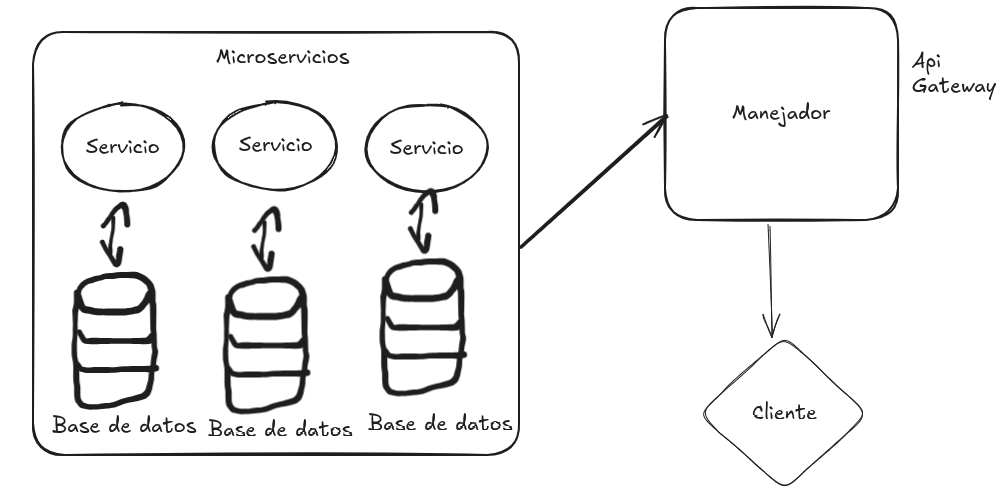
Articulos

# 1.Arquitectura de microservicios

La **arquit ectura de microservicios** surge como alternativa al estilo **monolítico**, donde las aplicaciones empresariales se construyen como una unión de tres partes: cliente, manejo de datos y servidor con lógica de dominio. Aunque este enfoque puede ser exitoso, los constantes cambios en el modelo de negocio hacen que la aplicación crezca, se complique su mantenimiento y su escalabilidad incremente la capacidad de toda la aplicación en lugar del segmento que lo requiere.

Para enfrentar este problema, la arquitectura de microservicios propone construir una aplicación como un conjunto de **servicios independientes**, incluso en diferentes lenguajes y equipos, lo que permite escalar solo el servicio que lo necesita. Ejemplos como Amazon, eBay y Netflix adoptaron este enfoque. Cada servicio tiene su propia **arquitectura hexagonal** y base de datos independiente, lo que asegura bajo acoplamiento y diversidad tecnológica.



# Reflexion:

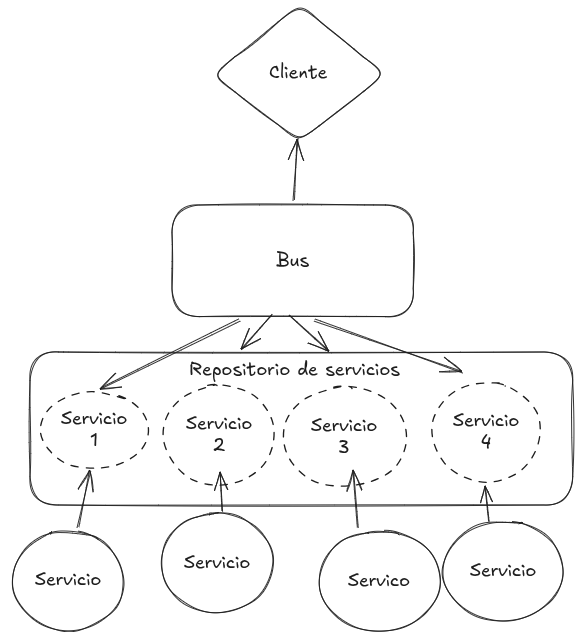
La arquitectura de microservicios no solo representa una solución efectiva para proyectos de gran escala, sino que, gracias a la independencia y flexibilidad entre servicios, permite reutilizar microservicios de un proyecto en otro, lo que ayuda a reducir el tiempo y los costos de desarrollo. Además, fomenta la creación de soluciones dinámicas y adaptables a los cambios en cualquier tipo de proyecto.

# Bibliografía:

Contreras, D. A. B. (2018). Arquitectura de microservicios. *Tecnología Investigación y Academia*, *6*(1), 36-46.

# 2.Arquitectura orientada a servicios

La arquitectura de software es una parte crucial de la ingeniería de software, encargada de estudiar, analizar y dibujar el mapa completo de un sistema. Se concentra mucho, en las conexiones entre sus piezas principales.  
En este ámbito, aparece la Arquitectura Orientada a Servicios (SOA) como un enfoque que une la TI con las exigencias de la empresa. SOA permite fusionar sistemas diversos, utilizando servicios independientes, estos se publican, encuentran y activan gracias a interfaces normales, sin importar el lenguaje o la plataforma empleada.  
El corazón de SOA late gracias a servicios autónomos que hablan a través de un bus de servicios y un repositorio. Facilita la reutilización, la integración y la gran flexibilidad. Hay servicios básicos, de negocio, de procesos o públicos, usando, frecuentemente, servicios web que usan estándares como XML, SOAP y WSDL.  
De esta manera, la productividad recibe un gran empujón, ademas de la seguridad y la flexibilidad organizacional frente a las transformaciones.



# Reflexion:

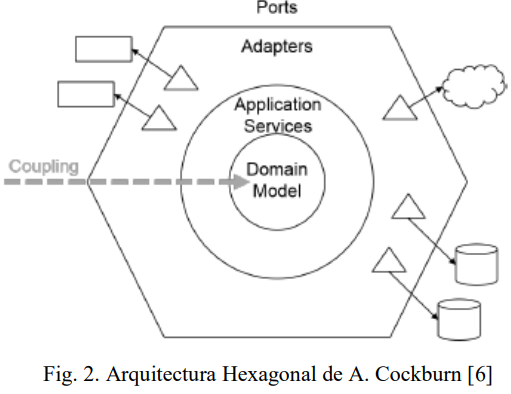
La arquitectura orientada a servicios puede reflejar un trabajo en equipo, cada persona representa un servicio el cual trabaja y funciona independientemente pero uniendo 2 o mas, se puede construir o hacer funcionar algo. El bus puede ser como un líder, el cual es el que solicita o requiere esos servicios para unificarlo o manejarlo para que se pueda usar.

# Bibliografía:

Martín, Y. E. (2012). Arquitectura de software. Arquitectura orientada a servicios. *Serie Científica de la Universidad de las Ciencias Informáticas*, *5*(1), 1-10.

# **3.Arquitectura hexagonal**

La arquitectura hexagonal es un patrón estructural que establece entradas y salidas en los bordes del diseño, permitiendo intercambiar manejadores sin cambiar el código del núcleo. De esta forma, la aplicación puede ser manejada por usuarios, programas, pruebas automatizadas o scripts por lotes, y desarrollada de manera aislada de dispositivos y bases de datos en tiempo de ejecución.  
  
A diferencia de la arquitectura en capas, la arquitectura hexagonal permite que la interfaz de usuario también se pueda intercambiar. El modelo resultante se compone del modelo de dominio, los servicios de aplicación, los puertos y los adaptadores. Cada lado del hexágono representa un puerto concreto, aunque en la práctica puede haber más puertos con su correspondiente adaptador.  
  
El modelo de dominio, como núcleo, es independiente de los servicios de aplicación y adaptadores. Esto facilita cambiar frameworks o herramientas escribiendo un nuevo adaptador. El interior del hexágono se mantiene libre de cuestiones tecnológicas, y la comunicación se da mediante interfaces llamadas puertos.



Reflexion:

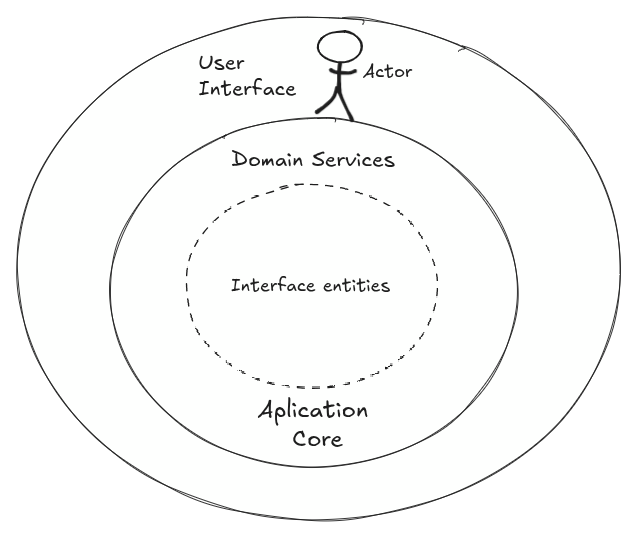
La arquitectura hexagonal propone que en lugar de construir sistemas dependientes de frameworjs, base de datos o interfaces graficas, planeta que el “corazon debe ser el dominio”, esto quiere decir las reglas y logica de negocio que realmente dan valor,

Bibliografia:

Cambarieri, M., Difabio, F., & García Martínez, N. (2020). Implementación de una Arquitectura de Software guiada por el Dominio. In *XXI Simposio Argentino de Ingeniería de Software (ASSE 2020)-JAIIO 49 (Modalidad virtual)*.

4.Arquitectura onion

La arquitectura onion combina las capas de Domain Driven Desing (DDD), que permite colocar en el centro la aplicación. En ese sentido, cada microservicio diseñado consta de este principio, donde la capada de interfaz de usuario, se encarga de la presentación de la información al usuario, lo que permite la menor exposición del usurario. En la capa de dominio, se establece la lógica general del desarrollo de la plataforma web, permitiendo establecer la capa que contiene los datos y lógica de manipulación de los mismo. En la capa de aplicación, se establecen los casos de uso, que permiten la activación por medio de la application core mediante interface de usuario. La capa de datos, queda la interacción de las bases de datos empeladas en el desarrollo de la plataforma web.



Reflexion:

Bibliografia:

Velásquez Meriño, H. D., & Viloria, J. (2025). Plataforma web con inteligencia artificial para la evaluación de habilidades de pensamiento crítico en estudiantes de la Corporación Universitaria del Caribe CECAR utilizando los TEST PC-B y TEST PC-A.