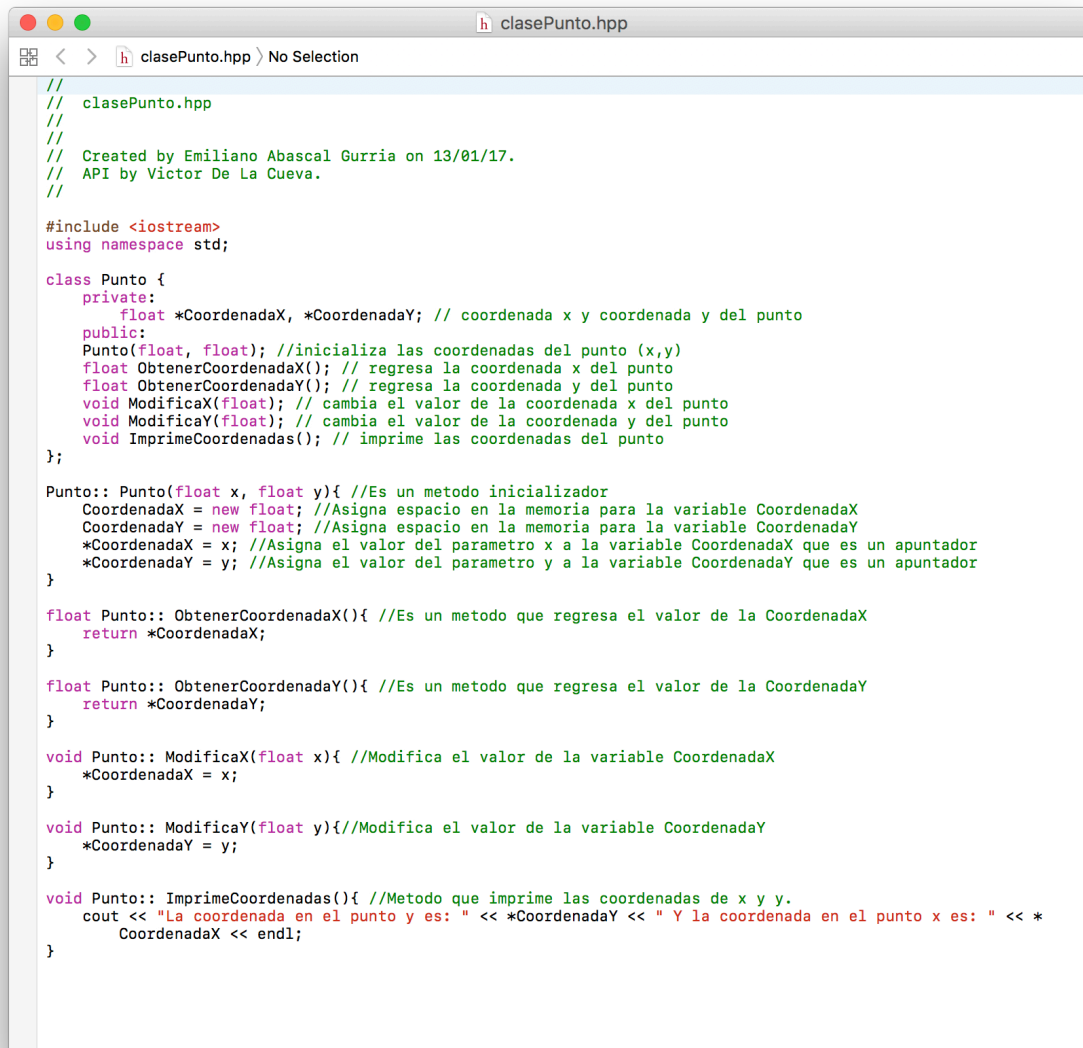


Documentación "Quick Sort"

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(); // retorna la coordenada x del punto
    float ObtenerCoordenadaY(); // retorna 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 retorna el valor de la CoordenadaX
    return *CoordenadaX;
}

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

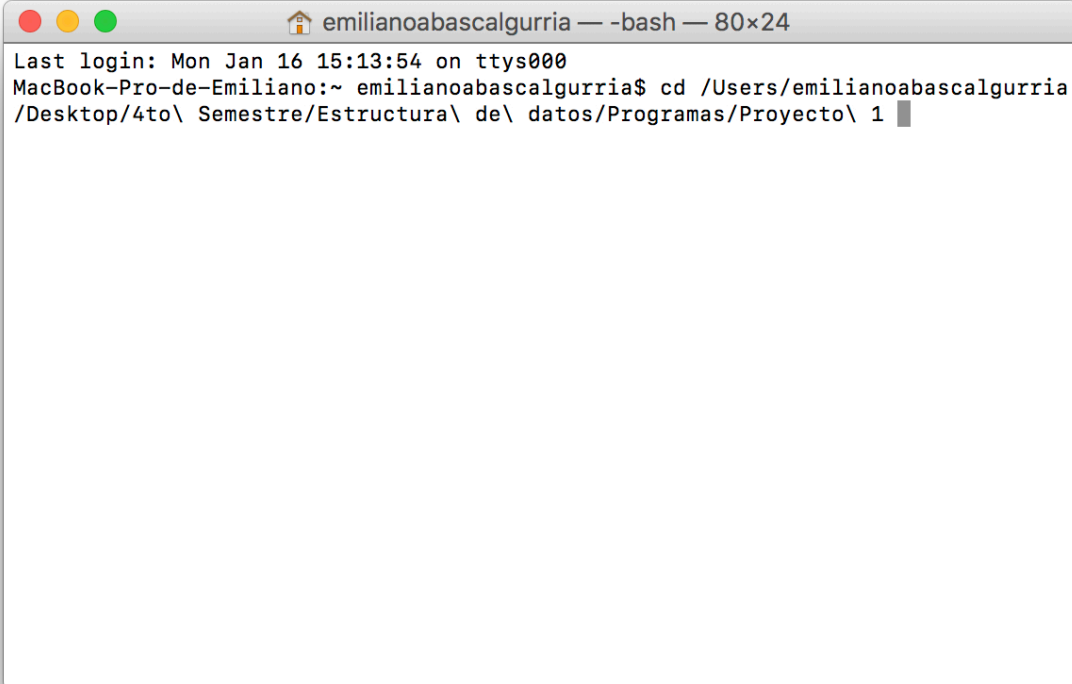
void Punto::ModificaX(float x){ //Modifica el valor de la variable CoordenadaX
    *CoordenadaX = x;
}

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++.
- Escribir el siguiente comando: `cd "El directorio donde se encuentra el programa"`. como se encuentra en el siguiente ejemplo:



```
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
```

- 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
```

03/2017

```
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: "quickSort" y "particiones", se llama a travez del archivo main, pero para que funcione debe haber un archivo .txt y el método "lecturaDatos", el cual se llamará desde el archivo main, en el cual se definirán los valores de los parámetros que acepta.

Algoritmos

- Funcion particion(int lo, int hi):

```
int i = lo
int j = hi
int v = lo
while (i < j)
    while (array[j] > array[v])
        j--
    while (i <= hi && array[i] < array[v])
        i++
    if (i < j)
        swapArr(i,j)
swapArr(v,j)
return j
```

- Funcion quickSort(int lo, int hi):

```
if (hi <= lo)
    return;
else
    int j = particion(lo, hi)
    quickSort(lo, j-1)
    quickSort(j+1, hi)
```

Descripción Técnica

- Funcion void quickSort(int lo, int hi):
 - Argumentos: lo, que es la parte inferior del arreglo, que es la mitad, y hi que es la parte superior
 - Variables: j, que es un entero igual a la llamada de la función partición, con los argumentos (lo, hi).

- Funcionalidad: Si hi es menor o igual a lo entonces se regresa valor vacío, sino, se declara la variable j que va a tener como valor el resultado de la función "particion" y se llama recursivamente la función quickSort con argumentos, lo y $j-1$, y se vuelve a llamar pero con los argumentos $j+1$ y hi .
- Función `particion(int lo, int hi)`:
 - Argumentos: lo , que es la parte inferior del arreglo, hi que es la parte superior.
 - Variables: i de tipo entero que va a ser igual a lo , j de tipo entero que va a ser igual a hi y v que es de tipo entero y es igual a lo , también es el pivote.
 - Funcionalidad: Mientras el valor de i sea menor que el valor de j , se harán estos procesos: Mientras el arreglo en la posición j sea mayor al arreglo en la posición del pivote entonces se reducirá el valor de j en 1, moviéndolo de derecha a izquierda, mientras i sea menor o igual a hi , y el valor del arreglo en la posición i sea menor que el arreglo en la posición del pivote, se aumentará el valor de i en uno, moviendo i de izquierda a derecha. Si el valor de i es mayor al valor de j , entonces se intercambian i por j . Se intercambian el pivote por j y se regresa el valor de j .