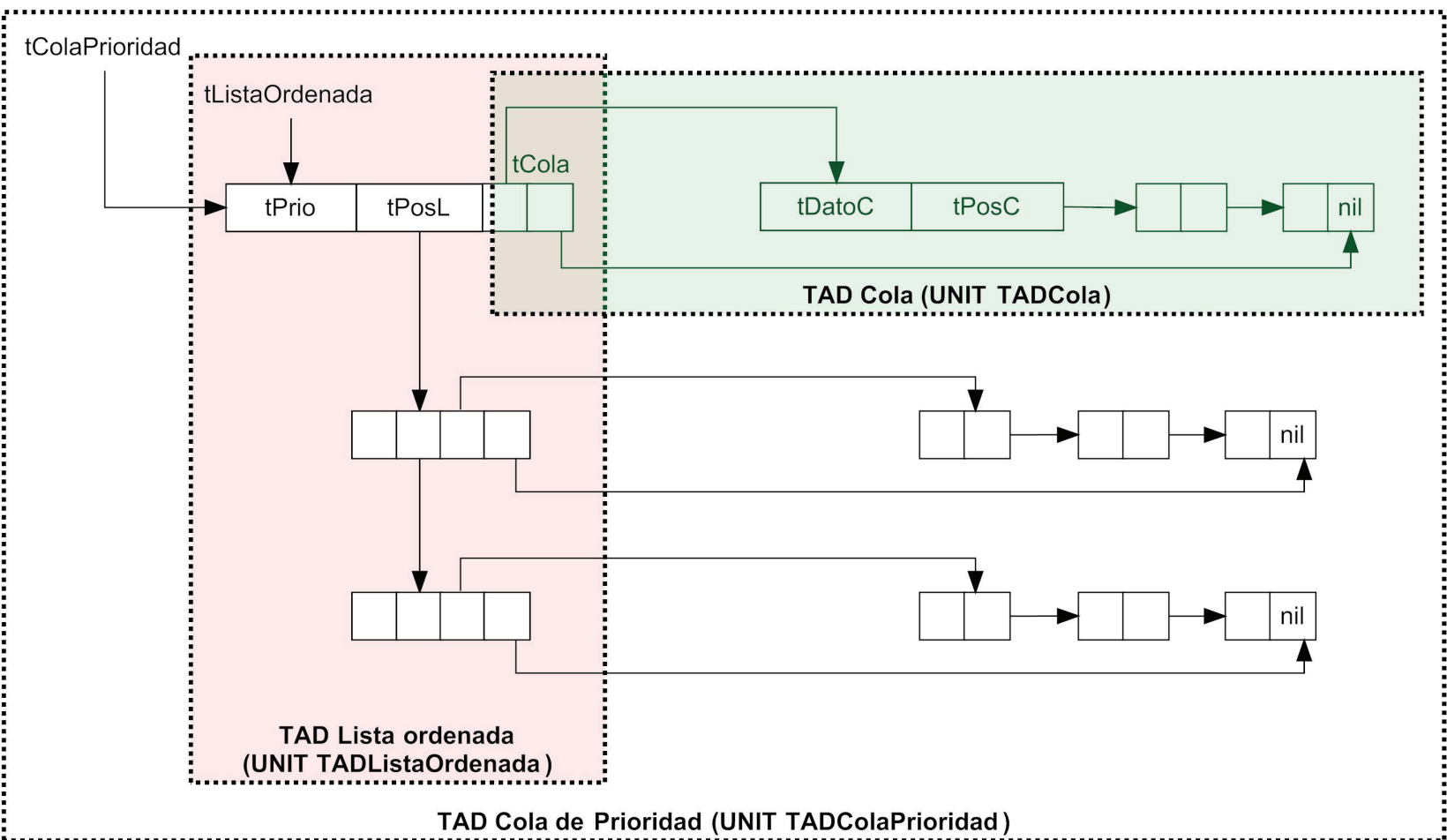




UNIVERSIDADE DA CORUÑA

Programación II

TAD Cola de Prioridad - Un Ejemplo de Implementación



```

UNIT TADCola;
INTERFACE
const
    NULO = nil;
type
    tDatoC = ...;
    tPosC = ^tNodoC;
    tNodoC = record
        datoC : tDatoC;
        sig : tPosC;
    end;
    tCola = record
        ini, fin : tPosC
    end;

procedure ColaVacía (VAR C: tCola);
function InsertarDatoCola (d: tDatoC; VAR C: tCola): boolean;
procedure EliminarDatoCola (VAR C: tCola);
function FrenteCola (C: tCola): tDatoC;
function EsColaVacía (C: tCola): boolean;
...

```

```

UNIT TADListaOrdenada;
INTERFACE
USES TADCola;
const
    MAXPRIO = ...;
type
    tPrioridad = 0..MAXPRIO;
    tDatoL = record
        prio: tPrioridad;
        Cola: tCola;
    end;
    tPosL = ^tNodoL;
    tNodoL = record
        datoL: tDatoL;
        sig: tPosL;
    end;
    tListaOrdenada = tPosL;

function BuscarDato (prio: tPrioridad; L: tListaOrdenada): tPosL;
function InsertarDato (prio: tPrioridad; VAR L: tListaOrdenada): boolean;
procedure ObtenerDato (p: tPosL; L: tListaOrdenada; var prio: tPrioridad; var C: tCola);
procedure ActualizaDato (L: tListaOrdenada; p: tPosL; C: tCola);
...

```

En gris, los detalles de la Lista Ordenada necesarios para usarla como parte de la Cola de Prioridad.

```
UNIT TADColaPrioridad;
INTERFACE
USES TADListaOrdenada, TADCola;
type
    tColaP = tListaOrdenada;

procedure ColaPVacia (VAR ColaP: tColaP);
function InsertarDatoColaP (dato: tDatoC; prio: tPrioridad;
                           VAR ColaP: tColaP): boolean;
procedure EliminarDatoColaP (var ColaP: tColaP);
function EsColaPVacia (ColaP: tColaP): boolean;
function FrenteColaP (ColaP: tColaP): tDatoC;

IMPLEMENTATION
function InsertarDatoColaP (d: tDatoC; prio: tPrioridad;
                           VAR ColaP: tColaP): boolean;

...

```

```
UNIT TADColaPrioridad;
INTERFACE
USES TADListaOrdenada, TADCola;
type
    tColaP = tListaOrdenada;
procedure ColaPVacia (VAR ColaP: tColaP);

function InsertarDatoColaP (dato: tDatoC; prio: tPrioridad;
                           VAR ColaP: tColaP): boolean;
procedure EliminarDatoColaP (var ColaP: tColaP);
function EsColaPVacia (ColaP: tColaP): boolean;
function FrenteColaP (ColaP: tColaP): tDatoC;

IMPLEMENTATION

function InsertarDatoColaP (d: tDatoC; prio: tPrioridad; VAR ColaP: tColaP):
    boolean;
    /* PreCondición: (Para ejercicio en clase) Se supone memoria suficiente para
        realizar la operación */

var
    pos: tPosL;
    C: tCola;
begin
    /* Existe esa prioridad? */
    pos:= BuscarDato (prio, ColaP);

    if pos = nulo then begin
        /* No existe => añadirla */
        InsertarDato (prio, ColaP);
        pos:= BuscarDato (prio, ColaP);
    end;

    /* Recuperar la cola para añadir el dato */
    ObtenerDato (pos, ColaP, prio, C);

    /* Actualizar */
    InsertarDatoCola (d, C);
    ActualizaDato (ColaP, pos, C);
    /* Sin restricciones de memoria. Devolvemos siempre TRUE */
    InstertarDatoColaP := TRUE;
end;
```

```
procedure EliminarDatoColaP (var ColaP: tColaP);  
/* PreCondición: La Cola de Prioridad no está vacía */  
var  
    posL: tPosL;  
    datoL: tDatoL;  
    Cola: tCola;  
    prioridad: tPrioridad;  
begin  
    /* Eliminamos el frente de la cola de máxima prioridad */  
    posL:= Primera(L);  
    ObtenerDato (PosL, ColaP, prioridad, Cola);  
    EliminarDatoCola (Cola);  
    /*Si una vez eliminado la Cola queda vacía, se elimina también el  
       nodo de la lista */  
    If EsColaVacía(Cola)  
    then eliminarPosicion(PosL, ColaP)  
    else ActualizarDato (ColaP, posL, Cola)  
end;
```