

# Estructuras de Datos

## Librería STL de C++

Prof. Luis Garreta

Ingeniería de Sistemas y Computación  
Pontificia Universidad Javeriana – Cali

10 de octubre de 2017

# Introducción a STL

- ▶ La STL es una colección de estructuras de datos y algoritmos de uso común.
- ▶ Está basada en plantillas utilizando programación genérica.
- ▶ Está diseñada para ser eficiente, evita el uso de funciones virtuales en favor de las plantillas evitando operaciones en tiempo de ejecución.
- ▶ La STL se podría dividir en tres grandes partes:
  - ▶ Contenedores (plantillas de estructuras de datos populares),
  - ▶ iteradores y
  - ▶ algoritmos.

# Introducción a los Contenedores

- ▶ Los contenedores de la STL son estructuras de datos capaces de contener casi cualquier tipo de objeto (hay algunas restricciones).
- ▶ Existen 3 tipos de clases contenedoras:
  - ▶ contenedores de primera clase,
  - ▶ adaptadores y
  - ▶ casi-contenedores.
- ▶ También podemos catalogar a los contenedores por el tipo. Existen 3 tipos de contenedores:
  - ▶ **Contenedores de secuencia:** estructuras de datos lineales tales como vectores y listas enlazadas
  - ▶ **Contenedores asociativos:** estructuras de datos no lineales que por lo general pueden localizar elementos almacenados en ellos rápidamente. Dichos elementos pueden almacenar conjuntos de valores, o pares clave/valor
  - ▶ **Adaptadores de contenedores:** versiones restringidas de los contenedores de secuencia

# Tipos de Contenedores

## Contenedores de secuencia (primera clase)

- ▶ **vector** Inserción y eliminación rápida en la parte final. Acceso directo a cualquier elemento.
- ▶ **deque** Inserciones y eliminaciones rápidas en la parte inicial o final. Acceso aleatorio a cualquier elemento.
- ▶ **list** Lista con enlace doble, inserción y eliminación rápida en cualquier parte.

## Contenedores asociativos (primera clase)

- ▶ **set** Búsqueda rápida, no se permiten duplicados.
- ▶ **multiset** Búsqueda rápida, se permiten duplicados
- ▶ **map** Asociación de uno a uno, no se permiten duplicados, búsqueda rápida basada en claves.
- ▶ **multimap** Asociación de uno a uno, se permiten duplicados, búsqueda rápida basada en claves.

## Adaptadores de contenedores

- ▶ **stack** Último en entrar primero en salir (UEPS).
- ▶ **queue** Primero en entrar, primero en salir (PEPS).
- ▶ **priority\_queue** El elemento de mayor prioridad siempre es el primero en salir.

# Funciones comunes de los contenedores

- ▶ La mayoría de contenedores de la STL proporcionan una funcionalidad similar.
- ▶ Hay muchas operaciones genéricas, como el método `size`, que se aplica a todos los contenedores.
- ▶ La siguiente tabla muestra los métodos comunes en todos los contenedores.
- ▶ Los operadores sobrecargados `operator<`, `operator<=`, `operator>`, `operator>=`, `operator==` y `operator!=` no se proporcionan para contenedores `priority_queue`.

# Tabla de Funciones Comunes

Funciones miembros comunes	Descripción
constructor predeterminado	Un constructor para crear un contenedor vacío. Por lo general, cada contenedor cuenta con varios constructores que proporcionan distintos métodos de inicialización.
constructor de copia	Un constructor que inicializa al contenedor para que sea una copia de un contenedor existente del mismo tipo.
destructor	La función destructora para encargarse de la limpieza, una vez que el contenedor ya no sea necesario.
empty	Devuelve true si no hay elementos en el contenedor, en caso contrario devuelve false.
insert	Inserta un elemento en el contenedor.
size	Devuelve el número de elementos que hay actualmente en el contenedor.
operator=	Asigna un contenedor a otro.
operator<	Devuelve true si el primer contenedor es menor que el segundo, en caso contrario devuelve false.
operator<=	Devuelve true si el primer contenedor es menor o igual que el segundo, en caso contrario devuelve false.
operator>	Devuelve true si el primer contenedor es mayor que el segundo, en caso contrario devuelve false.
operator>=	Devuelve true si el primer contenedor es mayor o igual que el segundo, en caso contrario devuelve false.
operator==	Devuelve true si el primer contenedor es igual que el segundo, en caso contrario devuelve false.
operator!=	Devuelve true si el primer contenedor es distinto que el segundo, en caso contrario devuelve false.
swap	Intercambia los elementos de dos contenedores.

# Tarea: Estructuras de Datos Básicas con Templates

Implementar las siguientes estructuras de acuerdo a lo visto en clase:

- ▶ Todas las clases derivan de la clase `vector` de la STL de C++
- ▶ Los archivos deben descargarse de la nube (con *git*: github o bitbucket)
- ▶ La evaluación se hará con herramientas linux consola (git, nano, g++)
- ▶ Tutorial STL vector: [http://www.codeguru.com/cpp/cpp/cpp\\_mfc/stl/article.php/c4027/C-Tutorial-A-Beginners-Guide-to-stdvector-Part-1.htm](http://www.codeguru.com/cpp/cpp/cpp_mfc/stl/article.php/c4027/C-Tutorial-A-Beginners-Guide-to-stdvector-Part-1.htm)
- ▶ Métodos de `vector`: <http://www.yolinux.com/TUTORIALS/LinuxTutorialC++STL.html>

