

COBERTURA DE CÓDIGO

COBERTURA DE CÓDIGO

La cobertura de código es una medida de la cantidad de líneas, declaraciones o bloques del código que se prueban utilizando un conjunto de pruebas automatizadas.

COBERTURA DE CÓDIGO - EJEMPLO A ANALIZAR

- Volvamos al problema de Sofía:

Sofía está jugando un juego. Tiene una bolsa con bolitas y mete la mano para sacar un puñado. Gana puntos según la cantidad de bolitas con las siguientes reglas:

- Si la cantidad de bolitas es menor que 10, gana dos puntos por cada bolita que sacó. Si no, un punto por cada una.
- Además, si la cantidad de bolitas que sacó es múltiplo de 3, gana 10 puntos. Si no, pierde 10 puntos.

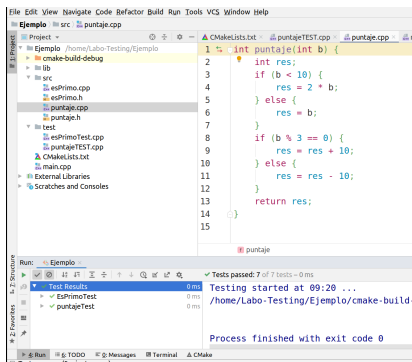
¿Cuántos puntos ganó Sofía?

¿Cómo garantizar la cobertura de líneas código de un proyecto de software?

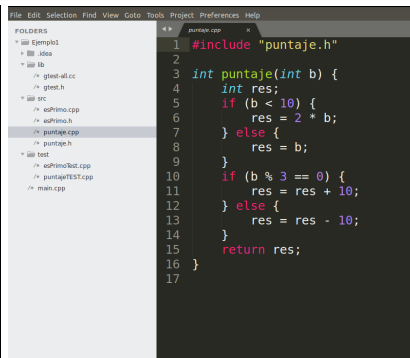
- Utilizando tests de caja Blanca, estos tests se basan en la estructura del código por eso nos servirán para nuestro propósito.

COBERTURA DE CÓDIGO

Primero implementar la solución y los Test de Caja Blanca



(A) CLion



(B) Sublime

LCOV

¿Qué herramienta utilizamos para medir la cobertura de líneas código de un proyecto de software?

- GCC gcov recopila datos para múltiples archivos de origen y crea páginas que contienen el código fuente marcado con información de cobertura.
- LCOV es una interfaz gráfica para la herramienta de prueba de cobertura de GCC gcov .

COMO UTILIZAR LCOV

LCOV necesita archivos .gcno y .gcda.

- Los archivos .gcno se generan en tiempo de compilación
- Los archivos .gcda se generan en tiempo de ejecución.

Para que nuestros proyectos generen los archivos necesitamos agregar FLAGS en la compilación.

COMO UTILIZAR LCOV - PRIMER PASO: AGREGAR FLAGS

- En CLion

Agregar las siguientes lineas al archivo CMakeList.txt del proyecto en caso de no tenerlas.

```
set_target_properties(mi_proyecto PROPERTIES COMPILE_FLAGS "-g --coverage")  
set_target_properties(mi_proyecto PROPERTIES LINK_FLAGS "-g --coverage")
```

De esta manera, cuando se compile el proyecto con tests, se generarán archivos con información de cubrimiento con extensión .gcno y cuando se corran los tests se generarán los archivos .gcda dentro de la carpeta de compilación del proyecto (cmake-build-debug)

COMO UTILIZAR LCOV - PRIMER PASO: AGREGAR FLAGS

- Desde la terminal

```
g++ -o NOMBRE lib/* src/ARCHIVO.h src/ARCHIVO.cpp test/*  
main.cpp -pthread -g --coverage
```

- Agregamos los FLAGS -g --coverage los cuales serviran para crear los archivos .gcno y .gcda
- Cuidado con el orden en que se pasan los archivos del código implementado (en este caso se encuentran dentro de src), para generar la información de cobertura es necesario poner los archivos .h antes que los correspondientes .cpp, de no hacerlo podrían generarse archivos .gcno vacíos.

De esta manera, cuando se compile el proyecto con tests, se generarán archivos con información de cubrimiento con extensión .gcno y cuando se corran los tests se generarán los archivos .gcda dentro de la carpeta de compilación del proyecto

EJECUTAR LCOV - SEGUNDO PASO: OBSERVAR COBERTURA

El siguiente paso es igual para los .gcno y los .gcda generados desde CLion y los generados compilando desde la terminal.

Desde la terminal ejecutar:

- `lcov --capture --directory A --output-file B`
- `genhtml B --output-directory C`

en donde

- *A* representa el directorio en donde se hayan generado los archivos .gcno y .gcda
- *B* representa el nombre de un archivo en donde querramos guardar la información que lcov genera.
- *C* representa un nombre de carpeta que el comando genHTML creará y en donde podrán encontrar un archivo `index.html` que contiene la información de cubrimiento.

EJEMPLO DE HTML GENERADO

- Informe salido del ejemplo de código

LCOV - code coverage report					
Current view: top level					
Test: cobertura					
Date: 2020-06-06 19:07:23					
			Hit	Total	
			Lines: 1221	4504	
			Functions: 793	2363	
Directory	Line Coverage		Functions		
/home/Labo-Testings/Ejemplo1	100.0 %	4 / 4	100.0 %	3 / 3	
/home/Labo-Testings/Ejemplo1/lib	26.1 %	892 / 3417	25.0 %	348 / 1395	
/home/Labo-Testings/Ejemplo1/src	94.1 %	16 / 17	100.0 %	6 / 6	
/home/Labo-Testings/Ejemplo1/test	100.0 %	35 / 35	100.0 %	32 / 32	
?	16.2 %	6 / 37	11.8 %	4 / 33	
?/bits	26.3 %	254 / 965	31.3 %	363 / 1158	
?/ext	48.3 %	14 / 29	25.7 %	39 / 152	
Generated by: LCOV version 1.13					