```
-- AUTORA: Elvira Mayordomo Cámara
-- PROYECTO: módulo de declaración del TAD listas con punto de interés con
-- punteros
-- FICHERO: listaspi2.ads
-- FECHA: 16-10-03

-- especificación listaspi
--  parámetro formal
--    género elemento
--  fpf
--  usa booleanos, pilas(elemento)
--    género lpi
--  operaciones
--    <_,_>: pila pila → lpi
--    listaVacía: → lpi
--    insertar: lpi elemento → lpi
--    borrar: lpi → lpi
--    parcial actual: lpi → elemento
--    esvacía: lpi → bool
--    principio: lpi → lpi
--    avanzar: lpi → lpi
```

```
--    alfinal: lpi → bool
-- dominios de definición p1,p2:pila; e:elemento
--    actual(<p1,apilar(p2,e)>)
-- ecuaciones p1,p2:pila; e:elemento
--    listaVacía = <pilaVacía, pilaVacía>
--    insertar(<p1,p2>,e) = <apilar(p1,e),p2>
--    borrar(<p1,pilaVacía>) = <p1,pilaVacía>
--    borrar(<p1,apilar(p2,e)>) = <p1,p2>
--    actual(<p1,apilar(p2,e)>) = e
--    esvacía(<pilaVacía, pilaVacía>) = verdad
--    esvacía(<apilar(p1,e),p2>) = falso
--    esvacía(<p1,apilar(p2,e)>) = falso
--    principio(<apilar(p1,e),p2>) = principio(<p1,apilar(p2,e)>)
--    principio(<pilaVacía,p1>) = <pilaVacía,p1>
--    avanzar(<p1,pilaVacía>) = <p1,pilaVacía>
--    avanzar(<p1,apilar(p2,e)>) = <apilar(p1,e),p2>
--    alfinal(<p1,pilaVacía>) = verdad
--    alfinal(<p1,apilar(p2,e)> = falso
-- fespec

generic
```

```
    type elemento is private;
package listaspi2 is
    type lpi is limited private;
    -- coste en memoria: una lista de n datos ocupa O(n)
    -- procedure crear(p1,p2:in pila;l:out lpi);
    -- Post: l=<p1,p2>
    -- Esta operación es auxiliar y no es necesario implementarla
    procedure   listaVacia(l:out lpi);
    -- Post: l=listaVacía
    -- coste en tiempo O(1)
    procedure insertar(l:in out lpi; e:in elemento);
    -- Pre: l=l0
    -- Post: l=insertar(l0,e)
    -- coste en tiempo O(1)
    procedure borrar(l:in out lpi);
    -- Pre: l=l0
    -- Post: l=borrar(l0)
    -- coste en tiempo O(1)
    function actual(l:lpi) return elemento;
    -- Pre: ¬alfinal(l)
    -- Post: actual(l)=actual(l)
```

```
    -- coste en tiempo O(1)
    function esVacia(l:lpi) return boolean;
    -- Post: esVacia(l)= esvacía(l)
    -- coste en tiempo O(1)
    procedure principio(l:in out lpi);
    -- Pre: l=l0
    -- Post: l=principio(l0)
    -- coste en tiempo O(1)
    procedure avanzar(l:in out lpi);
    -- Pre: l=l0
    -- Post: l=avanzar(l0)
    -- coste en tiempo O(1)
    function esFinal(l:lpi) return boolean;
    -- Post: esFinal(l)= alfinal(l)
    -- coste en tiempo O(1)
    procedure liberar(l: in out lpi);
    -- Libera la memoria dinámica accesible desde l, quedando l
    -- vacía.
    -- coste en tiempo O(n) para una lista de n elementos

private
```

```
        type unDato;
        type ptUnDato is access unDato;
        type unDato is
                record
                        dato:elemento;
                        sig:ptUnDato;
                end record;
        type lpi is
                record
                        prim, ptoInt, antPI: ptUnDato;
                end record;
end listaspi2;
```