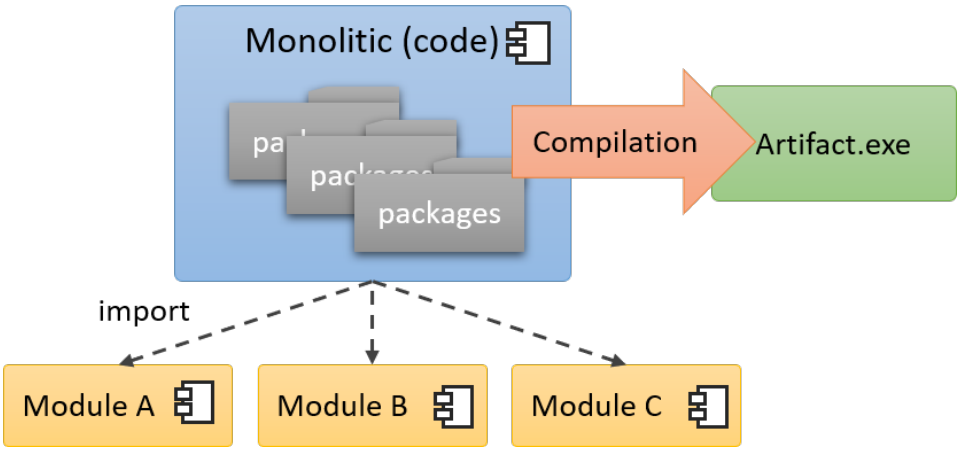
Microservicios

Arquitectura Monolítica

Esta es la arquitectura en la que la aplicación está contenida en un solo modelado de clases, posee una sola lógica de negocio, tiene una sola base de datos la cual está centralizada para toda la aplicación, además, todo el código fuente de la aplicación se encuentra en un solo proyecto, el cual se compila en un solo archivo ejecutable.



Ventajas

- Fácil de desarrollar y mantener
- No dependen de servicios o aplicaciones externas

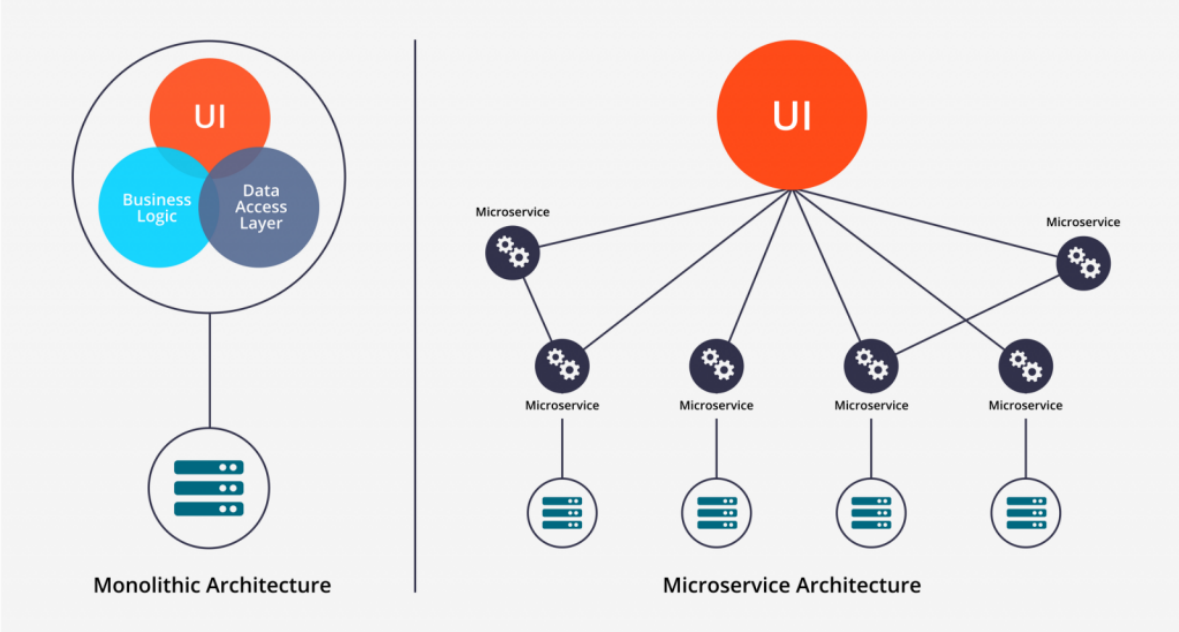
Desventajas

- Gran complejidad al crecer a gran escala
- Mayor probabilidad de fallos y cuellos de botella

Con las arquitecturas monolíticas, todos los procesos están estrechamente asociados y se ejecutan como un solo servicio. Esto significa que, si un proceso de una aplicación experimenta un pico de demanda, se debe escalar toda la arquitectura. Agregar o mejorar las características de una aplicación monolítica se vuelve más complejo a medida que crece la base de código. Esta complejidad limita la experimentación y dificulta la implementación de nuevas ideas. Las arquitecturas monolíticas aumentan el riesgo de la disponibilidad de la aplicación porque muchos procesos dependientes y estrechamente vinculados aumentan el impacto del error de un proceso.

Arquitectura de Microservicios

Es una arquitectura en la que cada uno de los módulos de las aplicaciones se encuentran divididos en varias secciones, lo que significa que cada módulo, tiene su propia base de datos, y la comunicación entre estos se realiza por medio de API's.



Ventajas

- Al tener los módulos separados, la aplicación puede funcionar de manera más fluida.
- Si falla un módulo, no falla toda la aplicación.

Desventajas

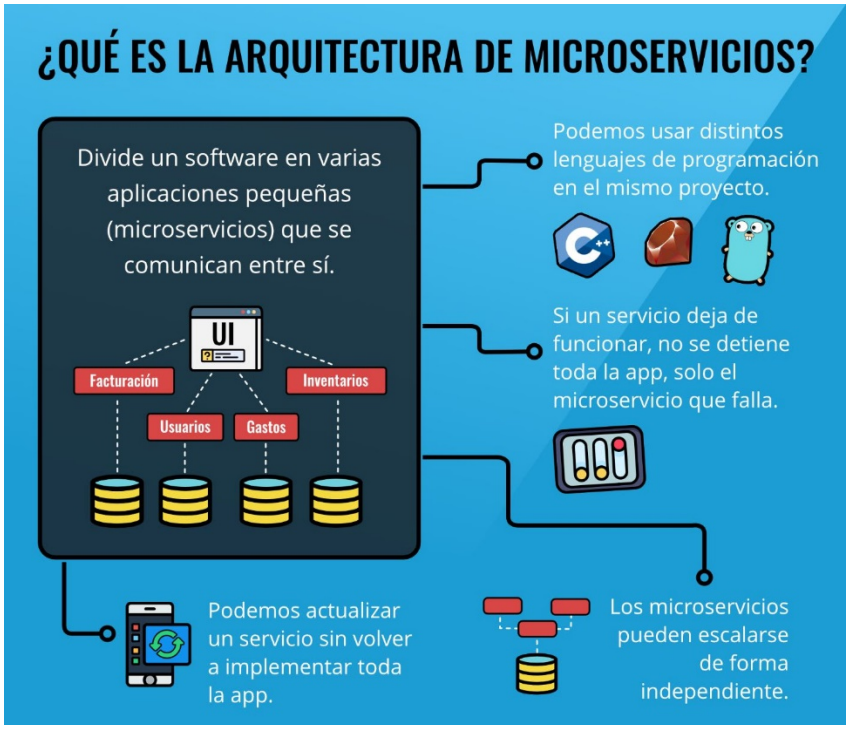
- Complejidad que puede alcanzar.
- Se pueden producir fallos de comunicación entre módulos (microservicios).
- Requiere mayor infraestructura.
- A mayor infraestructura, mayor costo.

Con una arquitectura de microservicios, una aplicación se crea con componentes independientes que ejecutan cada proceso de la aplicación como un servicio. Estos servicios se comunican a través de una interfaz bien definida mediante API ligeras. Los servicios se crean para las capacidades empresariales y cada servicio desempeña una sola función. Debido a que se ejecutan de forma independiente, cada servicio se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación.

Características de los microservicios

Autónomos
Cada servicio componente en una arquitectura de microservicios se puede desarrollar, implementar, operar y escalar sin afectar el funcionamiento de otros servicios. Los servicios no necesitan compartir ninguno de sus códigos o implementaciones con otros servicios. Cualquier comunicación entre componentes individuales ocurre a través de API bien definidas.

Especializados
Cada servicio está diseñado para un conjunto de capacidades y se enfoca en resolver un problema específico. Si los desarrolladores aportan más código a un servicio a lo largo del tiempo y el servicio se vuelve complejo, se puede dividir en servicios más pequeños.



Como ya se mencionó anteriormente, los microservicios son un enfoque arquitectónico para la construcción de aplicaciones, en las que estas se dividen en pequeños servicios independientes, cada uno de los cuales se encarga de realizar una tarea específica, estos servicios se comunican entre sí por medio del protocolo de comunicación HTTP.

Spring Boot es una excelente opción, ya que cuenta con una gran cantidad de características. En primer lugar, Spring Boot simplifica el proceso de configuración de la aplicación y la gestión de dependencias, cosa que permite a los desarrolladores centrarse en la lógica de negocio en lugar de en los detalles de

implementación. Y no solo eso, Spring Boot ofrece un gran número de bibliotecas y herramientas que facilitan el desarrollo de aplicaciones web, de bases de datos y de servicios REST, aumentando significativamente la productividad y los costos de desarrollo. Por último, Spring Boot es altamente escalable. Los desarrolladores pueden crear aplicaciones que crecerán y se adaptarán a medida que cambien los requisitos de negocio.

La elección de una arquitectura u otra, dependerá de la necesidad que tenga el escenario en cuestión, por lo general una arquitectura monolítica es suficiente para una aplicación pequeña y que se tiene previsto que no crecerá demasiado, por el contrario, si se trata de una aplicación empresarial, que tiene planeado crecer con el tiempo, una arquitectura de microservicios es una mejor opción.