

# Lenguaje de programación

## C++

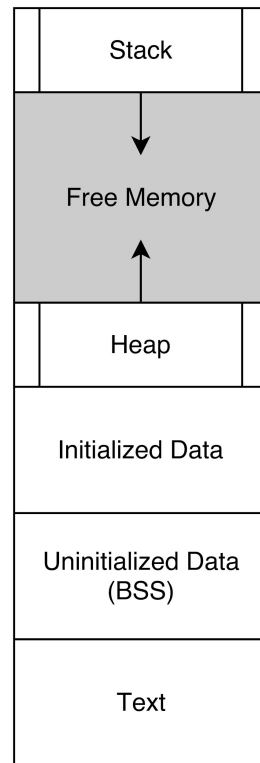
---

Ejemplos básicos

# Manejo de memoria

Ya lo hemos visto en C y excepto porque los nombres de las funciones que en C++ varían, el concepto y manejo de la memoria es el mismo.

- Stack almacena las variables u objetos locales
- Heap almacena las variables u objetos alocados dinámicamente
- Data guarda variables globales y estáticas, se distinguen en caso que estén inicializadas o no.
- Text contiene el código que se está ejecutando.



# Ejemplos

```
class Estudiante {
    char* nombre;
    int edad;
    float promedio;
public:
    // constructor
    Estudiante(const char* nombre, int edad, float promedio) {
        this->nombre = new char[strlen(nombre) + 1];
        strcpy(this->nombre, nombre);
        this->edad = edad;
        this->promedio = promedio;
    }
    void print() {
        cout<<nombre<<" "<<edad<<" "<<promedio<<endl;
    }
};
```

La clase es una estructura que a diferencia de struct de C, esta posee atributos y métodos. Como vemos podemos usar char\* como lo venían haciendo antes.

Por lo mencionado antes, corresponderia hacer un destructor para liberar la memoria reservada para el string.

```
~Estudiante(){
    delete(nombre);
}
```

# Ejemplos:

Posible main...

```
Estudiante *P;  
Estudiante J("Pablo", 25, 8.8);  
J.print();  
  
P = new Estudiante("Juan", 26, 7.5);  
P->print();
```

Vemos la diferencia en declarar al objeto localmente, y a un puntero del mismo.

El puntero, hasta que no se instancia, no puede acceder a métodos de la clase.

Al finalizar, correspondería liberar la memoria utilizando ***delete(P);***