



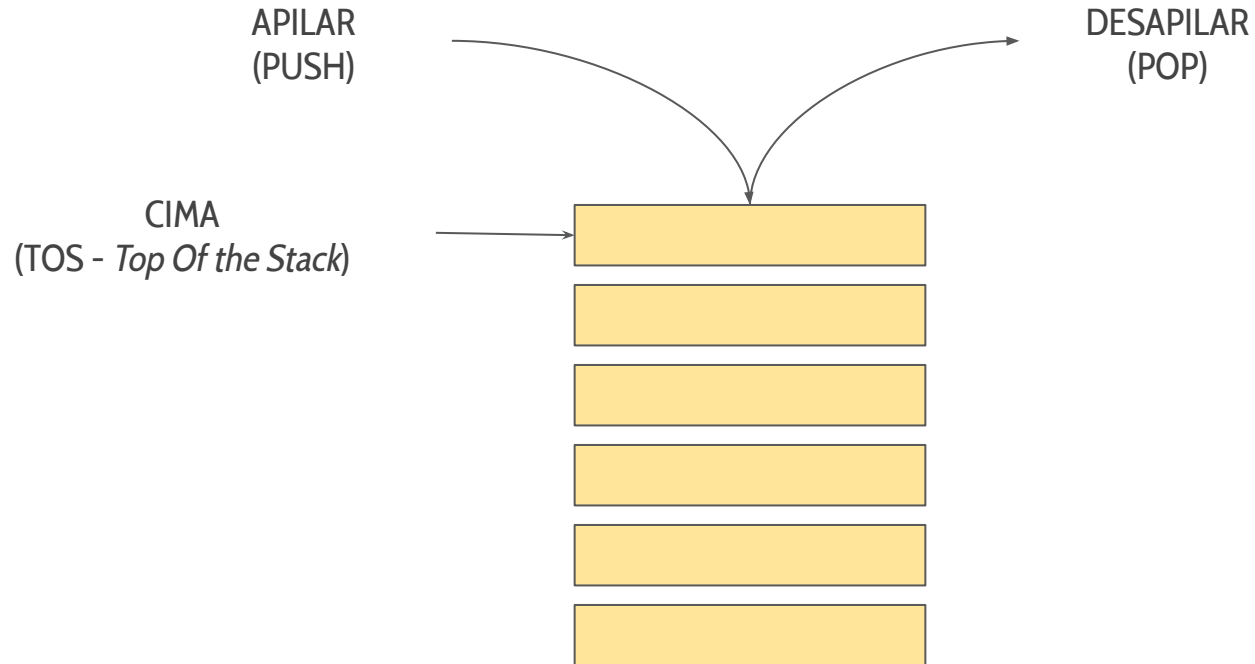
# Estructuras de datos básicas

AED I

2020

# Estructuras secuenciales: Pila

# pila (stack) - LIFO



# pila: operaciones

```
class Pila{
    private:
        struct nodo{
            int dato;
            nodo * siguiente;
        };
        nodo* inicio;
    public:
        Pila();
        void Apilar(int);
        int Desapilar();
        int Leer();
        bool estaVacía();
        int longitud();
        ~Pila();
};
```

# Ejercicio 2.05.

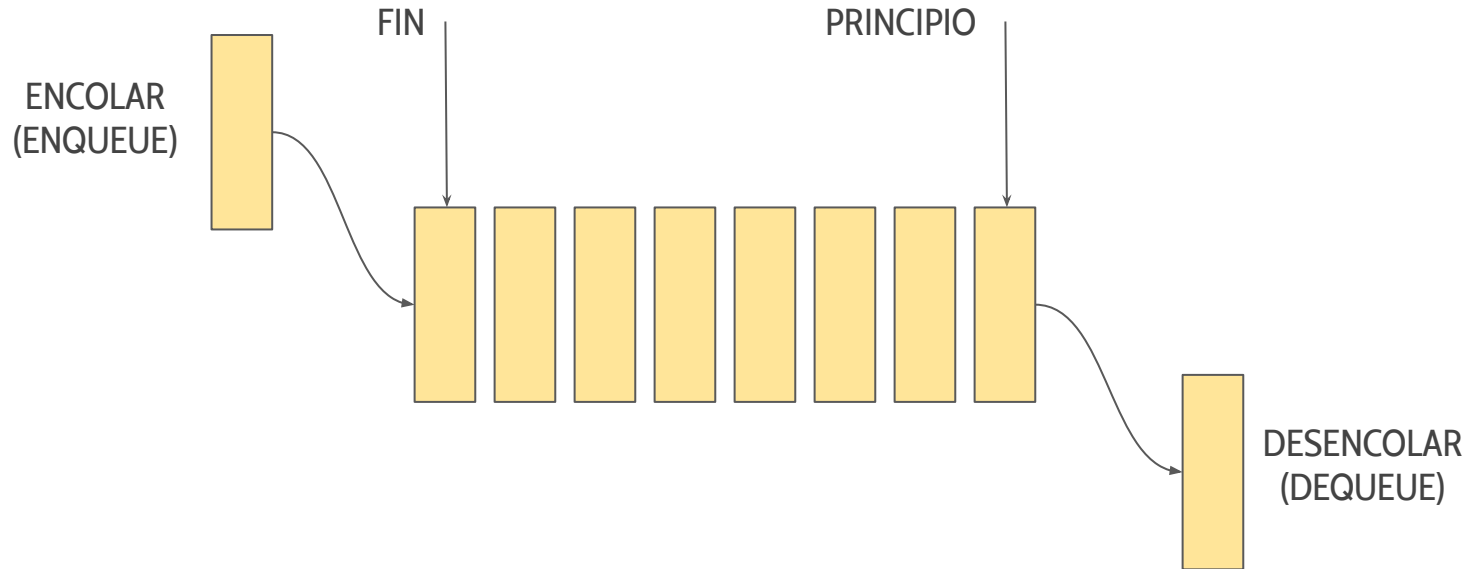
Implementar la TAD Pila

## Ejercicio 2.06.

Escribí un programa que mediante una pila valide si una secuencia de llaves, corchetes y paréntesis está balanceada.

# Estructuras secuenciales: Cola

# cola (queue) - FIFO





# cola: operaciones

```
class Cola{
    private:
        struct nodo{
            int dato;
            nodo * siguiente;};
        nodo* adelante;
    public:
        Cola();
        void Encolar(int);
        int Desencolar();
        int Leer();
        bool estaVacía();
        int longitud();
        ~Cola();
};
```

## Ejercicio 2.07.

Implementar la TAD Cola

## Ejercicio 2.08.

### Menor múltiplo formado por 0s y 9s

Creá un programa que dado un número entero calcule el menor múltiplo formado únicamente por dígitos 0s y 9s



Ingrese número: 5

El menor múltiplo formado por 0s y 9s es: 90

Ingrese número: 7

El menor múltiplo formado por 0s y 9s es: 9009

## Ejercicio 2.09.

Creá un programa que compare 2 listas enlazadas y determine si son iguales (secuencia de los mismos elementos)

## Ejercicio 2.10.

Detectá si una palabra o número es un palíndromo (capicúa). Utilice listas enlazadas o pilas para su resolución

**Método 1:** comparando una lista con la lista invertida

**Método 2:** usando una pila

Ingrese una palabra: `neuquen`

Es un palíndromo

Ingrese una palabra: `test`

No es un palíndromo

## Ejercicio 2.11.

Implementá un método que elimine los nodos repetidos de una lista

Pista: usá dos bucles anidados

## Ejercicio 2.12.

Creá una función que combine dos listas enlazadas previamente ordenadas. La lista resultante debe también estar ordenada

# La STL en c++

La Standard Template Library (STL) es una colección de estructuras de datos genéricas y algoritmos escritos en C++.

La STL proporciona una colección de estructuras de datos contenedoras y algoritmos genéricos que se pueden utilizar con estas estructuras.



# Pila (stack) en la STL

Funciones de la pila:

- `empty()`      Devuelve si la pila está vacía
- `size()`      Devuelve el tamaño de la pila
- `top()`      Devuelve una referencia al elemento de más arriba de la pila
- `push(g)`      Agrega el elemento g arriba en la pila
- `pop()`      Remueve el elemento de más arriba de la pila

# Ejemplo de stack

```
#include <iostream>
#include <stack>
using namespace std;
int main(){
    stack<int> s;
    s.push(10);
    s.push(15);
    s.push(30);
    s.push(1);
    cout << "El tamaño de la pila es: " << s.size() << endl;
    cout << "El elemento de arriba es: " << s.top() << endl;
    while(!s.empty()){
        cout << s.top() << endl;
        s.pop();
    }
    return 0;
}
```

# Cola (queue) en la STL

Funciones de la cola:

- `empty()`      Devuelve si la cola está vacía
- `size()`      Devuelve el tamaño de la cola
- `swap(x)`      Intercambia el contenido de la cola con aquella que se pasa por parámetro que debe ser del mismo tipo
- `top()`      Devuelve una referencia al elemento de más arriba de la pila
- `push(g)`      Agrega el elemento g al final de la cola
- `pop()`      Remueve el primer elemento de la cola

# Ejemplo de queue

```
#include <iostream>
#include <queue>
using namespace std;
int main(){
    queue<int> q;
    q.push(10);
    q.push(15);
    q.push(1);
    cout << "El tamaño de la cola es: " << q.size() << endl;
    cout << "El elemento de adelante es: " << q.front() << endl;
    cout << "El elemento de atrás es: " << q.back() << endl;
    while(!q.empty()){
        cout << q.front() << endl;
        q.pop();
    }
    return 0;
}
```

## Ejercicio 2.13.

Usá una pila para crear un programa que invierta una cadena.

Ingrese una palabra: `invertir`  
`ritrevni`

## Ejercicio 2.14.

Implementar el método `InsertarEnOrden()` que inserta en una lista previamente ordenada un nodo en la posición correspondiente según el orden ascendente