

Encabezados

- Necesidad
- Funcionamiento
- Separación en definiciones e implementación

Encabezados

- Imaginemos que tenemos muchos programas que tienen código en común. Esto puede llevar a los siguientes problemas:
 - El copiar y pegar el código cada vez que lo necesitamos, es un tarea propensa a errores.
 - Cada vez que realizamos un cambio en esa parte del código, es probable que también tengamos que cambiar el resto de las copias del código de otros programas.
- Por ejemplo:

```
struct Lista { ... };  
void agregarInicio(...) {...}  
void agregarFinal(...) {...}  
...  
  
int main() {  
    ...  
}
```

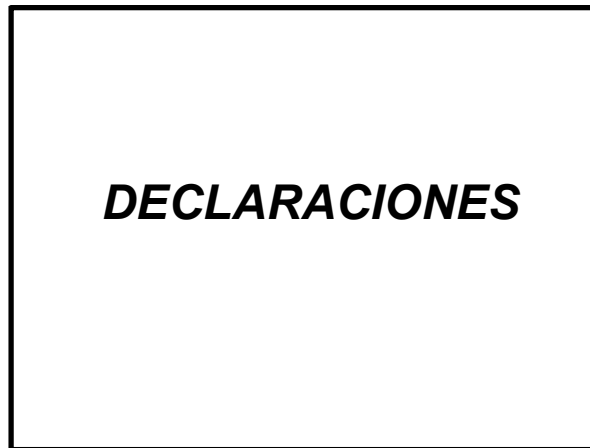
pilas.cpp

```
struct Lista { ... };  
void agregarInicio(...) {...}  
void agregarFinal(...) {...}  
...  
  
int main() {  
    ...  
}
```

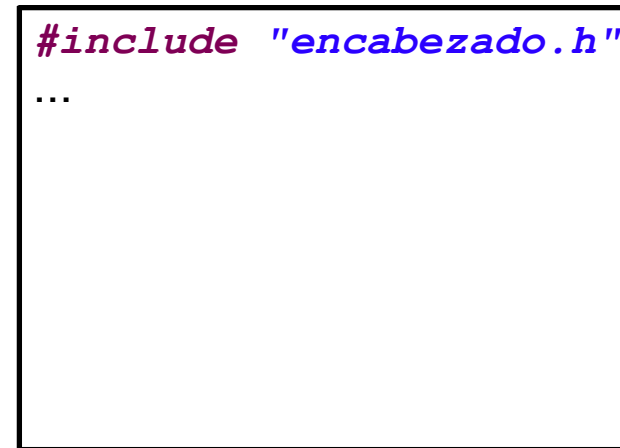
filas.cpp

Encabezados

- Un encabezado es un archivo que contiene declaraciones de C/C++ que podrán compartirse entre varios archivos de código fuente. La forma de utilizar un encabezado es incluyéndolo, con la directiva del preprocesador **#include**.
- La extensión de los encabezados es, por lo general, “.h” o “.hpp”.



encabezado.h



fuelle.cpp o
fuelle.h

Encabezados

- Hay que tener cuidado cuando se incluye el mismo código más de una vez en el mismo programa, ya que será procesado varias veces. La forma de evitar este problema, es encapsular en un condicional todo el contenido del encabezado. Por ejemplo:

```
#ifndef CONSTANTE  
#define CONSTANTE
```

DECLARACIONES

```
#endif
```

lista.h

```
#include "lista.h"  
...
```

pila.h

```
#include "lista.h"  
...
```

fila.h

```
#include "pila.h"  
#include "fila.h"  
...
```

main.cpp

Encabezados

- Una posibilidad que tenemos al trabajar con encabezados, es separar las declaraciones de estructuras y funciones de la implementación o definición de las mismas. Por ejemplo:

```
#ifndef LISTA_H
#define LISTA_H

DECLARACIONES

#endif
```

lista.h

```
#include "lista.h"

IMPLEMENTACION
```

lista.cpp

```
#ifndef PILA_H
#define PILA_H
#include "lista.h"

DECLARACIONES

#endif
```

pila.h

```
#include "pila.h"

IMPLEMENTACION
```

pila.cpp

Encabezados

```
#ifndef LISTA_H
#define LISTA_H

struct Nodo_lista...;

Nodo_lista * nueva_lista(int val, Nodo_lista * sig);
void agregar_final(Nodo_lista * &l, int val);
void mostrar_lista(Nodo_lista * l);

void vaciar_lista(Nodo_lista * &l);
#endif
```

lista.h

```
#include "lista.h"

Nodo_lista * nueva_lista(int val, Nodo_lista * sig)
{
    Nodo_lista * nuevo = new Nodo_lista;
    nuevo->val = val;
    nuevo->sig = sig;
    return nuevo;
}
...
```

lista.cpp

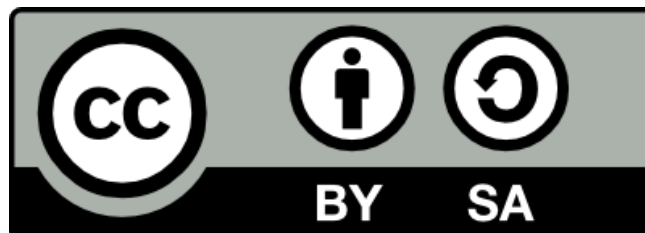
Federico Améndola

Martín Fernández

Consultas: laboratorio.ayda@alumnos.exa.unicen.edu.ar

Licencia creative commons

Atribución-Compartir Obras Derivadas Igual 2.5 Argentina



<http://creativecommons.org/licenses/by-sa/2.5/ar/>