



El futuro digital
es de todos

MinTIC



Complemento Fundamentos de Programación

Por: Willian Michel Vélez Candia

Misión
TIC 2022

UNIVERSIDAD
EL BOSQUE



Temario

Introducción.

¿Que es Fundamentos de Programación?

Introducción (¿Qué es Programar? - ¿Qué es un Programa? - ¿Qué es un Algoritmo?)

¿Qué es Dato?

¿Qué es un programa y Que es programar?

¿Procesos de entrada salida?

¿Que son los fundamentos de programación?

¿Qué es una variable?

¿Que son las sentencias?

¿Qué es una condición

¿Tipos de condicionales?

Misión
TIC2022



Introducción:



Complemento

Aporte

Proceso de Aprendizaje

El pensamiento computacional implica un conjunto de técnicas y habilidades de resolución de problemas, entre las cuales se hallan:

Descomposición de problemas: dividir un problema en problemas más pequeños y manejables



Introducción:



Existen diferentes formas de diseñar un lenguaje de programación y varios modos de trabajar para obtener los resultados que necesitan los programadores. Se trata de un conjunto de métodos sistemáticos aplicables en todos los niveles del diseño de programas para resolver problemas computacionales.

Convirtiéndose en un Paradigma.



... ¡ERRORES! En la Programación



Muchas personas comienzan a programar
intentando aprender un lenguaje...
¡ERROR!

puesto que no entienden qué están haciendo. ¡Es
como si quisieras aprender un idioma sin saber
hablar!



Es un Proceso para:



Aprender conceptos fundamentales de programación.

Abstraer problemas del mundo real, para representarlos con un algoritmo y, posteriormente, traducirlos a un programa

Pensar como un programador/a

Aprender un lenguaje de programación cualquiera.



Introducción:

(¿Qué es Programar?
¿Qué es un Programa?
¿Qué es un Algoritmo?)





Conceptualización:

¿Qué es Programar?

Es Definir una secuencia de acciones, para la consecución de un fin., por

Ejemplo: La activación de la alarma despertador del Celular.,, en este ejemplo le estas diciendo a tu celular, que necesitas que te despierte a “X” Hora.,, para ir a trabajar.



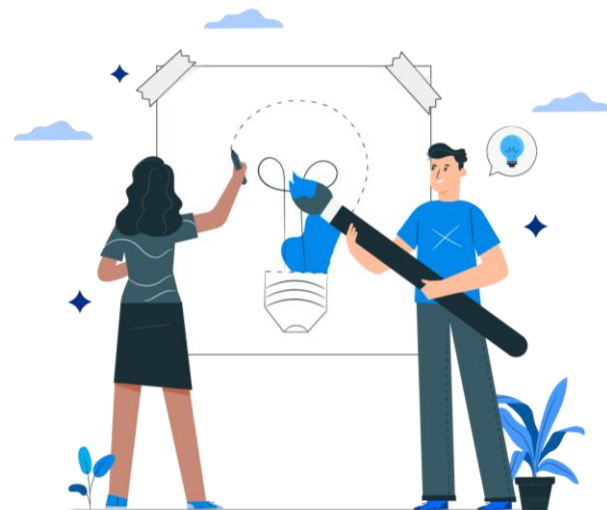
Conceptualización:

También podemos decir que Programar es el **Arte de solucionar problemas**, mediante la utilización del computador como herramienta.,

Es una **combinación** de **instrucciones**, de **datos**, para que finalmente se le pueda dar solución al problema inicial - bajo un concepto básico que es la **Algoritmia**.,

Y un **algoritmo** trata de buscar soluciones a un problema concreto.,. mediante una secuencia de **pasos ordenados**, finitos, y no ambigua para la solución de un problema.,

Ejemplo: paso a para fritar un huevo





Conceptualización:

¿Porque este ejemplo? – ¿Qué se necesita para freír un Huevo?

porque sigue una serie de pasos bien definidos, finitos

1. Ingredientes.
 - a. Un huevo
 - b. Aceite
 - c. Sal
2. Pasos
 - a. Calentar el Aceite
 - b. Una vez este caliente el Aceite (Abrir el Huevo)
 - c. Echar el huevo al sartén
 - d. Añadir un poco de sal
 - e. Esperar un par de minutos
3. Huevo Frito

Secuencia de instrucciones **finitas**, **ordenada** y no **ambigua** (Cada paso está claro) y como receta cumple la especificación de lo que es un algoritmo.



Conceptualización:

¿Qué es un Programa?

Datos de entrada -> Pasos o instrucciones, en donde se resuelve el problema -> >salida

Secuencia ordenada de instrucciones, que a partir de unos datos de entrada dan la solución a un problema y es parecido al concepto de algoritmo



Conceptualización:

Y programar es escribir un algoritmo, en un lenguaje que pueda ser entendido por el computador, para que este lo pueda interpretar y finalmente ejecutar.,

Los Pasos que debe seguir un programador son

1. Analizar (Problema)
2. Resolver (Algoritmo)
3. Implementar(Programa)



Conceptualización:

¿Qué es un Lenguaje de Programación?

Es la forma que se tiene, de escribir un programa que puede interpretar y que ejecuta un computador

Algoritmo: conjunto de instrucciones ordenadas, finitas no ambiguas, que dan la solución a un problema, y se escribe en un pseudocódigo (como se describe informalmente la solución a un problema), que se puede traducir a un lenguaje de programación, como una secuencia de instrucciones, que manipulan unos datos de entrada, para producir una salida entonces e tiene un programa que se puede ejecutar en un computador



Conceptualización:

Ejemplo **restar dos números**

Algoritmo en pseudocodigo (sin sitaxis , sin reglas., palabras inventadas)

Leer numero 1

Leer numero 2

Resultado = numero 1 le resto numero 2

Mostrar el resultado

Fin algoritmo



Conceptualización:

¿Qué es un Dato?

Es una representación simbólica (**numérica, alfabética, algorítmica**, etc.) de un atributo o **variable** cuantitativa. Los datos describen hechos empíricos, sucesos y entidades.

.- Conjunto de **caracteres** con algún significado, pueden ser numéricos, alfabéticos, o alfanuméricos.

Son hechos que describen sucesos e identidades.

Los datos tienden a convertirse en información para ofrecer un significado o ideas





Conceptualización:

¿Qué son los Fundamentos de Programación?

Son la base de la programación y para esto, durante el proceso de aprendizaje es necesario, comprender cómo se utilizan y están integrados por diversos métodos, herramientas, y una serie de pasos





Conceptualización:

¿Qué son los Fundamentos de Programación?

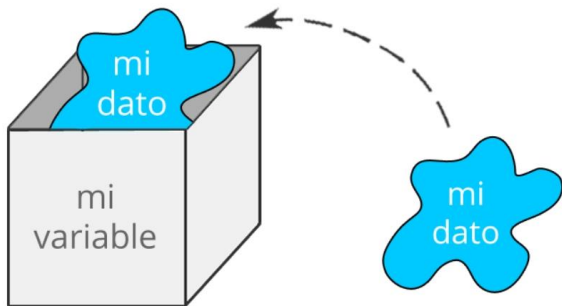


La computadora como unidad de procesamiento capaz de:
manipular, procesar, recuperar, y administrar información
los elementos que permiten la entrada de datos como el
input de datos y el output de datos



Conceptualización:

¿Qué es Variable?



Las variables son simples nombres que se usan para asociarles algún tipo de dato. Con este concepto en lenguaje de programación se declara una variable con el nombre que se quiera, ya depende de ti.



Conceptualización:

Tipos de Variables

Hay ciertas operaciones que no pueden realizarse entre variables de diferente tipo, cómo lo es sumar un **int** con un **str**.

Sin embargo, las asignaciones no tienen restricciones, su una variable antes era **int**, uno puede asignar un valor de otro tipo a esa variable:

```
>>> variable = 1989
>>> type(variable)
<type 'int'>
```

```
>>> variable = '1989'
>>> type(variable)
<type 'str'>
```

```
>>> variable = "Esteban"
>>> type(variable)
<type 'str'>
```



Conceptualización:

Características de las Variables

Las variables están presentes en todos los lenguajes de programación, estas corresponden a contenedores de información, las cuales pueden guardar números, letras, palabras, e incluso funciones en algunos lenguajes.

nombre = valor

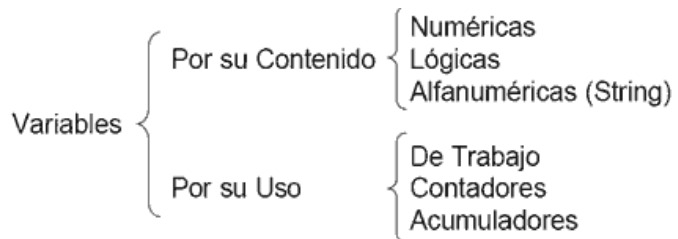
nombre = corresponde a el nombre que uno quiera darle a la variable.

valor = corresponde a la información que uno desea guardar en la variable.



Conceptualización:

Clasificación de las Variables



Es un espacio en la memoria de la computadora que permite almacenar temporalmente un dato durante la ejecución de un proceso, su contenido puede cambiar durante la ejecución del programa.

Para poder reconocer una variable en la memoria de la computadora, es necesario darle un nombre con el cual podamos identificarla dentro de un algoritmo.



Conceptualización:

Por su contenido

Variables Numéricas: Son aquellas en las cuales se almacenan valores numéricos, positivos o negativos, es decir almacenan números del 0 al 9, signos (+ y -) y el punto decimal.

Ejemplo:

iva = 0.19

pi = 3.1416

costo = 2500



Conceptualización:

Por su contenido

Variables Lógicas:

Son aquellas que solo pueden tener dos valores (cierto o falso) estos representan el resultado de una comparación entre otros datos.

Variables Alfanuméricas:

Esta formada por caracteres alfanuméricos (letras, números y caracteres especiales).



Conceptualización:

Por su uso

Variables de Trabajo: Variables que reciben el resultado de una operación matemática completa y que se usan normalmente dentro de un programa.

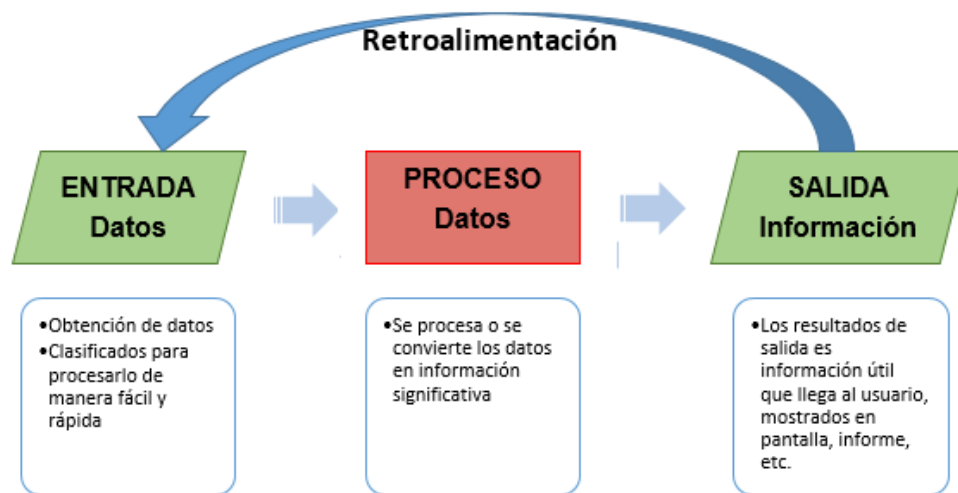
Ejemplo:

Suma = **a + b / c**

Acumuladores: Forma que toma una variable y que sirve para llevar la suma acumulativa de una serie de valores que se van leyendo o calculando progresivamente.



Procesos de las Variables y de la información





Procesos de las Variables y de la información

El sistema de **proceso** de información tiene tres componentes:

Entrada: Son los datos que se introducirán al procesador. Por teclado o pantalla

Proceso: Es la acción que acepta datos de **entrada** para ser procesados.

Salida: Es la información, es decir, es el resultado de los datos ya procesados.



El futuro digital
es de todos

MinTIC



Dentro de esos Procesos Existen

Operadores lógicos



Existen tres operadores lógicos:

and (y),
or (o),
not (no).

El significado semántico de estas operaciones es similar a su significado en inglés. Por ejemplo,



$x > 0$ and $x < 10$

es verdadero sólo cuando x es mayor que 0 y menor que 10.

$n \% 2 == 0$ or $n \% 3 == 0$

es verdadero si cualquiera de las condiciones es verdadera, es decir, si el número es divisible por 2 o por 3.

Estrictamente hablando, los operandos de los operadores lógicos deberían ser expresiones booleanas, pero Python no es muy estricto. Cualquier número distinto de cero se interpreta como “verdadero.”

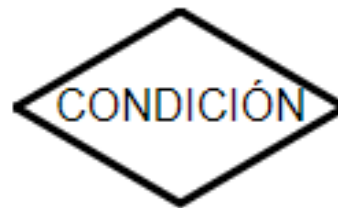


Otros Operadores

operador	comparación
==	es igual que
!=	es distinto de
<	es menor que
<=	es menor o igual que
>	es mayor que
>=	es mayor o igual que



Condicionales





Estructuras de Control Selectivas/Condicionales

Las estructuras selectivas se utilizan para tomar decisiones lógicas; de ahí que se suelen denominar también estructuras de decisión condicionales o alternativas. En Los lenguajes de programación., existen tres tipos de estructuras selectivas:

estructura selectiva simple: if
estructura selectiva doble: if-else
estructura selectiva múltiple: elif.



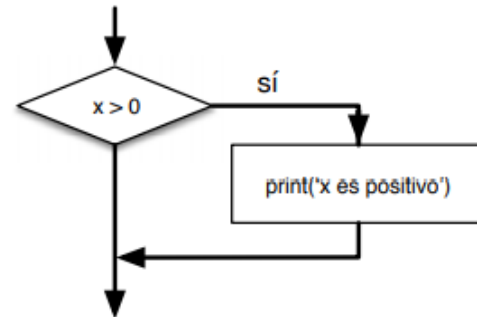
if

La Estructura Selectiva Simple

La estructura de control condicional fundamental es el **if**, y es equivalente al SI.
En estructura de Lenguaje selectiva simple, tiene la siguiente sintaxis:

si (condición)

 sentencia1
 sentencia2
 etc





if

La Estructura Selectiva Simple

La estructura de control condicional fundamental es el **if**, y es equivalente al SI.
En estructura de Lenguaje selectiva simple, tiene la siguiente sintaxis:

si (condición)

 sentencia1
 sentencia2
 etc

```
numero = int(input("Escriba un número Mayor que Cero: "))  
  
if numero > 0:  
    print("Gracias Usted Escribio un Numero Mayo que cero ")  
    print(f"escrito el número {numero}")
```




if - else

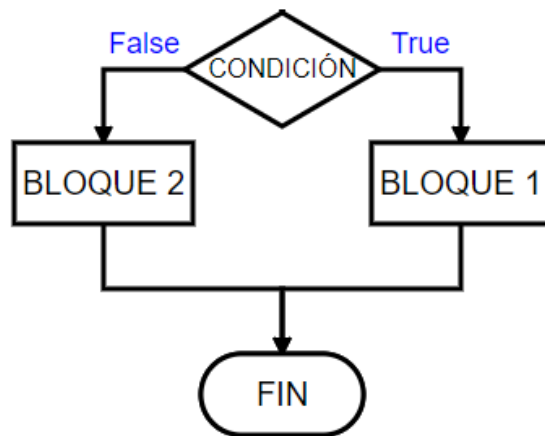
La estructura selectiva doble if-else es equivalente al si-sino de SL. En SL la estructura selectiva doble si-sino, tenía la siguiente sintaxis:

si (condición)

sentencia_si_1
sentencia_si_2
// ..

sino

sentencia_no_1
sentencia_no_2
//..





if - else

```
edad = int(input("¿Cuántos años tiene? "))

if edad < 18:

    print("Es usted menor de edad")
else:
    print("Es usted mayor de edad")

    print("¡Gracias Buen Día!")
```



if – elif -else

La Estructura de la sentencia condicional if ... elif ... else ...

if condición_1:

 bloque 1

elif condición_2:

 bloque 2

else:

 bloque 3

Si se cumple la condición 1,

se ejecuta el bloque 1

Si no se cumple la condición 1,

pero sí se cumple la condición 2,

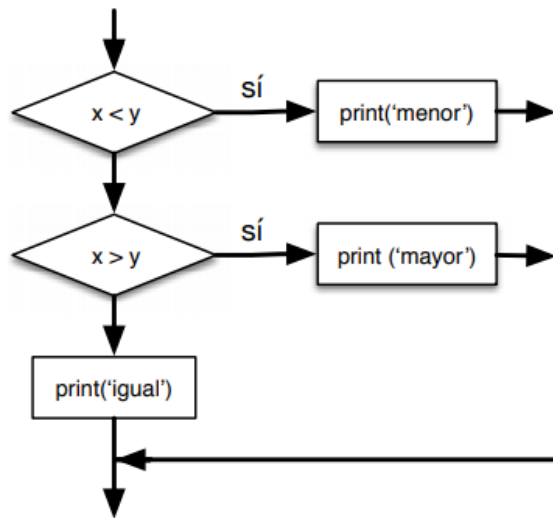
se ejecuta el bloque 2,

Si no se cumplen, ni la condición 1 ni la condición 2,

se ejecuta el bloque 3.



if – elif -else



if $x < y$:

print('x es menor que y')

elif

$x > y$:

print('x es mayor que y')

else:

print('x e y son iguales')



if – elif -else

```
edad = int(input("¿Cuántos años tiene? "))
```

```
if edad >= 18:  
    print("Es usted mayor de edad")
```

```
elif edad < 0:  
    print("No se puede tener una edad negativa")
```

```
else:  
    print("Es usted menor de edad")
```



El futuro digital
es de todos

MinTIC



Preguntas??