



## Actividad 2: Algoritmo push & pop

**Alumna: Hernández Vázquez Daniela**

**Profesor: M.I. Marco Antonio Martínez Quintana**

**Fecha: 15/06/2021**

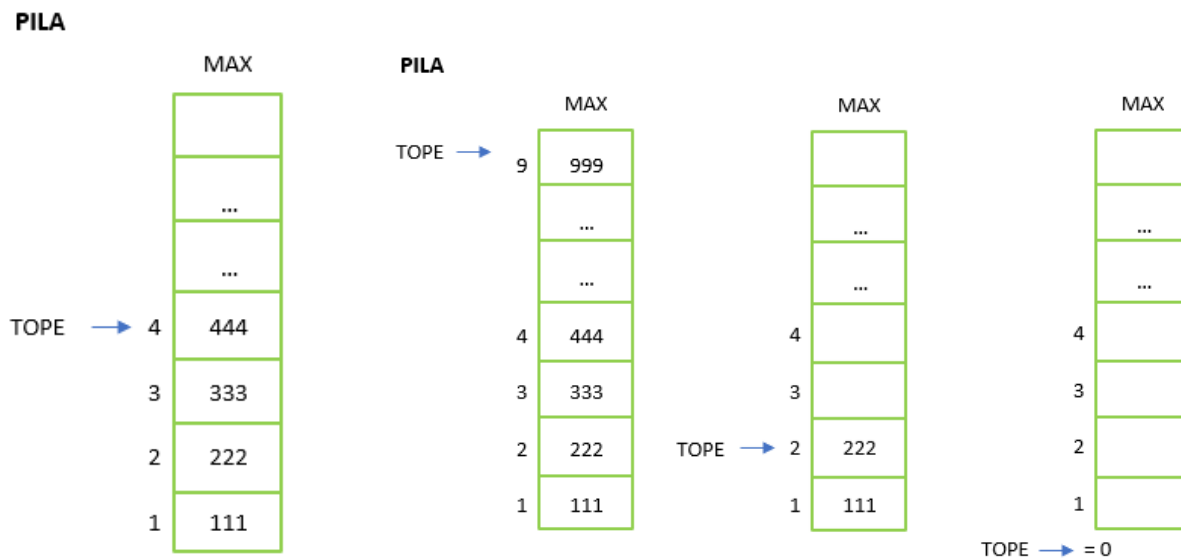
**2021-2**



## Actividades:

- Esta actividad consiste en realizar 2 algoritmos, uno correspondiente a la operación push y otro a la operación pop.

Anteriormente conocimos el concepto básico de pila la cual representa una estructura lineal de datos en que se puede agregar o quitar elementos únicamente por uno de los dos extremos conocido como tope (fig. 1). Es una estructura LIFO (last input, first output), esta tiene como característica que ultimo elemento que entra es el primero que sale.



Como sabemos la estructura de la pila no está definida como tal en los lenguajes de programación. En este caso buscaremos un algoritmo para las principales funciones de las pilas: Insertar un elemento *-push-*, Eliminar *-pop-*, Pila\_vacía, Pila\_llena.

### Algoritmo Pila\_vacía

*\*/Este algoritmo verifica si una estructura tipo pila -PILA- está vacía, asignando a Band el valor de verdad. La pila se implementa en un arreglo unidimensional. -TOPE- es un parámetro de tipo entero. BAND es un parámetro de tipo booleano /\**

Datos de entrada: Números Enteros

Datos de salida: Números Enteros

1. Inicio
2. Si (TOPE = 0)
  - Asignar VERDADERO a BAND //la pila está vacía
  - Si no
    - Asignar FALSO a BAND //la pila no está vacía
3. Fin

### Algoritmo Pila\_llena

\*/Este algoritmo verifica si una estructura tipo pila -PILA- está llena, asignando a BAND el valor de verdad. La pila se implementa en un arreglo unidimensional de MAX elementos. -TOPE- es un parámetro de tipo entero. BAND es un parámetro de tipo booleano /\*

Datos de entrada: Números Enteros

Datos de salida: Números Enteros

1. Inicio
2. Si (TOPE = MAX)
  - Asignar VERDADERO a BAND //la pila está llena
  - Si no
  - Asignar FALSO a BAND //la pila no está llena
3. Fin

### Algoritmo Push

\*/Este algoritmo agrega un elemento -DATO- a una estructura tipo pila -PILA-, mientras que esta no esté llena. Actualiza el valor -TOPE- de tipo entero, -MAX- es el número máximo de elementos que puede almacenar la pila /\*

Datos de entrada: Números Enteros

Datos de salida: Números Enteros

1. Inicio
2. Si (BAND = Verdadero)
  - imprimir" Desbordamiento, Pila llena"
  - Si no
  - Almacenar DATO en PILA[TOPE] y TOPE+1=TOPE
3. Fin

### Algoritmo Pop

\*/Este algoritmo quita un elemento -DATO- a una estructura tipo pila -PILA-, mientras que esta no esté vacía. Actualiza el valor -TOPE- de tipo entero /\*

Datos de entrada: Números Enteros

Datos de salida: Números Enteros

1. Inicio
2. Si (BAND = Verdadero)
  - imprimir" Subdesbordamiento, Pila vacía"
  - Si no
  - Hacer PILA[TOPE]=DATO y TOPE-1=TOPE
3. Fin

En este tipo de estructura solamente se tiene acceso a la parte superior de la pila, es decir, al último objeto apilado (denominado TOS, Top of Stack en inglés).

### *Conclusión:*

La estructura de la pila como lo menciona el principio del algoritmo que nos muestra las partes de esta están muy asociadas a las listas y a los arreglos, en este caso unidimensionales, con la diferencia de que en este caso solo se tiene acceso al elemento de la parte superior.

Anónimo. (28 abr 2021). Pila (informática). (10/06/21, 18:43), de wikipedia.org Sitio web: [Pila \(informática\) - Wikipedia, la enciclopedia libre](#)

Anónimo. (2008). Pilas y Colas. (10/06/21, 18:45), de uaeh.edu Sitio web: [Pilas y Colas \(uaeh.edu.mx\)](#)