

**Índice de Contenidos**

[Información del proyecto](#_heading=h.vmvh76j54a2m) 3

[Datos](#_heading=h.s4f9lcl1lg2c) 3

[Gerente de Proyecto](#_heading=h.d8pqr6dw6p39) 3

[Lista de Interesados (stakeholders) = programadores](#_heading=h.xqqho0g1trle) 3

[Descripción del proyecto](#_heading=h.rjw3xvqt6lqw) 4

[Objetivo de Negocio](#_heading=h.4thqik4ij7hj) 4

[Justificación del proyecto – Contexto](#_heading=h.xpx5wulqig9y) 4

[Problema-Necesidad](#_heading=h.fx5cwwsvlo7k) 5

[Requerimientos de alto nivel del Proyecto](#_heading=h.r91kpd63jugb) 6

[Descripción del Proyecto](#_heading=h.pkcu9hhctmln) 8

[Análisis Propuesta inicial (Diagnóstico de Alcances)](#_heading=h.2or9ql6npf1i) 8

[Objetivo del proyecto](#_heading=h.7eb71cvx3191) 9

[Alcances del proyecto](#_heading=h.l4j5an6z6j3u) 10

[Objetivos del desarrollo](#_heading=h.qmuvh53sjmyo) 11

[Descripción de la solución](#_heading=h.s4ln5jqtd5g9) 11

[Descripción del sistema ideado en base a los requerimiento y alcance](#_heading=h.ve1g3pn4okvp) 11

[Premisas y restricciones](#_heading=h.m671vbpmsser) 13

[Especificaciones técnicas de las herramientas de desarrollo 1](#_heading=h.11zjbejhxo6q)4

[Tipo de Infraestructura de Hardware y Sistemas de implementación y/o servicios a utilizar 1](#_heading=h.u17q9up1p2t3)6

# 

# 

# Información del proyecto

## Datos

|  | Empresa / Organización | Perfulandia |
| --- | --- | --- |
| Fecha de inicio/fin | 16/03/2025 - 8/04/2025 |
| Cliente | Empresa de Perfumes |
| Jefe de Proyecto | Bastian |

## Gerente de Proyecto

| **Nombre** | **Cargo** | **Departamento / División** |
| --- | --- | --- |
| Bastian Reyes | Scrum Master | Informática |

## Lista de Interesados (stakeholders)

| **Nombre** | **Tipo** | **Cargo** | **Departamento / División** |
| --- | --- | --- | --- |
| Cristian Lizama | Cliente | Full Stack Dev | Informática |
| Joshua Mardones | Cliente | DevOps | Informática |

# 

# Descripción del proyecto

## Objetivos de Negocio

| perfulandia SPA ha estado usando un sistema monolítico, pero con el tiempo se ha vuelto algo complicado de mantener y de hacerle cambios. Por eso, se está pensando en pasarse a una arquitectura de microservicios, que básicamente permite dividir el sistema en partes más pequeñas que se pueden manejar por separado.  La idea es que un equipo de desarrolladores trabaje con herramientas como Docker, Kubernetes y Terraform, para ir armando esta nueva estructura. Al tener el sistema dividido, es mucho más fácil escalar, arreglar errores o agregar cosas nuevas sin tener que tocar todo lo demás.  También se va a usar AWS para montar todo esto. Por ejemplo, Amazon EKS va a ayudar a manejar los contenedores, API Gateway se encarga de las solicitudes que llegan y Cognito se va a usar para que los usuarios se puedan loguear de forma segura. Para guardar los datos, se va a ocupar Amazon RDS con MySQL, que es confiable y funciona bien.  Otra parte importante es automatizar los despliegues, para que no se tenga que hacer todo a mano cada vez que se sube algo. Para eso se usarían Jenkins o GitHub Actions. Y para ver si todo anda bien, se usarían herramientas como Grafana y CloudWatch, que muestran cómo está funcionando el sistema en tiempo real.  En resumen, este cambio haría que el sistema sea más fácil de manejar, más rápido, y que la empresa pueda seguir creciendo sin quedar atrapada en una tecnología antigua. |
| --- |

## 

## 

## Justificación del proyecto – Contexto

| La verdad es que el sistema que usa actualmente Perfulandia SPA ya no está dando abasto. La empresa ha crecido bastante en poco tiempo, y eso ha dejado en evidencia varios problemas del sistema monolítico que tienen: se vuelve lento, se cae cuando hay mucha carga y cuesta mucho escalarlo o hacerle cambios sin afectar otras partes.  Todo esto complica no solo cómo se trabaja dentro de la empresa, sino también la experiencia de los clientes, que al final es clave. Si esto sigue así, podría terminar frenando el crecimiento del negocio. |
| --- |

## Problema-Necesidad

| Problema:  Perfulandia SPA se está enfrentando a dificultades debido a las limitaciones de su actual sistema monolítico ya que al inicio solo era necesario este sistema. El crecimiento acelerado de la empresa ha provocado fallos en el sistema, generando problemas de rendimiento y disponibilidad, lo que afecta negativamente las operaciones diarias y la satisfacción del cliente. Estos problemas pueden y van a ocasionar pérdidas de ventas, retrasos en la gestión de pedidos y dificultades en la administración de sucursales en distintas ciudades.  Necesidad:  La empresa Perfulandia SPA requiere una solución tecnológica escalable y eficiente que le permita gestionar su expansión sin comprometer la calidad del servicio y de este modo no perder clientes. Específicamente, necesita:   1. Migrar de una arquitectura monolítica a una de microservicios para mejorar la escalabilidad y la resiliencia del sistema. 2. Utilizar una base de datos MySQL optimizada para manejar grandes volúmenes de datos sin afectar el rendimiento. 3. Garantizar la disponibilidad y estabilidad del sistema para evitar interrupciones en las ventas y la gestión operativa. 4. Soportar el crecimiento nacional de la empresa, permitiendo la integración de nuevas sucursales sin problemas técnicos.   En conclusión, el principal desafío es modernizar la infraestructura tecnológica para garantizar un crecimiento sostenible y una mejor experiencia tanto para los clientes como para el equipo de la empresa |
| --- |

|  | Requerimientos de alto nivel del Proyecto |
| --- | --- |
| **1** | Arquitectura y Tecnologías La idea es migrar el sistema monolítico actual a una arquitectura basada en microservicios. Esto permitirá mejorar la escalabilidad, el rendimiento y la disponibilidad del sistema, adaptándose mejor a las necesidades de la empresa.  Para gestionar los microservicios, se implementarán herramientas como Docker y Kubernetes, que permitirán contenerizar los servicios y poder orquestarlos de manera eficiente.  Se utilizará AWS como plataforma en la nube, aprovechando sus servicios para facilitar la implementación y gestión de la infraestructura. Algunos de los servicios clave que se emplearán son:   * Amazon EKS, que nos ayudará a gestionar los contenedores de forma fácil y eficiente. * Amazon API Gateway para manejar las solicitudes que lleguen a los microservicios y dirigirlas a los lugares correctos. * Amazon Cognito para asegurarnos de que la autenticación y autorización de los usuarios sea segura y sencilla. * Amazon RDS con MySQL, que nos dará una base de datos relacional escalable y confiable, ideal para manejar grandes volúmenes de datos. * Terraform para automatizar la gestión de la infraestructura, lo que facilitará la creación y el mantenimiento de los entornos de manera eficiente. |
| **2** | Módulos y Funcionalidades Claves Gestión de Productos  * CRUD de productos con control de stock en tiempo real. * Sincronización del inventario entre múltiples sucursales.  Gestión de Ventas  * Registro y procesamiento de ventas al por mayor y al por menor. * Integración con medios de pago digitales.  Gestión de Clientes  * Registro y administración de clientes con historial de compras. * Implementación de programas de fidelización y descuentos personalizados.  Gestión de Sucursales  * Administración centralizada de información de cada sucursal. * Monitoreo en tiempo real del desempeño de cada tienda.  Gestión de Usuarios y Roles  * Control de acceso basado en roles (RBAC) con Amazon Cognito. * Administración de permisos para empleados y clientes.  Reportes y Análisis  * Generación de reportes de ventas, productos más vendidos y tendencias de mercado. * Implementación de dashboards con Grafana y AWS CloudWatch para análisis en tiempo real. |
| **3** | Estrategia de Migración   * Definir claramente el camino que vamos a seguir para pasar del sistema monolítico a los microservicios, para que todo esté claro desde el principio.   **Implementación Progresiva**   * Desarrollar e integrar los microservicios de manera gradual, para que no causemos interrupciones grandes en el funcionamiento del sistema. Lo vamos a hacer poco a poco, con cuidado.   **Capacitación del Equipo**   * Asegurarnos de que todo el equipo esté al tanto de las nuevas tecnologías y formas de trabajo que vamos a usar, para que todo marche bien y nadie quede atrás. |

# 

# Descripción del Proyecto

## Análisis Propuesta inicial (Diagnóstico de Alcances)

| Perfulandia SPA tiene un montón de clientes y ventas, así que su infraestructura necesita ser más flexible y capaz de crecer con el tiempo. Para eso, lo que se propone es pasar de un sistema monolítico a uno de microservicios en AWS, lo que va a hacer que todo sea más estable y rápido, y que pueda manejar el crecimiento sin caídas o problemas.  El objetivo es que todo el desarrollo, despliegue y mantenimiento sea de calidad, y para eso se van a usar herramientas que faciliten todo. La idea es migrar a microservicios, usar Cognito para la autenticación, AWS Load Balancer para distribuir las solicitudes, almacenar todo en Amazon RDS con MySQL y monitorear con Grafana y CloudWatch.  El proceso de migración va a ser organizado. Primero, se van a identificar los módulos del sistema monolítico, luego se van a crear APIs para cada uno. Después, se van a poner esos servicios en contenedores con Docker, orquestarlos con Kubernetes en EKS y usar SQS o EventBridge para la mensajería. Además, se usará Terraform para gestionar la infraestructura como código, y para automatizar los despliegues se va a usar CI/CD con Jenkins o GitHub Actions.  Claro, van a haber desafíos como mantener todo funcionando mientras se hace la transición, mejorar el rendimiento y capacitar a los equipos, pero los beneficios son claros. El sistema será más estable, escalable, y va a haber menos caídas o errores. Además, la experiencia de los usuarios va a mejorar. Lo más importante será monitorear todo el tiempo y hacer ajustes cuando sea necesario.  En resumen, todo el proyecto depende de hacerlo de forma ordenada y gestionarlo bien para que la transición sea lo más fácil posible. |
| --- |

## 

## Objetivo del proyecto

| El objetivo de este proyecto es crear una solución tecnológica en la nube de AWS, basada en microservicios, para que Perfulandia SPA pueda modernizar su sistema actual, mejorar su eficiencia, y estar lista para seguir creciendo de manera rápida. Además, la idea es ofrecer una mejor experiencia a los usuarios finales.  1.Migrar el sistema monolítico a microservicios que van a ser mucho más escalables.  2.Implementar servicios en AWS, como Amazon EKS, API Gateway, Cognito y RDS con MySQL.  3.Automatizar los procesos con CI/CD y hacer un monitoreo del sistema en tiempo real para ver cómo está funcionando todo.  Con todo esto, Perfulandia SPA va a poder asegurarse de que sus operaciones sigan funcionando sin problemas, mejorar la agilidad para desarrollar nuevos servicios, y estar lista para seguir creciendo por todo el país sin que la tecnología se quede atrás |
| --- |
|
|

## Alcances del proyecto

| Tiempo de Implementación  Este proyecto se estima que tomará unos 3 meses y 2 días.(tiempo ficticio) Con este tiempo, se espera que todo esté configurado en AWS, que se haga la transición de monolito a microservicios, y se validen todas las funcionalidades. Claro, el tiempo puede variar dependiendo de ciertos factores, pero esa es la idea inicial.  Servidor en la Nube  El servidor en la nube de AWS estará listo desde el principio. Vamos a usar servicios como Amazon EKS para manejar los contenedores, Amazon RDS con MySQL para las bases de datos, y Amazon Cognito para gestionar la autenticación de los usuarios. Todo esto se entregará al cliente cuando el proyecto esté terminado.  Personal para la Administración y Gestión  Cuando el proyecto termine, el cliente será el encargado de contratar a las personas que se encargarán de administrar todo lo que se implementó. Ese equipo tendrá que aprender a usar AWS, hacer el monitoreo con CloudWatch, y gestionar la infraestructura con Terraform.  Periodo de Prueba  Habrá un periodo de prueba de hasta 10 días después de entregar todo. En estos días se probará que los microservicios estén funcionando, que la base de datos en Amazon RDS esté bien y que todo esté comunicándose correctamente a través de API Gateway. El tiempo puede depender de lo rápido que el equipo de QA haga sus pruebas y ajustes.  Capacitación del Personal  La capacitación se hará en 5 días hábiles dentro del plazo de 3 meses. Será en las instalaciones del cliente, y se entrenará a 3 personas en cómo manejar AWS, hacer monitoreo con Grafana y CloudWatch, y gestionar la infraestructura con Terraform. |
| --- |

## Objetivos del desarrollo

| El objetivo de este proyecto es crear una solución tecnológica en la nube de AWS, basada en microservicios, para que Perfulandia SPA pueda modernizar su sistema actual, mejorar su eficiencia, y estar lista para seguir creciendo de manera rápida. Además, la idea es ofrecer una mejor experiencia a los usuarios finales.  podriamos decir que la solucion es;  Migrar el sistema monolítico a microservicios, que van a ser mucho más escalables.  Implementar servicios en AWS como Amazon EKS, API Gateway, Cognito y RDS con MySQL.  Automatizar los procesos con CI/CD y hacer un monitoreo del sistema en tiempo real para ver cómo está funcionando todo  Con todo esto, Perfulandia SPA va a poder asegurarse de que sus operaciones sigan funcionando sin problemas, mejorar la agilidad para desarrollar nuevos servicios, y estar lista para seguir creciendo por todo el país sin que la tecnología se quede atrás |
| --- |

# 

# Descripción de la solución y alcance

## Descripción del sistema ideado en base a los requerimientos y alcance

| 1. Migración a Microservicios: lo que vamos hacer es que el sistema monolítico que tienen ahora sea remplazado por uno de microservicios. Cada parte del sistema se va a hacer más flexible y fácil de mantener porque cada servicio se va a encargar de una función específica. Todo va a estar en Amazon EKS para manejar los contenedores y escalar automáticamente según se necesite.  2. Gestión de Productos y Ventas: Vamos a crear una plataforma que permita gestionar los productos, las ventas y las reservas. Los usuarios podrán ver y gestionar todo de manera fácil y rápida. Cada vez que alguien busque productos o haga una compra, todo se va a gestionar mediante microservicios independientes, lo que hará que el sistema sea más rápido y confiable.  3. Autenticación y Seguridad: La seguridad es clave, por eso vamos a usar Amazon Cognito para manejar la autenticación y asegurar que solo los usuarios autorizados tengan acceso a sus cuentas y puedan hacer transacciones sin riesgos. Todo va ser seguro para que los usuarios puedan gestionar sus pagos y reservas y tengan confianza.  4. Base de Datos en la Nube: Vamos a usar Amazon RDS con MySQL para gestionar todos los datos del sistema. Esto va a ser una base de datos escalable y con alta disponibilidad, lo que significa que el sistema va a funcionar de manera rápida y eficiente, incluso cuando haya más transacciones y crezca la cantidad de datos.  5. Plataforma Web: La plataforma web va a ser super fácil de usar y accesible desde cualquier navegador, Los usuarios podrán hacer búsquedas, hacer reservas, gestionar pagos y ver su historial de compras y reservas. Todo va a estar diseñado para que sea una experiencia fluida y rápida.  6. Automatización de Despliegues y Monitoreo: Vamos a automatizar el proceso de despliegue para que las actualizaciones del sistema sean más rápidas y sin errores. vamos a usar herramientas como Jenkins o GitHub Actions para hacer esto, y además todo el sistema va a estar monitoreado en tiempo real con CloudWatch y Grafana, así podemos ver cómo va funcionando todo y detectar posibles problemas antes de a los usuarios les pueda generar algún problema. |
| --- |

## 

## 

## Premisas y restricciones

| Premisas del Proyecto Uso Exclusivo de AWS:  Toda la infraestructura y los servicios se van a montar en la nube de AWS, aprovechando herramientas como EKS, RDS, Cognito y CloudWatch. Todo esto para asegurarnos de que la solución sea potente y se adapte a lo que necesitamos. Microservicios como Arquitectura  Principal: El sistema va a estar basado en microservicios, lo que nos va a dar mucha más flexibilidad y capacidad de escalar. Cada parte del sistema será independiente, y se va a comunicar con otras mediante APIs bien definidas. Desarrollo de una Plataforma Web  Accesible: entonces Vamos a desarrollar una plataforma web que los usuarios puedan acceder sin tener que descargar ninguna aplicación. Así, podrán hacer reservas, pagos y gestionar todo lo que necesiten directamente desde su navegador. Enfoque en la Seguridad: La seguridad es superimportante, especialmente cuando hablamos de pagos y datos sensibles. Usaremos Amazon Cognito para gestionar las cuentas de los usuarios y garantizar que todo esté bien protegido.  Escalabilidad del Sistema: El sistema tiene que ser capaz de manejar más tráfico sin que se caiga o se vuelva lento. La infraestructura va a ser capaz de escalar automáticamente según lo que necesite en cada momento.  Automatización de Despliegue;entonces Los despliegues del sistema van a ser automáticos gracias a los pipelines de CI/CD. Esto va a reducir el tiempo que tardamos en poner actualizaciones y mejoras, y también va a asegurar que todo salga bien .  Monitoreo Proactivo; Vamos a tener un monitoreo continuo para que podamos detectar cualquier posible problema antes de que afecte a los usuarios. Usaremos herramientas como CloudWatch para estar al tanto de cualquier fallo o disminución de rendimiento.  Restricciones del Proyecto  Tiempo de Implementación: Este proyecto tiene que estar listo en un tiempo específico. La primera fase incluye la migración a microservicios y el desarrollo básico de la plataforma web, y necesitamos entregarlo a tiempo.  Dependencia de AWS: Como estamos usando solo servicios de AWS, eso nos limita un poco en cuanto a opciones de personalización y depende de un único proveedor para toda la infraestructura.  Disponibilidad de Recursos: El equipo de desarrollo dependerá de la disponibilidad de gente con experiencia en herramientas como Kubernetes, Terraform, AWS y microservicios. Además, toda la infraestructura de desarrollo y pruebas se tendrá que manejar en la nube de AWS.  Adopción del Sistema: Cuando lancemos la nueva plataforma web, puede ser que los usuarios necesiten un poco de tiempo para acostumbrarse. Puede que necesitemos hacer algunos ajustes en la interfaz o dar capacitación para que la transición sea más fácil ordenalo con numeros no quiero que modifiques nada solo ponle numero |
| --- |

## 

## Especificaciones técnicas de las herramientas de desarrollo

| base de datos Gestiona Contenedores y microservicios etc Contenerizador    jenkins CI/CD Nube Infraestructura como codigo  Lenguaje de programación       * **AWS**: Es como un gran centro de datos en la nube donde puedes alojar servidores, bases de datos y otros recursos sin necesidad de comprar hardware. Ideal para escalar aplicaciones sin preocuparte por la infraestructura física. * **Terraform**: Es una herramienta que te permite definir y gestionar toda la infraestructura en código. En lugar de crear servidores o bases de datos manualmente en AWS, con Terraform escribes un archivo y lo despliegas automáticamente. * **Jenkins**: Es como un asistente de automatización para desarrolladores. Te ayuda a construir, probar y desplegar código sin intervención manual, asegurando que todo funcione antes de salir a producción. * **Kubernetes**: Maneja contenedores y aplicaciones de forma automática. Si tu aplicación necesita varios servidores, Kubernetes los organiza, los escala y los mantiene funcionando sin que tengas que intervenir constantemente. * **Docker**: Permite crear contenedores, que son como pequeñas cajas donde metes tu aplicación con todo lo necesario para que funcione en cualquier computadora o servidor sin problemas de compatibilidad. * **Java**: Un lenguaje de programación súper usado en el mundo empresarial. Se usa mucho para hacer aplicaciones web, microservicios y software de alto rendimiento. * **MySQL**: Es una base de datos donde se almacena la información de tu aplicación, como usuarios, compras o publicaciones. Funciona con SQL, un lenguaje para gestionar datos de manera eficiente.   Básicamente, todas estas herramientas trabajan en conjunto. |
| --- |

## 

## 

## 

## Tipo de Infraestructura de Hardware y Sistemas de implementación y/o servicios a utilizar

| La infraestructura propuesta en este proyecto estará completamente basada en AWS, lo que proporciona una solución escalable, eficiente, segura y altamente disponible de este modo nos aseguramos de que la empresa a futuro pueda seguir creciendo sin problemas ya que el nuevo sistema basado en microservicios les permitirá tener una estabilidad a futuro. EKS y RDS Cognito, CloudWatch y otras herramientas de AWS ocuparemos para asegurar que los servicios sean flexibles, con la capacidad de escalar automáticamente según las demandas del negocio y a la vez que se mantienen los más altos estándares de seguridad y rendimiento. Esta infraestructura permitirá a Perfulandia SPA continuar su crecimiento, manteniendo un sistema robusto y capaz de ofrecer un excelente servicio a sus usuarios. |
| --- |

# Organización del equipo Proyecto

## 

