

# Compiladores

# Ciclo de creación de un programa

## 1. Editor

- **Utilidad** → Creación y edición del código fuente.
- **Extensión** → *.c*

## 2. Preprocesador (*opcional*)

- **Utilidad** → Eliminación de comentarios, expansión de macros, etc.
- **Extensión** → *.i*
- **Aplicación** → *cpp*

## 3. Compilador (*opcional*)

- **Utilidad** → Creación del código fuente ensamblador.
- **Extensión** → *.s*
- **Aplicación** → *comp*

## 4. Ensamblador

- **Utilidad** → Creación del código objeto.
- **Extensión** → *.o*
- **Aplicación** → *as*

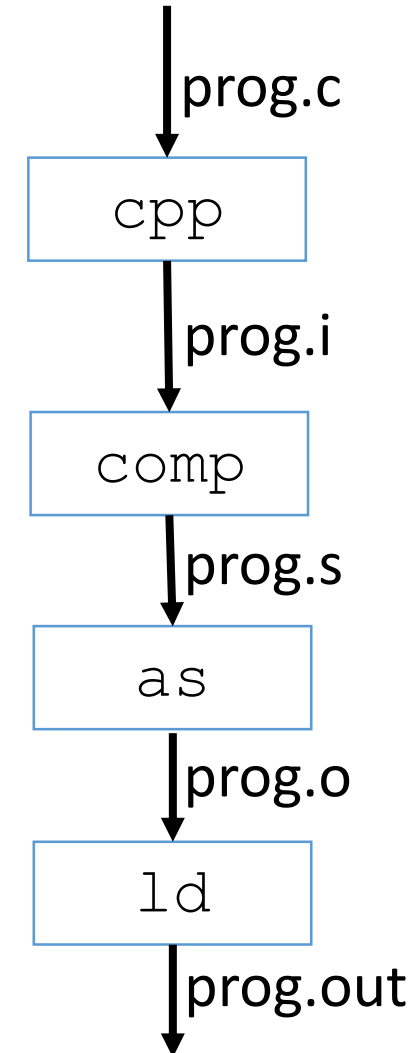
## 5. Enlazador

- **Utilidad** → Generación del ejecutable.
- **Extensión** → *.out*
- **Aplicación** → *ld*

## 6. Depurador (*opcional*)

- **Utilidad** → Detección de errores.
- **Aplicación** → *gdb*

Fase de compilación



# GCC (I) - Definición

## GNU Compiler Collection (GCC)

Es una suite de compiladores portable que soporta diferentes lenguajes como C, C++, Objective-C o Java. Es el encargado de **compilar**, **enlazar** y **generar** el archivo ejecutable a partir del archivo fuente.

### Parámetros

- **-E**: Para tras el preprocesado
- **-S**: Para tras el compilado (no ensamblar)
- **-c**: Para antes de enlazar
- **-o nombre**: Especifica el nombre del archivo de salida.

GCC es la interfaz real entre el usuario y las fases de compilación.

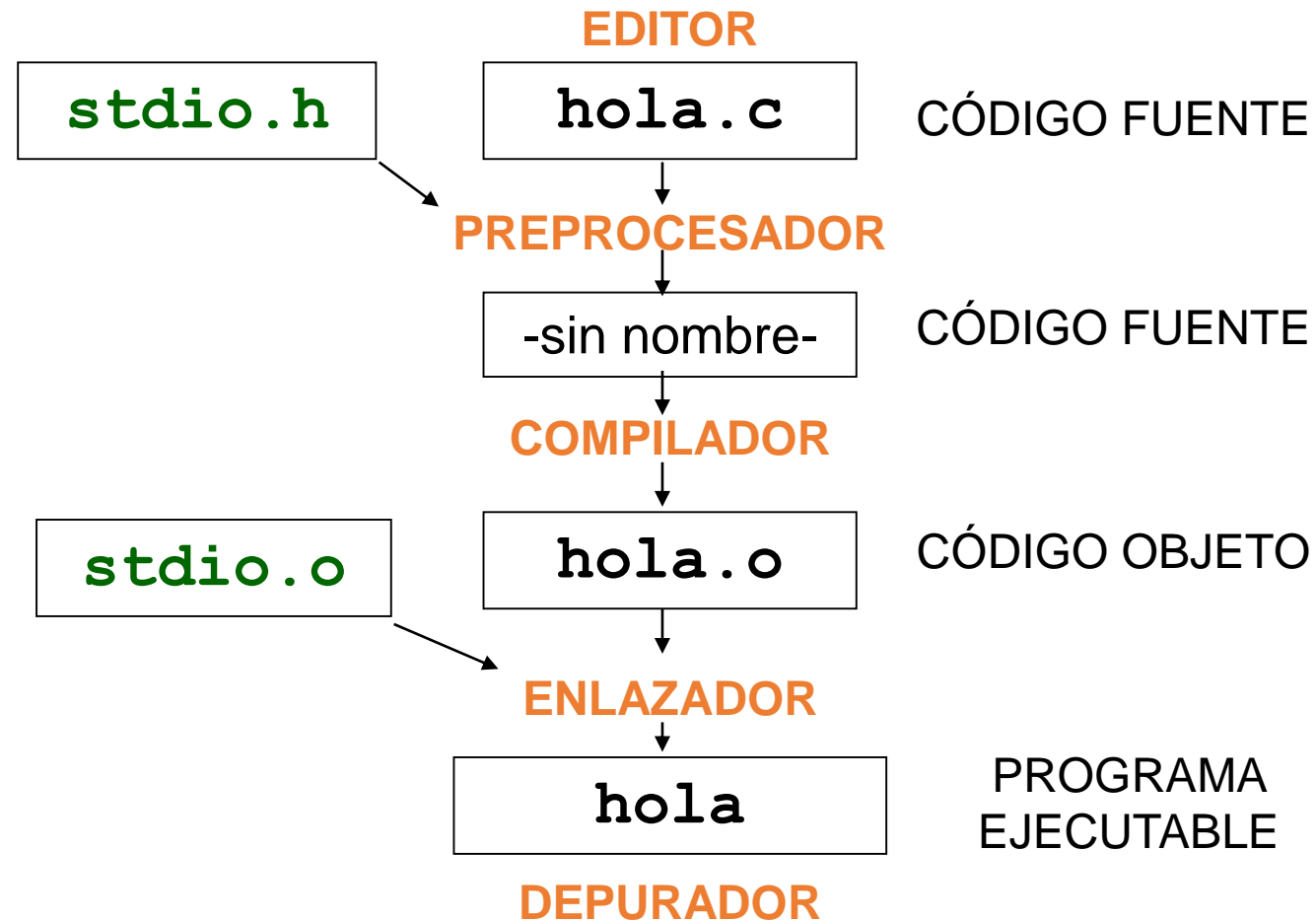
# "Hola mundo" en C

`hola.c`

```
/*  Mi primer programa en C  */  
#include <stdio.h> // Directiva  
int main () // Cuerpo principal.  
{  
    printf ("Hola mundo.\n");  
    return 0;  
} // Fin
```

`gcc hola.c -o hola`

# Del código fuente al ejecutable



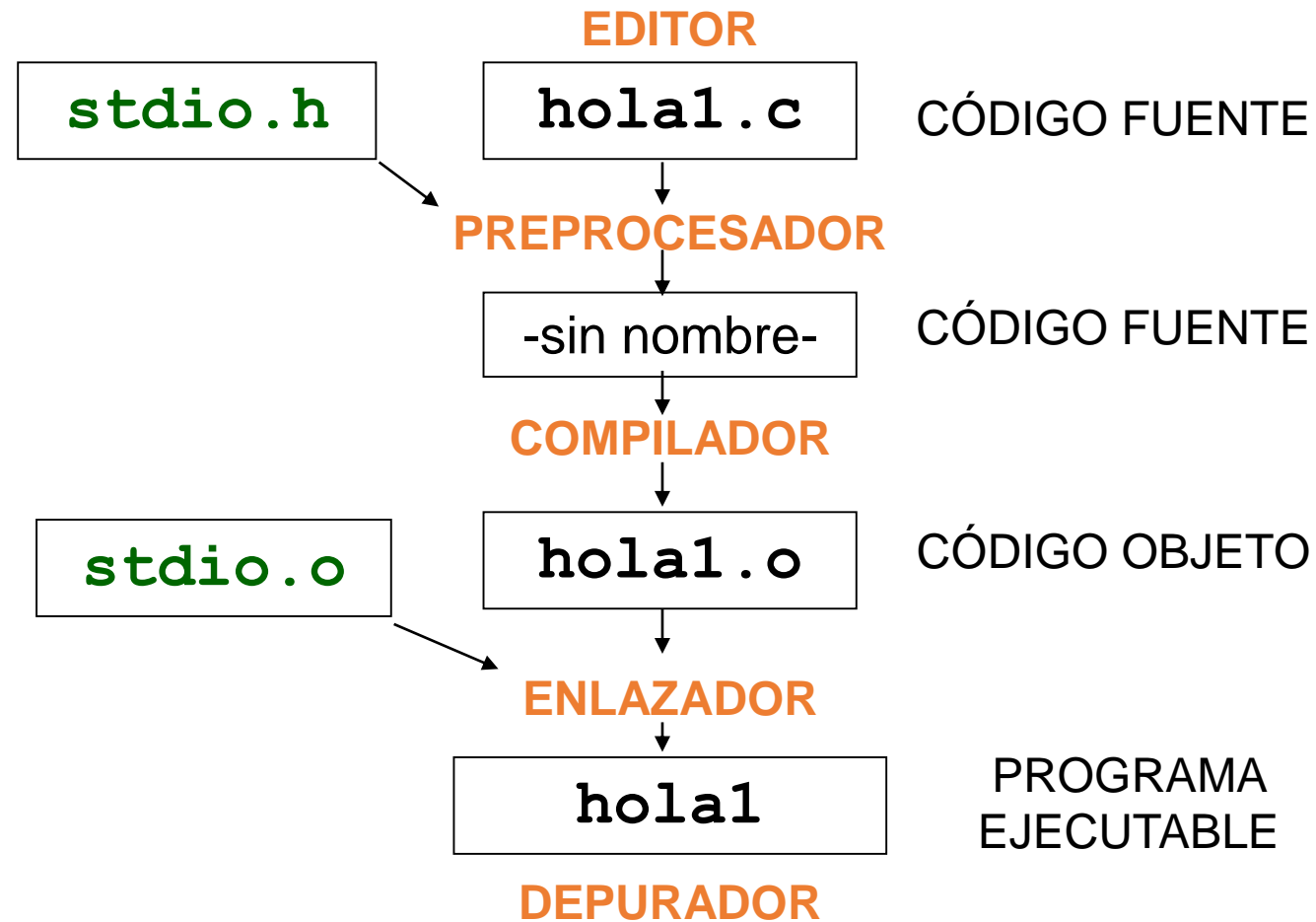
# Estructura de un programa (1)

**hola1.c**

<pre>/* hola1 (módulo único) */  #include &lt;stdio.h&gt;  void hola (void) {     printf ("Hola.\n"); }  void adios (void) // Sin { // acento     printf ("Adios.\n"); }</pre>	<pre>// (continuación)  int main () {     hola ();     adios (); // Sin acento      return 0; }  /* Fin */</pre>
--	--

**gcc hola1.c -o hola1**

# Del código fuente al ejecutable



# Estructura de un programa (2)

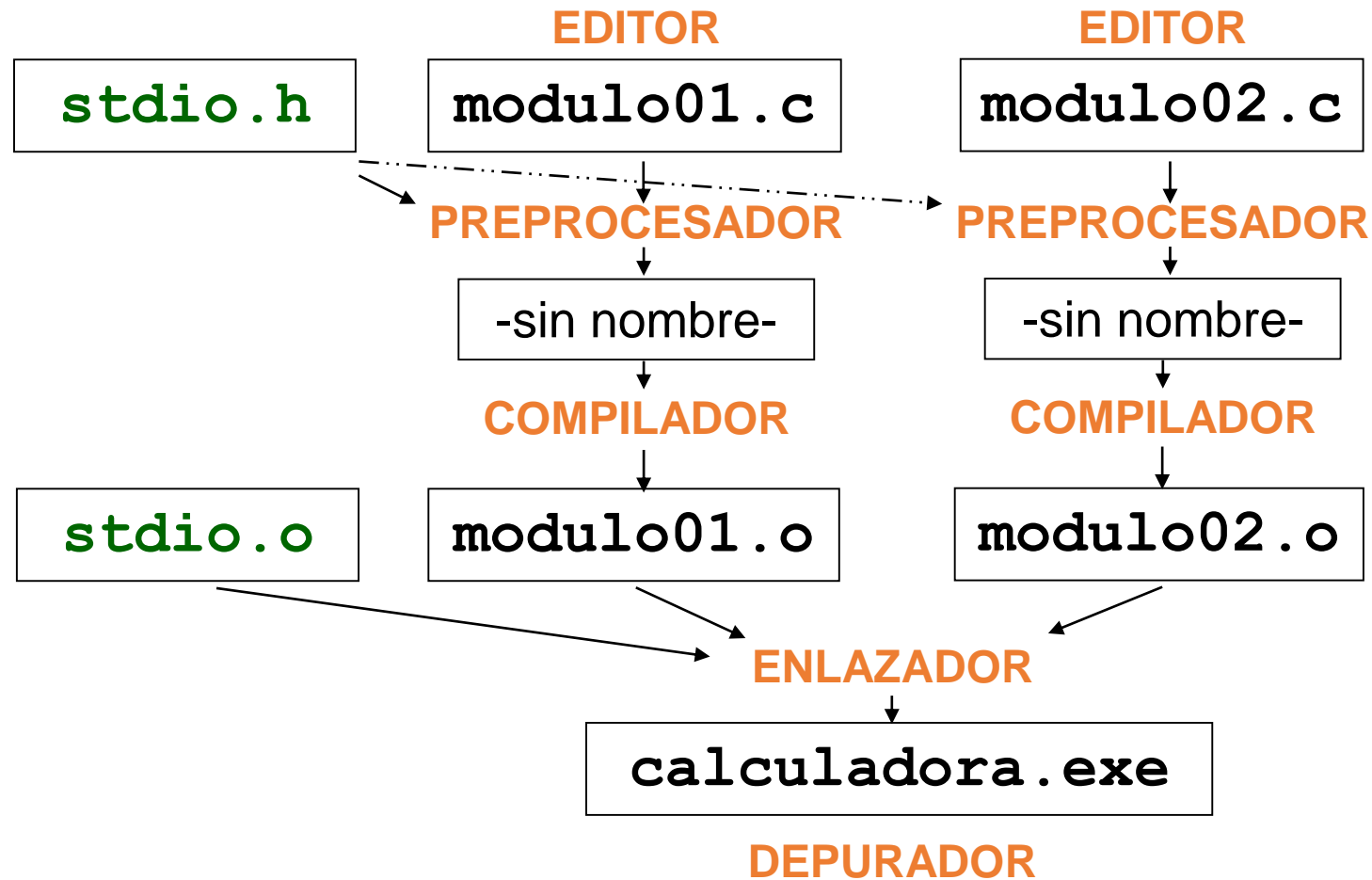
**hola2.c**

<pre>/* hola2 (módulo único) */  #include &lt;stdio.h&gt;  void hola (void); // OJO: void adios (void); // ";"  int main () {     hola ();     adios ();     return 0; }</pre>	<pre>// (continuación)  void hola (void) {     printf ("Hola.\n"); }  void adios (void) {     printf ("Adios.\n"); }  /* Fin */</pre>
--	---

**gcc hola2.c -o hola2**



# Programa con varios módulos



# Estructura de un programa (3)

## **hola3.c**

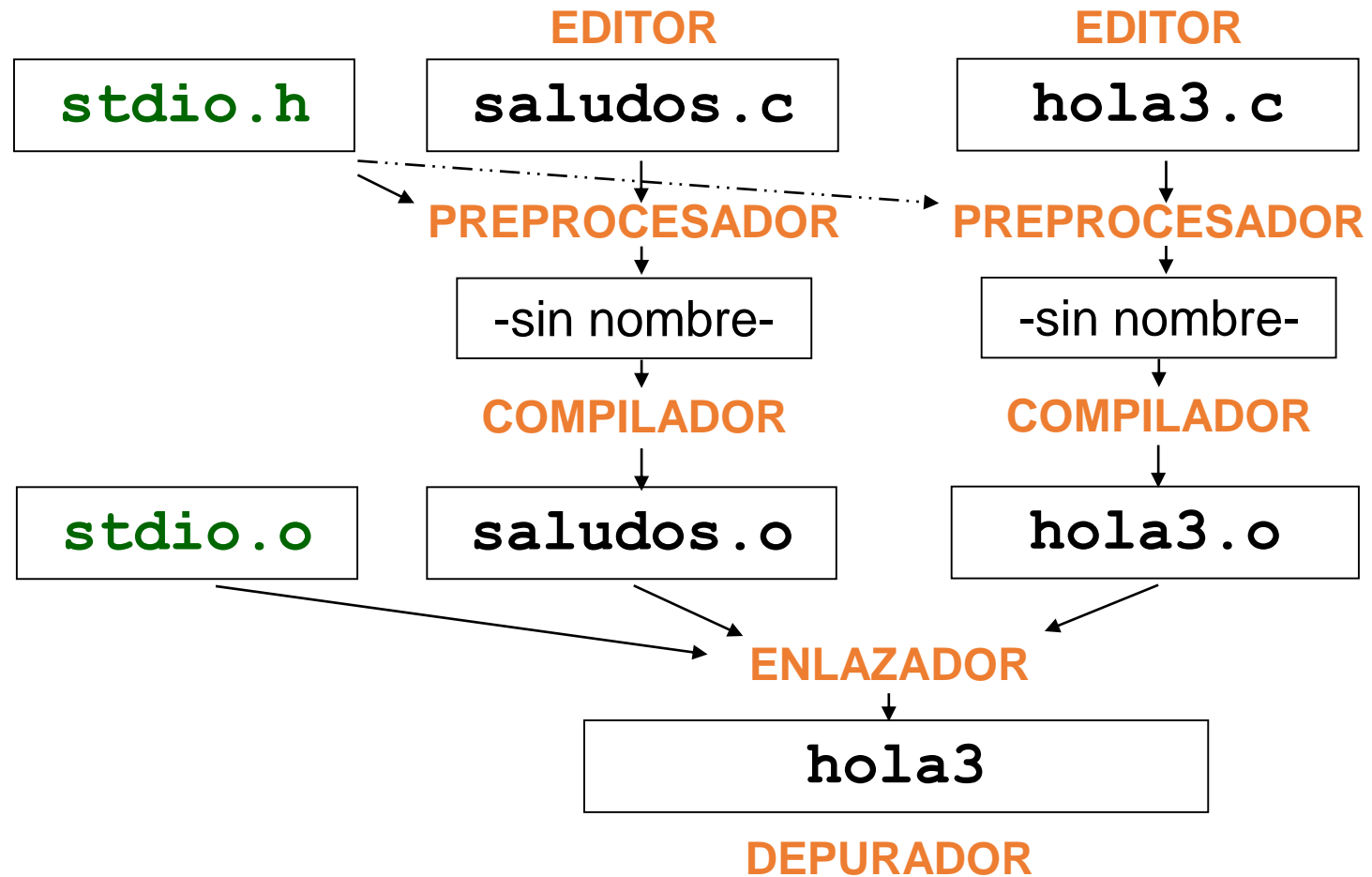
```
/* hola3 (módulo ppal) */  
  
void hola (void);  
void adios (void);  
  
int main ()  
{  
    hola ();  
    adios ();  
    return 0;  
}
```

## **saludos.c**

```
/* saludos.c (mód sec) */  
  
#include <stdio.h>  
  
void hola (void)  
{ printf ("Hola.\n"); }  
  
void adios (void)  
{ printf ("Adios.\n"); }
```

**gcc hola3.c saludos.c -o hola3**

# Programa con varios módulos



# Estructura de un programa (4)

## hola4.c

```
/* hola4 (módulo ppal) */  
  
#include "saludos.h"  
  
int main ()  
{  
    hola ();  
    adios ();  
    return 0;  
}
```

## saludos.h

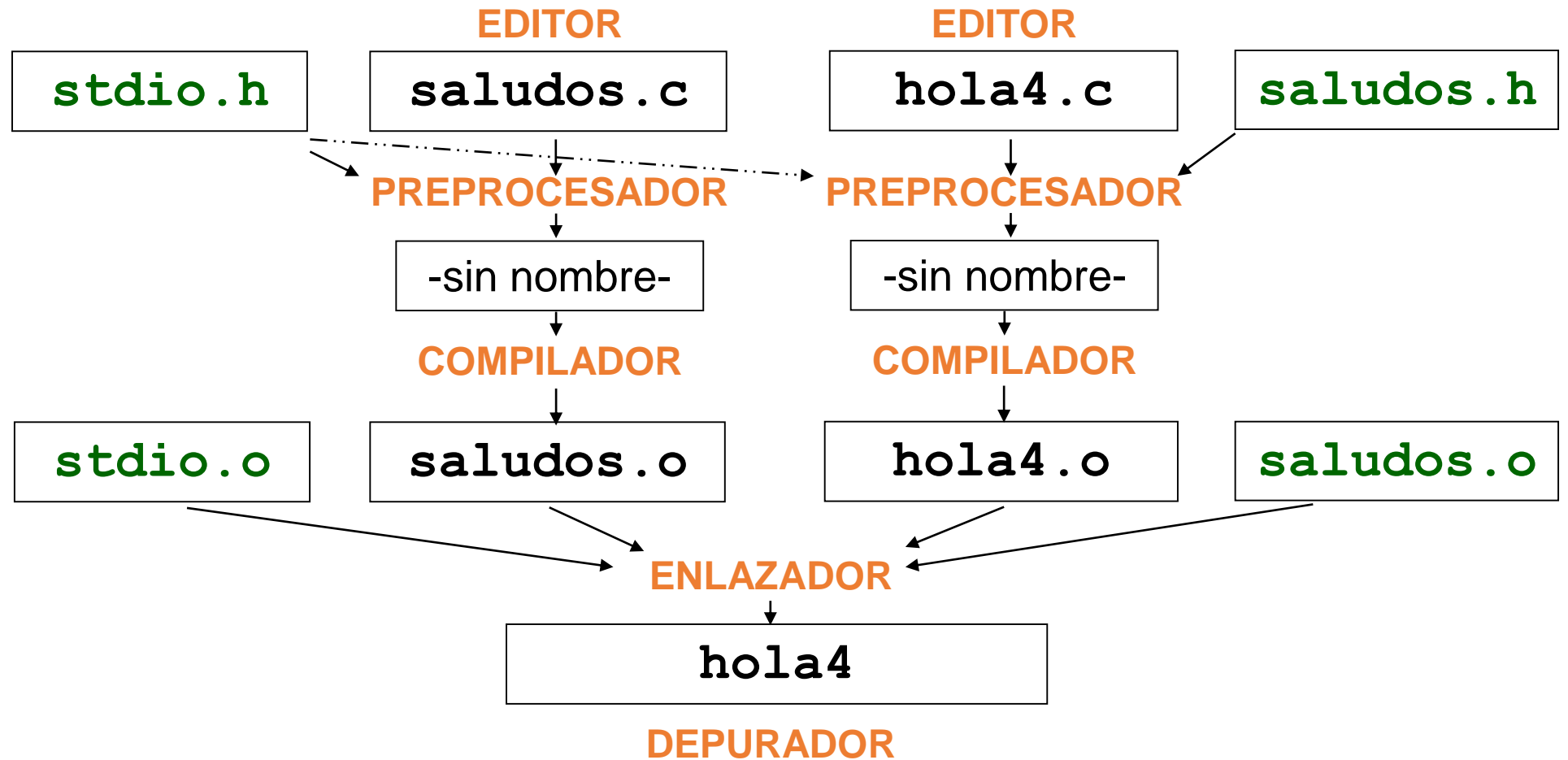
```
/* saludos.h: Cabeceras */  
void hola (void);  
void adios (void);
```

## saludos.c

```
/* saludos.c (mód sec) */  
  
#include <stdio.h>  
  
void hola (void)  
{ printf ("Hola.\n"); }  
  
void adios (void)  
{ printf ("Adios.\n"); }
```

**gcc hola4.c saludos.c -o hola4**

# Programa con varios módulos



# Estructura de un programa (5)

- Directivas (o directrices) para el preprocesador
  - Directivas de inclusión (**#include**)
  - Directivas de sustitución (**#define**)
- Definiciones y/o declaraciones
  - Definición de nuevos tipos de datos (**typedef**)
  - Declaración de las funciones
  - Variables globales
- Función **main()**
- Otras funciones

Hay más:  
**#ifdef ...**

Normalmente  
en los **.h**

