REPORTE DE PRÁCTICA

| Nombre Completo | Machado Sanchez Javier |
| --- | --- |
| Número de la Unidad a Evaluar | 3 |
| Tema a Evaluar | Lista Enlazada Simple Edades |
| Fecha de Entrega | 14 de Octubre del 2023 |

**Programa Fuente**

| using System;  namespace P27\_Lista\_Enlazada\_Simple\_Edades  {  internal class Program  {  //---------------------------------------------------------------------------------------------  //Declaración de la clase Nodo  public class Node  { //Campo para almacenar la edad  public short age;  //Campo para almacenar la referencia al siguiente nodo  public Node next;  }  //Fin de la clase Nodo  //---------------------------------------------------------------------------------------------  //---------------------------------------------------------------------------------------------  //Declaración de variables globales  public static Node startNode, endNode, tempNode;  public static int length\_list = 0;  //Fin variables globales  //---------------------------------------------------------------------------------------------  //---------------------------------------------------------------------------------------------  //Declaración de método para insertar  public static void AddElement(short element)  { //Condición para validar si el elemento a insertar es el primero  if (startNode == null)  { //Inicializando el primer nodo  startNode = new Node();  startNode.age = element; //Insertando dato  endNode = startNode;  endNode.next = null;  Program.length\_list++; //Contador de elementos totales  }  else  { //Inicializando el nodo auxiliar  tempNode = new Node();  tempNode.age = element; //Insertando dato  endNode.next = tempNode;  endNode = tempNode;  endNode.next = null;  Program.length\_list++; //Contador de elementos totales  }  }  //Fin método para insertar  //---------------------------------------------------------------------------------------------  //---------------------------------------------------------------------------------------------  //Declaración de método para eliminar  public static short DeleteElement()  {  short age\_temp;  if (startNode == endNode)  {  age\_temp = startNode.age;  startNode = null;  endNode = null;  Program.length\_list--;  }  else  {  age\_temp = startNode.age;  tempNode = startNode;  startNode = startNode.next;  Program.length\_list--;  }  return age\_temp;  }  //Fin método para eliminar  //---------------------------------------------------------------------------------------------  //---------------------------------------------------------------------------------------------  //Declaración de método imprimir la lista  public static void PrintElements()  {  if (Program.length\_list > 0)  {  if(Program.length\_list == 1) Console.WriteLine($"\n\n\tEsta es la edad de la lista:");  else Console.WriteLine($"\n\n\tEstas son las {Program.length\_list} edades de la lista:");  tempNode = startNode;  Console.Write("\n\t");  while (tempNode != null)  {  Console.Write($"[ {tempNode.age} ] ");  tempNode = tempNode.next;  }  }  else Console.WriteLine("\n\n\tLa lista se encuentra vacía.");  }  //Fin método para imprimir la lista  //---------------------------------------------------------------------------------------------  //---------------------------------------------------------------------------------------------  //Inicio Main  static void Main(string[] args)  {  //Declaración de variables locales  short age; char opcion\_menu; bool opcion\_bool, answer\_bool = false;  //Despliegue de menú  do  {  Console.Write("\n\t\t>> MENÚ DE OPCIONES <<" +  "\n\n\t1.- Insertar Elemento." +  "\n\t2.- Eliminar Elemento." +  "\n\t3.- Mostrar Elementos." +  "\n\t4.- Salir del Programa." +  "\n\n\tIngrese el número de la opción deseada: ");  //Condición para controlar excepción por formato incorrecto  if (char.TryParse(Console.ReadLine() ,out opcion\_menu))  {  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  //Switch case para controlar las opciones del menú  switch (opcion\_menu)  {  case '1':  {  do  {  Console.Write("\n\t\t>> INSERTANDO ELEMENTO <<" +  "\n\n\tPor favor, ingrese la edad que desea insertar: ");  //Condición para controlar excepción por formato incorrecto  opcion\_bool = short.TryParse(Console.ReadLine(), out age);  if (opcion\_bool)  { //Llamada a método para insertar  AddElement(age);  Console.Write($"\n\n\tSe ha insertado la edad {age} en la lista.");  PrintElements(); //Llamada a método para imprimir    do  { //Ciclo do while para controlar las repuestas del usuario  Console.Write("\n\n\t¿Desea insertar otra edad [S = Sí] [N = No] ?" +  "\n\tIngrese la letra de su respuesta: ");  //Llamada a DoAgain para controlar las respuestas  (answer\_bool, opcion\_menu) = DoAgain(Console.ReadLine(), answer\_bool);  } while (answer\_bool == false);  }  else  {  do  {  Console.Write("\n\n\tLa edad no se ha podido registrar." +  "\n\t¿Desea intentarlo de nuevo [S = Sí] [N = No] ?" +  "\n\tIngrese la letra se su respuesta: ");  //Llamada a DoAgain para controlar las respuestas  (answer\_bool, opcion\_menu) = DoAgain(Console.ReadLine(), answer\_bool);  } while (answer\_bool == false);  }  } while (opcion\_menu != 'N');  }  break;  case '2':  {  do  { //Condición para validar que la lista contenga elementos  if (length\_list > 0)  {  Console.Write("\n\t\t>> ELIMINANDO ELEMENTO <<");  PrintElements();  do  {  Console.Write("\n\n\t¿Desea eliminar una edad [S = Sí] [N = No] ?" +  "\n\tIngrese la letra de su respuesta: ");  //Llamada a DoAgain para controlar las respuestas  (answer\_bool, opcion\_menu) = DoAgain(Console.ReadLine(), answer\_bool);  } while (answer\_bool == false);  if (opcion\_menu == 'S')  { //Llamada al método para eliminar una edad  age = DeleteElement();  Console.Write( "\n\t\t>> ELIMINANDO ELEMENTO <<" +  $"\n\n\tSe ha eliminado la edad {age} correctamente.");  //Llamada a método para imprimir  PrintElements();  Console.Write("\n\n\tPresione la tecla <INTRO> para regresar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  else  {  Console.Write("\n\t\t>> NO HAY NADA QUE ELIMINAR <<");  Console.Write("\n\n\tNo se eliminará ninguna edad." +  "\n\tPresione la tecla <INTRO> para regresar al menú...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  }  else  {  Console.Write("\n\t\t>> NO HAY NADA QUE ELIMINAR <<");  Console.WriteLine("\n\n\tLa lista se encuentra vacía.");  Console.Write("\n\tPresione la tecla <INTRO> para regresar al menú...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  opcion\_menu = 'N';  }  } while (opcion\_menu != 'N');  }  break;  case '3':  {  if (length\_list > 0)  {  Console.Write("\n\t\t>> MOSTRANDO ELEMENTOS <<");  //Llamada a método para imprimir  PrintElements();  Console.Write("\n\n\tPresione la tecla <INTRO> para regresar al menú...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  else  {  Console.Write("\n\t\t>> NO HAY NADA QUE MOSTRAR <<");  //Llamada a método para imprimir  PrintElements();  Console.Write("\n\n\tPresione la tecla <INTRO> para regresar al menú...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  opcion\_menu = 'N';  }  }  break;  case '4':  {  Console.Write("\n\t\t:. SALIDA :." +  "\n\n\tGracias por utilizar nuestro programa" +  "\n\n\tPresione la tecla <Esc> para salir...");  while (Console.ReadKey().Key != ConsoleKey.Escape) ;  }  break;  default:  {  Console.Write("\n\t\t.: OPCIÓN INVALIDA :." +  "\n\n\tPor favor ingrese una opción existente" +  "\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  }  }  else  {  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  Console.Write("\n\t\t>> ALGO HA SALIDO MAL <<" +  "\n\n\tPor favor ingrese solamente un número." +  "\n\n\tPresione la tecla <INTRO> para regresar al menú...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  } while (opcion\_menu != '4');  }  //Fin Main  //---------------------------------------------------------------------------------------------  //Inicio método para preguntar hacer de nuevo  /\*Definición de método DoAgain(Hacer de nuevo) para hacer todo el procesamiento  \* de las respuestas leídas.  \* Answer bool es true cuando la respuesta es 'S' o 'N', por lo que se puede salir del  \* do while donde se hace la pregunta. En caso de ser cualquier otra respuesta se avisa  \* al usuario que la opción no existe y se vuelve a imprimir la pregunta  \*/  public static (bool, char) DoAgain(string read\_line, bool answer\_bool)  {  char opcion\_menu;  if (char.TryParse(read\_line, out opcion\_menu))  {  opcion\_menu = char.ToUpper(opcion\_menu);  if (opcion\_menu == 'S')  {  Console.Write("\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  return (answer\_bool = true, opcion\_menu);  }  else if (opcion\_menu == 'N')  {  Console.Write("\n\tPresione la tecla <INTRO> para regresar al menú...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  return (answer\_bool = true, opcion\_menu);  }  else  {  Console.Write("\n\n\tPor favor, ingrese una respuesta válida" +  "\n\tPresione la tecla <INTRO> para intentarlo de nuevo...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  return (answer\_bool = false, opcion\_menu);  }  }  else  {  Console.Write("\n\n\tPor favor, ingrese una respuesta válida" +  "\n\tPresione la tecla <INTRO> para intentarlo de nuevo...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  return (answer\_bool = false, opcion\_menu);  }  }  //Fin método para preguntar hacer de nuevo  //---------------------------------------------------------------------------------------------  }  } |
| --- |

**Ventana de Entrada de datos**

|  |
| --- |

**Ventana de Resultados**

|  |
| --- |

**EXPLICA BREVEMENTE QUÉ TE PARECIÓ EL PROBLEMA, QUE SE TE DIFICULTO Y COMO LO SOLUCIONASTE.**

| Nunca había visto este tema de las listas enlazadas así que fue un poco difícil entender su lógica y su utilidad en los programas. Gracias a videos de youtube y explicaciones con gráficos y esquemas es que pude entender esta parte del flujo de los datos en los nodos y también los beneficios que tiene trabajar en este tipo de listas.  Como casi todo el código se explicó en clase y también en un vídeo de youtube de la profe, no tuve mucho problema para implementar las listas enlazadas en c#, me enfoqué más en controlar las respuestas del usuario y evitar que el programa se rompa tan fácil o por algún error de dedo del usuario. |
| --- |