

Ejercicio 3

Lista()

vacia()

insertar(int dato)

getNumElementos()

getElemento(int posicion)

borrar(int dato)

Iterador getIterador()

posicion(int dato)

contiene(int dato)

Funciones()

```
public Lista sinRepeticiones(Lista lista) {
```

```
    int dato;
```

```
    for (int i = 0; i < lista.getNumElementos(); i++) {
```

```
        dato = lista.getElemento(i);
```

```
        int i = 1;
```

```
        while (lista.hasNext()) {
```

```
            if (dato == lista.getElemento(i))
```

```
                lista.borrar(lista.getElemento(i));
```

```
        }
```

```
    }
```

```
}
```

Iterador()

hasNext()

next()

elem

8 - 2 = 6

1

0

1

2

3

4

```
while (lista.hasNext()) {  
    int i = 1;  
    if (dato == lista.getElemento(i))  
        i++;  
}
```

Ejercicio 4 lista Enlazada

Lista

Nodo inicio

Lista.L()

Sin estructuras auxiliares
Un solo recorrido en la lista

Nodo

Nodo(int dato, Nodo sig)

Nodo getSiguiente()

Nodo setSiguiente()

int getDato()

void setDato(int dato)

2 4 3

3 4 2 2 4 3

clase lista

→ public void imprimir() {

Nodo actual = inicio;

while (actual != null) {

Nodo n = new Nodo(0, inicio);

n.setDato(actual.getDato());

inicio = n;

actual = actual.getSiguiente();

}

}

for each

2 → inicio ⇒ 2 2 n 3
↑ ↑
inicio fin

3 → inicio ⇒ 3 2 2 n 3
↑ ↑
inicio fin

Ejercicio 5

Lista

Nodo inicio

Lista ()

Nodo

Nodo (int dato, Nodo siguiente)

Nodo getSiguiente()

setSiguiente(Nodo siguiente)

int getDato()

setDato(int dato)

```
public void insertarCeros() {
```

```
    int suma = 0;
```

```
    Nodo actual = inicio;
```

```
    while (actual != null) {
```

```
        suma = suma + actual.getDato();
```

```
    } actual = actual.getSiguiente();
```

```
    Nodo actual = inicio;
```

```
    while (actual != null) {
```

```
        if (suma == actual.getDato()) {
```

```
            Nodo n = new Nodo(0, actual.getSiguiente());
```

```
            actual.setSiguiente(n);
```

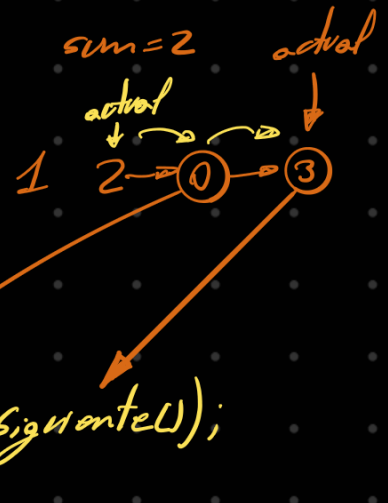
```
        } actual = actual.getSiguiente();
```

```
    }
```

```
    actual = actual.getSiguiente();
```

```
}
```

$$\begin{array}{r} 1 \ 2 \ 0 \ 3 \ 2 \ 0 \ - \ 6 \\ \hline \Sigma \neq 0 \end{array}$$



Doble salto

