Además debemos incluir el SerialClass.h y el SerialClass.cpp

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <locale.h>
#include <SerialClass.h>
#define MAX_BUFFER 200
#define PAUSA_MS 200


int menu(void)
{
	int opcion;
	printf("\n");
	printf("Menú Principal\n");
	printf("==============\n");
	printf("1 - Encender alarma.\n");
	printf("2 - Regìstro de datos.\n");
	printf("3 - Compobar estado de la conexión.\n");
	printf("4 - Salir de la aplicación\n");
	printf("Opción:");
	scanf_s("%d", &opcion);
	return opcion;
}

int main(void)
{
	Serial* Arduino;
	float distancia;
	char puerto[] = "COM3"; // Puerto serie al que está conectado Arduino
	int opcion_menu;
	setlocale(LC_ALL, "es-ES");
	Arduino = new Serial((char*)puerto);
	do
	{
		opcion_menu = menu();
		switch (opcion_menu)
		{
		case 1:
			printf("\nConectando alarma\n");
			distancia = leer_sensor_distancia(Arduino);
			if (distancia != -1)
				printf("\nDistancia: %f\n", distancia);
			break;
			// permite saber cuando alguien ha pasado por delante del
sensor
			// y de esta manera podemos recoger cuando ha sonado la
alarma
		case 2:
			printf("\Accediendo al registo de datos\n");

			// Incluir un fichero donde almacenar las
			// veces en las que ha sonado la alarma
			// y cuando esta se ha activado o desactivado
			break;
		case 3:
			verifica_sensores(Arduino, puerto);
			// permite saber si se ha realiado la conexion correctamente
con arduino
```

```
                        break;
                case 4:
                default: printf("\nOpción incorrecta\n");
                }
        } while (opcion_menu != 4);
        return 0;
}


void verifica_sensores(Serial* Arduino, char* port)
{
        if (Arduino->IsConnected())
        {
                printf("El estado de la conexión es correcto")
        }
        else
        {
                printf("\nNo se ha podido conectar con Arduino.\n");
                printf("Revise la conexión, el puerto %s y desactive el monitor
serie del IDE de Arduino.\n",port);
        }
}


float leer_sensor_distancia(Serial* Arduino)
{
        float distancia;
        int bytesRecibidos;
        char mensaje_recibido[MAX_BUFFER];

        bytesRecibidos = Enviar_y_Recibir(Arduino, "GET_DISTANCIA\n",
mensaje_recibido);
        if (bytesRecibidos <= 0)
        {
                printf("\nNo se ha recibido respuesta a la petición\n");
                distancia = -1;
        }
        else
        {
                printf("\nLa respuesta recibida tiene %d bytes. Recibido=%s\n",
bytesRecibidos,
                        mensaje_recibido);
                distancia = float_from_cadena(mensaje_recibido);
        }
        return distancia;
}
```