

### 3. Módulo Registro

El módulo Registro provee campos asociados con valores (cadenas de caracteres o *strings*). El módulo provee operaciones para crear un nuevo registro y obtener el conjunto de campos del registro en tiempo constante y definir un registro y obtener el valor correspondiente a determinado campo (en tiempo lineal respecto de la longitud del nombre del campo y del valor a definir).

#### Interfaz

se explica con: TAD REGISTRO

géneros: registro.

#### Operaciones básicas de Registro

NUEVO()  $\rightarrow res : \text{registro}$

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

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

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

**Descripción:** crea un registro vacío.

DEFINIR(**in/out**  $r : \text{registro}$ , **in**  $c : \text{nombre\_campo}$ , **in**  $v : \text{valor}$ )

**Pre**  $\equiv \{r = r_0\}$

**Post**  $\equiv \{r =_{\text{obs}} \text{definir}(r_0, c, v)\}$

**Complejidad:**  $\Theta(|c| + |v|)$

**Descripción:** si no existe el campo  $c$ , lo agrega al registro  $r$  y le asocia el valor  $v$ . Si el campo ya existe, reemplaza el valor existente con el valor  $v$ .

CAMPOS(**in**  $r : \text{registro}$ )  $\rightarrow res : \text{conj}(\text{nombre\_campo})$

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

**Post**  $\equiv \{res =_{\text{obs}} \text{campos}(r)\}$

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

**Descripción:** devuelve un conjunto que contiene todos los nombres de los campos del registro  $r$ .

**Aliasing:**  $res$  no es modificable y se devuelve por referencia.

•[•](**in**  $r : \text{registro}$ , **in**  $c : \text{nombre\_campo}$ )  $\rightarrow res : \text{valor}$

**Pre**  $\equiv \{c \in \text{campos}(r)\}$

**Post**  $\equiv \{res =_{\text{obs}} r[c]\}$

**Complejidad:**  $\Theta(|c| + |v|)$

**Descripción:** devuelve el valor asociado al campo  $c$  en el registro  $r$ .

#### Representación

Los registros se representan con una estructura de diccionario sobre Trie. La estructura diccionario cumple la restricción que debe tener el invariante de representación de un registro, es decir, que no puede haber campos repetidos. Por este motivo, al usar esta estructura el invariante se cumple siempre.

#### Representación del registro

registro se representa con  $\text{diccTrie}(\text{nombre\_campo}, \text{valor})$

donde  $\text{nombre\_campo}$  y  $\text{valor}$  son `string`

$\text{Rep} : \text{registro} \rightarrow \text{bool}$

$\text{Rep}(e) \equiv \text{true} \iff (\text{se cumple el invariante de representación de diccionario})$

$\text{Abs} : \text{estr } e \rightarrow \text{registro } r$

$\text{Abs}(e) \equiv r : \text{registro} \mid \text{campos}(r) =_{\text{obs}} \text{Campos}(e) \wedge (\forall c : \text{nombre\_campo})(c \in \text{campos}(r) \Rightarrow_L r[c] =_{\text{obs}} e[c])$   $\{\text{Rep}(e)\}$

## Algoritmos

---

---

**iNuevo** $(\rightarrow res : \text{registro})$

1:  $res \leftarrow \text{diccTrie}(string, string)::\text{Vacío}()$

Complejidad:  $\Theta(1)$

---

---

---

**iDefinir** $(\text{in/out } r : \text{registro}, \text{in } c : \text{nombre\_campo}, \text{in } v : \text{valor})$

1:  $r \leftarrow \text{diccTrie}(string, string)::\text{Definir}(r, c, v)$

Complejidad:  $\Theta(|c| + |v|)$

---

---

---

**iCampos** $(\text{in } r : \text{registro}) \rightarrow res : \text{conj}(\text{nombre\_campo})$

1:  $res \leftarrow \text{diccTrie}(string, string)::\text{Claves}(r)$

Complejidad:  $\Theta(1)$

---

---

---

**i[•]** $(\text{in } r : \text{registro}, \text{in } c : \text{nombre\_campo}) \rightarrow res : \text{valor}$

1:  $res \leftarrow \text{diccTrie}(string, string)::\text{Significado}(r, c)$

Complejidad:  $\Theta(|c| + |v|)$

---