



STL, compilación y make

Algoritmos y programación II

¿Qué vamos a ver hoy?

01

STL

Definición

02

STL vector

Definición, uso y ejemplos

03

Compilación

Teoría y compilación con terminal

04

Makefile

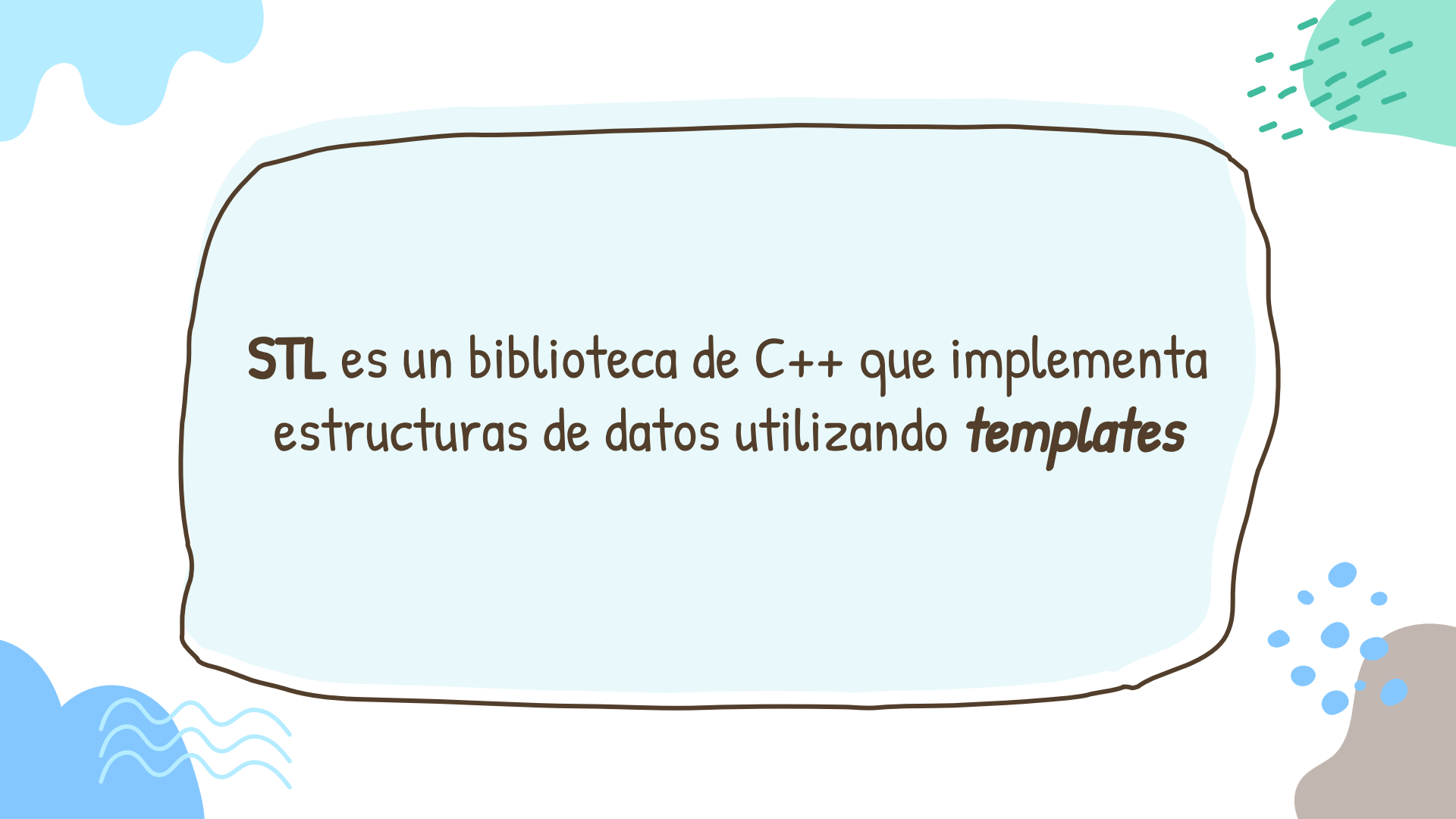
Armado de un make básico



01

STL

Standard Template
Library



STL es una biblioteca de C++ que implementa estructuras de datos utilizando *templates*

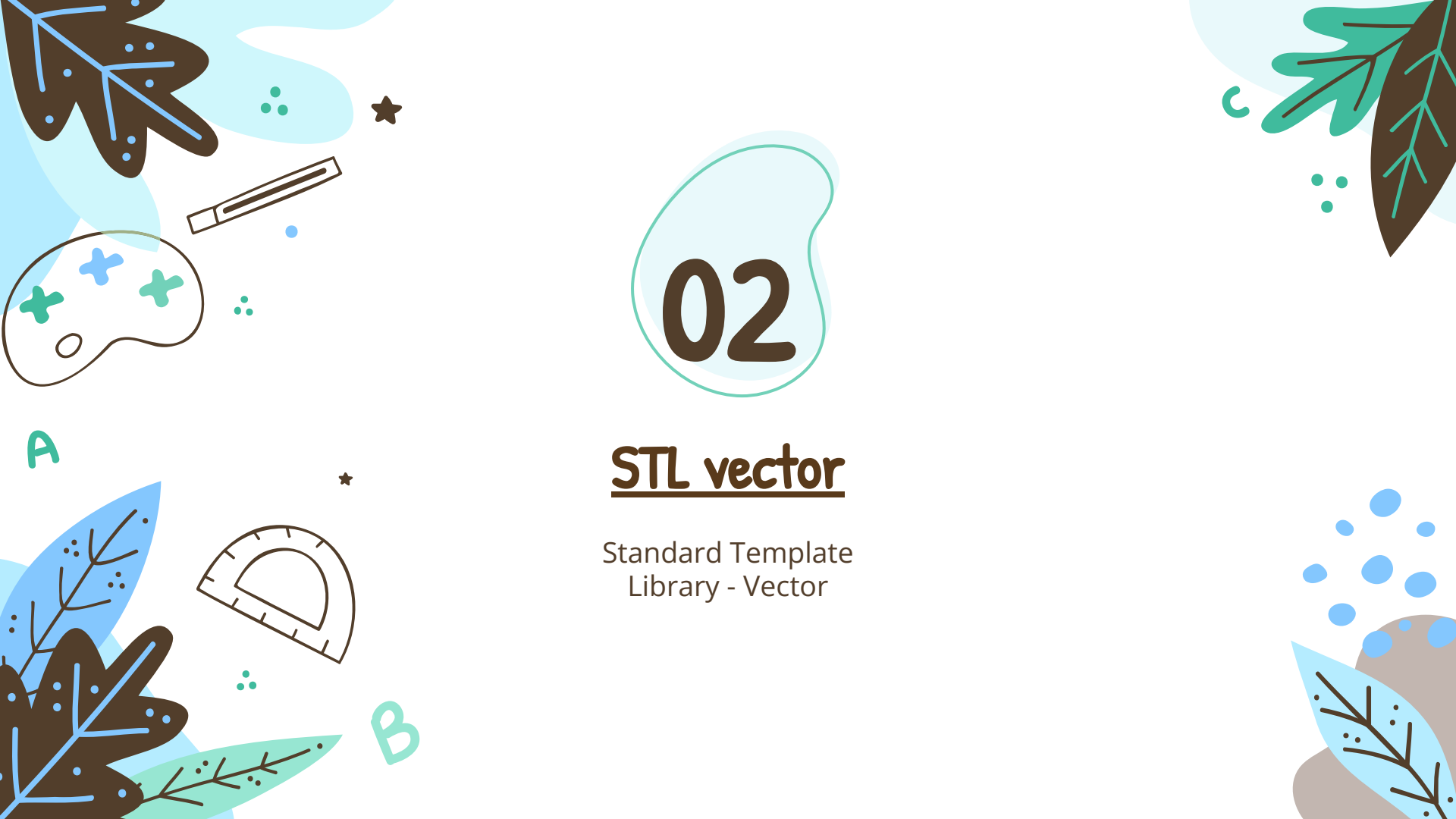
¿Lo uso siempre?

Pros

- Reutilizabilidad
- Eficiencia
- Gran comunidad

Contras

- Mala performance
- Curva de aprendizaje
- Falta de control
- Es “magico”



02

STL vector

Standard Template
Library - Vector


¿Qué es?

Dentro de STL

Vector es una clase de la biblioteca que implementa un

vector dinámico

Es aquel que se redimensiona segun la cantidad de elementos que debe contener. Pone en uso el concepto de *memoria dinámica*



```
vector<int> numeros = {1, 2, 3, 4, 5};
```

```
vector<string> frutas;
```

```
frutas.emplace_back( str: "Manzana"); // Crea el objeto dentro del vector no lo copia
```

```
string naranja = "Naranja";
```

```
frutas.push_back(naranja); // Usar si ya tenes un objeto creado y deseas agregar una copia de este
```

```
imprimirVectorEnteros( &: numeros);
```

```
imprimirVectorFrutas( &: frutas);
```

```
if (!numeros.empty()) {
```

```
    numeros.pop_back(); // Elimina el último elemento
```


```
    numeros.erase( first: numeros.begin() + 1, last: numeros.begin() + 3); // Elimina del segundo elemento al tercero
```

```
}
```

```
cout << "-----" << endl;
```

```
imprimirVectorEnteros( &: numeros);
```

```
imprimirVectorFrutas( &: frutas);
```





Ejercicio



Para poner en práctica lo aprendido de STL vector se deberá:

- Crear un vector de enteros
- Cargarlo con 10 números aleatorios
- Imprimir los números
- Buscar si está el número 3 en el vector
- Eliminar el 4to elemento del vector
- Imprimir los números restantes

Tiempo: 15 minutos





03

Compilación

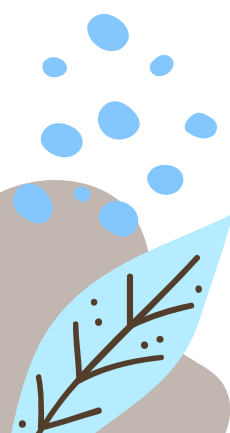
Teoría y compilación con
terminal



Declaración

Se declaran las cabeceras de las funciones → se encuentran **generalmente** en los headers de las bibliotecas (.h)

Salvo aquellas funciones de uso interno que se declaran en el .cpp



Definición

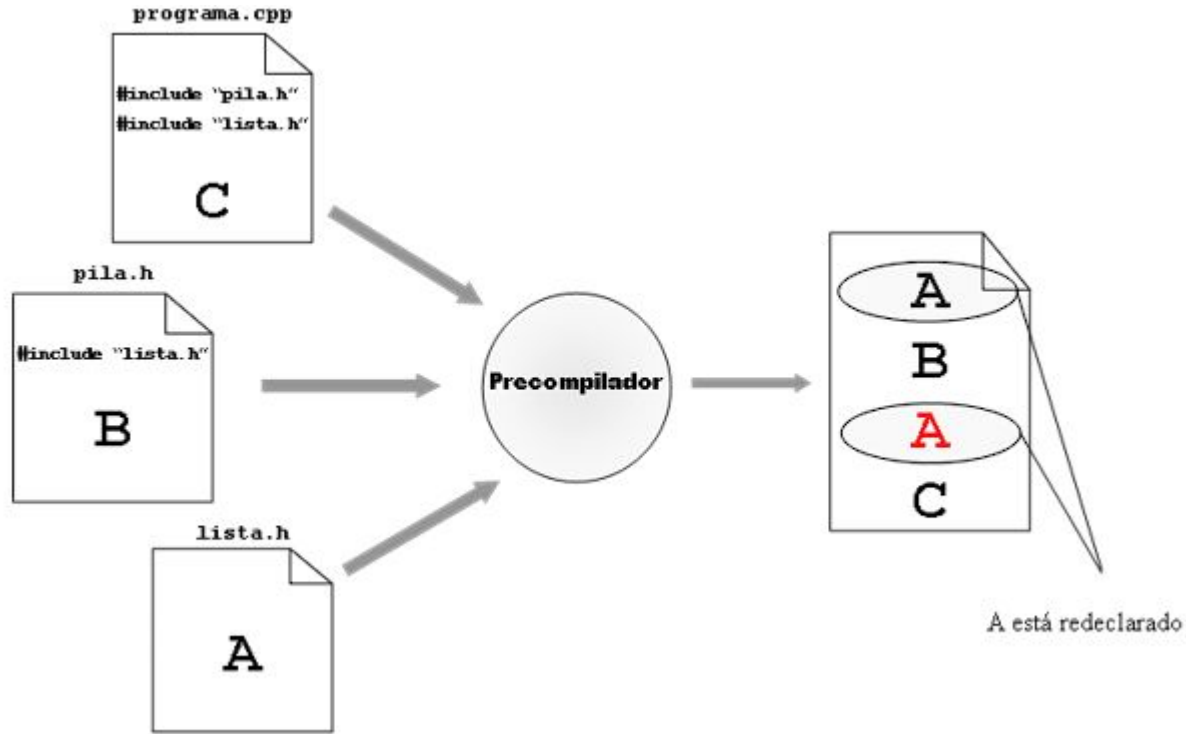
Es la implementación de las funciones declaradas en el .h → Se encuentran **siempre** en el .cpp



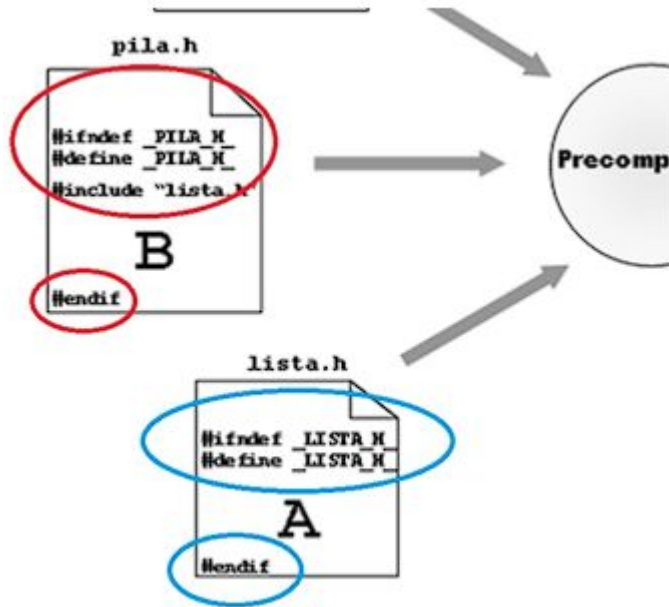
Proceso de compilación



Declaración múltiple



Declaración múltiple - solución



```
#ifndef _ARCHIVO_H
```

```
#define _ARCHIVO_H
```

```
// Código
```

```
#endif
```

Compilar en la terminal

```
g++ *.cpp -o programa -Wall -Werror -Wconversion
```

***.cpp**

Selecciona
todos los .cpp.
Se puede
poner de a
uno

-o programa

Le pone el
nombre
"programa"
al
ejecutable

-Wall

Activa
warnings
adicionales

-Werror

Todos los
warnings se
tratan como
errores

-Wconversion

Incluye warnings por
conversiones
implícitas de datos
que pueden cambiar
su valor

The background is white with decorative elements. In the top-left, there's a large brown leaf with blue veins and a light blue cloud-like shape. Below it is a paint palette with green and blue crosses and a pencil. In the bottom-left, there's a blue leaf, a brown leaf, and a protractor. In the top-right, there's a green leaf. In the bottom-right, there's a blue leaf and a cluster of blue dots. The number '04' is inside a light blue cloud shape in the center.

04

Makefile

Armado de un make
básico



Un **Makefile** es un archivo de texto que contiene **instrucciones para automatizar el proceso de compilación** de un programa.

Es especialmente útil cuando tenemos varios archivos fuente o están en distintos directorios.

Ejemplo

CXX = g++

CXX: contiene el nombre del compilador

CXXFLAGS = -Wall -Werror -Wconversion

Indica los flags para compilar

all:

\$(CXX) \$(CXXFLAGS) -o nombre main.cpp

Indica la linea de compilación llamando a CXX y CXXFLAGS

clean:

rm -f nombre *.o

Limpia los archivos de código objeto generados durante la compilación

Ejemplo con *.cpp

```
CXX = g++  
CXXFLAGS = -Wall -Werror -Wconversion  
  
all:  
    $(CXX) $(CXXFLAGS) -o programa *.cpp  
  
clean:  
    rm -f programa *.o
```

Define que tome todos los *.cpp
como fuente

Gracias!
Preguntas?

