|  |
| --- |
| **Ingeniero en Computación**  **2018-2**  **Materia:** Algoritmos y Estructuras de Datos / Clave 12098  **Alumno:** Carlos Daniel Orozco Bravo  **Matrícula:** 351285  **Maestro:** Pedro Núñez Yépiz  **Práctica No.:** 2  **Tema – Unidad:** Pilas  **Ensenada Baja California a 8 de diciembre del 2018** |

|  |
| --- |
| 1. **INTRODUCCIÓN** |
| En esta practica desarrollaremos un programa de computadora en lenguaje c, capaz de utilizar arreglos, para poder manipular una pila. |

|  |
| --- |
| 1. **COMPETENCIA** |
| En esta práctica se busca que el alumno desarrolle la competencia, de poder hacer y manipular una pila, desarrollado en un lenguaje c, utilizando arreglos. |

|  |
| --- |
| 1. **FUNDAMENTOS** |
| **Definición de pila**  Una pila es un tipo especial de lista abierta en la que sólo se pueden insertar y eliminar nodos en uno de los extremos de la lista. Estas operaciones se conocen como "push" y "pop", respectivamente "empujar" y "tirar". Además, las escrituras de datos siempre son inserciones de nodos, y las lecturas siempre eliminan el nodo leído.  Estas características implican un comportamiento de lista LIFO (Last In First Out), el último en entrar es el primero en salir.  El símil del que deriva el nombre de la estructura es una pila de platos. Sólo es posible añadir platos en la parte superior de la pila, y sólo pueden tomarse del mismo extremo.  El nodo típico para construir pilas es el mismo que vimos en el capítulo anterior para la construcción de listas:  struct nodo {  int dato;  struct nodo \*siguiente;  };  **Declaraciones de tipos para manejar pilas en C**  Los tipos que definiremos normalmente para manejar pilas serán casi los mismos que para manejar listas, tan sólo cambiaremos algunos nombres:  typedef struct \_nodo {  int dato;  struct \_nodo \*siguiente;  } tipoNodo;    typedef tipoNodo \*pNodo;  typedef tipoNodo \*Pila;  tipoNodo es el tipo para declarar nodos, evidentemente.  pNodo es el tipo para declarar punteros a un nodo.  Pila es el tipo para declarar pilas.    Es evidente, a la vista del gráfico, que una pila es una lista abierta. Así que sigue siendo muy importante que nuestro programa nunca pierda el valor del puntero al primer elemento, igual que pasa con las listas abiertas.  Teniendo en cuenta que las inserciones y borrados en una pila se hacen siempre en un extremo, lo que consideramos como el primer elemento de la lista es en realidad el último elemento de la pila. |

|  |
| --- |
| 1. **PROCEDIMIENTO** |
| Hacer un programa de pilas usando estructuras estaticas (arreglos). |

|  |
| --- |
| 1. **RESULTADOS Y CONCLUSIONES** |
| **Conclusiones:**  Se termino desarrollando un programa capaz de simular una pila utilizando arreglos. |

|  |
| --- |
| 1. **ANEXOS** |
| **Menú.**    **Con el 1 se hace push a la pila. Se agrego el 5, 6, 3, y 8 en ese orden.**      **Con el 2 se hace pop a la pila.**            **Con el 3 se entra al menú de la simulación de estacionamiento.**    **Con el 1 se mete un auto.**        **Con el 2 se saca un auto.** |

|  |
| --- |
| 1. **REFERENCIAS** |
| **JAVA SWING,**  Marc Loy, Robert Eckstein, Dave Wood, James Elliott, Brian Cole.  second edition.  O’Reilly. 2002.  **Como programar en C/C++**  H.M. Deitel/ P.J. Deitel  Segunda edición  Editorial: Prentice Hall.  ISBN:9688804711  **ESTRUCTURAS DE DATOS CON C Y C++**  YEDIYAH LANGSAM; MOSHE AUGENSTEIN  Segunda edición  Editorial: Prentice Hall.  ISBN: 9789688807989  **Ligas de Referencia:**  http://c.conclase.net/edd/index.php?cap=002#inicio |