

# Arquitectura Goal Challenge

- Basa en el patrón microservicios CQRS y DDD.
- CQRS es un patrón de arquitectura que separa los modelos para leer y escribir datos.
- Lecturas mediante EF Core<sup>1</sup>.
- Lecturas y escrituras basadas en Entity Framework Core<sup>2</sup>.
- Uso de un mediador (MediatR) para la comunicación de los microservicios (Commands).
- Toda la arquitectura está basa en la tecnología .NET 6.
- Comandos y Queries hacen una sola cosa, principio de responsabilidad única de SOLID (single responsibility principle)
- Contenedor de inversión de dependencias con Autofac aplicando el principio de Inversión de dependencia SOLID
- Inyección de interfaces necesarias en cada método o caso de uso para cumplir el principio de segregación de interfaces.

Nota 1: Habitualmente en mis arquitecturas, suelo usar Dapper (micro ORM) para la ejecución de las consultas, todo con dynamic como resultado de la operación. El rendimiento es muy superior a las consultas que pueda hacer EF Core en muchas circunstancias.

Nota 2: Debido a la naturaleza de la prueba usando EF inMemory, las consultas se han hecho en EF Core, de todas formas, he implementado un proyecto de ayuda o helper que expone más sencillamente las herramientas más comunes para usar Dapper (GoalChallenge.Data.Dapper).

## Representación de las dependencias a nivel de proyecto

