

MasterClass

Web API

1-Sobre el placer invisible del back end

2- Sobre OOP y Arquitectura (esta vez, con memes)

3-Sobre el flujo de datos (sin memes)

4-Dónde estamos, qué queremos, de dónde venimos, quiénes somos, por qué hacemos tantas preguntas, cuál es el sentido de la vida en el tiempo infinito del universo intergaláctico. Y esas cosas. (pendiente)

5-Katas C# OOP y apuntes sintaxis. (pendiente)

Sobre el placer invisible del backend <3

De no saber aún tanto o todo

Pero sentir que tira

Y empezar a investigar

Como si fuera un juego

Trazar líneas y relaciones

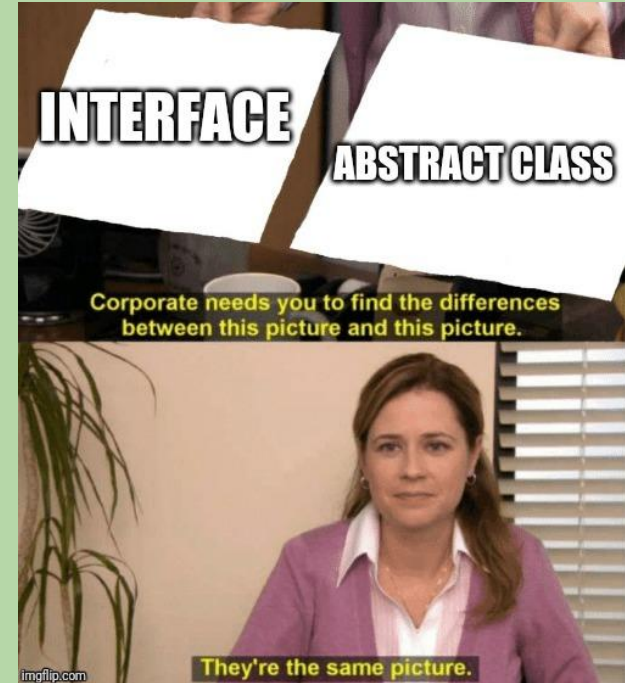
En el infinito mundo de las formas

Que a veces nos sobrepasa

Y por eso es tan bello



Clases e Interfaces...



Un breve apunte sobre las clases...

Clase

- Tipo de objeto

- Heredable

- Instanciable

- Puede tener atributos y métodos **definidos** o “concretos” (incluso si es una clase abstracta).

Un breve apunte sobre las interfaces...

Interfaz

- Rol

- Implementable

- No instanciable

- No puede tener atributos y métodos **definidos**

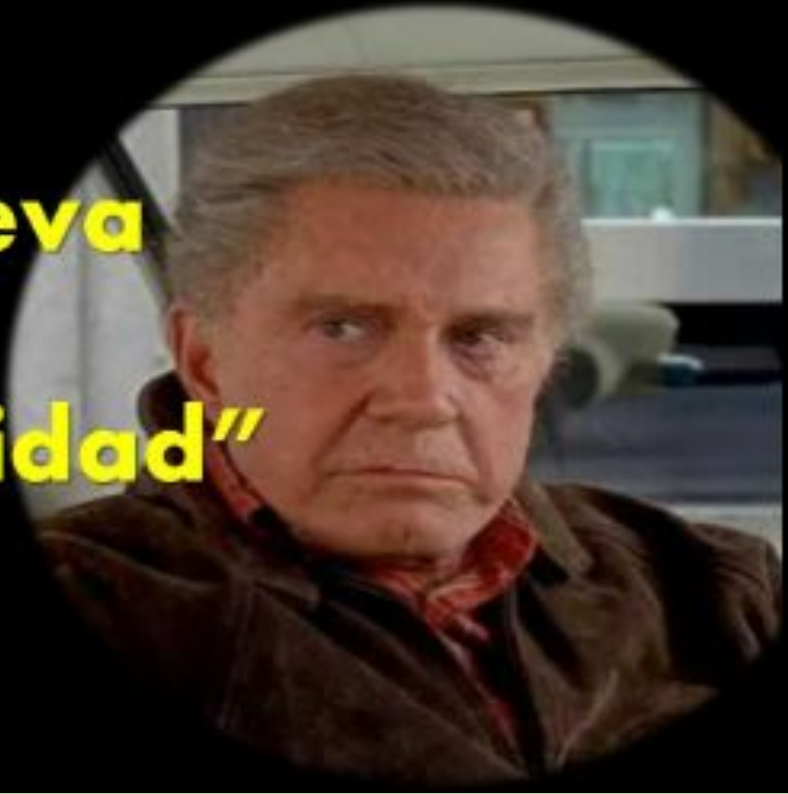
Un poquito más...

Si hay una función en el código que deban poder cumplir clases que no sean necesariamente del mismo tipo, entonces esa función se define en una interfaz.

La interfaz fuerza a las clases que la implementan a tener ciertos métodos, sin importar del tipo que sean.

Les da “poder” o “autorización” a las clases que la implementan, pero también “responsabilidad”...acaso vean venir el siguiente spider-meme...

“Un gran poder conlleva una gran responsabilidad”



La interfaz “da” a la clase la posibilidad de cumplir un rol...
PERO la fuerza a la obligación de responsabilizarse por ciertos métodos.

Ajum...parece que
ApuService no está
implementando bien
los métodos de
IBomberoService,
o sí?
Acaso...?



Apuntirijillo!

En el cielo infinito y misterioso de la programación, las interfaces sirven para muchas cosas que podemos comprender dentro del concepto de “rol”, no solamente para definir dependencias (como las que venimos inyectando en Program.cs).



Arquitectura, capas y flujo de datos

Tipos de arquitectura, según representación gráfica oficial de microsoft:



Apunte sobre el segundo tipo, que es el propuesto en el proyecto:

- La solicitud se presenta al Controlador
- El o los Servicio/s la tramita/n y hace/n lo que haya que hacer
- Si es necesario interactuar con una base de datos y/o hacer validaciones de negocio, el Servicio llama a la Lógica
- La Lógica hace las validaciones necesarias, verifica/opera según la lógica de negocio y en función de ello, si es necesario, luego impacta en la base de datos
- La base de datos hace lo suyo y devuelve una respuesta.
- Al llegar al final del recorrido luego va retornando capa por capa hasta el Controlador, que responde a la solicitud inicial.

Controlador



Servicios



Lógica



Base de Datos

O sea más o menos...

HAY QUE LLAMAR AL **AGUA**
PARA QUE APAGUE AL FUEGO
EL AGUA NO QUIERE APAGAR EL FUEGO
EL FUEGO NO QUIERE QUEMAR EL PALO
EL PALO NO QUIERE PEGARLE AL LOBO
EL LOBO NO QUIERE SACAR A LA CHIVA
LA CHIVA NO QUIERE SALIR DE AHÍ.



Parece engorroso todo esto de las capas y tal, lo séeeee.

Dénle una vuelta y ténganlo en cuenta.

En todo caso, es decisión suya y pueden hacer la arquitectura como consideren mejor.