

Analizar un código a través de lenguajes de programación C++ y Code:: Blocks

Analyze a code through C ++ programming languages and Code:: Blocks

Patricio Rigoberto Sangopanta¹ Cajas, Bryan Antonio Mérelo Gil², Edwin Edison Quinatoa Arequipa³.

RESUMEN

El presente artículo trata acerca del análisis a realizar de un código mediante estructuras anidadas de datos, las cuales están agrupadas en datos de tipo enteros, char, flotantes caracterizándose para su organización y las relaciones entre los datos que lo constituyen. Para desarrollar un programa aplicando valores constantes que hace referencia a datos ya permanentes y variables tipo enteros al igual que contadores para obtener valores acumulados y suma total de viajes, como gavetas para n empresas, las cuales se reparte los productos ya especificados en el programa con ello obtener la venta mayor y ordenamiento, como su búsqueda mediante un código autogenerado, hemos tomado un tiempo de ejecución en el programa para lo cual se utilizará el lenguaje de programación C++ y Code: Blocks, ya que es un lenguaje flexible que trabaja con la programación estructurada con la posibilidad de definir archivos de código fuente, y para la debía ejecución del programa se lo realiza en Dev c++ y Code: Blocks. Además del uso de estructuras también se emplea funciones con parámetros y a su vez por valor, por referencia que devuelvan o retornen valores dentro del código. La programación modular se compone de tareas específicas que se declaran independientemente, y estas a su vez buscan ser resueltos por partes, esto trata de dividir el programa en sub-programas que permitan la reducción del código para un mejor entendimiento y de esta forma pueda ser aplicada en cualquier IDE de desarrollo como es el Code: Blocks.

Palabras claves: código, desarrollo, estructuras anidadas, funciones, programa

ABSTRACT:

His article deals with the analysis to be performed of a code through nested data structures, which are grouped in integer, char, floating data characterizing itself for its organization and the relationships among the data that constitute it. To develop a program applying constant values that refers to data already permanent and variables type integers as well as counters to obtain accumulated values and total sum of trips, such as drawers for n companies, which are distributed the products already specified in the program with This means obtaining the biggest sale and ordering, as its search through a self-generated code, we have taken a time of execution in the program for which the programming language C ++ and Code: Blocks will be used, since

Recibido 20 de febrero del 2018; revisión aceptada 20 de marzo 2019

¹ Universidad Técnica de Cotopaxi, Ecuador, patricio.sangopanta8434@utc.edu.ec

² Universidad Técnica de Cotopaxi, Ecuador, bryan.mereolo8546@utc.edu.ec

³ Universidad Técnica de Cotopaxi, Ecuador, edwin.quinatoa@utc.edu.ec

it is a flexible language that works with the structured programming with the possibility of defining source code files, and for the execution of the program was done in Dev c ++ and Code: Blocks. In addition to the use of structures, functions are also used with parameters and, in turn, by value, by reference, which return or return values within the code. Modular programming is composed of specific tasks that are declared independently, and these in turn seek to be resolved by parts, this is to divide the program into sub-programs that allow the reduction of the code for a better understanding and thus can be applied in any development IDE as is the Code: Blocks.

Key words: code, development, nested structures, functions, program.

INTRODUCCIÓN

El presente artículo tiene como finalidad dar a conocer acerca de las estructuras anidadas de datos y su aplicación en la programación; la estructura de dato se caracteriza por su organización y las operaciones que define, en el cual se basa en ciertos aspectos como los datos, proceso y resultados que ayudan a realizar un análisis acerca del problema que queremos dar una solución, de manera que la estructura pueda conservar su tamaño y mantener su tiempo de ejecución, y en muchos de los casos el tamaño no se puede modificar aunque exista desperdicios de memoria.

La estructura de datos puede contener elementos que estén compuestos por otras pequeñas estructuras, es decir, tienen tareas independientemente que agrupa los datos obtenidos con los que trabajara nuestro código, incluyendo punteros que pueden ser del mismo tipo. Para este programa se ha empleado ciertos aspectos de funciones con parámetros, con el único objetivo de realizar nuestro programa de manera mejora y más sencilla, como entendible para el usuario y así memorizar el código de nuestra parte de Dev.

En el desarrollar de un código se ha utilizado los IDE de desarrollo Dev C++ y Code::blocks que han permitido tener una comparación sobre el tamaño que genera de acuerdo a su tiempo de ejecución, y para ello también se ha agregado el uso del lenguaje de programación c++ que realiza la manipulación de objetos y se caracteriza por ser multiparadigma.

El IDE de desarrollo Code::blocks es un entorno integrado libre que de igual manera se utilizado para el desarrollo de programas en lenguaje C++. Se lo conoce también como un entorno multiplataforma porque puede ser de uso libre para distintos sistemas operativos aunque cuenta con una licencia de GNU.

En un algoritmo además de hacer uso del lenguaje C, de la misma manera se establece el empleo de funciones en un código que consiste en dividir en partes el programa en subprogramas, en los cuales tienen un programa principal conocido como main ya que mediante esta función los programas empiezan a ejecutarse, esta función debe encontrarse de forma obligatoria dentro del

código porque se encarga de enviar información a las otras funciones ya sean de valor o por referencia.

Finalmente se utiliza métodos de ordenamientos como, método de burbuja, al igual se utiliza método de búsqueda mediante un código para una visualización ya que es importante conocer cada uno de los ítems ingresados en esta parte del programa, estructura de datos también se la utiliza en diferentes puntos ya que cumple con ciertos parámetros que describen que los datos aparecerán cuando el programa termine y la cantidad de datos no debe ser de mayor tamaño ya que se denominaría como un arreglo de estructuras o llamado tipo arrays.

DESARROLLO

A. Estructura de datos

En [1] se menciona que las estructuras de datos en C++ se pueden entender como un tipo de dato compuesto (no complejo). Las estructuras de datos permiten almacenar de manera ordenada una serie de valores dados en una misma variable. Las estructuras de datos más comunes son los vectores o arreglos y las matrices, aunque hay otras un poco más diferentes como son el struct y las enumeraciones. Es una forma particular de organizar datos en una computadora para que puedan ser utilizados de manera eficiente.

Las estructuras de datos son un medio para manejar grandes cantidades de datos de manera eficiente para usos tales como grandes bases de datos y servicios de indización de Internet. Por lo general, las estructuras de datos eficientes son clave para diseñar algoritmos eficientes. Algunos métodos formales de diseño y lenguajes de programación destacan las estructuras de datos, en lugar de los algoritmos, como el factor clave de organización en el diseño de software.

B. Lenguaje de programación

En [2] menciona que un lenguaje de programación es una herramienta que nos permite comunicarnos e instruir a la computadora para que realice una tarea específica. Cada lenguaje de programación posee una sintaxis y un léxico particular, es decir, forma de escribirse que es diferente en cada uno por la forma que fue creado y por la forma que trabaja su compilador para revisar, acomodar y reservar el mismo programa en memoria.

C. Lenguaje de programación C++

En [2] también explica sobre el tipo de lenguaje a utilizar como es el C++ es un lenguaje de programación orientado a objetos que toma la base del lenguaje C y le agrega la capacidad de abstraer tipos como en Smalltalk.

La creación de C++ tuvo el objetivo de extender al exitoso lenguaje de programación C con mecanismos que permitieran la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje.

Posteriormente se obtuvo facilidades de programación genérica, que se sumó a los otros dos paradigmas que ya estaban admitidos tanto en programación estructurada y la programación

orientada a objetos. Por esto se suele decir que el C++ es un lenguaje de programación multiparadigma.

D. Funciones

En [4] explica que una función es una sección de un programa que calcula un valor de manera independiente al resto del programa. Es un conjunto de líneas de código que realizan una tarea específica y puede retornar un valor. Las funciones pueden tomar parámetros que modifiquen su funcionamiento. Son utilizadas para descomponer grandes problemas en tareas simples y para implementar operaciones que son comúnmente utilizadas durante un programa y de esta manera reducir la cantidad de código. Cuando una función es invocada se le pasa el control a la misma, una vez que esta finalizó con su tarea el control es devuelto al punto desde el cual la función fue llamada.

Variables locales: las variables locales solo pueden ser utilizadas únicamente en la función que hayan sido declaradas. Es una variable que está declarada dentro de un subprograma, y se dice que es local al subprograma, lo que la caracteriza es que su valor sólo está disponible mientras se ejecuta el subprograma.

Variables globales: Estas variables globales se crean durante la ejecución del programa, y son globales porque son llamadas, leídas, modificadas. Desde cualquier función y se definen antes del programa principal (main). Son variables declaradas en la sección correspondiente a esta labor en el programa principal y a diferencia de las locales, su valor está disponible tanto en el cuerpo del programa principal como en el de cualquiera de los subprogramas declarados.

Tipos de funciones

Funciones con Parámetros

Las funciones normalmente operan sobre ciertos valores pasados a las mismas ya sea como constantes literales o como variables, aunque se pueden definir funciones que reciban parámetros como explica según [4]. Existen dos formas en C++ de pasar parámetros a una función:

Parámetros por referencia.- si en una declaración de función se declaran parámetros por referencia, a los mismos no se les podrá pasar valores literales ya que las referencias apuntan a objetos (variables o funciones) residentes en la memoria, pueden ser alterados por la función que los reciba.

Parámetros por valor.- si un parámetro es declarado para ser pasado por valor, el mismo puede pasarse como una constante literal o como una variable, los parámetros pasados por valor o copia no pueden ser alterados por la función que los recibe, es decir, la función puede manipular a su antojo al parámetro, pero ningún cambio hecho sobre este se reflejará en el parámetro original.

Funciones void.- para escribir funciones que no regresen valor alguno, y para ello podemos declarar a la función como void. La palabra reservada void es utilizada para declarar funciones sin valor de retorno y también para indicar que una función específica no requiere de parámetros.

Funciones que no reciben ni retornan.- son las más sencillas para poder utilizarlas tenemos que saber cómo crearlas y como llamarlas. Una función se crea de esta forma general:

```
Tipo nombre () {  
}
```

El “tipo” se refiere al tipo de dato (int, float,void,char) y en las funciones que no retornan es la función void.

Funciones que reciben y no retornan.- esta función es capaz de recibir datos. Su sintaxis es:

```
Tipo nombre (tipo_var1 nombre_var1, tipo_var2 nombre_var2) {  
}
```

El “tipo” y “nombre” hacen referencia a la función void.

Funciones que no reciben y retornan.- recibir es enviar datos del main a la función. Retornar es enviar datos de la función al main. Para retornar datos hay que utilizar el “return” y no usar el “void” como tipo.

Funciones que reciben y retornan. - esta función se encarga de recibir las variables y de retornar el resultado.

Librerías .h

En [6] menciona que las librerías son un conjunto de funciones que usamos frecuentemente y que por lo general están agrupadas de acuerdo a su utilidad, por ejemplo se utiliza el iostream que tiene funciones para controlar la entrada y salida de datos.

I. IDE de desarrollo

En [7] hace referencia a IDE que es un entorno de programación que ha sido empaquetado como un programa de aplicación, o sea, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Sus siglas significan integrated development environment es un programa informático compuesto por un conjunto de herramientas de programación. Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. El lenguaje Visual Basic, por ejemplo, puede ser usado dentro de las aplicaciones de Microsoft Office, lo que hace posible escribir sentencias Visual Basic en forma de macros para Microsoft Word. Ofrecen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, funciona como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto.

A. Dev C++

En [8] señala que el Dev C++ es un IDE para crear aplicaciones utilizando el lenguaje de programación C++, que ocupa muy poco tamaño en el disco duro, ideal para crear programas pequeños en las que solo sea necesario demostrar el uso de estructuras de control y estructuras de datos, estas aplicaciones se pueden compilar rápidamente y ejecutar en forma de consola. Es

un compilador y entorno de desarrollo para C/C++ consiste en un editor de múltiples ventanas integrado con un compilador que nos permitirá una compilación, un enlace y una ejecución de aplicaciones rápida y de fácil uso.

B. Code::Blocks

En [9] indica que el code::blocks es un entorno de desarrollo integrado de código abierto, que soporta múltiples compiladores, que incluye GCC, Clang y Visual C++. Se desarrolló en C++ usando wxWidgets como el kit de herramientas GUI. Utilizando una arquitectura de complemento, sus capacidades y características están definidas por los complementos proporcionados.

Actualmente la mayor parte de la funcionalidad viene provista por los complementos incluidos predeterminadamente. No es un IDE autónomo que acepta complementos, sino que es un núcleo abstracto donde los complementos se convierten en una parte vital del sistema. Esto lo convierte en una plataforma muy dinámica y potente, no solo por la facilidad con que puede incluirse nueva funcionalidad, sino por la capacidad de poder usarla para construir otras herramientas de desarrollo tan solo añadiendo complementos.

II. Fragmento del código

A. Enunciando del problema

Realizar un programa que me permita ingresar n empresas, y n viajes en el cuál por cada viaje tome 3 tipos de productos los cuales son constantes, cada uno tendrá su peso en kilos debido a eso se genere una cantidad de viajes por cada empresa y totalidad de gavetas.

Condiciones:

- Si las cantidades de kilos llevados por una empresa es mayor a 10.000 aporta el 5%.
- Cantidad mediante un rango $15 < x < 22$.

Nota: En el programa se debe incluir un menú de opciones que permita ingresar, búsqueda, mayor venta y ordenamiento.

B. Análisis del Problema

Proveedor: 1 código: 01

Nombre	RUC	Dirección	Teléfono
La Favorita	564654645	Quito	0998454345

Cliente: 1 Código: 001

Nombre	Apellido	Dirección	Teléfono
Bryan	Merelo	Guamani	0996456546

Producto: 1 Código: 0001

Tipo de productos:

- 1.- Brocoli
- 2.- Frejol
- 3.- Papas

Tipo	Gavetas	Calculo
1	16	1600

C. Descripción de un fragmento de código con captura de pantalla

Función principal main

En el siguiente fragmento se puede observar las líneas de código que nos indican el ingreso del nombre, apellido y cedula. Esto nos sirve para poder identificar cada uno de los datos ingresados para al momento de la impresión concuerde con el mismo.

```
24 int main() {
25     cout<<"Ingrese los proveedores\n";
26     cin>>n0;
27     system("cls");
28     for(int i=0;i<n0;i++){
29         fflush(stdin);
30         cout<<"Proveedor NI"<<i+1<<endl;
31         cout<<"Codigo\n0"<<i+1<<endl;
32         cout<<"Nombre\n";
33         cin.getline(pp[i].nprov,20);
34         cout<<"RUC\n";
35         cin.getline(pp[i].rucprov,20);
36         cout<<"Direccion\n";
37         cin.getline(pp[i].dirprov,40);
38         cout<<"Telefono\n";
39         cin.getline(pp[i].teleprov,20);
40         cout<<"Ingrese numero de clientes\n";
41         cin>>pp[i].nncli;
42         cout<<"Ingrese costo de viaje\n";
43         cin>>pp[i].ncostovia;
44         pp[i].Ttol=pp[i].ncostovia*pp[i].nncli;
45         for(int j=0;j<pp[i].nncli;j++){
46             system("cls");
47             fflush(stdin);
48             cout<<"Cliente NI"<<j+1<<endl;
49             cout<<"Codigo\n00"<<j+1<<endl;
50             cout<<"Nombre\n";
51             cin.getline(pp[i].c[j].ncli,20);
52             cout<<"Apellido\n";
```

Figura 1. Fragmento de la función principal main.

```
C:\Users\PC\Documents\main.exe
Proveedor N 1
Codigo
01
Nombre
Favorita
RUC
67657567
Direccion
Quito
Telefono
09876576
Ingrese numero de clientes
2
```

Figura 2. Ejecución del código de la función principal main.

```
C:\Users\PC\Documents\main.exe
Cliente N 1
Codigo
001
Nombre
Juan
Apellido
Merelo
Direccion
Guamani
Telefono
09956765
Ingrese numero de productos
2
```

Figura 3. Ejecución del código del cliente.

Funciones

El siguiente fragmento nos muestra las líneas de código que se encuentran en la función void inicio lo cual nos permite ingresar los datos que son necesarios para el desarrollo, pero cabe recalcar que esta función no imprime los datos, para esto se utiliza otra función void imprimir

```
25 void ingreso();
26 void imprimir();

void ingreso(){
    cout<<"Ingrese los proveedores\n";
    cin>>n0;
    system("cls");
    for(int i=0;i<n0;i++){
        fflush(stdin);
        cout<<"Proveedor NI"<<i+1<<endl;
        cout<<"Codigo\n0"<<i+1<<endl;
        cout<<"Nombre\n";
        ...
    }
}

void imprimir(){
    system("cls");
    //ordenar inicio :)
    for(int i=0;i<=n0;i++){
        for(int j=0;j<n0-1;j++){
            if(pp[j].Ttol>pp[j+1].Ttol){
                aux=pp[j].Ttol;
                pp[j].Ttol=pp[j+1].Ttol;
                pp[j+1].Ttol=aux;
            }
        }
    }
}
```

Figura 4. Fragmento del código en funciones.

```
Ordenamiento de precios de proveedor de mayor a menor
-----
Proveedor 1:
1200
```

Figure 5: Ordenamiento de Proveedores.

```
Proveedor N°1
Codigo: 01
Nombre: Favorita
RUC: 9696976
Direccion: Quito
Telefono: 09858656
---> numero de clientes: 1
Costo de viaje: 100
---> Cliente N°1
Codigo: 001
Nombre: Bryan
Apellido: Merelo
Direccion: Guamani
Telefono: 09875676
---> Numero de productos: 1
---> Producto N°1
Codigo: 0001
Producto: Frejol
Gavetas: 16
```

Figura 6. Datos impresos en la consola.

```
C:\Users\PC\Documents\programas\merelo2.exe
Usted tiene 1 Proveedores :
Desea buscar algun proveedor?
  Ingrese codigo
01

Nombre :Favorita
Direccion :Quito
Telefono :09858656
Total a pagar : 1200
Costo de viaje :100
Clientes -> 1

Desea buscar algun cliente del proveedor seleccionado?
  Ingrese codigo
001

Nombre :Bryan
Apellido :Merelo
Direccion :Guamani
Telefono :09875676
Productos :1

Desea buscar algun producto del cliente seleccionado?
  Ingrese codigo
0001

Numero Gavetas :16
Tipo de producto :2
```

Figura 7. Búsqueda por código.

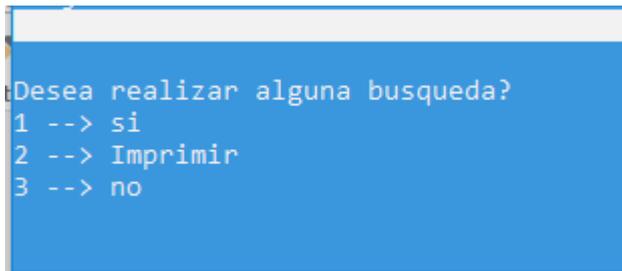


Figura 8. Menú básico.

Code::blocks

En esta parte del código muestra la función en una librería que tiene la finalidad de ingresar e imprimir los datos, por lo que en la línea 1 y 2 se puede observar el uso de las librerías.h, las mismas que son muy importantes ya que nos ayudan a realizar la correcta llamada de la función .h dentro del main para el proceso adecuado de los cálculos a realizar y una respectiva ordenación de código entre el main, funciones.h y estructuras.h.

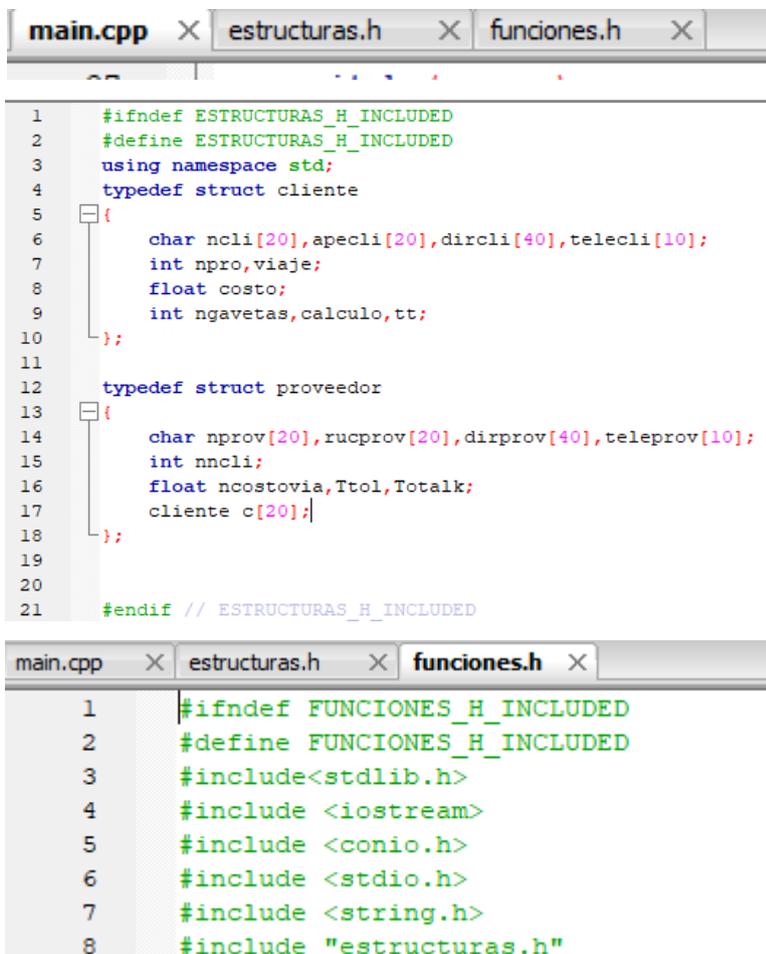


Figura 9. Fragmento del código en Code::Blocks.



Figura 10. Ejecución del código (Menu de opciones) en Code::blocks.

V. Aspectos de análisis

Para realizar el análisis del código se tenía en cuenta tres aspectos importantes: las líneas de código, tiempo de ejecución y el tamaño que genera el archivo tanto en el main y funciones dentro del Dev C++ y en las librerías .h conjuntamente con el Code::Blocks.

En las siguientes tablas se muestra los valores obtenidos en las tres formas de programación:

Tabla 1. Líneas de código.

Programación	Líneas de código
Función principal main()	63
Funciones	254
Librerías .h	26
Sumatoria Total	\sum 343

Tabla 2. Tiempo de ejecución.

Programación	Tiempos de compilación	Tiempo de ejecución
Funcion principal main()	0,84s	54,53s
Funciones	0,78s	51,89s
Librerías .h	0,3s	43,01s
Sumatoria Total	\sum 1,92	\sum 149,43

```
-----
Process exited after 54.53 seconds with return value 0
Presione una tecla para continuar . . . █
```

Figura 11. Tiempo de ejecución del main.

```
-----
Process exited after 51.89 seconds with return value 0
Presione una tecla para continuar . . . █
```

Figura 12. Tiempo de ejecución del código en funciones.

Code::blocks

Tamaño del archivo que genera

Tabla 3. Tamaño de archivo generado.

Programación	Tamaño del archivo
Funcion principal main()	1,34 KB
Funciones	10,5 KB
Librerías .h	1,11 KB
Sumatoria Total	\sum 12,95 KB

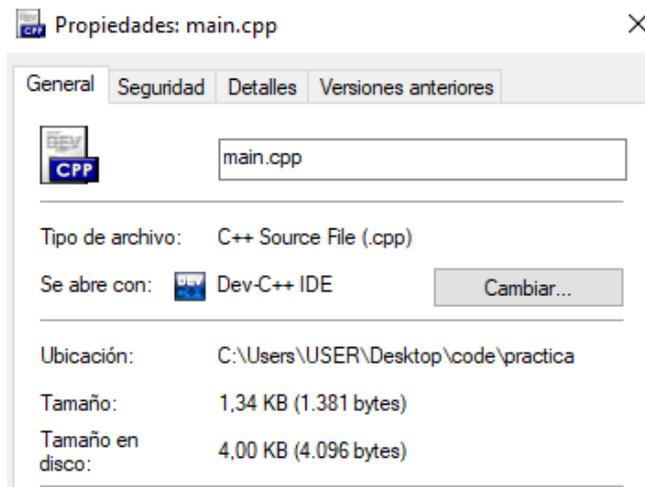


Figura 3. Tamaño del archivo de la función principal.

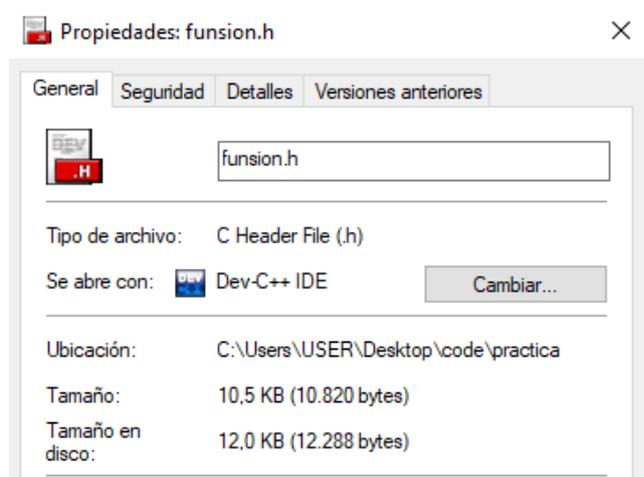


Figura 4. Tamaño del archivo del código en funciones.

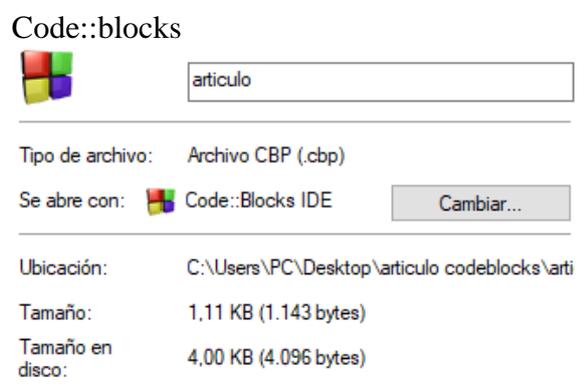


Figura 5. Tamaño del archivo del código en Code::blocks.

Resultados del análisis:

En esta sección se interpretara los resultados obtenidos mediante porcentajes de acuerdo al análisis mencionado anteriormente tomando en cuenta los aspectos importantes relacionados con las tres formas de programación del lenguaje C++ y Code::blocks.

Líneas de código

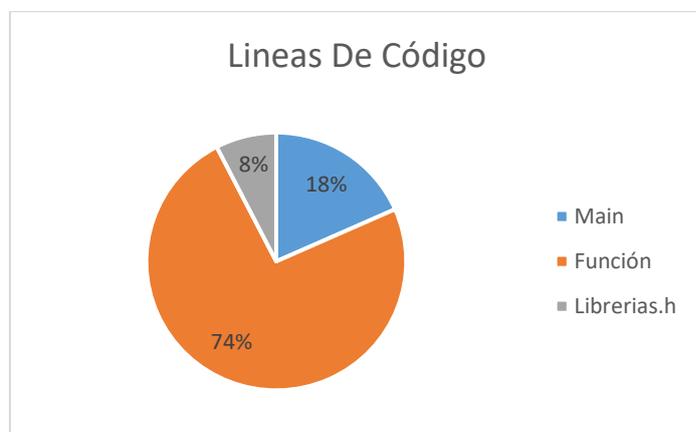


Figura 6. Porcentaje equivalente a líneas de código.

Interpretación: En el gráfico se puede observar que el mayor número de líneas de código se encuentra en las funciones ya que en esta parte se realiza la declaración de nuevas funciones vacías o con parámetros a ocupar, para el proceso debido a realizar, mientras que las del main y las librerías consta con menos líneas de código, por motivo que solo son declaraciones de estructuras y llamamientos dentro de ellas, la mayoría de desarrollo se genera dentro de las funciones.

Tiempo de ejecución

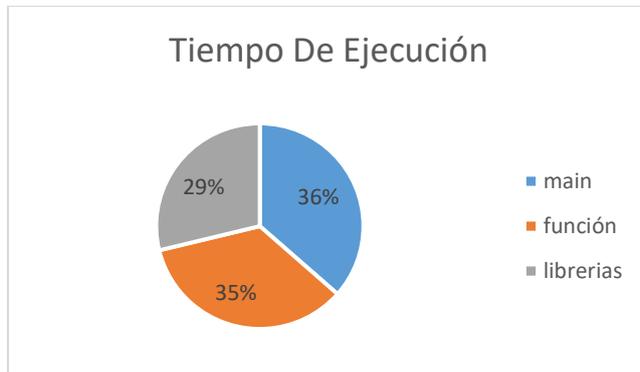


Figura 7. Porcentaje equivalente al tiempo de ejecución.

Interpretación: En el gráfico se puede observar que el mayor número de porcentaje del tiempo de ejecución llega a ser el main.

Tamaño del archivo

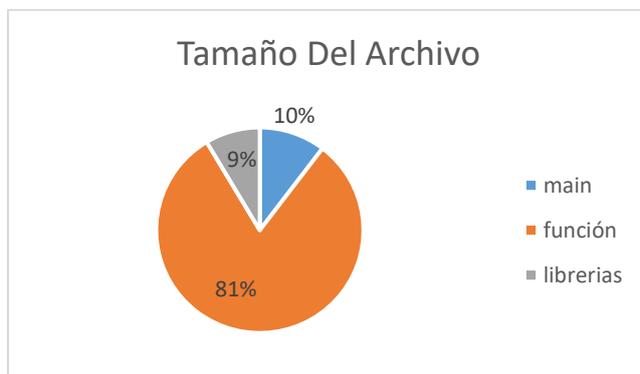


Figura 8. Porcentaje equivalente al tamaño del archivo.

Interpretación: En el gráfico se puede observar que el mayor número de porcentaje del tamaño del archivo es el código en funciones mientras que la función principal main tiene un rango menor y las librerías.h es el mínimo de tamaño que genera el archivo de Code::Blocks.

CONCLUSIONES

Se concluye que el programa que se utiliza con mayor frecuencia es el Dev C++ ya que ocupa poco espacio en el disco con la debida aplicación del lenguaje de programación C, de manera que se ha relacionado con plataformas de código abierto que permite que la programación sea más factible de aprender e incorporar.

Se concluye que la utilización de un IDE de desarrollo en cierto porcentaje nos ha facilitado la programación ya que son programas de gran importancia y sobre todo cambiantes ya que se puede aplicar en cualquiera de ellos porque son gratuitos y se los implementa en las líneas de

código para buscar una solución en el momento de compilar el programa y así poder corregir errores.

Se concluye que es fundamental conocer ciertos aspectos básicos acerca, de nuestro proyecto y tener un amplio conocimiento ya sea en realizar un artículo como las diferentes formas de la programación.

REFERENCIAS

- [1] Pascal, «Estructura de Datos,» 2011, [En línea]. Available: <https://www.programarya.com/Cursos/C++/Estructuras-de-Datos>.
- [2] Oracle, «Lenguaje de Programacion,» 16 03 2015. [En línea]. Available: <https://aprendiendoarduino.wordpress.com/2015/03/26/lenguaje-de-programacion-c/>.
- [3] E. Pascal, «Documento Programacion Modular,» 2008. [En línea]. Available: <http://teleformacion.edu.aytolacoruna.es/PASCAL/document/modular.htm> .
- [4] Anonimo, «Funciones con Parametros,» 13 08 2013. [En línea]. Available: <https://www.tutorialesprogramacionya.com/cya/detalleconcepto.php?punto=17&codigo=17&inicio=15>.
- [5] F. B. y. R. Ferris, «Introduccion a Ficheros,» 2009. [En línea]. Available: <file:///C:/Users/Ficheros/Documents/AED.Tema.07.pdf>.
- [6] Anonimo, «Librerias estaticas,» 2011. [En línea]. Available: http://www.programacionenc.net/index.php?option=com_content&view=article&id=63:creacion-y-uso-de-una-libreria-estatica-h&catid=37:programacion-cc&Itemid=55.
- [7] D. Maldonado, «¿Que son los IDE de desarrollo?,» 2012, 24 10. [En línea]. Available: https://www.ecured.cu/IDE_de_Programaci%C3%B3n.
- [8] E. Perdomo, «IDEde desarrollo: Dev C++,» 27 08 2009. [En línea]. Available: <https://eperdomo89.wordpress.com/2009/08/27/dev-c-%C2%BF-que-es/>.
- [9] Anonimo, «Tipos Simples y estructuras de Control,» [En línea]. Available: file:///C:/Users/TECNOMANIA/Documents/01_Tipos-Simples-y-Estructuras-de-Control.pdf. [Último acceso: 13 07 2018].