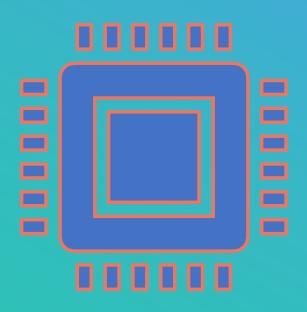


VISUALIZACIÓN FUNCIONAMIENTO DE CÓDIGO JAVA



CÓMO FUNCIONA UN VECTOR (VERSIÓN JAVA 8.0)

VISUALIZE CODE EXECUTION

Learn Python, Java, C, C++, JavaScript, and Ruby

[consulta online 14/9/2020]

http://www.pythontutor.com/java.html#mode=edit

```
Write code in Java 8 V
   1 public class ListaEstatica {
          private int [] v;
          private int cant;
          public void inicializarLista() {
              v = new int[10];
              cant = 0;
          public boolean listaVacia() {
  10
  11
              return cant == 0;
  12
  13
  14
          public void valores() {
              int actual = 0;
  15
              while (actual < cant) {
  16
                  System.out.print(v[actual] + " - ");
  17
  18
                  actual = actual + 1;
  19
  20
  21
  22
          public boolean existe(int x) {
              int actual = 0;
  23
              while (actual < cant && v[actual] !=x )
  24
                  actual++;
  25
              return actual != cant;
  26
  27
  28
```

CONTINUACIÓN

```
public void agregar(int x) {
30
           if (cant == 0) {
31
               v [0] = x;
32
               cant ++;
           } else {
               int actual = cant - 1;
35
               while (actual >= 0 \&\& x <= v[actual])
                  v[actual + 1] = v[actual];
37
                  actual --;
38
39
               v[actual + 1] = x;
40
               cant++;
41
42
43
       public void eliminarInicio() {
45
           if (cant != 0) {
46
               cant--;
               for (int i = 0; i< cant; i++)
47
48
                  v[i] = v[i+1];
49
50
51
52
       public void eliminarFinal() {
53
           if (cant != 0) {
54
               cant--;
55
56
57
       public void eliminar(int x) {
58
           if (cant != 0) {
59
60
               int i = 0;
61
               while (i < cant && v[i] != x)
                  i++;
                cant --;
                for (; i< cant; i++)
                   v[i] = v[i+1];
66
67
                               68
                               69
                                        public static void main(String[] args) {
                                             ListaEstatica l = new ListaEstatica();
                               70
                                            l.inicializarLista();
                               71
                                            1.agregar(3);
                               72
                               73
                                            1.agregar(1);
                                            1.agregar(2);
                               74
                                            1.agregar(4);
                               75
                                            1.valores();
                               76
                                             System.out.println();
                               77
                               78
                                            1.valores();
                               79
                               80
```

Visualize Execution

```
Print output (drag lower right corner to resize)
                               Java 8
                          (known limitations)
                   if (cant != 0) {
       59
       60
                       int i = 0;
       61
                        while (i < cant && v[i] != x)
                           i++;
                                                                                           Objects
                                                                             Frames
       63
                        cant --:
                                                                       main:79
                                                                                            ListaEstatica instance
                        for (; i< cant; i++)
       64
       65
                            v[i] = v[i+1];
       66
                                                                       Return
                                                                                             cant | 4
                                                                              void
                                                                        value
       67
       68
       69
               public static void main(String[] args) {
       70
                   ListaEstatica 1 = new ListaEstatica();
                   1.inicializarLista();
       72
                   1.agregar(3);
                   1.agregar(1);
       73
       74
                   1.agregar(2);
                   1.agregar(4);
                  1.valores();
                  System.out.println();
                  1.valores();
                                                CONTINUACIÓN
       80 }
                            Edit this code
ine that just executed
next line to execute
                           < Prev
```

0

CÓMO FUNCIONA UNA LISTA ENLAZADA (VERSIÓN JAVA 8.0)

VISUALIZE CODE EXECUTION
Learn Python, Java, C, C++, JavaScript, and Ruby
[consulta online 14/9/2020]
http://www.pythontutor.com/java.html#mode=edit

```
Write code in Java 8 V
     public class ListaEjemplo {
          public class Nodo{
              int x;
              Nodo sig:
          private Nodo inicio;
   8
          public void inicializarLista() {
            inicio = null;
  10
  11
  12
          public boolean listaVacia() {
  13
            return inicio == null;
  14
  15
  16
          public void valores() {
  17
              Nodo actual = inicio;
              // recorre toda la lista
  18
  19
              while (actual != null) {
  20
                  System.out.print(actual.x + " - ");
  21
                  actual = actual.sig;
  22
  23
  24
  25
          public boolean existe(int x) {
              Nodo actual = inicio;
  26
              while (actual != null && actual.x !=x )
  27
  28
                  actual = actual.sig;
              return actual != null;
  29
  30
```

CONTINUACIÓN

31

```
32
        public void agregar(int x) {
33
          Nodo n = new Nodo();
         n.x = x;
34
35
          if (inicio == null || x < inicio.x) {</pre>
36
             n.sig = inicio;
37
             inicio = n;
38
          } else {
39
             Nodo actual = inicio, ant = null;
40
             while (actual != null \&\& x >= actual.x) {
41
                 ant = actual;
                 actual = actual.sig;
42
43
             n.sig = actual;
45
             ant.sig = n;
46
47
48
49
        public void eliminarInicio() {
          if (inicio != null)
50
51
          inicio = inicio.sig;
52
53
       public void eliminarFinal() {
55
           if (inicio != null) {
56
                if (inicio.sig ==null)
57
                   eliminarInicio();
58
                else {
59
                   Nodo actual = inicio, ant = null;
                                                       69
                                                               public void eliminar(int x) {
60
                   while (actual.sig != null) {
                                                       70
                                                                   if (inicio != null) {
61
                       ant = actual;
                                                       71
                                                                        if (inicio.x == x)
62
                       actual = actual.sig;
63
                                                       72
                                                                            eliminarInicio();
64
                   ant.sig = actual.sig;
                                                       73
                                                                        else {
65
                                                       74
                                                                            Nodo actual = inicio, ant = null;
66
                                                       75
                                                                            while (actual != null && actual.x !=x ) {
67
                                                       76
                                                                                ant = actual;
                                                       77
                                                                                actual = actual.sig;
                                                       78
                                                       79
                                                                            if (actual != null)
                                                       80
                                                                                ant.sig = actual.sig;
                                                       81
                                                       82
                                                       83
                                                       84
                                                       85
                                                               public static void main(String[] args) {
                                                       86
                                                                   ListaEjemplo l = new ListaEjemplo();
                                                       87
                                                                   l.inicializarLista();
                                                       88
                                                                   1.agregar(3);
                                                       89
                                                                   1.agregar(1);
                                                       90
                                                                   1.agregar(2);
                                                       91
                                                                   1.agregar(4);
                                                       92
                                                                   1.valores();
                                                       93
                                                                   System.out.println();
                                                       94
                                                                   1.valores();
                                                       95
                                                       96
                                                       97
```

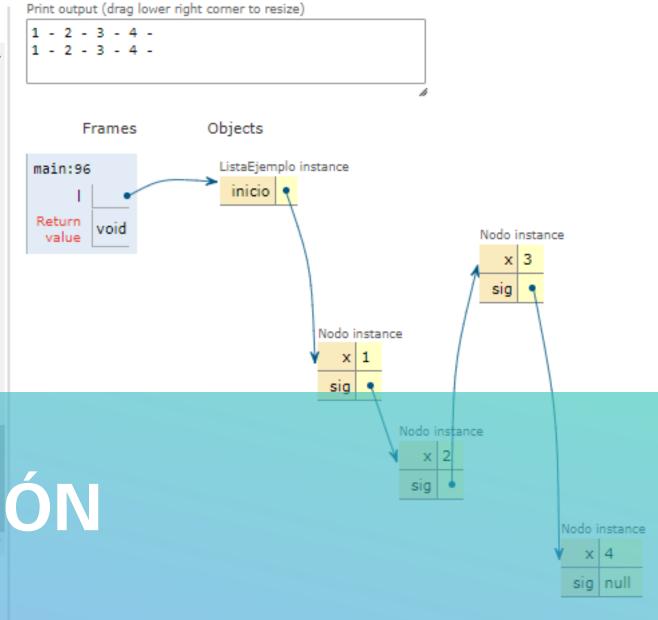
Visualize Execution

```
Java 8
                     (known limitations)
76
                       ant = actual;
77
                       actual = actual.sig;
78
79
                   if (actual != null)
                       ant.sig = actual.sig;
80
81
82
83
84
85
       public static void main(String[] args) {
86
           ListaEjemplo 1 = new ListaEjemplo();
87
           1.inicializarLista();
88
           1.agregar(3);
           1.agregar(1);
89
90
           1.agregar(2);
           1.agregar(4);
91
           1.valores();
           System.out.println();
           1.valores();
                  CONTINUACIÓN
97 O<sub>}</sub>
```

Edit this code

<< First | < Prev | Next > | Last >> |

line that just executednext line to execute



TRABAJO PRÁCTICO OBLIGATORIO



Para realizar una versión preliminar

- ✓ La prueba puede realizarse con la implementación estática
- ✓ No se requiere usar archivos para el ingreso de datos a la cola. Esta puede cargarse con valores constantes.

Sólo para una prueba final se requiere cargar los archivos y utilizar la implementación adecuada para la información recibida en los archivos.

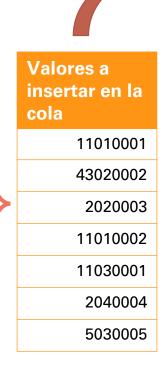
Por ser un volumen de solicitudes importantes el programa sólo deberá recorrer la cola de solicitudes *una vez*.

Lista de las 5 solicitudes más recientes.

Ejemplo. Se tiene la siguiente información. De las solicitudes menos a las más recientes

Id Persona	Proveedor	Película
11	YouTube	Batman
43	Netflix	Robocop
2	Netflix	Misión imposible
11	YouTube	Robocop
11	Vimeo	Batman
2	НВО	Duro de matar
5	Vimeo	Nueve reinas

Codigo	proveedor	Codigo	película
1	youtube	1	Batman
2	netflix	2	Robocop
3	Vimeo	3	Misión imposible
4	НВО	4	Duro de matar
		5	Nueve reinas



Valores dentro de la estructura final
5030005
2040004
11030001
11010002
2020003

ld	Prov	Película
5	3	5
2	4	4
11	3	1
11	1	2
2	2	3

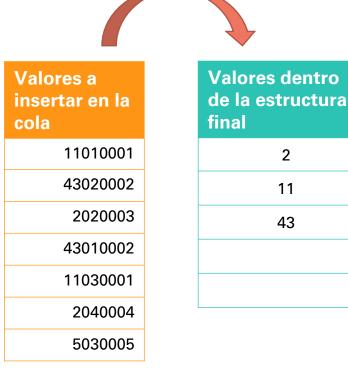
Listado de personas que solicitaron la mayor cantidad de películas, ordenadas por identificador.

Ejemplo. Se tiene la siguiente información. De las solicitudes menos a las más recientes

	ld	Persona	Pr	oveedor	Película	
	11		Yo	uTube	Batman	
	43		Ne	etflix	Robocop	
	2		Ne	etflix	Misión imposible	
	43		Yo	uTube	Robocop	
	11		Vir	meo	Batman	
	2		HE	80	Duro de matar	
	5		Vir	meo	Nueve reinas	
Codiç	go	proveed	or	Codigo	película	
1		youtube		1	Batman	
2		netflix		2	Robocop	
3		Vimeo		3	Misión imposible	
4		НВО		4	Duro de matar	/

5

Nueve reinas



Lista de películas que fueron más solicitadas con la empresa a la que fue solicitada.

Ejemplo. Se tiene la siguiente información. De las solicitudes menos a las más recientes

	ld	Persona	Pr	oveedor	Película
	11		Yo	uTube	Batman
	43		Ne	etflix	Robocop
	2		Ne	etflix	Misión imposible
	43		Yo	uTube	Robocop
	11		Vii	meo	Batman
	2		HE	80	Duro de matar
	5		Vii	meo	Nueve reinas
Codiç	go	proveed	or	Codigo	película
1		youtube		1	Batman
2		netflix		2	Robocop
3		Vimeo		3	Misión imposible
4		НВО		4	Duro de matar
				5	Nueve reinas



•	
Valores of de la esti	
1	1, 3
2	1, 2

Ranking de las 10 películas más solicitadas, ordenadas por cantidad de veces que fueron solicitadas

Ejemplo. Se tiene la siguiente información. De las solicitudes menos a las más recientes. Para el ejemplo puede haber varias soluciones posibles.

ld	Persona	Pr	oveedor	Película	
11		Yo	uTube	Batman	
43		Ne	tflix	Robocop	
2		Ne	tflix	Misión imposible	
43		Yo	uTube	Robocop	
11		Vir	neo	Batman	
2		НВ	0	Duro de matar	
5		Vir	neo	Nueve reinas	
Codigo	proveed	or	Codigo	película	
ſ	youtube		1	Batman	
2	netflix		2	Robocop	
3	Vimeo		3	Misión imposible	
4	НВО		4	Duro de matar	
			5	Nueve reinas	

		•		
Valores a insertar en la cola	Valores of de la estre final (alte	ructura	Valores of de la estifinal. (alt 2)	ructura
11010001	2	2	2	2
43020002		_		_
2020003	1	2	1	2
43010002	5	1	4	1
11030001	4	1	5	1
2040004	3	1	3	1
5030005				
			d que fueron s igual, pued	

variar su orden

Lista de personas que solicitaron películas iguales a distintas compañías, indicando película y compañía.

lores dentro de la

1 (1, 3)

2 (2, 1)

ructura final

Ejemplo. Se tiene la siguiente información. De las solicitudes menos a las más recientes.

	ld	Persona	Pr	oveedor	Película
	11		Yo	uTube	Batman
	43		Ne	etflix	Robocop
	2		Ne	etflix	Misión imposible
	43		Yo	uTube	Robocop
	11		Vii	meo	Batman
	2		HE	30	Duro de matar
	5		Vii	meo	Nueve reinas
Codi	go	proveed	or	Codigo	película
1		youtube		1	Batman
2		netflix		2	Robocop
3		Vimeo		3	Misión imposible
4		НВО		4	Duro de matar
				5	Nueve reinas

TRABAJO PRÁCTICO OBLIGATORIO

+

0



Lectura de archivos de texto en lenguaje JAVA.

Existen varias formas. Se detalla una de las formas en el siguiente ejemplo.

ARCHIVO DE TEXTO

```
141; EL SEXTO SENTIDO; Kanopy
500; AVATAR VERSION EXTENDIDA; Amazon Prime Video
778; MISION SECRETA; YouTube
264; EL QUINTO INFIERNO; Cine.Ar
881; QUIEN QUIERE SER MILLONARIO; Tubi
591; POMPEYA; Netflix
427; BLOOD - EL ULTIMO VAMPIRO; CBS All Access
304; MORGANA; YouTube
816; HECTOR Y EL SECRETO DE LA FELICIDAD; Vimeo
147; ESCAPE; Open Culture
838; PALABRAS ROBADAS; Internet Archive
567; PEQUEYOS TRAVIESOS AL RESCATE; Amazon Prime Video
994; ASESINOS DE ELITE; Roku Channel
822; HONDO (JOHN WAINE); IMDbTV
264; EL HOBBIT LA DESOLACION DE SMOUG; AppleTV +
453; ENTRE EL ODIO Y EL AMOR; Amazon Prime Video
744; PELIGROSA COMPAYÖA; YouTube
337; BLADE TRINITY; Vimeo
841; SACRIFICE; FandangoNow
142; EL HOBBIT UN VIAJE INESPERADO; Netflix
690; EL OCASO DE UN ASESINO; Hulu
193; EL OJO DEL MAL 2; Vimeo
123; NAUFRAGO; Roku Channel
856; CAMBIO DE PLANES; Kanopy
```

Código de lectura

```
public void cargarMovimientos(String archivo) {
    try {
        FileReader arch = new FileReader(archivo);
        BufferedReader buffer = new BufferedReader(arch);
         int dni, valor;
         String línea, nombreProveedor, nombrePelicula;
         buffer.readLine();
         linea = buffer.readLine();
         while(linea != null) {
              String[] lista = linea.split(";");
              dni = Integer.valueOf(lista[0]);
              nombreProveedor = lista[2];
              nombrePelicula = lista[1];
              linea = buffer.readLine();
         arch.close();
      catch (Exception e) {
        System.out.println("Error de lectura. Arch: "+ archivo + ": " + e);
```