

# Pilas y Colas

Emmanuel Cruz Hernández

[emmanuel\\_cruzh@ciencias.unam.mx](mailto:emmanuel_cruzh@ciencias.unam.mx)

3 de noviembre de 2020

# Contenido

## 1 Pilas

- Operaciones
- Aplicaciones

## 2 Colas

- Operaciones
- Aplicaciones

## 3 Bibliografía

Una pila es una estructura de datos cuya inserción y eliminación se basa en el principio **LIFO** (*last-in, last-out*).

Es una estructura que cuenta con las siguientes características:

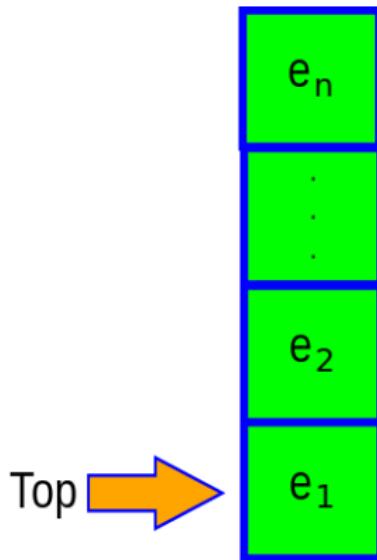
- Dinámica
- Homogénea
- Lineal
- Acceso directo

# Origen del concepto

La palabra *pila* se deriva de la metáfora de una pila de platos en un dispensador de platos para cafetería.



# Representación gráfica



# Formas de implementación

Algunas de las implementaciones para pilas son las siguientes:

- Pilas con Arreglos
- Pilas basadas en referencias

# Diferencia

A diferencia de las formas de implementación de listas, las pilas tienen la misma complejidad en tiempo con ambas implementaciones. La diferencia radica en la complejidad en espacio.

# Operaciones sobre pilas

- `push(e)`: agrega el elemento *e* en el tope de la lista.
- `pop()`: remueve y regresa el elemento en el tope de la pila.
- `top()`: regresa el elemento en el tope de la lista, sin eliminarlo.
- `isEmpty()`: regresa un booleano que indica si la pila está vacía.
- `clear()`: elimina todos los elementos de la estructura.

# Operaciones de colas

Se pueden consultar las operaciones de forma visual, en el siguiente enlace:

[https://docs.google.com/presentation/d/  
1aHoSn9fhaZg0M0xnKj1X1JuzUgAp704Ixhu49MExGTg/edit?usp=  
sharing](https://docs.google.com/presentation/d/1aHoSn9fhaZg0M0xnKj1X1JuzUgAp704Ixhu49MExGTg/edit?usp=sharing)

# Complejidad en tiempo

Método	Complejidad
<code>push(<i>e</i>)</code>	$O(1)$
<code>pop()</code>	$O(1)$
<code>top()</code>	$O(1)$
<code>size()</code>	$O(1)$
<code>isEmpty()</code>	$O(1)$
<code>clear()</code>	$O(1)$

# Aplicaciones de pilas

- Es una estructura muy útil para balancear símbolos en una expresión.
- Se usan para convertir expresiones con operadores infijos a expresiones con operaciones postfijos.
- Puede almacenar llamadas a métodos. Usualmente una pila se relaciona a la recursión.

# Colas

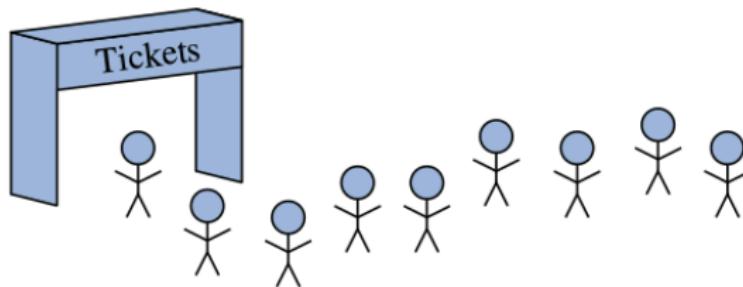
Una cola es una estructura de datos cuya inserción y eliminación se basa en el principio **FIFO** (*first-in, first-out*).

Es una estructura que cuenta con las siguientes características:

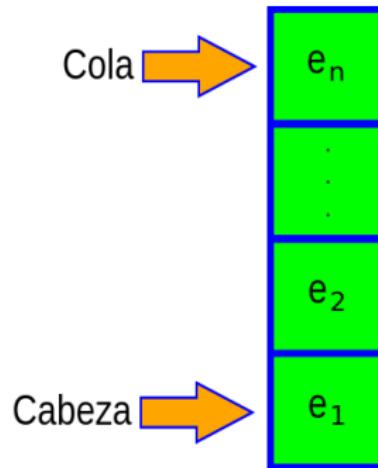
- Dinámica
- Homogénea
- Lineal
- Acceso directo

# Origen del concepto

La palabra *cola* se deriva de la metáfora de una fila de personas esperando por un servicio.



# Representación gráfica



# Formas de implementación

Algunas de las implementaciones para pilas son las siguientes:

- Pilas con Arreglos
- Pilas basadas en referencias

# Diferencia

Las colas tienen la misma complejidad en tiempo con ambas implementaciones. La diferencia radica en la complejidad en espacio al igual que en las pilas.

# Operaciones sobre colas

- enqueue(*e*): agrega el elemento *e* al final de la cola.
- dequeue(): elimina y regresa el primer elemento de la cola.
- first(): regresa el primer elemento de la cola, sin eliminarlo.
- size(): regresa el número de elementos contenidos.
- isEmpty(): regresa un booleano que indica si la cola está vacía.
- clear(): elimina todos los elementos de la estructura.

# Operaciones de colas

Se pueden consultar las operaciones de forma visual, en el siguiente enlace:

[https://docs.google.com/presentation/d/  
1iCidhFhpmq4xQwhEY-zGaYTSGKB3NPge57iogBrt85M/edit?usp=  
sharing](https://docs.google.com/presentation/d/1iCidhFhpmq4xQwhEY-zGaYTSGKB3NPge57iogBrt85M/edit?usp=sharing)

# Complejidad en tiempo

Método	Complejidad
enqueue( <i>e</i> )	O(1)
dequeue()	O(1)
first()	O(1)
size()	O(1)
isEmpty()	O(1)
clear()	O(1)

# Aplicaciones de colas

- Las llamadas a grandes empresas generalmente se colocan en una cola cuando todos los operadores están ocupados.
- En la espera de usuarios por el uso de algún recurso cuando todos están ocupados. El usuario que ha esperado más tiempo es el primero en usar el recurso cuando está disponible.

# Bibliografía

-  GOODRICH, M.T., TAMASSIA, R. Y GOLDWASSER, M.H., *Data Structures and Algorithms in Java*, Wiley, Sexta Edición, 2014.
-  WEISS, M. A., *Data Structures and Algorithm Analysis in Java*, Pearson, Tercera Edición, 2012.