

## Programación de Arrays En C++

### *Programming of arrays in C++*

Ronny Richard Arauz León<sup>1</sup>, Leonardo Steve Pilapaña Tipan<sup>2</sup>, Edwin Edison Quinatoa Arequipa<sup>3</sup>

#### RESUMEN

En el presente artículo se presenta una breve base teórica de la programación empleando el lenguaje C++, enfocándose en el desarrollo del tema de arrays y sus métodos de ordenación. Añadiendo temas necesarios referentes a la programación con el lenguaje C++, ya que estos temas van a ser necesarios para poder dar solución al problema planteado, de dicho problema se explicará el análisis, fragmentos del código empleado para brindar una solución al ejercicio. Una vez terminado el ejercicio se ha de comenzar con el análisis de los IDES'S que se emplearan para el desarrollo del programa, añadiendo un análisis sobre los métodos más utilizados en el ordenamiento de datos de un arrays, tomando en cuenta investigaciones posteriores se le determinara una ponderación a cada método para lograr validar cuál de ellos es el más empleado, todo esto se realiza a través de unas tablas porcentuales de la cual se tomaran los datos obtenidos y se determinara las conclusiones sobre estos parámetros a analizar.

**Palabras clave:** Programación, paradigma, flujograma, programación modular.

#### ABSTRACT

In the present article a brief theoretical basis of the programming is presented using the C ++ language, focusing on the development of the theme of arrays and their methods of arrangement. Adding necessary topics regarding the programming with the C ++ language, since these topics will be necessary to be able to solve the problem, from this problem the analysis will be explained, fragments of the code used to provide a solution to the exercise. Once the exercise is finished, it is necessary to begin with the analysis of the IDES'S that will be used for the development of the program, adding an analysis on the most used methods in the ordering of data of an arrays, taking into account later investigations. determine a weighting to each method to achieve validate which of them is the most used, all this is done through a percentage tables from which the data obtained and determine the conclusions on these parameters to be analyzed.

**Keywords:** Programming, paradigm, flow diagram, arrays, modular programming.

Recibido 18 de febrero de 2019, aceptado 25 de marzo de 2019.

<sup>1</sup> Universidad Técnica de Cotopaxi, Latacunga, Cotopaxi, Ecuador, [ronny.arauz3082@utc.edu.ec](mailto:ronny.arauz3082@utc.edu.ec)

<sup>2</sup> Universidad Técnica de Cotopaxi, Latacunga, Cotopaxi, Ecuador, [leonardo.pilapana8496@utc.edu.ec](mailto:leonardo.pilapana8496@utc.edu.ec)

<sup>3</sup> Universidad Técnica de Cotopaxi, Latacunga, Cotopaxi, Ecuador, [edwin.quinatoa@utc.edu.ec](mailto:edwin.quinatoa@utc.edu.ec)

## INTRODUCCIÓN

Para poder estudiar y comprender cualquier lenguaje de programación primero se debe conocer cuáles son los conceptos y terminologías que soporta esta, es decir que dependiendo de los conceptos y terminologías que se aprenda, se sabrá que método de programación podemos realizar con lo que se ha aprendido, pero para el desarrollo de este trabajo se empleara el método de programación main, puesto que se ha de realizar una comparación del mismo código en diferentes IDE'S de desarrollo.

Como el artículo se basa en la programación de arrays en el lenguaje c++, se ha empleado tres de los ide's más usados para el estudio de este lenguaje, donde uno de ellos ha sido desarrollo en el propio lenguaje c++ y por este motivo lo lleva en su nombre dando a indicar que el ide Dev c++ fue desarrollado sobre este lenguaje.

Pasando de la conceptualización y de la herramienta utilizada se ha de presentar un problema planteado para desarrollar empleando vectores, este problema planteado permitirá mostrar la declaración de los vectores, su manipulación además de realizar la aplicación de un método de ordenación de datos que puede contener un array.

Dando por último se ha de analizar el código en los diferentes ide's según los parámetros establecidos donde uno de ellos ha de demostrar que ide es el más rápido en compilar el código generado por el ejercicio planteado, terminado el análisis de todos los parámetros se ha de presentar a través de graficas porcentuales los resultados obtenidos en el análisis.

Terminado todo lo anterior se debe tener presente que este artículo se ha realizara para dar a conocer a los lectores más métodos y formas para obtener un buen resultado al realizar la programación de vectores en el lenguaje c++, este lenguaje viene siendo uno de los más importantes de aprender para aquel que se inicia en el área de la programación.

### A. C++:

En primer lugar, C++ es un lenguaje de programación multiparadigma la cual admite emplear la programación orientada a objetos y la programación estructurada, pero este lenguaje no se desarrolló de esta manera en un principio, ya que este lenguaje es una mejora de lo que fue el lenguaje c el cual se quedaba corto para la resolución de algunos problemas es en este punto donde el amor de los programadores por este lenguaje hace que surjan nuevos lenguajes basados en el cómo el objective C, pero de todas estas versiones destaco el C++ debido a que se lo considero un lenguaje hibrido destacándolo sobre los demás lenguajes orientados a objetos, tal como se hace referencia.

“En palabras de Bjarne Stroustrup, el hecho de que C++ no sea un lenguaje orientado a objetos puro es más una ventaja que un inconveniente, ya que lo hace más versátil y adecuado para un mayor número de aplicaciones”[1, p. 30].

Este lenguaje de programación también se considera un lenguaje de alto nivel ya que permite el uso de números, letras y otros símbolos que están en el teclado, para luego transformarlo a un lenguaje de bajo nivel o lenguaje de maquina el cual se maneja en código binario que va a ser el que entiende la computadora para que a través de varias transformaciones el código escrito en c++ termina en un programa.

Pero para que podamos generar un programada se tiene que escribir el código y para eso necesitaremos varias herramientas las cuales son:

- Un compilador que pueda correr C++
- Un editor de texto o mejor un IDE<sup>4</sup>
- Nociones básicas de programación
- Y una gran cantidad de paciencia

Ya con todas las herramientas listas e instaladas en el ordenador se puede comenzar el aprendizaje de este lenguaje.

### **B. IDE:**

Los IDE'S son entornos de desarrollo de software estos son un conjunto de herramientas como lo son un compilador, un editor de texto, un depurador y un constructor de interfaz gráfica, estos IDE'S se pueden presentar como una sola aplicación o puede ser un complemento el cual puede interactuar con otros programas como en el caso de Microsoft Excel el cual incluye un complemento de Visual Basic, con el cual se puede escribir código como una especie de macro para poder ejecutar tareas, estos IDE'S además ofrecen una interfaz amigable para el programador y no solo trabajan en un lenguaje sino hay varios que pueden manejar varios lenguajes tal como el caso de eclipse[2].

Para considerar usar cierto IDE este necesita tener ciertas características como ser multiplataforma, integración con framework populares, soportar varios lenguajes sino se quiere ser específico en uno solo por ejemplo en el caso que se quiera crear un programa en java con conexión a una base de datos se debe usar NetBeans ya que este soporta Java y a la vez php o MySQL.

---

<sup>4</sup> *Integrated Drive Electronics* o en español entorno de desarrollo integrado.

En la actualidad existen una gran cantidad de IDE'S en el mercado tanto por licencia como gratuitos, pero en este trabajo está enfocado en el lenguaje C++, para este lenguaje existen varios IDE'S, pero se verán los más conocidos para manejar este lenguaje.

**a. Microsoft Visual Studio:**

Es un IDE específicamente para sistemas operativos Windows este IDE también es multiplataforma puesto que fue desarrollado para manejar C, pero con los avances que ha tenido actualmente puede soportar C++, Visual Basic, Java, Python entre otros más.

Durante las actualizaciones que este IDE ha tenido cada vez ha ido mejorando a niveles de interfaz, facilidad de lectura y escritura además de ayudas para los desarrolladores con plantillas, este IDE se encuentra en su versión 2017 pero la empresa Microsoft se encuentra por lanzar al mercado la nueva versión del 2019[3].

**b. Dev C++:**

Bloodshed Dev-C ++ es un IDE que está diseñado para el lenguaje de programación C y C ++ y mantiene una licencia de código abierto lo que quiere decir que es un software libre, pero se encuentra disponible solo para el OS de Windows [4].

**c. Code::Blocks:**

Code: blocks es un IDE de código abierto, que soporta múltiples compiladores, que integra GCC y Visual C++. Se desarrolló en C++ usando wxWidgets como el kit de herramientas GUI. Utilizando una arquitectura de complemento, Code: blocks está orientado hacia C, C++ y Fortran. Debido a que Dev-C++ y otros IDE'S están diseñados para los lenguajes C y C++, pero está creado en Delphi, en ese momento surgió la motivación de crear un IDE hecho en los lenguajes propios como son C y C++. Con este propósito se desarrolló Code::Blocks.

**C. Métodos de programación:**

Para la programación en el lenguaje C++ se la puede realizar mediante varias formas o métodos como el método de funciones en el cual el código se divide en partes dejando una función principal el cual va a ser el primero en ejecutar y el que se encargara de buscar cual es la estructura que se codifico este método se lo aplica junto con el método de librerías.h en la programación estructurada, donde el problema se lo divide en subproblemas o más conocido como divide y vencerás.

Pero nos centraremos en el método el cual se va a evaluar a continuación que es la programación solamente en el main o en la función principal.

#### **a. Programación en el main:**

Es la función principal de la estructura de todo programa ya que este va a ser el punto de arranque por donde va a comenzar a buscar funciones o directamente ejecutar el programa codificado.

Además, se encarga de mantener el orden en el que van a ser ejecutadas las funciones que se hayan declarado, para su correcta declaración en el código esta debe seguir ciertos lineamientos como, se declara después de las librerías y de la inicialización de los objetos no globales, no se puede utilizar en ningún otro lugar del código ni realizar recursividad del mismo peormente realizar la llamada de esta en otra función declarada [5].

Cuando se realiza la codificación de un programa solo en la función main se presentan varios problemas si se trata de un problema complejo, por lo general cuando se codifica en el main se ve que a medida que avanza el código se vuelve más tediosa y difícil de comprender, la poca reutilización de código y se presenta una dificultad para poder programar, pero sobre todo siempre se verá código repetido.

#### **D. Librerías:**

Las librerías son un conjunto de funciones que usamos frecuentemente y que por lo general están agrupadas de acuerdo a su utilidad, por ejemplo, la iostream tiene funciones para controlar la entrada y salida de datos.

“Las librerías de E/S de C++, llamada IOSTREAMS, proporciona una forma segura y fiable de utilizar E/S mediante los flujos. Es preferible a la de C llamada Stdio, por la mejor comprobación de tipos para la lectura/escritura”[1, p. 374].

#### **E. Variables o los tipos de datos:**

En C++ un variable es un espacio de memoria reservado donde se almacena el valor de un cierto tipo de dato y puede ser modificado. Las variables pueden almacenar todo tipo de dato tanto cadenas, valores enteros, decimales, por otra parte, una constante es un espacio de memoria designado solo para ese valor y este no puede ser modificado [6].

#### **F. Sentencias:**

Las sentencias o las estructuras de control son los elementos básicos que van a controlar el flujo en el que el programa se da además de dividir el código dependiendo que se necesite realizar, estas sentencias también tienen sus palabras reservadas las cuales no pueden ser utilizadas como variables, las sentencias de control pueden ser cíclicas o de opción según sea el caso que se necesite [6]. Dentro de estas sentencias se encuentra el if, if-else, while, do while, for, switch.

### G. Arrays:

Los arrays o arreglos unidimensionales y llamados popularmente vectores no son nada más que una estructura de datos de un mismo tipo que permite almacenar valores en un solo bloque de memoria juntos, para esto en su declaración se le debe proporcionar una dimensión en su declaración.

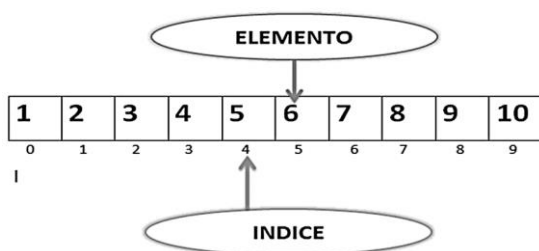
40	55	63	17	22	68	89	97	89
0	1	2	3	4	5	6	7	8

**Figura 1.** Array de nueve elementos

En la figura 1 se observa cómo se representa un vector de nueve elementos de tipo numérico con sus partes respectivas.

Existen varias formas para poder declarar un vector esto ya depende del programador que forma le parece mejor emplear en su código, pero por lo general al vector se le asigna primero el tipo de dato seguido del nombre que se le vaya a dar y por último se le da una dimensión, ejemplo `int a[9];`, esto nos quiere decir que es un vector de tipo entero llamado “a” el cual tiene una dimensión de 9 espacios.

Para poder ingresar valores dentro de un vector o modificarlos se hace con la estructura de control cíclica `for`, esta emplea el índice del vector para poder ir de posición en posición del vector y saber en qué espacio de ese vector se va a ingresar, eliminar o modificar el dato ahí alojado, el índice del vector siempre inicia en cero por regla general, pero no hay problema si se comienza en uno, el problema es que se estaría desperdiciando un espacio de memoria.



**Figura 2.** Partes de un arrays.

En la figura 2 se observa las partes que conforman un vector desde elemento que lo compone como el índice que indica la posición de espacio de memoria asignado.

Por ejemplo, si queremos ver qué valor se encuentra en la posición dos del vector se tendrá que manejar con el valor del índice donde veremos que según la figura 2 el valor que se encuentra en la posición 2 se encuentra el valor número 3 esto se debe a que se inicializó el vector en cero.

## METODOLOGÍA

### H. Desarrollo de un ejercicio como ejemplo para la aplicación de los temas desarrollados.

Se desarrolló un ejercicio para poder realizar un análisis comparativo entre los 3 métodos de programación, los cuáles van a ser por main donde se realizará el ejercicio solamente en el programa principal. El segundo método va a ser mediante funciones, lo cual se explicó en líneas anteriores, por lo que se va a dividir en programas más pequeños para llegar al resultado, y por último se empleara el método de librerías.h en el cual se van a crear librerías para poder almacenar ciertas funciones y llamarlas desde el programa principal para obtener un código más limpio.

#### a. Enunciado:

Realizar un programa en el cual se trabajará con arrays, dicho programa debe contar con un menú de opciones que permita al usuario ingresar valores a 2 arrays y realizar su multiplicación, insertar nuevos elementos a los vectores, modificar los elementos existentes, eliminar o encerrar los elementos que el usuario ingrese y ordenar los elementos de los vectores. Este programa deberá contar con las respectivas validaciones.

#### b. Análisis del ejercicio:

Para lograr resolver el problema se empleará variables globales, estas variables declaradas se van a utilizar en el menú, en los vectores añadiéndole un flag o bandera para limitar el ingreso en ciertas opciones ya que en la primera ejecución del programa no van a existir datos por lo que sería contraproducente tener habilitado la opción eliminar, imprimir y ordenar los datos.

Variables empleadas: Int op, op1, vec1[ ], vec2[ ], vec3[ ]; flag=0, vsize;

Luego para el resto del programa se utilizarán variables locales si son necesarias.

Las variables Int aumento, vsize1 se van a usar para insertar nuevos elementos, int posi, elemento para modificar los valores dentro de los vectores e int reduc, posición para eliminar los datos de los vectores o llevarlos a cero. Estas variables se las tratara como locales dentro del main y están declaradas en cada opción que se vaya a realizar.

Comenzando con el proceso del desarrollo de los casos vamos a separar su desarrollo en la cantidad de opciones a usar:

- Primero se va a ingresar a pedir al usuario que ingrese datos en la opción uno op1 = 1;
- Entramos en el caso 1 u opción 1 ahí realizaremos el ingreso de datos a los vectores, pidiendo su tamaño, primeramente -> vsize = 3.
- Se ingresará los datos en los vectores Vec1[3] y vec2[3] con un tamaño de 3 tomando el 3 del vsize.



**Tabla 1:** valores ingresados en los vectores

Int i < vsize	Vec1	Vec2
1	1	4
2	2	5
3	3	6

En la tabla 1 se observa la tabla con los valores ingresados en los dos vectores. Una vez ingresado los datos a los valores retornaremos al menú de opciones para esta vez imprimir los resultados de la multiplicación de los valores de los vectores pasando primeramente por el desarrollo de la multiplicación que se realiza a través de un for  $i=0; i<vsize; i++$ .

Continuando se ingresa en la opción 2 => op1=2.

Se imprimirá los vectores vec1, vec2 y vec3.ya cumplido el proceso de multiplicación que se detalla más abajo:

Donde vsize=3.

**Tabla 2.** Resultado de la multiplicación entre vectores.

Int i < vsize	Vec1	Vec2	Vecr
1	1	4	4
2	2	5	10
3	3	6	18

En la tabla 2 se observa el resultado obtenido entre la multiplicación de los vectores uno y dos. Una vez realizado esto ya podremos navegar por las opciones, así como ingresar nuevos elementos al nuevo vector o eliminar dependiendo que queramos hacer en el programa.

Por ejemplo, si vamos a insertar nuevos elementos se utilizarán las variables locales declaradas respectivamente en cada caso que queramos entrar en el menú.

Caso 3: -> op1 = 3 // insertar elementos

Variables locales a usar -> int aumento, vsize1;

Pedimos cuantos elementos desea ingresar -> aumento = 1

Entonces se sumará vsize + aumento dando a vsize1 el nuevo valor  $3 + 1 = 4$

Este valor será nuestro nuevo tamaño en los vectores donde a través de un for realizaremos el ingreso de datos igualando el valor de i al tamaño anterior para solamente ingresar en la posición disponible del vector:

```
for (int i=vsize1; i<vsize; i++) -> for(int i=3;i<4; i++)
```

Ahora se ejecuta el ingreso de los datos de la siguiente manera:

- se ingresará en vec1[ i ] -> vec1[ 3 ].
- se ingresará en la posición 3 del vector 1 el valor de 9 y se ingresará en vec2[ i ] -> vec2[3].
- se ingresará en la posición 3 del vector 2 valor de 9.



Ojo no confundir el tamaño del vector con las posiciones que este tiene en este caso se dice que los vectores tienen un tamaño de 4 pero se ingresa en la posición 3 ya que los vectores se comienzan a contar desde cero por ese motivo la condición del for es que se repite el bucle cuando el tamaño es menor al iterador con incremento de 1, siguiendo tendríamos los Vectores actualizados -> opción imprimir.

**Tabla 3** Resultado de los vectores actualizados.

Int i < vsize	Vec1	Vec2	Vecr
0	1	4	4
1	2	5	10
2	3	6	18
3	9	9	81

En la tabla 3 se observa el nuevo tamaño del vector y del resultado obtenido. Ahora entraremos en el caso 4 que es el de modificar los elementos del vector.

- Caso 4: -> op1 = 4 //modificar elementos

Retornando de nuevo al menú de opciones vamos a ingresar a la opción 4 la cual va a modificar los valores del vector pidiendo al usuario la posición del vector y en que vector desea modificar.

- int posi, elemen, n; -> variables locales a usar.

Se pedirá al usuario cuantos valores se editarán claro sumando el tamaño de ambos vectores, ósea  $4*2 = 8$  valores capaces que se pueden editar.

En este caso ingresaremos -> elemen = 1 editar un solo elemento.

Ingresaremos la posición del vector que desea modificar:

- posi = 3 -> entonces ingresaremos a la posición tres y pedimos a cuál vector modificar
- n = 1 -> ingresaremos en el vector 1 en la posición tres el valor de 8
- $vec1[ posi ] = 8$  ->  $vec1[ 3 ] = 8$ .

**Tabla 4.** resultado de la actualización de vectores

z	Vec1	Vec2	Vecr
0	1	4	4
1	2	5	10
2	3	6	18
3	8	9	72

En la tabla 4 se observa los nuevos datos que se ingresaron en los vectores a través del caso 4. Ahora ingresaremos al caso 5 para resolver en el cual vamos a eliminar los datos o llevarlos a cero sin modificar el tamaño de los vectores.

- Caso 5: -> op1 = 5 // eliminar elementos o llevar a cero.
- int reduc,posicion; -> variables a usar de manera local.
- se pedirá cuantos valores se desea eliminar en este caso vamos a tomar este valor para ambos vectores.

- `reduc = 1` -> eliminaremos un elemento.
- ingrese la posición que se desea eliminar el elemento.
- `posicion = 3` -> se eliminarán los valores que se encuentran en la posición 3 de ambos vectores.
- `vec1[posicion]=0;` -> `vec1[ 3 ]=0;`
- `vec2[posicion]=0;` -> `vec2[ 3 ]=0;`
- y se retornara a imprimir con los valores actualizados.

**Tabla 5.** Resultado con la actualización de los vectores.

Int i < vsize	Vec1	Vec2	Vecr
0	1	4	4
1	2	5	10
2	3	6	18
3	0	0	0

En la tabla 5 se observa el resultado del ingreso al caso 5 donde se eliminan o se enceran los valores de los vectores según su posición indicada por el usuario.

Continuando con el caso 6 y el ultimo por resolver donde se debe de ordenar los elementos del vector de forma ascendente, para lograr esto se emplean los algoritmos que se han desarrollado por terceros en este caso se emplea el algoritmo o método de selección.

El método de ordenamiento por selección es uno de los métodos que necesita un orden de  $n^2$  de operaciones para ordenar un vector de  $n$  datos.

“Este método consiste en encontrar el menor de todos los elementos del vector e intercambiarlo con el que está en la primera posición luego el segundo más pequeño y así sucesivamente hasta ordenar todos los elementos del vector” [7].

Pasos de su funcionamiento:

- Coger la primera posición y del resto del vector buscar el valor más pequeño
- Comparar el menor con el de la primera posición
- Si es verdadero se intercambian los valores
- Para falso no se intercambian se pasa al siguiente
- Cogemos la segunda posición y buscamos del resto del vector el más pequeño
- Volvemos a comparar el menor con el valor de la posición en la que nos encontremos
- Intercambiamos si el  $v$  sino seguimos
- Y así sucesivamente hasta ordenar el vector.

Condición a cumplir para generar el intercambio de valores:

- El número menor de las posiciones sobrantes del vector debe ser menor al valor de la posición usada o  $\text{Menor} < V[\text{POS}]$ .

**Tabla 3.** resultado de los vectores ordenados

Int $i < \text{vsize}$	Vec1	Vec2	Vecr
0	0	0	0
1	1	4	4
2	2	5	10
3	3	6	18

En la tabla 6 se observa el resultado del método de ordenamiento por selección empleado en el caso 6. Para cumplir con la solución de este caso no es necesario emplear el método de selección en este caso depende del programador que método es más conveniente de usar en su código, de esta manera se da por terminado el programa en el cual se añadirán validaciones dependiendo los datos que necesiten ser validados.

### c. Fragmentos de código

Este fragmento de código va a realizar lo que es la eliminación o puesta a cero los vectores en la posición ingresada por el usuario.

```

for(int i=0;i<reduc;i++)
{
do{
system("cls");
cout<<"\n\t|-----|\n";
cout<<"\t| POS \t VECTOR 1 \t VECTOR 2 |\n";
cout<<"\t|-----|\n";
for(int i=0;i<vsize;i++)
{
cout<<"\t\t"<<i<<"\t| "<<vec1[i]<<" ]\t [ "<<vec2[i]<<" ]"<<endl;
}
cout<<"\t|-----|\n";
cout<<"\t| INGRESE LA POSICION DEL ELEMENTO |\n";
cout<<"\t| QUE DESEA ELIMINAR |\n";
cout<<"\t| SE TOMARA EL VALOR PARA AMBOS VECTORES |\n";
cout<<"\t|-----|\n";
cout<<"\tingrese el numero de la posicion : ";
cin>>posicion;
if(posicion < 0 || posicion>vsize-1)
{
cout<<"\t|-----|\n";
cout<<"\t| LA POSICION NO EXISTE INGRESE NUEVO VALOR |\n";
cout<<"\t|-----|\n";
system("pause");
}
}while(posicion < 0 || posicion>vsize-1);
vec1[posicion]=0;
vec2[posicion]=0;
}
if(reduc==vsize)
{
flag=0;
}
system("cls");
cout<<"\n\t|-----|\n";
cout<<"\t| SE HAN ELIMINADO EXITOSAMENTE LOS ELEMENTOS |\n";
cout<<"\t|-----|\n";
}

```

**Figura 3.** Fragmento caso 5.

En la figura 3 se observa un fragmento de la codificación del caso 5 o el caso para eliminar o encerrar los datos del vector según la posición indicada por el usuario, esta opción se encuentra bloqueada al inicio de la ejecución ya que en un principio no existen datos en lo vectores ni tampoco tienen tamaño.

En el siguiente fragmento de código se mostrará el método de selección empleado para el ordenamiento de los datos de los vectores de forma ascendente.

```

system("cls");
cout<<"\t|-----|\n";
cout<<"\t|          BIENVENIDO          |\n";
cout<<"\t|-----|\n";
cout<<"\t|  ESTA OPCION ORDENARA LOS VECTORE |\n";
cout<<"\t|          EXISTENTES          |\n";
cout<<"\t|-----|\n\n";
system("pause");
for(int i=0;i<vsize;i++)
{
  vec3[i]=vecl[i]*vec2[i];
}
cout<<"\t|-----|\n";
cout<<"\t|          VISULIZACION DE VECTORES |\n";
cout<<"\t|-----|\n";
cout<<"\t| VECTOR 1 | \t| VECTOR 2 | \t| VECTOR R |<<endl;
cout<<"\t|-----|\n";
for(int i=0;i<vsize;i++)
{
  cout<<"\t| "<<vec1[i]<<" | "<<"\t\t| "<<vec2[i]<<" | "<<"\t\t| "<<vec3[i]<<" |<<endl;
}
int aux,minimo;
for(int i=0;i<vsize;i++)
{
  minimo=i;
  for(int j=i+1;j<vsize;j++)
  {
    if(vecl[j]<vecl[minimo])
    {
      minimo=j;
    }
  }
  aux=vecl[i];
  vecl[i]=vecl[minimo];
  vecl[minimo]=aux;
}
...

```

**Figura 4.** Codificación del método de selección.

En la figura 4 se observa el código escrito para el ordenamiento de los datos de los vectores, empleando el método de selección.

#### **d. Análisis comparativo.**

En esta sección del trabajo se ha de tomar en cuenta tres parámetros o aspectos que se tomaran en cuenta tanto del código realizado como de las herramientas empleadas para su desarrollo.

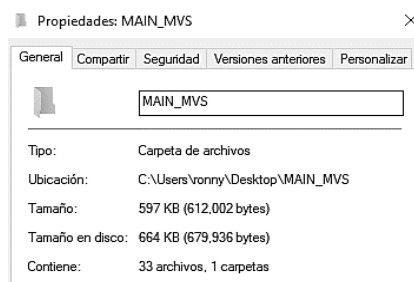
Los parámetros que se han analizado son:

1. Tamaño de proyecto creado por los tres IDE'S más utilizados
2. Tiempo de compilación del código por los tres IDE'S más utilizados
3. El método de ordenación más utilizado para realizar esta acción

#### **1. Tamaño del proyecto generado por los tres IDE'S más usados:**

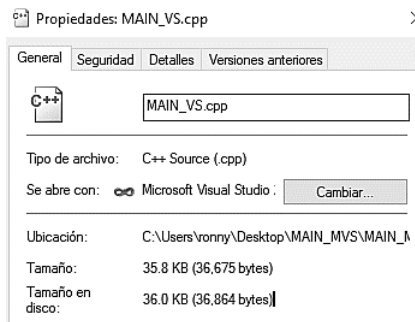
Para comenzar se tomó el aspecto tamaño del proyecto generado donde se llegó a visualizar la primera gran diferencia que existe entre estos 3 IDE'S.

Microsoft Visual Studio: En este primer IDE el proyecto alcanzo un tamaño de 597 KB (612,002 bytes) y en el archivo .cpp se alcanzó un tamaño de 35.8 KB (36,675 bytes).



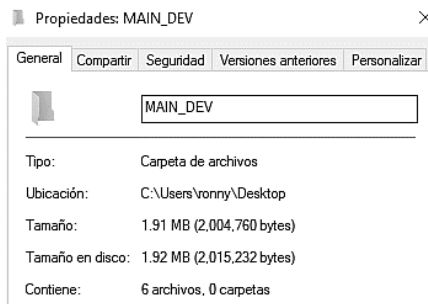
**Figura 5.** propiedades del proyecto en MVS.

En la figura 5 se observa las propiedades del proyecto que se creó para ejecutar el código en el IDE de Microsoft Visual Studio.



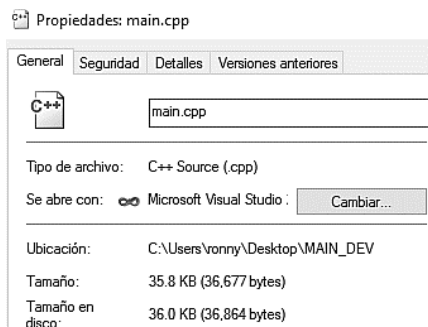
**Figura 6.** propiedades del archivo .cpp.

En la figura 6 se presentan las propiedades del archivo .cpp en donde se aloja el código creado. Dev C++: en este IDE el proyecto alcanzó un tamaño de 1.91 MB (2,004,760 bytes) y el archivo .cpp obtuvo un tamaño de 35.8 KB (36,677 bytes).



**Figura 7.** propiedades de proyecto en devc++.

En la figura 7 se presentan las propiedades del proyecto generado por el IDE dev c++.



**Figura 8.** propiedades del archivo .cpp en dev.

En la figura 8 se observan las propiedades del archivo .cpp perteneciente al IDE dev c++. Code::Blocks: en este último IDE el proyecto alcanzo un total de 181 KB (186,072 bytes) y el archivo .cpp alcanzó un tamaño de 35.8 KB (36,675 bytes).

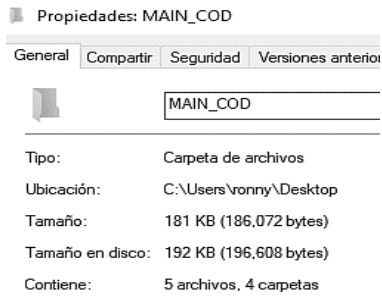


Figura 9. propiedades del proyecto en cb.

En la figura 9 se presentan las propiedades del proyecto generado en el IDE de Code::Blocks.

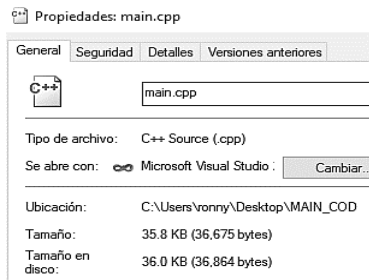


Figura 10. propiedades del archivo .cpp en cb.

En la figura 10 se observan las propiedades del archivo .cpp donde se desarrolló el código en el IDE de Code::Blocks.

## 2. Tiempo de compilación de los tres IDE'S

En esta sección se envió a compilar el código realizado en los tres IDE'S más utilizados para el manejo del lenguaje C++, hay que tener en cuenta que los tiempos de compilación pueden variar dependiendo varios factores como el hardware del equipo que se emplee o las veces que se ha compilado el proyecto.

- Microsoft Visual Studio: 0.26 seg.
- Dev C++: 3.23 seg.
- Code::Blocks: 3 seg.

## RESULTADOS

Datos estadísticos de los resultados obtenidos:

- Tamaño del archivo del proyecto generado por los IDE'S

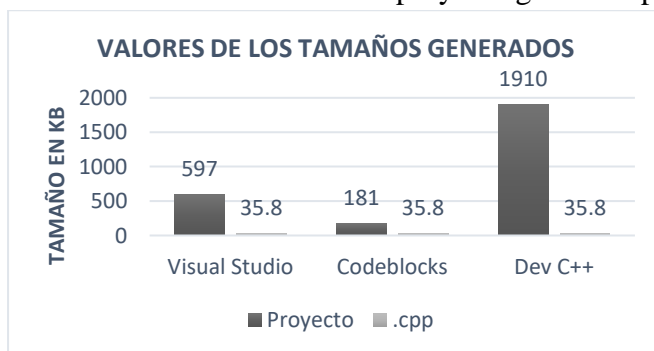
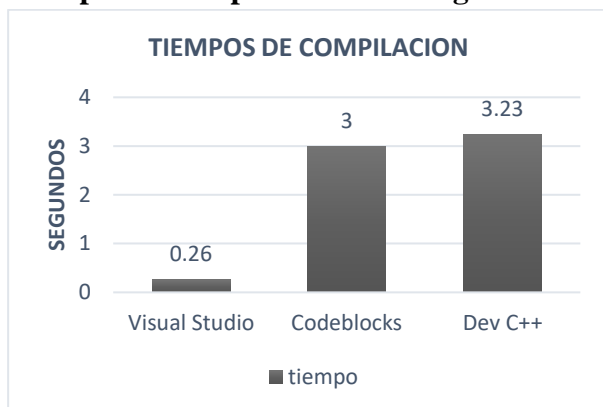


Gráfico 1. Grafica de los tamaños generados.

En el gráfico 1 se observan los valores obtenidos de los tres IDE'S al crear el proyecto para el desarrollo del ejercicio planteado.

A partir de esta tabla ya se puede tener una idea de que IDE puede ser más efectivo emplear al momento de generar un proyecto.

- **Tiempos de compilación del código en los tres IDE'S**



**Gráfico 2.** Grafica con los tiempos de compilación.

En el gráfico 2 se puede evidenciar los tiempos de compilación referentes a cada IDE que se utilizó. En este aspecto hay que tener en cuenta que los valores del tiempo de compilación pueden variar dependiendo a diferentes factores como lo son el hardware del equipo que se vaya a emplear, la cantidad de veces que se compila el código, entre otros factores más.

## CONCLUSIONES

Según los valores que se obtuvo a través del análisis que se ha realizado, bajo los parámetros establecidos se puede determinar que el IDE Microsoft Visual Studio es la mejor opción en cuanto a velocidad de compilación, aunque en la parte de la creación de proyectos se quede atrás del Code::Blocks esto no difiere tanto, ya que al Visual Studio se lo considera como un IDE de alto nivel para desarrollos de proyectos en comparación de los demás IDE'S, puestos que estos están enfocados para la enseñanza o para la práctica de los lenguajes que estos pueden manejar.

por ultimo topando el tema de los métodos de ordenamiento se pudo definir que, el método más famoso o más utilizado depende mucho de en qué situación se necesite emplear estos algoritmos, en el desarrollo de este parámetro se verifico que el método burbuja es más utilizado por estudiantes y profesores para aprender el manejo y el desarrollo de este método, mientras que los demás métodos como cuentan con un desarrollo más complejo y más elaborado lo emplean las personas que ya manejan este tema de ordenación de arrays, pero hay que tener en cuenta que el uso de cualquiera de estos métodos difiere directamente de la cantidad de conocimiento que se maneje sobre los métodos y el control de los arrays que se manejen.



## REFERENCIAS

- [1] G. Aburruzaga, I. Medina, and P. Francisco, *Fundamentos C++*, 2a. ed. Madrid(España): Universidad de Cadiz, 2006.
- [2] EcuRed contributors, “IDE de Programación.” [Online]. Available: [https://www.ecured.cu/index.php?title=Especial:Citar&page=IDE\\_de\\_Programación&id=2881136](https://www.ecured.cu/index.php?title=Especial:Citar&page=IDE_de_Programación&id=2881136). [Accessed: 05-Mar-2019].
- [3] colaboradores de wikipedia, “Detalles bibliográficos de Microsoft Visual Studi,” 2009. [Online]. Available: [https://es.wikipedia.org/w/index.php?title=Microsoft\\_Visual\\_Studio&oldid=114858532](https://es.wikipedia.org/w/index.php?title=Microsoft_Visual_Studio&oldid=114858532). [Accessed: 06-Mar-2019].
- [4] colaboradores de Wikipedia, “Detalles bibliográficos de Dev-C++,” 2019. [Online]. Available: <https://es.wikipedia.org/w/index.php?title=Dev-C%2B%2B&oldid=114235733>. [Accessed: 06-Mar-2019].
- [5] cppconference, “Funcion Main,” 2018. [Online]. Available: [https://es.cppference.com/mwiki/index.php?title=c++/language/main\\_function&oldid=32412](https://es.cppference.com/mwiki/index.php?title=c++/language/main_function&oldid=32412). [Accessed: 29-Mar-2019].
- [6] Joyanes Aguilar Luis, *Programación en c++ Algoritmos estructura de datos y objetos.*, 2a.ed. Madrid(España): McGraw-Hill España, 2006.
- [7] EcuRed contributors, “Datos bibliográficos sobre Algoritmo de ordenamiento por selección,” *EcuRed.* [Online]. Available: [https://www.ecured.cu/index.php?title=Algoritmo\\_de\\_ordenamiento\\_por\\_selección&oldid=1424584](https://www.ecured.cu/index.php?title=Algoritmo_de_ordenamiento_por_selección&oldid=1424584). [Accessed: 01-Mar-2019].
- [8] Microsoft, “Microsoft Visual Studio.” USA, 1997.
- [9] Bloodshed Software and Johan Mes, “Dev-C++.” USA, 2005.
- [10] The Code::Blocks team, “Code::Blocks.” USA, 2005.
- [11] O. Vallejos and G. Arduino, “ALGORITMIA DE LA ORDENACIÓN EFICIENTE Necesidad de Ordenar Eficientemente Análisis Algorítmico.” pp. 569–578, 2012.