**Respuestas UT3 – PD3**



Se creará una instancia de Lista y de Nodo. El nodo se establecerá como cabeza de la lista.

Precondiciones:

- No hay ninguna en específica.

Postcondiciones:

- Los nodos deberían quedar insertados en la lista

- La lista tendría como largo la cantidad de nodos insertados.

Seudocódigo:

crear nuevoNodo

crear otroNodo

crear una Lista

cabezaDeLista <- nuevoNodo

insertar otroNodo en la lista creada.



Se insertará un nodo al final de una lista con un método “insertar”.

Precondiciones:

- Debe existir una lista de nodos

Postcondiciones:

- El nodo seleccionado quedará insertado al final de la lista

- El largo de la lista sumará 1

Seudocódigo:

nuevo nodoInsertable

insertar nodoInsertable en Lista



Se tomará una clave, la cual será comparada con la clave de cada nodo, hasta encontrar el equivalente. Al encontrar un nodo equivalente, es retornado. Si no lo encuentra, devuelve un error.

Precondiciones:

* Debe existir una lista.
* La lista debe contener 1 o más elementos.

Postcondiciones:

* Si el elemento se encuentra en la lista, se devolverá el elemento, sino un error.

Seudocódigo:

ElementoABuscar <- Clave

Para cada Nodo en Lista:

Si Clave = Clave de Nodo:

Devolver Nodo

FinSi

FinPara

Devolver error

Fin



Se creará un método imprimir, el cual vaya recorriendo nodo por nodo con un bucle, imprimiendo cada uno en consola.

Precondiciones:

* Debe existir una lista con 1 elemento o más.
* Los nodos deben contener un atributo Dato no vacío para ser impreso.

Poscondiciones:

* Quedarán impresos los datos de cada nodo, en este caso en consola.

Seudocódigo:

Para cada Nodo en Lista:

Imprimir Nodo.Dato

FinPara



Primero se debe fijar que el nodo Cabeza de lista no sea el que hay que eliminar. Si es el primero, se debe asignar al siguiente como cabeza.

Luego, se recorre nodo por nodo, revisando la clave del nodo siguiente. Si el nodo siguiente es el que hay que eliminar, la referencia del nodo actual al próximo se debe adelantar un lugar, salteando al eliminado y dejándolo fuera.

Precondiciones:

* Debe existir una lista con al menos un elemento para buscar.

Poscondiciones:

* La lista terminará con un nodo menos, ya que uno será excluido de los enlaces.

Seudocódigo:

Si Clave de CabezaDeLista = ClaveAEliminar

Cabeza de Lista <- CabezaDeLista.siguiente

FinSi

Para cada Nodo en Lista:

Si Siguiente.Clave = ClaveAEliminar:

Nodo.siguiente = Nodo.siguiente.siguiente

Devolver true

FinSi

FinPara

Devolver False

Fin