# Universidad de Costa Rica

# Ciencias de Computación e Informática

# CI-1201 Programación II

# Profesor Adolfo Di Mare

Indentación y documentación de la clase Acumulador

# Gabriel Castañaza B31520

## Introducción

“El objetivo de esta tarea es tomar un programa C++ completo para agréguele la documentación y aprender a compilarlo. Se debe tomar la implementación de la clase Acumulador y darle formato de acuerdo a las convenciones de programación definidas en el curso.”

Debe hacerse una documentación interna y externa. Construir un sitio en Internet donde se suba el código fuente y su documentación.

Para generar la documentación es necesario usar Doxygen y generar un archivo de configuración.

## Descripción del problema

Para realizar esta tarea, es necesario cumplir varios objetivos. Indentar un programa que esta sin indentar y sin ningún salto de línea. Para ello hay que seguir los lineamientos detallados aquí: <http://www.di-mare.com/adolfo/p/Indentacion.htm>

A ese mismo programa hay que agregarle la documentación y generar un sitio html con esa documentación usando el programa Doxygen.

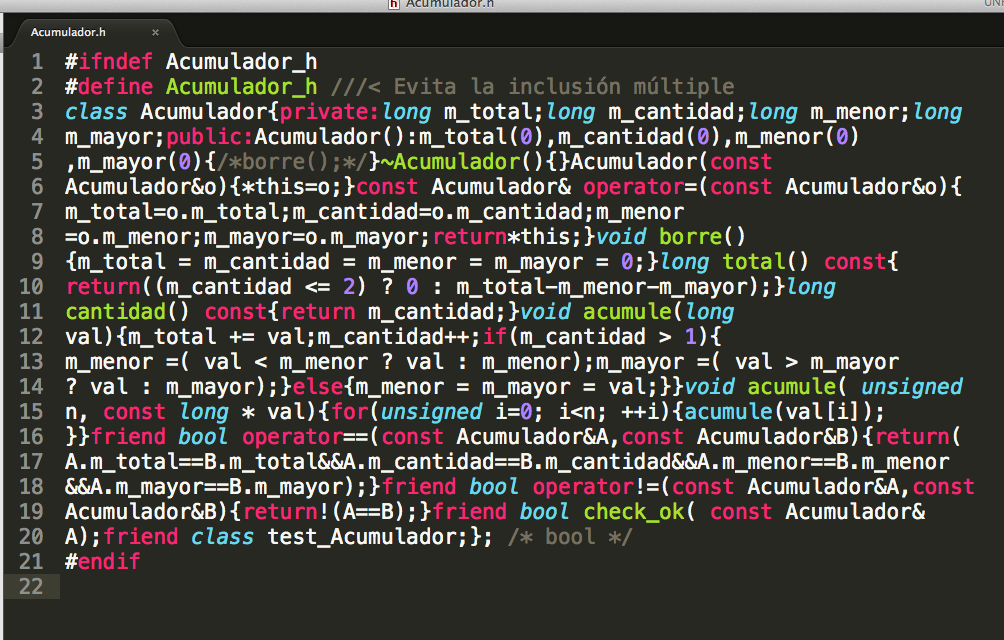
Además hay que compilar el programa C++ y ejecutarlo, así como entender su funcionamiento para escribir una guía de uso.

Para poder completar estos objetivos, hay que descargar el .zip con los archivos de la tarea

<http://www.di-mare.com/pub/Pg2-T1/>

## Indentar el Programa C++

Se nos pide indentar un programa que viene sin formato, el nombre del archivo es “Acumulador.h”.



Para ello, hay varias maneras de lograrlo algunas de ellas vistas en clase, voy a detallar una de esas maneras, pero usando otro software. Voy a usar otro software, debido a que la computadora que uso, tiene Mac OS. Además, este software es el que uso regularmente para programar.

El programa que utilicé para realizar se llama “Sublime Text 2” que entre una de sus ventajas esta, manejar expresiones regulares, para buscar y reemplazar texto:



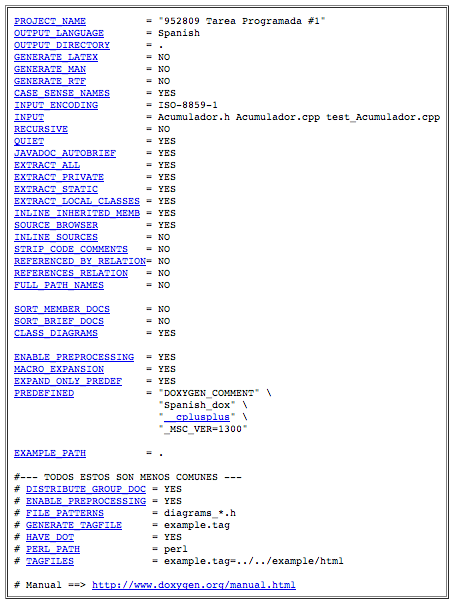
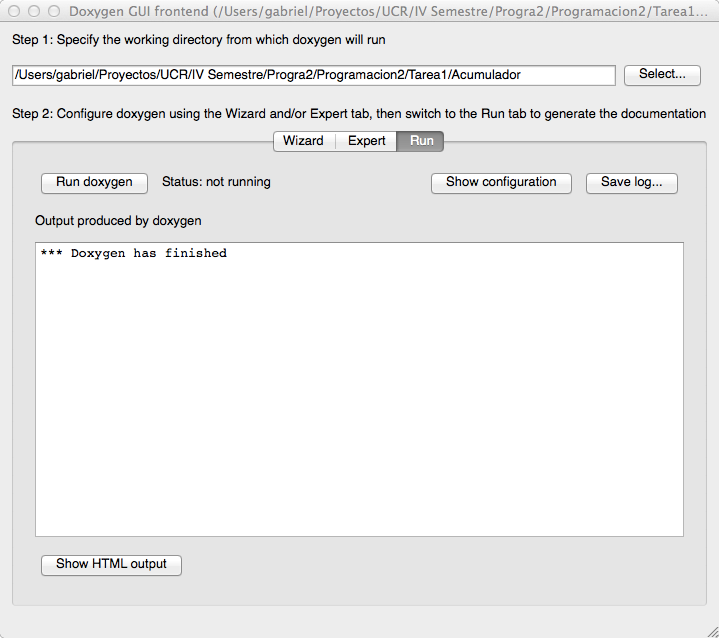
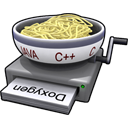
Lo que permite buscar todos los punto y coma ( finales de línea ) y reemplazarlos agregando un salto de línea (\n). Además hay que repetir el proceso con la llave de inicio y de final ( {} ).

Este proceso da como resultado, un código que puede indentarse con mas sencillez. Basta con seguir estas simples reglas:

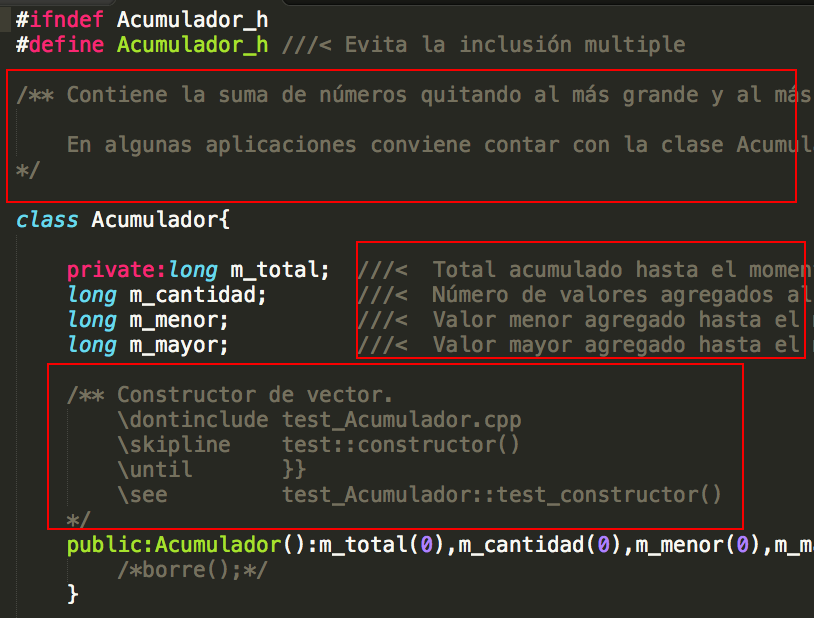
* Siempre se indenta 4 espacios el código encerrado entre corchetes '{' y '}'.
* El primer corchete está en el mismo renglón que la instrucción.
* A partir del último corchete '}' se termina la indentación.

## Escribir la Documentación

Para escribir la documentación del programa, hay que tomar en cuenta el uso del programa Doxygen, el cual contiene muchas configuraciones, pero para simplificar la tarea, el profesor nos proveyó un archivo de configuración.



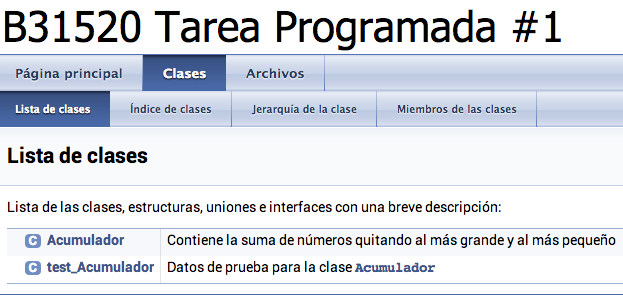
Este archivo de configuración (Acumulador.dxg), debe cargarse en el Doxygen para poder generar la documentación. Hay que entrar en la pestaña “Run” y presionar el botón “Run doxygen” Esto generará una carpeta llamada “html” con toda la documentación del programa, tomando como fuente los archivos: Acumulador.h, Acumulador.cpp, test\_Acumulador.cpp. De los cuales hay que completar la documentación siguiendo los lineamientos de doxygen en el archivo “Acumulador.h”



Como se aprecia en la imagen, hay que tomar en cuenta lo siguiente:

* Que cada clase tenga un ejemplo BUnit
* Que cada método tenga una descripción corta
* Los métodos tengan una descripción completa
* Cada método tenga un ejemplo BUnit

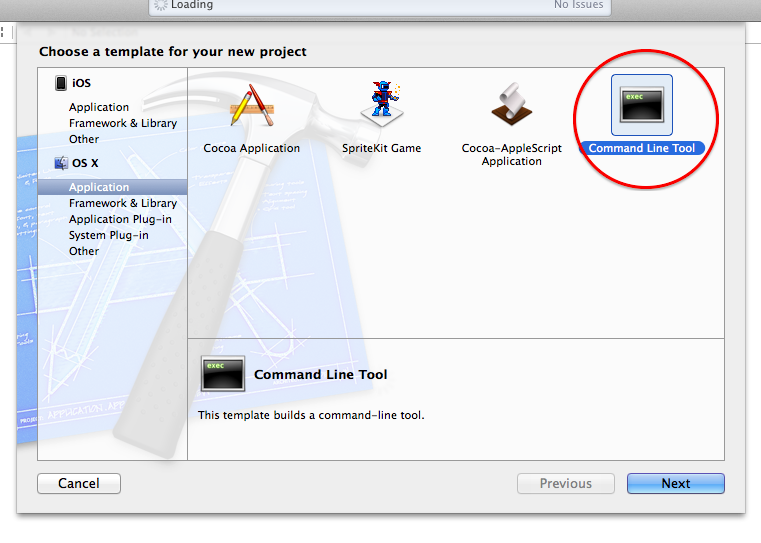
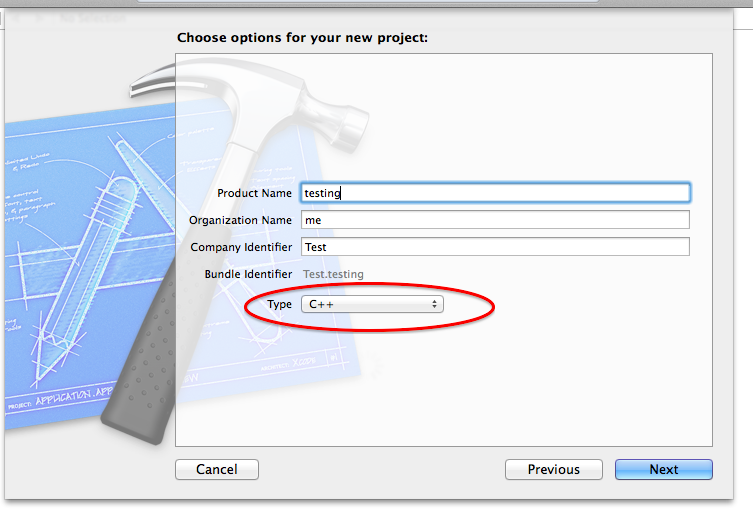
Luego de agregar estas descripciones, lo que doxygen genera es un sitio html como este:



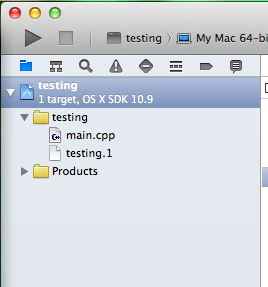
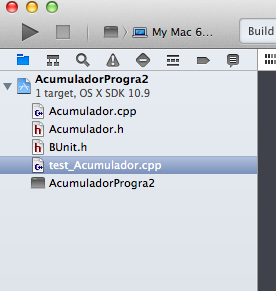
## Compilar y ejecutar el programa

Debido a que mi computadora tiene Mac OS y el programa visto en clases (CodeBlocks) no se instalo correctamente, para compilar C++ la forma que queda es usar el IDE por excelencia de Mac OS: XCode.

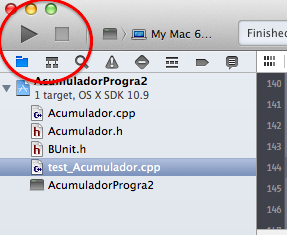
Lo primero que hay que hacer es hacer un proyecto nuevo del tipo “Command Line Tool”

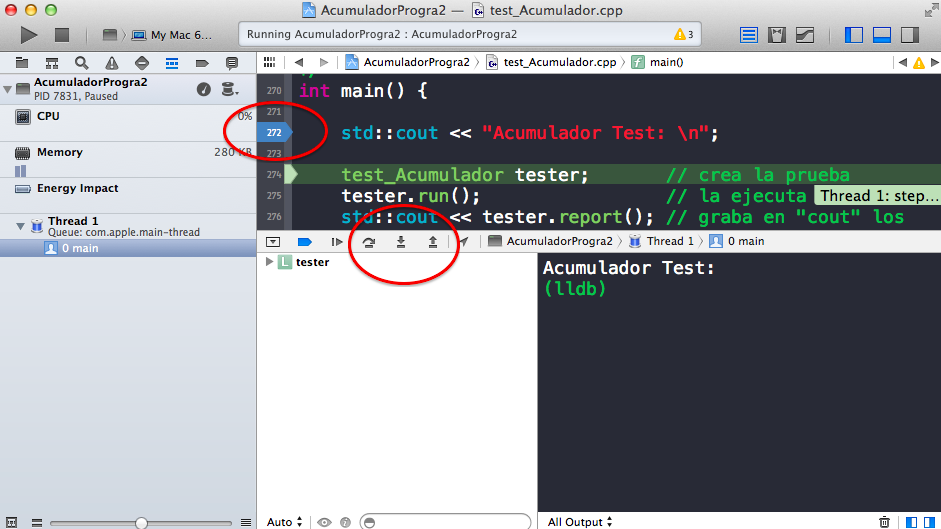
 

Luego escoger como lenguaje “C++”. Esto genera algunos archivos que hay que eliminar. Para agregar los archivos del proyecto que descargamos.

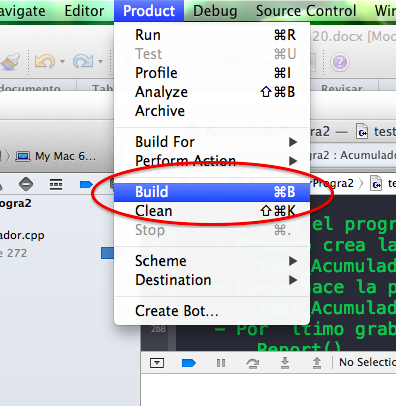
 

Pero es necesario reemplazar el archivo “BUnit.h” por el que se tiene que descargar de la página <http://www.di-mare.com/adolfo/p/BUnit/BUnit.zip>

Luego hay que ejecutar el programa, colocando un breakpoint en el método main(). Para ver la ejecución del programa paso a paso. Y para ello se hace uso de los botones: Step over, step into, step out.



Para solamente compilar el programa, se hace con la opción “comando+B”, o el menú Product > Build



## Documentación

La clase Acumulador es una clase contenedora que guarda la suma de números quitando al más grande y al más pequeño.

En algunas aplicaciones conviene contar con la clase Acumulador que sirve para sumar un montón de valores numéricos, pero eliminando del total tanto al más grande como al más pequeño de todos los valores. Por ejemplo, el acumulador aplicado a los valores (1,5,2,2,3,3,4,5) sólo sumaría 5+2+2+3+3+4 y eliminaría el 1 y alguno de los 5's, que son los valores el menor y el mayor de toda la secuencia.

La documentación generada por Doxygen, se encuentra en el siguiente enlace: <http://soygcm.github.io/Programacion2/Tarea1/B31520/html/>