

Estructuras de datos lineales: Listas.

El problema con los arreglos es que cuando se crean se les da un tamaño fijo, si se desea agregar un campo nuevo se tendrá que crear un nuevo arreglo con el nuevo tamaño.

Con las listas se da un crecimiento dinámico.

Lista enlazada simple:

Node nodo; → null

```
public class lista{
```

```
    private Nodo inicio;
```

```
    public void lista( ){ }
```

```
}
```

```
Lista lista = new Lista( );
```

```
Nodo pimero = new Nodo("Ant");
```

```
lista inicio = primero;
```

```
System.out.println(lista.inicio.nombre); // IMPRIME Ant
```

```
if (lista.inicio == null);
```

```
if (lista.inicio.siguiente == null);
```

```
public class Nodo{
```

```
    String nombre;
```

```
    Node siguiente;
```

```
    public void Nodo (String nombre){ //CONSTRUCTOR
```

```
        this.nombre = nombre;
```

```
    }
```

```
}
```

```
Nodo p = new Nodo("Bat");  
lista.inicio.siguiente = p;  
Nodo p = new Nodo("Cat");  
lista.inicio.siguiente = p;
```

Eliminar un nodo

```
//ELIMINAMOS EL NODO ANT  
_bat = lista.inicio.siguiente;  
lista.inicio = null;  
lista.inicio = _bat;
```

Agregamos el nodo Pig al inicio de la lista

```
p = new Nodo("Pig");  
lista.inicio = p;
```

```
p.siguiente = _bat;
```

```
lista.inicio.siguiente = _bat;
```



Hacen lo mismo

Agregamos el nodo Dog entre el nodo Pig y el nodo Bat

```
_pig = lista.inicio;
```

```
_bat = pig.siguiente;
```

```
_bat = lista.inicio.siguiente;
```



Hacen lo mismo

```
p = new Nodo("Dog");
```

```
_pig.siguiente = p;
```

```
p.siguiente = _bat;
```

Para imprimir todos los elementos

```
p = lista.inicio;

While (p.siguiente != null){

System.out.println(p.nombre);

p = p.siguiente.

}
```

Para eliminar el ultimo nodo

```
p = lista.inicio;

While (p.siguiente.siguiente != null) p=p.siguiente;

p.siguiente = null;
```

Para insertar en el centro con While (Nodo Cat)

```
p = lista.inicio;

While (p.nombre!= "Dog"){

    p = p.siguiente;

}

Nodo nodo = new Nodo("Cat");

_bat = p.siguiente;

p.siguiente = nodo;

nodo.siguiente = _bat;
```

Lista Doblemente enlazada:

```
public class Nodo{

    Object elemento;

    Nodo siguiente;

    Nodo anterior;

}
```

```
public class Listad{  
    Nodo inicio;  
    Nodo final;  
}
```