



ALGORITMOS Y ESTRUCTURAS DE DATOS II

Ingeniería Técnica en Informática de Gestión
Ingeniería Técnica en Informática de Sistemas

CURSO 2002/03

Implementación del TAD Pila

Representación estática

```
módulo pilas
importa defTipoElemento
exporta (* La documentación del módulo de definición se deja como ejercicio para el alumno *)
constante max = 100
tipo pila

acción crearPila (var p:pila)
acción apilar (var p:pila; e:elemento)
acción desapilar (var p:pila)
función cima (p:pila): elemento
función esVacía (p:pila): booleano

implementación
tipo pila = registro
    dato: tabla [1..max] de elemento;
    indCima: 0..max;
fregistro;

acción crearPila (var p:pila);
    p.indCima:= 0
facción

acción apilar (var p:pila; e:elemento);
    p.indCima:= p.indCima +1;
    p.dato[p.indCima]:= e;
facción

acción desapilar (var p:pila);
    p.indCima:= p.indCima -1;
facción

función cima (p:pila): elemento;
    retorna (p.dato[p.indCima]);
ffunción

función esVacía (p:pila): booleano;
    retorna (p.indCima = 0);
ffunción

fmódulo
```

Representación dinámica

módulo pilas
importa defTipoElemento
exporta (* La documentación del módulo de definición se deja como ejercicio para el alumno *)
tipo pila

acción creaPila (var p:pila)
acción apilar (var p:pila; e:elemento)
acción desapilar (var p:pila)
función cima (p:pila): elemento
función esVacía (p:pila): booleano
acción asignar (var nueva: pila; vieja: pila);
acción liberar (var p: pila);

implementación

tipos pila = puntero a unDato
unDato = registro
dato: elemento;
sig: pila;

fregistro

ftipos

acción creaVacía (var p:pila);
p:= nil

facción

acción apilar (var p:pila; e:elemento);
var aux: pila; fvar
aux:= p;
reservar(p);
p^.dato:= e;
p^.sig:= aux

facción

acción desapilar (var p:pila);
var aux: pila; fvar
aux:= p;
p:= p^.sig;
liberar (aux)

facción

función cima (p:pila): elemento;
retorna (p^.dato)

ffunción

función esVacía (p:pila): booleano;
retorna (p = nil);

ffunción

acción asignar (**var** nueva: pila; vieja: pila);
(* duplica la representación de la pila vieja guardándola en la pila nueva *)

var aux: pila **fvar**

si esVacía (vieja) **entonces** creaVacía (nueva)

sino

reservar (nueva);
nueva^.dato:= vieja^.dato;
aux:= nueva;
vieja:= vieja^.sig;

mientras vieja ≠ nil **hacer**

reservar (aux^.sig);
aux:= aux^.sig;
aux^.dato:= vieja^.dato;
vieja:= vieja^.sig;

fmientras

aux^.sig:= nil

fsi

facción

acción liberar (**var** p: pila);

var aux: pila **fvar**

aux:= p;

mientras p ≠ nil **hacer**

p:= p^.sig;
liberar (aux);
aux:= p;

fmientras

facción

fmódulo