

/* Contar con un filtro */

```
method cantidadEmpresas() {  
    return propiedades.count { propiedad => propiedad.sosEmpresa() }  
}
```

/* Find or else */

```
method hipotecar() {  
    const propiedad = propiedades.findOrElse(  
        { propiedad => propiedad.puedeHipotecarse() },  
        { throw new DomainException(message = "Dinero insuficiente") }  
    )  
    propiedad.hipotecarPara(self)  
}
```

/* Dominio de numeros */

```
method tirarUnDado() {  
    return 1.randomUpTo(7)  
}
```

/* Para no poner igual */

```
method esDe(unDuenio) {  
    return duenio.equals(unDuenio)  
}
```

/* Recursividad */

```
method casillerosDesde(unCasillero) {  
    const casillero = casilleros.first()  
    if (unCasillero.equals(casillero)) {  
        return casilleros  
    } else {  
        casilleros.remove(casillero)  
        casilleros.add(casillero)  
        return self.casillerosDesde(unCasillero)  
    }  
}
```

/* Times */

```
method pasarMinutos(minutos) {  
    minutos.times { i => self.pasarUnMinuto() }  
}
```

/* Sorted by */

```
method misionesOrdenadasPorPuntos() = misiones.sortedBy({  
    mision1 , mision2 => mision1.puntosRecompensa() <  
    mision2.puntosRecompensa()  
})
```

/* Agregar una lista */

```
method agregarHuevos(huevos){  
    huevosBuscados.addAll(huevos)  
}
```

```
/* Indice */
```

```
method denunciaRegistradaPor(unaVictima){  
    return !denuncias.isEmpty() and denuncias.get(nroDenuncia-1).victima() ==  
    unaVictima  
}
```

```
/* flatMap */
```

```
method elementosDeNiniosGlotos() {  
    return self.niniosGlotos().flatMap { ninio => ninio.elementos() }  
}
```

```
/* Remover los que cumplen */
```

```
method limpiezaPacks() {  
    packs.removeAllSuchThat({ pack => pack.esInutil()})  
}
```