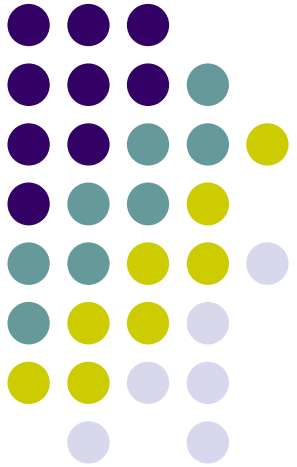
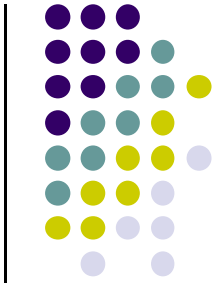
COLAS CIRCULARES

ESTRUCTURAS DE DATOS
2006



Prof. Ing. M.Sc. Fulbia Torres

COLAS CIRCULARES

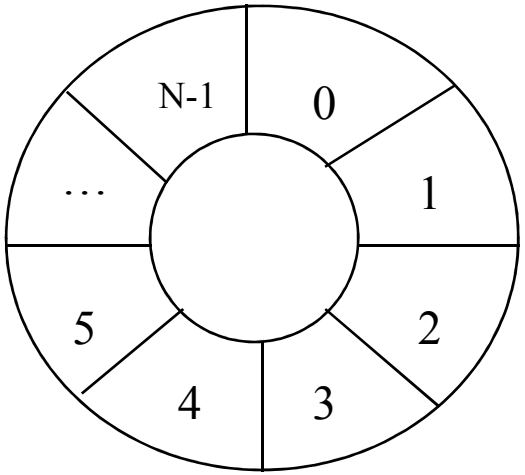


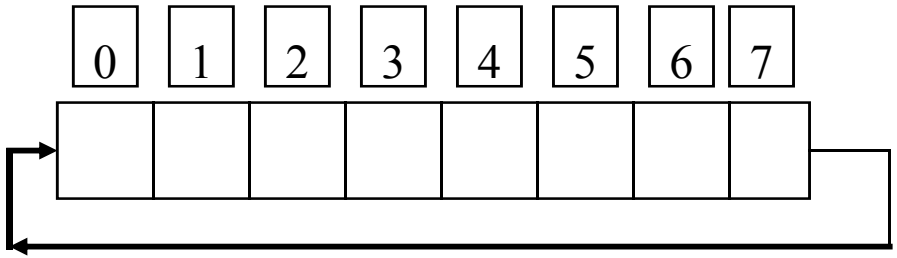
IMPLEMENTACIÓN COLA CIRCULAR (Utilizando Arreglos)

La solución que se describe implica reutilizar las componentes del vector que contenían elementos ya eliminados. Esto es, cuando durante el proceso de añadido lleguemos al final del vector, comenzaremos a llenar de nuevo las componentes iniciales del mismo si se encuentran vacías. Para lograr esto manejaremos el vector como si fuese un **"vector circular".** Esto significa que no consideraremos la componente MAX del vector como la última del mismo, sino que consideraremos que la siguiente componente a ésta es otra vez la primera del vector

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES





Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

DEFINICIÓN

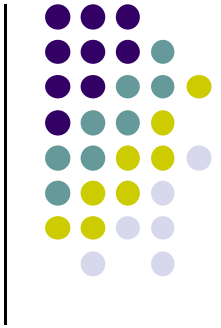
define MAX número máximo de elementos

```
struct {
          tipo_base  datos [MAX];
          int frente,final;
        } cola;

struct  cola  c;
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES



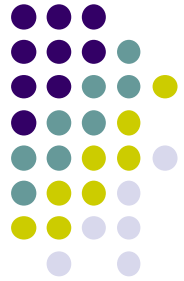
Con esta definición de **Cola** las operaciones asociadas especificadas en el TAD quedarían del siguiente modo:

Crea una cola vacía.

```
cola CrearColacirc (struct cola *c)
{
        (*c).frente = 0;
        (*c).final = MAX-1;
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

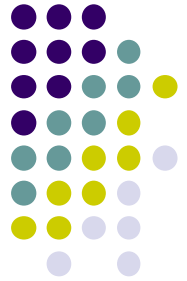


Dada una cola c y un valor, del tipo base, forma una nueva cola con el valor agregado por el *final*.

```
void InsertarColacirc (struct cola *c, tipo_base valor)
{
        if (!ColaLLenacirc (c)
        {
            c.final = siguiente(final);
             c.datos[c.final]=valor;
        }
        else
        {
            cout << “Cola Overflow”;
             exit(1);
        }
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES



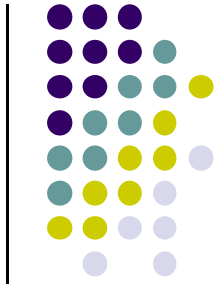
Dada una cola c, elimina el elemento indicado por el valor del *frente*.

```
tipo_base RemoverColacirc (struct cola *c)
{   tipo_base x;

        if (!ColaVaciacirc (c))
            {  x=(*c).datos [(*c).frente]);
               (*c).frente = siguiente((*c).frente );
                return x;
             }
        else
         {  cout << ”Cola Underflow \n";
              exit (1);
          }
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

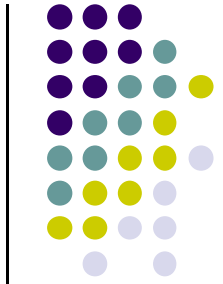


Devuelve el valor verdadero si la cola está vacía y falso en caso contrario.

```
int ColaVaciacirc (struct cola *c)
{
            if  ( (*c).frente == siguiente((*c).final))
               return  (1);
            else return ( 0);
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

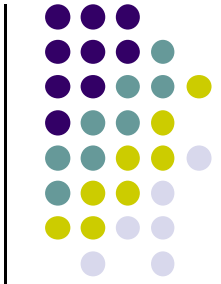


Devuelve el valor verdadero si la cola está vacía y falso en caso contrario.

```
int ColaLlenacirc (struct cola *c)
{
            if  ( (*c).frente == siguiente(siguiente((*c).final)))
               return  (1);
           else return ( 0);
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

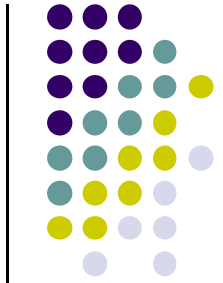


Devuelve el valor del elemento que está apuntado por frente.

```
tipo_base InfoColacirc (struct cola *c)
{
        if ColaVaciacirc (c) { cout << ”Cola Underflow \n";
                                exit (1);
                              }
       else return ((*c).datos[(*c).frente]);
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES



Teoría del resto.

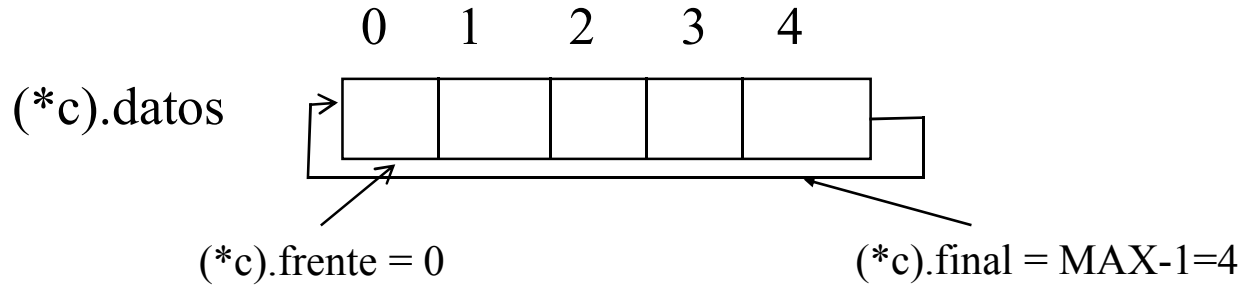
```
Int  siguiente(int r)
{
        return (r + 1) % MAX;
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

Corrida Implementación Circular de Colas.

CrearColacir (c):

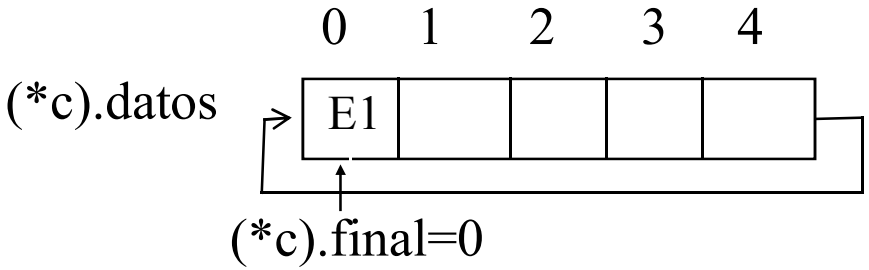


Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

Corrida Implementación Circular de Colas.

InsertarColacir (c,E1):



(*c).frente = 0

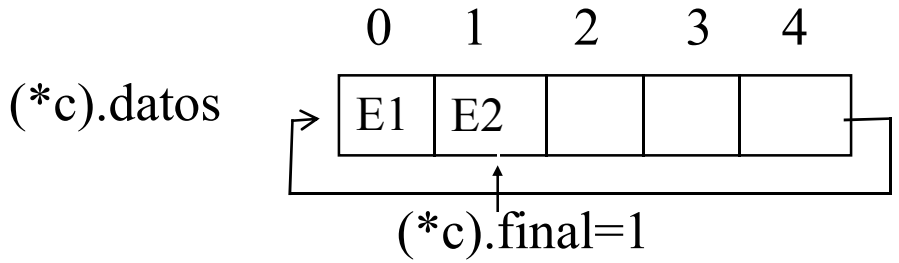
```
void InsertarColacirc (struct cola *c, tipo_base valor)
{
        if (!ColaLLenacirc (c)
        {
            c.final = siguiente(final);
            c.datos[c.final]=valor;
        }
        else
        {
            cout << "Cola Overflow";
            exit(1);
        }
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

Corrida Implementación Circular de Colas.

InsertarColacir (c,E2):



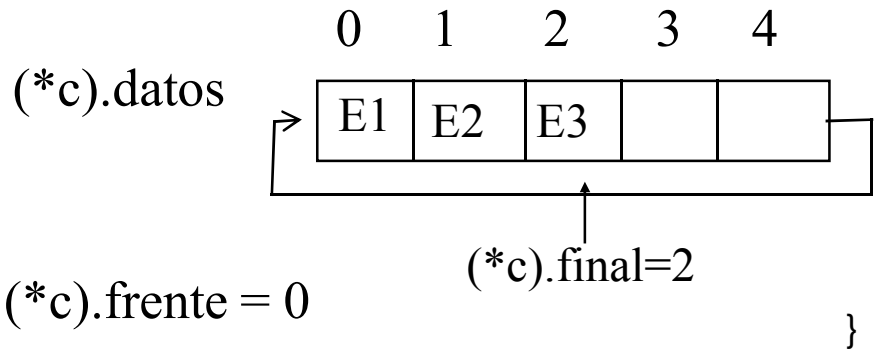

```
void InsertarColacirc (struct cola *c, tipo_base valor)
{
        if (!ColaLLenacirc (c)
        {
            c.final = siguiente(final);
            c.datos[c.final]=valor;
        }
        else
        {
            cout << "Cola Overflow";
            exit(1);
        }
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

Corrida Implementación Circular de Colas.

InsertarColacir (c,E3):




```
void InsertarColacirc (struct cola *c, tipo_base valor)
{
        if (!ColaLLenacirc (c)
        {
            c.final = siguiente(final);
            c.datos[c.final]=valor;
        }
        else
        {
            cout << "Cola Overflow";
            exit(1);
        }
}
```

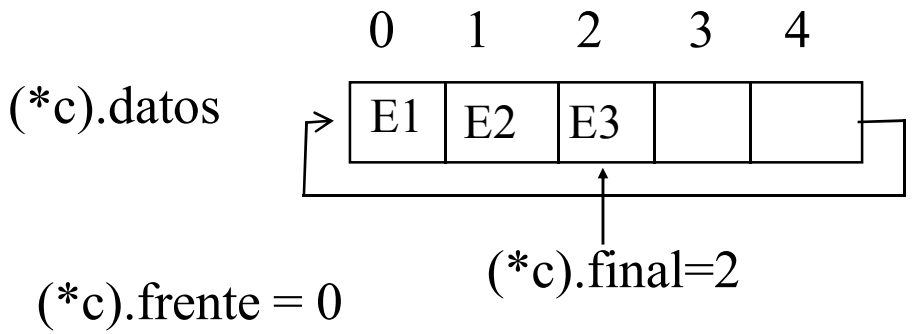
Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

Corrida Implementación Circular de Colas.

InfoColacir (c): (*c).datos[(*c).frente]

Mostrará E1 sin modificar frente

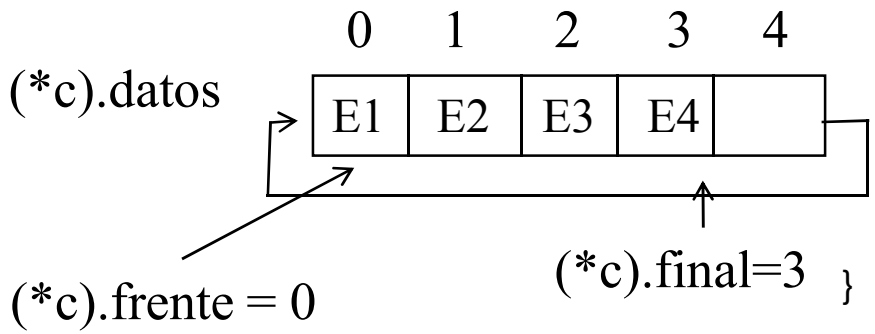


Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

Corrida Implementación Circular de Colas.

InsertarColacir (c,E4):



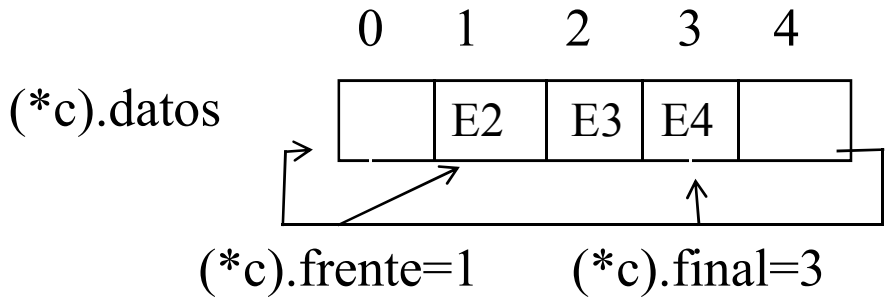

```
void InsertarColacirc (struct cola *c, tipo_base valor)
{
        if (!ColaLLenacirc (c)
        {
            c.final = siguiente(final);
            c.datos[c.final]=valor;
        }
        else
        {
            cout << "Cola Overflow";
            exit(1);
        }
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

Corrida Implementación Circular de Colas.

RemoverColacir (c):




```
tipo_base RemoverColacirc (struct cola *c)
{   tipo_base x;

      if (!ColaVaciacirc (c))
          {  x=(*c).datos [(*c).frente]);
             (*c).frente = siguiente((*c).frente );
             return x;
               }
      else
       {  cout << "Cola Underflow \n";
                exit (1);
        }
}
```

Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

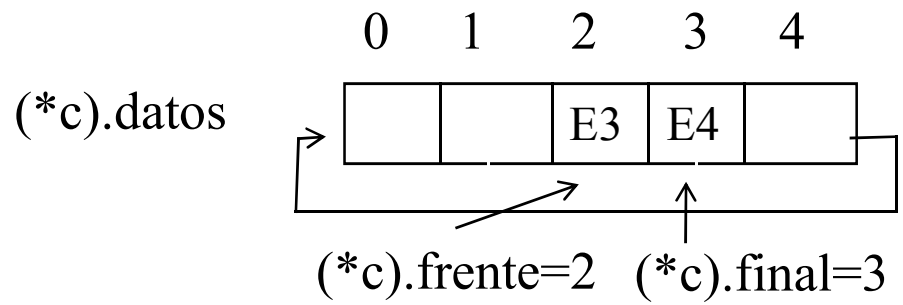
COLAS CIRCULARES

Corrida Implementación Circular de Colas.

RemoverColacir (c):

```
tipo_base RemoverColacirc (struct cola *c)
{   tipo_base x;

      if (!ColaVaciacirc (c))
          {  x=(*c).datos [(*c).frente]);
             (*c).frente = siguiente((*c).frente );
             return x;
               }
      else
       {  cout << "Cola Underflow \n";
              exit (1);
        }
}
```



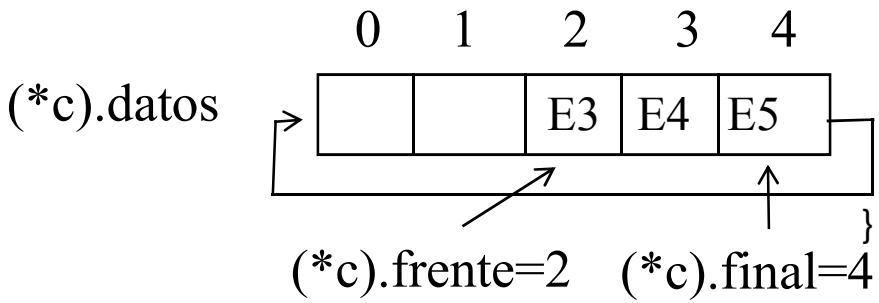
Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

Corrida Implementación Circular de Colas.

InsertarColacir (c,E5):

```
void InsertarColacirc (struct cola *c, tipo_base valor)
{
        if (!ColaLLenacirc (c)
        {
            c.final = siguiente(final);
            c.datos[c.final]=valor;
        }
        else
        {
            cout << "Cola Overflow";
            exit(1);
        }
```



Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

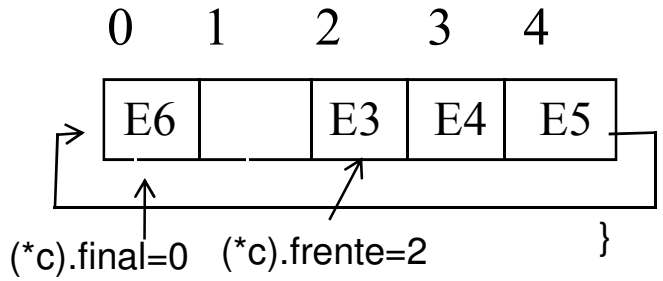
COLAS CIRCULARES

Corrida Implementación Circular de Colas.

InsertarColacir (c,E6):

```
void InsertarColacirc (struct cola *c, tipo_base valor)
{
        if (!ColaLLenacirc (c)
        {
            c.final = siguiente(final);
            c.datos[c.final]=valor;
        }
        else
        {
            cout << "Cola Overflow";
            exit(1);
        }
```

(*c).datos



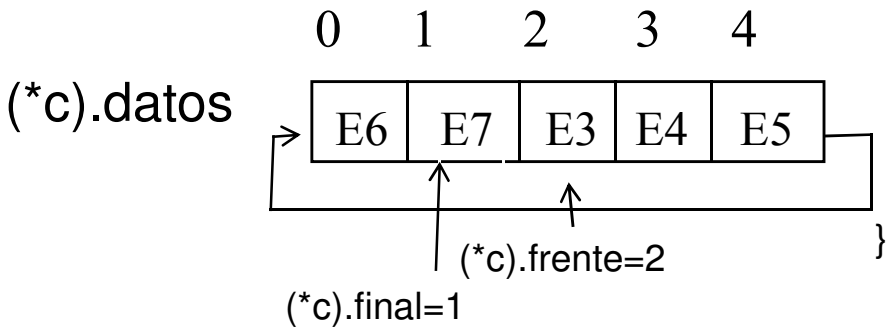
Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

COLAS CIRCULARES

Corrida Implementación Circular de Colas.

InsertarColacir (c,E7):

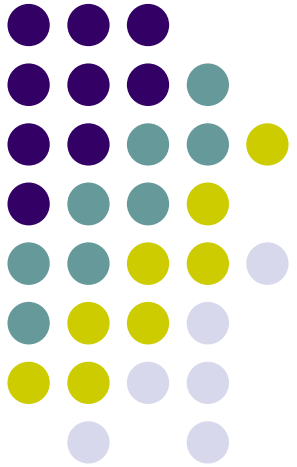
```
void InsertarColacirc (struct cola *c, tipo_base valor)
{
        if (!ColaLLenacirc (c)
        {
            c.final = siguiente(final);
            c.datos[c.final]=valor;
        }
        else
        {
            cout << "Cola Overflow";
            exit(1);
        }
```



Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006

GRACIAS POR SU ATENCIÓN

HASTA LA PRÓXIMA CLASE



Ing. M.Sc. Fulbia Torres
Asignatura: Estructuras de Datos
Barquisimeto 2006