



La tecnología utilizada fue .net core 8 debido a que es un framework moderno, multiplataforma que puede ser contenerizado y escalable en múltiples ambientes ya sea en la nube u on-premise

Dentro de la solución se pueden visualizar múltiples patrones de programación, como son repositoryPattern, UnitOfWork, Shared Library Pattern, Dependency Injection & Separation of Concerns, MemoryCache (Cache-Aside Pattern), Event-Driven Architecture (EDA)

El objetivo fue realizar la solución apegado a los conceptos de clean architecture de .net, en este caso los servicios se dividieron en 3 soluciones para que fueran lo más independiente posible. Esto brinda un código más limpio, mantenible, escalable y reutilizable, además de que aísla las fallas, ya que si el servicio de inventario se cae los demás no se ven afectados.

Adicionalmente se incluye un proyecto para abarcar la autenticación y autorización de usuarios... Este proyecto puede ser totalmente independiente e incluso tener su base de datos aparte, ya que su único objetivo es emitir un jwt token con las acciones permitidas que tiene el

usuario, las mismas estarán relacionada con los roles. Cada endpoint desarrollado contiene la acción que se requiere para poder consumirlo

Los microservicios de Productos e Inventario, están totalmente separados, sin embargo la capa de Dominio e Infraestructura decidí hacerla compartida ya que los modelos se relacionan entre sí, y la base de datos para un tema de productos - inventario lo más probable es que la misma sea compartida.