

Colas

Una **cola** es una estructura de datos que almacena elementos en una lista y permite acceder a los datos por uno de los dos extremos de la lista. Un elemento se inserta en la cola (parte final) de la lista y se suprime o elimina por la frente (parte inicial, Cabeza) de la lista. Las aplicaciones utilizan una cola para almacenar elementos en su orden de aparición o concurrencia.

Los elementos se eliminan (se quitan) de la cola en el mismo orden en que se almacenan y, por consiguiente, una cola es una estructura de **tipo FIFO** (first-in/first-out, primero en entrar/primeramente en salir).

Operaciones en una Cola

Las acciones que están permitidas en una cola son:

- Creación de una cola vacía.
- Verificación de que una cola está vacía.
- Añadir un dato al final de una cola.
- Eliminación de los datos de la cabeza de la cola.

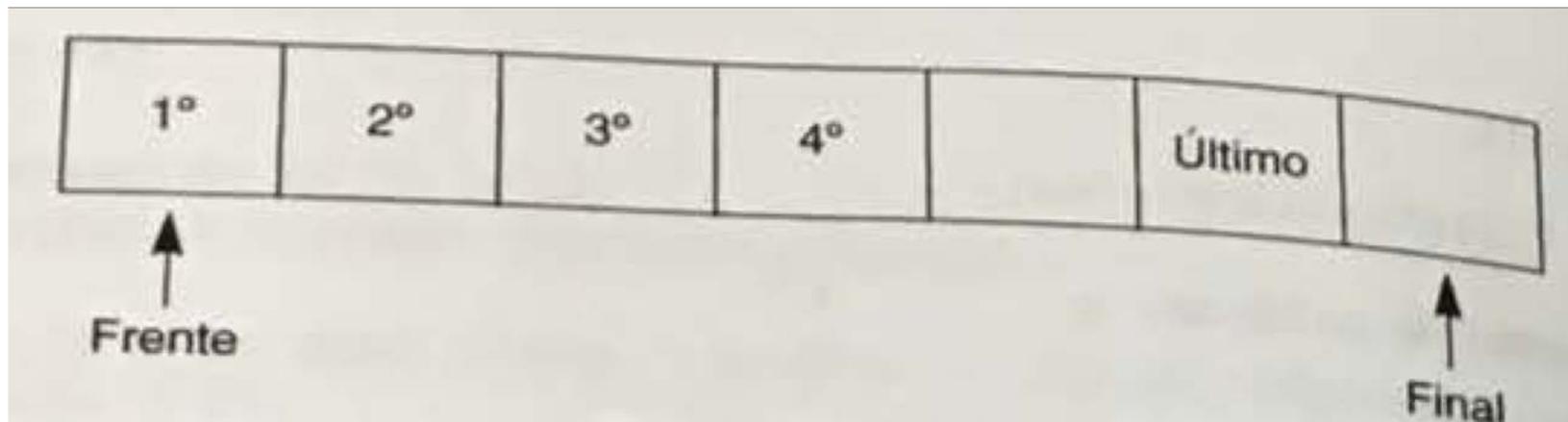


Figura: Una cola.

Los datos se almacenan de un modo lineal y el acceso a los datos sólo está permitido en los extremos de la cola.

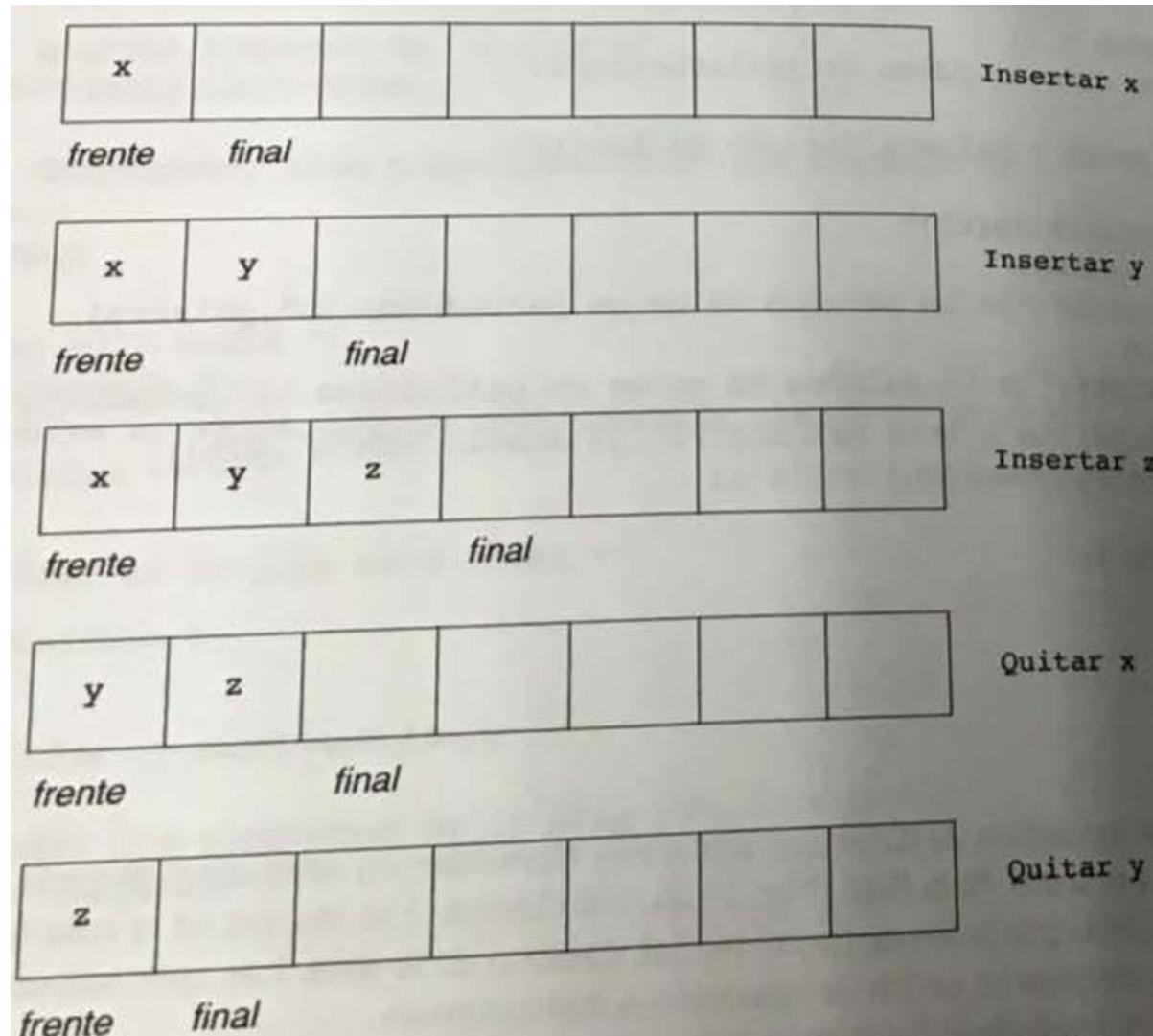


Figura: Operaciones de *Insertar* y *Quitar* en una Cola.

Especificaciones de una Cola

Las operaciones que sirven para definir una cola son:

- **TipoDato** **Dato que se almacena en la pila.**
- **InsertarQ** Toma un element del TipoDato y lo inserta en el final de la cola.
- **EliminarQ** Elimina (quita) y devuelve el element de la Cabeza o frente de la cola.
- **Qvacía** **Comprobar si la cola está vacía.**
- **Qllena** **Comprobar si la pila está llena de elementos.**
- **FrenteQ** Devuelve el valor del elemento en el frente de la cola, sin eliminar el elemento y, por tanto, no modifica la cola.

colaarray.h

El código para crear una cola requiere de la librería *colaarray.h* y se muestra a continuación:

```
#include<stdio.h>
#include<stdlib.h>
#define MaxTamQ 100

typedef struct
{
int Frente,Final;
TipoDato listaQ[MaxTamQ];
}
Cola;

void CrearCola(Cola* Q);

void MostrarCola(Cola* Q);

void InsertarQ(Cola* Q,TipoDato elemento);

TipoDato EliminarQ(Cola* Q);

void BorrarCola(Cola* Q);

TipoDato FrenteQ(Cola Q);

int Qvacia(Cola Q);

int LongitudQ(Cola Q);

int Qllena(Cola Q);
```

```
void CrearCola(Cola* Q)
{
Q->Frente=0;
Q->Final=0;
}
void MostrarCola(Cola* Q)
{
while(!Qvacía(*Q))
printf("%d",EliminarQ(Q));
}
void InsertarQ(Cola* Q,TipoDato elemento)
{
if(Qllena(*Q))
{
puts("imposible");
exit(1);
}
Q->Final=(Q->Final+1)%MaxTamQ;
Q->listaQ[Q->Final]=elemento;
}
TipoDato EliminarQ(Cola* Q)
{
TipoDato aux;
if(Qvacía(*Q))
{
puts("IMPOSIBLE");
exit (1);
}
```

```
Q->Frente=(Q->Frente+1)%MaxTamQ;
aux=Q->listaQ[Q->Frente];
return aux;
}
TipoDato Frente(Cola Q)
{
TipoDato aux;
{
if(Qvacia(Q))
puts("IMPOSIBLE");
exit(1);
}
return (Q.listaQ)[Q.Frente+1]%MaxTamQ;
}
int Qvacia(Cola Q)
{
return(Q.Frente==Q.Final);
}
int Qllena(Cola Q)
{
return (Q.Frente==(Q.Final+1)%MaxTamQ);
}
```

cola.c

Entregar el proyecto de cola.c