

## Refactorización del código del proyecto Pokemon :

## • Renombrado

- Renombrado de respuesta a multiplicador para mas expresividad

```
fun efectividad(otro:Tipo):Double{
    var respuesta = 0.0
    if (supereficaces[otro.tipo]?.contains(this.tipo) == true) respuesta = 2.0 else{
        if (listanomuy[otro.tipo]?.contains(this.tipo) == true) respuesta = 0.5 else respuesta = 1.0
    }
    return respuesta
}
```

- 

```
fun efectividad(otro:Tipo):Double{
    var multiplicador = 0.0
    if (supereficaces[otro.tipo]?.contains(this.tipo) == true) multiplicador = 2.0 else{
        if (listanomuy[otro.tipo]?.contains(this.tipo) == true) multiplicador = 0.5 else multiplicador = 1.0
    }
    return multiplicador
}
```

- 

- Renombrado de danio a potencia en ataque porque nos daba toc y porque es mas expresivo, danio suena mas a representar el daño definitivo que hace el ataque

```
class Ataque(nombre: String, tipo:Tipo, danio:Int) {

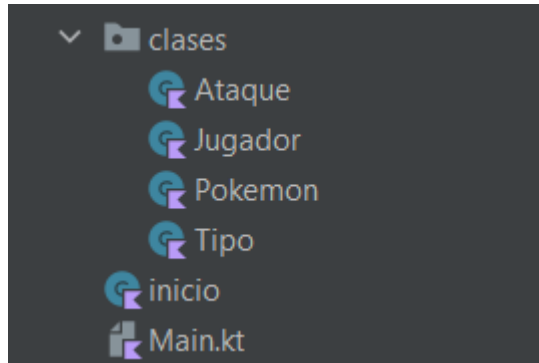
    var nombre : String
    var tipo : Tipo
    var potencia : Int

    fun efectividad(otro:Tipo):Double{
        var multiplicador = 0.0
        if (supereficaces[otro.tipo]?.contains(this.tipo) == true) multiplicador = 2.0 else{
            if (listanomuy[otro.tipo]?.contains(this.tipo) == true) multiplicador = 0.5 else multiplicador = 1.0
        }
        return multiplicador
    }

    fun atacar(otro:Tipo){
        val danio = this.potencia * multiplicador
        otro.recibirDanio(danio)
    }
}
```

-

- Sustitucion de bloques
  - -
- Campos encapsulados
  - Ya lo están
- Movimieto de clases



- 
- Extraer interfaz
  - -
- Cambios de parámetros
  - Efectividad ahora recibe directamente el ataque y saca por su cuenta el tipo, para que la comprension en recibir ataque sea mejor

```
fun efectividad(otro:Tipo):Double{
    var multiplicador = 0.0
    if (supereficaces[otro.tipo]?.contains(this.tipo) == true) multiplicador = 2.0 else{
        if (listanomuy[otro.tipo]?.contains(this.tipo) == true) multiplicador = 0.5 else multiplicador = 1.0
    }
    return multiplicador
}
```

```
fun efectividad(ataque: Ataque):Double{
    var multiplicador = 0.0
    if (supereficaces[ataque.tipo.tipo]?.contains(this.tipo) == true) multiplicador = 2.0 else{
        if (listanomuy[ataque.tipo.tipo]?.contains(this.tipo) == true) multiplicador = 0.5 else multiplicador = 1.0
    }
    return multiplicador
}
```

```
fun recibir_ataque(ataque: Ataque) {
    vida -= (ataque.potencia*tipo.efectividad(ataque))
}
```

- 
- Mover a otro nivel
  - -
- Borrado seguro
  - Borrado de fuerza de la clase pokemon, no se usa en ningun momento

```
class Pokemon(nombre:String, lore:String, tipo:Tipo, fuerza:Int, vida:Int) {
    var nombre : String
    var lore : String
    var tipo : Tipo
    var fuerza : Int
    var vida : Double
}
```

```
class Pokemon(nombre:String, lore:String, tipo:Tipo, vida:Int) {
    var nombre : String
    var lore : String
    var tipo : Tipo
    var vida : Double
}
```