

## Explicación de Actividad 2.2

### Caso de Prueba

```
LinkedList<string> lista;
lista.create("HOLA", 0);
lista.create("CRAYOLA", 1);
lista.create("AQUI", 2);
lista.create("ESTOY", 3);
// lista.print();
cout << "-----" << endl;
lista.create("BIENVENIDO", 2);
lista.create("AQUI", 2);
cout << "Read: " << lista.read("HOLA") << endl;
lista.update("CRAYOLA", "SEBAS");
// lista.print();
cout << "-----" << endl;

lista.del("AQUI");
lista.del("ESTOY");
// lista.print();
cout << "-----" << endl;
```

### Output

Contenido de la LinkedList

[0] - HOLA  
[1] - CRAYOLA  
[2] - AQUI  
[3] - ESTOY  
-----

Read: 0

Contenido de la LinkedList

[0] - HOLA  
[1] - SEBAS  
[2] - AQUI  
[3] - BIENVENIDO  
[4] - AQUI  
[5] - ESTOY  
-----

Contenido de la LinkedList

[0] - HOLA  
[1] - SEBAS  
[2] - BIENVENIDO  
-----

### Explicación

Lo que realice en el .cpp para probar las funciones fue algo muy corto para asegurar de que funcionaran de manera óptima. Lo primero que hice fue crear un vector de tipo string, llamado "lista". Utilizando este vector, implemente la función "create" para añadir los

primeros cuatro datos(visibles en el output). Mantuve la función “print” para desplegar los resultados dentro del vector cada vez que se realizará un cambio.

Después utilice la función “read” para leer el dato “HOLA”, la que después desplegó su posición de “0”. Con esto implementa la función “create” de nuevo para implementar un valor en medio del vector y validar que las posiciones sean modificadas de manera adecuada. En ese caso implementé “BIENVENIDO” en la posición “2” y nuevamente lo despliega con la función print.

Para terminar, valide la función “del” para eliminar “AQUI” y “ESTOY” del vector. Terminando esto lo despliegue por medio del “print” por última vez para visualizar los datos dentro del vector. Fue un reto realizar el delete ya que se tenía que considerar la eliminación de datos repetidos.En general, pude implementar las 4 funciones pedidas de manera efectiva. Inclusive, fui capaz de implementar funciones alternas para mantener de manera organizada el código.