

Fundamentos de Programación
Taller de Repaso para el
Parcial 2

Profesor: Daniel Wilches Maradei [dmaradei@eisc.univalle.edu.co]

Monitor: Christian Alejandro Trujillo [catruero@univalle.edu.co]

Este taller (y el parcial # 2) solo podrá ser entregado por medio del Campus Virtual. Si usted olvidó su contraseña, debe entrar a la página web del Campus y escribir a la persona encargada de retornar las contraseñas.

Este taller NO puede ser enviado por correo (tampoco el parcial)

Fecha límite de entrega: Domingo 3 de junio

Pilas (Stacks)

Tiempo estimado para finalizar: 1 hora

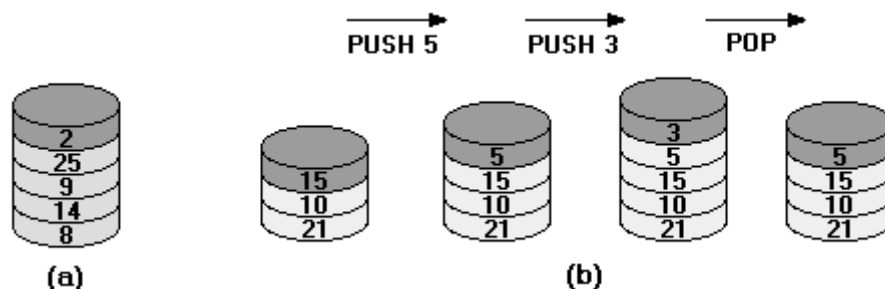
Enunciado:

Una **Pila** es una estructura de almacenamiento de datos que contiene un conjunto de elementos ordenados según el tiempo que llevan dentro de la Pila, de modo que el ultimo elemento insertado este antes que los demás (contrario al orden de inserción).

Además de una **Pila** solo puede retirarse un elemento a la vez, y siempre debe ser el elemento que esstá de primero en la Pila (es decir, el último que fue insertado), es decir, que esté en el tope.

En la figura (a) se muestra una **Pila** que contiene los elementos 2,25,9,14,8. El elemento sombreado es el que está en el tope de la **Pila**. Según nuestra definición previa, si un elemento es ingresado a esta pila, debería quedar sobre este elemento.

En la figura (b) se muestra una **Pila** que contiene los elementos 15,10,21 y se muestra qué sucede cuando se le agregan (push) y se le retiran elementos (pop). En la primer transición, se le agrega el número 5, y ya que el 15 estaba en el tope de la **Pila**, el 5 queda sobre el 15, resultado en la **Pila** que contiene los elementos 5,15,10,21. De similar manera para la segunda transición, donde al agregar 3 a la nueva **Pila** quedan los elementos: 3,5,15,10,21.



Cuando a esta nueva **Pila** se le retira el último número (se le hace POP), entonces queda convertida en la **Pila** 5,15,10,21, es decir, al hacer POP se le retira el número que estaba en el tope de la **Pila**: el 3.

1. Valor de este punto: 50%

Usted debe realizar de manera correcta, con base en la estructura definida abajo y al enunciado de este taller, la implementación de estas 3 funciones: **Push**, **Pop** y **CrearPilaVacía**. Definidas según los siguientes contratos:

```
; La siguiente estructura representa una Pila como las
; definidas arriba.
; El parámetro valor es un número
; El parámetro subpila es una Pila que contiene el resto de los
; elementos
(define-struct Pila (valor subpila))

; Esta función inserta un valor en el tope de la Pila y retorna la
; Pila resultante de esta operación.
; Push : valor Pila -> Pila

; Esta función retira el valor que está en el tope de la Pila y retorna
; una lista con dos elementos: el primero es el elemento retirado,
; y el segundo es la Pila que resulta de esta operación.
; Pila resultante de esta operación.
; Pop : Pila -> (list número Pila)

; Esta función retorna una Pila vacía.
; CrearPilaVacía : -> Pila
```

2. Valor de este punto: 50%

Los siguientes son ejemplos del uso de la **Pila**, estos deben funcionar a la perfección:

```
(define miPila (CrearPilaVacía))
; miPila debe quedar con algún valor que usted reconozca como
; una pila vacía. Por ejemplo: empty

(push 21 miPila)
; Debe retornar (make-Pila 21 empty), o en vez de empty, el valor que
; usted haya escogido.

(push 10 (push 21 miPila))
; Debe retornar (make-Pila 10 (make-Pila 21 empty))

(push 15 (push 10 (push 21 miPila)))
; Debe retornar (make-Pila 15 (make-Pila 10 (make-Pila 21 empty)))

(pop (push 15 (push 10 (push 21 miPila))))
; Debe retornar (list 15 (make-Pila 10 (make-Pila 21 empty)))

(cdr (pop (push 15 (push 10 (push 21 miPila)))))
; Debe retornar (make-Pila 10 (make-Pila 21 empty))
```



```
(define miPila2
  (cdr (pop (push 15 (push 10 (push 21 miPila))))))
; miPila2 debe quedar con el valor
; (make-Pila 10 (make-Pila 21 empty))

(pop miPila2)
; Debe retornar (list 10 (make-Pila 21 empty))
```