

Programando en la nube con C++

Programming in the cloud with C++

Laura Adriana Garay¹

lgaray@ucse.edu.ar

Universidad Católica de Santiago del Estero. Departamento Académico San Salvador,
San Salvador de Jujuy, Jujuy, Argentina

Mónica Liliana Mamani²

monicaliliana.mamani@ucse.edu.ar

Universidad Católica de Santiago del Estero. Departamento Académico San Salvador,
San Salvador de Jujuy, Jujuy, Argentina

Mariano Alfredo Sumbaino³

Universidad Católica de Santiago del Estero. Departamento Académico San Salvador,
San Salvador de Jujuy, Jujuy, Argentina

¹ Especialista en Enseñanza de la Educación Superior (Universidad Católica de Cuyo)

Educador Internacional En Ingeniería-IGIP (International Society for Engineering Pedagogy)

Ingeniera en Computación (Universidad Católica de Sgo del Estero)

Profesor Asociado de la materias Estructura de Datos, Matemática Discreta, Sistemas Operativos I y Sistemas Operativos II de la carrera de Ing. en Informática de UCSE-DASS.

Ing. en Computación (Universidad Católica de Sgo del Estero-Departamento Académico San Salvador)

² Experta Universitaria en Enseñanza de la Educación Superior (Universidad Católica de Sgo del Estero-Departamento Académico San Salvador)

Profesora Adjunta de la materia Estructura de Datos y Matemática Discreta de la carrera de Ing. en Informática de UCSE-DASS

³ Alumno 4to año de la carrera Ing. en Informática de UCSE-DASS.

Diplom. Universitario en Video Juego (UCSE-DASS)

Resumen

En la mayoría de las materias iniciales de la carrera Ing. en Informática relacionadas a la programación, los alumnos programan en IDEs de escritorio, con las limitaciones que su utilización conlleva, entre ellas, dependencia del equipo en el cual se ha instalado dicho entorno.

Actualmente la NUBE es una herramienta tecnológica que, entre otras, brinda el servicio de DESARROLLO en diferentes plataformas.

Iniciar a los alumnos en la utilización de esta herramienta, no solo los capacita en su utilización, sino que tienen la posibilidad de experimentar el trabajo colaborativo en el diseño de un programa, por más simple que este sea.

Desde la cátedra Estructura de Datos de la carrera de Ingeniería en Informática de la Universidad Católica de Santiago del Estero del Departamento Académico San Salvador, se propone, determinar las ventajas de usar entornos de programación en línea/online en la resolución de problemas básicos utilizando diferentes estructuras de datos (pilas, colas, listas, etc.).

Palabras clave

Programación, nube, estructura de datos

Abstract

In most of the initial subjects of the Bachelor's Degree in Computer Engineering linked to programming, students program in desktop IDEs, with the limitations that their use entails, including a dependence on the equipment on which this environment was installed.

Currently, the CLOUD is a technological tool that, among others, provides the DEVELOPMENT service on different platforms.

Introducing students to the use of this tool not only trains them in its application, but also they have the possibility of experiencing collaborative work in the design of a program, no matter how simple it may be.

From the Data Structure chair of the Bachelor's Degree in Computer Engineering of the Catholic University of Santiago del Estero from San Salvador Academic Department, there is a purpose to determine the advantages of using online programming environments in solving basic problems using different data structures (Stacks, queues, lists, etc.).

Key Words

Programming, cloud, data structure

Introducción Justificación del problema.

Especificar, proyectar y desarrollar software es una de las competencias específicas de la carrera Ingeniería en Informática. Para contribuir a esta competencia, desde el primer año de estudio la enseñanza de la programación está presente. En la mayoría de los casos, la programación se realiza sobre un IDE (Integrated Development Environment) instalado en la computadora de trabajo, lo cual conlleva tareas de instalación del entorno a utilizar, que si bien no tiene mayores dificultades si presenta una dependencia respecto al equipo donde se instaló dicho entorno.

Hoy en día, el uso de la nube se ha incrementado considerablemente en diversos aspectos y el desarrollo de programas no es una excepción. Por tal razón, desde la cátedra Estructura de Datos de la carrera de Ingeniería en Informática de la Universidad Católica de Santiago del Estero del Departamento Académico San Salvador, se propuso en el marco de un proyecto de investigación iniciar a los alumnos en el uso de este servicio desde la programación en la resolución de problemas básicos y con el uso de las diferentes estructuras de datos, con la finalidad de que puedan descubrir las ventajas de utilizar un entorno on-line frente a uno de escritorio, como así también analizar los diferentes inconvenientes que éste puede presentar

Antecedentes

Actualmente durante el cursado de esta materia como de otras relacionadas con la programación, a saber, Fundamentos de Informática y Programación I, la práctica en programación se realiza mediante IDEs de escritorio, para lo cual, en el caso de realizar la práctica en los laboratorios de la Universidad, la instalación es realizada por personal técnico, y si los alumnos cuentan con su propio equipo, la instalación es su responsabilidad. En esta asignatura el objetivo es el estudio de las diferentes estructuras de datos, tales como Pilas, Colas, Listas, Arboles y Grafos utilizando Dev C++ como entorno de desarrollo (IDE de escritorio) para la implementación y prueba de estas estructuras en la resolución de diferentes ejercicios básicos.

Según Sitepoint⁴, utilizar la nube para programar tiene las siguientes virtudes:

- Son algo más que un editor de código: igual que los IDEs se podrían definir como editores de código con esteroides, las plataformas de desarrollo en la nube podrían pasar por ser IDEs con esteroides. Este tipo de herramientas son más bien Plataformas de Desarrollo como Servicio (Development Platform as a Service- dPaaS), que ofrecen una serie de funcionalidades sobre costes y productividad más elevadas.

⁴ Aldo Ziflaj (September 29, 2015). Developing in the Cloud: an Introduction. <https://www.sitepoint.com/developing-in-the-cloud-an-introduction/>

- Menos tiempo de implementación: este tipo de plataformas reducen mucho los tiempos de instalación. Menos tiempo, menos costes.
- Programa desde cualquier sitio: no es necesario que el programador esté delante de la máquina para continuar programando su producto. Al estar alojada en la nube, es posible hacerlo desde cualquier sitio y dispositivo (computadora o tableta). Único requisito: estar conectado a la Red
- Trabajo colaborativo en tiempo real: una de las grandes ventajas es que varios desarrolladores pueden estar trabajando en el mismo proyecto a la vez y utilizar servicios de chat online para comunicarse.
- Personalización del entorno de desarrollo: este tipo de herramientas permiten instalar dependencias para los proyectos de forma independiente. Cuando se programa en local, en muchas ocasiones se pueden tener complicaciones porque las dependencias para unos proyectos afectan a otros o perjudican a otras aplicaciones web.

Teniendo en cuenta las ventajas mencionadas, y la forma de trabajo actual en diferentes materias de programación, desde este proyecto se pretende iniciar y mostrar a los alumnos una forma distinta de trabajo para una de las tareas básicas de un informático, como lo es la Programación.

Marco teórico referencial.

Estructuras de Datos⁵: En el desarrollo de programas, existe una fase previa a la escritura del programa, esta es el diseño del algoritmo que conducirá a la solución del problema, en esta fase también deberá considerarse la estructura de datos que se va a utilizar. El término estructura de datos se refiere a la forma en que la información está organizada dentro de un programa. La correcta organización de datos puede conducir a algoritmos más simples y eficientes.

Estructura de datos: Conjunto de variables agrupadas y organizadas de cierta forma para representar un comportamiento.

Las estructuras de datos según su tamaño en memoria se clasifican en: Estructuras de datos estáticas: Son aquellas cuyo tamaño en memoria es fijo, por ejemplo, los arreglos.

Estructuras de datos dinámicas: Son las estructuras que permiten variar su tamaño en memoria de acuerdo con las necesidades del ambiente, por ejemplo, listas enlazadas.

La Nube⁶: Es un modelo tecnológico en el cual encontramos muchos servicios. Uno de ellos es almacenar información en línea y después acceder a ella desde cualquier lugar, usando cualquier dispositivo que nos permita conectarnos a internet. Existen plataformas como Google Drive, OneDrive, iCloud, Dropbox, entre muchas otras, desde las cuales se puede

⁵ Cairó, Osvaldo Guardati, Silvia (2006). Estructuras de Datos. Mc Graw Hill

⁶ GCFGlobal. GCF Aprende Libre. <https://edu.gcfglobal.org/es/como-funciona-la-nube/>

hacer esto. Ofrece servicios tales como:

Almacenamiento de archivos: Permite almacenar archivos en servidores de internet y después acceder a ellos desde cualquier lugar, usando un dispositivo con conexión a internet. Con esto no es necesario que los archivos estén guardados en la computadora y no se corre el riesgo de perderlos en caso de que el equipo sufra algún daño.

Por lo general, este es el servicio que la mayoría de las personas conoce y maneja. Hay diferentes plataformas como OneDrive, iCloud, Google Drive, Dropbox, entre otras, que se encargan de proveer este servicio. Usar estas plataformas es muy sencillo, solamente se debe ingresar a ellas y crear una cuenta. La mayoría dará cierta cantidad de espacio de forma gratuita, pero si se necesita más, se pueden adquirir paquetes más grandes.

Infraestructura: Algunas compañías que manejan una gran cantidad de información contratan empresas especializadas como Amazon Web Services que cuentan con el hardware o máquinas necesarias para procesar datos.

Computación o Software: En este caso, se presta los servicios de software, es decir, se alquila por utilizar un programa. Suele estar ligado al servicio de infraestructura, porque no solo se contrata el programa, sino además la máquina que lo procesará.

Objetivos

Objetivo General

- Determinar las ventajas de usar entornos de programación on-line en la resolución de problemas básicos con estructuras de datos

Objetivos Específicos

- Buscar y analizar entornos de programación on-line que soporten C++
- Comparar entornos de programación on-line frente al de escritorio

Estrategia metodológica

Este proyecto se desarrolló en el marco de la cátedra Estructura de Datos, correspondiente al 2do año de la carrera de Ingeniería en Informática.

Las actividades realizadas consistieron en

- BÚSQUEDA Y ANÁLISIS DE ENTORNOS DE PROGRAMACIÓN ON LINE que soporten la programación en C++, lenguaje que los alumnos vienen aplicando desde la materia Fundamentos de Informática.
- DETERMINACIÓN DEL ENTORNO MAS ADECUADO para los trabajos prácticos con los estudiantes
- DETERMINACIÓN DE VENTAJAS Y DESVENTAJAS del entorno usado en las prácticas
- COMPARACIÓN de los entornos de programación on-line vs. el de escritorio
- CONCLUSIONES respecto a cuál de los entornos resultó más adecuado para el trabajo de los estudiantes

Resultados alcanzados

Realizada la búsqueda de IDEs en la nube que permitan la programación en C++, se obtuvieron los siguientes resultados.

Ide's en la nube para C++

IDE's	DESCRIPCIÓN	CARACTERÍSTICAS
IDEOne	Su nombre, auto explicativo de la misión que cumple: <i>un solo IDE y vía Web</i> . Este IDE de programación online nos permite programar, compilar y depurar código en más de 60 lenguajes de programación diferentes, siendo una de las plataformas más completas disponibles en la red. Podemos destacar los lenguajes C, C++ , C#, Java, PHP y Bash.	Las conexiones con los servidores de IDEOne se realizan a través de HTTPS para evitar que usuarios no autorizados puedan capturar código perteneciente a un usuario en particular. Además, al registrarse en la plataforma, se podrá configurar los entornos como "privados" para que solo el dueño, tenga acceso a su código y nadie más. Antes de ejecutar el proyecto, se deben de ingresar los datos en el flujo de entrada (stdin), los cuales se tomarán en conjunto con el código fuente creado, para su posterior compilación y ejecución sobre los cuales se obtendrán los resultados esperados.
CodePad	A diferencia de otros compiladores online, Codepad no tiene esa parte ligera de IDE con debugger, sino que tiene como fin el poder pegar rápidamente un texto y que nos devuelva una página que podamos pasarle a cualquier persona con el código y el resultado tras ejecutarlo. No admite una gran cantidad de lenguajes, pero si varios de los más usados, como: C, C++ , D, Haskell, Lua, OCaml, PHP, Perl.	Esta herramienta se caracteriza por ser mucho más simple y rápida que IDEOne. En su página web simplemente debemos pegar el código y pulsar sobre el botón "submit". El código se envía al servidor donde se compila y ejecuta o interpreta para mostrar el resultado. Además, nos facilita una URL para compartir el proyecto con otras personas, pudiendo ayudar estas a mejorar el código. Esta IDE, no permite trabajar con valores variables, lo cual resulta una desventaja al querer trabajar con proyectos en los cuales se soliciten el ingreso de datos para realizar cálculos y la posterior obtención de resultados.
Repl.it	Poderoso entorno interactivo en la nube, para lenguajes de programación. Repl.it, soporta la codificación de cualquier lenguaje compilado en su navegador, desde Python hasta C++ o incluso Swift para aplicaciones de iOS.	En dicho entorno se puede ejecutar código sin instalar nada, es un IDE basado en el navegador, y tiene incorporado un compilador que opera desde cualquier sistema operativo. A diferencia del resto de las IDE'S mencionadas, en Repl.it el flujo de ingreso de datos es solicitado en tiempo de ejecución.

Tabla 1: Descripción IDEs en la nube

De acuerdo con lo investigado se consideraron las siguientes ventajas y desventajas que presentaban cada uno de los IDEs considerados

	IDEOne	CodePad	Repl.it
←Z→	<p>Las conexiones con sus servidores, evitan que usuarios no autorizados, tomen código de un usuario en particular</p> <p>Funciona como herramienta complementaria, donde se pueden realizar pruebas con fragmentos de código</p> <p>Es una IDE vía web que nos permite escribir comentarios al estilo post de un blog</p>	<p>Facilita una URL para compartir los proyectos con otros usuarios</p> <p>Permite compartir el editor</p> <p>Posee resaltado de sintaxis</p> <p>Permite copiar/pegar partes importantes del código para compartir online</p>	<p>Es de fácil funcionamiento</p> <p>Si el proyecto tiene la habilidad de recargar automáticamente el navegador cuando se ejecuta algún cambio en el código, este navegador también puede realizarlo</p> <p>Puede utilizarse desde cualquier computadora o dispositivo móvil</p> <p>Es útil para la enseñanza, permitiendo crear clases virtuales e invitar a los alumnos a llevar un control de su progreso</p>
SA ←Z→	<p>No es recomendable para un desarrollador que recién está comenzando a programar, puesto que el mecanismo de ingreso de datos es poco intuitivo</p> <p>Tiene límites de tiempo por ej. Para compilar, 10'; en la ejecución, 5' y límite de memoria, de 256 MB</p> <p>No posee autocompletado de sintaxis</p> <p>No permite el uso de cabeceras (headers .hpp)</p>	<p>No admite una gran cantidad de lenguajes</p> <p>Es lento al momento de la depuración de un código</p> <p>No permite el uso de cabeceras (headers .hpp)</p>	<p>No permite el uso de cabeceras (headers .hpp)</p>

Tabla 2 : IDEs en la nube-Ventajas y Desventajas

Prueba y Práctica de cada uno de los IDEs: consideradas las características de cada uno de los IDEs, se procedió a la prueba de estos, mediante la realización de ejercicios simples de programación con la finalidad de determinar lo necesario para llevar a cabo su implementación

A continuación, se muestra uno de los códigos con los que se trabajó en las tres plataformas a fin de proceder a la elección de una de ellas, en este código se muestra la carga y muestra de dos matrices cuadradas, y la obtención del producto de los elementos de la diagonal principal:

```
Ejemplo.cpp
1  #include <iostream>
2  #include <stdlib.h>
3
4  using namespace std;
5
6  void cargaMatriz(int M[10][10],int tam){
7      int i, j, c=1;
8      for (i=0;i<tam;i++){
9          for (j=0;j<tam;j++){
10             cout<<c++<<" nro entero: ";
11             cin>>M[i][j];
12         }
13     }
14 }
15
16 void muestraMatriz(int M[10][10],int tam){
17     int i, j;
18     for (i=0;i<tam;i++){
19         for (j=0;j<tam;j++){
20             cout<<M[i][j]<<"\t";
21         }
22         cout<<endl;
23     }
24 }
25
26 int main(){
27     int t, i, j, m;
28
29     int A[10][10]; //declara la matriz A
30     int B[10][10]; //declara la matriz B
31
32     do {
33         cout<<"Ingrese el tamaño de la matriz (Entre 2 y 10): ";
34         cin>>t;
35     }while ((t<2)||t>10);
36
37     system ("cls");
38
39     //CARGA MATRIZ A
40     cout<<"Carga de la matriz A (debe ingresar "<t*t<<" nros enteros)"<<endl;
41     cargaMatriz(A,t);
42
43     cout<<"Carga de la matriz B (debe ingresar "<t*t<<" nros enteros)"<<endl;
44     cargaMatriz(B,t);
45
46     cout<<endl<<"Muestra de las matrices cargadas"<<endl;
47     cout<<endl<<"Matriz A"<<endl;
48     muestraMatriz(A,t);
49     cout<<endl<<"Matriz B"<<endl;
50     muestraMatriz(B,t);
51
52     cout<<endl<<"Producto de las diagonales principales: "<<endl;
53     for (i=0;i<t;i++){
54         int j=0;
55         while (j<i){
56             cout<<" "<<"\t";
57             j++;
58         }
59         cout<<A[i][i]*B[i][i]<<endl;
60     }
61 }
62
```

Imagen 1: Código Ejemplo

1ra Plataforma: CodePad

url: <http://codepad.org/>

- 1- Se realiza el registro correspondiente en dicho compilador online
- 2- Se configura la cuenta creada, definiendo el lenguaje a trabajar como predeterminado
- 3- Se crea (escribe) el nuevo código

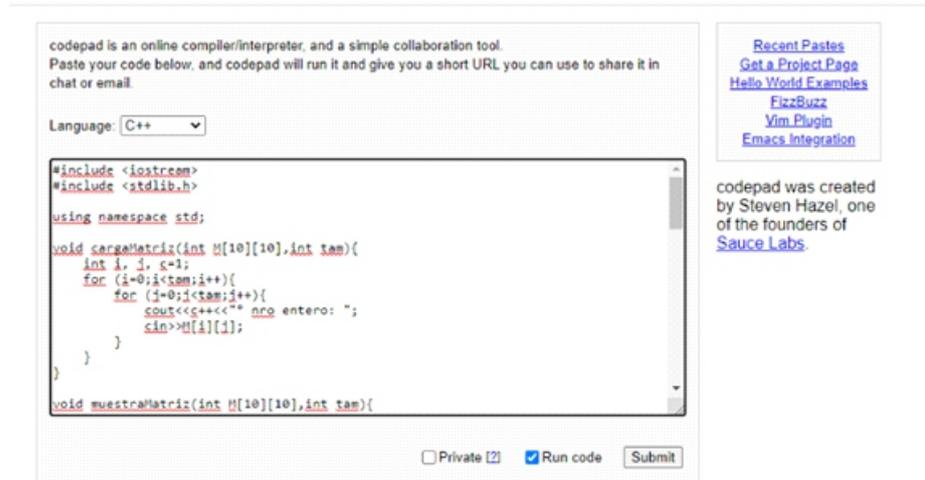


Imagen 2: CodePad. Codificación

La plataforma no permite trabajar con ingreso de datos.

logaray - C++, pasted just now.

public (make private)

```

1  #include <iostream>
2  #include <stdlib.h>
3
4  using namespace std;
5
6  void cargaMatriz(int M[10][10],int tam){
7      int i, j, c=1;
8      for (i=0;i<tam;i++){
9          for (j=0;j<tam;j++){
10             cout<<c++<<" nro entero: ";
11             cin>>M[i][j];
12         }
13     }
14 }
15
16 void muestraMatriz(int M[10][10],int tam){
17     int i, j;
18     for (i=0;i<tam;i++){
19         for (j=0;j<tam;j++){
20             cout<<M[i][j]<<"\t";
21         }
22         cout<<endl;
23     }
24 }
25
26 int main(){
27     int t, i, j, m;
28
29     int A[10][10]; //declara la matriz A
30     int B[10][10]; //declara la matriz B
31
32     do {
33         cout<<"Ingrese el tamaño de la matriz (Entre 2 y 10): ";
34         cin>>t;
35     }while ((t<2)||t>10);
36
37     system ("cls");
38
39     //CARGA MATRIZ A
40     cout<<"Carga de la matriz A (debe ingresar " <<t<<" nros enteros)"<<endl;
41     cargaMatriz(A,t);
42
43     cout<<"Carga de la matriz B (debe ingresar " <<t<<" nros enteros)"<<endl;
44     cargaMatriz(B,t);
45
46     cout<<endl<<"Muestra de las matrices cargadas"<<endl;
47     cout<<endl<<"Matriz A"<<endl;
48     muestraMatriz(A,t);
49     cout<<endl<<"Matriz B"<<endl;
50     muestraMatriz(B,t);
51
52     cout<<endl<<"Producto de las diagonales principales: "<<endl;
53     for (i=0;i<t;i++){
54         int j=0;
55         while (j<i){
56             cout<<" " <<"\t";
57             j++;
58         }
59         cout<<A[i][i]*B[i][i]<<endl;
60     }
61 }

```

Output:

```

1  Ingrese el tamaño de la matriz (Entre 2 y 10): Ingrese el tamaño de la matriz (Entre 2 y 10): Ingrese
2  Timeout

```

Imagen 3: CodePad. Resultado de la ejecución

Al quitar el ingreso de los datos, se puede observar que el programa obtiene los resultados establecidos.

```

1 #include <iostream>
2 #include <stdlib.h>
3
4 using namespace std;
5
6 void cargaMatriz(int M[10][10],int tam){
7     int i, j, c=1;
8     for (i=0;i<tam;i++){
9         for (j=0;j<tam;j++){
10            M[i][j]=c++;
11        }
12    }
13 }
14
15 void muestraMatriz(int M[10][10],int tam){
16     int i, j;
17     for (i=0;i<tam;i++){
18         for (j=0;j<tam;j++){
19             cout<<M[i][j]<<"\t";
20         }
21         cout<<endl;
22     }
23 }
24
25 int main(){
26     int t, i, j, m;
27
28     int A[10][10]; //declara la matriz A
29     int B[10][10]; //declara la matriz B
30
31     t=3;
32
33     //CARGA MATRIZ A
34     cout<<"Carga de la matriz A (debe ingresar " <<t<<" nros enteros)"<<endl;
35     cargaMatriz(A,t);
36
37     cout<<"Carga de la matriz B (debe ingresar " <<t<<" nros enteros)"<<endl;
38     cargaMatriz(B,t);
39
40     cout<<endl<<"Muestra de las matrices cargadas"<<endl;
41     cout<<endl<<"Matriz A"<<endl;
42     muestraMatriz(A,t);
43     cout<<endl<<"Matriz B"<<endl;
44     muestraMatriz(B,t);
45
46     cout<<endl<<"Producto de las diagonales principales: "<<endl;
47     for (i=0;i<t;i++){
48         int j=0;
49         while (j<i){
50             cout<<" "<<"\t";
51             j++;
52         }
53         cout<<A[i][i]*B[i][i]<<endl;
54     }
55 }
56

```

Output:

```

1 Carga de la matriz A (debe ingresar 9 nros enteros)
2 Carga de la matriz B (debe ingresar 9 nros enteros)
3
4 Muestra de las matrices cargadas
5
6 Matriz A
7 1 2 3
8 4 5 6
9 7 8 9
10
11 Matriz B
12 1 2 3
13 4 5 6
14 7 8 9
15
16 Producto de las diagonales principales:
17 1
18 25
19 81

```

Imagen 4: CodePad.Ejecución sin ingreso de datos

2da Plataforma: IdeOne

url: <https://ideone.com/>

- 1- Se realiza el registro correspondiente en dicho compilador online
- 2- Se configura la cuenta creada, definiendo el lenguaje a trabajar como predeterminado
- 3- Se crea/escrbe el nuevo código

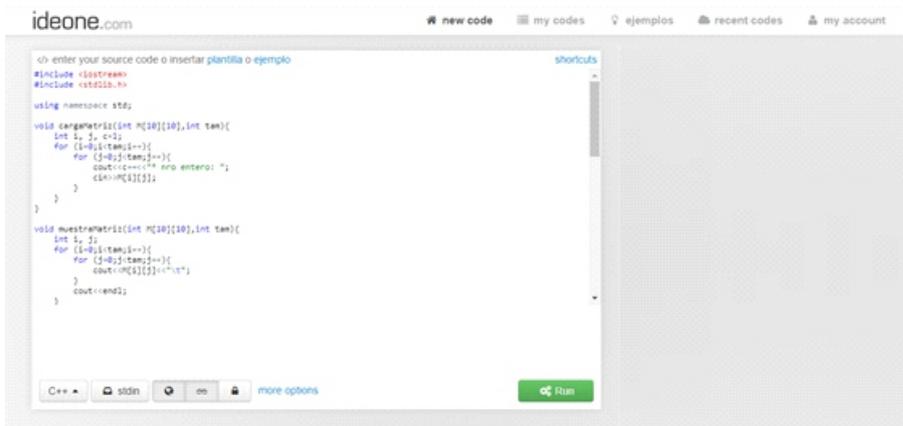


Imagen 5: IdeOne. Codificación

4- Este compilador permite trabajar con ingreso de datos, pero éstos no se otorgan durante la ejecución, sino que se registran previamente a la ejecución en el espacio del flujo de entrada (stdin), esto significa que se deben especificar todos y cada uno de los valores ingresados que el programa necesita para su ejecución, los mismos serán compilados en conjunto con el código fuente creado.

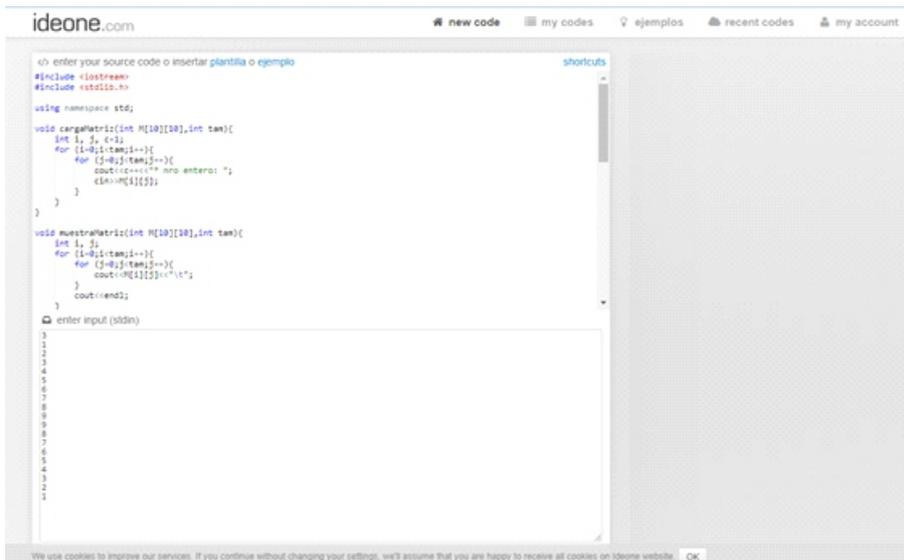


Imagen 6: IdeOne. Registro de los valores de ingreso

5- Al finalizar la compilación, se ejecuta el proyecto, realizando los cálculos con datos ingresados en STDIN devolviendo resultados.

```

1. #include <iostream>
2. #include <string.h>
3.
4. using namespace std;
5.
6. void cargaMatriz(int M[10][10],int tam){
7.     int i, j, c=1;
8.     for (i=0;i<tam;i++){
9.         for (j=0;j<tam;j++){
10.            cout<<""<<" nro entero: ";
11.            cin>>M[i][j];
12.        }
13.    }
14. }
15.
16. void muestraMatriz(int M[10][10],int tam){
17.     int i, j;
18.     for (i=0;i<tam;i++){
19.         for (j=0;j<tam;j++){
20.            cout<<M[i][j]<<" ";
21.        }
22.        cout<<endl;
23.    }
24. }
25.
26. int main(){
27.     int t, i, j, m;
28.
29.     int A[10][10]; //declara la matriz A
30.     int B[10][10]; //declara la matriz B
31.
32.     do {
33.         cout<<"Ingrese el tamaño de la matriz (Entre 2 y 10): ";
34.         cin>>t;
35.     }while ((t<2)||t>10);
36.
37.     system ("cls");
38.
39.     //CARGA MATRIZ A
40.     cout<<"Carga de la matriz A (debe ingresar " <<t<<" nros enteros)"<<endl;
41.     cargaMatriz(A,t);
42.
43.     cout<<"Carga de la matriz B (debe ingresar " <<t<<" nros enteros)"<<endl;
44.     cargaMatriz(B,t);
45.
46.     cout<<endl<<"Muestra de las matrices cargadas"<<endl;
47.     cout<<endl<<"Matriz A"<<endl;
48.     muestraMatriz(A,t);
49.     cout<<endl<<"Matriz B"<<endl;
50.     muestraMatriz(B,t);
51.
52.     cout<<endl<<"Producto de las diagonales principales: "<<endl;
53.     for (i=0;i<t;i++){
54.         int s=0;
55.         while (j<i){
56.             cout<<" " <<"\t";
57.             j++;
58.         }
59.         cout<<A[i][i]*B[i][i]<<endl;
60.     }
61. }
62.

```

Imagen 7: IdeOne. Ejecución (1ra parte)

```
stdin copy
3
1
2
3
4
5
6
7
8
9
9
8
7
6
5
4
3
2
1

stdout copy
Ingrese el tamaño de la matriz (Entre 2 y 10): Carga de la matriz A (debe ingresar 9 nros enteros)
1° nro entero: 2° nro entero: 3° nro entero: 4° nro entero: 5° nro entero: 6° nro entero: 7° nro entero:
8° nro entero: 9° nro entero: Carga de la matriz B (debe ingresar 9 nros enteros)
1° nro entero: 2° nro entero: 3° nro entero: 4° nro entero: 5° nro entero: 6° nro entero: 7° nro entero:
8° nro entero: 9° nro entero:
Muestra de las matrices cargadas

Matriz A
1 2 3
4 5 6
7 8 9

Matriz B
9 8 7
6 5 4
3 2 1

Producto de las diagonales principales:
9
  25
   9
```

Imagen 8: IdeOne. Ejecución (2da parte)

3ra Plataforma: Repl.it

url: <https://replit.com/>

- 1- Se realiza el registro correspondiente en dicho compilador online
- 2- Se configura la cuenta creada, definiendo el lenguaje a trabajar como predeterminado
- 3- Se crea/escrbe el nuevo código

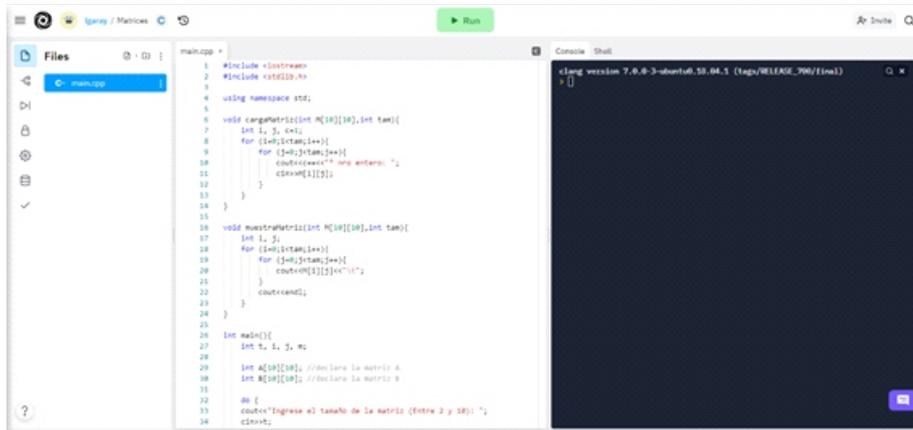


Imagen 9: Repl.it. Codificación

4- Esta plataforma divide la página permitiendo visualizar el código fuente y el espacio donde se sigue la ejecución del programa, como lo sería en una plataforma de escritorio.

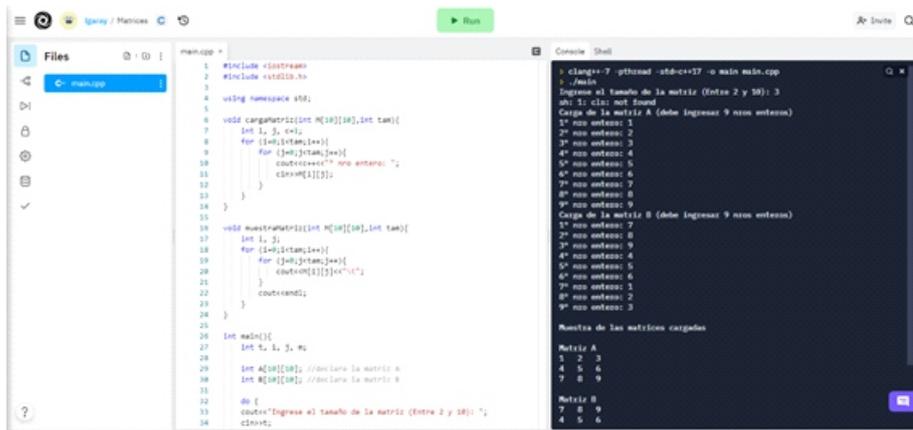


Imagen 10: Repl.it. Ejecución (1ra parte)

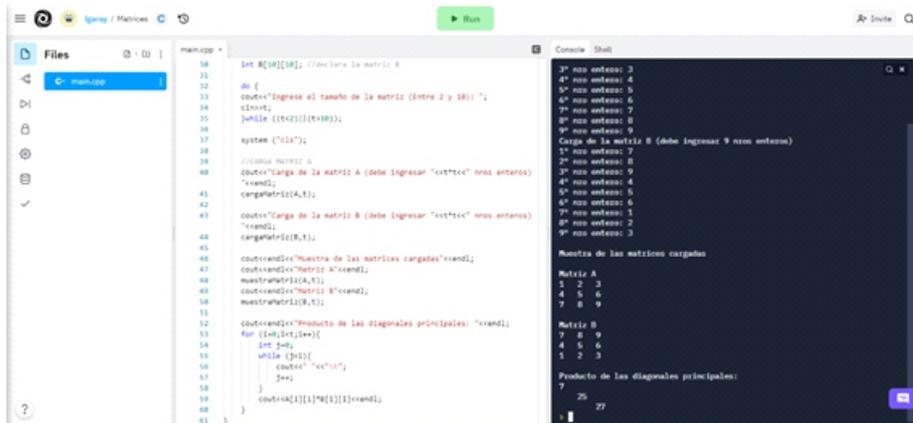


Imagen 11: Repl.it. Ejecución (2da parte)

Por las pruebas realizadas se determinó que, de los entornos de desarrollo online analizados, resulta ser el más eficiente Repl.it, ya que permite el ingreso de valores sin solicitarlos a priori (antes de la ejecución) pues se solicitan en tiempo de ejecución.

IDE's de Escritorio vs IDE's en la nube

A partir de lo investigado y lo trabajado en el presente proyecto, se elaboró la siguiente tabla comparativa.

Dadas las características y las necesidades en la resolución de los ejercicios planteados, el IDE en la nube seleccionado para la realización de los diferentes ejercicios prácticos fue Repl.it.

A continuación, se muestra una tabla comparativa entre el IDE de Escritorio y el IDE en la nube.

IDE de Escritorio (DEV C++)	IDE en la Nube (Repl.it)
Permite la utilización de librerías personalizadas, muy útil al trabajar con tipos de datos abstractos (Pilas, Colas y Listas)	No permite trabajar con librerías personalizadas. Por lo que las pruebas se realizaron siempre en un único archivo del tipo cpp y se trabajó con las librerías predeterminadas del lenguaje.
No permite trabajar en colaboración sobre un mismo proyecto	A través de la opción de Compartir, un mismo proyecto puede ser accedido por los integrantes de un equipo sin que cada uno deba tener la copia respectiva
Al instalarse se debe tener en cuenta el SO instalado en la computadora	Sea cual fuere el SO de la computadora el entorno de desarrollo funciona perfectamente

Tabla 3: IDE de escritorio vs IDE online

Luego de realizadas las actividades programadas en el presente proyecto, podemos concluir que los resultados obtenidos fueron **Satisfactorios**. Se logró el objetivo planteado a un nivel básico, teniendo en cuenta que se experimentó con ejercicios básicos, correspondientes a las primeras unidades de la materia.

Conclusiones

La cátedra siempre utilizó entornos de escritorio para la realización de los ejercicios prácticos, sea el DEV C++, Code Block, e incluso Zinjai. Cualquiera fuere, el mismo debe

instalarse en la computadora y lo realizado en ella, deberá ser grabado en la misma o, si debe ser transportado, se utilizará un medio de almacenamiento portable o en su defecto utilizar algún almacenamiento en la nube. Para continuar lo trabajado en otro equipo, en primer lugar, deberá instalarse en él el mismo entorno o en su defecto alguno compatible con el primero.

Al trabajar en un IDE on-line, no se requiere instalación, no es necesario almacenar lo realizado en algún espacio adicional y la compatibilidad no es un inconveniente. Además, se cuenta con la gran ventaja del trabajo colaborativo, lo cual contribuye a desarrollar una de las competencias más importantes de un futuro ingeniero en informática, “el trabajo en equipo”, sin necesidad de estar en el mismo espacio físico a la hora de resolver un problema.

Bibliografía

- Cairó, Osvaldo Guardati, Silvia (2006). Estructuras de Datos. Mc Graw Hill
- Joyanes Aguilar, Luis Zahonero Martinez, Ignacio (1998). Estructura de datos: Algoritmos, Abstracción y Objetos. Mc Graw Hill
- GFCGlobal. GFCAprendeLibre. <https://edu.gcfglobal.org/es/como-funciona-la-nube/>
- Desarrolladores, transformación digital (03 de noviembre, 2015). Desarrollo en la nube: ventajas y plataformas para programadores. <https://www.bbvaapimarket.com/es/mundo-api/desarrollo-en-la-nube-ventajas-y-plataformas-para-programadores/>
- Rubén Velasco (03 de marzo, 2016). Cómo programar, compilar y probar código desde la nube. <https://www.redeszone.net/2016/03/03/como-programar-compilar-y-probar-codigo-desde-la-nube/>
- Ideone.com (29 enero, 2010). Programa, compila y ejecuta in the cloud. <https://robleshermoso.wordpress.com/2010/01/29/ideone-com-programa-compila-y-ejecuta-in-the-cloud/>
- <https://adictec.com/los-mejores-ides-en-la-nube/>

Agradecimientos

Al Departamento Académico San Salvador de la Universidad Católica de Santiago del Estero, a la Ing. Mónica Mamaní, y al alumno Mariano Alfredo Sumbaino, quienes participaron activamente de este proyecto de investigación.