

## Droide



Villanueva Carlos Alberto | FPR-2021-2 | 2/12/2021 11:55:16 | 90 minutos

Villanueva Carlos Alberto

FPR-2021-2

2/12/2021 11:55:16

**class Droide :****public Droide ()**
**ArrayList<Droide> droidesFueraDeServicio    ArrayList<Pieza> piezasOperativas    ArrayList<Pieza> piezasNoOperativas**

droidesFueraDeServicio = new ArrayList&lt;&gt;()

piezasOperativas = new ArrayList&lt;&gt;()

piezasNoOperativas == new ArrayList&lt;&gt;()

**class Droide :****private ResultadoReparacion autoRepararse ()**
**Pieza pieza    int i = 0    Pieza piezaOperativa    boolean encuentreUnaPieza = false**

i &lt; this.piezasNoOperativas.size()

pieza = piezasNoOperativas.get(i)

/\* Guardo la referencia de la pieza no Operativa, y la uso para busca en la lista de piezas Operativas con el nombre \*/

/\* porque no queremos encontrar la misma pieza que esta rota, sino una con el mismo nombre \*/

piezaOperativa = buscarPiezaOperativa(pieza.getNombre())

piezaOperativa != null

V

F

/\* reemplazar la pieza rota con la pieza operativa \*/

reemplazarPieza(pieza, PiezaOperativa)

encuentreUnaPieza = true

```

/* Si nunca entro al while, el droide esta completamente operativo, si entro un vez, al menos la reparacion es Parcial */
/* y si entro una vez pero el aun tiene piezas para reparar (size() > 0) la reparacion es Imposible */

return obtenerResultadoReparacion(encontreUnaPieza)

```

**class Droide :**

**private Pieza buscarPiezaOperativa ( String nombrePieza )**

**int i = 0   Pieza piezaBuscada = null   Droide droide**

```

i < droidesFueraDeServicio.size() && piezaBuscada == null

```

```

droide = droideFueraDeServicio.get(i)

```

```

/* Busamos la pieza requerida en la lista de piezas operativas del droide en la posicion que estamos recorriendo */

```

```

piezaBuscada = droide.getPiezaOperativa(nombrePieza)

```

```

/* Se aumenta siempre el indice, porque si encontramos la pieza, ya no va a entrar por while */

```

```

/* ya que es != de null o porque nos quedamos sin droides */

```

```

i++

```

```

return piezaBuscada

```

**class Droide :**

**private Pieza getPiezaOperativa ( String nombrePieza )**

**int i = 0   Pieza piezaEncontrada = null   Pieza pieza**

```

i < piezasOperativas.size() && piezaEncontrada == null

```

```

pieza = piezasOperativas.get(i)

```

```

pieza.mismaPieza(nombrePieza)

```

V

F

Villanueva Carlos Alberto

FPR-2021-2

2/12/2021 11:55:16

Villanueva Carlos Alberto

FPR-2021-2

2/12/2021 11:55:16

	/* remove() devuelve el objeto que estoy eliminando, por eso se puede asignar con la eliminacion */	i++
	/* piezaEncontrada = piezasOperativas.remove(i) */	
	/* OTRA FORMA */	
	piezaEncontrada = piezasOperativas.get(i)	
	piezasOperativas.remove(i)	
return piezaEncontrada		

**class Droide :**

**public void reemplazarPieza ( Pieza piezaDaniada , Pieza piezaOperativa )**

piezasNoOperativas.remove(piezaDaniada)

piezasOperativas.add(piezaOperativa)

Villanueva Carlos Alberto

FPR-2021-2

2/12/2021 11:55:16

**class Pieza :**

**private boolean mismaPieza ( String nombre )**

return this.nombre.equals(nombre)

Villanueva Carlos Alberto

FPR-2021-2

2/12/2021 11:55:16

**class Droide :**

**public void obtenerResultadoReparacion ( boolean encuentreUnaPieza )**

ResultadoReparacion resultado = REPARACION\_IMPOSIBLE

/\* piezasNoOperativas.size() == 0 \*/

piezasNoOperativas.isEmpty()

resultado = ResultadoReparacion.COMPLETAMENTE\_OPERATIVO

encuentreUnaPieza

```
resultado = ResultadoReparacion.REPARACION.PARCIAL
```

```
return resultado
```