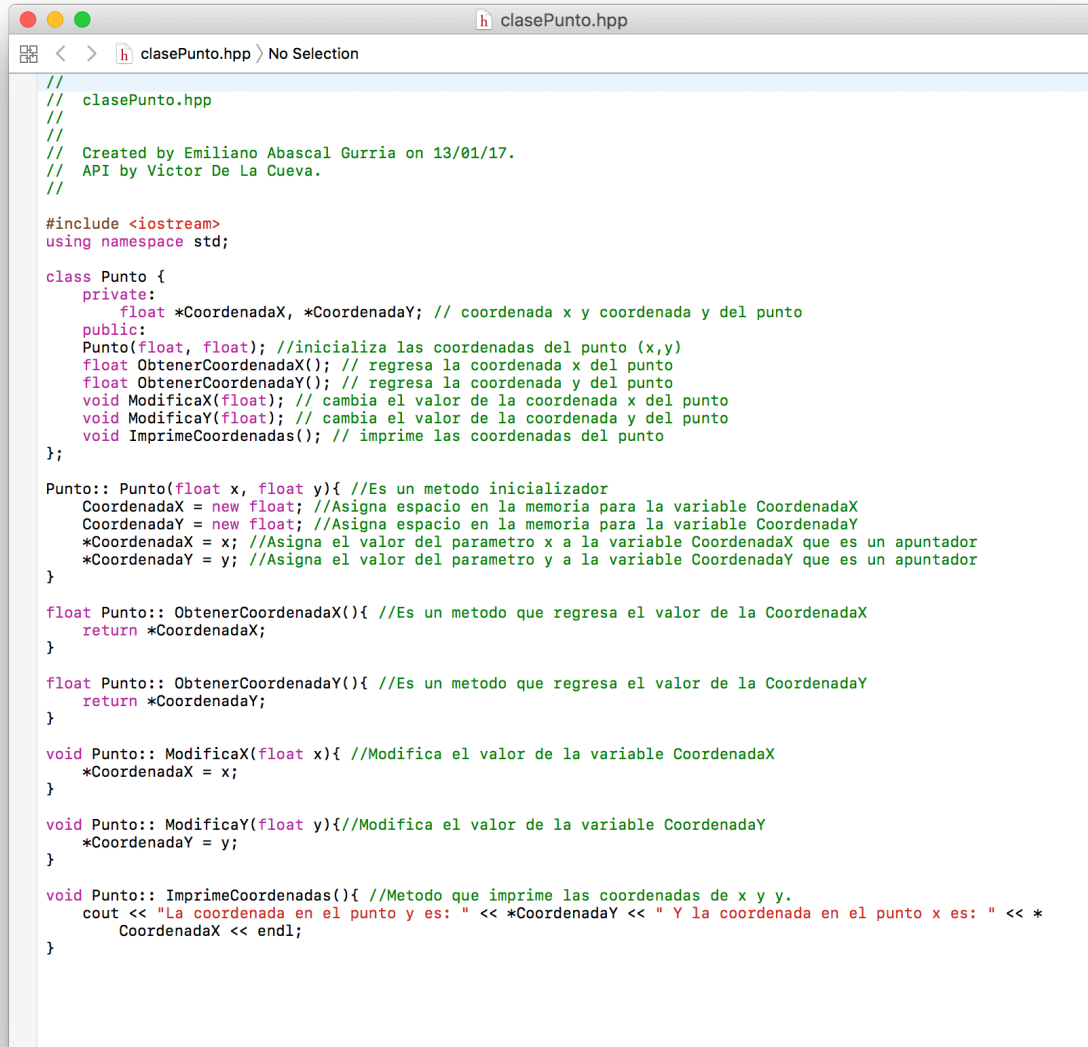


# Documentación Proyecto 1

Emiliano Abascal Gurría

A01023234

Instituto Tecnológico de Estudios Superiores de Monterrey  
Campus Santa Fe



```
//
// clasePunto.hpp
//
// Created by Emiliano Abascal Gurria 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;
}

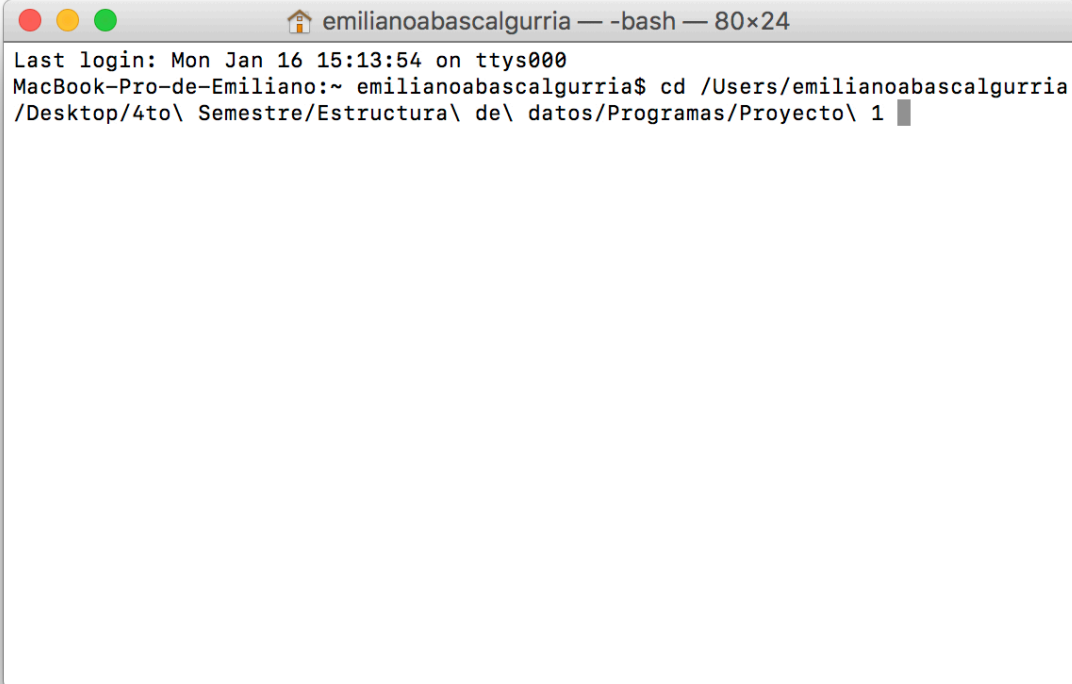
void Punto::ModificaX(float x){ //Modifica el valor de la variable CoordenadaX
    *CoordenadaX = x;
}

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;
}
```

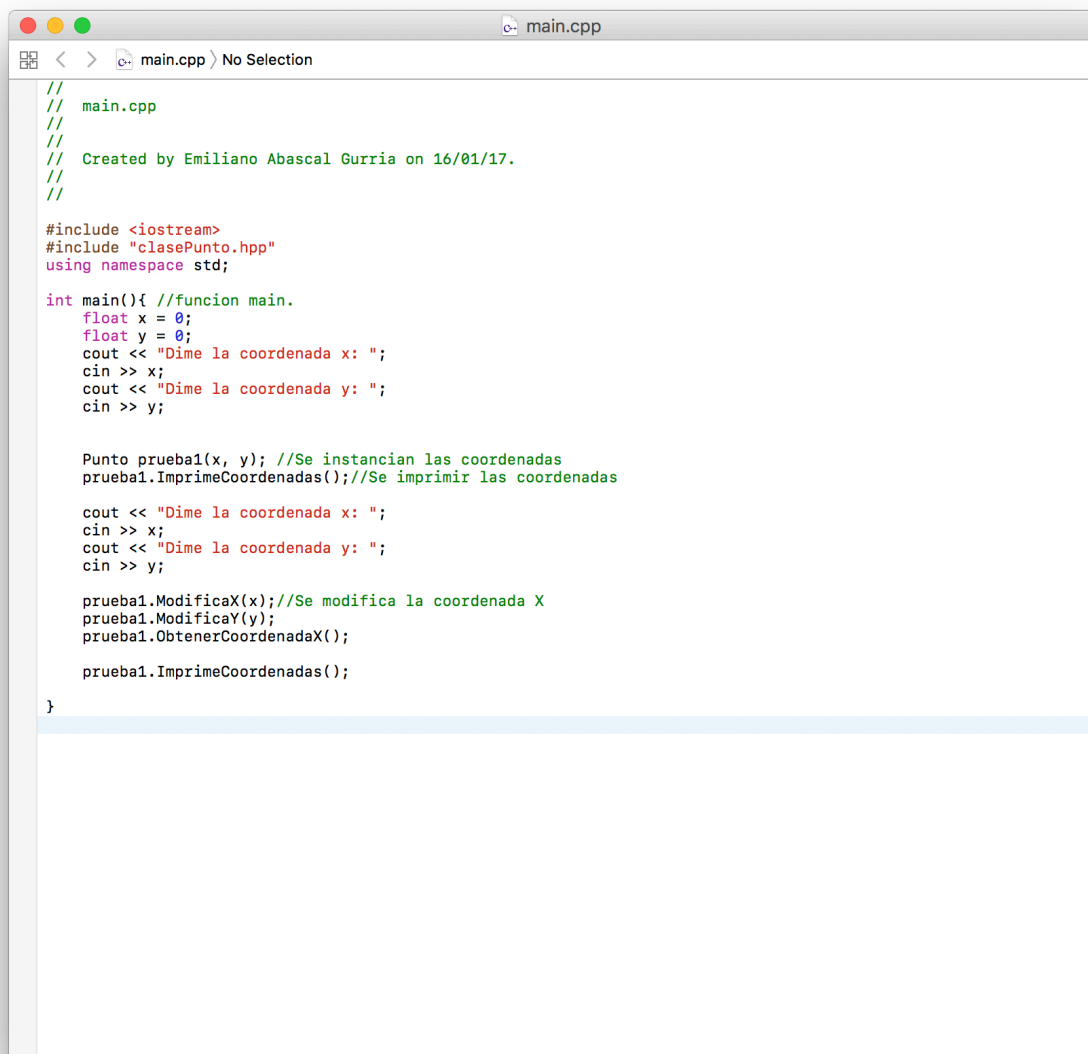
# Manual de usuario:

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



```
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
```

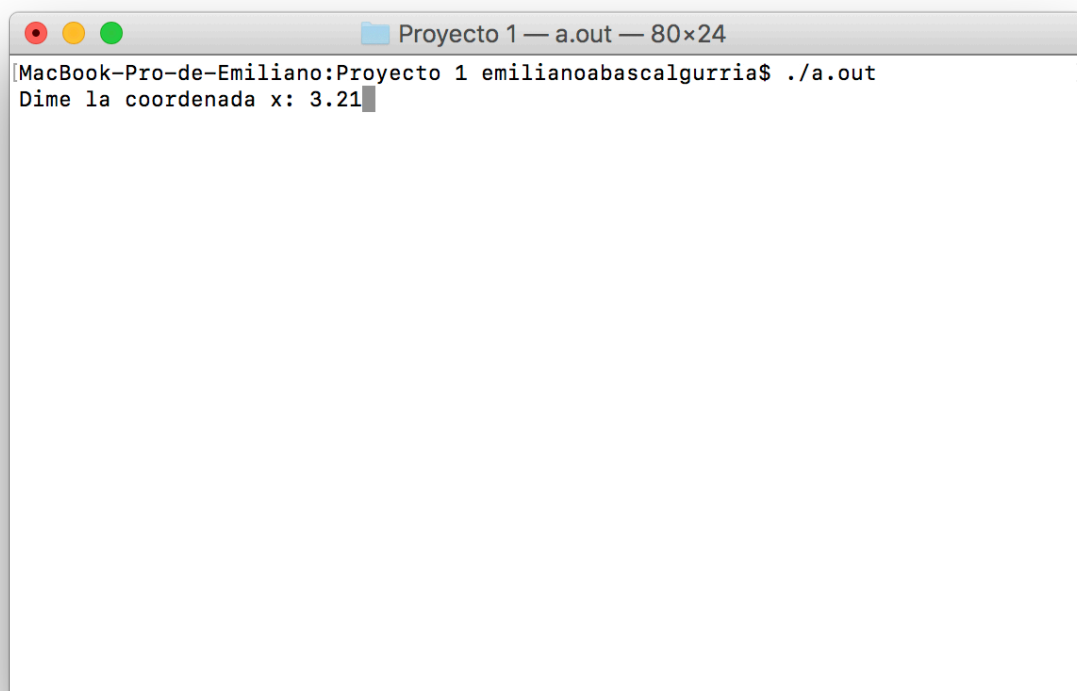
- Ya que se está en el directorio se requiere hacer un archivo `main.cpp` en el cual se importara el archivo de la siguiente manera: `#include "clasePunto.hpp"` y se instanciaran los objetos a continuación un ejemplo.



The screenshot shows a code editor window titled "main.cpp". The code is written in C++ and includes comments in Spanish. It defines a class "Punto" (not fully shown) and a "main" function. The main function initializes variables x and y to 0, prompts the user for coordinates, reads the input, and then uses a "Punto" object to modify and retrieve the coordinates. The code is as follows:

```
//  
// main.cpp  
//  
// Created by Emiliano Abascal Gurria on 16/01/17.  
//  
//  
  
#include <iostream>  
#include "clasePunto.hpp"  
using namespace std;  
  
int main(){ //funcion main.  
    float x = 0;  
    float y = 0;  
    cout << "Dime la coordenada x: ";  
    cin >> x;  
    cout << "Dime la coordenada y: ";  
    cin >> y;  
  
    Punto prueba1(x, y); //Se instancian las coordenadas  
    prueba1.ImprimeCoordenadas(); //Se imprimir las coordenadas  
  
    cout << "Dime la coordenada x: ";  
    cin >> x;  
    cout << "Dime la coordenada y: ";  
    cin >> y;  
  
    prueba1.ModificaX(x); //Se modifica la coordenada X  
    prueba1.ModificaY(y);  
    prueba1.ObtenerCoordenadaX();  
  
    prueba1.ImprimeCoordenadas();  
}
```

- Al tener un archivo main.cpp regresa a la terminal y escribe el siguiente comando g++ main.cpp y da enter.
- Si no hay ningún problema, se creara un archivo ".out".
- Escribe en la terminal ./a.out y el programa se correra, como se demuestra en al ejemplo a continuación.



The screenshot shows a terminal window titled "Proyecto 1 — a.out — 80x24". The prompt is "[MacBook-Pro-de-Emiliano:Proyecto 1 emilianoabascalgurria\$ ]". The user has entered the command "./a.out", and the program has executed, displaying the prompt "Dime la coordenada x: 3.21" with a cursor.

```
Proyecto 1 — a.out — 80x24
MacBook-Pro-de-Emiliano:Proyecto 1 emilianoabascalgurria$ ./a.out
Dime la coordenada x: 3.21
Dime la coordenada y: 6.18
```

```
Proyecto 1 — a.out — 80x24
MacBook-Pro-de-Emiliano:Proyecto 1 emilianoabascalgurria$ ./a.out
Dime la coordenada x: 3.21
Dime la coordenada y: 6.18
La coordenada en el punto y es: 6.18 Y la coordenada en el punto x es: 3.21
Dime la coordenada x: 67.12
```

```
Proyecto 1 — a.out — 80x24
MacBook-Pro-de-Emiliano:Proyecto 1 emilianoabascalgurria$ ./a.out
Dime la coordenada x: 3.21
Dime la coordenada y: 6.18
La coordenada en el punto y es: 6.18 Y la coordenada en el punto x es: 3.21
Dime la coordenada x: 67.12
Dime la coordenada y: 98.1
```

```
Proyecto 1 — -bash — 80x24
MacBook-Pro-de-Emiliano:Proyecto 1 emilianoabascalgurria$ ./a.out
Dime la coordenada x: 3.21
Dime la coordenada y: 6.18
La coordenada en el punto y es: 6.18 Y la coordenada en el punto x es: 3.21
Dime la coordenada x: 67.12
Dime la coordenada y: 98.1
La coordenada en el punto y es: 98.1 Y la coordenada en el punto x es: 67.12
MacBook-Pro-de-Emiliano:Proyecto 1 emilianoabascalgurria$
```

## Algoritmo

1. Inicio.
2. Nueva clase "Punto" con los siguientes métodos:
  1. float ObtenerCoordenadaX()
  2. float ObtenerCoordenadaY()
  3. void ModificaX(float)
  4. void ModificaY(float)
  5. void ImprimeCoordenadas()
  6. Punto(float, float)
3. Y atributos:
  1. float \*CoordenadaX
  2. float \*CoordenadaY
4. Funcionamiento:
  1. Punto(float x, float y):
    1. Se asigna espacio en memoria para CoordenadaX y CoordenadaY.
    2. Se asigna el valor de la variable "x" a \*CoordenadaX.
    3. Se asigna el valor de la variable "y" a \*CoordenadaY.
  2. float ObtenerCoordenadaX():
    1. Regresa el valor de la variable \*CoordenadaX.
  3. float ObtenerCoordenadaY():
    1. Regresa el valor de la variable \*CoordenadaY.
  4. void ModificaX(float x):
    1. Se asigna el valor de la variable "x" a la variable \*CoordenadaX.
  5. void ModificaY(float y):
    1. Se asigna el valor de la variable "y" a la variable \*CoordenadaY.
  6. void ImprimeCoordenadas():
    1. Se imprime en consola: La coordenada en el punto y es: (Valor de \*CoordenadaY) Y la coordenada en el punto x es: (Valor de \*CoordenadaX).
5. El objeto se deberá instansiar en un archivo main.cpp
6. Fin.

# Descripción Técnica

```
Punto:: Punto(float x, float y){ //Es un método 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 parámetro x a la variable CoordenadaX que es un
apuntador
    *CoordenadaY = y; //Asigna el valor del parámetro y a la variable CoordenadaY que es un
apuntador
}

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

float Punto:: ObtenerCoordenadaY(){ //Es un método 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 en consola 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;
}
```