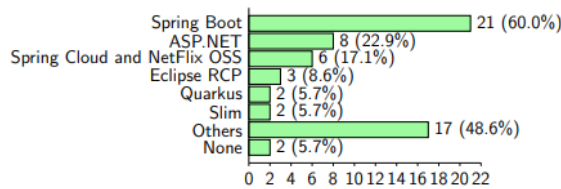


Figure 6: Frameworks used.

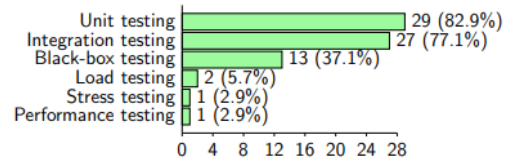


Figure 8: Types of test applied to verify microservices.

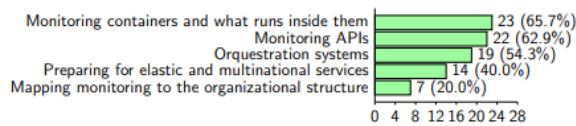


Figure 9: Principles of monitoring microservices used.

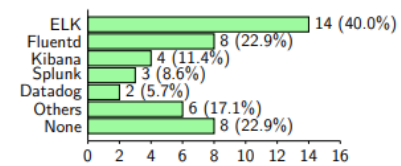


Figure 11: Log management tools used.

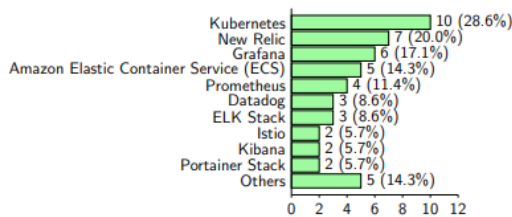


Figure 10: Monitoring tools used for microservices.

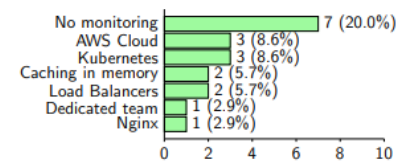


Figure 12: Management of network overhead.

Anexo 5 – Tecnologías y prácticas usadas por empresas que operan con microservicios [13]

Costos identificado (Items)			Costo mensual	Detalle	Año 1	Año 2	Año 3	Año 4	Año 5	
*Se asume q	Costos Cloud				Multiplificador de costos Cloud (en base al costo del año 1)					
x Cliente	Máquinas virtuales				\$270,000	1	1.50	2.50	3.75	5.00
x Grupo	Firewall				\$620,800	1	1	1	1	1
x Cliente	Almacenamiento				\$108,000	1	1.50	2.50	3.75	5.00
x Grupo	App Gateway				\$45,000	1	1	1	1	1
x Grupo	Virtual network				\$121,500	1	1	1	1	1
x Cliente	Respaldos				\$24,300	1	1.50	2.50	3.75	5.00
FreshDesk (Soporte)										
Costo fijo					\$135,000	Sistema				
Sueldo de trabajadores										
Equipo de desarrollo					\$6,000,000	2 personas				
Equipo analista					\$7,500,000	4 personas + 1 soporte				
Cyberseguridad					\$2,500,000	1 persona				
Marketing										
Publicidad					\$1,000,000	1 persona				
Equipo de ventas					\$3,000,000	2 persona				
Costos adicionales										
Personal de aseo					\$460,000	1 persona				
Arriendo oficina					\$630,000	primeros dos años				
Arriendo oficina más grande					\$1,500,000	últimos tres años				
Notebooks					\$500,000	1668 1TB i7-1165G7				
Valor del dólar considerado					\$900					
*Usar el N° de clientes para calcular costos y horas vendidas para las utilidades										
Proyección de clientes anuales					Año 1	Año 2	Año 3	Año 4	Año 5	
N° clientes totales					20	30	50	75	100	
N° clientes a proveer con VM					16	24	40	60	80	
Implementaciones del Ceptinel					4	10	20	25	25	
*Se redujo a 5 años porque la startup planea venderse en el futuro, por lo que no se considera más a futuro a pesar de que continúe su operación										
Total de ingresos acumulados					\$432,000,000	\$669,600,000	\$1,134,000,000	\$1,674,000,000	\$2,187,000,000	
*Se usa el promedio de todos los sueldos core para nuevos trabajadores										
Proyección de trabajadores en el futuro					Año 1	Año 2	Año 3	Año 4	Año 5	
Tamaño efectivo del equipo					11	11	16	16	24	
Sueldo del equipo acumulado anual					\$240,000,000	\$240,000,000	\$249,090,909	\$249,090,909	\$263,636,364	
Resumen ingresos y costos										
Clientes					20	30	50	75	100	
Ingresos					\$432,000,000	\$669,600,000	\$1,134,000,000	\$1,674,000,000	\$2,187,000,000	
Costos					-\$72,754,800	-\$411,375,600	-\$508,148,109	-\$604,700,109	-\$715,797,564	
Total					\$59,245,200	\$258,224,400	\$625,851,891	\$1,069,299,891	\$1,471,202,436	
Tamaño equipo por año										
Equipo core					7	11	18	26	35	
Diferencial					0	4	7	9	9	
Expectativa real					7	7	10	10	18	
Equipo completo					11	11	16	16	24	
Ingresos principales										
Ítem					Valor					
Licencia mensual					\$1,350,000					
Manejo del Cloud					\$450,000					
Implementación					\$5,400,000					
*Sumar 2 comerciales (Listo)										
*Incluye horas base de roadmap										
*Cobro mensual										
*Se hace una vez para cada nuevo cliente										
Suelto por encargados del aseo										
*Considerar arriendo de nuevo espacio de trabajo										
*Incluir costo por nuevos pcs										
Activo					Valor	Vida útil	Depreciación	Valor residual 3	Valor residual 1y	
Notebooks					\$500,000	5	100000	\$200,000	\$400,000	

Ítem	Valor Premium	Valor Enterprise	Valor básico
Licencia mensual	\$ 1,350,000	\$ 1,000,000	\$ 350,000
Manejo del Cloud	\$ 450,000	\$ 450,000	\$ 150,000
Implementación	\$ 5,400,000	\$ 4,500,000	\$ 1,000,000

Anexo 6 – Total de ingresos y costos considerados junto a análisis varios para el flujo de caja. Fuente: Elaboración propia

Apagar la máquina virtual o cambiar su tamaño

Feedback Descargar como CSV Descargar como PDF Posponer Descartar Crear alerta Configurar regla de recomendación

Seleccionar	Máquina virtual	Acciones recomendadas	Posibles ahorros anuales basados en...	Suscripción	Regla de recomendación	Detalles adicionales**	Última actualización	Acción
<input type="checkbox"/>	[Icono] [Nombre oculto]	Cambiar el tamaño de Standard_D4s_v5 a... Ver patrones de uso	205.34 USD (12%)	Patrocinio de Microsoft Azure	Uso de CPU < 100%	CPU (%) - 2 Red (%) - 0 Memoria (%) - 67 Periodo de...	22-11-2023, 11:31	Posponer Descartar
<input type="checkbox"/>	[Icono] [Nombre oculto]	Cambiar el tamaño de Standard_D4as_v5 a... Ver patrones de uso	1,990.94 USD (63%)	Patrocinio de Microsoft Azure	Uso de CPU < 100%	CPU (%) - 27 Red (%) - 0 Memoria (%) - 44 Periodo de...	22-11-2023, 11:32	Posponer Descartar
<input type="checkbox"/>	[Icono] [Nombre oculto]	Cambiar el tamaño de Standard_D4s_v3 a... Ver patrones de uso	857.09 USD (50%)	Patrocinio de Microsoft Azure	Uso de CPU < 100%	CPU (%) - 1 Red (%) - 0 Memoria (%) - 15 Periodo de...	22-11-2023, 11:44	Posponer Descartar
<input type="checkbox"/>	[Icono] [Nombre oculto]	Cambiar el tamaño de Standard_D4s_v5 a... Ver patrones de uso	857.09 USD (50%)	Patrocinio de Microsoft Azure	Uso de CPU < 100%	CPU (%) - 4 Red (%) - 0 Memoria (%) - 18 Periodo de...	22-11-2023, 11:24	Posponer Descartar
<input type="checkbox"/>	[Icono] [Nombre oculto]	Cambiar el tamaño de Standard_D8as_v5 a... Ver patrones de uso	3,178.37 USD (50%)	Patrocinio de Microsoft Azure	Uso de CPU < 100%	CPU (%) - 2 Red (%) - 0 Memoria (%) - 25 Periodo de...	22-11-2023, 11:25	Posponer Descartar

Anexo 7 – Recomendaciones de ahorro de Azure [1]

DESCRIPCION		BASE Incluye Soporte 5 x 8 + HH Consultoría		
Grupo de Servicios	Detalle de Servicio	AML/Compliance	Prevención de Fraude	Abuso de Mercado
Reglas de Negocio Base	5 reglas Base (definir librería BASE por vertical)	VERDADERO	VERDADERO	VERDADERO
	Parametrización	VERDADERO	VERDADERO	VERDADERO
Gestión de Alertas	Panel de Alertas y Casos	VERDADERO	VERDADERO	VERDADERO
	Dashboard de Estadística de Gestión de Casos, Asignación de Tareas y Monitoreo de Alertas y Casos (matriz relaciones)	VERDADERO	VERDADERO	VERDADERO
Servicio de Datos - Screening Personas	Listas Internacionales Públicas (ONU, OFAC, CAN, AUS, EU)	VERDADERO	VERDADERO	VERDADERO
	Busqueda PEP por país	VERDADERO	VERDADERO	VERDADERO
Reportes Normativos	Base de Noticias Adversas	FALSO	FALSO	FALSO
	Normativa UAF (2 informes)	VERDADERO	VERDADERO	FALSO
Reglas Tiempo Real	Regulación Bursátil (1)	FALSO	FALSO	VERDADERO
	Normativa CMF	VERDADERO	VERDADERO	FALSO
Interfaz de Conexión Estándar a Datos (capacidad de conectarse con distintos protocolos)	API de ejecución de reglas integradas al Negocio	FALSO	FALSO	FALSO
	Dashboard monitoreo de reglas Tiempo Real	FALSO	FALSO	FALSO
Dashboard Configurables	Describir Servicio. Definir Input de datos hacia Ceptinel. Servicio REST Document	VERDADERO	VERDADERO	VERDADERO
	Módulo de Data Connection para sincronización de lectura de datos	VERDADERO	VERDADERO	VERDADERO
Editor de Reglas	Habilitación de Widget basados en Gráficos, Listas, Eventos/Issues, Reglas	FALSO	FALSO	FALSO
	CRUD de Dashboards	FALSO	FALSO	FALSO
Librería de Procesos por industria	Widget para Monitoreo Reglas tiempo Real	FALSO	FALSO	FALSO
	Transformaciones 50, conectores a datos 16 y conectores de salidas 22	FALSO	FALSO	FALSO
Repositorio de Datos - Modelamiento (re-definir el servicio de cara a cliente) Add-Ons	Creación de Reglas con Editor gráfico [No code]	FALSO	FALSO	FALSO
	Control de Versiones de Reglas	FALSO	FALSO	FALSO
Portal Ético Add-Ons	Auditoría de Reglas, historial de ejecuciones	FALSO	FALSO	FALSO
	Reglas de la librería NO BASE	FALSO	FALSO	FALSO
Análisis Predictivo - Reglas Sugeridas Add-Ons	Mantenedor de Esquemas y creación de Listas para almacenar datos	FALSO	FALSO	FALSO
	Repositorio de Archivos con capacidad limitada	VERDADERO	VERDADERO	VERDADERO
Portal Ético Add-Ons	Sitio de Denuncias	VERDADERO	VERDADERO	VERDADERO
	Módulo de Investigación y de Casos y Denuncias	VERDADERO	VERDADERO	VERDADERO
Análisis Predictivo - Reglas Sugeridas Add-Ons	Módulo de Aprendizaje - ML (Random Forest)	FALSO	FALSO	FALSO
	Dashboard con Operaciones Inusuales entregadas por el modelo	FALSO	FALSO	FALSO

DESCRIPCION		ENTERPRISE (Personalización) Soporte Extendido (7 x 9) (7 x 24) + HH Consultoría					
Grupo de Servicios	Detalle de Servicio	AML/Compliance	Prevención de Fraude	Abuso de Mercado	Auditoría Continua	Monitoreo de Procesos	
Reglas de Negocio Base	5 reglas Base (definir librería BASE por vertical)	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO	
	Parametrización	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO	
Gestión de Alertas	Panel de Alertas y Casos	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
	Dashboard de Estadística de Gestión de Casos, Asignación de Tareas y Monitoreo de Alertas y Casos (matriz relaciones)	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
Servicio de Datos - Screening Personas	Trazabilidad de Alertas y Casos (matriz relaciones)	FALSO	FALSO	FALSO	FALSO	FALSO	
	Listas Internacionales Publicas (ONU, OFAC, CAN, AUS, EU)	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO	
	Busqueda PEP por país	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO	
	Base de Noticias Adversas	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO	
Reportes Normativos	Normativa UAF (2 informes)	VERDADERO	VERDADERO	FALSO	FALSO	FALSO	
	Regulación Bursátil (1)	FALSO	FALSO	VERDADERO	FALSO	FALSO	
	Normativa CMF	VERDADERO	VERDADERO	FALSO	FALSO	FALSO	
Reglas Tiempo Real	API de ejecución de reglas integradas al Negocio	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
	Dashboard monitoreo de reglas Tiempo Real	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
Interfaz de Conexión Estándar a Datos (capacidad de conectarse con distintos protocolos)	Describir Servicio. Definir Input de datos hacia Ceptinel. Servicio REST Document	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
	Módulo de Data Connection para sincronización de lectura de datos	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
Dashboard Configurables	Habilitación de Widget basados en Gráficos, Listas, Eventos/Issues, Reglas	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
	CRUD de Dashboards	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
	Widget para Monitoreo Reglas tiempo Real	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
Editor de Reglas	Transformaciones 50, conectores a datos 16 y conectores de salidas 22	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
	Creación de Reglas con Editor gráfico [No code]	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
	Control de Versiones de Reglas	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
	Auditoría de Reglas, historial de ejecuciones	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
Librería de Procesos por industria	Reglas de la librería NO BASE	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO	
Repositorio de Datos - Modelamiento (re-definir el servicio de cara a cliente) Add-Ons	Mantenedor de Esquemas y creación de Listas para almacenar datos	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
Almacenamiento de Archivos (diferencia por Capacidad de Archivos en BASE/EP)	Repositorio de Archivos con capacidad limitada	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO	
	Sitio de Denuncias	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO	
	Módulo de Investigación y de Casos y Denuncias	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO	
	Modelo de Aprendizaje - ML (Random Forest)	FALSO	FALSO	FALSO	FALSO	FALSO	
Análisis Predictivo - Reglas Sugeridas Add-Ons	Dashboard con Operaciones Inusuales entregadas por el modelo	FALSO	FALSO	FALSO	FALSO	FALSO	

DESCRIPCION		PREMIUM Ejecutivo de Soporte Dedicado + HH Consultoría				
Grupo de Servicios	Detalle de Servicio	AML/Compliance	Prevención de Fraude	Abuso de Mercado	Auditoría Continua	Monitoreo de Procesos
Reglas de Negocio Base	5 reglas Base (definir librería BASE por vertical)	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO
	Parametrización	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO
Gestion de Alertas	Panel de Alertas y Casos	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
	Dashboard de Estadística de Gestion de Casos, Asignacion de Tareas y Monitor Trazabilidad de Alertas y Casos (matriz relaciones)	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
Servicio de Datos - Screening Personas	Listas Internaciones Publicas (ONU, OFAC, CAN, AUS, EU)	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO
	Busqueda PEP por país	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO
	Base de Noticias Adversas	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO
Reportes Normativos	Normativa UAF (2 informes)	VERDADERO	VERDADERO	FALSO	VERDADERO	FALSO
	Regulación Bursatil (1)	FALSO	FALSO	VERDADERO	FALSO	FALSO
Reglas Tiempo Real	Normativa CMF	VERDADERO	VERDADERO	FALSO	FALSO	FALSO
	API de ejecución de reglas integradas al Negocio	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
Interfaz de Conexión Estándar a Datos (capacidad de conectarse con distintos protocolos)	Dashboard monitoreo de reglas Tiempo Real	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
	Describir Servicio. Definir Input de datos hacia Ceptinel. Servicio REST Document	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
Dashboard Configurables	Módulo de Data Connection para sincronización de lectura de datos	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
	Habilitación de Widget basados en Graficos, Listas, Eventos/Issues, Reglas	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
	CRUD de Dashboards	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
Editor de Reglas	Widget para Monitoreo Reglas tiempo Real	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
	Transformaciones 50, conectores a datos 16 y conectores de salidas 22	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
Librería de Procesos por industria	Creacion de Reglas con Editor grafico [No code]	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
	Control de Versiones de Reglas	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
Repositorio de Datos. Modelamiento (re-definir el servicio de cara a cliente) Add-Ons	Auditoria de Reglas, historial de ejecuciones	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
Almacenamiento de Archivos (diferencia por Capacidad de Archivos en BASE/EUP)	Reglas de la librería NO BASE	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO
	Mantenedor de Esquemas y creación de Listas para almacenar datos	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
Portal Etico Add-Ons	Repositorio de Archivos con capacidad limitada	VERDADERO	VERDADERO	VERDADERO	VERDADERO	VERDADERO
	Sitio de Denuncias	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO
Análisis Predictivo - Reglas Sugeridas Add-Ons	Módulo de Investigación y de Casos y Denuncias	VERDADERO	VERDADERO	VERDADERO	FALSO	FALSO
	Modelo de Aprendizaje - ML (Random Forest)	VERDADERO	FALSO	FALSO	FALSO	FALSO
	Dashboard con Operaciones Inusuales entregadas por el modelo	VERDADERO	FALSO	FALSO	FALSO	FALSO

Anexo 8 – Separación de permisos según plan contratado [1]

Referencias

- [1] Propiedad de Ceptinel
- [2] Garzas, Javier (19 Junio, 2015) ¿Qué es eso de los microservicios?
<https://www.javiergarzas.com/2015/06/microservicios.html>
- [3] Fernandez, T. (2022, September 19). Microservices Architecture: Advantages and Drawbacks. Semaphore. <https://semaphoreci.com/blog/microservice-architecture>
- [4] Bucchiarone, A., Dragoni, N., Dustdar, S., Lago, P., Mazzara, M., Rivera, V., & Sadovykh, A. (Eds.). (2020). Microservices. Springer International Publishing. <https://doi.org/10.1007/978-3-030-31646-4>
- [5] Colanzi, T., Amaral, A., Assunção, W., Zavadski, A., Tanno, D., Garcia, A., & Lucena, C. (2021, September). Are we speaking the industry language? The practice and literature of modernizing legacy systems with microservices. In *15th Brazilian Symposium on Software Components, Architectures, and Reuse* (pp. 61-70). Page 4
- [6] de la Torre, C., Wagner, B., & Rousos, M. (2023, March 21). NET Microservices Architecture for Containerized .NET Applications (EDITION v7.0 - Updated to ASP.NET Core 7.0). Co-Authors: Cesar de la Torre, Sr. PM, .NET product team, Microsoft Corp.; Bill Wagner, Sr. Content Developer, C+E, Microsoft Corp.; Mike Rousos, Principal Software Engineer, DevDiv CAT team, Microsoft. Page 26
- [7] Ibid., Page 67
- [8] Ibid., Page 69
- [9] Ibid., Page 27
- [10] Kolny, M. (2023, March 22). Scaling up the Prime Video audio/video monitoring service and reducing costs by 90%. Retrieved from <https://www.primevideotech.com/video-streaming/scaling-up-the-prime-video-audio-video-monitoring-service-and-reducing-costs-by-90>
- [11] Constanzo, M. (2019). ¿Qué es el patrón MVC?. Platzi. <https://platzi.com/tutoriales/1248-pro-arquitectura/5466-que-es-el-patron-mvc/>
- [12] Google cloud. Platform as a service (paas) vs. Infrastructure as a service (iaas) vs. Software as a service (saas). <https://cloud.google.com/learn/paas-vs-iaas-vs-saas?hl=es-419>

[13] Colanzi, T., Amaral, A., Assunção, W., Zavadski, A., Tanno, D., Garcia, A., & Lucena, C. (2021, September). Are we speaking the industry language? The practice and literature of modernizing legacy systems with microservices. In *15th Brazilian Symposium on Software Components, Architectures, and Reuse* (pp. 61-70). Page 5-7