
Laboratorio 11

Punteros, POO y Pilas

1. Competencias

1.1. Competencias del curso

Conoce, comprende e implementa programas usando punteros y POO del lenguaje de programación C++.

1.2. Competencia del laboratorio

Conoce, comprende e implementa programas usando punteros y POO del lenguaje de programación C++.

2. Equipos y Materiales

- Un computador.
- IDE para C++.
- Compilador para C++.

3. Marco Teórico

3.1. Punteros con pilas

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.

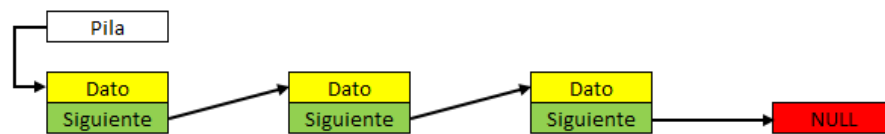
Estas características implican un comportamiento de lista LIFO (Last In First Out), el último en entrar es el primero en salir.

La analogía de la 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.

Los tipos que definiremos normalmente para manejar pilas serán casi los mismos que para manejar listas.

Es evidente, a la vista del gráfico, que una pila es una lista enlazada. Así que sigue siendo muy importante que nuestro programa nunca pierda el valor del puntero al primer elemento, igual que pasa con las listas enlazadas.

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



3.2. Operaciones básicas con pilas

Las pilas tienen un conjunto de operaciones muy limitado, sólo permiten las operaciones de "push" y "pop":

Push: Añadir un elemento al inicio de la pila.

Pop: Leer y eliminar un elemento del inicio de la pila.

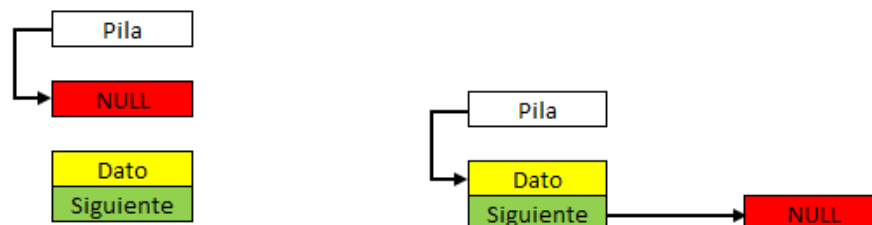
1. Push, insertar elementos

a) Push en una pila vacía

Partiremos de que ya tenemos el nodo a insertar y, por supuesto un puntero que apunte a él, además el puntero a la pila valdrá NULL.

El proceso es muy simple, bastará con que:

1. *Nodo->siguiete apunte a NULL.*
2. *Pila apunte a nodo.*



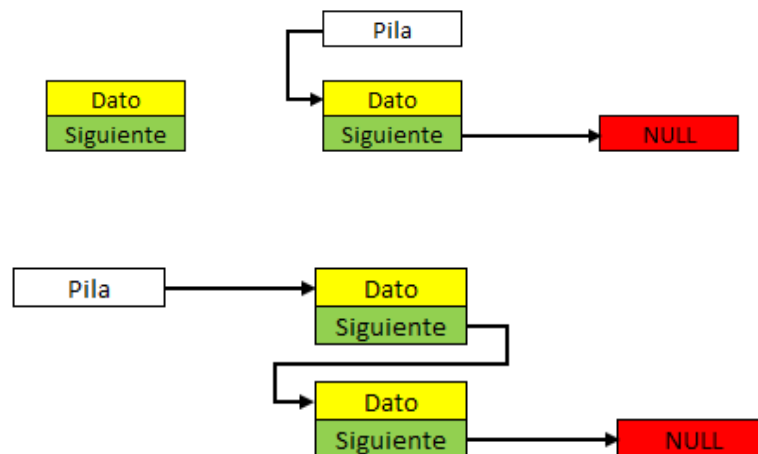
b) Push en una pila no vacía

Podemos considerar el caso anterior como un caso particular de éste, la única diferencia es que podemos y debemos trabajar con una pila vacía como con una pila normal.

De nuevo partiremos de un nodo a insertar, con un puntero que apunte a él, y de una pila, en este caso no vacía.

El proceso sigue siendo muy sencillo:

1. Hacemos que nodo->siguiente apunte a Pila.
2. Hacemos que Pila apunte a nodo.



2. Pop, leer y eliminar un elemento

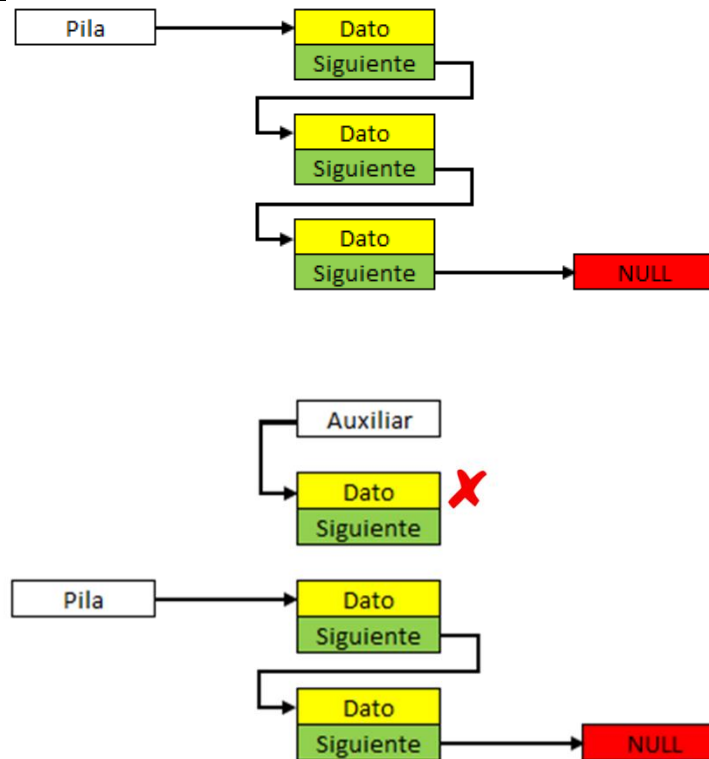
a) Eliminar un elemento de pila

Ahora sólo existe un caso posible, ya que sólo podemos leer desde un extremo de la pila.

Partiremos de una pila con uno o más nodos, y usaremos un puntero auxiliar, nodo:

1. Hacemos que auxiliar apunte al primer elemento de la pila, es decir a Pila.
2. Asignamos a Pila la dirección del segundo nodo de la pila: Pila->siguiente.
3. Guardamos el contenido del auxiliar para devolverlo como retorno, recuerda que la operación pop equivale a leer y borrar.
4. Liberamos la memoria asignada al primer nodo, el que queremos eliminar.

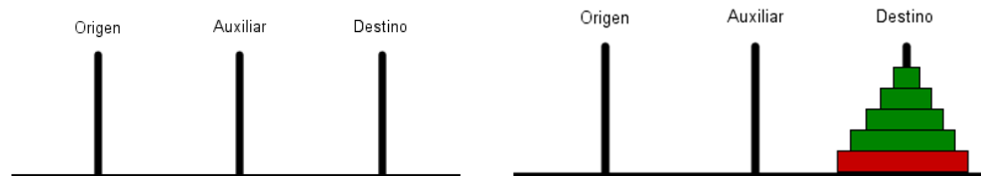
Si la pila sólo tiene un nodo, el proceso sigue siendo válido, ya que el valor de Pila->siguiente es NULL, y después de eliminar el último nodo la pila quedará vacía, y el valor de Pila será NULL.



4. Ejercicios

Resolver los siguientes ejercicios planteados:

1. Defina una Pila que permita insertar elementos utilizando clases.
2. Sobre el ejercicio anterior, adecue el programa para eliminar elementos de una Pila.
3. Implemente un algoritmo para buscar elementos de la Pila.
4. Escribir un programa que dé la solución al problema de las Torres de Hanoi para N discos, utilizando pilas, las cuales representen cada uno de los postes:



5. Entregables

Al final estudiante deberá:

1. Compactar el código elaborado y subirlo al aula virtual de trabajo. Agregue sus datos personales como comentario en cada archivo de código elaborado.
2. Elaborar un documento que incluya tanto el código como capturas de pantalla de la ejecución del programa. Este documento debe de estar en formato PDF.
3. El nombre del archivo (comprimido como el documento PDF), será su LAB11_GRUPO_A/B/C_CUI_1erNOMBRE_1erAPELLIDO.

(Ejemplo: LAB11_GRUPO_A_2022123_PEDRO_VASQUEZ).

4. Debe remitir el documento ejecutable con el siguiente formato:

LAB11_GRUPO_A/B/C_CUI_EJECUTABLE_1erNOMBRE_1erAPELLIDO

(Ejemplo: LAB11_GRUPO_A_EJECUTABLE_2022123_PEDRO_VASQUEZ).

En caso de encontrarse trabajos similares, los alumnos involucrados no tendrán evaluación y serán sujetos a sanción.