```
-- AUTORA: Elvira Mayordomo Cámara
-- PROYECTO: módulo de implementación del TAD listas con punto de interés a partir
-- de pilas limitadas
-- FICHERO: listaspi.adb
-- FECHA: 16-10-03

package body listaspi is

    procedure crear(p1,p2:in pila;l:out lpi) is
    -- Post: l=<p1,p2>
    -- coste en tiempo: para una lista de n datos tarda O(n)
    -- Esta operación es auxiliar y no es necesario implementarla
    begin
        copiar(p1,l.piladcha);
        copiar(p2,l.pilaizda);
    end crear;

    procedure   listaVacia(l:out lpi) is
    -- Post: l=listaVacía
    -- coste en tiempo O(1)
    begin
        creaVacia(l.piladcha);
        creaVacia(l.pilaizda);
```

```
end listaVacia;

procedure insertar(l:in out lpi; e:in elemento) is
-- Pre: l=<p1,p2> ∧ (¬esllena(p1))
-- Post: l=insertar(<p1,p2>,e)
-- coste en tiempo O(1)
begin
      apilar(l.pilaizda,e);
end insertar;

procedure borrar(l:in out lpi) is
-- Pre: l=l0
-- Post: l=borrar(l0)
-- coste en tiempo O(1)
begin
      if not esfinal(l) then desapilar(l.piladcha); end if;
end borrar;

function actual(l:lpi) return elemento is
-- Pre: ¬alfinal(l)
-- Post: actual(l)=actual(l)
-- coste en tiempo O(1)
begin
```

```
        return cima(l.piladcha);
end actual;

function esVacia(l:lpi) return boolean is
-- Post: esVacia(l)= esvacía(l)
-- coste en tiempo O(1)
begin
        return (esVacia(l.piladcha) and esVacia(l.pilaizda));
end esVacia;

procedure principio(l:in out lpi) is
-- Pre: l=<p1,p2> ∧ (tamaño(p1)+tamaño(p2)<=capacidad)
-- Post: l=principio(<p1,p2>)
-- coste en tiempo: si la lista tenía m datos antes del punto de interés
-- tarda O(m)
begin
        while (not esVacia(l.pilaizda)) loop
                apilar(l.piladcha,cima(l.pilaizda));
                desapilar(l.pilaizda);
        end loop;
end principio;

procedure avanzar(l:in out lpi) is
```

```
    -- Pre: l=<p1,p2> ∧ (esvacía(p2) ∨ (¬esllena(p1))
    -- Post: l=avanzar(l0)
    -- coste en tiempo O(1)
    begin
        if not esVacia(l.piladcha) then
            apilar(l.pilaizda,cima(l.piladcha));
            desapilar(l.piladcha);
        end if;
    end avanzar;

    function esFinal(l:lpi) return boolean is
    -- Post: esFinal(l)= alfinal(l)
    -- coste en tiempo O(1)
    begin
        return esVacia(l.piladcha);
    end esFinal;

end listaspi;
```