

# KIERAS RULES 2021

## #1 A #5

### Regla #1

Cada módulo con su  
.c/.h



→ Corresponde a  
una funcionalidad  
separada!

### Regla #2

Siempre usar  
"#include  
GUARDS"

### Regla #3

Todas las  
declaraciones  
necesarias para  
usar un módulo →  
Deben aparecer en  
su .h



Este .h se usa para  
acceder al módulo

### Regla #4

El .h sólo contiene  
DECLARACIONES

El .c contiene las  
DEFINICIONES  
( y su inicializac.)

El módulo .c debe  
#incluir el .h

### Regla #5

En un módulo, las  
**variables**  
**globales** para todo  
el programa:



a) Se DECLARAN  
como **extern** en el .h



b) Se DEFINEN e  
INICIALIZAN al  
principio del .c

Ej: en .h va: **extern  
int pepe;**  
en .c va al principio  
**int pepe=0;**

#### REFERENCIA:

[Kieras, 2012], David Kieras, EECS Dept. University of Michigan, 19/12/2012 –  
originalmente escrito para C++, pero utilizado para C indistintamente.

# KIERAS RULES 2021

## #6 A #11

### Regla #6

Las variables, structs o funciones de **uso interno** del .c no deben estar en el .h



→ Se declaran como **static** al principio del .c

Ej.: `int pepe2;`  
Solo usado por el .c se declara al principio  
`static int pepe2 = 0;`  
(el .h ni se entera)  
Idem para funciones

### Regla #7

Cada .h debe **"#incluir"** sólo los otros .h que se requieran para compilar correctamente

→no mas!

### Regla #8

"FORWARD DECLARATION"

Si solo se requiere en header1.h →  
Un único elemento X de header2.h...

.. Se puede en header1.h sólo declarar ese X (y no header2.h completo)

### Regla #9

EI .h DEBERIA COMPILAR CORRECTAMENTE POR SI MISMO!

Armar un test .c e ir **#incluyendo** uno a uno todos los .h



Ver que compilen correctamente

### Regla #10

Un archivo modulo1.c debe **"#incluir"** al principio:



a) modulo1.h



b) Luego cualquier otro xxx.h que modulo1.c requiera

### Regla #11

Nunca! Hacer un **"#include"** de un archivo .c

#### REFERENCIA:

[Kieras, 2012], David Kieras, EECS Dept. University of Michigan, 19/12/2012 – originalmente escrito para C++, pero utilizado para C indistintamente.