

IMPLEMENTACION ESTATICA

```
typedef int TElementoC;  
typedef struct {  
    TElementoC datos[50];  
    int pri, ult; } TCola;  
void iniciaC (TCola *C) {  
    (*C).pri=-1;  
    (*C).ult=-1;  
}  
int vacíaC(TCola C){  
    return C.pri== -1;  
}  
void poneC (TCola *C, TElementoC dato) {  
    if ((*C).ult != 49) {    // no llena  
        if ((*C).pri== -1)  
            (*C).pri = 0;  
        (*C).ult += 1;  
        (*C).datos[(*C).ult]=dato;  
    }  
}  
void sacaC (TCola *C, TElementoC *dato) {  
    if ((*C).pri != -1) {    // !vacíaC(*C)  
        *dato = (*C).datos[(*C).pri];  
        if ((*C).pri == (*C).ult)  
            iniciaC(C);  
        else  
            (*C).pri +=1;  
    }  
}  
TElementoC consultaC (TCola C){  
    if (C.pri != -1)  
        return C.dato[C.pri];  
    //else return -1;  
}
```

IMPLEMENTACION DINAMICA

```
typedef int TElementoC;
```

```
typedef struct {  
    TElementoC dato;  
    struct nodo * sig;} nodo;
```

```
typedef struct {  
    nodo *pri, *ult;} TCola;
```

```
void iniciaC (TCola *C){  
    (*C).pri=NULL;  
    (*C).ult=NULL;  
}
```

```
int vaciaC(TCola C){  
    return C.pri==NULL;  
}
```

```
void poneC (TCola *C, TElementoC dato) {  
    nodo * aux;  
    aux = (nodo *) malloc (sizeof(nodo));  
    aux->dato = dato;  
    aux->sig = NULL;  
    if ((*C).pri==NULL)  
        (*C).pri=aux;  
    else  
        (*C).ult->sig=aux;  
    (*C).ult=aux;  
}
```

```
void sacaC (TCola *C, TElementoC *dato){  
    nodo * aux;  
    if ((*C).pri !=NULL) {  
        aux = (*C).pri;  
        *dato = aux->dato;  
        if ((*C).pri == (*C).ult)  
            iniciaC(C);  
        else  
            (*C).pri = (*C).pri->sig;  
        free(aux);  
    }  
}
```

```
TElementoC consultaC (TCola C){  
    if (C.pri !=NULL)  
        return C.pri-> dato;  
    //else return -1;  
}
```

IMPLEMENTACION ESTATICA CIRCULAR

```
typedef int TElementoC;
typedef struct {
    TElementoC datos[50];
    int pri, ult; } TCola;

void iniciaC (TCola *C){
    (*C).pri=-1;
    (*C).ult=-1;
}

int vacíaC(TCola C){
    return C.pri==0;
}

void poneC (TCola *C, TElementoC dato) {
    if ! ((*C).ult==49 && (*C).pri==0 || (*C).ult+1==(*C).pri) { // no llena
        if ((*C).pri==0){ //vacía
            (*C).pri = 0;
            (*C).ult = 0;
        }
        else
            if (*C).ult == 49
                (*C).ult = 0;
            Else
                (*C).ult += 1;
        (*C).datos[(*C).ult]=dato;
    }
}

void sacaC (TCola *C, TElementoC *dato) {
    if ((*C).pri != -1) { // !vacíaC(*C)
        *dato = (*C).datos[(*C).pri];
        if ((*C).pri == (*C).ult)
            iniciaC(C);
        else
            if (*C).pri == 49
                (*C).pri = 0;
            Else
                (*C).pri += 1;
    }
}

TElementoC consultaC (TCola C){
    if (C.pri !=-1)
        return C.dato[C.pri];
    //else return -1;
}
```