

“UNIVERSIDAD POLITÉCNICA DE PÉNJAMO”



Nombre del Alumno: Brandon Iván Márquez Morales.

Grado y Grupo: 1° “A”

Carrera: Ingeniería en Software.

Materia: “Algoritmos”

Trabajo: Reporte de códigos documentados en los diferentes lenguajes y Pseudocódigo.

Docente: Miguel Ángel Saldaña Cabeza.

04/10/2020

Índice.

1.-índice

2.-Introducción

3.-Marco teórico

4.-Desarrollo

5.-Conclusión

6.-Referencias

Introducción.

En el presente reporte se hablará del desarrollo de un algoritmo sobre mostrar las tablas de multiplicar del uno al diez utilizando con un ciclo de tal manera que se le pregunte al usuario que, si quiere repetir el programa teniendo que ingresar lo que se pide, principalmente mostrando su diagrama de flujo hecho a mano donde seguidamente se hará su pseudocódigo en el programa de lenguaje natural "PSeInt".

Posteriormente se mostrará la codificación en tres lenguajes de programación diferentes los cuales son: Python, Ruby y C++.

En cada lenguaje programación mencionado y el lenguaje natural o mejor dicho PSeInt se mostrará su documentación donde se explicará a detalle y paso a paso que significa cada cosa, para que sirve y como funciona, De esta manera será muy fácil aprender y recordar cada estructura o sentencia de cada uno de los lenguajes de programación y el PSeInt o lenguaje natural.

Marco teórico.

¿Qué es un algoritmo?

*Un algoritmo es la receta para resolver un conjunto de problemas.

*Es un conjunto de instrucciones que indican paso a paso que hay que hacer.

*Un algoritmo es conceptual y puede ser descrito usando el lenguaje natural que usamos día a día, diagramas de flujo o pseudocódigo.

*Un algoritmo puede implementarse en diferentes lenguajes de programación.

¿Qué es un diagrama de flujo?

Un diagrama de flujo es un diagrama que describe un proceso, sistema o algoritmo informático. Se usan ampliamente en numerosos campos para documentar, estudiar, planificar, mejorar y comunicar procesos que suelen ser complejos en diagramas claros y fáciles de comprender. Los diagramas de flujo emplean rectángulos, óvalos, diamantes y otras numerosas figuras para definir el tipo de paso, junto con flechas conectoras que establecen el flujo y la secuencia. Pueden variar desde diagramas simples y dibujados a mano hasta diagramas exhaustivos creados por computadora que describen múltiples pasos y rutas.

¿Qué es un lenguaje de programación?

En informática, se conoce como lenguaje de programación a un programa destinado a la construcción de otros programas informáticos. Su nombre se debe a que comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano.

Dicho lenguaje está compuesto por símbolos y reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye el código fuente de una aplicación o pieza de software determinado. Así, puede llamarse también lenguaje de programación al resultado final de estos procesos creativos.

¿Qué es un programa?

Un programa es un conjunto de pasos lógicos escritos en un lenguaje de programación que nos permite realizar una tarea específica. El programa suele contar con una interfaz de usuario, es decir, un medio visual mediante el cual interactuamos con la aplicación. Algunos ejemplos son la calculadora, el navegador de internet, un teclado en pantalla para el celular, etc.

¿Qué es Python?

Es un lenguaje de programación versátil multiplataforma y multiparadigma que se destaca por

su código legible y limpio. Una de las razones de su éxito es que cuenta con una licencia de código abierto que permite su utilización en cualquier escenario.

¿Qué es la instrucción “print” en Python?

Una de las acciones básicas e imprescindibles que tiene que realizar un programa es la de mostrar información por pantalla: texto, números, resultados...

Para mostrar texto en Python utiliza la función `print()`, cuya sintaxis es:

```
>>>print('texto')
```

texto

```
>>>cadena = "Esto es una cadena"
```

```
>>>print(cadena)
```

Esto es una cadena

¿Qué es Ruby?

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Combina una sintaxis inspirada en Python y Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Lisp, Lua, Dylan y

CLU. Ruby es un lenguaje de programación interpretado en una sola pasada y su implementación oficial es distribuida bajo una licencia de software libre.

¿Qué es “puts” en Ruby?

La función puts se usa para escribir una secuencia de caracteres al flujo de salida estándar, Además de que este da un salto de línea, Ejemplo:

```
#include <stdio.h>
```

```
int *puts(const char *s);
```

¿Qué es C++?

C++ es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido.

¿Qué es el tipo de dato “int” en C++?

Este tipo de datos permite trabajar con cualquier número que se encuentre dentro del conjunto numérico de los enteros, por ejemplo 1, 10, 56, 32, etc. Para designar este tipo de datos, se utilizan las siguientes palabras:

* int – 16 bits

- * long – 32 ó 64 bits (dependiendo del compilador y procesador)

¿Qué es el tipo de dato “float” en C++?

Este tipo de datos permite trabajar con cualquier número que se encuentre dentro del conjunto numérico de los reales, en otras palabras cualquier número, incluidos los que tienen fracciones decimales, por ejemplo 3.456, 7.231, 65.43, etc. Para designar este tipo de datos, se utilizan las siguientes palabras:

- * float – 16 bits
- * double – 32 bits

¿Qué es una librería?

En programación, una librería es un archivo o conjunto de archivos que se utilizan para facilitar la programación. Las librerías, también llamadas "frameworks", consisten en archivos de código a los que llamamos al principio de la página, por ejemplo, una librería JavaScript será un archivo en JavaScript que insertamos al principio de la página.

¿Para que sirve la librería `#include<iostream>` en C++?

Esta librería sirve para Entrada y Salida de datos en C++ i: input, o: output, stream: corriente.

¿Para qué sirve “using namespace std;” en C++?

Instrucción para indicar al programa que se usará de manera estándar el flujo de entrada y salida de datos y poner: `cout<<` y `cin>>` sin estar que poniendo siempre “`std : :`” antes de ellos.

¿Qué es y para que sirve “int main() {}” en C++?

Es la función principal de C++ que sirve para poner dentro todo el código a desarrollar.

¿Qué es y para qué sirve “return 0;” en C++?

Retorno para que el programa detecte que ha finalizado todo correctamente.

¿Para qué sirve la instrucción “cout<<” en C++?

La instrucción `cout` es utilizada como método de salida de datos por pantalla (realiza una impresión en pantalla). La sintaxis de uso de esta instrucción (forma de escribir correctamente la instrucción) es la siguiente:

```
cout<<"Mensaje a imprimir";
```

¿Para qué sirve la instrucción “cin>>” en C++?

La instrucción `cin`, es utilizada para la entrada de datos por medio del teclado. La sintaxis de uso de esta instrucción es la siguiente:

```
cin>>NombreDeLaVariable;
```

¿Qué es PSeInt?

PSeInt es la abreviatura de los estados de computación de PSeudo Intérprete, una herramienta educativa creada en Argentina, utilizada principalmente por estudiantes para aprender los fundamentos de la programación y el desarrollo de la lógica. Es un software muy popular de su tipo y es ampliamente utilizado en universidades de Latinoamérica y España.

Utiliza pseudocódigo para la solución de algoritmos.

¿Qué es un ciclo?

Un bucle o ciclo, en programación, es una secuencia que ejecuta repetidas veces por una instrucción/un trozo de código, hasta que la condición asignada a dicho bucle deja de cumplirse. Los tres bucles más utilizados en programación son el bucle while, el bucle for y el bucle do-while.

¿Qué es una variable?

Una variable es donde se guarda (y se recupera) datos que se utilizan en un programa.

Cuando escribimos código, las variables se utilizan para:

Guardar datos y estados.

Asignar valores de una variable a otra.

Representar valores dentro de una expresión matemática.

Mostrar valores por pantalla.

Todas las variables deben ser de un tipo de datos, ya sea un dato de tipo primitivo, como un número o texto, o un dato abstracto, como un objeto que se ha creado.

Lo habitual es que haya una serie de restricciones. La mayoría de los lenguajes de programación tienen estas condiciones:

Pueden ser letras, números y el símbolo `_`. Con estos límites:

no puede tener espacios.

no empezar con un número.

no puede ser una palabra reservada. Por ejemplo, `if`, `for`, `while`...

Algunos lenguajes pueden permitir empezar con carácter especial, como `$`.

¿Qué son los datos de entrada y salida?

En un programa, los datos de entrada son los que la computadora va a procesar. Los datos de salida son datos derivados, es decir, obtenidos a partir de los datos de entrada. Por esta razón, a los datos de salida se les considera más significativos que a los datos de entrada. Ambos tipos de datos son información (textos, imágenes, sonidos, vídeos,...) que maneja la

computadora. Sin embargo, en un sentido más filosófico, a los datos de entrada se les considera la materia prima de los datos de salida, considerados estos como la verdadera información.

¿Qué es una cadena de caracteres?



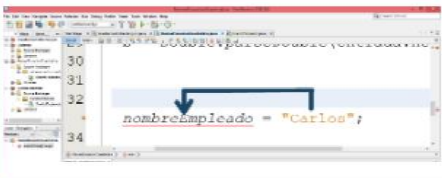
Una cadena también llamada "string" (así es más común porque casi todos los lenguajes de programación son en inglés son un tipo de datos que representan texto. Se llaman cadenas porque se componen de caracteres únicos encadenados, es decir ligados entre sí. Pueden ser asignadas a una variable (dato introducido por el usuario del cual el valor o descripción puede cambiar con cada ejecución del programa) o combinarlas.

¿Qué es el ciclo for?

Los ciclos for son lo que se conoce como estructuras de control de flujo cíclicas o simplemente estructuras cíclicas, estos ciclos, como su nombre lo sugiere, nos permiten ejecutar una o varias líneas de código de forma iterativa, conociendo un valor específico inicial y otro valor final, además nos permiten determinar el tamaño del paso entre cada "giro" o iteración del ciclo.

¿Qué es una operación en programación?

Una vez que se comprende el concepto de variable y/o constante, sus correspondientes tipos de datos y las expresiones que pueden escribirse al combinar operadores y operandos es posible realizar operaciones, es decir acciones que conduzcan a disponer de variables con valores que pueden obtenerse ya sea por parte de los usuarios de los programas o a su vez como resultado de la ejecución de una expresión. Ejemplo:

OPERACIONES	REPRESENTACIÓN GRÁFICA
LECTURA	
ESCRITURA	
ASIGNACIÓN	

¿Qué es imprimir en programación?

Permite llevar hacia la salida estándar (la pantalla) los valores (datos) obtenidos de la evaluación de una lista de argumentos.

En programación, los argumentos (también llamados parámetros) se emplean para transferir datos de un

programa (o subprograma) llamante a otro llamado, y viceversa (del llamado al llamante).

Descripción del Algoritmo.

“Se pide mostrar las tablas de multiplicar de dicho numero que el usuario elija”

Principalmente se agrega un ciclo para que una vez que termine la operación se le pregunte al usuario si desea repetir el programa o hacer u otro cálculo y a esto se le conoce como: Mientras (En el lenguaje natural que conocemos o Pseudocódigo) y Ciclo While en la codificación. Y esto se va a repetir siempre y cuando se cumpla una condición.

Dentro del ciclo posteriormente se le pide un numero ya sea entero o decimal al usuario y después se le da lectura al dato, una vez este guardado el dato se utiliza el ciclo for ó para donde principalmente se le da un nombre a nuestra variable, después se asigna el límite hasta donde va a llegar nuestra variable y finalmente se asigna el incremento o decremento.

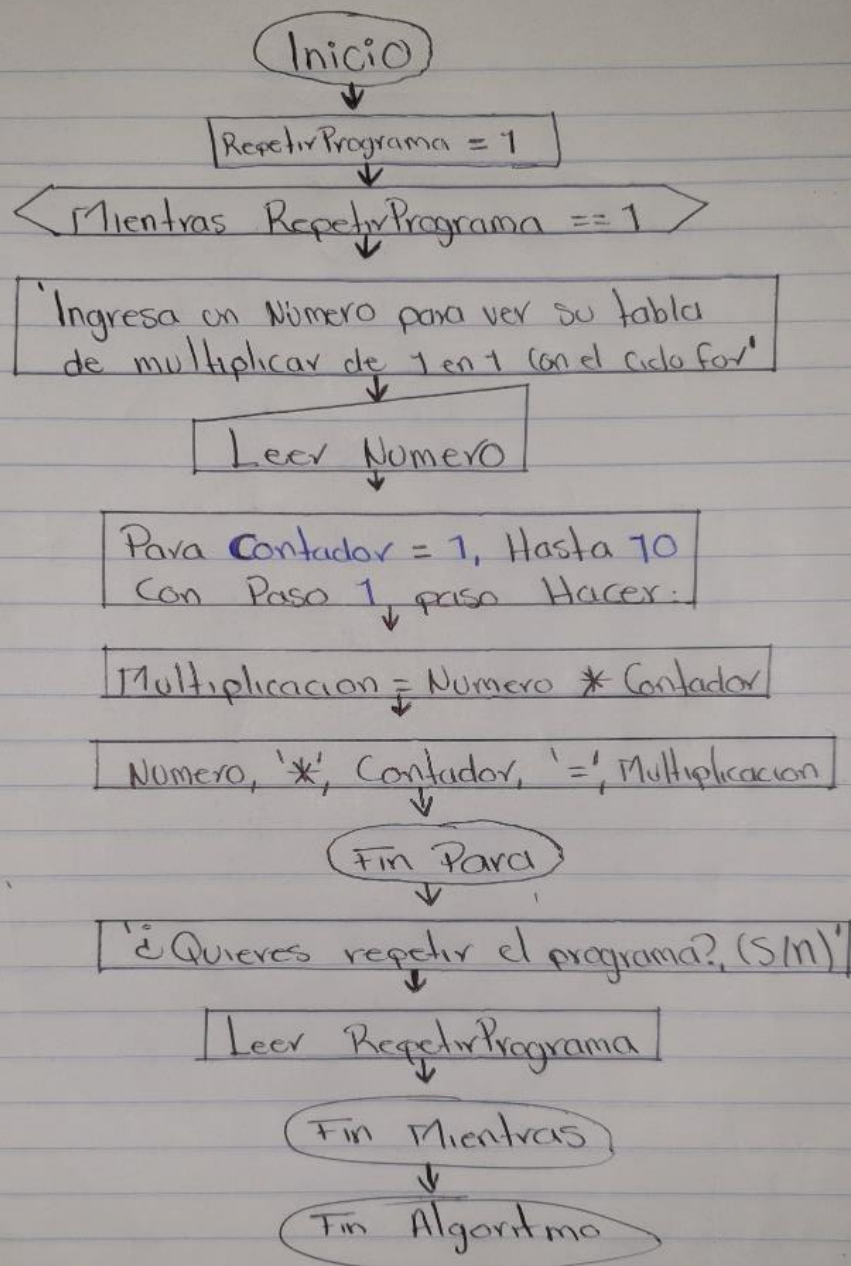
Después se hace la operación en este caso una multiplicación del número ingresado por el usuario y nuestra variable asignada en el ciclo for.

Finalmente imprimimos el resultado de la multiplicación con número ingresado por el usuario, el signo de multiplicación, la variable del ciclo for, el signo de igual y el resultado de la multiplicación y esto quedaría así: Numero,"*",Contador,"=", Multiplicación.

Y por último se le pregunta al usuario que si desea hacer otro calculo, si la condición al inicio dentro del ciclo while se cumple se repite el programa, si no s cumple termina el Algoritmo.

Desarrollo.

"Diagrama de flujo"



ISW 1ºA Brandon Iván Márquez Mordales

Pseudocódigo documentado en el programa “PSelnt”:

```
1  Algoritmo CicloFor //Aquí se ingresa el nombre del Algoritmo a desarrollar
2
3  RepetirPrograma = "s"; //Se esta declarando una variable "RepetirPrograma" igual a el caracter: "s" para despues compararla y usarla en una funcion para repetir el programa
4
5  Mientras RepetirPrograma == "s" Hacer //En esta funcion se esta comparando que si cuando se lea la variable "RepetirPrograma" es igual a "s" se va a repetir todo lo que-
6  //esta dentro de la funcion "Mientras"
7
8  Escribir "Ingresa un úmero para ver su tabla de Multiplicar de 1 en 1"; //La instrucción "Escribir" Sirve para imprimir un mensaje en pantalla o el resultado de algo
9
10 Leer Numero; //Aquí se lee el dato de entrada ingresado por el usuario y se almacena en la variable "Numero"
11
12 //La funcion "Para" sirve para asignar un valor a una variable, "Hasta" sirve para indicar hasta que cantida es limite que llegara el valor inicial
13 //Y "Con Paso 1 paso Hacer" Sirve para inidicar en que catidad se ir haciendo un incremento o un decremento
14 Para Contador = 1 Hasta 10 Con Paso 1 paso Hacer
15
16     Multiplicacion = Numero * Contador; //En esta líea se está haciendo una asignación a la variable "Multiplicacion" y se le asigna la multiplicacion
17     //de la variable "Numero" por la variable "Contador"
18
19     //En esta línea se imprimira en pantalla la variable "Numero" ingresada por el usuario, un caracter de tipo cadena: "*", La variable "Contador",-
20     //un caracter de tipo cadena: "=" y la variable "Multiplicacion" que es el resultado de la Multiplicacion de: Numero * Contador
21     Escribir Numero, "*", Contador, "=", Multiplicacion;
22
23
24 Fin Para //Esta instrucción esta indicando y cerrando la finalizacion del Ciclo llamado "Para"
25 Escribir "¿Quieres repetir el programa?, (s/n)"; //Se imprime en pantalla y se pregunta que si quiere repetir el programa contestando "s" para si y "n" para no
26 Leer RepetirPrograma; //Se lee la variable "Repetir Programa" para comprobar si se va o no a repetir el programa de dicha condicion en la instruccion Mientras
27 Fin Mientras //Esta instrucción esta indicando y cerrando la finalizacion del Ciclo llamado "Mientras"
28 FinAlgoritmo //Se esta indicando el cierre del programa o del Algoritmo
```

Código documentado en el lenguaje de programación “Python”:

```
CicloFor.py  CicloFor.rb

1 RepetirPrograma = "s" #Se está declarando variable llamada "RepetirPrograma" igual al caracter "s"
2 #Se usa el ciclo "while" y sirve para repetir ciertas acciones siempre y cuando se cumpla una condición
3 #Su lectura es: "Mientras la variable "RepetirPrograma" se igual a el caracter "s" se va repetir todo lo que está después de los dos puntos ":"
4 while (RepetirPrograma == "s"):
5     #En esta línea se le está asignando a la variable "Numero" su tipo de dato entero (int) e indicando que es un dato de entrada (input)-
6     #donde se está indicando que el valor ingresado por el usuario es de entrada y se almacenará en la variable "Numero"
7     #Y a la vez en la misma línea se imprime un mensaje en pantalla dentro de paréntesis () y comillas dobles: ""
8     #Y por último "\n" al final de la cadena significa un salto de línea, es decir la siguiente impresión de cadena o resultado se imprimirá-
9     # en la siguiente línea.
10    Numero = int(input("Ingrese un Número para ver su tabla de Multiplicar de 1 en 1 con el Ciclo for:\n"))
11    #La sentencia for es una secuencia que ejecuta repetidas veces por una instrucción/un trozo de código, hasta que la condición asignada-
12    # a dicho bucle deja de cumplirse
13    #Su lectura es: para Contador en el rango y entre paréntesis se coloca donde inicia después separado de una coma donde termina
14    #En este caso es un incremento ya en donde termina el rango es positivo, si queremos que decremente el valor se pone negativo
15    for Contador in range(1,10):
16        #En esta línea se hace una asignación a la variable "Multiplicacion" el valor de la Multiplicación de la variable "Numero" por la variable "Contador"
17        Multiplicacion = Numero * Contador
18        #La instrucción "print" sirve para imprimir una cadena de caracteres o el resultado de una variable y todo lo que se imprima va dentro de: ()
19        #Su lectura es: Imprimir la variable "Numero", carácter de cadena "*", variable "Contador", carácter de cadena "=" y variable "Multiplicacion"
20        print(Numero, "*", Contador, "=", Multiplicacion)
21        #Se le pregunta al usuario si quiere repetir el programa y posteriormente se hace la lectura de la variable "RepetirPrograma"-
22        #para comprobar si se cumple la condición de la asignación para repetir el programa
23    RepetirPrograma = input("¿Quieres repetir el programa?, (s/n)\n")
24
```

Código documentado en el lenguaje de programación “Ruby”:

```
CicloFor.py  CicloFor.rb

1  #Para representar una variable en Ruby, primero se pone el nombre de la variable y despues se le asigna el valor en este caso se le asigna: "s"
2  #Posteriormente esta variable se usará para compararla y darle lectura al final para ver si se cumple o no la condicion para repetir el programa
3  RepetirPrograma = "s"
4  #Para representar el ciclo while en Ruby se pone while despues el nombre de la variable declarada que es "RepetirPrograma" y se compara-
5  #con los signos: "==" y por último se coloca el valor asignado a comparar
6  #Su lectura es: mientras la variable RepetirPrograma se igual a el caracter "s", Se va a repetir todo lo que esta dentro del while o ciclo
7  #y se finaliza con "end" para indicar que ahí termina lo que se estaría ciclando
8  while RepetirPrograma == "s"
9  #La instrucción "puts" sirve para imprimir un mensaje en pantalla hacia el usuario dando como resultado una cadena de caracteres o el resultado-
10 #de una variable y su lectura es: Imprimir (entre comilllas "" se pone lo que queramos imprimir) que es lo que esta en letras color aqua
11   puts "Ingresa un Número para ver sus tabala de Multiplicar de 1 en 1 con el Ciclo for"
12 # "gets.chomp" es la instrucción que sirve para leer una variable de tipo cadena o string que es una cadena de caracteres
13 # "gets.to_i" es la instrucción que sirve para leer una variable de tipo de dato entero como por ejemplo: 10
14 # "gets.to_f" es la instrucción que sirve para leer una variable de tipo flotante o entero y decimal como por ejemplo: 10 y 10.5(Acepta los 2 tipos).
15   Numero = gets.to_i#Se esta declarando una variable con instrucción "gets.to_i" para almacenar el dato digitado por el usuario en la variable"Numero"
16   for Contador in (1..10)#Se usa el ciclo "for" en un rango de 1-10 y se pone entre: () separado de dos puntos despues de la inicialización
17     Multiplicacion = Numero * Contador#Se está asignando una variable a una operación en este caso una multiplicación de la variable "Numero"-
18     #por la variable "Numero"
19     print Numero, "*", Contador, "=", Multiplicacion, "\n"#en Ruby tambien se utiliza "print" al igual que "python" que sirven para imprimir-
20     #La diferencia es que puts da un salto de línea y print no lo hace para ello en print se usa al final: "\n"
21   end #Esta instrucción "end" sirve para indicar que hasta aquí termina el ciclo for
22   puts "¿Quieres repetir el programa?, (s/n)"#Se le pregunta al usuario si quiere repetir el programa con "puts"
23   RepetirPrograma = gets.chomp#Se lee la variable "RepetirPrograma" para posteriormente compararla y ver si si o no se hace el ciclo
24 end#Esta instrucción "end" sirve para indicar que hasta aquí termina el ciclo while
25
```

Ruby\CicloFor.rb 25:1

CRLF UTF-8 Ruby GitHub Git (0)

Código documentado en el lenguaje de programación "C++":

```
1 #include <iostream> //Librería para Entrada y Salida de datos en C++ i:input, o:output, stream:corriente.
2 using namespace std; //Instrucción para indicar al programa que se usará de manera estándar el flujo de entrada y salida de datos (cout, cin)
3 int main() { //Función principal de C++ donde dentro de ella va todo el código
4     int Numero, Multiplicacion, Contador; //Declaración de variables de tipo entero (int) separados por: , y al final va: ;
5     string RepetirPrograma = "s"; //Declaración y asignación de variable de tipo string o cadena y su asign. se pone entre: ""
6     while (RepetirPrograma == "s") { //Uso del ciclo while para hacer una comparación y si se cumple se cicla lo de adentro de: {}
7         //Se imprime mensaje en consola con la instrucción: cout<<
8         cout<<"Ingrese un Numero para ver su tabla de Multiplicar\n";
9         cin>>Numero; //Se lee variable "Numero" con la instrucción: cin>>
10        //Se usa ciclo donde: Se inicializa una variable, Se indica su límite y su incremento o decremento en este caso: incremento
11        //Su lectura es: Para Contador igual a 1 hasta menor o igual a 10, Contador va ir incrementando hasta su límite
12        for (Contador=1; Contador<=10; Contador++){
13            //Se hace una asignación a la variable "Multiplicación" igual a la variable "Numero" por la variable "Contador"
14            Multiplicacion = Numero * Contador;
15            //Se imprime en consola las variables: "Numero", "Contador", "Multiplicacion" y caracteres de cadena: "*" y "="
16            // "\n" sirve para dar un salto de línea
17            cout<< Numero << "*" << Contador << "=" << Multiplicacion << "\n";
18        }
19        //Se usa instrucción "cout<<" para preguntar al usuario si quiere repetir el programa "\n" sirve para dar un salto de línea
20        cout<<"Desea repetir el programa, (s/n)\n";
21        cin>>RepetirPrograma; //Se us instrucción "cin>>" para leer la variable "RepetirPrograma" y posteriormente compararla
22    }
23    return 0; //Retorno para que el programa detecte que a finalizado todo correctamente
24 }
25
```

Conclusión.

Me he llevado un aprendizaje más captado ya que ya tenía conocimiento sobre las preguntas del marco teórico, pero ahora que me puse a investigar algunos otros conceptos diferentes me ha quedado cien por ciento claro todo sobre los fundamentos de programación cosa que es muy buena para así seguir aprendiendo más cosas nuevas y tener la facilidad de comprender mejor y más fácil gracias a estos conocimientos adquiridos.

Sobre el desarrollo del algoritmo en el diagrama de flujo, pseudocódigo (PSeInt), Codificación en Python, Ruby y C++, no tuve ningún problema, En el diagrama de flujo que lo hice en el programa “DFD” fue muy sencillo ya que solo fue colocar las figuras correspondientes a cada paso del Algoritmo y pues su uso o se podría decir sintaxis lo se usar muy bien ya que, ya lo he manejado antes, pero personalmente a mi me gusta más de desarrollar el diagrama de flujo a mano.

En Python y Ruby simplemente si no recordaba alguna de sus instrucciones o sintaxis consulte el material de apoyo que está en Classroom en la clase de algoritmos (lenguajes.docx) pero en realidad no fue el caso ya que con todas las prácticas que hemos realizado he logrado aprender la sintaxis de Python y Ruby.

En el caso de C++, Ahí si no tengo problema ya que yo ya estoy familiarizado con los lenguajes de la familia “C” en este caso vengo de programar desde hace 2 años programando en el lenguaje “C” y “C#” y pues su sintaxis es muy parecida y se me facilita mucho en C++.

Pero también debo decir que me he llevado una buena experiencia al codificar el algoritmo en estos lenguajes ya que me ayuda a mejorar bastante para tener conocimiento de nuevos lenguajes y avanzar más.

Referencias.

Robledano. (2020, 3 julio). *Qué es Python: Características, evolución y futuro*. OpenWebinars.net. [https://openwebinars.net/blog/que-](https://openwebinars.net/blog/que-es-python/#:%7E:text=%20Qu%C3%A9%20es%20Python%3A%20Caracter%C3%ADsticas%2C%20evoluci%C3%B3n%20y%20futuro,Python.%20Simplificado%20y%20r%C3%A1pido%3A%20Este%20lenguaje...%20More%20)

[es-](https://openwebinars.net/blog/que-es-python/#:%7E:text=%20Qu%C3%A9%20es%20Python%3A%20Caracter%C3%ADsticas%2C%20evoluci%C3%B3n%20y%20futuro,Python.%20Simplificado%20y%20r%C3%A1pido%3A%20Este%20lenguaje...%20More%20)

[python/#:%7E:text=%20Qu%C3%A9%20es%20Python%3A%20Caracter%C3%ADsticas%2C%20evoluci%C3%B3n%20y%20futuro,Python.%20Simplificado%20y%20r%C3%A1pido%3A%20Este%20lenguaje...%20More%20](https://openwebinars.net/blog/que-es-python/#:%7E:text=%20Qu%C3%A9%20es%20Python%3A%20Caracter%C3%ADsticas%2C%20evoluci%C3%B3n%20y%20futuro,Python.%20Simplificado%20y%20r%C3%A1pido%3A%20Este%20lenguaje...%20More%20)

colaboradores de Wikipedia. (2020b, septiembre 5). *Bucle (programación)*. Wikipedia, la enciclopedia libre.

[https://es.wikipedia.org/wiki/Bucle_\(programaci%C3%B3n\)#:%7E:text=Un%20bucle%20o%20ciclo%2C%20en%20programaci%C3%B3n%2C%20es%20una,condici%C3%B3n%20asignada%20a%20dicho%20bucle%20deja%20de%20cumplirse.](https://es.wikipedia.org/wiki/Bucle_(programaci%C3%B3n)#:%7E:text=Un%20bucle%20o%20ciclo%2C%20en%20programaci%C3%B3n%2C%20es%20una,condici%C3%B3n%20asignada%20a%20dicho%20bucle%20deja%20de%20cumplirse.)

(2019, 7 mayo). *¿Qué es una variable? Las variables se suelen utilizar en muchos ámbitos diferentes, como por ejemplo, en matemáticas. Pero aquí te Read more*. Lenguajes de programación. <https://lenguajesdeprogramacion.net/diccionario/que-es-una-variable-en-programacion/>

CarlosPes.com. (2015). *Definición de Datos de Entrada y Datos de Salida - Concepto y Significado*. ConceptoDe.

http://www.carlospes.com/minidiccionario/datos_de_entrada_y_de_salida.php#:%7E:text=En%20un%20programa%2C%20los%20datos%20de%20entrada%20son,m%C3%A1s%20significativos%20que%20a%20los%20datos%20de%20entrada.

(2010a). *¿Qué es una cadena de caracteres?* Yahoo.

<https://es.answers.yahoo.com/question/index?qid=20091208165723AAECeLy>

S.E.M.A. (2012). *ProgramarYA.com*. ProgramarYA. <https://www.programarya.com/Cursos/C%20%20/Ciclos/Ciclo-for#:%7E:text=Los%20ciclos%20for%20son%20lo%20que%20se%20conoce,iterativa%2C%20que%20nos%20permite%20ejecutar%20de%20manera%20>

CarlosPes.com. (2013). *La Función de Imprimir de Carlos Pes*.

http://www.carlospes.com/curso_de_lenguaje_c/01_09_01_la_funcion_printf.php#:%7E:text=La%20funci%C3%B3n%20printf%20permite%20llevar%20hacia%20la%20salida,para%20transferir%20datos%20de%20un%20programa%20%28o%20

colaboradores de Wikipedia. (2020b, septiembre 2). *Ruby*. Wikipedia, la enciclopedia libre. <https://es.wikipedia.org/wiki/Ruby>

colaboradores de Wikipedia. (2020e, septiembre 24). *C++*. Wikipedia, la enciclopedia libre.

<https://es.wikipedia.org/wiki/C++#Ejemplos>

colaboradores de Wikipedia. (2020d, septiembre 6). *PSeInt*. Wikipedia, la enciclopedia libre.

<https://es.wikipedia.org/wiki/PSeInt#:~:text=PSeInt%20es%20un%20software%20libre%20educativo%20multiplataforma%20dirigido,a%20personas%20que%20se%20inician%20en%20la%20programaci%C3%B3n.>

J.P.E.R.E.Z. (2012b). *¿Qué es puts?* Arquitectura de sistemas.

http://www.it.uc3m.es/pbasanta/asng/course_notes/input_output_function_puts_es.html

Acosta, I. C. (2020, 21 enero). *TIPOS DE DATOS EN C++*. Geek Electrónica. <https://geekelectronica.com/tipos-de-datos-en-c/>

ADR Formación. (2020, 22 abril). *La función Print() en Python*. adr formación.

https://www.adrformacion.com/knowledge/programacion/2__print_.html#:~:text=En%20Python%203%20print%20%28%29%20es%20una%20funci%C3%B3n%2C,comas%20o%20a%20trav%C3%A9s%20de%20las%20cadenas%20%27f%27.

(2012). *Librerías para la programación web*. Aprende Web. <https://aprende->

[web.net/librerias/#:~:text=En%20programaci%C3%B3n%2C%20una%20librer%C3%ADa%20es%20un%20archivo%20o,javascript%20que%20insertamos%20al%20principio%20de%20la%20p%C3%A1gina.](https://aprende-web.net/librerias/#:~:text=En%20programaci%C3%B3n%2C%20una%20librer%C3%ADa%20es%20un%20archivo%20o,javascript%20que%20insertamos%20al%20principio%20de%20la%20p%C3%A1gina.)

¿Da igual usar `#include` o `#include <iostream>`? (2015, 31 diciembre). Stack Overflow en español.

<https://es.stackoverflow.com/questions/1959/da-igual-usar-include-iostream-o-include-iostream>

`c++` - ¿Para qué sirve `using namespace std`? (2012). stackoverrun. <https://stackoverrun.com/es/q/5159492>

Díaz, R. (2009, 10 marzo). *Uso de la función “main” en C/C++*. int main en español. <https://blog.r2d2rigo.es/2009/03/10/uso-de-la-funcin-main-en-cc/>

C. (2020, 10 junio). *return (Instrucción) (C)*. Microsoft Docs. <https://docs.microsoft.com/es-es/cpp/c-language/return-statement-c?view=vs-2019>

Acosta, I. C. (2020b, febrero 4). *ENTRADA Y SALIDA DE DATOS EN C++ (cin y cout)*. Geek Electrónica. <https://geekelectronica.com/entrada-y-salida-de-datos-en-c-cin-y-cout/>

¿Qué es un diagrama de flujo. (2012). Lucidchart. <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo#:~:text=Un%20diagrama%20de%20flujo%20es%20un%20diagrama%20que,complejos%20en%20diagramas%20claros%20y%20f%C3%A1ciles%20de%20comprender.>

¿Qué es un algoritmo? (2010). edu JAVA. <http://www.edu4java.com/es/conceptos/que-es-un-algoritmo-que-es-un-programa.html#:~:text=Un%20algoritmo%20es%20la%20receta%20para%20resolver%20un,indican%20paso%20a%20paso%20que%20hay%20que%20hacer>.

J.T.R.E. (2010c). *¿Qué es un lenguaje de programación?* Concepto.de. <https://concepto.de/lenguaje-de-programacion/>

L.F.R.T. (2012d). *¿Qué es un programa?* / *Desarrollar Inclusión*. Desarrollar Inclusión | Portal de tecnología inclusiva de CILSA. <https://desarrollarinclusion.cilsa.org/tecnologia-inclusiva/que-es-un-programa/#:~:text=%C2%BFQu%C3%A9%20es%20un%20programa%3F%20Un%20programa%20es%20un,visual%20mediante%20el%20cual%20interactuamos%20con%20la%20aplicaci%C3%B3n>.