

# Ayudantia 1

- Carlos Lagos - [carlos.lagosc@usm.cl](mailto:carlos.lagosc@usm.cl)
- Nangel Coello - [nangel.coello@usm.cl](mailto:nangel.coello@usm.cl)
- Rafael Baeza - [rafael.baezam@usm.cl](mailto:rafael.baezam@usm.cl)
- Benjamín Cárdenas - [benjamin.cardenas@usm.cl](mailto:benjamin.cardenas@usm.cl)

# Contenidos

- *Introducción a la Biblioteca Estándar de C++: STL*

# ¿Qué es la STL?

Es un conjunto de clases y funciones basadas en plantillas (templates).

Implementa **estructuras de datos** y **algoritmos** básicos (listas, pilas, colas, búsqueda, orden, etc).

Además agrega objetos iteradores para operar y/o acceder a las estructuras de datos.

# Componentes de la STL

La STL se compone de 4 elementos principales:

1. **Contenedores** (Estructuras de datos)
2. **Algoritmos**
3. **Iteradores**

# Contenedores

- Contenedores de secuencia  
Array, Vector, Deque, List, Forward List
- **Adaptadores de contenedores**  
Stack, Queue, Priority Queue
- **Contenedores asociativos**  
Set, Map, Multiset, Multimap
- **Contenedores asociativos desordenados**  
Unordered Set, Unordered Map, Unordered Multiset, Unordered Multimap

# Algoritmos Manipulativos (Modifican o reordenan)

Función	Operación
<b>copy</b>	Copia elementos de un rango a otro.
<b>fill</b>	Asigna un valor a todos los elementos de un rango.
<b>transform</b>	Aplica una función a cada elemento.
<b>replace</b>	Reemplaza elementos por un nuevo valor.
<b>swap</b>	Intercambia el contenido de dos variables.
<b>reverse</b>	Invierte el orden de los elementos.
<b>rotate</b>	Rota los elementos de un rango.
<b>remove</b>	Elimina elementos con un valor específico.
<b>unique</b>	Elimina duplicados consecutivos.

# Algoritmos Manipulativos (Modifican o reordenan)

```
#include<vector>
#include<algorithm>
#include<iostream>

using namespace std;

int main(){
    vector<int> ejemplo = {1, 2, 3, 4, 5, 6, 7};
    vector<int> copia_ejemplo(ejemplo.size());

    copy(ejemplo.begin(), ejemplo.end(), copia_ejemplo.begin());
    reverse(copia_ejemplo.begin(), copia_ejemplo.end());

    for(int i : copia_ejemplo) cout << i << ' ';

    return 0;
}

//output: 7 6 5 4 3 2 1
```

# Algoritmos No Manipulativos (Operan sin modificar

Función	Operación
<b>max_element</b>	Encuentra el elemento máximo.
<b>min_element</b>	Encuentra el elemento mínimo.
<b>accumulate</b>	Suma los elementos de un rango.
<b>count</b>	Cuenta ocurrencias de un elemento.
<b>find</b>	Busca el primer elemento que cumple una condición.
<b>is_permutation</b>	Comprueba si dos rangos son permutaciones.
<b>is_sorted</b>	Verifica si un rango está ordenado.
<b>partial_sum</b>	Calcula la suma acumulada.



# Iteradores

Objetos similares a punteros que sirven para recorrer los contenedores.

# Beneficios de la STL

- **Código genérico y reutilizable:** Escribe algoritmos que funcionan con múltiples tipos.
- **Enfoque en la solución de problemas:** Menos código de bajo nivel y más énfasis en la lógica.
- **Gestión de memoria:** Reduce errores comunes como pérdidas de memoria.

# Limitaciones de la STL

- **Sobrecarga de rendimiento:** La generalidad puede afectar la eficiencia en casos específicos.
- **Complejidad al depurar:** Las abstracciones pueden dificultar la identificación de errores.
- **Control de memoria:** Menor control sobre detalles de asignación y liberación.
- **Integración con estructuras personalizadas:** Algunas estructuras a medida pueden requerir soluciones específicas.

# ¡Gracias!

Preguntas y comentarios.