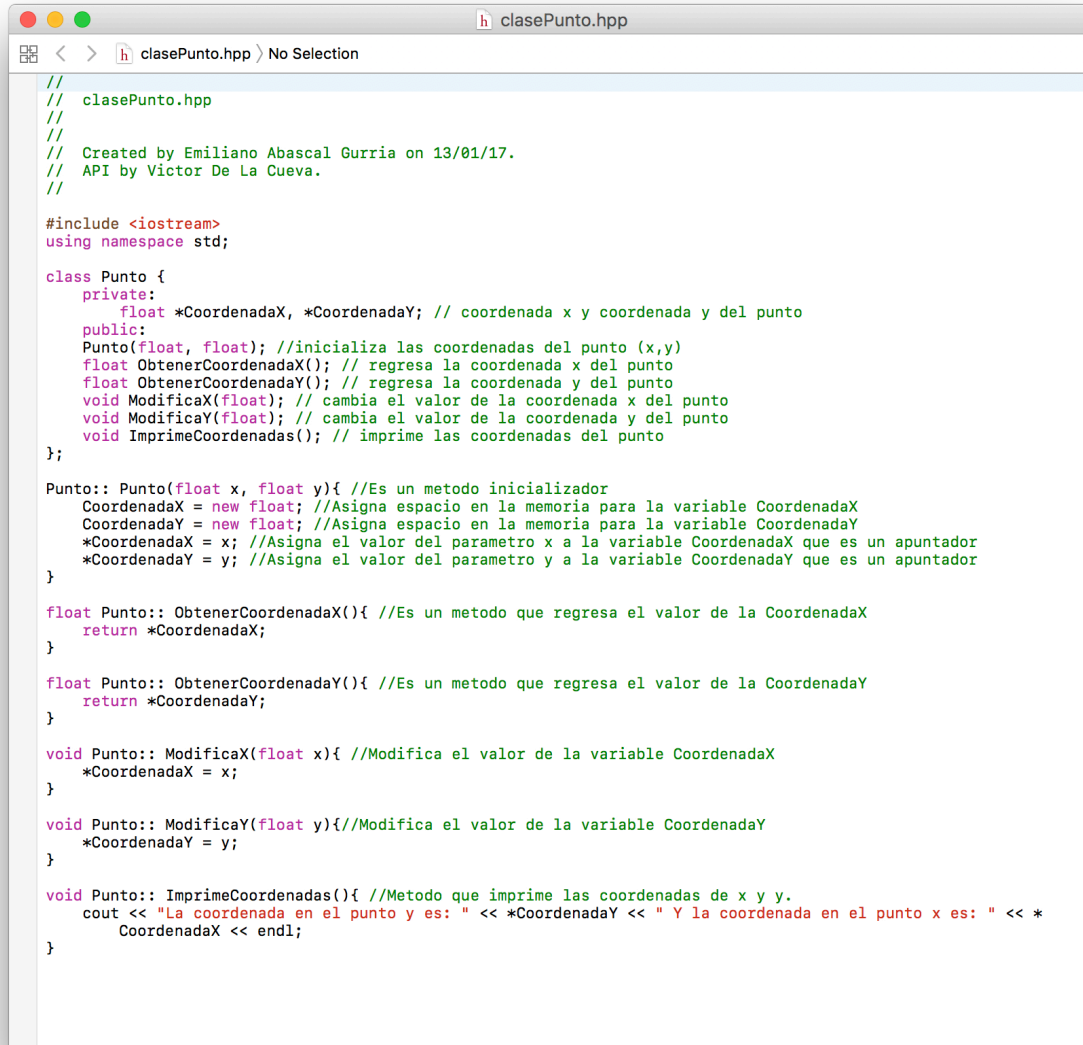


Documentación "Merge Sort"

Emiliano Abascal Gurría

A01023234

Instituto Tecnológico de Estudios Superiores de Monterrey
Campus Santa Fe



```
//
// clasePunto.hpp
//
// Created by Emiliano Abascal Gurría on 13/01/17.
// API by Victor De La Cueva.
//

#include <iostream>
using namespace std;

class Punto {
private:
    float *CoordenadaX, *CoordenadaY; // coordenada x y coordenada y del punto
public:
    Punto(float, float); //inicializa las coordenadas del punto (x,y)
    float ObtenerCoordenadaX(); // regresa la coordenada x del punto
    float ObtenerCoordenadaY(); // regresa la coordenada y del punto
    void ModificaX(float); // cambia el valor de la coordenada x del punto
    void ModificaY(float); // cambia el valor de la coordenada y del punto
    void ImprimeCoordenadas(); // imprime las coordenadas del punto
};

Punto::Punto(float x, float y){ //Es un metodo inicializador
    CoordenadaX = new float; //Asigna espacio en la memoria para la variable CoordenadaX
    CoordenadaY = new float; //Asigna espacio en la memoria para la variable CoordenadaY
    *CoordenadaX = x; //Asigna el valor del parametro x a la variable CoordenadaX que es un apuntador
    *CoordenadaY = y; //Asigna el valor del parametro y a la variable CoordenadaY que es un apuntador
}

float Punto::ObtenerCoordenadaX(){ //Es un metodo que regresa el valor de la CoordenadaX
    return *CoordenadaX;
}

float Punto::ObtenerCoordenadaY(){ //Es un metodo que regresa el valor de la CoordenadaY
    return *CoordenadaY;
}

void Punto::ModificaX(float x){ //Modifica el valor de la variable CoordenadaX
    *CoordenadaX = x;
}

void Punto::ModificaY(float y){ //Modifica el valor de la variable CoordenadaY
    *CoordenadaY = y;
}

void Punto::ImprimeCoordenadas(){ //Metodo que imprime las coordenadas de x y y.
    cout << "La coordenada en el punto y es: " << *CoordenadaY << " Y la coordenada en el punto x es: " << *
        CoordenadaX << endl;
}
```

Manual de usuario:

- Una vez obtenido el software ubicarlo en un directorio de preferencia, por ejemplo en el escritorio.
- Abrir la terminal o un compilador para correr programas de C++.



```
emilianoabascalgurria — -bash — 80x24
Last login: Mon Jan 16 15:13:54 on ttys000
MacBook-Pro-de-Emiliano:~ emilianoabascalgurria$ cd /Users/emilianoabascalgurria
/Desktop/4to\ Semestre/Estructura\ de\ datos/Programas/Proyecto\ 1
```

- Escribir el siguiente comando: `cd "El directorio donde se encuentra el programa"`. como se encuentra en el siguiente ejemplo:
 - Ya que se está en el directorio se requiere hacer un archivo `main.cpp` en el cual se importara el archivo de la siguiente manera: `#include "Arreglo.hpp"` y se llamarán a las funciones del programa, también se deberá incluir la librería `"string"`.
- Al tener un archivo `main.cpp` regresa a la terminal y escribe el siguiente comando `g++ main.cpp` y da enter.
- Se debe tener un archivo `.txt` del cual se leerán los valores para el arreglo de una de las funciones del programa, la función `"lecturaDatos"` se debe correr antes que todas las otras funciones ya que es la que le da valores al arreglo.
- Si no hay ningún problema, se creara un archivo `".out"` en el directorio donde se encuentran los archivos `"main.cpp"` y `Arreglo.hpp`.
- Escribe en la terminal `./a.out` y el programa se correrá, como se demuestra en al ejemplo a continuación.

```
Proyecto2 — pi@raspberrypi: ~/Desktop — -bash — 80x24
iMacdeEmiliano2:Proyecto2 emilianoabascalgurria$ g++ main.cpp
```

2/2017

```
Proyecto2 — pi@raspberrypi: ~/Desktop — -bash — 80x24
iMacdeEmiliano2:Proyecto2 emilianoabascalgurria$ ./a.out
```

```
Proyecto2 — pi@raspberrypi: ~/Desktop — -bash — 95x23
iMacdeEmiliano2:Proyecto2 emilianoabascalgurria$ ./a.out
Anidacion incorrecta para: '(()))'.
limpiaString para: 'aaaaabbbbccccdddeeeeggggggffffffiiiiiii' es: 'abcdegfi'.
La suma de digitos del numero: 5 es: 15.
El substring 'hola' esta 2 veces en el string 'hola Alejandro, hola Emiliano, soy Victor.'.
iMacdeEmiliano2:Proyecto2 emilianoabascalgurria$
```

Estrutura General

El programa contiene varios métodos pero habrá un enfoque en los siguientes: "mergeSort" y "merge", se llama a travez del archivo main, pero para que funcione debe haber un archivo .txt y a los métodos "lecturaDatos" y "print", los cuales se llamarán desde el archivo main, en el cual se definirán los valores de los parámetros que aceptan.

Algoritmos

- Funcion merge(int ab,int mid, int ar):

```
int x = ab
int y = mid + 1
int z = ab
int arrb[t]
for(int x = ab, x <= ar, x++)
    arrb[x] = array[x]
while (x <= mid && y <= ar)
    if (arrb[x] < arrb[y])
        array[z] = arrb[x]
        z++
        x++
    else
        array[z] = arrb[y]
        z++
        y++
while (x <= mid)
    array[z] = arrb[x]
    z++;
    x++;
while (y <= ar)
    array[z] = arrb[y]
    z++;
    y++;
```

- Funcion mergeSort(int ab, int ar):

```
if(ab < ar)
    int mid = ab + (ar - ab)/2
    mergeSort(mid + 1,ar)
    mergeSort(ab, mid)
    merge(ab, mid, ar)
```

Descripción Técnica

- Funcion void merge(int ab, int mid, int ar):
 - Argumentos: ab, que es la parte inferior del arreglo, mid, que es la mitad, y ar que es la parte superior
 - Variables: x, que es un entero igual a ab, y que es un entero con el valor de mid más uno, z que es un entero igual a ab, arrb, que es un arreglo auxiliar.
 - Funcionalidad: Se copia el valor del arreglo original en el arreglo arrb, si el valor del arreglo arrb en la posicion y es menor al valor del arreglo arrb en la posicion izquierda o (x), entonces el valor del arreglo array en la posicion z tendra el valor del arreglo arrb en la posicion y, si lo anterior no se cumple, el valor del arreglo array en la posición z guardará el valor del arreglo arrb en la posicion x, mientras el tamaño de x sea menor o igual al valor de la mitad del tamaño del arreglo, el valor del arreglo array en la posición z guardará el valor de el arreglo arrb en la posicion x, mientras el tamaño de y sea menor o igual al valor de la ultima posicion del arreglo array, el arreglo array en la posición z guardará el valor de arrb contenido en y.
- Funcion mergeSort(int ab, int ar):
 - Argumentos: ab, que es la parte inferior del arreglo, ar que es la parte superior.
 - Variables: mid que es un entero que es igual a $ab + (ar - ab)/2$
 - Funcionalidad: Se saca la mitad de la dimension del arreglo, se ordena la mitad derecha, llamando su misma funcion, se ordena mitad izquierda, llamando su misma funcion y se hace "Merge Sort" de las dos mitades, llamando la funcion merge.