

TC1031.11: Programación de estructuras de datos y algoritmos fundamentales (GPO 11)

Casos prueba de la act 2.2

CASOS DE PRUEBA ACT 1.3					
No.	Precondiciones	Resumen	Pasos	Resultado Esperado	Resultado obtenido
1	Tener la función de inserción "add" e inicializar una lista	El usuario desea añadir elementos a lista cuando este se presente vacía.	<ol style="list-style-type: none"> 1. Inicializar la lista 2. Usar la función add con el dato a añadir o usar la función add con el elemento y el índice 0 	El elemento que se quiere añadir se añade a la lista	La lista añade el elemento que el usuario deseo añadir.
2	Tener la función de inserción "add", tener inicializado la lista y tener un elemento en ella.	El usuario desea añadir un elemento al principio de la lista convirtiéndolo en su primer elemento.	<ol style="list-style-type: none"> 1. Usar el método add a la lista. 2. Dentro del método escribir el valor a añadir y el índice 0 	El elemento que se desee añadir aparezca al inicio de la lista.	Al imprimir la lista el elemento añadido aparece al principio.
3	Tener la función de inserción "add", tener inicializado la lista y tener mas de un elemento en ella.	El usuario desea añadir un elemento en medio o en una posición n en la lista.	<ol style="list-style-type: none"> 1. Usar el método add a la lista. 2. Dentro del método escribir el valor a añadir y el índice donde quiera el usuario. 	El elemento se añada en dicho índice en la lista.	Al imprimir la lista el elemento que se quiere añadir en n posición aparece.

TC1031.11: Programación de estructuras de datos y algoritmos fundamentales (GPO 11)

Casos prueba de la act 2.2

4	Tener la función de inserción "add", tener inicializado la lista y tener más de un elemento en ella.	El usuario desea añadir un elemento al final de la lista.	<ol style="list-style-type: none">1. Usar el método add en la lista.2. Dentro del método añadir el elemento que desee al final en la lista.	El elemento se añade al final de la lista y se muestra en pantalla.	Al imprimir la lista el elemento que se quiso añadir aparece al final.
---	--	---	--	---	--

Explicación/Reflexión:

Para la realización de esta actividad se implemento e concepto de sobrecarga de funciones ya que se crearon dos funciones con el mismo nombre "add", en uno nadamas como parámetro el dato que se quiera añadir, cubriendo el caso en donde el usuario quiera añadir un elemento al final de la lista y el otro con parámetros de el dato que se quiera añadir y un entero llamado index siendo este el índice en donde se quiera añadir dicho elemento, cubriendo en ese ultimo en caso que el usuario quiera añadir un elemento al principio de la lista o añadir un elemento en n posición, a continuación presentaremos las funciones:

- Método add con parámetro solo del dato:

```
//función para insertar nodo al final de la lista
void add(T data) {
    if (!first) {
        // En caso si la lista esta vacía.
        first = new node<T>;
        first->data = data;
        first->next = NULL;
    }
    // Cuando la lista no esta vacía.
    else {
        //Creación de un nuevo nodo.
        node<T>* newNode = new node<T>;
        newNode->data = data;
        newNode->next = NULL;
        //Creación de un nodo temporal "last" apuntando al primer nodo.
        node<T>* last = this->first;
        //Bucle while que recorre el puntero hasta que cuando nodolast->next = NULL
        while (last->next != NULL){
            last = last->next;
        }
        //el nodolast->next = al primer nodo creado con el elemento que se añadio.
        last->next = newNode;
    }
}
```

Casos prueba de la act 2.2

En este método lo primero que pregunta es si la lista esta vacía con el if(), en caso que lo sea crea la lista y añade el elemento deseado en ella, pero en caso que la lista no este vacía se crea un nuevo nodo, que su parte data se añadira el elemento deseado y su next será igual a NULL, creamos un nodo puntero temporal llamado "last" que es igual al primer nodo de la lista y este entra en un bucle while endonde el nodo puntero last ira recorriendo la lista hasya que su parte next sea NULL, una vez llegado a ese periodo, se actualiza el nodo "last" next sería igual al primer nodo creado con el dato añadido, cubriendo el caso en cuando el usuario quiera añadir un elemento al final de la lista.

- Método add con parámetros del dato y el índice;

```
//función para insertar nodo en n posición
void add(T data, int index){
    node<T>* temp1 = new node<T>;
    temp1->data = data;
    temp1->next = NULL;
    //En caso que el usuario quiera insertar el nodo al principio
    if(index == 0){
        temp1->next = this->first;
        this->first = temp1;
        return;
    }
    node<T>* temp2 = this->first;
    //iterador para buscar la posición-1
    for(int i = 0; i < index-1; i++){
        temp2 = temp2->next;
    }
    //inserción del nodo a la lista
    temp1->next = temp2->next;
    temp2->next = temp1;
}
```

En este método add tiene como parámetros el dato el cual el usuario desea añadir y un entero llamado index, una vez llamado el método primero creamos un nodo temporal que este tendrá en su parte data el elemento que el usuario desee y su next será un NULL, hacemos un condicional en caso que el usuario quiera añadir el elemento al inicio de la lista, ósea que el index sea 0, entonces en caso que el condicional sea true el nodo creado ->next será igual al puntero first que este apunta al primer nodo anteriormente y después recorremos el puntero first a que apunte a nuestro nodo creado para así que el elemento se añada al principio de la lista.

TC1031.11: Programación de estructuras de datos y algoritmos fundamentales (GPO 11)

Casos prueba de la act 2.2

En caso que el índice sea diferente de 0 creamos otro nodo temporal pero este apuntara al puntero first por lo que también apuntaremos al primer nodo de la lista y entonces entraremos en un bucle for en donde inicializamos un entero $i = 0$ e iremos sumando uno y este tiene que ser menor al índice pero -1 ya que necesitamos el nodo previo al índice donde insertaremos nuestro elemento ya que el previo tiene la dirección del siguiente.

Una vez hayamos encontrado nuestro nodo previo, nuestro nodo creado al inicio del metodo next será igual al segundo nodo temporal next ósea que nuestro primer nodo creado next será igual al elemento en el cual recorreremos ese lugar por el nodo creado y después el segundo nodo next será igual al primer nodo creado ósea este será nuestro anterior nodo de la lista.

Fotos de los casos de prueba:

Caso prueba uno:

```
//Inicializamos la lista
FLinked<int> list;
//Caso prueba 1: Añadir nuestro primer elemento, cuando la lista esta vacía.
list.add(1);

//Mostramos la lista inicial
std::cout << "Mostramos la lista inicial: " << std::endl;
list.implist();
```

```
Mostramos la lista inicial:
1
```

Caso prueba dos:

```
//Caso prueba 2: Añadir elemento al inicio de la lista.
std::cout << "Insertamos el elemento 7 al principio de la lista con el metodo add(data, index)" << std::endl;
list.add(7,0);
list.implist();
```

```
Insertamos el elemento 7 al principio de la lista con el metodo add(data, index)
7 1
```

Caso prueba tres:

```
//Caso prueba 3: Añadir elemento en medio o en n posición de la lista.
std::cout << "Insertamos el elemento 5 a la mitad de la lista con metodo add(data, index)" << std::endl;
list.add(5,1);
list.implist();
```

```
Insertamos el elemento 5 a la mitad de la lista con metodo add(data, index)
7 5 1
```

Casos prueba de la act 2.2

Caso prueba cuatro:

```
//Caso prueba 4: Añadir elemento al final de la lista.  
std::cout << "Insertamos el elemento 9 al final de la lista con el metodo add(data)" << std::endl;  
list.add(9);  
list.implist();
```

```
Insertamos el elemento 9 al final de la lista con el metodo add(data)  
7 5 1 9
```

Ambos métodos fueron creadas gracias al apoyo de dos fuentes de video:

CodeBeauty. (2021). Youtube: "How to insert a new node in a linked list in C++?(at the front, at the end, after a given node)". Recuperado de: https://youtu.be/RNMIDj62o_o

Mycodeschool. (2013). Youtube: "Linked list in C/C++ - Insert a node at nth position". Recuperado de: <https://youtu.be/IbvsNF22Ud0>