REPORTE DE PRÁCTICA

| Nombre Completo | Machado Sanchez Javier |
| --- | --- |
| Número de la Unidad a Evaluar | 3 |
| Tema a Evaluar | Manejando Pilas Dinámicas |
| Fecha de Entrega | 03 de Octubre del 2023 |

**Programa Fuente**

| using System;  using System.Collections.Generic;  namespace P21\_Manejando\_Pilas\_Dinamicas  {  internal class Program  {  static void Main(string[] args)  {  Console.Title = "P21 Manejando Pilas Dinámicas";  //Declaración de pila dinamica  Stack<string> PilaMascotas = new Stack<string>(40);  //Declaración de variables para menú  int opcion\_menu;  //Declaración de variables para procedimientos  string nombre\_mascota;  //Despliegue de menú  do  {  Console.Write("\n\t\t.: MENÚ OPCIONES :." +  "\n\n\t[1] - Insertar Mascotas." +  "\n\n\t[2] - Eliminar Mascotas." +  "\n\n\t[3] - Buscar Mascotas." +  "\n\n\t[4] - Mostrar Mascotas." +  "\n\n\t[5] - Salida del Programa." +  "\n\n\tIngrese el número de la opción deseada: ");    opcion\_menu = int.Parse(Console.ReadLine());  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  //Control de opciones del menú  switch (opcion\_menu)  {  case 1:  { //Condición para validar estado de la pila  do  {  if (PilaMascotas.Count < 40)  {  Console.Clear();  //Captura de datos  Console.Write("\n\t\t.: INSERTAR MASCOTAS :." +  "\n\n\tIngrese el nombre de la mascota: ");  nombre\_mascota = Console.ReadLine();  //Procedimiento para insertar el nombre  PilaMascotas.Push(nombre\_mascota);  //Pregunta para ingresar otro nombre  Console.Write($"\n\n\tSe ha insertado la mascota {nombre\_mascota}" +  $"\n\n\t¿Desea ingresar otra mascota [1.Sí][2.No]? " +  $"\n\n\tIngrese el número de la opción: ");  opcion\_menu = int.Parse(Console.ReadLine());  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  else  { //Pila llena  Console.WriteLine("\n\n\t.: LA PILA ESTÁ LLENA :.");  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  opcion\_menu =2;  }  } while (opcion\_menu == 1);  }  break;  case 2:  {  //Condición para validar estado de la pila  do  {  if (PilaMascotas.Count > 0)  {  Console.Clear();  //Captura de datos  Console.Write("\n\t\t.: ELIMINAR MASCOTAS :." +  "\n\n\tIngrese el nombre de la mascota a eliminar: ");  nombre\_mascota = Console.ReadLine();  //Condición para válidar que el nombre se encuentre en la pila  if (PilaMascotas.Contains(nombre\_mascota) == true)  {  //Eliminación de mascota y pregunta para hacer otra iteración  Console.Write($"\n\n\tSe ha eliminado la mascota {PilaMascotas.Pop()}");  }  else  {  Console.WriteLine($"\n\n\tEl nombre {nombre\_mascota} no se encuentra en la pila");  }  Console.Write($"\n\n\t¿Desea eliminar otra mascota [1.Sí][2.No]? " +  $"\n\n\tIngrese el número de la opción: ");  opcion\_menu = int.Parse(Console.ReadLine());  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  else  { //Pila Vacía  Console.WriteLine("\n\n\t.: LA PILA ESTÁ VACÍA :.");  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  opcion\_menu = 2;  }  } while (opcion\_menu == 1);    }  break;  case 3:  {  do  {  if (PilaMascotas.Count > 0)  {  Console.Clear();  //Captura de datos  Console.Write("\n\t\t.: BUSCAR MASCOTAS :." +  "\n\tIngrese el nombre de la mascota a buscar: ");  nombre\_mascota = Console.ReadLine();  //Procedimiento de busqueda  if (PilaMascotas.Contains(nombre\_mascota) == true) Console.WriteLine($"\n\n\tEl nombre {nombre\_mascota} sí se encuentra en la pila");  else Console.WriteLine($"\n\n\tEl nombre {nombre\_mascota} no se encuentra en la pila");  //Pregunta para hacer otra iteración  Console.Write("\n\n\t¿Desea buscar otra mascota [1.Sí][2.No]? " +  "\n\n\tIngrese el número de la opción: ");  opcion\_menu = int.Parse(Console.ReadLine());  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  else  { //Pila Vacía  Console.WriteLine("\n\n\t.: LA PILA ESTÁ VACÍA :.");  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  opcion\_menu = 2;  }  } while (opcion\_menu == 1);  }  break;  case 4:  {  //Condición para validar estado de la pila  if (PilaMascotas.Count > 0)  {  //Captura de datos  Console.Write("\n\t\t.: MOSTRAR MASCOTAS :." +  "\n\n\tSe han encontrado las siguientes mascotas: \n");  //Procedimiento para mostrarlos  foreach(string nombre in PilaMascotas)  {  Console.WriteLine($"\n\t{nombre}");  }  //Mensaje para regresar al menú  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  else  { //Pila Vacía  Console.WriteLine("\n\n\t.: LA PILA ESTÁ VACÍA :.");  Console.Write("\n\n\tPresione la tecla <INTRO> para continuar...");  while (Console.ReadKey().Key != ConsoleKey.Enter) ; Console.Clear();  }  }  break;  case 5:  {  //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;  }  } while (opcion\_menu != 5);  }  }  } |
| --- |

**Ventana de Entrada de datos**

|  |
| --- |

**Ventana de Resultados**

|  |
| --- |

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

| No tuve mayores complicaciones, trabajar con pilas dinámicas es mucho más sencillo que con las estáticas debido a que los métodos ya están programados en la librería. Solamente tuve conflicto en la parte de mostrar todos los elementos de la cola, pues con peek solamente obtienes el último ingresado -que es muy útil en ciertos procedimientos- y no es posible mostrar todos los elementos; al menos yo no encontré manera. Solamente pude lograrlo por medio de un foreach, no sé si esté del todo correcto pero al menos funcionó |
| --- |