



Templates C++

Este documento se hizo con propósitos educativos, mucho del conocimiento e información del mismo se adjudica al Ing. Luis Humberto González Guerra.

Ozym4ndias | Roberto A. Pérez

=====

Templates en C++

La mayoría de las veces se requiere hacer una función o una clase la cual sería la misma funcionalidad, pero cambiaría el tipo de dato, y se tendría que hacer una por cada tipo, y esto no sería del todo eficiente.

=====

Motivación

Por ejemplo si una función obtuviera el menor de dos enteros y otra el menor de dos flotantes, funcionalmente es lo mismo pero con diferente tipo de dato. Tener dos funciones sería ineficiente.

=====

Templates

El uso de Templates (Platillas) es una característica del lenguaje de programación C++ en el cual permite que funciones y clases operen con tipos de datos genéricos, permitiendo con esto que una función o clase trabaje en muchos tipos de datos sin tener que reescribirse para cada uno.

=====

Ejemplo función

```
template <typename T>
T menor(T uno, T dos) {
    if (uno < dos) {
        return uno;
    }
    return dos;
}
```

```
template <class T>
class Value{
    public:
        Value();
        Value(T);
        T getValue;
        void setValue(T value);
    private:
        T value;
};
```

```
template <class T>
Value<T>::Value(T value){
    this->value = value;
}
```

```
template <class T>
T Value<T>::getValue(){
    return this->value;
}
```

=====

