

Colas



Dr. Jaime Osorio Ubaldo

Introducción

- 1 La teoría de colas es el estudio matemático de las colas.

Introducción

- 1 La teoría de colas es el estudio matemático de las colas.
- 2 El matemático danés Agner Krarup Erlang, trabajador de la Copenhagen Telephone Exchange, publicó el primer artículo sobre la teoría de colas en 1909.

Introducción

- 1 La teoría de colas es el estudio matemático de las colas.
- 2 El matemático danés Agner Krarup Erlang, trabajador de la Copenhagen Telephone Exchange, publicó el primer artículo sobre la teoría de colas en 1909.
- 3 El problema que Erlang abordó fue el de tratar de diseñar una red de teléfonos: ¿cuántas redes troncales se necesitan para una determinada cantidad de llamadas entre habitantes de un pueblo y otro?

Introducción

- 1 La teoría de colas es el estudio matemático de las colas.
- 2 El matemático danés Agner Krarup Erlang, trabajador de la Copenhagen Telephone Exchange, publicó el primer artículo sobre la teoría de colas en 1909.
- 3 El problema que Erlang abordó fue el de tratar de diseñar una red de teléfonos: ¿cuántas redes troncales se necesitan para una determinada cantidad de llamadas entre habitantes de un pueblo y otro?
- 4 Esta teoría se puede aplicar a diversas situaciones como comercio, negocios, industrias, transportes, logísticas para el mejoramiento de sus procesos de atención al cliente.

Cola



Cola



Una cola es una colección ordenada de elementos a partir de la cual se puede eliminar elementos de un extremo (llamado frente de la cola) y se puede agregar elementos en el otro extremo (llamado final de la cola).

Cola



Una cola es una colección ordenada de elementos a partir de la cual se puede eliminar elementos de un extremo (llamado frente de la cola) y se puede agregar elementos en el otro extremo (llamado final de la cola).

A la cola se le conoce como estructuras de datos de tipo FIFO (First In First Out).

Cola



Una cola es una colección ordenada de elementos a partir de la cual se puede eliminar elementos de un extremo (llamado frente de la cola) y se puede agregar elementos en el otro extremo (llamado final de la cola).

A la cola se le conoce como estructuras de datos de tipo FIFO (First In First Out). **Ejemplos.**

- 1 Colas en los supermercados, bancos.

Cola



Una cola es una colección ordenada de elementos a partir de la cual se puede eliminar elementos de un extremo (llamado frente de la cola) y se puede agregar elementos en el otro extremo (llamado final de la cola).

A la cola se le conoce como estructuras de datos de tipo FIFO (First In First Out). **Ejemplos.**

- 1 Colas en los supermercados, bancos.
- 2 Cola de impresiones.

Cola



Una cola es una colección ordenada de elementos a partir de la cual se puede eliminar elementos de un extremo (llamado frente de la cola) y se puede agregar elementos en el otro extremo (llamado final de la cola).

A la cola se le conoce como estructuras de datos de tipo FIFO (First In First Out). **Ejemplos.**

- 1 Colas en los supermercados, bancos.
- 2 Cola de impresiones.
- 3 Cola de tareas a ejecutar en una computadora.

Es posible aplicar esta estructura de datos en la simulación del funcionamiento de los

- 1 Bancos.

Es posible aplicar esta estructura de datos en la simulación del funcionamiento de los

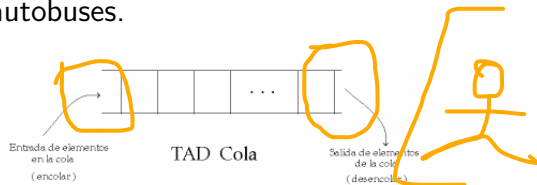
- 1 Bancos.
- 2 Líneas aéreas.

Es posible aplicar esta estructura de datos en la simulación del funcionamiento de los

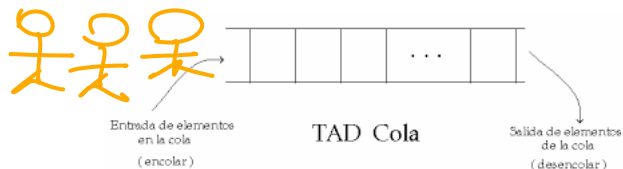
- 1 Bancos.
- 2 Líneas aéreas.
- 3 Casetas de cobro de autopistas.

Es posible aplicar esta estructura de datos en la simulación del funcionamiento de los

- 1 Bancos.
- 2 Líneas aéreas.
- 3 Casetas de cobro de autopistas.
- 4 Terminal de autobuses.



Operaciones



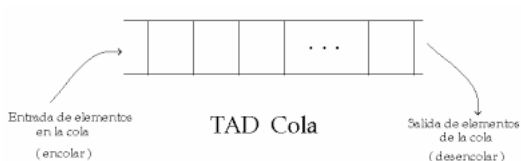
- Encolar . ✓

Operaciones



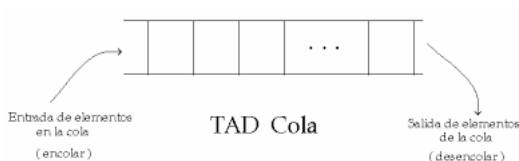
- Encolar .
- Desencolar.

Operaciones



- Encolar .
- Desencolar.
- Determinar el primer elemento de la cola.

Operaciones

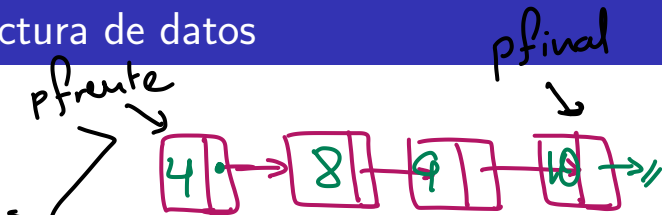


- Encolar .
- Desencolar.
- Determinar el primer elemento de la cola.
- Determinar el último elemento de la cola.

Definir la estructura de datos

```
struct nodocola{  
    int dato;  
    nodocola *sig;  
};
```

```
typedef nodocola *pnodocola;  
pnodocola pfrente;  
pnodocola pfinal;  
pfrente = NULL;  
pfinal = NULL;
```



pfrente → //

pfinal → //

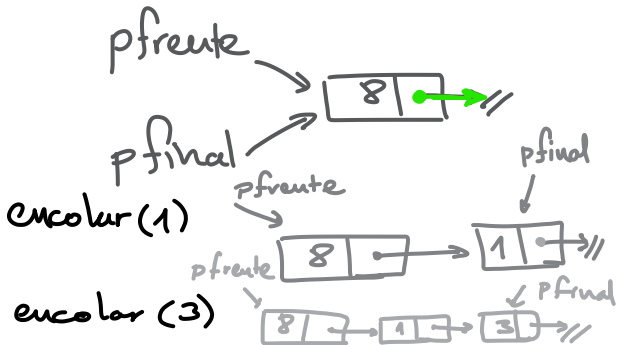
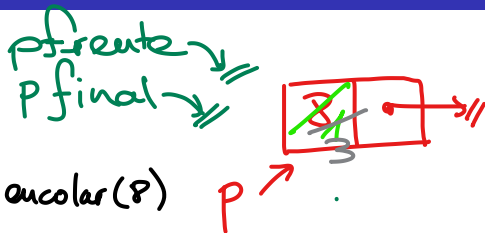
Obs :

```
int *p  
p = new int  
*p = 8 delete p
```

~~p → 8~~

Encolar

```
void encolar(int x){  
  pnodocola p; ✓  
  p = new nodocola; ✓  
  (*p).dato = x; ✓  
  (*p).sig = NULL; ✓  
  if (pfrente == NULL)  
    pfrente = p;  
  else  
    (*pfinal).sig = p; ✓  
  pfinal = p; ✓  
};
```



Desencolar

①

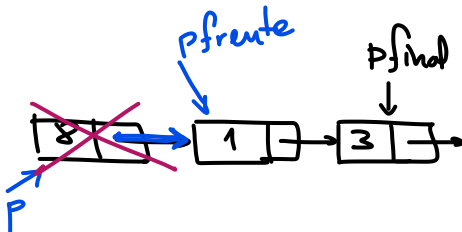
```
void desencolar(){  
    pnodocola p;  
    p = pfrente;  
    if ((*p).sig == NULL){  
        pfrente = NULL;  
        pfinal = NULL ;  
    }  
    else  
        pfrente = (*p).sig;  
    delete p;  
}
```

pfrente pfinal



desencolar()
pfrente pfinal

②



Problema

Para la atención en un banco se forman dos colas clasificadas como CLIENTES y VISITANTES. Escribir un programa que simule la atención de las personas, siguiendo las siguientes reglas:

- Las personas se ubican en las colas que les corresponde dependiendo si son clientes o visitantes.
- Si hay personas en las dos colas se atienden en el siguiente orden: dos de CLIENTES y luego uno de VISITANTES, luego se continua con esta secuencia.

Cola en un Banco

```
#include "banco.h"
int main(){
    cola colaVisitante, colaCliente;
    int operacion, cont=0;
    char tipoCliente, dato;
    do{ cout<< "Elija 1 para ingresar, 2 para atender y 0 para terminar";
        cin>> operacion;
        if(operacion==1){
            cout<< "Escriba c para cliente y escriba v para visitante" <<endl;
            cin>>tipoCliente;
            cin>>dato;
            if(tipoCliente=='c') colaCliente.encolar(dato);
            if(tipoCliente=='v') colaVisitante.encolar(dato);
        }else if(operacion==2){
```


Cola en un banco

```
if(cont < 2 && colaCliente.vacia() == false){  
    colaCliente.desencolar();  
    cont++; }  
else if(colavisitante.vacia() == false){  
    colaVisitante.desencolar();  
    cont=0; }  
else if(colavisitante.vacia() == true && colaCliente.vacia() == false){  
    cont=0;  
    colaCliente.desencolar(); }  
else  
    cout << "Las colas estan vacias" << endl;  
}  
colaCliente.imprimir();  
colaVisitante.imprimir();  
} while(operacion != 0);  
}
```

Handwritten notes and diagrams:

- A large red 'X' is next to the first if statement.
- An orange bracket groups the two middle if statements.
- Red arrows point from the left margin to the first, second, and third if statements.
- Handwritten text: *Clientes* with three boxes labeled $c=1$, $c=2$, and $c=0$, each with a red 'X' over it.
- Handwritten text: *Visitante* with two boxes, each with a red 'X' over it, and an arrow pointing to the right.
- Handwritten text: *cont = 0* next to the `cont=0;` lines.
- Handwritten text: *Clientes* and *Visitante* with red 'X' marks over the words.

Tarea

En el problema anterior, cada vez que se atiende un cliente debe imprimir su nombre y su tipo de cliente.