Práctica 6

EDD

Matú Hernández Diana

Rojo Mata Daniel

Diseño de TDA Nodo

Especificación

- TDA: **Nodo** $\Big($**Valores:** $\{null\} \cup \{(e, s) : e \in T, s \in RefNodo\}$;
  Campos: elemento, siguiente;
  Operaciones: DarSiguiente, DarElemento, ModificarElemento, ModificarSiguiente $\Big)$

Sintaxis

- ***NodoNuevo (T)** $\rightarrow$ **Nodo**
  - Descripción: Crea un nuevo nodo con el elemento proporcionado y el siguiente nodo inicializado como null.
  - Entrada: Un elemento a del tipo T.
  - Salida: Un nuevo nodo con el elemento a y el siguiente nodo inicializado como null.
- **DarSiguiente (Nodo)** $\rightarrow$ **Referencia de tipo Nodo**
  - Descripción: Devuelve la referencia al siguiente nodo del nodo dado.
  - Entrada: Un nodo de tipo Nodo.
  - Salida: Un nodo, que es el siguiente nodo del nodo dado, o null si el nodo dado es null.
- **DarElemento (Nodo)** $\rightarrow$ **T**

- Descripción: Devuelve el elemento del nodo dado.
- Entrada: Un nodo de tipo Nodo.
- Salida: El elemento del nodo dado, o null si el nodo dado es null.

- **ModificarElemento (Nodo, T) → Nodo**
  - Descripción: Modifica el elemento del nodo dado con el nuevo elemento proporcionado.
  - Entrada: Un nodo de tipo Nodo y un nuevo elemento del tipo T.
  - Salida: El nodo modificado con el nuevo elemento, o un nuevo nodo con el nuevo elemento si el nodo dado es null.
- **ModificarSiguiente (Nodo, T) → Nodo**
  - Descripción: Modifica la referencia al siguiente nodo del nodo dado con el nuevo nodo proporcionado.
  - Entrada: Un nodo de tipo Nodo y un nuevo nodo de tipo Nodo.
  - Salida: El nodo modificado con el nuevo siguiente nodo, o un nuevo nodo con el nuevo siguiente nodo si el nodo dado es null.

Semántica

- Sea $n \in Nodo$ con n distinto de *null*, y $a, d \in T$ (genérico).

Constructor

```
*NodoNuevo(a) =>
    elemento <- a
    siguiente <- Null
```

Método DarSiguiente

```
DarSiguiente (null) => ERROR
DarSiguiente (*NodoNuevo(a)) =>
    entonces *NodoNuevo(a).siguiente

OPCIONAL:
DarSiguiente (n) =>
    entonces n.siguiente
```

Método DarElemento

```
DarElemento (null) => null
DarElemento (*NodoNuevo(a)) =>
    entonces *NodoNuevo(a).elemento

OPCIONAL:
DarElemento (n) =>
    entonces n.elemento
```

Método ModificarElemento

```
ModificarElemento (null, a) =>
    entonces *NuevoNodo(a)
ModificarElemento (*NodoNuevo(a), d ) =>
```

```
    entonces DarElemento(*NuevoNodo(a)) <- d

OPCIONAL:
ModificarElemento (n, d ) =>
    entonces n.elemento <- d
```

Método ModificarSiguiente

```
ModificarSiguiente (null, a) => entonces ERROR
ModificarSiguiente (*NodoNuevo(a), d ) =>
    entonces DarElemento(DarSiguiente(*NuevoNodo(a))) <- d

OPCIONAL:
ModificarSiguiente (n, d ) =>
    entonces n.siguiente.elemento <- d
```