



Ingeniería en Computadores

Algoritmos y Estructuras de Datos II

Tarea Extra clase 2

Profesor:

Jose Isaac Ramírez Herrera

Estudiantes:

Jose Ignacio Granados Marín (2018319698)

Juan Ignacio Navarro Navarro (2019039662)

IIS-2020

En este documento se presentan los principales resultados que se obtuvieron con de la implementación de pruebas unitarias, pruebas de integración y análisis de cobertura de código en la Tarea Extra clase 1. Es importante mencionar que el IDE utilizado fue Qt Creator por lo que se seleccionó el propio framework de dicho entorno de desarrollo para la implementación de las pruebas. Dicha herramienta es conocida como QTest.

Todas las pruebas unitarias y pruebas de integración implementadas se encuentran en el proyecto con dirección en el repositorio de Server/qtest. Y los resultados del análisis de cobertura de código están localizados en Server/coverage/html/index.html.

A continuación, se presentarán los resultados obtenidos:

1. Pruebas unitarias (Server/qTest/Source/tst\_testing.cpp):

```
//Unit testing method implementation

void Testing::test_actualEdges() {

    QVector<int> result = graph.actualEdges(graph.matrix, 0);

    QCOMPARE(result, actualEdgesNode);

}

void Testing::test_findEdge() {

    bool result = graph.findEdge(actualEdgesNode, 3);

    QCOMPARE(result, true);

}

void Testing::test_findDistance() {

    int distance = 5;

    int result = graph.findDistance(graph.matrix, 0, 3);

    QCOMPARE(result, distance);

}
```

```

void Testing::test_createMatrix() {
    QVector< QVector<int> > resultFloydWarshallMatrix = graph.createMatrix(graph.matrix, nodes, infinite);
    QCOMPARE(resultFloydWarshallMatrix, floydWarshallMatrix);
}

void Testing::test_printPath() {
    QString pathString = "3";
    QString result = graph.printPath(path, 0, 2);
    QCOMPARE(result, pathString);
}

void Testing::test_printSolution02() {
    QString result = graph.printSolution(path, nodes, 0, 2);
    QCOMPARE(result, message02);
}

void Testing::test_floydWarshall() {
    QString result = graph.floydWarshall(floydWarshallMatrix, nodes, infinite, 0, 2);
    QCOMPARE(result, message02);
}

```



## 2. Resultados de las pruebas unitarias del server:

```

17:28:31: Starting /home/juan/Documents/Tarea2-Datos2-UnitTesting/Datos-2-Tarea-1-Sockets-Punteros/build-qTest-Desktop_Qt_5_15_1_GCC_64bit-
Debug/qTest ...
***** Start testing of Testing *****
Config: Using QtTest library 5.15.1, Qt 5.15.1 (x86_64-little_endian-lp64 shared (dynamic) release build; by GCC 5.3.1 20160406 (Red Hat
5.3.1-6))
PASS : Testing::initTestCase()
PASS : Testing::test_actualEdges()
PASS : Testing::test_findEdge()
PASS : Testing::test_findDistance()
PASS : Testing::test_createMatrix()
PASS : Testing::test_printPath()
PASS : Testing::test_printSolution02()
PASS : Testing::test_floydWarshall()
PASS : Testing::test_printSolution31()
PASS : Testing::test_printSolution01()
PASS : Testing::test_printSolution03()
PASS : Testing::cleanupTestCase()
Totals: 12 passed, 0 failed, 0 skipped, 0 blacklisted, 0ms
***** Finished testing of Testing *****





```

### 3. Resultados de la cobertura del proyecto:

LCOV - code coverage report					
Current view: top level		Hit		Total	Coverage
Test: coverage-filtered.info		Lines:		151	159 95.0 %
Date: 2020-10-15 17:32:11		Functions:		27	29 93.1 %
Directory	Line Coverage			Functions	
/usr/include/c++/9		85.7 %	6 / 7	88.9 %	8 / 9
Server		95.4 %	145 / 152	95.0 %	19 / 20

Generated by: [LCOV version 1.14](#)

### 4. Resultados de la cobertura del server:

LCOV - code coverage report					
Current view: <a href="#">top level</a> - Server		Hit		Total	Coverage
Test: coverage-filtered.info		Lines:		145	152 95.4 %
Date: 2020-10-15 17:32:11		Functions:		19	20 95.0 %
Filename	Line Coverage			Functions	
graph.cpp		93.8 %	90 / 96	100.0 %	8 / 8
main.cpp		100.0 %	5 / 5	100.0 %	1 / 1
serverclass.cpp		100.0 %	12 / 12	100.0 %	3 / 3
widget.cpp		97.4 %	38 / 39	87.5 %	7 / 8

Generated by: [LCOV version 1.14](#)

### 5. Pruebas de integración (Server/qTest/Source/tst\_testing.cpp):

```
//Métodos de las pruebas de integración

void Testing::test_printSolution31(){
    QString result = graph.printSolution(path, nodes, 3, 1);

    QCOMPARE(result, message31);
}

void Testing::test_printSolution01(){
    QString result = graph.printSolution(path, nodes, 0, 1);

    QCOMPARE(result, message01);
}

void Testing::test_printSolution03(){
    QString result = graph.printSolution(path, nodes, 0, 3);

    QCOMPARE(result, message03);
}
```