

Algorítmica y Programación II

Tipos de datos recursivos
Pilas y Colas

Tipo de dato recursivo – Estructuras lineales

- Otras operaciones que debo implementar son:
 - Buscar un elemento
 - Eliminar un elemento
 - Crear la estructura (destruir*)
- Implementar otras estructuras
 - Pila
 - Cola

Pila dinámica

- › Tal como se vio en capítulos anteriores, la pila es una estructura que permite adicionar o eliminar componentes por el principio o tope de la misma.

Type

```
TElemento = String;  
  
TPila = ^TNode;  
TNode = Record  
    info : TElemento;  
    sig  : TPila  
end;
```

Pila – Operaciones

```
procedure crear(var pila:TPila);
```

```
var e:TElemento;
```

```
begin
```

```
    pila := nil;
```

```
end;
```

¿Qué sucede si la misma variable de tipo pila la vuelvo a utilizar, alcanza con llamar a la operación crear?

```
function vacia(const pila:TPila):boolean;
```

```
begin
```

```
vacia := pila = nil;
```

```
end;
```

Pila – Operaciones

```
procedure meter(var pila:TPila;  
                const E:TElemento);  
  
var  
    aux:TPila;  
  
begin  
    new(aux);  
    aux^.info := e;  
    aux^.sig := pila;  
  
    pila := aux;  
  
end;
```

Pila – Operaciones

```
procedure sacar(var pila:TPila;  
                var e:TElemento);  
  
var  
    aux:TPila;  
  
begin  
    aux := pila;  
    pila := pila^.sig;  
    e := aux^.info;  
  
    dispose(aux);  
  
end;
```

Cola dinámica

- › Como definimos anteriormente, la cola es una estructura que permite adicionar elementos y atenderlos en orden.

```
TElemento = Integer;  
TCola = record  
    cab, fin : ^TNode;  
  
end;  
TLista = ^TNode;  
TNode = record  
    info: TElemento;  
    sig : TLista;  
  
end;
```

Cola – Operaciones

```
procedure crear(var cola:TCola);
```

```
var e:TElemento;
```

```
begin
```

```
    cola.cab := nil;
```

```
    cola.fin := nil;
```

```
end;
```

```
function vacia(const cola:TCola):boolean;
```

```
begin
```

```
vacia := cola.cab = nil;
```

```
end;
```


Cola – Operaciones

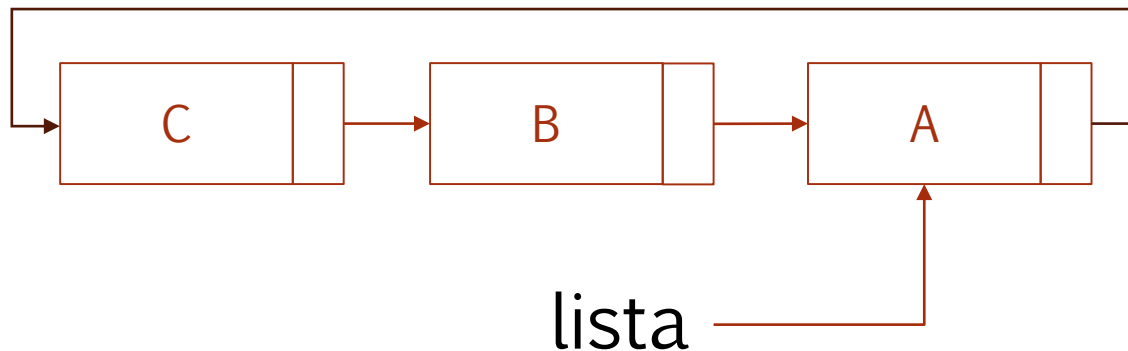
```
procedure meter(var cola:TCola; const E:TElemento);  
var      aux:TLista;  
  
begin  
  new(aux);  
  aux^.info := e;  
  aux^.sig := nil;  
  
  // verifico si es el primer nodo a insertar  
  if(cola.fin = nil)then  
    cola.cab := aux  
  else  
    cola.fin^.sig := aux;  
  
  cola.fin := aux;  
end;
```

Cola – Operaciones

```
procedure sacar(var cola:TCola; var E:TElemento);  
var  
  cabeza:^TNodo;  
begin  
    cabeza := cola.cab;  
    E := cabeza^.info;  
  
    cola.cab := cabeza^.sig;  
  
    //verifico si era el último  
    if(cola.cab = nil)then  
        cola.fin := nil;  
    dispose(cabeza);  
end;
```

Otras implementaciones

- › Cuando nos referimos a la implementación de una cola o una pila dinámica, en general, pensamos en las implementaciones que definimos anteriormente.
- › Si el puntero de una lista referencia al último nodo, que TDA podríamos crear?



TDA - Bibliografía

- Data Structures. Nalle Dale.
- Intermediate Problem Solving and Data Structures - Helman P.- Veroff R.
- Estructuras de datos en C. A, Tenenbaum - Y, Langsam – M, Augenstein.
- https://en.wikipedia.org/wiki/List_of_data_structures