

Estructuras de Datos y Algoritmos - Algoritmos 1

Trabajo Práctico de Taller N° 1

Año 2003

Ejercicio 1:

Implementar en *Haskell* los tipos abstractos Stack (Pila) y Queue (Cola) genéricos con las siguientes funciones:

```
push      :: a -> Stack a -> Stack a
pop       :: Stack a -> Stack a
top       :: Stack a -> a
empty_stack :: Stack a -> Bool

enqueue   :: a -> Queue a -> Queue a
dequeue   :: Queue a -> Queue a
front     :: Queue a -> a
empty_queue :: Queue a -> Bool
```

Ejercicio 2:

Implementar en *Pascal* los tipos abstractos de datos Stack y Queue con las funciones equivalentes a la implementación en *Haskell* utilizando variables estáticas.

Ejercicio 3:

Implementar en *Pascal* el tipo abstracto de datos List (Lista) de elementos de tipo

`integer`

utilizando variables dinámicas. Dicha implementación deberá proveer, al menos, las siguientes acciones y funciones:

```
procedure empty_list(var l:list);
```

Esta acción toma una lista y la vacía

```
procedure insert_element(e: integer; pos: integer; var l:list);
```

Toma un valor entero, una posición y una lista, e inserta el valor en la posición correspondiente de dicha lista.

```
procedure delete_element(pos:integer; var l:list);
```

Toma una posición y una lista, y elimina el elemento ubicado en dicha posición.

```
function length_list(l:list): integer;
```

Toma una lista y devuelve su longitud.

```
function get_element(pos: integer; l:list): integer;
```

Toma una posición y una lista, y devuelve el elemento ubicado en dicha posición.