



El futuro digital
es de todos

MinTIC

Hechos

QUE

CONECTAN



CICLO 1

EJE TEMÁTICO 1

**FUNDAMENTOS DE
LENGUAJE PYTHON**

Universidad
Industrial de
Santander



Mision
TIC 2022

Fundamentos de lenguaje Python

Ruta de aprendizaje

**Conceptos
básicos de
programación**

**Modelos de
desarrollo de
software**

**Introducción a
Python**

**Tipos de
datos e
instrucciones
básicas**

**Operadores
aritmético
lógicos**

Fundamentos del lenguaje Python

Este recurso pretende dar las bases para interpretar los conceptos básicos de programación de computadores y desarrollo de software e introducir a los tipos de datos e instrucciones básicas del lenguaje de programación Python.

Palabras clave: Computador, Fundamentos, Python, Software.



Introducción

Un lenguaje de programación es un conjunto de instrucciones que son interpretadas por una máquina para producir una serie de salidas o resultados. Pensemos, por ejemplo, en la preparación de un sándwich de jamón con los siguientes ingredientes: dos rodajas de pan, dos rodajas de jamón, una rodaja de queso y 4 torrijas de tomate. Si el proceso para su preparación es llevado a cabo de manera distinta a como debería ser, posiblemente todo resulte en un desastre; en ese sentido, se podría ver al autor de la receta del sándwich como el programador, y a la persona que está usando la receta para preparar el sándwich como la máquina o el computador, que interpreta las instrucciones dadas por el programa (receta) para obtener una salida (un sandwich).

Los lenguajes de programación facilitan el desarrollo de aplicaciones, y a su vez, estas aplicaciones nos facilitan el día a día, como por ejemplo, el manejo de nuestras actividades diarias por medio de diferentes aplicaciones móviles.

Introducción a los conceptos básicos de programación de computadores.

Conceptos básicos

Existen conceptos que son comunes a todos los lenguajes de programación, independientemente del lenguaje de programación seleccionado para aprender; tener estos conceptos claros es de vital importancia, ya que nos ayudará a dominar mejor el lenguaje que se esté aprendiendo, así como también los lenguajes que se deseen aprender en el futuro. Estos conceptos serán explicados a continuación.

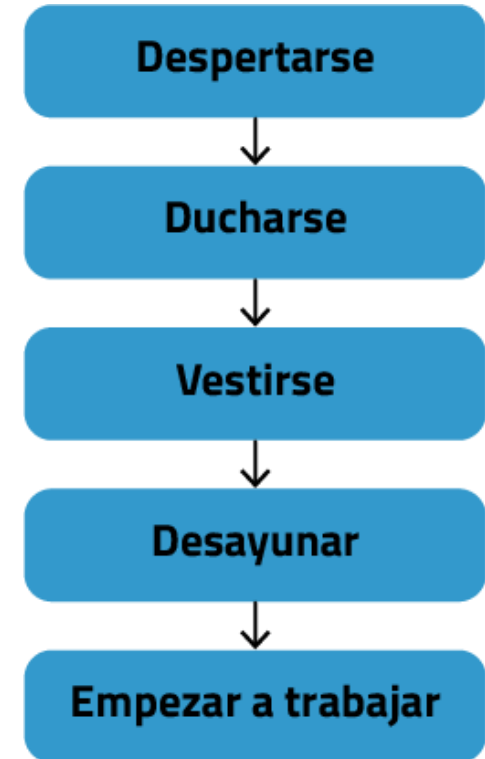


Introducción a los conceptos básicos de programación de computadores.

Conceptos básicos

Algoritmo: es una serie de comandos o instrucciones que deben ser seguidos en un orden específico para solucionar un problema o ejecutar una acción.

Una herramienta que nos ayuda a ver gráficamente esta serie de comandos, pasos o instrucciones, son los diagramas de flujo, en donde cada instrucción es implementada en un recuadro seguido por una flecha que indica el siguiente paso o la siguiente instrucción que debe ser ejecutada.



Introducción a los conceptos básicos de programación de computadores.

Conceptos básicos

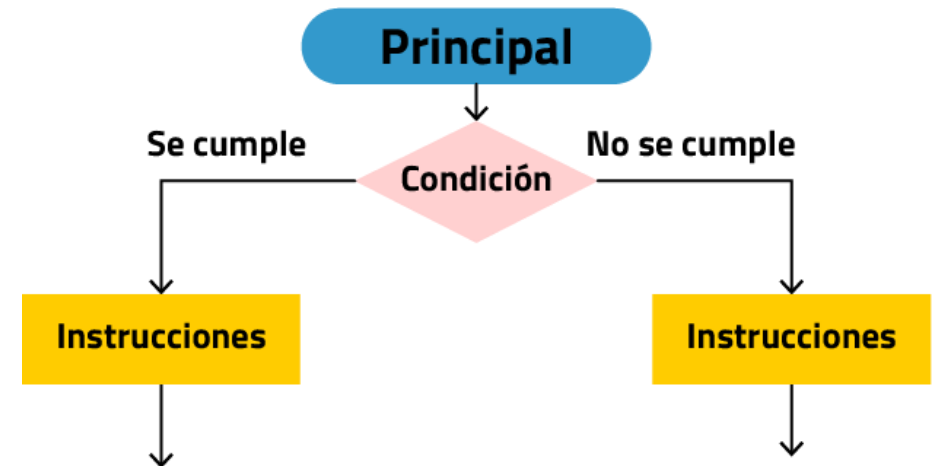
Estructuras condicionales: las estructuras condicionales nos ayudan a incorporar comportamientos que dependen de cierta condición, como por ejemplo, que la temperatura del agua alcance cierto valor para ingresar a la ducha.

En programación es muy recurrente la toma de decisiones y en la mayoría de lenguajes de programación, su sintaxis consta de las palabras claves `if(condition)... then... else`, que traducido al español es, `sí(condición) entonces ... de lo contrario...`

Introducción a los conceptos básicos de programación de computadores.

Conceptos básicos

Estructuras condicionales: en los diagramas de flujo, una estructura condicional se implementa mediante un rombo, en donde se escribe la condición. De este rombo salen dos posibles caminos, uno de ellos se sigue si la condición es verdadera o se cumple, y el otro camino, se sigue si NO se cumple.



Introducción a los conceptos básicos de programación de computadores.

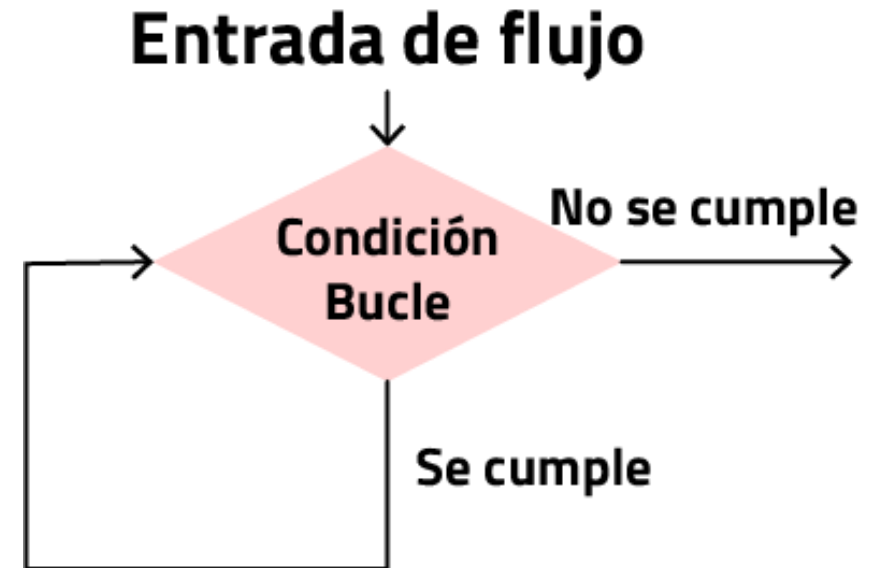
Conceptos básicos

Estructuras de bucle: son las encargadas de realizar tareas o instrucciones de manera recurrente, ya sea un cierto número de veces o hasta que cierta condición se cumpla. Podríamos mejorar nuestro ejemplo de tomar una ducha en una ciudad fría si en lugar de incorporar a nuestro algoritmo una estructura condicional (sí el agua tiene la temperatura adecuada entonces me baño), incorporamos una estructura de bucle (mientras el agua no tenga la temperatura indicada, espere a que caliente).

Introducción a los conceptos básicos de programación de computadores.

Conceptos básicos

Estructuras de bucle: En los diagramas de flujo, una estructura condicional se implementa mediante un rombo en donde se escribe la condición de permanencia, seguido de dos caminos, uno a seguir mientras la condición de permanencia se cumpla y otro a seguir cuando la condición de permanencia se deje de cumplir



Revisión de lo aprendido

Complete los siguientes textos con las palabras **mientras**, **si**, **entonces** y **de lo contrario** según corresponda:

Sume las notas obtenidas en el parcial 1, 2 y 3 del estudiante, luego divida la suma entre tres para obtener el promedio. _____ la nota es menor a 3, _____ el estudiante reprueba la materia, _____ el estudiante aprueba la materia.

_____ la persona es hombre, _____ deberá cotizar 1000 semanas de pensión _____ sea menor a 62 años, _____ es mujer deberá cotizar 1000 semanas de pensión _____ sea menor a 57 años

_____ la temperatura no sea la indicada, la caldera se mantendrá encendida. Tenga precaución _____ la temperatura supera los 120 °C, _____ deberá oprimir parada de emergencia, _____ deberá estar atento al tablero de control

Introducción a los conceptos básicos de programación de computadores.

Conceptos básicos

Composición de un sistema computacional:

Las computadoras o sistemas computacionales se pueden definir como **procesadores de datos**. Algunos ejemplos de sistemas computacionales son: las calculadoras, los celulares y los relojes inteligentes. De acuerdo al propósito para los que fueron creados, los sistemas computacionales pueden ser de **propósito específico**, es decir, diseñados para una tarea específica como monitorear los niveles de agua de un río, o de **propósito general**, los cuales pueden ser programados para una tarea deseada, por ejemplo, los computadores de escritorio.

Introducción a los conceptos básicos de programación de computadores.

Conceptos básicos

Composición de un sistema computacional:

Los sistemas computacionales se encuentran conformados por un conjunto de elementos funcionales que se interrelacionan entre sí. Estos elementos son:

- **El hardware:** las partes físicas (teclado, cámara, unidades de almacenamiento, etc.)
- **El sistema operativo:** el intermediario entre el hardware y las aplicaciones (Windows, Linux, iOS)
- **Las aplicaciones:** es el software diseñado a partir de un conjunto de algoritmos escritos en un determinado lenguaje de programación y utilizados por el usuario final.
- **Los usuarios:** son las personas que finalmente usan el software desarrollado o la aplicación.

Revisión de lo aprendido

Para cada uno de los elementos, indique si corresponde al hardware, al sistema operativo o a las aplicaciones.

- Teclado _____
- Microsoft Word _____
- Driver para la impresora _____
- Mac OS _____
- Mousse _____
- Monitor _____
- Google Chrome _____
- Linux _____
- Microsoft Excel _____

Introducción a los conceptos básicos de programación de computadores.

Modelos de desarrollo de software

Ciclo de vida de desarrollo de software

(SDLC): Un ciclo de vida de desarrollo de software (SDLC) incluye los procesos de software utilizados para especificar y transformar los requisitos de software en un producto de software entregable. Está dividido en 4 categorías como se muestra en la imagen.



1. Procesos primarios

Incluye los procesos de análisis, diseño, desarrollo, operación y mantenimiento del *Software*.

2. Procesos de soporte

Son aplicados de forma intermitente o continua a lo largo del ciclo de vida de un producto *software*, para respaldar los procesos primarios, incluyendo procesos de gestión de la configuración, aseguramiento de calidad, verificación y validación.

Ciclo de Vida del Software

3. Procesos organizacionales

Proporcionan soporte en el proceso de ingeniería de *software*, incluyendo, capacitación, análisis de medición de proceso, gestión de infraestructura, reutilización, mejora de procesos organizacionales y gestión de modelos de ciclo de vida de *software*.

4. Procesos de proyectos cruzados

Como reutilización, línea de productos de *software*; pueden implicar más de un proyecto de *software* en una organización.

Introducción a los conceptos básicos de programación de computadores.

Modelos de desarrollo de software

Fases de desarrollo de software: Las fases en el desarrollo del software varían en la forma como son aplicadas según el modelo de ciclo de vida de software utilizado, pero de manera general, se implementan las siguientes fases:



Introducción a los conceptos básicos de programación de computadores.

Modelos de desarrollo de software

Modelos Básicos de ciclos de vida del software: La naturaleza intangible y maleable del software permite una amplia variedad de modelos de ciclo de vida de desarrollo de software que van desde modelos lineales, en los que las fases del desarrollo de software se logran secuencialmente, a modelos iterativos, en los que el software se desarrolla en incrementos de funcionalidad creciente en ciclos iterativos. A continuación, se muestran gráficamente estos modelos:

Introducción a los conceptos básicos de programación de computadores.

Modelos de desarrollo de software

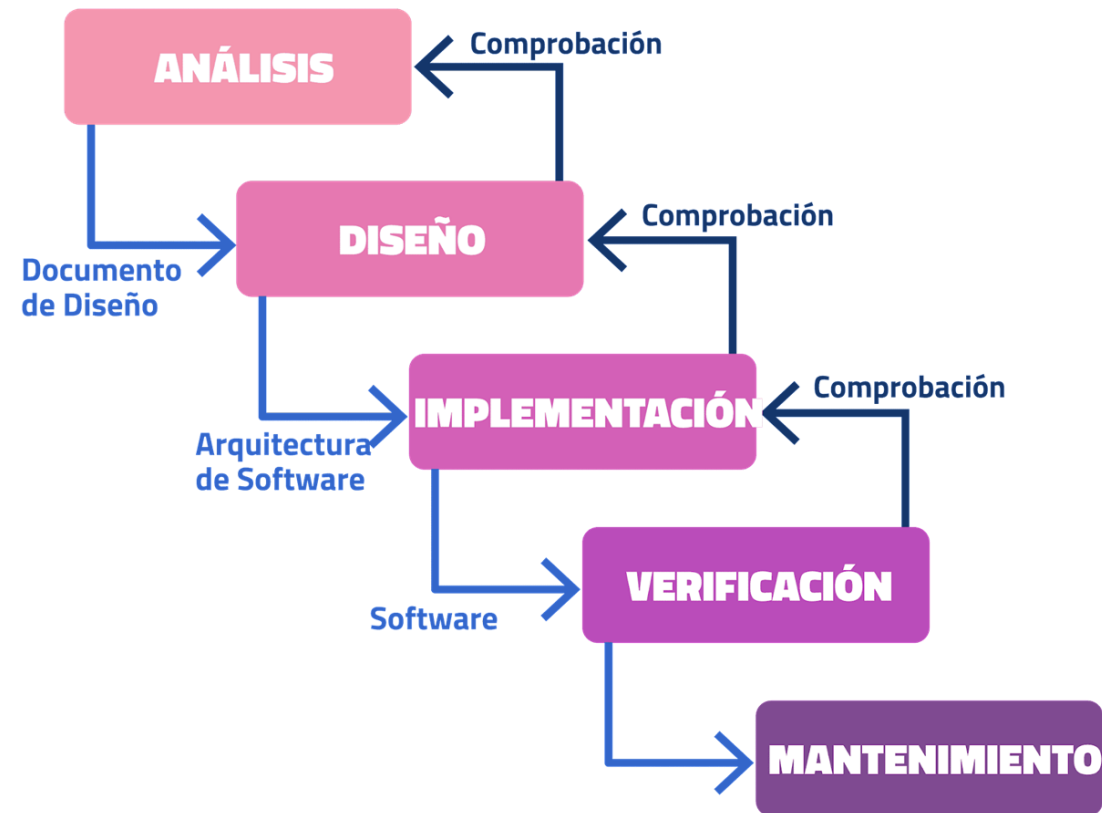
Ciclo en V



Introducción a los conceptos básicos de programación de computadores.

Modelos de desarrollo de software

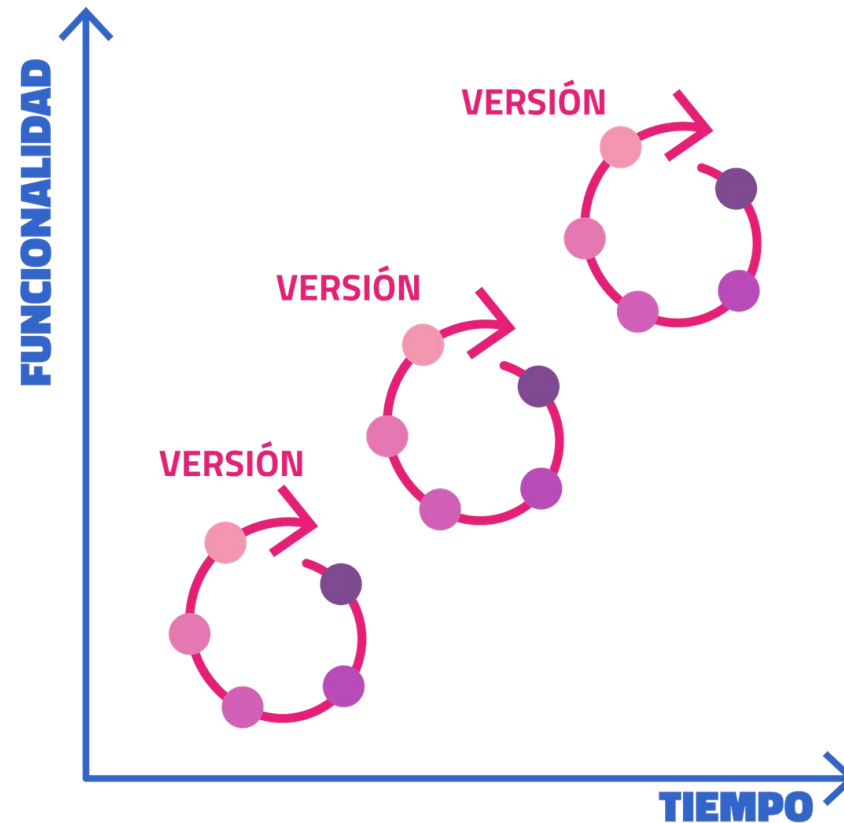
Ciclo en cascada



Introducción a los conceptos básicos de programación de computadores.

Modelos de desarrollo de software

Ciclo iterativo



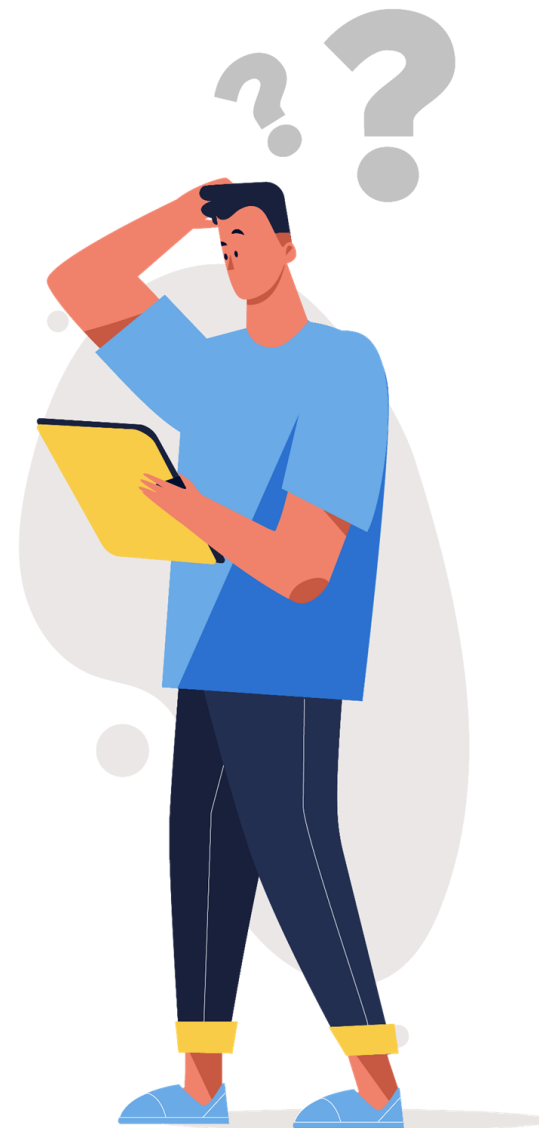
Revisión de lo aprendido

Selección múltiple con única respuesta:

1. Los procesos de análisis, diseño, desarrollo, operación y mantenimiento del Software son denominados:

Opciones de respuesta

- a. Procesos Primarios
- b. Procesos de soporte
- c. Procesos Organizacionales
- d. Procesos de proyectos cruzados

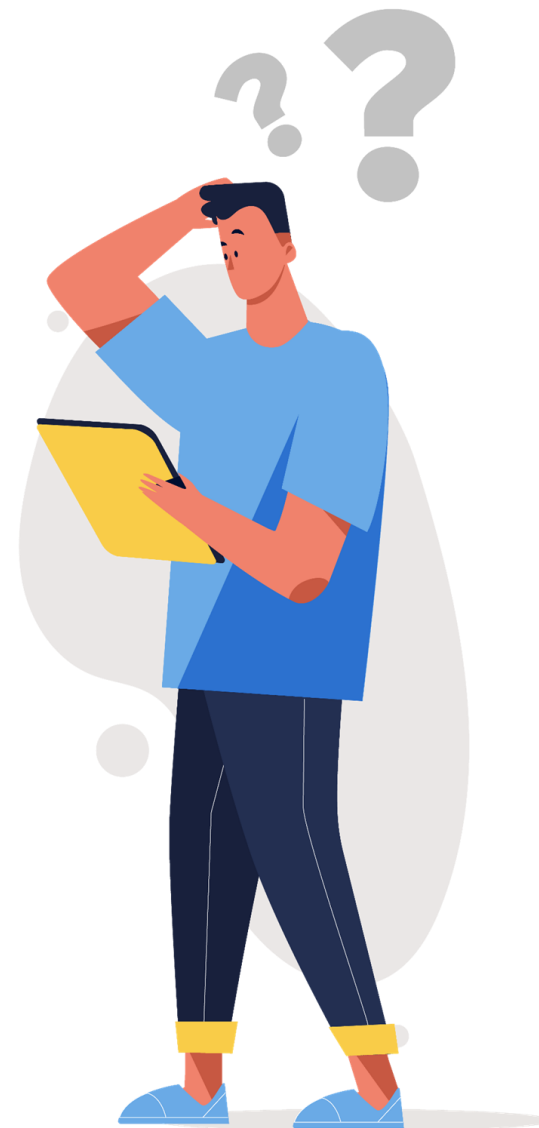


Revisión de lo aprendido

2. Cuando las fases del desarrollo de software se logran de manera secuencial con retroalimentación y entrega de un producto único, se suelen usar los modelos:

Opciones de respuesta

- a. Lineales
- b. Iterativos
- c. Organizacionales
- d. Misionales



Revisión de lo aprendido

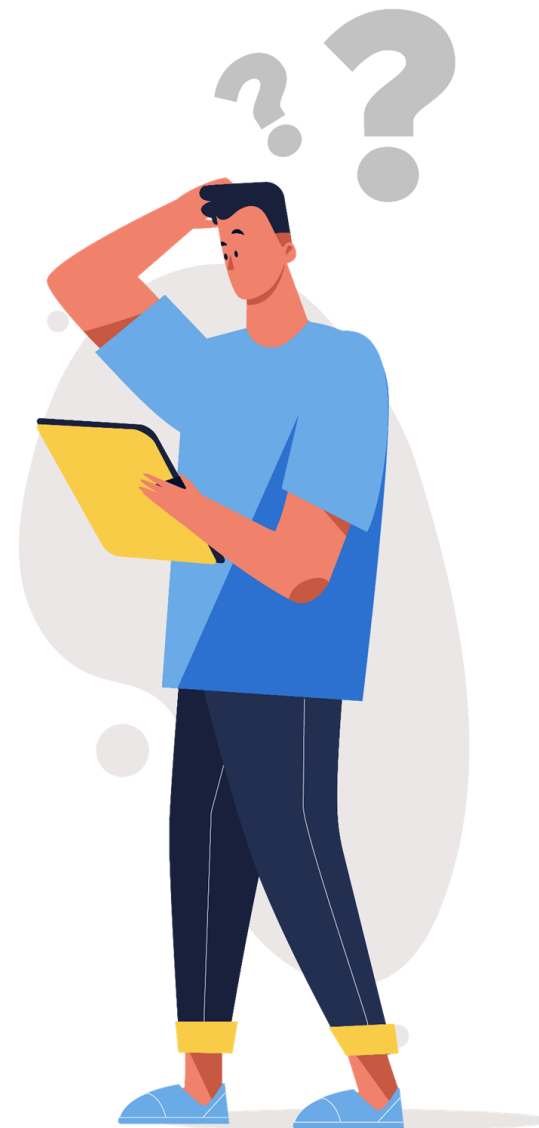
3. Cuando las fases del desarrollo de software se realizan con incrementos de funcionalidad creciente en ciclos iterativos o ágiles, se suelen usar los modelos:

Opciones de respuesta

a.Lineales

b.Iterativos

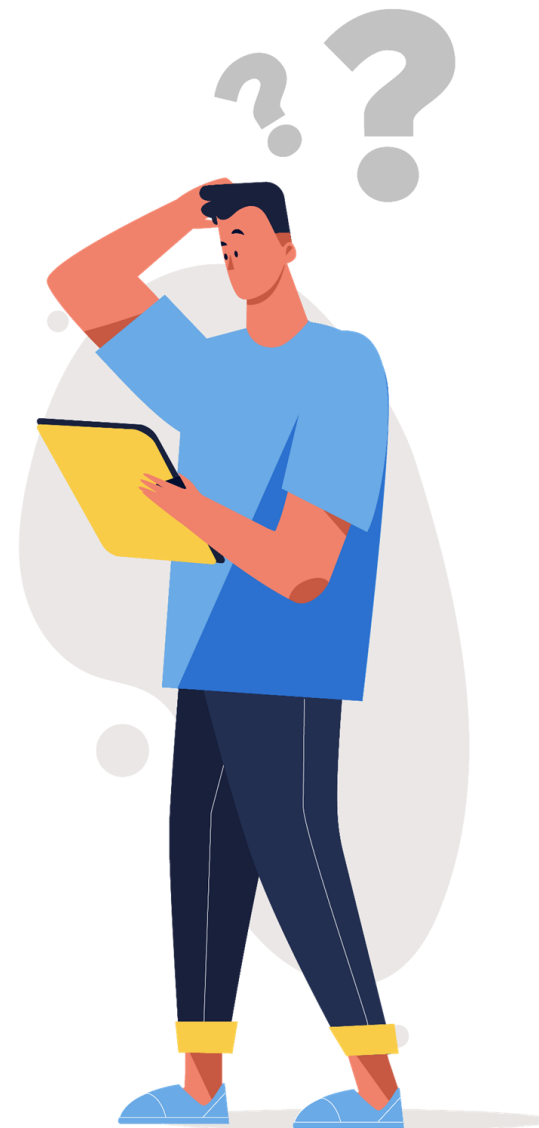
c.Organizacionales



Revisión de lo aprendido

4. Debido a la naturaleza intangible y maleable del Software, para su desarrollo existe una gran variedad de:
Opciones de respuesta

- a. Usuarios
- b. Modelos de ciclo de vida
- c. Mantenimientos
- d. Conjuntos



Introducción a los conceptos básicos de programación de computadores.

Introducción al lenguaje de programación Python

Python es un lenguaje de programación gratuito, cuya principal característica es su sintaxis simple y fácil de aprender, esto ha hecho que Python tenga una buena comunidad de programadores alrededor del mundo. El hecho de que sea rápido realizar un prototipo de software en este lenguaje de programación, permite a las empresas un importante ahorro de recursos tanto de tiempo como de dinero para poder lanzar productos al mercado ágilmente, compatible con diferentes plataformas y sistemas operativos.

Introducción a los conceptos básicos de programación de computadores.

Introducción al lenguaje de programación Python

En palabras cortas, un entorno de desarrollo integrado (**IDE**), es un programa que ofrece un entorno con múltiples herramientas que le ayudan al desarrollador a programar. Puede estar enfocado a un solo lenguaje de programación o a varios. Específicamente, un IDE está compuesto por un editor de texto, un compilador, un intérprete y un depurador.

Entre los IDE más comunes para hacer desarrollo en el lenguaje de programación Python se encuentran:

Pycharm: jetbrains.com/pycharm

IDLE: Viene por defecto al descargar Python.

Thonny: <https://thonny.org/>.

Visual Studio Code: <https://code.visualstudio.com/>.

Spyder: <https://www.spyder-ide.org/>

Introducción a los conceptos básicos de programación de computadores.

Introducción al lenguaje de programación Python

Al crear código en algún lenguaje de programación, es necesario que pase por un proceso para ser ejecutado, ya sea en la misma máquina o en alguna otra plataforma, por ejemplo, un navegador web. A esto se le llama comúnmente “traducir el programa” y es lo que nos permite hacer los intérpretes.

Si estás en algún computador que no tiene Python instalado y necesitas modificar algún programa de urgencia, podrás modificar tu programa desde el navegador web del computador; para ello, es importante que conozcas algunos intérpretes en línea.

A continuación, se mostrarán los más usados.

Programiz: <https://www.programiz.com/python-programming/online-compiler/>.

Paiza.io: <https://paiza.io/es>

Online GDB: https://www.onlinegdb.com/online_python_interpreter.

Revisión de lo aprendido

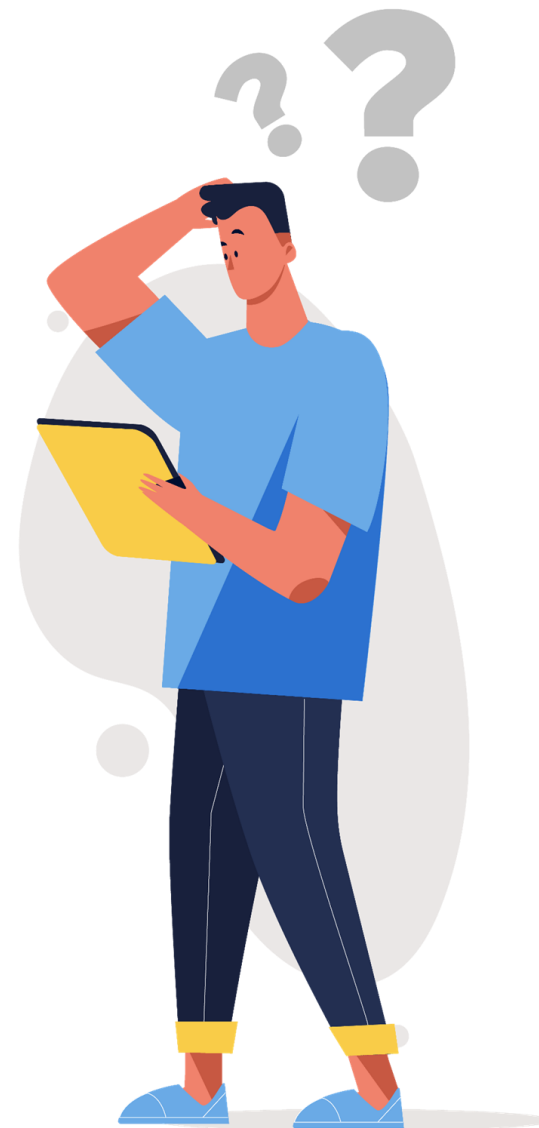
¿Qué es un IDE?

- A. Un intérprete capaz de traducir lo codificado en lenguaje de máquina.
- B. Un programa que nos ayuda a escribir código.
- C. Un entorno de desarrollo integrado que ayuda al programador a realizar todo el desarrollo de su programa.
- D. Un entorno de desarrollo integrado que ayuda al programador a identificar errores en la escritura del programa.

Revisión de lo aprendido

El IDE que viene instalado por defecto al instalar Python es:

- A. IDLE
- B. Spyder
- C. Visual Studio Code
- D. Atom



Introducción a los conceptos básicos de programación de computadores.

Tipos de datos e instrucciones básicas

Tipos de datos y declaración de variables: En informática, un dato es una unidad de información eficiente para mover o procesar. Si el valor de los datos cambian con el tiempo, estos datos son almacenados en variables; por el contrario, si no cambian de valor con el tiempo, son almacenados en constantes. Un ejemplo de un dato variable puede ser la temperatura ambiente, y de un dato constante, puede ser la talla de una camiseta.

En Python, la sintaxis para declarar una variable es:

nombre_de_variable = valor

Es importante tener en cuenta que para el nombre de variables, NO se pueden usar:

- Espacios en blanco.
- Caracteres especiales como \$ %&/()=?¿!,
- Comenzar una definición de variables con números.
- Usar palabras reservadas, es decir palabras que hagan parte de la sintaxis de Python como if, else, while, entre otras.

Introducción a los conceptos básicos de programación de computadores.

Tipos de datos e instrucciones básicas

Tipos de datos fundamentales en Python: Los tipos de datos básicos de Python son los numéricos (enteros, punto flotante y complejos), las cadenas de caracteres y los booleanos. En la siguiente tabla se describen los tipos de datos fundamentales de Python:



CATEGORÍA	TIPO DE DATO	CARACTERÍSTICA	EJEMPLO DE ASIGNACIÓN EN PYTHON
Numéricos	Entero (int)	No tienen parte decimal y pueden ser negativos o positivos	x=-4
	Flotantes (float)	Tienen parte decimal y pueden ser positivos o negativos	x=3.5502
	Complejos (complex)	Tienen una parte real y una parte imaginaria	X= 1.51 + 6j
Cadenas de texto	Cadenas de texto (string)	Se pueden conformar de diferentes tipos de caracteres.	cadena1 = 'comillas simples' cadena2 = "comillas dobles"
Booleanos	Booleanos (boolean)	Solo puede tener un valor de Verdadero o Falso.	aT = True aF = False
Colecciones de datos	Conjuntos (set)	Colección de datos sin elementos que se repiten	pla ={'pastelito', 'jamon', 'papa', 'empanadilla'}
	Listas (list)	Colección de datos, los elementos pueden repetirse.	b = ['2.36', 'elefante', 1010, 'rojo']
	Tuplas (tuple)	Lista que no se puede modificar después de su creación.	tupla = (1234, 5678, 'Hola Mundo Python!')
	Diccionarios (Dictionary)	Define los datos uno a uno entre un campo y un valor	datos_basicos = { "nombres": "Pedro", "apellidos": "Perez Pinto", "numero": "852963" }

Introducción a los conceptos básicos de programación de computadores.

Tipos de datos e instrucciones básicas

- **Entrada y salida estándar:** Para pedir información al usuario, debe utilizar las funciones integradas en el intérprete del lenguaje, así como los argumentos de línea de comandos.

raw_input()

La forma general de mostrar información por pantalla es mediante una consola de comando; generalmente, podemos mostrar texto y variables separándolos con comas, para este se usa la sentencia

print()



Introducción a los conceptos básicos de programación de computadores.

Tipos de datos e instrucciones básicas

- **Transformación de datos (casting):** Antes de todo, es importante tener en cuenta que al momento de asignar un valor a una variable, Python implícitamente asigna el tipo a la variable de acuerdo al tipo de dato que almacena.

Python nos ofrece la opción de explícitamente decir qué tipo de variable queremos, a esto se le llama transformación de tipo de dato o también casting o casteo.

Python incluye una gran variedad de funciones nativas para realizar distintas operaciones. Las más comunes son: `float()` , `str()`, `int()`, `list()`, `set()` y otras no tan comunes como: `hex()`, `oct()`, `bin()`.

Introducción a los conceptos básicos de programación de computadores.

Tipos de datos e instrucciones básicas

Ejemplo:

Convertir a cadena de texto (string): `str()`

```
anios = 56
print = ("Edad: " + str(anios))
```

Convertir a entero: `int()`

```
Cant = "26"
costo = 1500 * int(Cant)
print (costo)
```

Introducción a los conceptos básicos de programación de computadores.

Operadores aritméticos lógicos

- **Operadores aritméticos:** estos son la base de todo cálculo matemático que se pretenda realizar en todo lenguaje de programación, es decir, Python no es la excepción. La siguiente tabla relaciona los operadores aritméticos y su implementación en Python:



OPERACIÓN	DESCRIPCIÓN	EJEMPLO
Suma (+)	Consiste en hacer una adición a un número o una variable. Para realizar una suma en Python, basta con poner un número o una variable, seguido del símbolo suma (+) y otro número o variable	manzanas=5 print(manzanas+5)
Resta (-)	La resta, NO es una operación conmutativa, es decir, el orden en el que se escriba la operación SÍ importa. Se implementa con el operador (-)	a=1 b=10 print(a-b)
Multiplicación (*)	Este operador es conmutativo y se implementa con el símbolo asterisco (*)	print(5*5)
División (/)	Este operador no es conmutativo y siempre retorna un valor flotante. Se implementa con /.	print(11/2)
División entera (//)	Retorna la parte entera del resultado de una división, es decir sin su parte decimal. Se implementa con //	print(11//2)
Potenciación (**)	Multiplica la base, por sí mismo, las veces que indique el exponente. La base estará al lado izquierdo del operador (**) y el exponente al lado derecho.	print(3**2)
Módulo (%)	Devuelve el residuo, o también llamado resto, de una división.	print(5%2)

Introducción a los conceptos básicos de programación de computadores.

Operadores aritméticos lógicos

- **Operadores de comparación:** Sirven para comparar entre al menos un par de valores y siempre devuelven un valor booleano. La sintaxis es la misma que los operadores aritméticos, se escribe un valor, número o variable a la derecha, luego el símbolo del operador, y por último, el otro valor, número o variable.

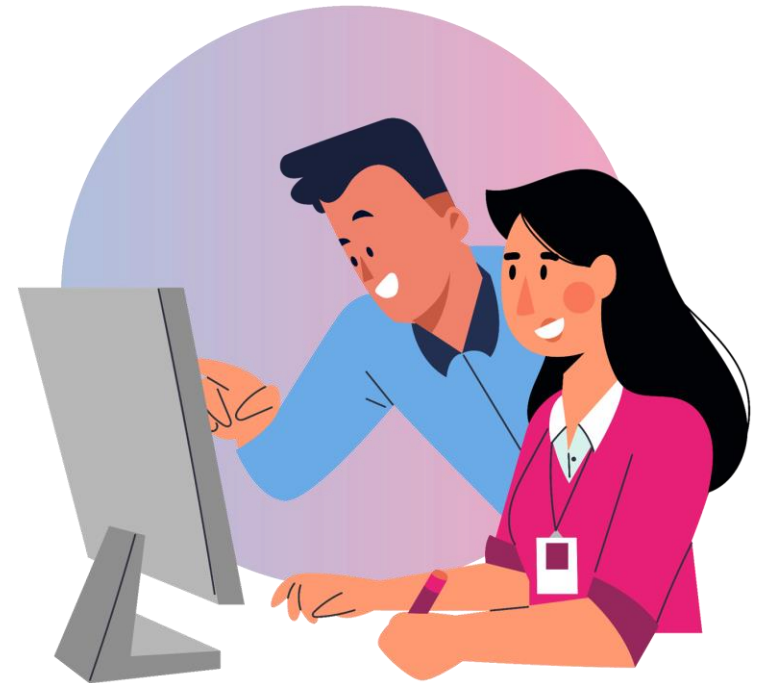


OPERACIÓN	DESCRIPCIÓN	EJEMPLO
Igualdad (==)	Compara si ambos valores dados para el operador son iguales, en caso de serlo, devolverá True, en caso contrario, devolverá False.	x=5 print(x==5)
Diferencia (!=)	Lo opuesto a la igualdad. Si los valores son diferentes devolverá True, de lo contrario devolverá False.	x=5 y=5 print(x!=y)
Mayor que (>)	Este operador devuelve True si el número de la izquierda es mayor que el de la derecha, False en caso contrario.	print(20>5)
Menor que (<)	Devolverá True si el número de la izquierda es menor al de la derecha.	print(20<5)
Mayor o igual que (>=)	Similar al mayor que, la única diferencia es que también devolverá True si ambos números son iguales.	x=5 print(x>=5)
Menor o igual que (<=)	Es similar al menor que, con la diferencia de que devolverá True si ambos números son iguales	print(3<=5)

Introducción a los conceptos básicos de programación de computadores.

Operadores aritméticos lógicos

- **Operadores lógicos:** Python incluye tres operadores lógicos base: el AND, OR y NOT.
Es importante tener en cuenta que, los operadores lógicos funcionan solo con valores o variables booleanas, a su vez, devuelve como respuesta valores booleanos.



OPERACIÓN	DESCRIPCIÓN	EJEMPLO
AND	En español significa "y", solo devolverá True si ambos valores, el de la izquierda y el de la derecha, son verdaderos. Para usar este operador basta con escribir un valor booleano, luego la palabra reservada "and" en minúscula y por último otro valor booleano.	print(True and False)
OR	En español significa "o", es un operador que, al contrario del AND, devuelve a todo verdadero a menos que ambos valores, el de la izquierda y la derecha del operador.	x=5 y=5 print (x>y or x==5)
NOT	Que en español significa "no", lo que hace es negar, o más específicamente, invertir el valor booleano que viene después de él.	print(not True)

- Tabla de la verdad: **AND**

INPUT		OUTPUT
A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

- Tabla de la verdad: **OR**

INPUT		OUTPUT
A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

- Tabla de la verdad: **NOT**

INPUT	OUTPUT
A	NOT A
0	1
1	0

En las tablas anteriores el número **1** representa el **True** y el número **0** representa el **False**.

Introducción a los conceptos básicos de programación de computadores.

Operadores aritméticos lógicos

- **Precedencia matemática PEMDAS:** especifica el orden en que son realizadas las operaciones matemáticas en una expresión donde se tiene más de un operador. El orden es:
 1. **Paréntesis**
 2. **Exponentes** (o raíces)
 3. **Multiplicación y división**
 4. **Adición y Sustracción** (sumas y restas)

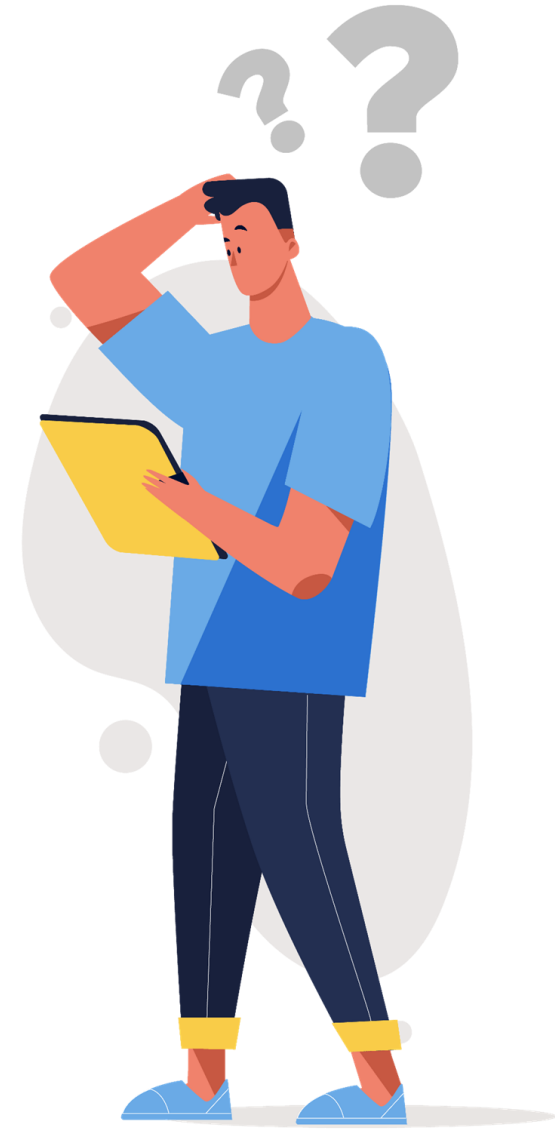
Ese orden se puede conocer de forma muy sencilla, acordándose del siguiente acrónimo: PEMDAS, donde cada letra del acrónimo hace referencia al orden de las operaciones, en el orden respectivo. Ahora que conoces al PEMDAS.

Python sigue muy estrictamente dicho orden, con un pequeño agregado, incluso al no hacer operaciones matemáticas, podemos hacer que Python ejecute otros operadores, primero con respecto a otros, si encerramos dicha operación dentro de un paréntesis.

Revisión de lo aprendido

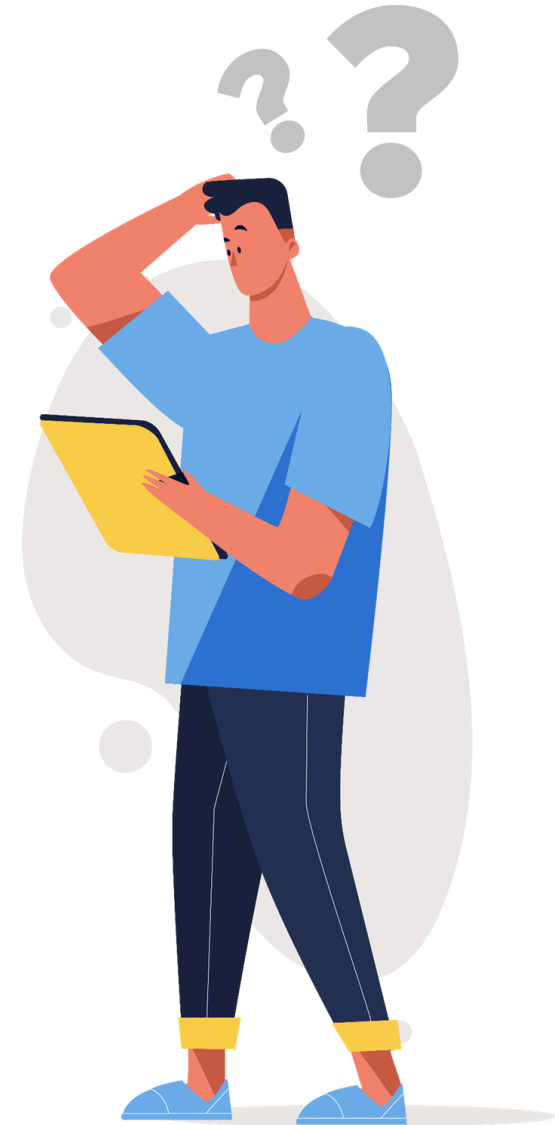
1. Coloque al frente de cada variable su tipo, según corresponda.

- i. $x = 1+3$: _____
- ii. nombre = "Jairo" : _____
- iii. $\pi = 3.141596$: _____
- iv. fuerza = "18" : _____



Revisión de lo aprendido

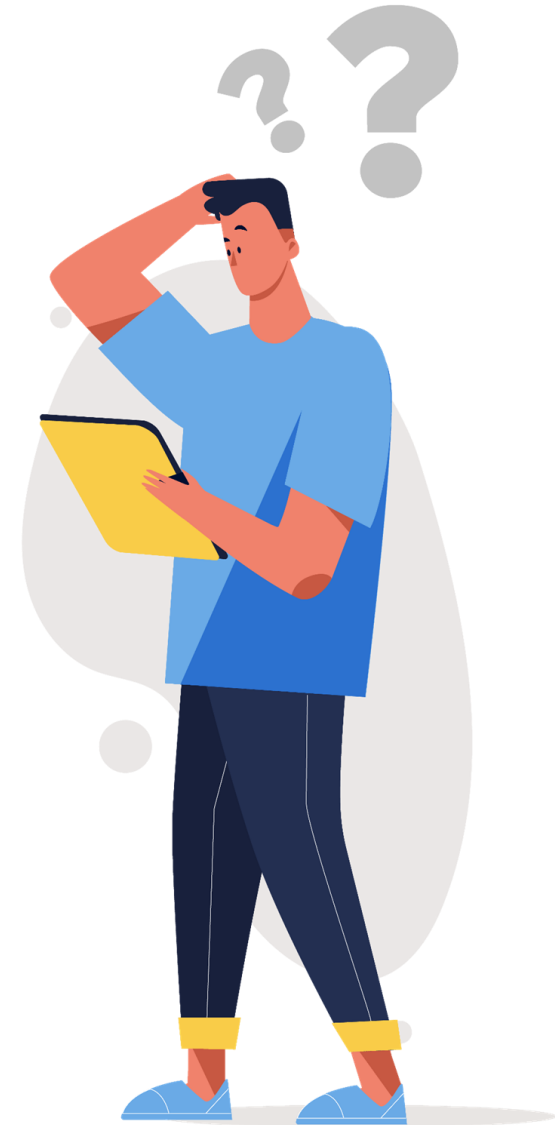
2. Es la siguiente variable un string. `magnitud = "3.45"`
- a. Verdadero
 - b. Falso



Revisión de lo aprendido

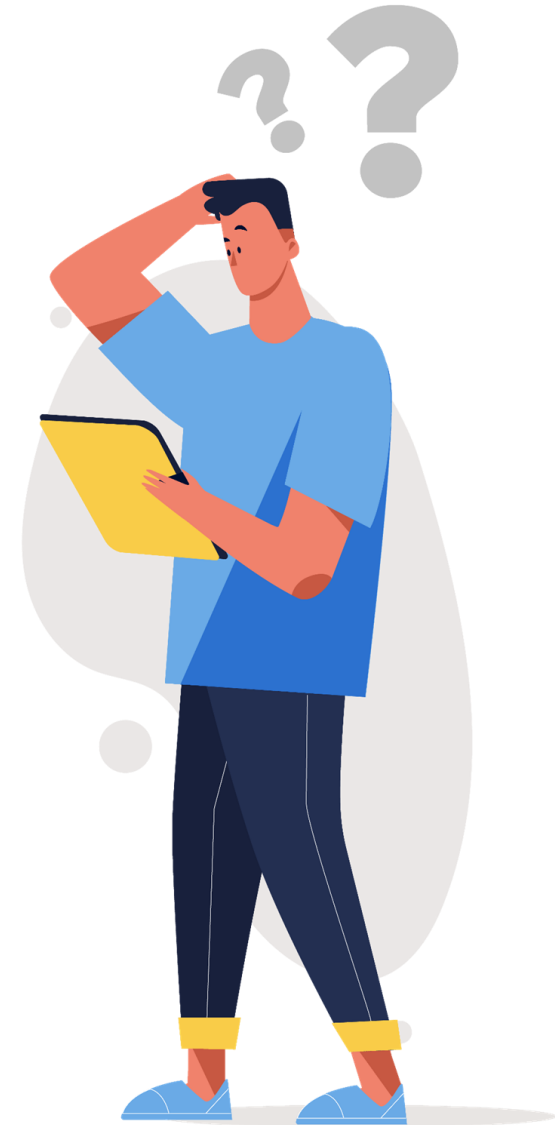
3. ¿Cuál es el valor final de la siguiente operación?
 $\text{Ver_1} = "35" + 8$

- a. error
- b. 358
- c. 43



Revisión de lo aprendido

4. Para **asignar un valor cadena a una variable**, se debe colocar entre comillas simples
- a. Verdadero
 - b. Falso

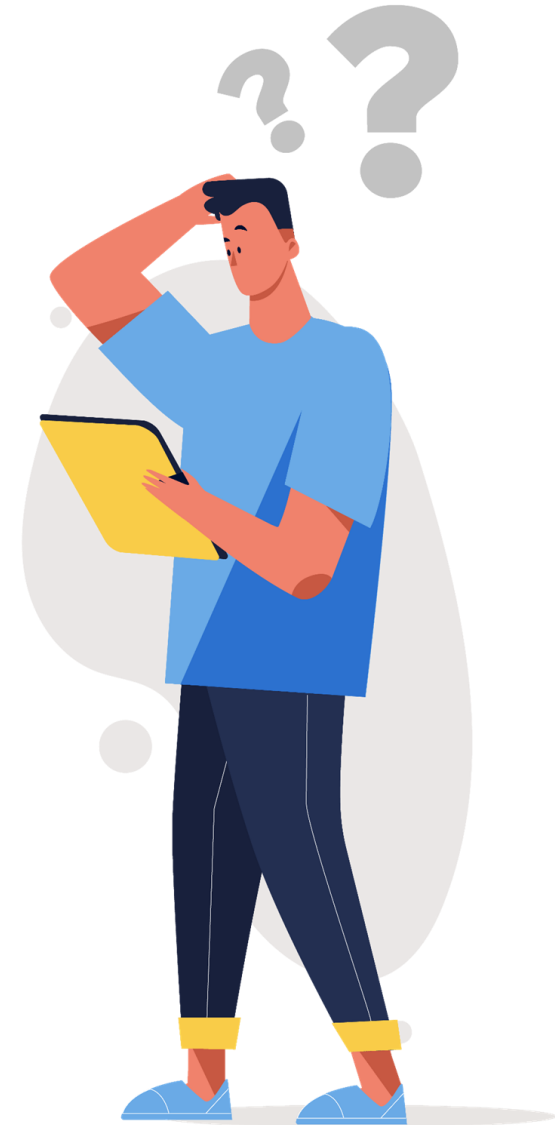


Revisión de lo aprendido

5. Este es el resultado de la siguiente operación

$$35 == 35.0$$

- a. Verdadero
- b. Falso





El futuro digital
es de todos

MinTIC

Hechos

QUE

CONECTAN



CICLO 1

EJE TEMÁTICO 1

**FUNDAMENTOS DE
LENGUAJE PYTHON**

Universidad
Industrial de
Santander



Mision
TIC2022