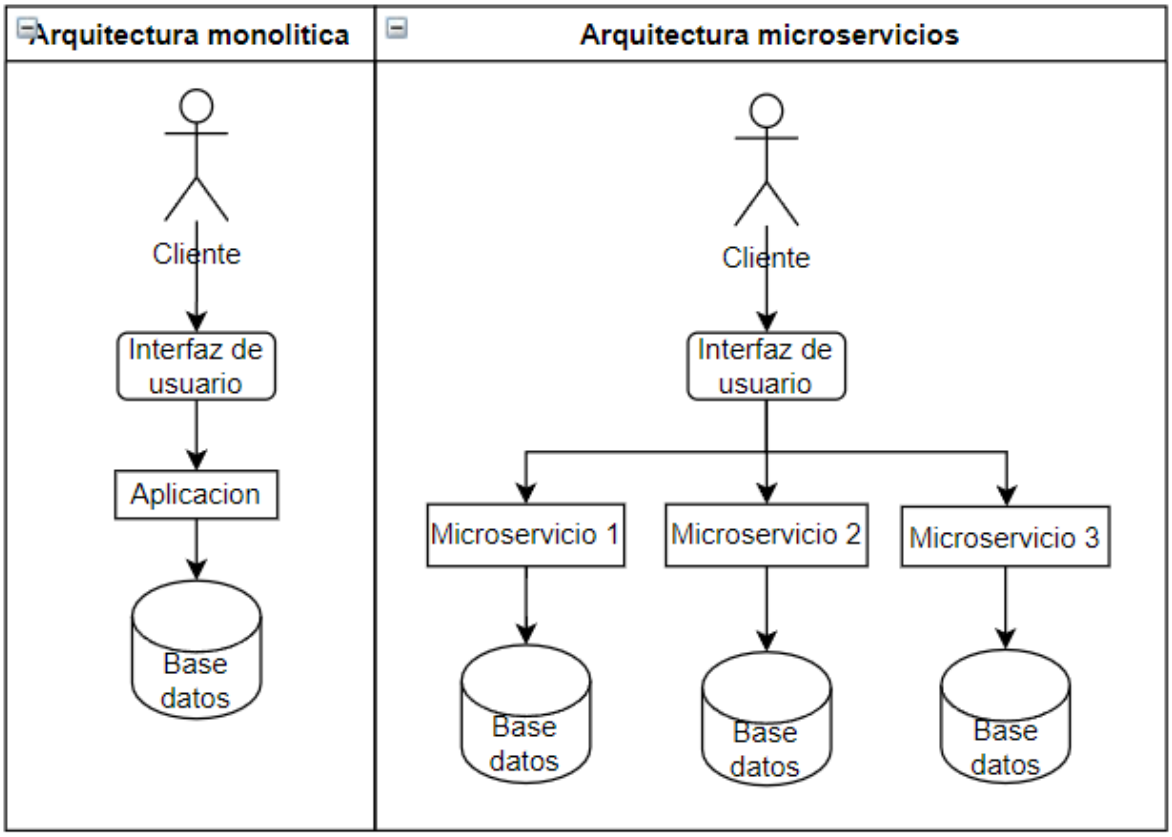
**Virtualización y programa distribuido en la nube con AWS**

Hace un tiempo se utilizaban mucho las arquitecturas monolíticas, donde los desarrollos de software se estructuran de tal forma que los aspectos funcionales de quedan acoplados en un mismo programa, donde todo se encuentra centralizado en un mismo servidor y no hay separación de módulos o funcionalidades. Esta forma de trabajar reduce el tiempo de implementación pero aumenta los tiempos de despliegue y de implementación de mejoras o nuevas funcionalidades ya que implica volver a cargar todo el producto relacionado.

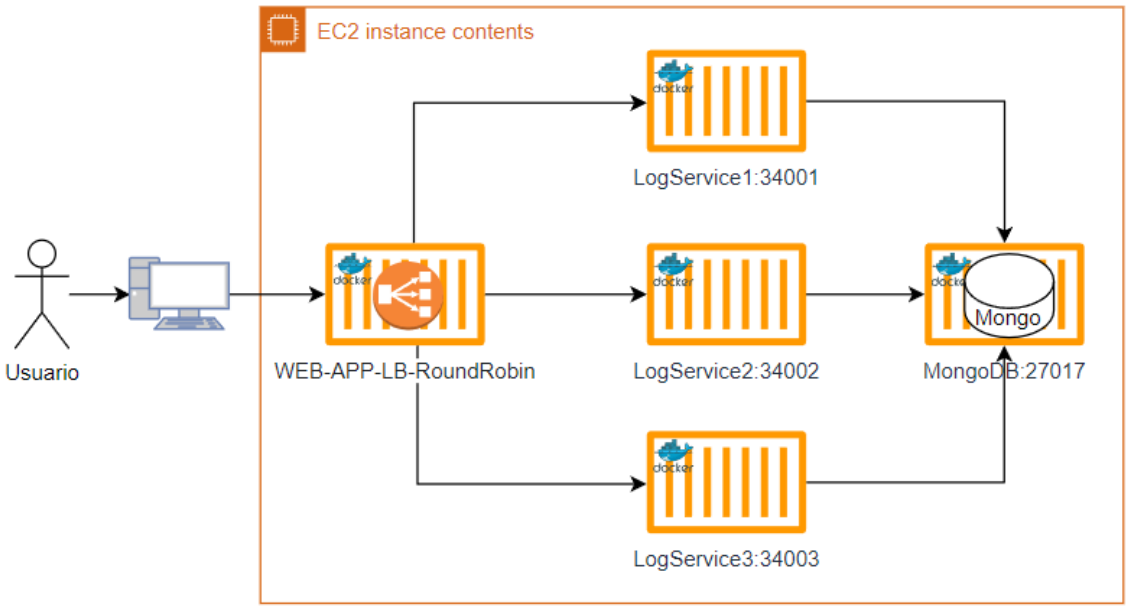
Por estas y otras dificultades de las arquitecturas monolíticas surgen las arquitecturas de microservicios, esta arquitectura tiene como objetivo separar los diferentes componentes de una aplicación con el fin de que cada componente sea una funcionalidad y una aplicación independiente por si misma que al sumarse en conjunto con otras funcionalidades en conjunte formen toda una aplicación funcional desacoplada. Una de las mayores ventajas que trae implementar esta arquitectura, es que facilita la modularidad y la escalabilidad frente a distintos niveles.




Los microservicios están muy ligados al concepto de contenedor, una unidad estándar de software que empaqueta el código junto a todas de sus dependencias para que el servicio o aplicación se ejecute de forma rápida y fiable de un entorno de software informático a otro. Por otro lado tenemos el concepto *Docker*, una

plataforma de contenedores que incluyen en ellos todo lo necesario para que dicho software se ejecute de forma aislada y eficiente.

La arquitectura que se mostrara a continuación cumple de cierta manera una arquitectura de microservicios donde se empaquetan cada uno de los componentesn en contendedores *Docker,* esta arquitectura está compuesta de un balanceador de cargas, tres aplicaciones REST y una base de datos NoSql de Mongo y esta aplicación desplegada en  la nube de AWS.




1 *Imagen*

El componente principal de esta arquitectura es la aplicación REST la cual tiene una petición POST que recibe una cadena, la cual es persistida en la base de datos de mongo y luego devuelve las 10 cadenas persistidas más recientes. Para tener mayor disponibilidad y escalabilidad de la aplicación se tienen tres instancias encargado de almacenar la cadena ingresada por el usuario. Las tres instancias desplegadas deberán tener conexión con la base de datos de MongoDB para poder persistir las cadenas ingresadas por el usuario.

Para poder tener acceso a las tres instancias de los servicios REST se utiliza un balanceador de carga Round-Robin, el cual reparte las cargas uno a uno por cada instancia desplegada. *ver.imagen1.* Se ha desplegado la aplicación en la nube de AWS en una máquina virtual con una instancia EC2.

Se puede acceder a nuestra aplicación digitando directamente la URL, luego el Front-End llamado WEB-APP-LB-RoundRobin se recibe y se asignan las peticiones a cada una de las instancias desplegadas llamadas LogServicex, estas instancias utilizan los puertos 34001,34002 y 34003 para comunicarse usan la librería de Mongo Morphia para la conexión con la base de datos MongoDB.