

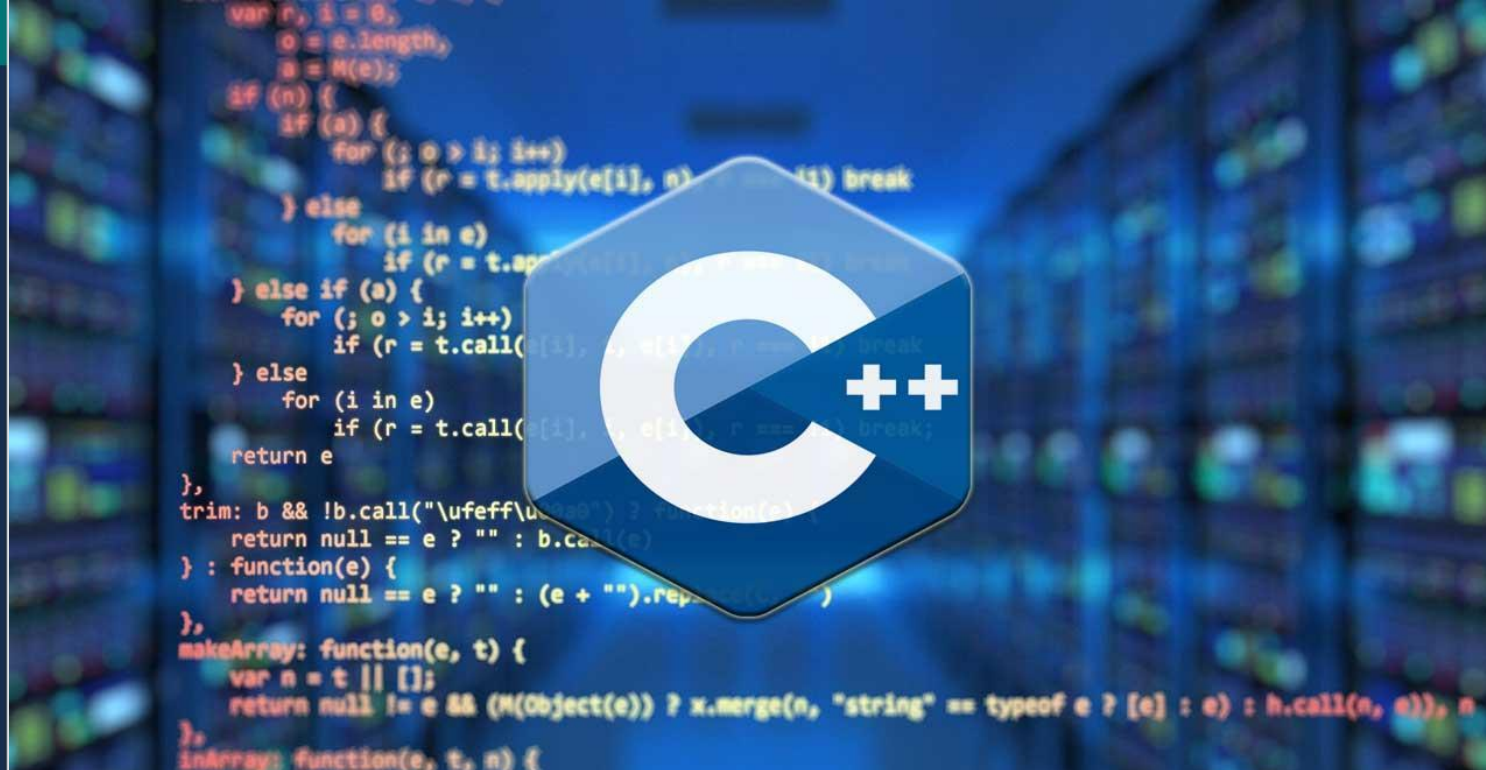
**POSTULACIÓN DE MATERIAL
ACADÉMICO**

FACULTAD DE INGENIERÍA

**PROGRAMA INGENIERÍA
SISTEMAS**



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación
y extensión de programas de pregrado, aplicando todos los
requisitos de las normas ISO implementadas en sus sedes
Neiva y Pitalito"

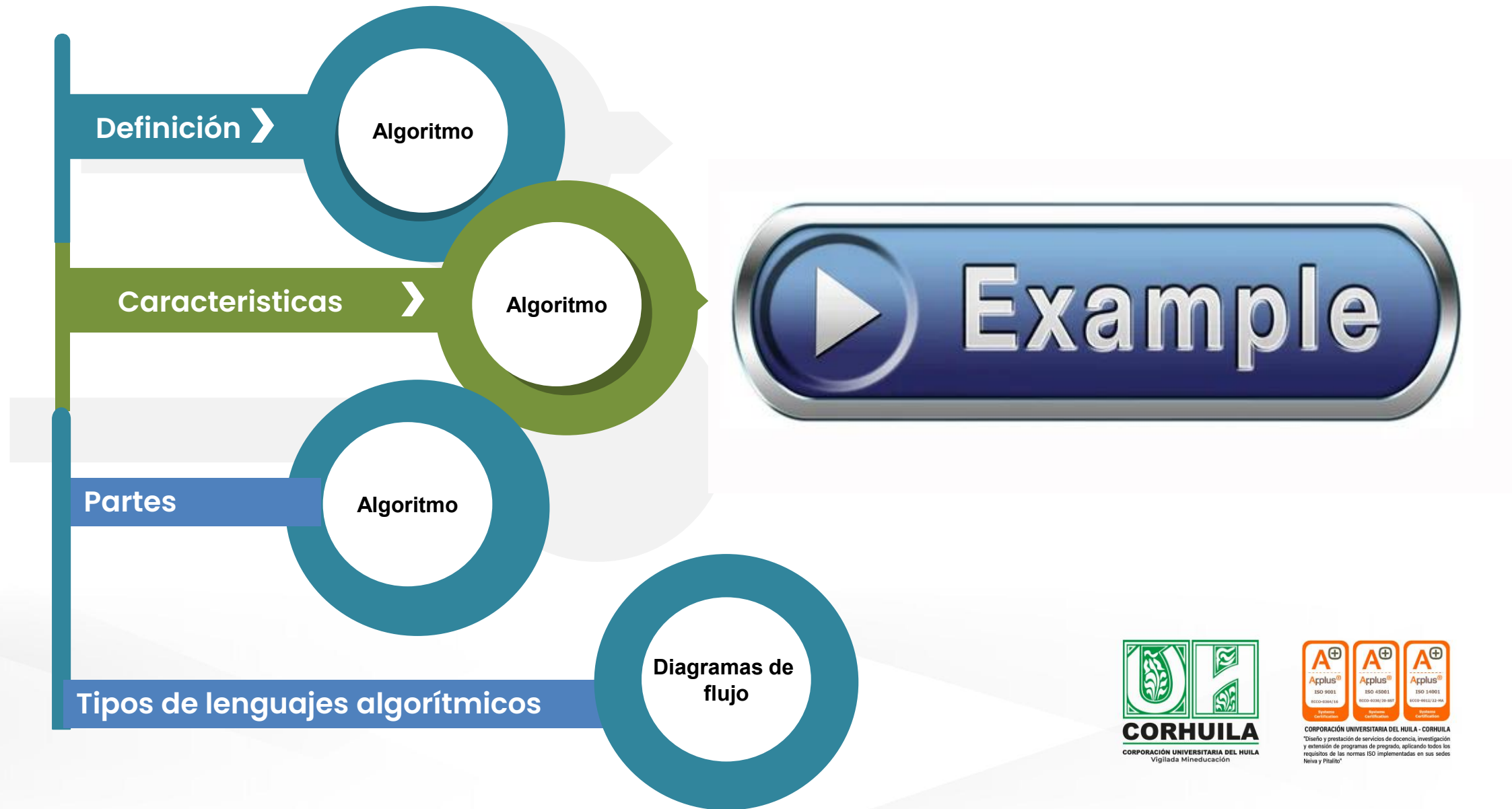


**Asignatura fundamentos de
programación**

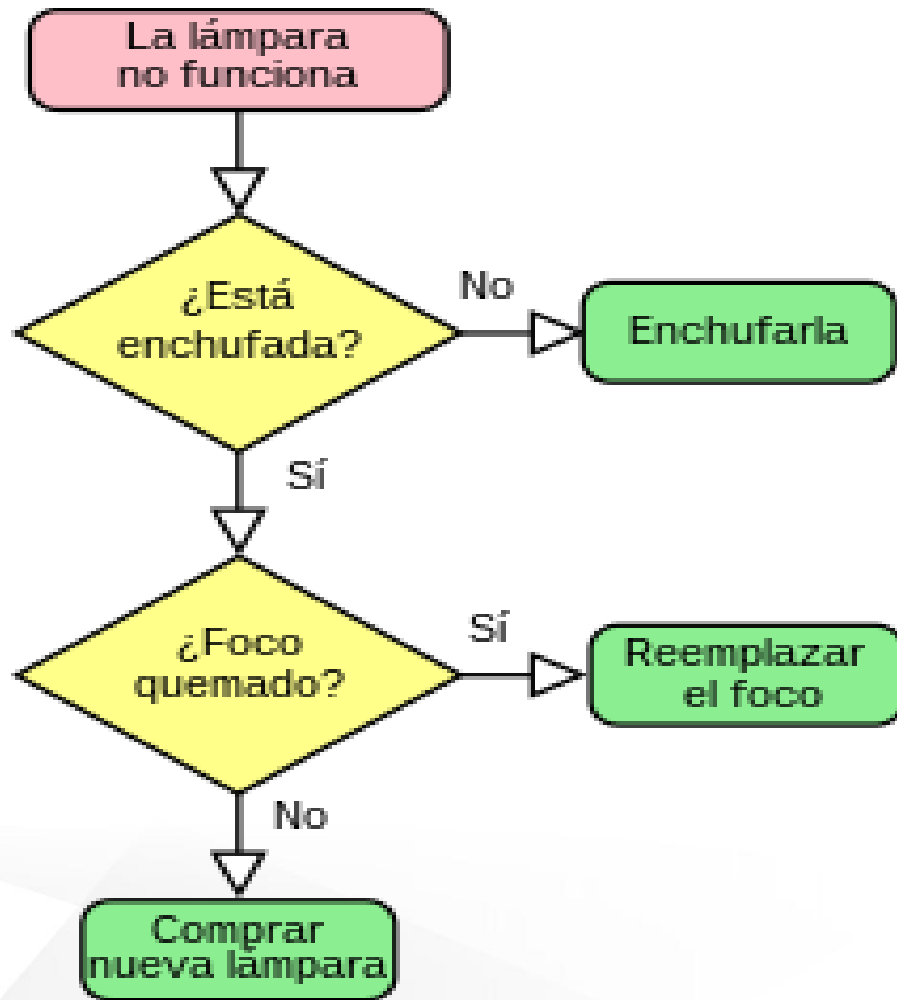
**Unidad 1: Introducción a Conceptos
de Programación**

Docente: Edisney Garcia Perdomo

Contenido



Algoritmo

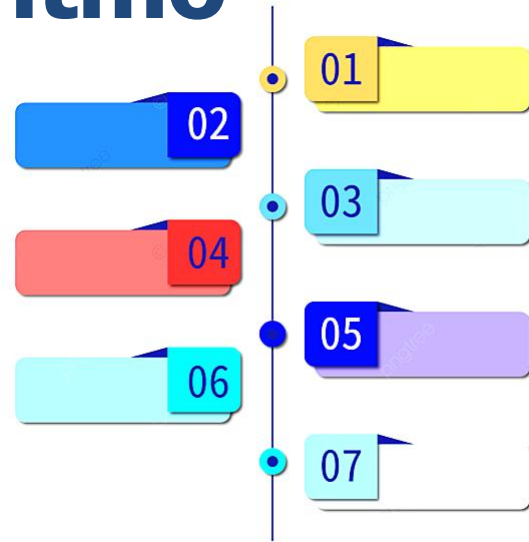


Fuente: <https://onx.la/d9023>

Un algoritmo es una secuencia de pasos lógicos, finitos y bien definidos que se siguen para resolver un problema o realizar una tarea específica. En programación, un algoritmo establece instrucciones precisas que guían el proceso de entrada, procesamiento y salida de datos, permitiendo que una computadora o persona ejecute la tarea de manera estructurada y eficiente.



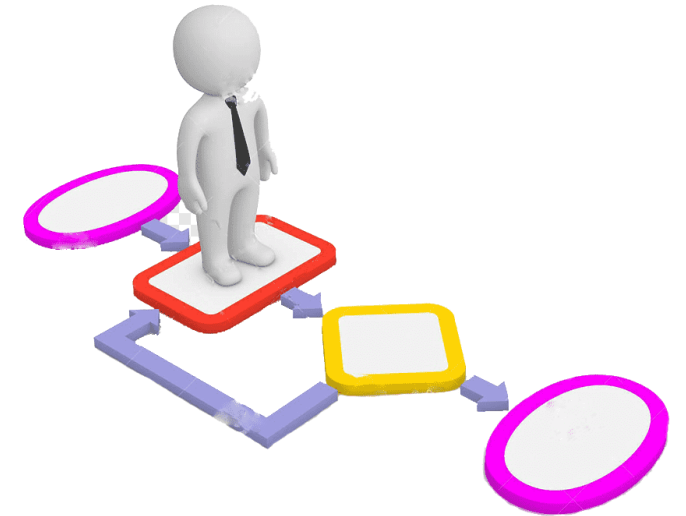
Algoritmo



Cualquier problema de la vida real se soluciona por medio de un algoritmo con múltiples representaciones.

Pasos secuenciales: significa que deben ser ejecutados uno después de otro.

Pasos ordenados: quiere decir que deben llevar un orden casi obligatorio (en la mayoría de los casos).



Fuente: <https://onx.la/d9023>



Algoritmo

Un algoritmo es una serie de pasos organizados, para dar solución a un problema específico.

Partes de un algoritmo

Entrada

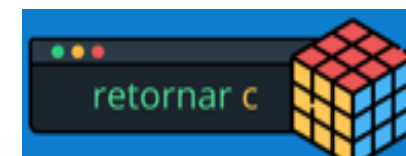
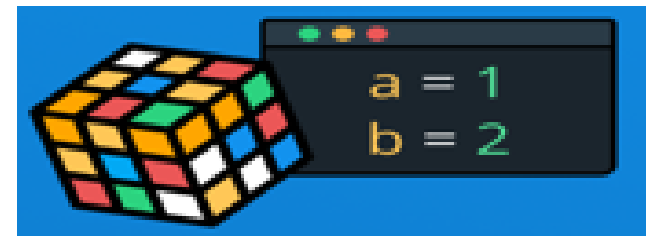
Son los datos que se le dan a un algoritmo

Proceso

Operación que se realiza con los datos.

Salida

Resultado final que se obtiene de la operación.
En este caso sera 3



Fuente: <https://onx.la/47c3d>

Características



Preciso

Tienen que resolver el problema, sin errores.



Definido

Si ejecuta el algoritmo varias veces, los datos de salida serán iguales.



Finito

Debe tener un inicio y fin.



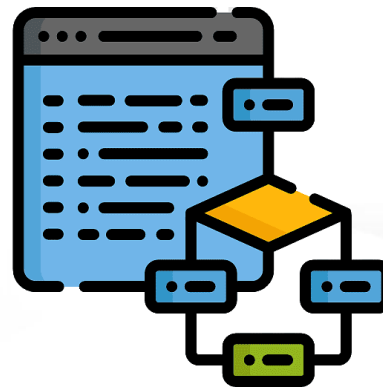
CORPORACIÓN UNIVERSITARIA DEL HUILA
Vigilada Mineducación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Otras características de los algoritmos

- Los resultados de los cálculos deben ser exactos, de manera rigurosa.
- Deben constar de pasos claros, precisos y no ambiguos.
- Los algoritmos son esenciales en la informática porque proporcionan una base estructurada para el desarrollo de programas y aplicaciones, independientemente del lenguaje de programación utilizado.

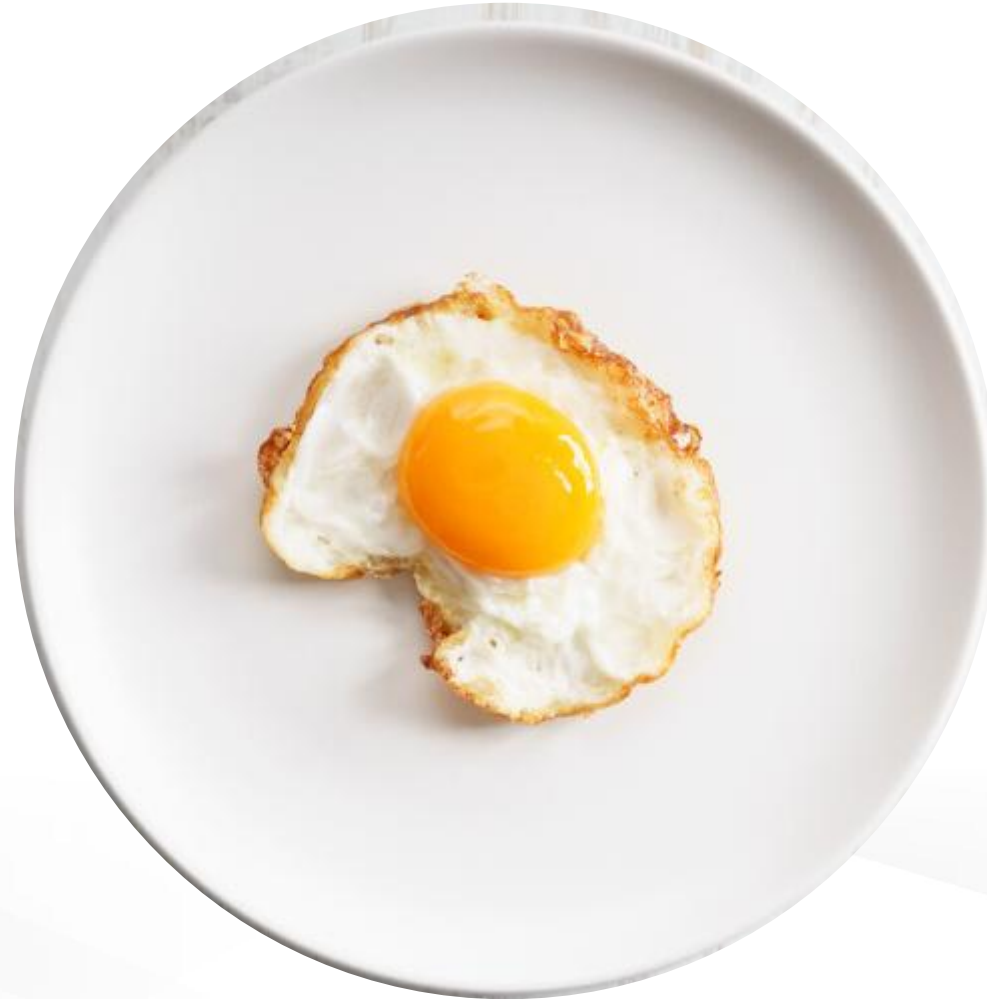


Fuente: <https://onx.la/d9023>



Algoritmo

Elaborar el algoritmo para la preparación de un huevo frito.



Fuente: <https://onx.la/47c3d>



Algoritmo

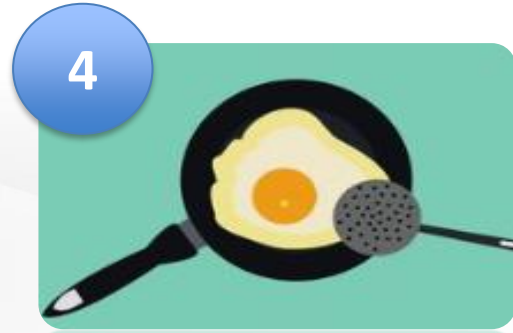
Solución del algoritmo para la preparación de un huevo frito

1. Alistar ingredientes.
2. Ponga una sartén con aceite a calentar.
3. Rompa un huevo y viértalo en el sartén.
4. Deje cocinar el huevo sin tocarlo por 3 minutos.
5. Emplatar.



Secuencia de pasos del algoritmo para preparar un huevo frito:

Paso a paso



Fuente: <https://onx.la/49fd6>



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Señor estudiante realice el siguiente ejercicio en clase: ¿Cuál es su algoritmo para llenar un vaso de agua?



Fuente: <https://onx.la/bcedc>



Actividad Independiente:

Realizar un algoritmo para ir a la universidad.



Fuente: <https://onx.la/119ed>

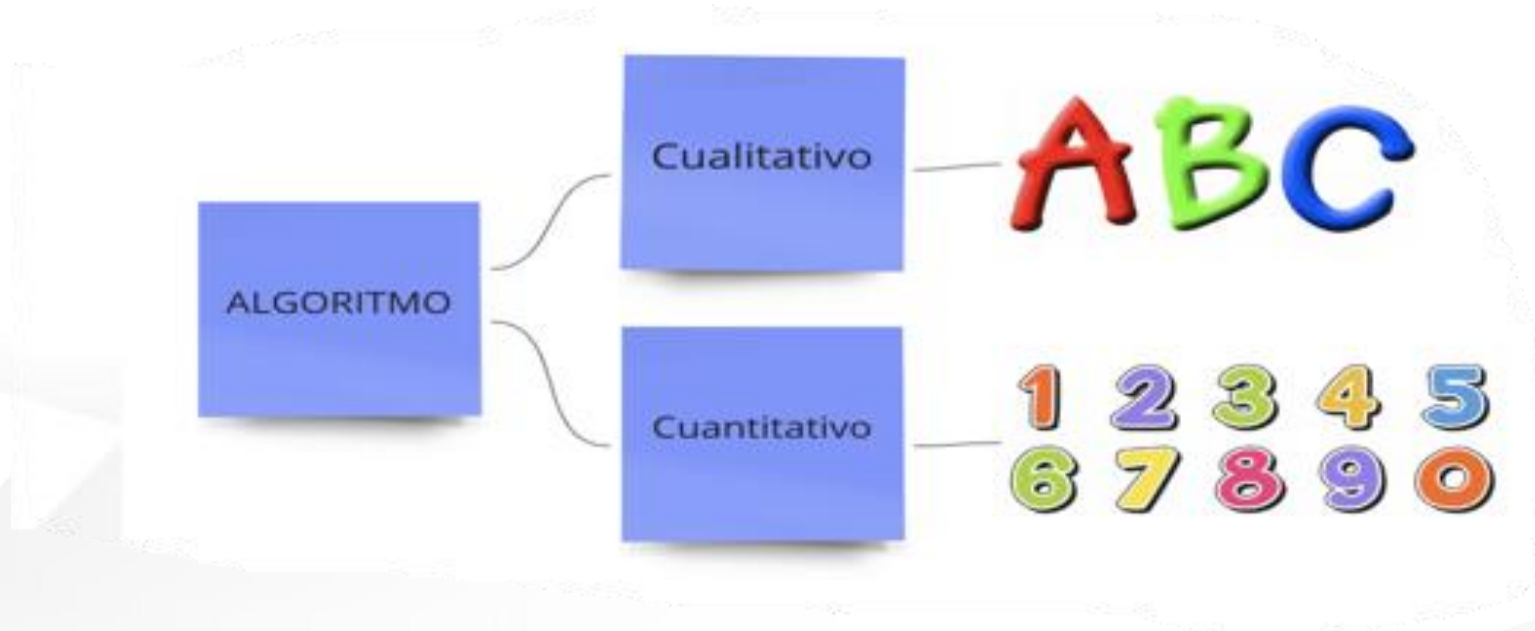


CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Tipos de Algoritmos

Cualitativos: Son aquellos en los que se describen los pasos utilizando palabras.

Cuantitativos: Son aquellos en los que se utilizan cálculos numéricos para definir los pasos del proceso.



Tipos de Algoritmos

Ejemplo _ Cuantitativos:

- Obtener el año en el que nacido un a persona.

Pasos: Variables?

edad=Edad de la persona

año=Año actual

añoNac=Año nacimiento de la persona

1. Preguntar edad
2. $\text{añoNac} = \text{año} - \text{edad}$
3. Decirle añoNac

1. Conocer la edad de la persona.
2. Saber el año actual.
3. Al restar el año actual con la edad de la persona procedemos a calcular su año de nacimiento.



Vigilada Mineducación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

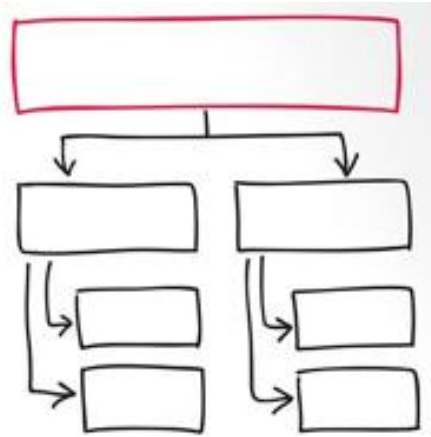
Existen diferentes formas de representar la estructura de un algoritmo, las más conocidas son:

- Graficas
- No graficas



Tipos de Lenguajes Algorítmicos

- **Gráficos:** Es la representación gráfica de las operaciones que realiza un algoritmo (diagrama de flujo).



Fuente: URL corta.at/RqKWD

- **No Gráficos:** Representa en forma descriptiva las operaciones que debe realizar un algoritmo (pseudocódigo).

```
1 Proceso NUMERO_MAYOR
2   Escribir 'Ingresa A: ';
3   Leer A;
4   Escribir 'Ingresa B: ';
5   Leer B;
6   Si A > B Entonces
7     Escribir 'El mayor es A';
8   Sino
9     Escribir 'El mayor es B';
10  FinSi
11 FinProceso
12
```



CORPORACIÓN UNIVERSITARIA DEL HUILA
Vigilada Mineducación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Diferencia entre un diagrama de flujo y un Pseudocódigo

Pseudocódigo:

INICIO

Base, Altura: ENTERO

ESCRIBA "Diga la Base: "

LEA Base

ESCRIBA "Diga la Altura"

LEA Altura

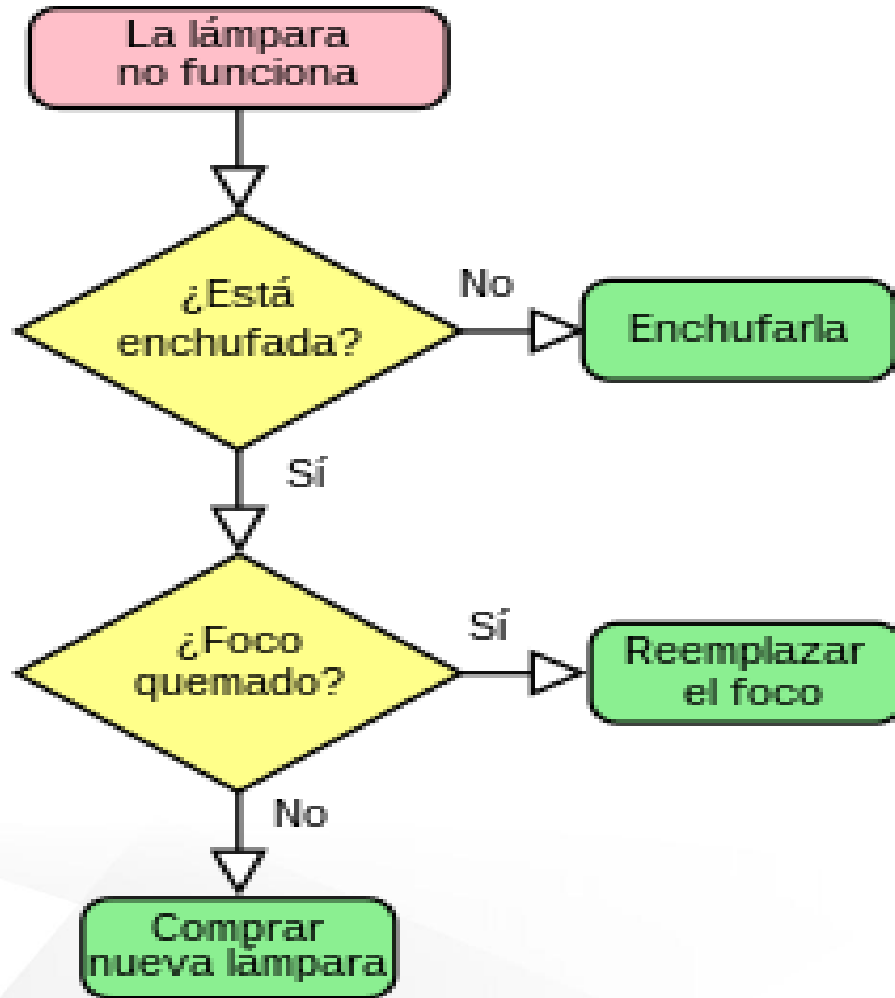
ESCRIBA "Area del Triangulo = ",
(BASE*ALTURA)/2

FIN

Diagrama de flujo:



Diagramas de flujo



Fuente: <https://onx.la/d9023>

Un diagrama de flujo es una representación gráfica de los pasos secuenciales que se deben seguir para realizar un proceso o resolver un problema. En programación, los diagramas de flujo se utilizan para ilustrar la lógica de un algoritmo, mostrando cada paso en forma de símbolos conectados por flechas que indican el flujo del proceso.

Diagramas de flujo

Los símbolos más comunes en los diagramas de flujo incluyen:

Óvalo: Representa el inicio y el fin del diagrama.

Rectángulo: Indica una instrucción o proceso específico, como una operación matemática o la asignación de valores.

Rombo: Utilizado para decisiones o condicionales, donde el flujo del diagrama se divide en función de una condición (por ejemplo, "sí" o "no").

Paralelogramo: Representa la entrada y salida de datos (como leer un valor o mostrar un resultado).

Flechas: Muestran la dirección del flujo del proceso, conectando los diferentes símbolos..



Diagramas de flujo

SIMBOLOS FUNDAMENTALES



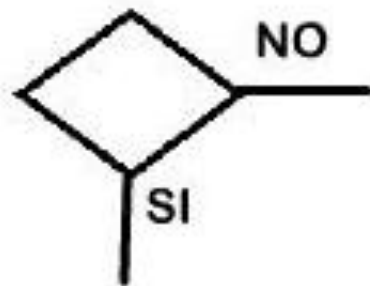
INICIO Y FIN DEL PROCESO



REALIZAR UN PROCESO (OPERACIÓN MATEMÁTICA POR EJEMPLO)



ENTRADA DE DATOS Y/O SALIDA DE DATOS



TOMAR UNA DECISIÓN (UNA PREGUNTA). LA RESPUESTA A LA PREGUNTA PUEDE SER SI O NO

Los diagramas de flujo son herramientas visuales poderosas para la planificación y comprensión de la lógica en programación.

Fuente: Elaboración propia



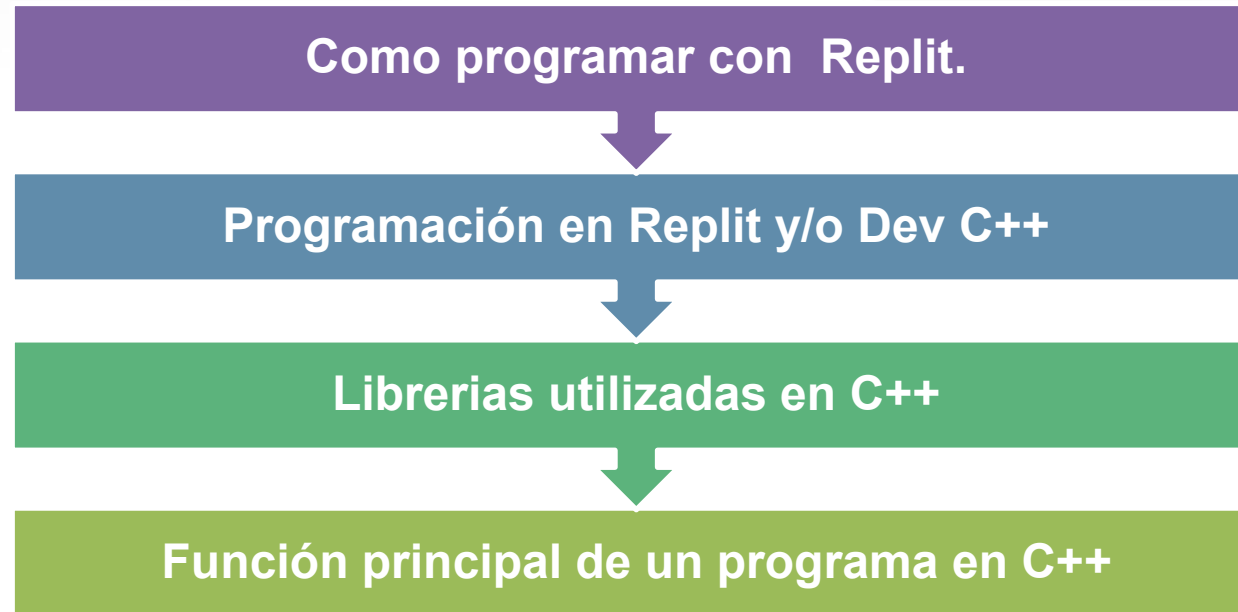
CORPORACIÓN UNIVERSITARIA DEL HUILA
Vigilada Mineducación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

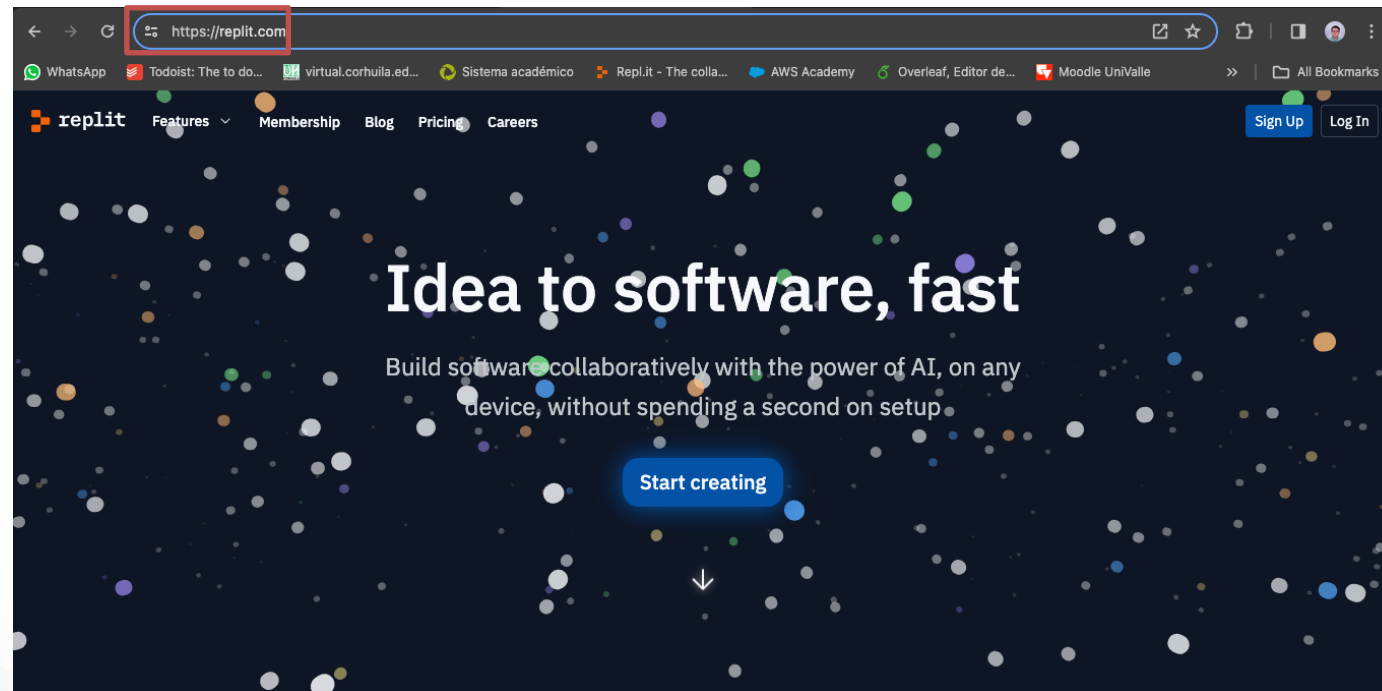
Unidad 1: Introducción al lenguaje de programación C++

Temas a desarrollar



Como utilizar Replit en linea

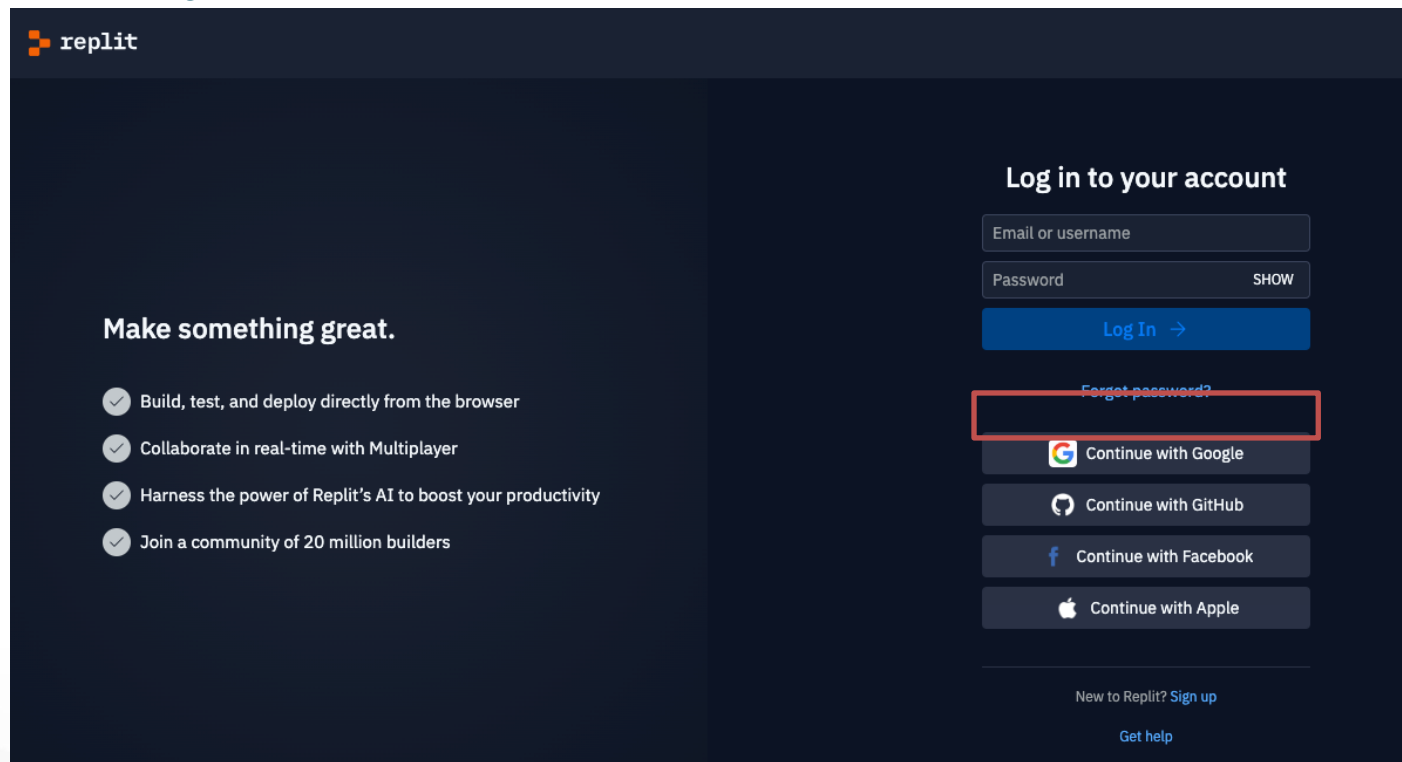
Ingresa a internet y copie el
URL →



Replit

Clic en el paso de marcado con color naranja →

 Continue with Google

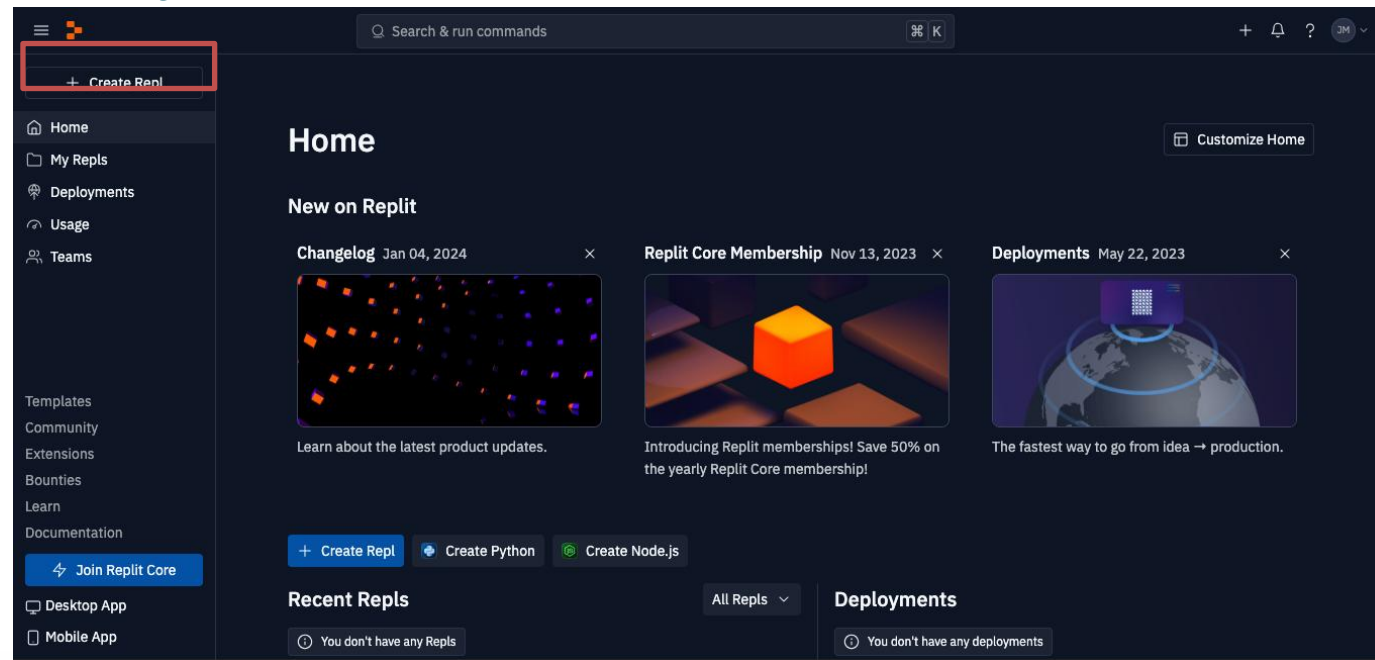


The screenshot shows the Replit login interface. On the left, there is a section titled "Make something great." with four bullet points: "Build, test, and deploy directly from the browser", "Collaborate in real-time with Multiplayer", "Harness the power of Replit's AI to boost your productivity", and "Join a community of 20 million builders". On the right, the "Log in to your account" section contains a form with "Email or username" and "Password" fields, a "Log In →" button, and a "Forgot password?" link highlighted with a red box. Below the form are social login options: "Continue with Google", "Continue with GitHub", "Continue with Facebook", and "Continue with Apple". At the bottom, there are links for "New to Replit? Sign up" and "Get help".



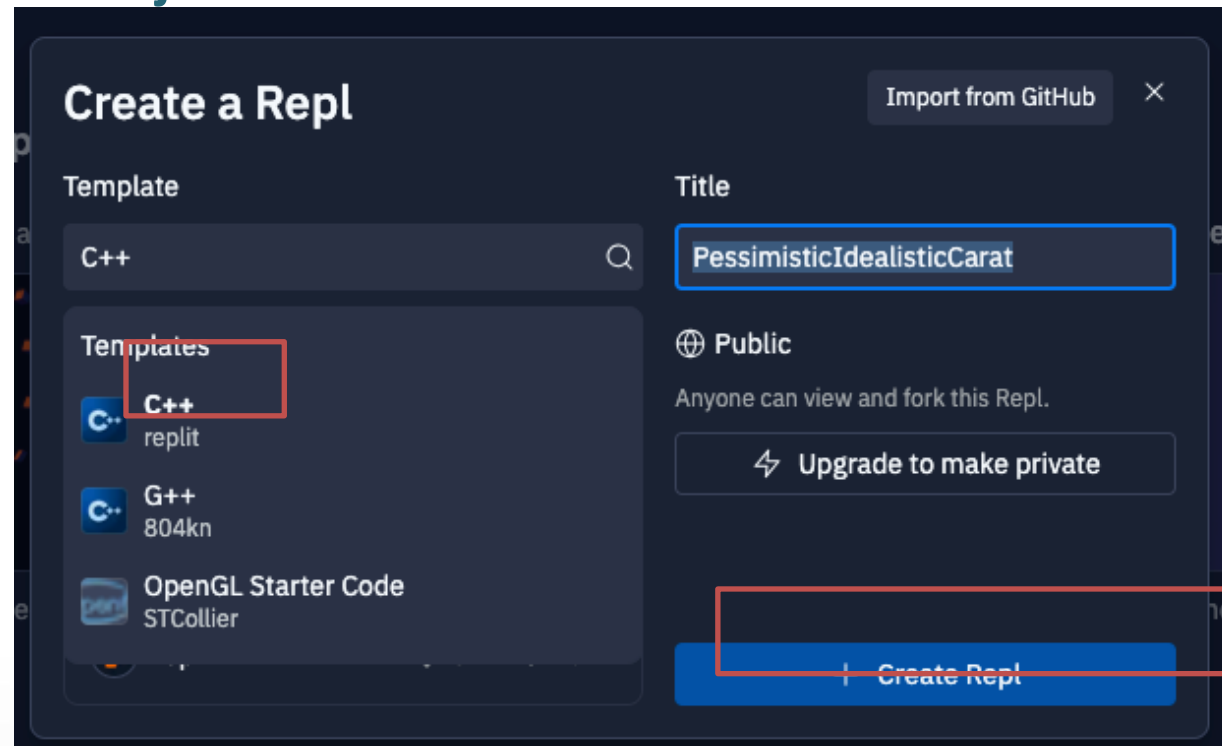
Replit

Clic en el paso de marcado con color naranja
naranja →



Replit

Clic en el paso de marcado con color naranja →



Replit

Inicie a programar teniendo en cuenta la explicación del docente o revise la siguiente Programa con Replit y/o Dev C++ presentación →

```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello World!\n";
5 }
```



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Programa con Replit y/o Dev C++



Programa C++

Las librerías en C++ son conjuntos de código preescrito que puedes usar en los programas para no tener que escribir todo desde cero. Estas librerías contienen funciones, clases y variables que permiten realizar tareas comunes, como manejar la entrada y salida de datos, realizar cálculos matemáticos o trabajar con cadenas de texto.



Programa C++

Por ejemplo, la librería `<iostream>` es una de las más usadas en C++. Te permite realizar operaciones de entrada y salida de datos. Como:

Imprimir en la pantalla usando `std::cout` o leer datos del teclado usando `std::cin`.



Programa C++

iostream: Contiene los algoritmos estándar, es quizá la más usada e importante.

La sintaxis es la siguiente:

#include <nombre de la librería> o alternativamente
#include "nombre de la librería".

```
1 #include <iostream>
```

C++

El programa en C++ este se compone de dos partes:

Librerías Función principal

Librería

#include <iostream>

```
[*] Sin Nombre1  
1 //Primer programa en C++  
2  
3 #include<iostream>  
.
```



C++

Fusión principal Int main() {

La función principal en C++ es llamada `main()` y es el punto de entrada de cualquier programa. Es decir, es donde comienza la ejecución de tu código. Todo programa en C++ debe tener una función `main()` porque es desde ahí que el sistema operativo sabe por dónde empezar a ejecutar el programa.

```
5 int main(){  
6  
7  
8  
9  
10 }
```



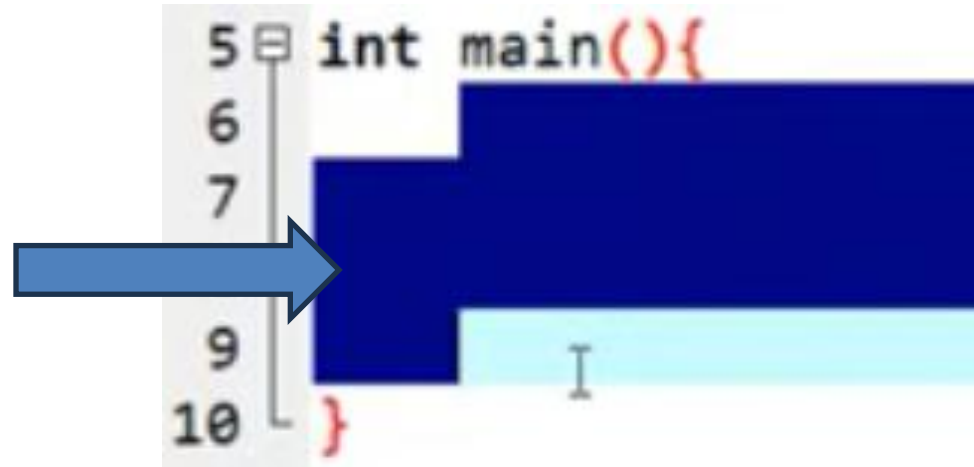
C++

Fusión principal

```
int main( ) {
```

En este espacio se
escribe el código a
utilizar

```
}
```



```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     string nombre;
5     cout << "Cual es su nombre\n";
6     cin >> nombre;
7     cout << " hola " << nombre << endl;
8 }
```



C++

```
1  #include <iostream>
2  using namespace std;
3  int main() {
4      string nombre;
5      cout << "Cual es su nombre\n";
6      cin >> nombre;
7      cout << " hola " << nombre <<endl;
8  }
```



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

C++

Using namespace std;

```
using namespace std;
```

Es para poder imprimir algo en pantalla.

Así se ponen los flujos de entrada y salidas super sencillos.

```
cout<<"Hola mundo :D";
```

Imprime en pantalla todo lo que se escribe dentro del `cout<<" "` ;



Línea 7

```
int main() {  
    cout<<"Hola mundo :D";  
  
    return 0;  
}
```

Nota: Tener cuidado al escribir el código es sensible a las mayúsculas y minúsculas.



C++

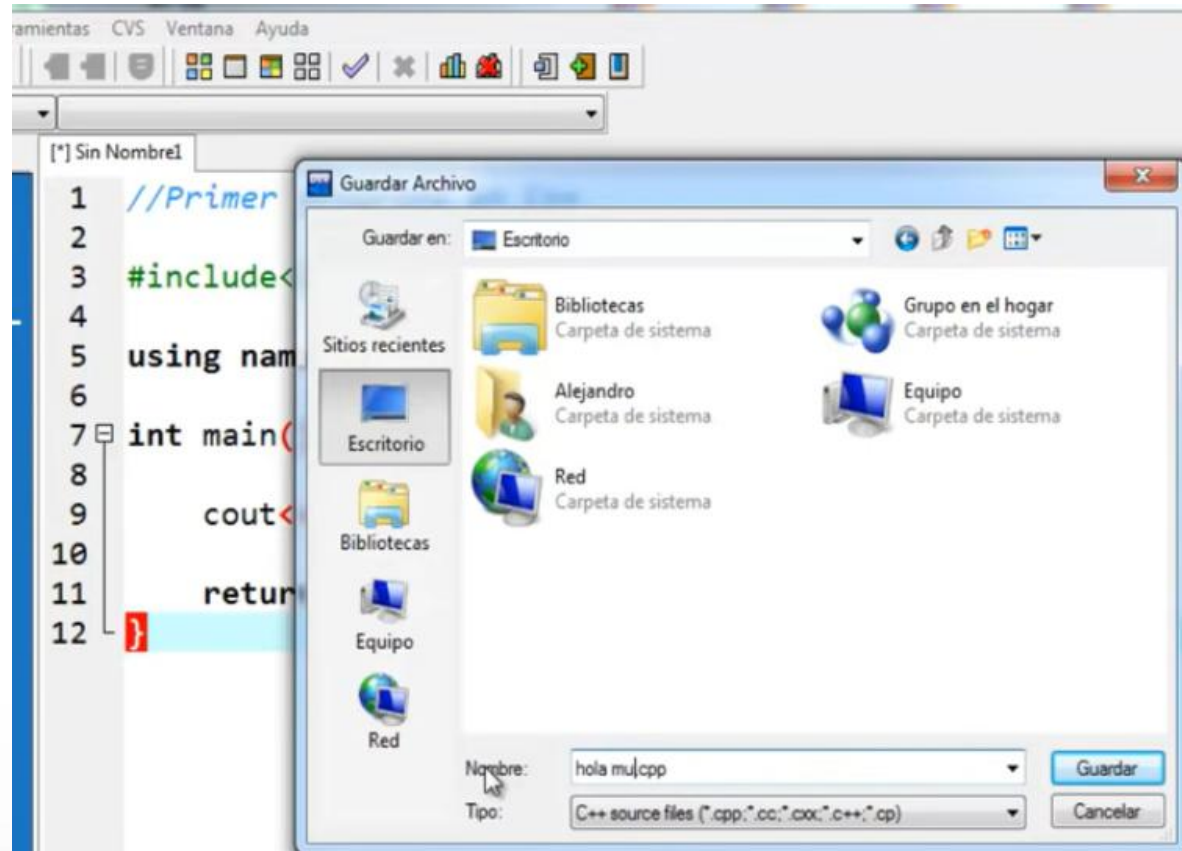
Se ha declarado una función de tipo entera, se le escribe el **Retur 0** antes de finalizar para que detecte que el programa ha finalizado correctamente.

```
1 //Primer programa en C++
2
3 #include<iostream>
4
5 int main(){
6
7
8
9     return 0;
10 }
```



C++

Se guarda con extensión `.cpp`



C++

Se ejecuta y se ha creado el primer programa en C++

```
hola mundo.cpp
1 //Primer programa en C++
2
3 #include<iostream>
4
5 using namespace std;
6
7 int main()
8 {
9     cout<<"Hola mundo :D";
10
11     return 0;
12 }
```

```
C:\Users\Alejandro\Desktop\hola mundo.exe
Hola mundo :D
-----
Process exited with return value 0
Press any key to continue . . .
```



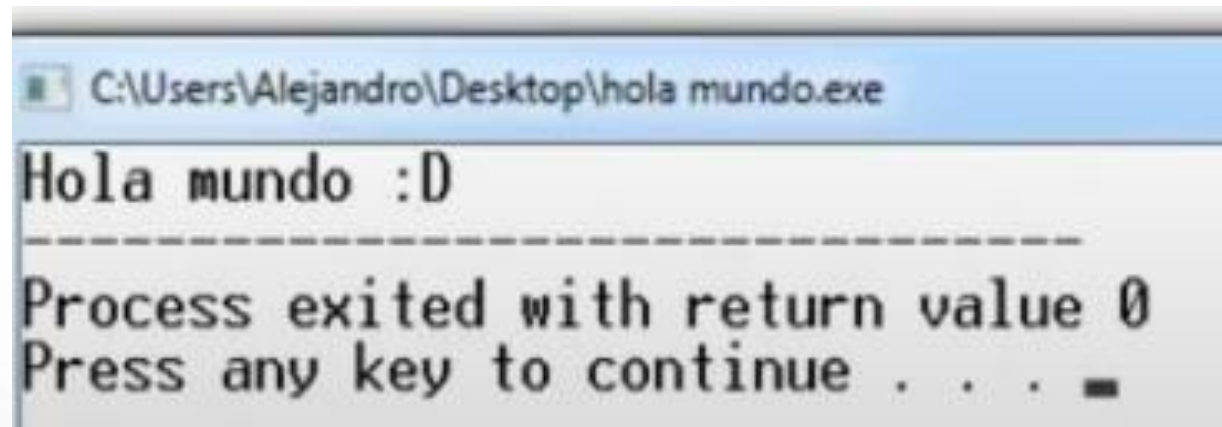
CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

C++

Al ejecuta se observa que la línea punteada está muy junta, para separar se incluye en:

```
Cout<<" Hola mundo :D<<endl;
```

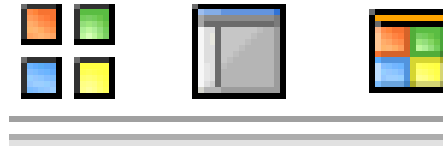
```
cout<<"Hola mundo :D"<<endl;
```



```
C:\Users\Alejandro\Desktop\hola mundo.exe
Hola mundo :D
-----
Process exited with return value 0
Press any key to continue . . .
```



C++



Al ejecutar y observe como queda:
Compilar y ejecutar F11

```
hola mundo.cpp ×
1 //MI primer proyecto
2 #include<iostream>
3 using namespace std;
4 int main() {
5     cout<<"Hola mundo :D";
6
7     return 0;
8 }
9
```



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

C++

Al ejecutar y observe como queda:

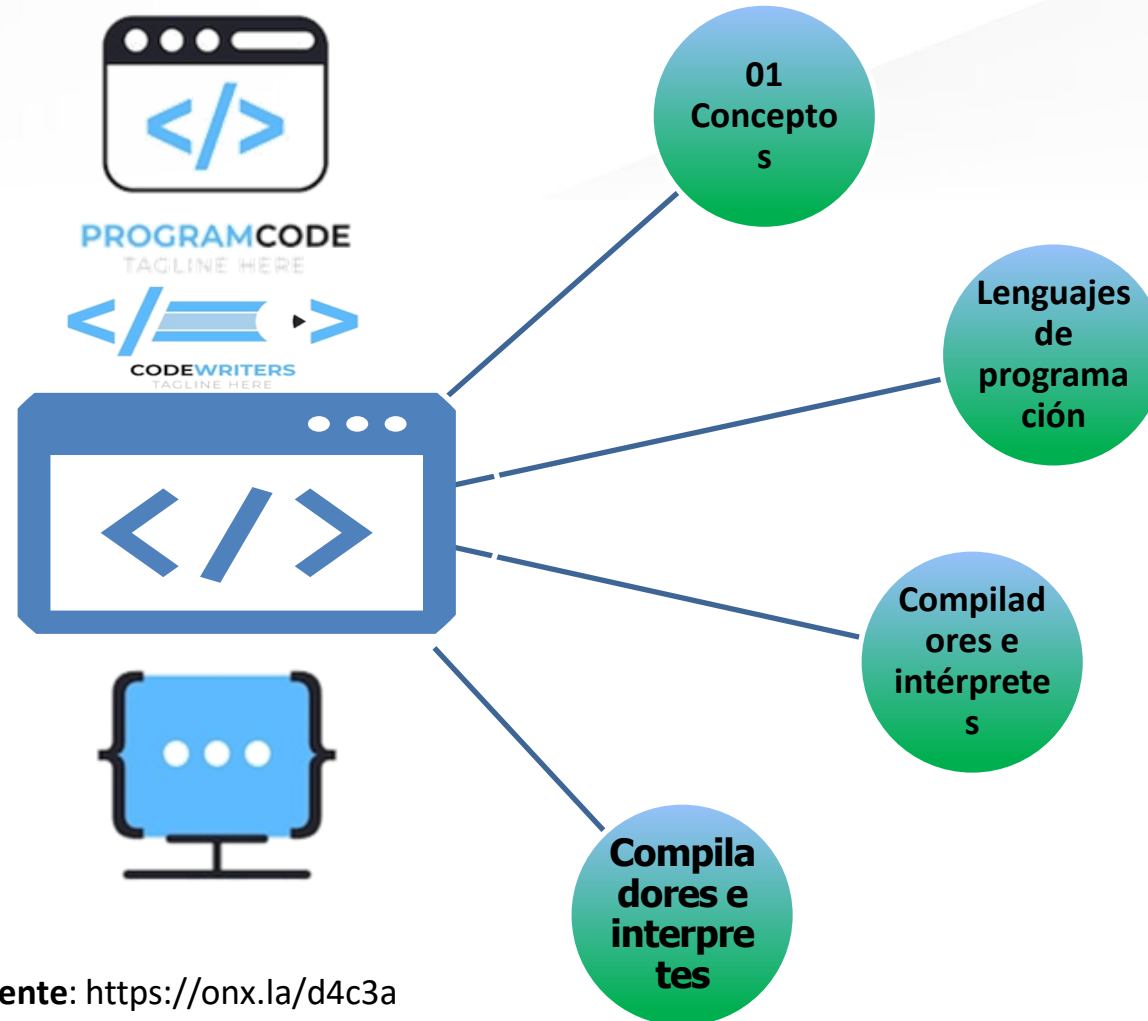
```
hola mundo.cpp ×
1 //MI primer proyecto
2 #include<iostream>
3 using namespace std;
4 int main() {
5     cout<<"Hola mundo :D";
6
7     return 0;
8 }
9
```



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Unidad 1.1: Introducción a Conceptos de Programación

Temas a desarrollar



Fuente: <https://onx.la/d4c3a>

Lenguajes de Programación

Es un conjunto de símbolos, caracteres y reglas (programas) que les permiten a las personas comunicarse con la computadora.

Los lenguajes de programación tienen un conjunto de instrucciones que nos permiten realizar operaciones de entrada/salida, calculo, manipulación de textos, lógica/comparación y almacenamiento/recuperación.



Fuente: <https://onx.la/14e8b>



Lenguajes de Programación

Proporciona las herramientas de programación que necesitan los desarrolladores de software.



Fuente: <https://onx.la/14e8b>



Lenguajes de Programación



Con la aparición de los lenguajes de programación suelen aparecer diferentes versiones de un mismo lenguaje.

Surgió la necesidad de estandarizarlos para que fueran más universales.

Las organizaciones que se encargan de regularizar los lenguajes son ANSI (Instituto de las Normas Americanas) e ISO (Organización de Normas Internacionales).

Lenguajes de Programación



Existen diversos lenguajes de programación, cada uno con sus propias características y aplicaciones. Algunos ejemplos comunes de lenguajes de programación son:



Un lenguaje versátil utilizado en desarrollo de software, sistemas operativos y programación de juegos.



Conocido por su sintaxis clara y legible, es utilizado en una amplia variedad de aplicaciones, desde desarrollo web hasta inteligencia artificial.

Fuente: <https://onx.la/01ef3>

Lenguajes de Programación



Utilizado en el desarrollo de aplicaciones empresariales, aplicaciones móviles (Android) y sistemas embebidos.

JavaScript



Conocido por su simplicidad y flexibilidad, a menudo se utiliza en desarrollo web, especialmente con el framework Ruby on Rails.



Un lenguaje de scripting ampliamente utilizado para el desarrollo web y la creación de aplicaciones dinámicas.

Fuente: <https://onx.la/01ef3>

Paradigmas de programación



Los paradigmas de programación son enfoques o estilos fundamentales para diseñar, estructurar y organizar el código en la programación informática. Cada paradigma tiene sus propias reglas, conceptos y prácticas que definen cómo se deben escribir y organizar los programas.

Los lenguajes de programación Python o JavaScript, son lenguajes multiparadigmas.



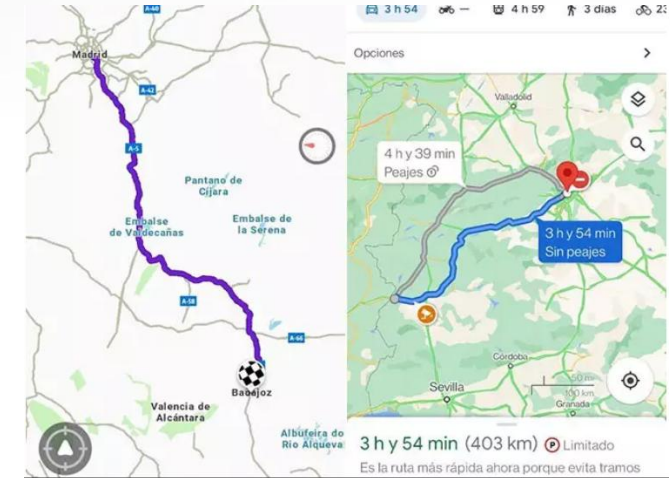
Fuente <https://shorturl.at/RqKWD>

Es una forma diferente de programar, de resolver un problema es equivalente a un mapa.

Es como el Weiss....busca la mejor opción para llegar a un sitio.

Son estilos de programación documentadas (no inventadas), de acuerdo a la necesidad que se quiera resolver.

- Formas de programar.
- Buenas prácticas.



Fuente <https://shorturl.at/RqKWD>

Paradigmas de programación



Algunos de los paradigmas de programación más comunes son:

- Paradigma imperativo
- Paradigma declarativo
- Paradigma estructurado o secuencial, entre otros

Paradigmas de programación



■ Paradigma imperativo/Procedimental:

Características: Se centra en describir paso a paso cómo lograr un resultado específico. Utiliza estructuras de control como bucles y condicionales.

Ejemplos de lenguajes: C, Pascal.

Se escribe paso por paso el programa ejemplo, hacer un arroz con pollo.

- Pechuga de pollo
- Zanahoria en cuadros
- 2 tallos de apio
- 1 cebolla cabezona blanca en cuadritos
- 2 dientes de ajo
- 3 ramitas de cilantro
- 1 cucharadita de pimienta
- 1 cucharadita de sal

Paradigmas de programación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, Investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

- **Paradigma declarativo**

Características: Describe "qué" se debe lograr en lugar de "cómo" hacerlo. Incluye programación lógica y programación funcional.

Ejemplos de lenguajes: Haskell, SQL.

Le dice al Sistema lo que debe hacer en SQL le dice traer todas las ventas del mes de abril.

Ejemplo traer datos que comiencen con un numero

Paradigmas de programación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, Investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

- **Programación Lógica:**

Características: Se basa en la lógica formal y la inferencia. Los programas expresan declarativamente relaciones lógicas y reglas.

Ejemplos de lenguajes: Prolog.

Paradigmas de programación

▪ Paradigma estructurado o secuencial

Es el primer paradigma que aprendemos todos a programar, es el estructurado secuencial o de arriba hacia abajo.

Donde se declaramos variables:

ejemplo Nombre = Pepito

Declaramos función cuadrado

```
function cuadrado(n){  
    return n * n  
}  
  
Let num = cuadrado(2);
```

Fuente: Elaboración propia

```
variables.js x  
1  var cadena="hola esto es un String"  
2  var entero=5  
3  var decimal=4.5  
4  var caracter='c'  
5  var booleana=true
```

Fuente: Elaboración propia

Paradigmas de programación



▪ Paradigma Programación orientada a objetos

Es el más usado en el mundo, este separa al sistema en entidades llamadas Objetos.

Características: Organiza el código en torno a "objetos" que encapsulan datos y comportamientos. Los conceptos clave incluyen clases, objetos, encapsulamiento, herencia y polimorfismo

Objeto tiene características y funciones

Funciones: (multiplicar dos números)

Funciones: Las cosas que el usuario puede hacer en una red social, registrar el usuario y la contraseña.

Ejemplos de lenguajes: Java, C++, Python

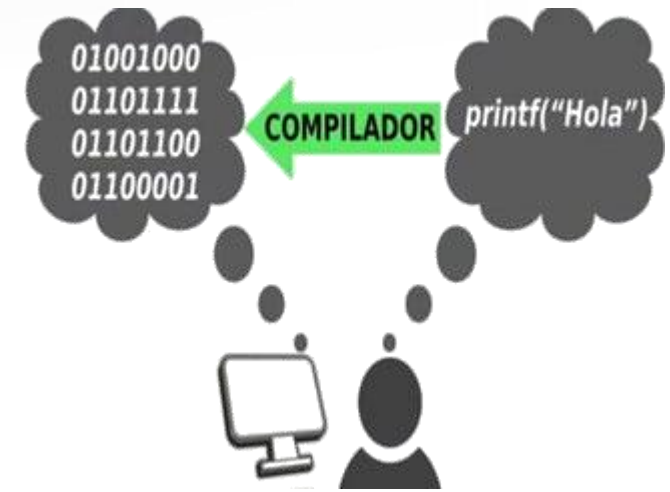
Compiladores e interpretes



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

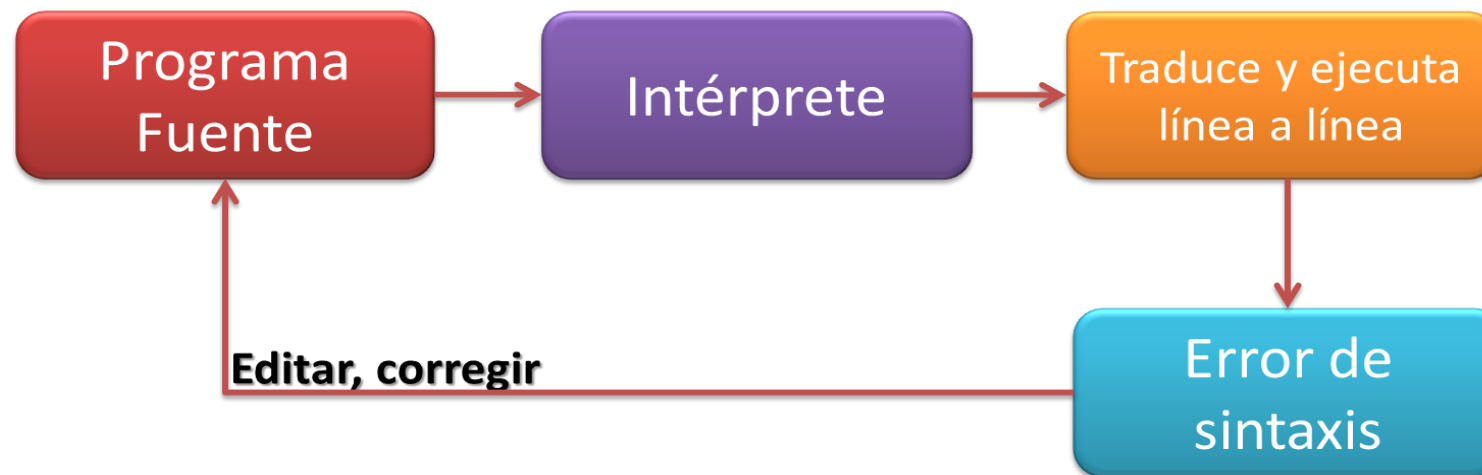
Un compilador es un programa que traduce todo el código fuente de un programa escrito en un lenguaje de alto nivel (como C o Java) a lenguaje de máquina antes de su ejecución. Esto significa que el programa entero se compila en un archivo ejecutable que luego puede ser ejecutado por la computadora sin necesidad de recompilarse.

En resumen, los compiladores suelen ser más rápidos en la ejecución final, pero los intérpretes permiten una revisión más inmediata de los resultados y son comunes en lenguajes como Python.



Fuente: <https://onx.la/417d7>

Por otro lado, un intérprete traduce el código fuente línea por línea y lo ejecuta directamente, sin generar un archivo ejecutable. Esto permite ver los resultados de inmediato, pero puede ser más lento en ejecución que un programa compilado, ya que cada línea se traduce y ejecuta en tiempo real.



Fuente: Elaboración propia

Unidad 1.2 : Introducción a Conceptos de Programación



Temas para desarrollar

Análisis de un problema.



Pasos para la representación de un algoritmo (entradas, procesos, salidas).



Resolución de problemas en un lenguaje de programación.



Conceptos básicos (tipos de datos, variables, constantes, expresiones, operadores).

Análisis de un problema

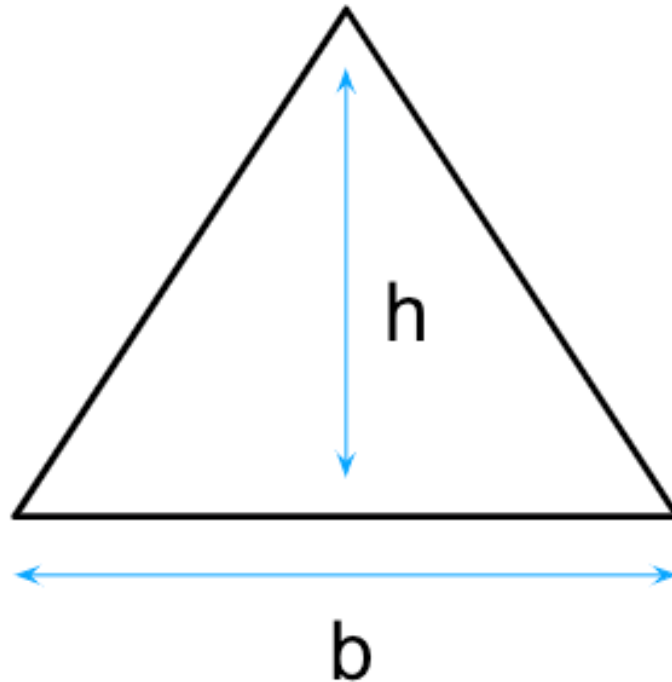
En programación es el primer paso en el proceso de desarrollo de un programa. Consiste en entender completamente el problema que se desea resolver, identificando los datos de entrada, los requisitos y las restricciones. En esta etapa, el programador descompone el problema en partes más pequeñas y define los pasos necesarios para llegar a una solución. El análisis incluye:

- Definir claramente los objetivos del programa.
- Determinar las entradas y salidas necesarias.
- Identificar cualquier restricción o condición específica.
- Dividir el problema en tareas más pequeñas y manejables.



Análisis de un problema

Problema: Desarrollar un programa que permita calcular el área de un triángulo dados su base y su altura.



$$\text{área}=(b \cdot h) / 2$$

Análisis de un problema

Problema: Desarrollar un programa que permita calcular el área de un triángulo dados su base y su altura.

1. Analizar el problema

Consiste en identificar los datos de entrada y de salida.



Análisis de un problema

Problema: Desarrollar un **programa** que permita calcular el área de un triángulo dados su base y su altura.

1. Analizar el problema

Consiste en identificar los datos de entrada y de salida.



Análisis de un problema

Problema: Desarrollar un **programa** que permita calcular el área de un triángulo dados su base y su altura.

1. Analizar el problema

Consiste en identificar los datos de entrada y de salida.

Las entradas y salidas se conocen como **variables** y pueden ser de tres tipos.

- **Real:** 1.2, 13.7, 3.5
- **Entero:** 1, 500, 1000
- **Texto:** “Maria”, “calle 23 a No. 34-18”



Análisis de un problema

Problema: Desarrollar un **programa** que permita calcular el área de un triángulo dados su base y su altura.

1. Analizar el problema

Consiste en identificar los datos de entrada y de salida.



Análisis de un problema

Problema: Desarrollar un **programa** que permita calcular el área de un triángulo dados su base y su altura.

1. Analizar el problema

Consiste en identificar los datos de entrada y de salida.

Digite la base: 3

Digite la altura: 5

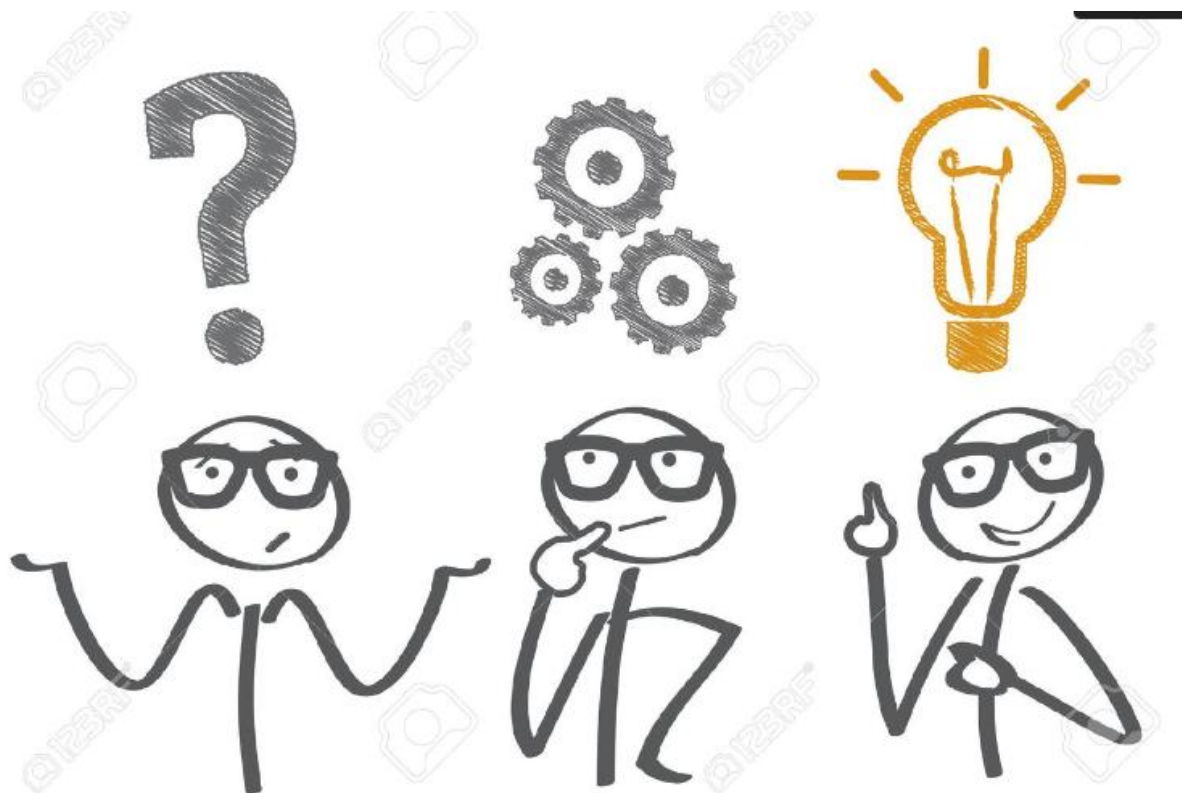


El área del triángulo es: 7.5



Lo que sí es importante notar, es que de un algoritmo sale el código en un **lenguaje de programación.**

Cuando se elabora un algoritmo se debe tener en cuenta lo siguiente:



Fuente: Elaboración propia

- Tener claro cuál es el problema que va a solucionar.
- Establecer un objetivo que permita medir la solución del problema.

Pasos para representar un algoritmo

Pseudocódigo

Inicio

Fin



Pasos para representar un algoritmo

Pseudocódigo

Inicio

base, altura : entero

area : real

Fin



Pasos para representar un algoritmo

Pseudocódigo

Inicio

base, altura : entero

area : real

preguntar(base)

preguntar (altura)

Fin



Pasos para representar un algoritmo

Pseudocódigo

Inicio

base, altura : entero

area : real

preguntar(base)

preguntar (altura)

$area = (base * altura) / 2$

Fin



Pasos para representar un algoritmo

Pseudocódigo

Variables



Inicio

base, altura : entero
area : real

Entradas



preguntar(base)
preguntar (altura)

Procesos



$area = (base * altura) / 2$

Salidas



mostrar(area)

Fin

Pasos del algoritmo



Pasos para representar un algoritmo

Pseudocódigo

Inicio

base, altura : entero

area : real

preguntar(base)
preguntar (altura)

Digite la base:
Digite la altura:

$area = (base * altura) / 2$

El área del
triángulo es:

mostrar(area)

Fin



Pasos para representar un algoritmo

Pseudocódigo

Inicio

base, altura : entero

area : real

preguntar(base)

preguntar (altura)

$area = (base * altura) / 2$

mostrar(area)

Fin

Prueba de Escritorio

base	altura	area
1	2	1
4	5	10
2	3	3



Pasos para representar un algoritmo

El pseudocódigo puede utilizarse como lenguaje común para que un equipo de desarrolladores se comuniquen y colabore en un proyecto, independientemente de sus conocimientos de programación.

Variables
→

Entradas
→

Procesos
→

Salidas
→

Pseudocódigo

Inicio

base, altura : entero
area : real

preguntar(base)
preguntar (altura)

area = (base * altura) / 2

mostrar(area)
Fin

```
main.cpp x +
main.cpp > f main
1 #include <iostream>
2 using namespace std;
3 // Inicio
4 int main() {
5     // Variables
6     int base;
7     int altura;
8     float area;
9     // Entradas
10    cout << "Digite la base: ";
11    cin >> base;
12    cout << "Digite la altura: ";
13    cin >> altura;
14    // Proceso
15    area = (base * altura) / 2;
16    // Salidas
17    cout << "El area del triangulo es: " << area << endl;
18 // Fin
19 }
```



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Resolución de problemas en un lenguaje de programación C++

Código→

```
main.cpp x +
Ejecutar → Run
main.cpp
1 #include <iostream>
2 using namespace std;
3 // Inicio
4 int main() {
5     // Variables
6     int base;
7     int altura;
8     float area;
9     // Entradas
10    cout << "Digite la base: ";
11    cin >> base;
12    cout << "Digite la altura: ";
13    cin >> altura;
14    // Proceso
15    area = (base * altura) / 2;
16    // Salidas
17    cout << "El area del triangulo es: " << area << endl;
18    // Fin
19 }
```

Console Shell

```
Run
Digite la base: 4
Digite la altura: 5
El area del triangulo es: 10
```

Resultado por pantalla→



CORPORACIÓN UNIVERSITARIA DEL HUILA
Vigilada Mineducación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA

"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Ejercicios

Problema 1: Desarrollar un **programa** que permita convertir grados Celsius a Fahrenheit.

$$F = (9/5)*C + 32$$

Digite los grados en Celsius: 20



El valor en Fahrenheit es: 68



CORPORACIÓN UNIVERSITARIA DEL HUILA
Vigilada Mineducación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Ejercicios

Problema2: Desarrollar un programa para calcular la nota definitiva de un estudiante en una asignatura. Los datos son la nota de la parte teórica de la asignatura que representa el 70% de la definitiva y la nota de la práctica que representa el 30%. El programa debe mostrar un mensaje que indique si el estudiante está aprobado o reprobado



Análisis →

Entrada

NT: nota de teoría. Tipo: Real.

NP: nota de práctica. Tipo: Real.

Proceso

Calcular la nota definitiva

$$N_{def} = 0.7 \times NT + 0.3 \times NP$$

Determinar si el estudiante está aprobado o reprobado

$N_{def} \geq 9.5$ Estudiante aprobado

$N_{def} < 9.5$ Estudiante reprobado

Salida

N_{def} : nota definitiva. Tipo: Real.

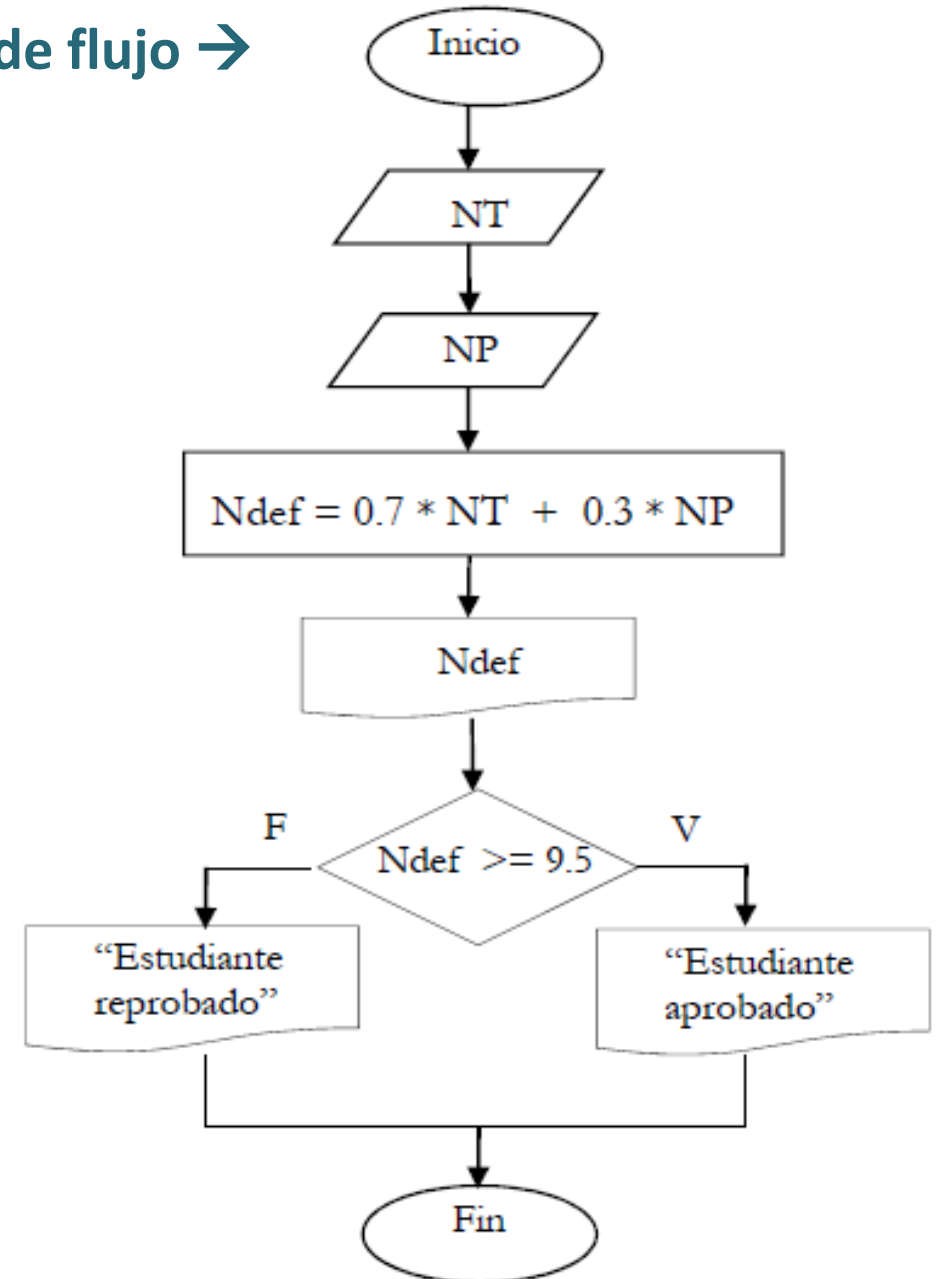
Mensaje que indica si el estudiante está aprobado o reprobado en la asignatura.



Algoritmo

0. Inicio
1. Leer nota de teoría (NT)
2. Leer nota de práctica (NP)
3. $N_{def} = 0.7 * NT + 0.3 * NP$
4. Mostrar nota definitiva (Ndef)
5. Si $N_{def} \geq 9.5$ entonces
 - 5.1 Mostrar “El estudiante está aprobado”
de lo contrario
 - 5.2 Mostrar “El estudiante está reprobado”
- Fin de si
6. Fin

Diagrama de flujo →



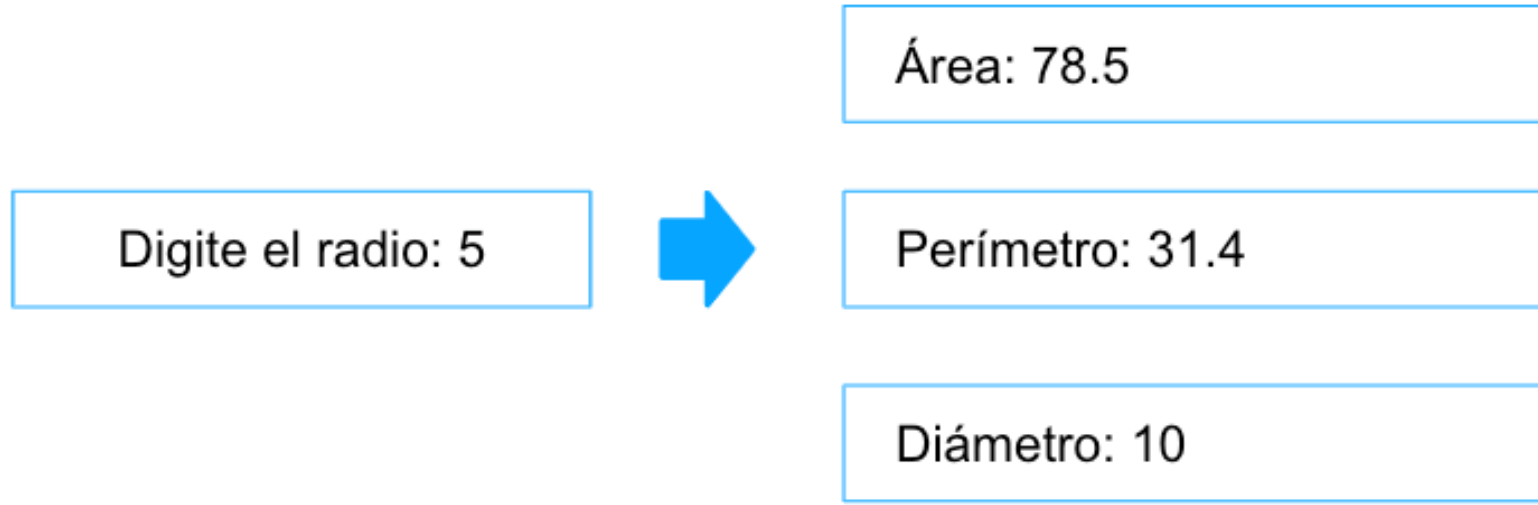
PREGUNTAS



Ejercicios

Actividad independiente: Realizar el siguiente ejercicio implementando lo explicado en clase: Análisis del problema identificando cada uno de los pasos.

Problema 2: Desarrollar un **programa** que permita calcular el área, el perímetro y el diámetro de un círculo de radio r .



Tipos de datos

Problema 1: Desarrollar un **programa** que permita convertir grados Celsius a Fahrenheit.

$$F = (9/5)*C + 32$$



Palabras reservadas

Cada lenguaje de programación tiene sus propias **palabras reservadas** como parte de su sintaxis (forma de escribir).

```
main.cpp x +
main.cpp
1 #include <iostream>
2 using namespace std;
3 // Inicio
4 int main() {
5     // Variables
6     int base;
7     int altura;
8     float area;
9     // Entradas
10    cout << "Digite la base: ";
11    cin >> base;
12    cout << "Digite la altura: ";
13    cin >> altura;
14    // Proceso
15    area = (base * altura) / 2;
16    // Salidas
17    cout << "El area del triangulo es: " << area << endl;
18    // Fin
19 }
```



Tipos de datos

Tipo de dato	Ejemplo	Palabra reservada en C++
Entero	2, 98, 130	int
Real o decimal	3.5, 9.87, 10.20	float
Cadena	“Hola”, “Calle 23 # 15.18”	string

Fuente: Elaboración propia



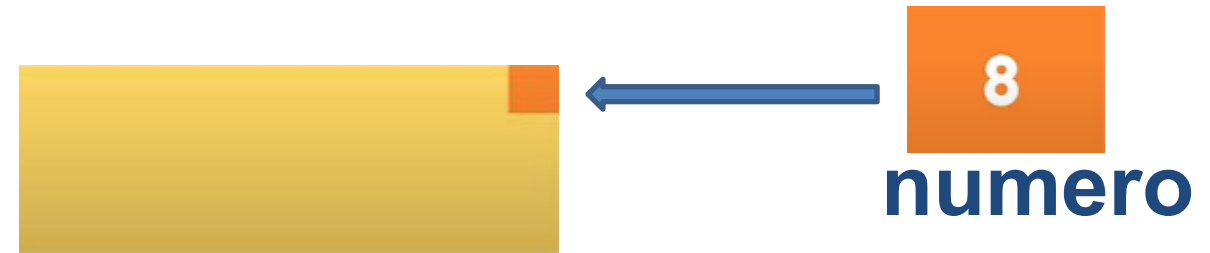
CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Variables

Es un espacio de la memoria reservado para almacenar un valor, correspondiente a un tipo de dato (numérico, decimal, cadena).

Es modificable a lo largo de un programa.

Las variables tienen un identificador o nombre.



Fuente: Elaboración propia

Variables

Una variable tiene dos fases :

Primero la declaramos: En C++ tipo de dato y nombre.

Ejemplo: int edad;

Luego la inicializamos: Significa darle un valor.

Ejemplo: int edad = 0;

```
#include <iostream>
using namespace std;

// Inicio
int main(){
    // Declarar la variable numero como entero e inicializar con 5
    int numero = 5;
    cout << numero << endl;
    // Declarar la variable decimal como float e inicializar con 3.5
    float decimal = 3.5;
    cout << decimal << endl;
    // Declarar la variable cadena como string e inicializar con hola
    string cadena = "Hola";
    cout << cadena << endl;
}
```



Variables

¿Cómo nombrar una variable?

- Sin espacio entre caracteres
- No puede iniciar con números
- No puede incluir caracteres especiales.
- No puede llevar el mismo nombre de una palabra reservada



Datos clientes

4toCorte

Número

int



Variables



Datos clientes

4toCorte

Número

int



datosClientes

cuartoCorte

numero

entero



Variables

Estilos para nombrar un variable

PascalCase

La primera letra de cada palabra va en mayúscula.

NumeroPersonas

camelCase

La primera letra de cada palabra va en mayúsculas, excepto la primera.

numeroPersonas.



Ejercicio 1

Realice un programa en C++ que pregunte un nombre y lo muestre en pantalla.

Análisis →

Digite su nombre: Cristian

Hola Cristian



Ejercicio 1

una variable
permite
guardar un
valor

```
main.cpp x
20 #include <iostream>
21 using namespace std;
22
23 // Inicio
24 int main() {
25     // Datos de entradas
26     //nombre: texto
27     string nombre;
28
29     // Preguntar nombre
30     cerr << "Digite su nombre: ";
31     cin >> nombre;
32
33     // Datos de salidas
34     //Mostrar nombre
35     cout << "Hola " << nombre;
36 // Fin
37 }
```

← Inicio

nombre : texto
preguntar (nombre)
mostrar (nombre)
Fin

nombre	Salida
jose	Hola jose
santiago	Hola santiago
isabella	Hola isabella



Ejercicio 1

permite
solicitar un
dato al
usuario

```
main.cpp x
20 #include <iostream>
21 using namespace std;
22
23 // Inicio
24 int main() {
25     // Datos de entradas
26     //nombre: texto
27     string nombre;
28
29     // Preguntar nombre
30     cerr << "Digite su nombre: ";
31     cin >> nombre;
32
33     // Datos de salidas
34     //Mostrar nombre
35     cout << "Hola " << nombre;
36 // Fin
37 }
```

Inicio

nombre : texto

preguntar (nombre)

mostrar (nombre)

Fin



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Ejercicio 1

```
main.cpp x
20 #include <iostream>
21 using namespace std;
22
23 // Inicio
24 int main() {
25     // Datos de entradas
26     //nombre: texto
27     string nombre;
28
29     // Preguntar nombre
30     cerr << "Digite su nombre: ";
31     cin >> nombre;
32
33     // Datos de salidas
34     //Mostrar nombre
35     cout << "Hola " << nombre;
36     // Fin
37 }
```

permite
mostrar un
mensaje al
usuario

Inicio

nombre : texto

preguntar (nombre)

mostrar (nombre)

Fin



CORPORACIÓN UNIVERSITARIA DEL HUILA
Vigilada Mineducación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación
y extensión de programas de pregrado, aplicando todos los
requisitos de las normas ISO implementadas en sus sedes
Neiva y Pitalito"

Ejercicio 1

The screenshot shows a C++ IDE with a file named `main.cpp`. The code is as follows:

```
20 #include <iostream>
21 using namespace std;
22
23 // Inicio
24 int main() {
25     // Datos de entradas
26     //nombre: texto
27     string nombre;
28
29     // Preguntar nombre
30     cerr << "Digite su nombre: ";
31     cin >> nombre;
32
33     // Datos de salidas
34     //Mostrar nombre
35     cout << "Hola " << nombre;
36 // Fin
37 }
```

The console output shows the execution of the program:

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Digite su nombre: Cristian
Hola Cristian>
```

Annotations with red arrows:

- A red arrow points from the text "permite ejecutar la aplicación" to the green "Run" button.
- Two red arrows point from the text "imprime un mensaje y el valor de la variable" to the `cout << "Hola " << nombre;` line in the code.
- A red arrow points from the text "La variable almacena el valor Cristian" to the `cin >> nombre;` line in the code.

imprime un
mensaje y el valor
de la variable

La variable
almacena el valor
Cristian



CORPORACIÓN UNIVERSITARIA DEL HUILA
Vigilada Mineducación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación
y extensión de programas de pregrado, aplicando todos los
requisitos de las normas ISO implementadas en sus sedes
Neiva y Pitalito"

Constantes

Una constante es muy similar a una variable, la única diferencia es que NO es modificable, es decir, no cambia durante la ejecución del algoritmo o programa.

Ejemplo:

$\pi = 3.1416;$

gravedad = 9.81;



Expresiones

Las expresiones son combinaciones de constantes, variables, símbolos de operación, paréntesis y nombres de funciones especiales.

Ejemplo:

$$a + (b + 3) / c$$



Ejercicio 2

Realice un programa en C++ que permita calcular el IVA (19%) de una venta.

Análisis→

Digite el valor de la venta:
120000



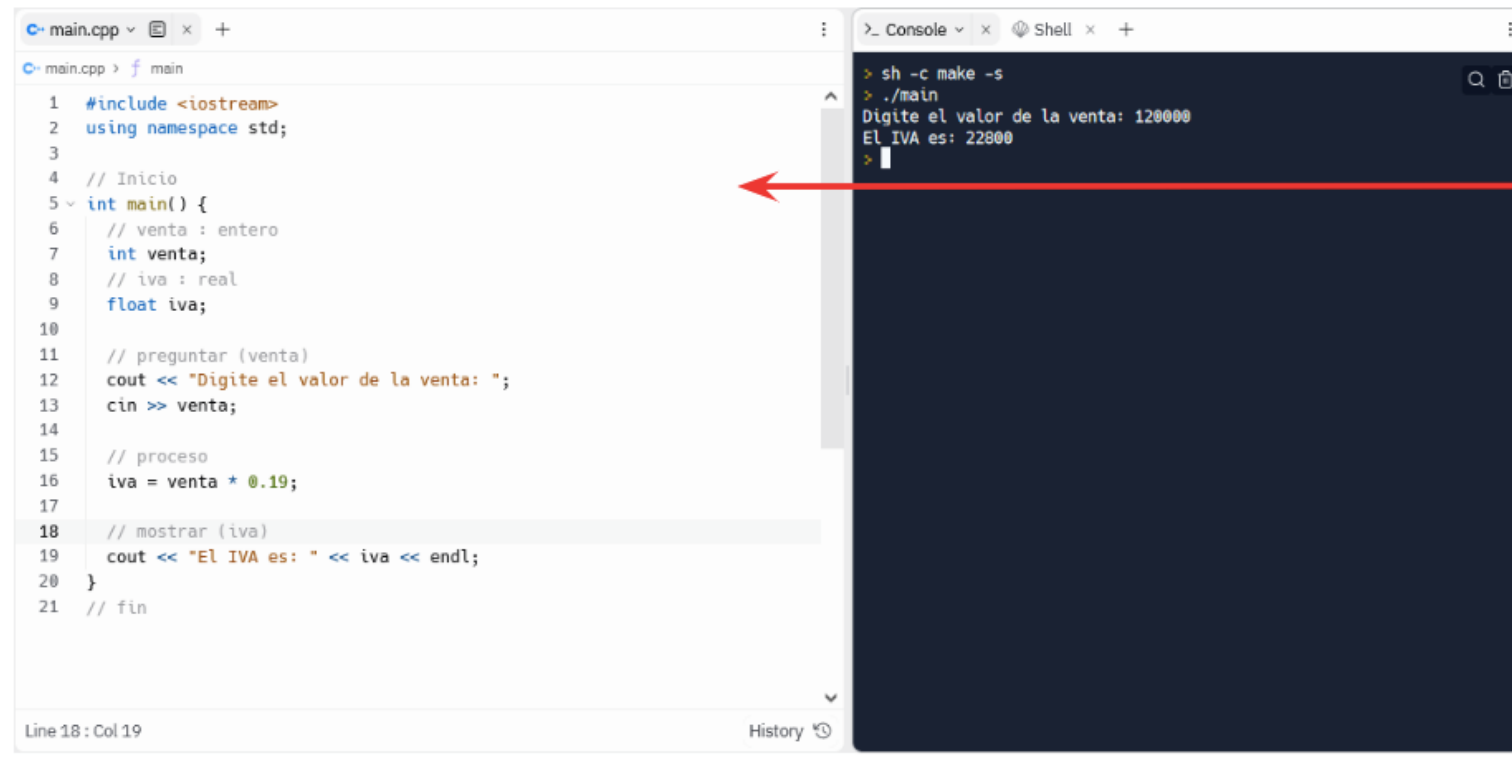
El IVA es: 22800



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Ejercicio 2

Realice un programa en C++ que permita calcular el IVA (19%) de una venta.



The image shows a C++ IDE with two windows. The left window displays the source code for a program that calculates the 19% IVA on a sale. The right window shows the terminal output where the program is executed, demonstrating the input and output.

```
main.cpp | Console | Shell
```

```
main.cpp > f main
```

```
1 #include <iostream>
2 using namespace std;
3
4 // Inicio
5 int main() {
6     // venta : entero
7     int venta;
8     // iva : real
9     float iva;
10
11     // preguntar (venta)
12     cout << "Digite el valor de la venta: ";
13     cin >> venta;
14
15     // proceso
16     iva = venta * 0.19;
17
18     // mostrar (iva)
19     cout << "El IVA es: " << iva << endl;
20 }
21 // fin
```

```
> sh -c make -s
> ./main
Digite el valor de la venta: 120000
El IVA es: 22800
>
```

Inicio

venta : entero

iva : real

preguntar (venta)

iva = venta * 0.19

mostrar (iva)

Fin

Ejercicio 3

Realice un programa en C++ que solicite el radio de un círculo y calcule el diámetro y área

Análisis →

Digite el radio: 2.3



Diámetro: 4.6
Área: 16.6106



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Ejercicio 3

```
main.cpp x
25 #include <iostream>
26 // Se utiliza esta línea para dar espacio al std, donde
    // se encuentran las librerías estándar de C++ (Ejemplo:
    // cout, que sirve para impresiones en pantalla)
27 using namespace std;
28
29 // Inicio
30 int main() {
31     // Datos de entradas
32     // radio: real
33     float radio;
34     //diámetro, área: real
35     float diametro, area;
36
37
38     // Preguntar radio
39     cerr << "Digite el radio: ";
40     cin >> radio;
41
42     // procesos
43     diametro = 2 * radio;
44     area= 3.14 * radio * radio;
45
46
47     // Datos de salidas
48     // mostrar diámetro, área
49     cout << "Diámetro: " << diametro << endl;
50     cout << "Área: " << area;
51 // Fin
52 }
```

Inicio

radio : real

diámetro, área : real

preguntar (radio)

diámetro = 2*radio

área = 3.14*radio*radio

mostrar (diámetro)

mostrar (área)

Fin



Ejercicio 3

```
main.cpp x Console Shell
25 #include <iostream>
26 // Se utiliza esta línea para dar espacio al std, donde
    // se encuentran las librerías estándar de C++ (Ejemplo:
    // cout, que sirve para impresiones en pantalla)
27 using namespace std;
28
29 // Inicio
30 int main() {
31     // Datos de entradas
32     // radio: real
33     float radio;
34     //diametro,area: real
35     float diametro, area;
36
37
38     // Preguntar radio
39     cerr << "Digite el radio: ";
40     cin >> radio;
41
42     // procesos
43     diametro = 2 * radio;
44     area= 3.14 * radio * radio;
45
46
47     // Datos de salidas
48     // mostrar diametro, area
49     cout << "Diametro: " << diametro <<endl;
50     cout <<"Área: "<< area;
51 // Fin
52 }
```

```
> clang++-7 -pthread -std=c++17 -o main main.cpp
> ./main
Digite el radio: 2.3
Diametro: 4.6
Área: 16.6106>
```



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"

Operadores

Existen tres tipos de operadores:

- Aritméticos
- Relacionales
- Lógicos



Operadores

Aritméticos

Nombre	Símbol	Ejemplo
	o	
Suma	+	$6 + 2 = 8$
Resta	-	$6 - 2 = 4$
Multiplicación	*	$6 * 2 = 12$
División	/	$6 / 2 = 3$
Módulo (guarda el residuo de la división)	%	$6 \% 2 = 0$ $5 \% 2 = 1$

Fuente: Elaboración propia



Operadores

Relacionales

Nombre	Símbol	Ejemplo
	o	
Menor	<	$5 < 3$
Menor o igual	<=	$5 <= 5$
Mayor	>	$5 > 3$
Mayor o igual	>=	$5 >= 3$
Igual	==	$5 == 5$
Diferente	!=	$5 != 5$

Fuente: Elaboración propia



Operadores

Lógicos

Nombre	Símbol	Ejemplo
	o	
And	&&	$(5 < 3) \ \&\& \ (8 > 3)$
Or		$(5 < 3) \ \ (8 > 3)$

Fuente: Elaboración propia



Actividad independiente

Realizar los siguientes ejercicios implementando cada uno de los pasos explicados (análisis del problema, prueba de escritorio, diagrama de flujo, pseudocódigo).

1. Realizar un programa en C++ que muestre el resultado de la suma de dos números enteros.
2. Una tienda de tenis ofrece el 20 % de descuento sobre el total de la compra, el cliente desea saber cuánto debe pagar por la compra.



Referencias bibliográficas

- Deitel, P. &. (2021). Como programar C++. USA: Pearson.
- Jiménez, J. &. (2015). Fundamentos de programación. Diagramas de flujo, Diagramas N-S, Pseudocódigo y Java. México D.F.: Alfaomega.
- Joyanes, L. &. (2014). Programación en C, C++, Java y UML. Madrid, España: McGraw-Hill.
- Joyanes, L. (2008). Fundamentos de programación, estructura de datos y objetos. Madrid, España: Mc Graw Hill.
- Villalobos, R. (2014). Fundamentos de Programación C++ - Más 100 Algoritmos codificados. Perú: Macro.



Thank
you!



CORHUILA

CORPORACIÓN UNIVERSITARIA DEL HUILA
Vigilada Mineducación



CORPORACIÓN UNIVERSITARIA DEL HUILA - CORHUILA
"Diseño y prestación de servicios de docencia, investigación y extensión de programas de pregrado, aplicando todos los requisitos de las normas ISO implementadas en sus sedes Neiva y Pitalito"