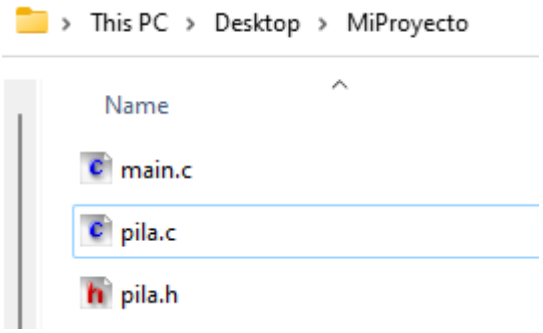
Instrucciones para la instalación de librerías

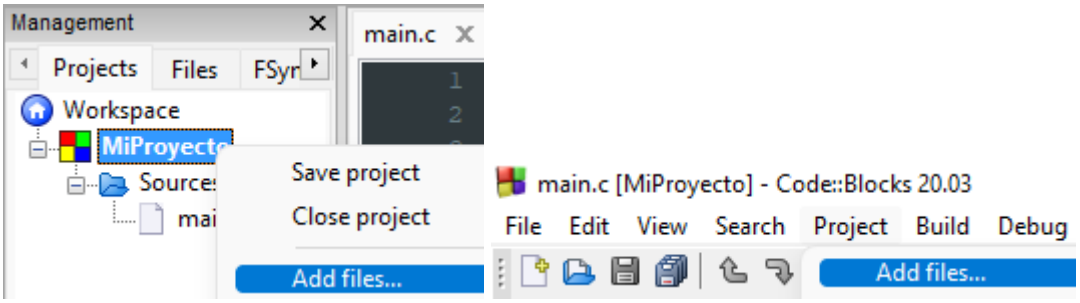
Librería de Pilas

Para instalar la librería de pilas usando Code::Blocks se deben realizar los siguientes pasos:

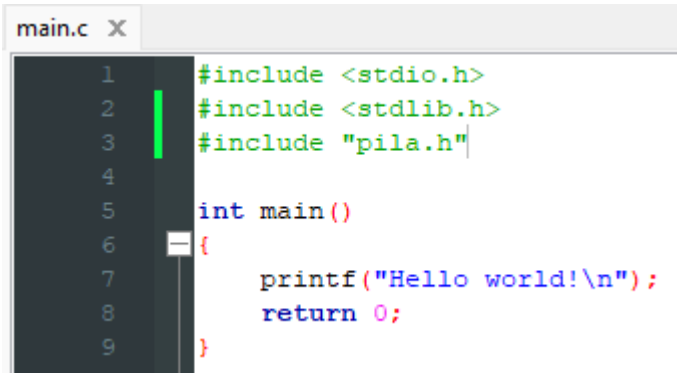
1. Tener un proyecto abierto
2. Copiar los dos archivos (pila.c y pila.h) en la misma carpeta donde se encuentra el main.c del proyecto utilizando el explorador de archivos, no el IDE



3. Hacer click en el menú desplegable Proyect o hacer click derecho en el nombre del proyecto (panel Management, por defecto a la izquierda, se abre con shift + F2) y seleccionar “Add files…”, seleccionar ambos archivos (pila.c y pila.h), darle Open y finalmente Ok



4. En el archivo main.c agregar la directiva de preprocesador `#include "pila.h"`



Información adicional

Agregar los archivos al proyecto le indica al IDE que compile todos los archivos .c, enlazarlos y generar el ejecutable cuando se inicia el proceso de compilación. ¿Y los .h? Estos archivos se compilan indirectamente ya que el preprocesador se encarga de pegar este contenido donde hay un `#include`.

¿Por qué `"pila.h"` y no `<pila.h>`? Los archivos de cabeceras que se indican entre picos son librerías precompiladas, lo que aliviana la carga del compilador y enlazador. Los archivos que se indican entre comillas son aquellos a los que se puede acceder desde el directorio (carpeta) del proyecto. Escribir `"pila.h"` y `"./pila.h"` es lo mismo ya que son dos maneras de ubicar el archivo desde el directorio actual. Si esto carece de sentido no se preocupe, tomará sentido si hace un proyecto con suficientes archivos como para ameritar una estructura de proyecto más compleja (más carpetas).

Cómo crear librerías propias

Puede surgir la motivación de escribir una librería agrupando funciones relacionadas (alta cohesión, bajo acoplamiento) o simplemente alguna función útil de uso recurrente. Para lograrlo necesitará de al menos dos archivos, uno de cabeceras y uno de código fuente.

Si cuenta con un proyecto abierto alcanza con ir a File > New > File… y ahi elegir C/C++ header (luego C/C++ source) y utilizar el botón de los tres puntos y ponerle un nombre al archivo, esto automáticamente rellenará el campo con la ruta absoluta del archivo, por ejemplo “C:\Users\...\MiProyecto\utiles.h”.

En la misma pantalla donde se indica el nombre y ruta del archivo hay tres casillas de confirmación (que deben estar tildadas) y un campo que indica el header guard word, no importa qué diga siempre y cuando no esté vacío, el IDE se encargará de generar un macro de preprocesador que evitará las múltiples definiciones de una librería. ¿Cuándo ocurre esto? Cuando hay un proyecto suficientemente grande como para que más de un archivo de código fuente incluya las utilidades de una librería propia, suponga que existe un `#include "pila.h"` en main.c y también en alumno.c, por dar un ejemplo.

Sólo queda hacer lo mismo con el archivo de código fuente (proceso similar pero más sencillo) e incluir el archivo de cabeceras tanto en el archivo de código fuente de la librería como en el main.c o donde sea que se utilice la librería.

Aclaración: No recomiendo crear una librería sin un proyecto abierto, esto significará que los archivos siendo creados y editados no se podrán compilar o probar hasta incluirlos en otro proyecto, entorpeciendo el desarrollo de la librería.