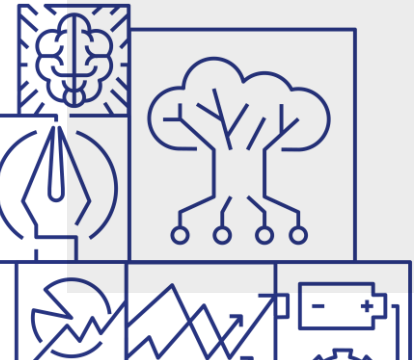




Filminas de la semana 4

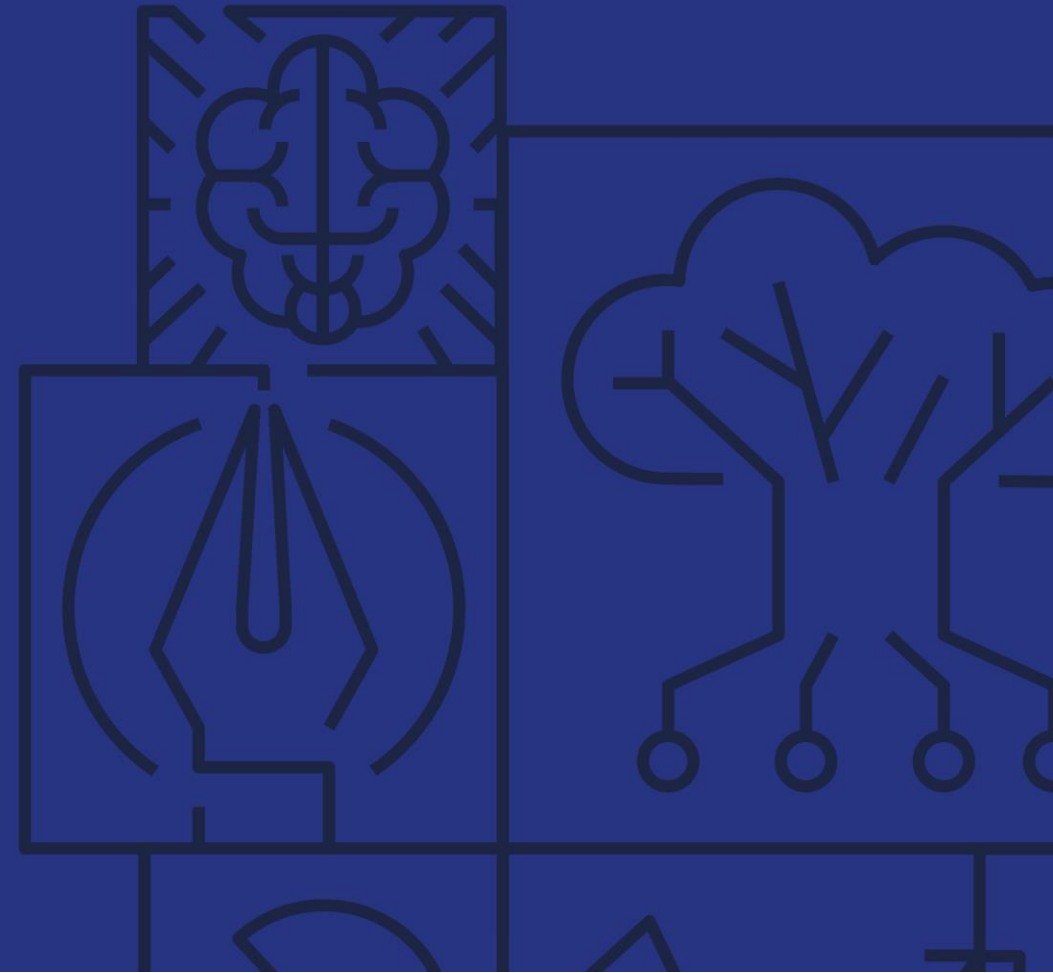
Agenda

- Revisión de la tarea sobre Pilas
- Presentación de la estructura Cola mediante ejemplos de la cotidianidad
- Acercamiento a la estructura de referencias dinámicas en Java
- Revisión del atributo frente y último, así como los métodos encola y atiende
- Discusión de la práctica “Utilizando colas en un programa útil”



fidÉlitas
U n i v e r s i d a d

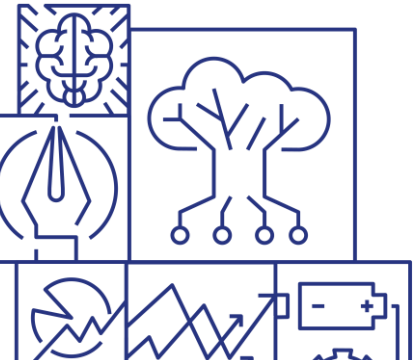
Colas



Concepto de cola



Una cola es una estructura de datos de acceso restrictivo, en la cual el primer elemento ingreso es el primero en ser extraído de la estructura.



Ejemplos de colas en el mundo cotidiano:

- Una fila en un Banco
- Una fila en la parada del bus
- La fila del supermercado
- Elementos que hacen “fila”



último

frente



Características de una cola

- Sólo se pueden **colocar** elemento en parte anterior al último elemento (**encola**).
- Sólo se puede **tomar** el elemento que está al frente de la cola (**atiende**)

Implementando la clase NodoC en Java

Un nodo es un elemento de una estructura de datos. Así entonces, primero definiremos una clase NodoC que contendrá la información importante para la Cola.

Se tienen 2 atributos: dato, de tipo String; atrás, de tipo NodoC.

```
public class NodoC {  
    private String dato;  
    private NodoC atrás;  
  
    .  
    .  
}
```

Implementando la clase Cola en Java

- La clase **Cola** sólo tiene dos métodos:
 - **encola**: que coloca un elemento en la Cola en la parte **de atrás**.
 - **atiende**: que toma elementos **del frente** de la Cola

```
public class Cola {  
    ..  
    public void encola(NodoC n) {  
        ..  
    }  
    public NodoC atiende() {  
        ..  
    }  
}
```


Veamos cómo se crea la clase NodoC en Java

```
1 package lec04;
2 public class NodoC {
3     private String dato; // Donde se almacena la información
4     private NodoC atras; // Referencia al objeto que está "atras" del actual
5
6     public String getDato() { return dato; } // devuelve el dato
7     public void setDato(String dato) { this.dato = dato; } //fija el dato
8
9     public NodoC getAtras() { return atras; } //devuelve la referencia al objeto que está atras
10    public void setAtras(NodoC atras) { this.atras = atras; } //Fija la referencia del objeto de atras
11
12    public NodoC(String dato) { this.dato = dato; } //constructor con el dato
13    public NodoC() { } //constructor default
14
15    @Override
16    public String toString() { return dato; } //retorna la info en String
17 }
```

Y ahora hacemos la clase Cola

```
1 package lec04;
2 public class Cola {
3     private NodoC frente; //Referencia al primer objeto de la Cola
4     private NodoC ultimo; //Referencia al último objeto de la Cola
5
6     public void encola(NodoC d) { //Coloca un elemento en la Cola
7         if (frente==null) { //Si la cola está vacía el elemento
8             frente=d; //se asigna tanto al frente como al
9             ultimo=d; //ultimo, porque son los mismos
10        } else { //en cambio, si la Cola tiene elementos
11            ultimo.setAtras(d); //Al ultimo le enlace el nuevo...
12            ultimo=d; //y actualizo el atributo ultimo
13        }
14    }
15
16    public NodoC atiende() { //Atiende el primer elemento de la Cola
```

Finalicemos la clase Cola

```
16  public NodeC atiende() { //Atiende el primer elemento de la Cola
17      NodeC aux=frente;    //aux referencia al primero de la Cola
18      if (frente!=null) { //Si la Cola tiene elementos...
19          frente=frente.getAtras(); //actualizo el primero como el anterior
20          aux.setAtras(null); // le elimino la referencia al que atiendo
21      }
22      return aux; //Retorno el auxiliar que tomó la referencia del primero
23  }

24  public String toString() { //Inspecciono la cola (internamente) sin eliminar
25      String s=""; // Defino s como un String vacío
26      NodeC aux=frente; // pongo en un auxiliar (aux) el frente actual
27      while (aux!=null) { // mientras aux no se nulo se hace el ciclo
28          s+=aux+"\n"; //Concateno aux a s (se llama al toString de aux)
29          aux=aux.getAtras(); //actualizo aux como el de atras...
30      }
31      return s; // retorno s
32  }
33  }
34  }
```


Finalmente, hagamos la clase principal

```
1 package lec04;
2 public class Lec04 {
3     public static void main(String args[]) {
4         Cola laCola=new Cola();
5         laCola.enqueue(new NodoC("Juan"));
6         laCola.enqueue(new NodoC("Pedro"));
7         laCola.enqueue(new NodoC("Sebastian"));
8         laCola.enqueue(new NodoC("Daniela"));
9         laCola.enqueue(new NodoC("Andres"));
10        System.out.println("La cola original");
11        System.out.print(laCola);
12        System.out.println("Atiendo 2");
13        NodoC nodo=laCola.atiende();
14        System.out.println(nodo);
15        System.out.println(laCola.atiende());
16        System.out.println("Encolo uno más");
17        laCola.enqueue(new NodoC("Andrea"));
18        System.out.print(laCola);
19    }
20 }
```

La salida que muestra el programa

```
La cola original
Juan
Pedro
Sebastian
Daniela
Andres
Atiendo 2
Juan
Pedro
Encolo uno más
Sebastian
Daniela
Andres
Andrea
BUILD SUCCESSFUL (total time: 0 seconds)
```

Práctica

- Crearemos un proyecto **Lec04** donde se definirá:
 - La clase **Dato**
 - La clase **Cola**
 - La clase **Rutinas**
- E incorporaremos funcionalidad en la clase principal de Lec04 para ver el funcionamiento de una Cola.

Taller sobre colas

1. Trabaja en la clase Rutinas
2. Cree un atributo private Cola cola1, cree un método que inserte algunos elementos en cola1
3. Cree el método public boolean encuentra(int x), que recibe un número x como parámetro y utilizando sólo Colas:
 1. Retorne falso si no se encuentra o
 2. Retorne verdadero si se encuentra.La Pila original debe prevalecer con los mismos valores y orden al finalizar la ejecución del método.
4. Cree el método que “extrae(int x)” que extrae un número de una cola si se encuentra, la cola debe prevalecer igual (menos el número sacado).
5. Cree un método que ordene ascendentemente la cola de números de una cola dada. En este caso la cola cambia el orden de los elementos.

Para todos los métodos debe utilizar únicamente Colas como estructuras de apoyo.

**Hasta la próxima.
Nos vemos.**

Tema de la siguiente clase: I Examen Parcial.

