



Universidad Tecnológica Nacional
Facultad Regional Córdoba

Algoritmos y Estructuras de Datos

Ficha Nro. 19

Pilas y Colas

Temario

- Estructura de datos nativas y abstractas
- Pilas
- Colas

Introducción

- Estructuras de datos “nativas”: son estructuras de datos provistas por los lenguajes de programación como las tuplas, rangos, listas entre otros.
- Estructuras o “tipos abstractos”: se refiere a estructuras de datos que el programador necesita y no esta disponible en su lenguaje de trabajo → entonces el programador debe implementarla, combinando elementos que están disponible en el lenguaje. Por ejemplo la definición de un tipo Alumno o Libro.

Abstracción

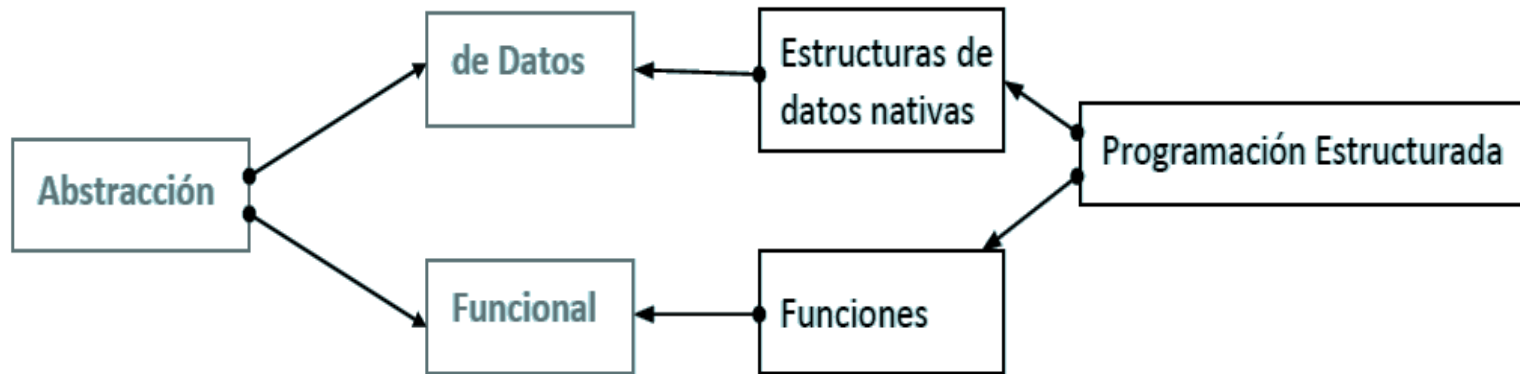
- La creación de un nuevo tipo abstracto se conoce como **implementación del tipo abstracto** e implica realizar un mecanismo conceptual denominado abstracción.
- **Abstracción:** es un proceso mediante el cual se intenta captar y aislar la información y el comportamiento esencial de una entidad u objeto del dominio de un problema.
- Este mecanismo de abstracción se descompone en dos aspectos:
 - **Abstracción de datos:** identificación de datos significativos de una entidad.
 - **Abstracción Funcional:** identificación de procesos relevantes.

Mecanismo del proceso de abstracción

Figura 1: Esquema de un proceso de abstracción.

Ejemplo: Programa para control de préstamos de libros en una biblioteca...

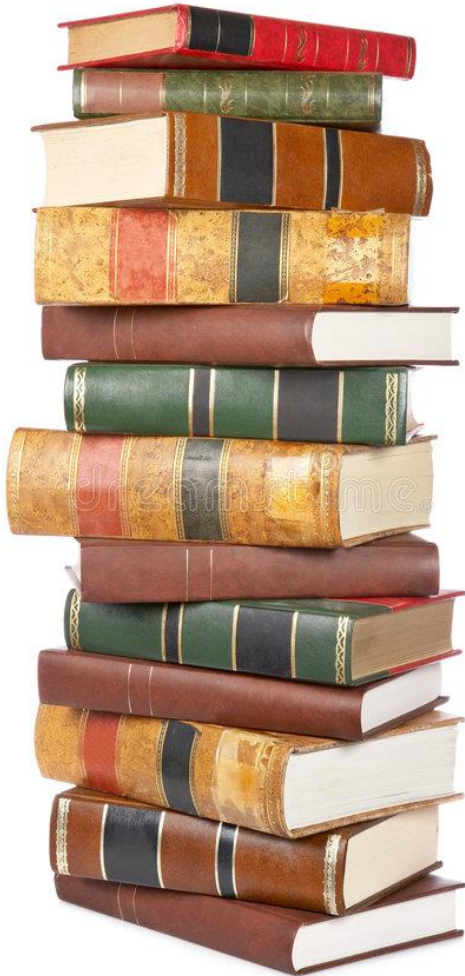
- ☑ Libro \Rightarrow {ISBN, título, autor, copias disponibles} \Rightarrow abstracción de datos adecuada.
- ☒ Libro \Rightarrow {color de tapa, peso, largo, ancho, tipo de letra} \Rightarrow abstracción de datos inadecuada.
- ☑ Libro \Rightarrow {buscar(libro), prestar(libro), recibir(libro)} \Rightarrow abstracción funcional adecuada.
- ☒ Libro \Rightarrow {calcular_peso(libro), superficie_tapa(libro)} \Rightarrow abstracción funcional inadecuada.



Pilas

- Una pila es una estructura lineal, en el cual los elementos se organizan de forma tal que:
 - Uno de los elementos se ubica al principio (en el frente o tope de la pila).
 - Los demás se enciman o apilan uno sobre otro a partir del primero en ser insertado → y este queda ubicado al fondo de la pila.

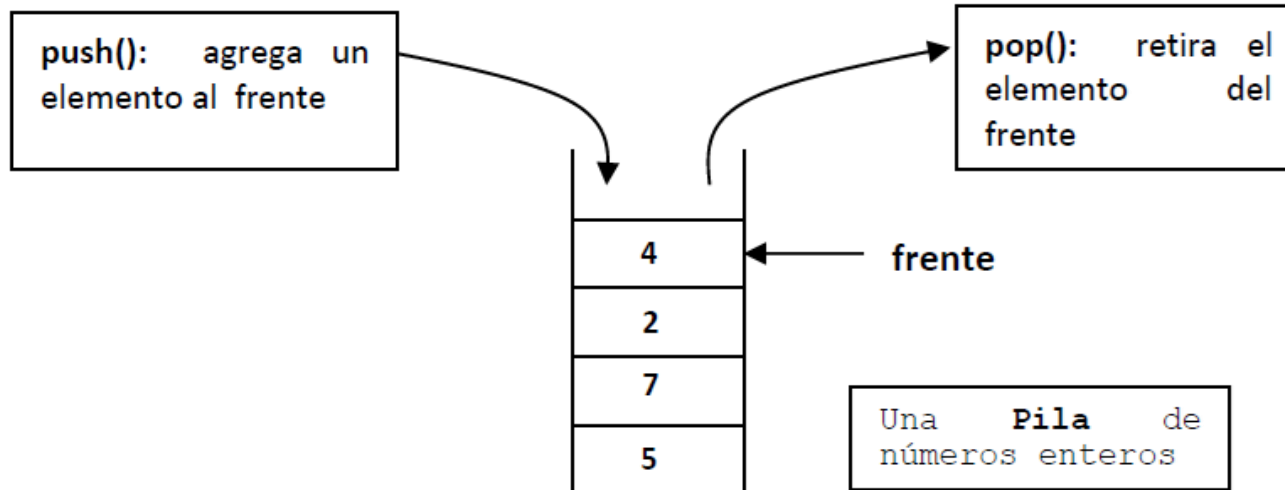
Pilas: ejemplos...



Pilas – Esquema Conceptual

Ejemplo: valores a cargar en la pila → 5 7 2 4

Figura 2: Esquema conceptual de una pila de números enteros.



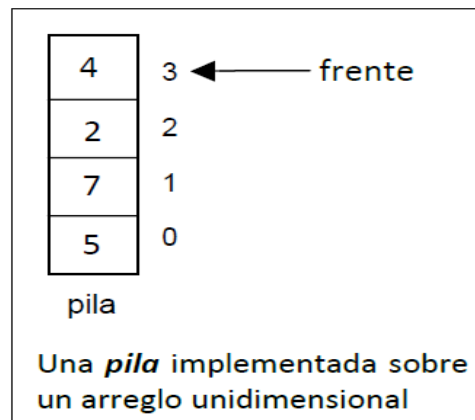
- Aplicación: las pilas son útiles en situaciones donde se debe invertir la secuencia de entrada.
- Esta forma de procesamiento con inversión de entradas se denomina **LIFO** (Último en entrar –Primero en salir)

Pilas: un modo de implementación

- La forma mas simple de implementar una pila en un programa, es usando **un arreglo unidimensional** como soporte.

Ejemplos: valores a cargar en la pila → **5 7 2 4**

Figura 3: Esquema de una pila implementada sobre un arreglo unidimensional.



- En este caso el **frente de la pila** es el último elemento del arreglo como se muestra en la Figura 3.

Pilas: un modo de implementación en Python

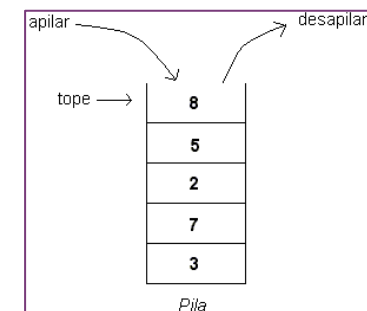
```
__author__ = 'Catedra de AED'

def init(): # Crea y retorna una pila vacía
    pila = []
    return pila

def esta_vacia(pila) # Chequea si la pila está vacía.
    n=len(pila)
    return n == 0

def add(pila, x): # Agrega un elemento a la pila.
    pila.append(x)

def remove(pila, x): # Elimina elemento de la pila.
    x = None
    if not esta_vacia(pila):
        x = pila[-1]
        del pila[-1]
    return x
```



Pila: un modo de implementación en Python

#Script Principal...

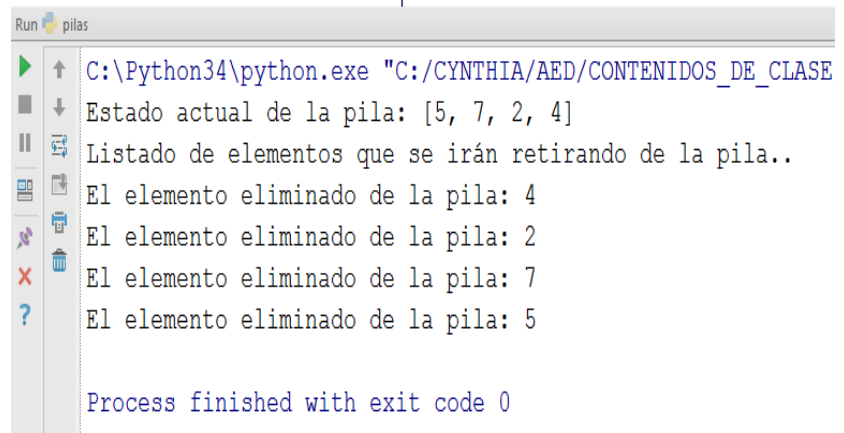
```
def test():
    p1 = init()
    add(p1, 5)
    add(p1, 7)
    add(p1, 2)
    add(p1, 4)

    print('Estado actual de la
    pila:', p1)

    while not esta_vacia(p1):
        print('Eliminado', remover(p1))

if __name__ == "__main__":
    test()
```

Captura de ejecución sobre el
contenido de la pila.



```
Run pilas
C:\Python34\python.exe "C:/CYNTHIA/AED/CONTENIDOS_DE_CLASE
Estado actual de la pila: [5, 7, 2, 4]
Listado de elementos que se irán retirando de la pila..
El elemento eliminado de la pila: 4
El elemento eliminado de la pila: 2
El elemento eliminado de la pila: 7
El elemento eliminado de la pila: 5
Process finished with exit code 0
```

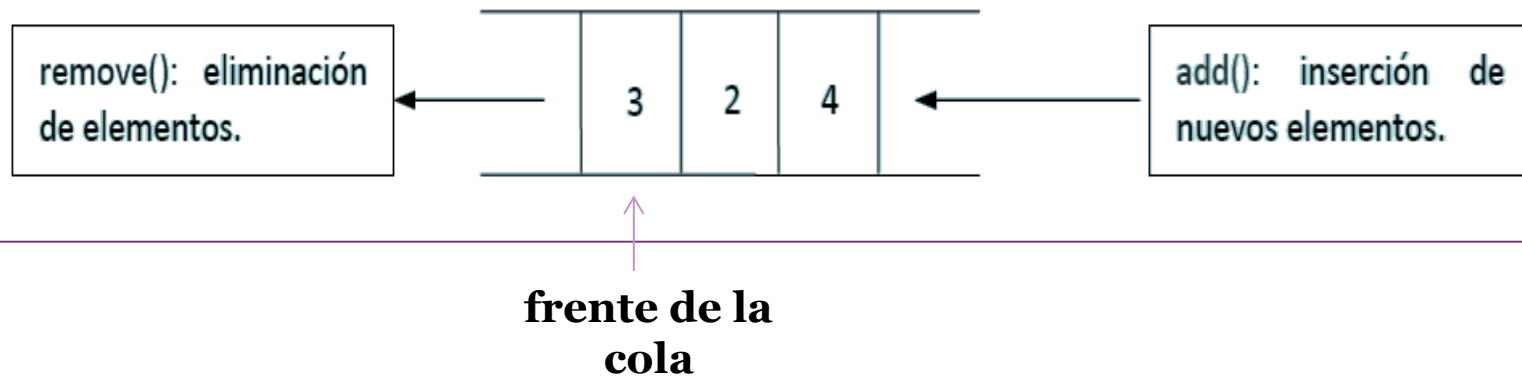
Colas

- Una **cola** es una **estructura lineal**, en el cual los elementos se organizan uno detrás de otro, quedando uno de ellos al principio de la cola (al frente) y otro en el último lugar (o al fondo).
 - La operación de **inserción** se hace de manera tal que el **nuevo componente** queda **último en la cola**.
 - La **operación de eliminación** se hace considerando el elemento que se encuentra en la primera posición.

Colas- Esquema Conceptual

Ejemplo: valores a cargar en la cola → 3 2 4

Figura 4: Esquema conceptual de una cola de números enteros.



- La forma de organizar los elementos en una cola se denomina **FIFO (Primero en entrar-Primero en Salir)**

Colas - Ámbito de aplicación

- Las colas son muy útiles en programas que requieren efectuar una simulación de situaciones de espera frente a un servicio. Ejemplo: ventanilla de un cajero de un banco.



Colas: un modo de implementación en Python

```
__author__ = 'Catedra de AED'

def init(): #Crea y retorna una pila vacía
    cola= []
    return cola

def esta_vacia(cola) #Chequea si la cola está vacía.
    n=len(cola)
    return n == 0

def add(cola, x): #Agrega un elemento a la cola.
    cola.append(x)

def remove(cola, x): #Elimina elemento de la cola.
    x = None
    if not esta_vacia(cola):
        x = cola[0]
        del cola[0]
    return x
```

Colas: un modo de implementación en Python

#Script Principal..

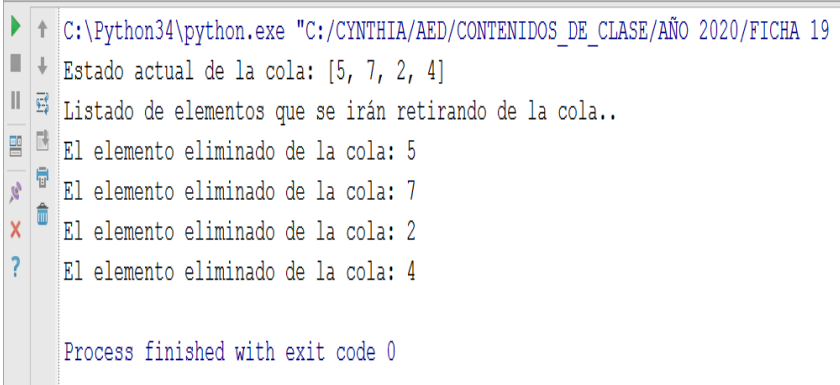
```
def test():
    c1 = init()
    add(c1, 5)
    add(c1, 7)
    add(c1, 2)
    add(c1, 4)

    print('Estado actual de la
    cola:', c1)

    while not esta_vacia(c1):
        print('Eliminado', remover(c1))

if __name__ == "__main__":
    test()
```

Captura de ejecución sobre el
contenido de la cola.



```
C:\Python34\python.exe "C:/CYNTHIA/AED/CONTENIDOS_DE_CLASE/AÑO 2020/FICHA 19
Estado actual de la cola: [5, 7, 2, 4]
Listado de elementos que se irán retirando de la cola..
El elemento eliminado de la cola: 5
El elemento eliminado de la cola: 7
El elemento eliminado de la cola: 2
El elemento eliminado de la cola: 4

Process finished with exit code 0
```


Bibliografía

- [1] V. Frittelli, Algoritmos y Estructuras de Datos, Córdoba: Universitas, 2001.
- [2] Python Software Foundation, "Python Documentation," 2015. [Online]. Available: <https://docs.python.org/3/>. [Accessed 24 February 2015].
- [3] M. Pilgrim, "Dive Into Python - Python from novice to pro," 2004. [Online]. Available: <http://www.diveintopython.net/toc/index.html>. [Accessed 6 March 2015].