```
-- AUTOR: Elvira Mayordomo Cámara
-- PROYECTO: módulo de declaración del TAD historias
-- FICHERO: historias.ads
-- FECHA: 22-10-03

-- especificación historias
--   usa fechas, cadenas, bool - -fechas tiene operaciones "<" y ">"
--   géneros página, historia
--   operaciones
--    creapágina: cadena fecha → página
--    url: página → cadena
--    lafecha: página → fecha
--
--    historiaVacía: → historia
--    parcial insertar: historia página → historia
--    borrar: historia cadena → historia
--    está: historia cadena → bool
--    parcial buscar: historia cadena → fecha
--    esvacía: historia → bool
```

-- **parcial** últimafecha: historia $\rightarrow$ fecha

-- **dominios de definición** h:historia; p:página; c:cadena
-- $\neg$esvacía(h) $\rightarrow$ últimafecha(h)
-- insertar(historiaVacía,p)
-- $\neg$(lafecha(p)<últimafecha(h)) $\rightarrow$ insertar(h,p)
-- está(h,c) $\rightarrow$ buscar(h,c)
-- **ecuaciones** c: cadena; f: fecha; p,p1,p2: página; h: historia
-- url(creapágina(c,f)) = f
-- lafecha(creapágina(c,f)) = f
-- url(p1)=url(p2) $\rightarrow$ insertar(insertar(h,p1),p2) = insertar(h,p2)
-- url(p1)$\neq$url(p2) $\wedge$ lafecha(p1)= lafecha(p2) $\rightarrow$
-- insertar(insertar(h,p1),p2) = insertar(insertar(h,p2),p1)
-- borrar(historiaVacía,c) = historiaVacía
-- url(p)=c $\rightarrow$ borrar(insertar(h,p),c) = borrar(h,c)
-- url(p) $\neq$c $\rightarrow$ borrar(insertar(h,p),c) = insertar(borrar(h,c),p)
-- está(historiaVacía,c) = falso
-- está(insertar(h,p),c) = (url(p)=c) $\vee$ está(h,c)
-- url(p)=c $\rightarrow$ buscar(insertar(h,p),c) = lafecha(p)

```
--    url(p) ≠c → buscar (insertar(h,p),c) = buscar(h,c)
--    esvacía(historiaVacía) = verdad
--    esvacía(insertar(h,p)) = falso
--    últimafecha(insertar(h,p)) = lafecha(p)
-- fespec

with fechas, Ada.Strings.Unbounded, ustrings;
use fechas, Ada.Strings.Unbounded, ustrings;
package historias is
      type pagina is
            record
                  elurl: ustring;
                  lafecha: fecha;
            end record;
      type historia is limited private;
      procedure creaVacia(h:out historia);
      -- Post: h=historiaVacía
      -- coste en tiempo O(1)
      procedure insertar(h:in out historia; url:in ustring; f:in fecha);
      -- Pre: (h=h0) ∧ (¬(lafecha(creapágina(url,f))<últimafecha(h)) ∨ esvacía(h))
```

```
-- Post: h=insertar(h0, creapágina(url,f))
-- coste en tiempo: para una historia de n datos tarda O(n)
procedure borrar(h:in out historia; url:in ustring);
-- Pre: h=h0
-- Post: h=borrar(h0,url)
-- coste en tiempo: para una historia de n datos tarda O(n)
function esta(h:historia; url:ustring) return boolean;
-- Post: esta(h,url)= está(h,url)
-- coste en tiempo: para una historia de n datos tarda O(n)
function buscar(h:historia; url:ustring) return fecha;
-- Pre: está(h,url)
-- Post: buscar(h,url)=buscar(h,url)
-- coste en tiempo: para una historia de n datos tarda O(n)
function esVacia(h:historia) return boolean;
-- Post: esVacia(h)= esvacía(h)
-- coste en tiempo O(1)
function ultimafecha(h:historia) return fecha;
-- Pre: ¬esvacía(h)
-- Post: ultimafecha(h)=últimafecha(h)
-- coste en tiempo O(1)
```

```
private
        type unDato;
        type historia is access unDato;
        type unDato is
                record
                        dato: pagina;
                        sig: historia;
                end record;
end historias;
```