

## 1. Aclaraciones

## 2. Módulo Diccionario Trie( $\alpha$ )

### Interfaz

### Representación

## 3. Módulo Cola de Prioridad Heap( $\alpha$ )

### Interfaz

### Representación

## 4. Módulo Conjunto AVL( $\alpha$ )

### Interfaz

#### parámetros formales

**géneros**  $\alpha$

**función** COPIAR( $\text{in } a : \alpha \rightarrow res : \alpha$ )

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res =_{\text{obs}} a\}$

**Complejidad:**  $\Theta(\text{copy}(a))$

**Descripción:** función de copia de  $\alpha$ 's

**se explica con:** SECUENCIA( $\alpha$ ), ITERADOR BIDIRECCIONAL( $\alpha$ ).

**géneros:** lista( $\alpha$ ), itLista( $\alpha$ ).

### Operaciones básicas de lista

VACÍA()  $\rightarrow res : \text{lista}(\alpha)$

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res =_{\text{obs}} <>\}$

**Complejidad:**  $\Theta(1)$

**Descripción:** genera una lista vacía.

AGREGARADELANTE( $\text{in/out } l : \text{lista}(\alpha), \text{in } a : \alpha \rightarrow res : \text{itLista}(\alpha)$ )

**Pre**  $\equiv \{l =_{\text{obs}} l_0\}$

**Post**  $\equiv \{l =_{\text{obs}} a \bullet l_0 \wedge res = \text{CrearItBi}(<>, l) \wedge \text{alias}(\text{SecuSuby}(res) = l)\}$

**Complejidad:**  $\Theta(\text{copy}(a))$

**Descripción:** agrega el elemento  $a$  como primer elemento de la lista. Retorna un iterador a  $l$ , de forma tal que Siguiente devuelva  $a$ .

**Aliasing:** el elemento  $a$  agrega por copia. El iterador se invalida si y sólo si se elimina el elemento siguiente del iterador sin utilizar la función ELIMINARSIGUIENTE.

### Operaciones del iterador

CREARIT( $\text{in } l : \text{lista}(\alpha) \rightarrow res : \text{itLista}(\alpha)$ )

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res =_{\text{obs}} \text{crearItBi}(<>, l) \wedge \text{alias}(\text{SecuSuby}(it) = l)\}$

**Complejidad:**  $\Theta(1)$

**Descripción:** crea un iterador bidireccional de la lista, de forma tal que al pedir SIGUIENTE se obtenga el primer elemento de  $l$ .

**Aliasing:** el iterador se invalida si y sólo si se elimina el elemento siguiente del iterador sin utilizar la función `ELIMINARSIGUIENTE`.

`CREARITULT(in l: lista( $\alpha$ ))  $\rightarrow$  res : itLista( $\alpha$ )`

**Pre**  $\equiv \{\text{true}\}$

**Post**  $\equiv \{res =_{\text{obs}} \text{crearItBi}(l, <>) \wedge \text{alias}(\text{SecuSuby}(it) = l)\}$

**Complejidad:**  $\Theta(1)$

**Descripción:** crea un iterador bidireccional de la lista, de forma tal que al pedir `ANTERIOR` se obtenga el último elemento de  $l$ .

**Aliasing:** el iterador se invalida si y sólo si se elimina el elemento siguiente del iterador sin utilizar la función `ELIMINARSIGUIENTE`.

## Representación

### 5. Módulo Restricción

Interfaz

Representación

### 6. Módulo Grafo

Interfaz

Representación

### 7. Módulo Ciudad

Interfaz

Representación