REPORTE DE PRÁCTICA

| Nombre Completo | Machado Sanchez Javier |
| --- | --- |
| Número de la Unidad a Evaluar | 6 |
| Tema a Evaluar | Método de Hash Mitad al Cuadrado |
| Fecha de Entrega | 20 de Noviembre del 2023 |

**Programa Fuente**

| using System;  namespace P46\_Metodo\_de\_Hash\_Mitad\_al\_cuadrado  {  internal class Program  {  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Inicio Método Buscar  public static bool Buscar(int[] NewArray, int key, ref int position)  {  //Inicio Declaración de variables locales  bool flag = false;  int index, anti\_collision;  double key\_pow;  string string\_key\_pow;  position = 0;  //Fin Declaración de variables locales  //Inicio Bloque de instrucciones  //Generando índice con mitad al cuadrado  key\_pow = Math.Pow(key, 2) + 1;  string\_key\_pow = key\_pow.ToString();  index = (string\_key\_pow.Length + 1) / 2;  if (NewArray[index] == key) { flag = true; position = index; }  else  {  anti\_collision = index + 1;  while(anti\_collision < NewArray.Length && NewArray[anti\_collision] != key &&  NewArray[anti\_collision] != 0 && anti\_collision != index)  {  anti\_collision++;  if (anti\_collision >= NewArray.Length) anti\_collision = 0;  }  if (NewArray[anti\_collision] == key) { flag = true; position = anti\_collision; }  else { flag = false; position = -1; }  }  return flag;  //Fin Bloque de instrucciones  }  //Fin Método Buscar  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Inicio Método Mostrar  public static void Mostrar(int[] Arreglo)  {  for (int i = 0; i < Arreglo.Length; i++)  {  if(i == 0) Console.Write($"\n\t[{i + 1}] - {Arreglo[i]} , ");  else if(i % 5 == 0) Console.Write($"\n\t[{i + 1}] - {Arreglo[i]} , ");  else if(i == Arreglo.Length - 1) Console.Write($"[{i + 1}] - {Arreglo[i]}. ");  else Console.Write($"[{i + 1}] - {Arreglo[i]} , ");  }  }  //Fin Método Mostrar  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Inicio Método Hashing  public static int[] Hashing(int[] OriginalArray, int[] NewArray)  {  //Inicio declaración de variables locales  int index, anti\_collision;  double key\_pow;  string string\_key\_pow;  //Fin declaración de variables locales  //Inicio bloque de instrucciones  for (int i = 0; i < OriginalArray.Length; i++)  {  key\_pow = Math.Pow(OriginalArray[i], 2) + 1;  string\_key\_pow = key\_pow.ToString();  index = (string\_key\_pow.Length + 1) / 2;    while (NewArray[index] != 0)  {  anti\_collision = index + 1;  if (anti\_collision >= OriginalArray.Length) index = 0;  else index = anti\_collision;  }  NewArray[index] = OriginalArray[i];  }  return NewArray;  //Fin bloque de instrucciones  }  //Fin Método Hashing  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  static void Main(string[] args)  {  Console.Title = "P46 Método Hash Mitad al Cuadrado";  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Declaración de variables  char option\_menu;  int matricula;  int[] MatriculasArreglo = new int[50];  int[] AuxMatriculasArreglo = new int[50];  int IndexElement = 0;  Random random = new Random();  //Fin Declaración de variables  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //Despliegue de menú  do  {  Console.Write("\n\t\t.: MENÚ OPCIONES :." +  "\n\n\t[1] - Insertar Matrículas." +  "\n\n\t[2] - Buscar Matrículas." +  "\n\n\t[3] - Mostrar Matrículas." +  "\n\n\t[4] - Salida del Programa." +  "\n\n\tIngrese el número de la opción deseada: ");  if (char.TryParse(Console.ReadLine(), out option\_menu))  {  option\_menu= char.ToUpper(option\_menu);  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  //Control de opciones del menú  switch (option\_menu)  {  case '1':  {  //Captura de datos  Console.Write("\n\t\t.: INSERTAR MATRÍCULAS :." +  $"\n\n\tA continuación se insertarán las {MatriculasArreglo.Length} matrículas en el arreglo aleatoriamente: ");  for (int i = 0; i < MatriculasArreglo.Length; i++)  {  matricula = random.Next(1, 1000);  Console.Write($"\n\n\t{i + 1}.- Ingrese la matrícula del alumno: {matricula}");  MatriculasArreglo[i] = matricula;  Console.Write($"\n\tSe ha ingresado la matrícula {MatriculasArreglo[i]} correctamente");  }  //Llamada a método para generar los nuevos índices  AuxMatriculasArreglo = Hashing(MatriculasArreglo, AuxMatriculasArreglo);  Console.Write("\n\n\tSe han ingresado las matrículas correctamente" +  "\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  case '2':  {  //Captura de datos  Console.Write("\n\t\t.: BUSCAR MATRÍCULAS :." +  "\n\n\tIngrese el sueldo del empleado a buscar: ");  matricula = int.Parse(Console.ReadLine());  //Procedimiento de búsqueda  if (Buscar(AuxMatriculasArreglo, matricula ,ref IndexElement)) Console.WriteLine($"\n\n\tLa matrícula {matricula} se encuentra en la posición {IndexElement + 1} de la lista");  else Console.WriteLine($"\n\n\tLa matrícula {matricula} no se encuentra en la lista");  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  case '3':  {  //Captura de datos  Console.Write("\n\t\t.: MOSTRAR MATRÍCULAS :." +  $"\n\n\tA continuación se mostrarán las {MatriculasArreglo.Length} matrículas del arreglo: ");  Console.WriteLine("\n\n\tArreglo original:");  Mostrar(MatriculasArreglo);  Console.WriteLine("\n\n\tArreglo con hashing:");  Mostrar(AuxMatriculasArreglo);  Console.Write("\n\n\tSe han mostrado los sueldos correctamente" +  "\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  break;  case '4':  {  //Caso 4. Salida del programa  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:  {  //En caso de ingresar una opción inválida  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.: 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();  }  } while (option\_menu != '4');  }  }  } |
| --- |

**Ventana de Entrada de datos**

|  |
| --- |

**Ventana de Resultados**

|  |
| --- |

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

| Un tema muy interesante y muy ingenioso para obtener nuevos índices para el arreglo. Tuve problemas para realizar este programa debido a que la posición no me la arrojaba correctamente. Al final noté que mi función estaba retornando el valor de “Index” cuando en realidad debería de retornar el valor de index sin colisiones “anti\_collision”.  Tardé como unos 30 minutos en darme cuenta pero al final pude solucionarlo. |
| --- |