



El futuro digital
es de todos

MinTIC

Introducción A la Programación

»
Misión TIC 2022

xxx



Misión
TIC 2022



El futuro digital
es de todos

MinTIC



x
x
x

Fundamentos de Programación - Python

Temas

- Presentación Formador
- Presentación Misión TIC – Ciclo 1
- Pensamiento lógico estructurado
- Variables – Constantes – Operadores aritméticos
- Pseudocódigo - Algoritmos
- Diagramas de Flujo



Presentación Ciclo 1 – Misión TIC



El futuro digital
es de todos

MinTIC



Presentación Formador



Presentación Ciclo 1 – Misión TIC



El futuro digital
es de todos

MinTIC



Presentación Misión TIC – Ciclo 1



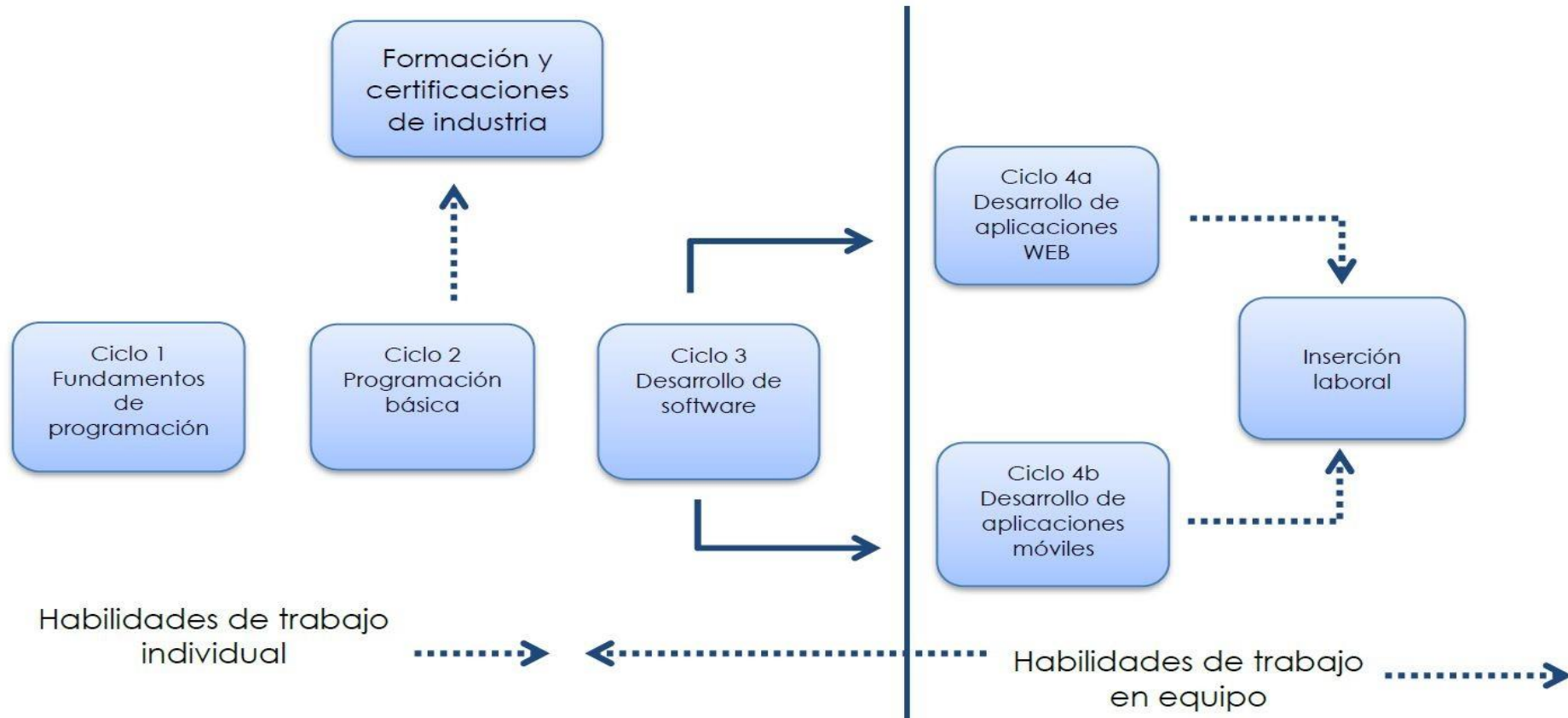
Presentación Ciclo 1 – Misión TIC



El futuro digital
es de todos

MinTIC

» Estructura Misión TIC 2022





Ciclo 1: Fundamentos de Programación en Python

Presentación Ciclo 1 – Misión TIC



El futuro digital
es de todos

MinTIC

» Perfil de Egreso



El tripulante que haya culminado con éxito este ciclo estará en la capacidad de:

- Desarrollar un programa monousuario para resolver los requerimientos planteados por un tercero
- Construir un programa trabajando de manera individual
- Construir un programa utilizando un lenguaje imperativo (Python)
- Construir un programa siguiendo el ciclo completo de vida de desarrollo, que comienza con la identificación y documentación de los requerimientos funcionales y termina con un conjunto de pruebas unitarias.
- Construir un programa con una interfaz de consola
- Construir un programa que maneje estructuras de datos lineales en memoria principal
- Construir un programa que maneje archivos de texto para almacenar información persistente (formato JSON / CSV)

Presentación Ciclo 1 – Misión TIC



El futuro digital
es de todos

MinTIC

» Estructura Ciclo 1 – Fundamentos de Programación – Ejes temáticos

X
X
X

Semana	Unidad de aprendizaje	Contenido	Tipo	Actividad de aprendizaje
1	Introducción a la Programación	Conceptualización e introducción al curso	Sesiones de videoconferencia (TEAMS)	Reto 1
		Algoritmos, pensamiento lógico y estructurado		
		Instalación y configuración de Python 3, Pseint y Pycharm, hola mundo en IDLE y Pseint		
2	Formalización y Diagramación de Soluciones	Variables	Sesiones de videoconferencia (TEAMS)	Reto 2
	Estructuras de Control	Entrada, Salida y Operadores Aritméticos		
		Condicionales		
		Ciclos		
3	Estructuras de Datos	Validaciones y captura de excepciones	Sesiones de videoconferencia (TEAMS)	Reto 3
		Listas		
		Listas: Ciclos		
		Listas: Operadores Especiales		
4		Listas dentro de listas	Sesiones de videoconferencia (TEAMS)	Reto 4
		Tuplas y Diccionarios		
		Funciones y Métodos		
5	Ordenamiento y Búsqueda	Métodos de Ordenamiento Iterativo	Sesiones de videoconferencia (TEAMS)	
		Métodos de Búsqueda Iterativo		
		Recursividad		
6		Métodos y Ordenamiento Recursivo	Sesiones de videoconferencia (TEAMS)	Reto 5
7	Archivos y JSON	Lectura y escritura de archivos	Sesiones de videoconferencia (TEAMS).	
		JSON		
		Finalizar Reto 5		



El futuro digital
es de todos

MinTIC

» Tecnologías Empleadas - Instalación

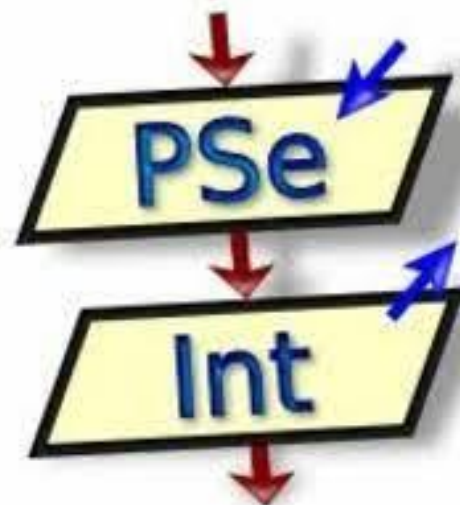


PYTHON

<https://www.python.org/downloads/>



Visual Studio Code



<http://pseint.sourceforge.net/?page=descargas.php>



PyCharm



Pensamiento **lógico** estructurado

Estructurado: Manejo de estructuras de control (lógicas -mentales) – Secuencia, **condicional**, ciclo

Lógico: Secuencia lógica de pasos, existe un proceso1, luego proceso 2 , luego proceso 3 (Orden)



Variables

Constante: Algo que no cambia, que es fijo

35 grados

3

“Sergio Medina”

Variable: Algo que cambia

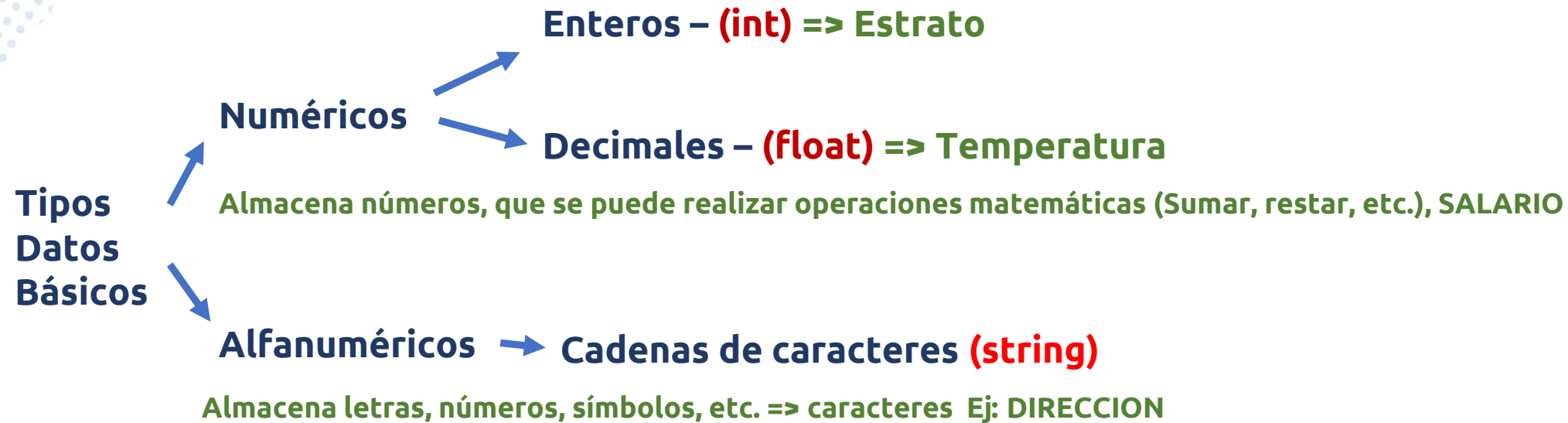
Temperatura

Estrato

Nombre_persona



Variables





Práctica Variables

```
Python 3.9.2 (tags/v3.9.2:13f6953, Oct 7 2021)
Type "help", "copyright", "credits() or "license()" for more
>>> numerol=10
>>> numerol
10
>>> type(numerol)
<class 'int'>
>>> numero2=34.5
>>> numero2
34.5
>>> type(numero2)
<class 'float'>
>>> nombre="Sergio"
>>> nombre
'Sergio'
>>> type(nombre)
<class 'str'>
>>>
```




Operadores aritméticos

Fundamentos de Programación - Python



El futuro digital
es de todos

MinTIC



x
x
x

Python



Operadores aritméticos - Jerarquía (Solo mencionarlos para utilizarlos)

Operador	Nombre	Nivel Jerárquico
()	Paréntesis	1er nivel
**	Potencia	2do nivel
*	Multiplicación	3er nivel
/ //	División	
+	Suma	4to nivel
-	Resta	



De izquierda a derecha

%

Módulo (Residuo)



Práctica Operadores aritméticos

```
Python Desactivar audio Detener video Seguridad
Type "help", "copyright",
>>> 7/2
3.5
>>> 7//2
3
>>> 7%2
1
>>> 7+(5*2)
17
>>> 7+5*2
17
>>> 5+10/2-4*3
-2.0
>>> n1=40
>>> n1
40
>>> n2=15
>>> type(n1)
<class 'int'>
>>> type(n2)
<class 'int'>
>>> n3=n1/n2
>>> n3
2.6666666666666665
>>> type(n3)
<class 'float'>
>>>
```



Práctica Operadores aritméticos

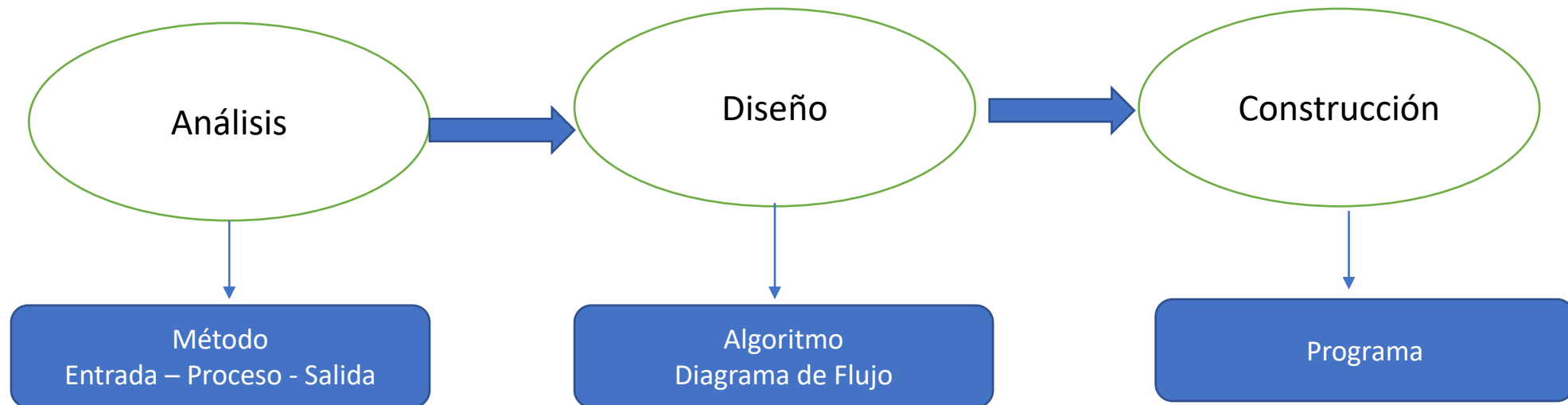
```
x Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021) [AMD64] on win32
x Type "help", "copyright", "credits" or "license()" for more
x >>> n1="5"
>>> type(n1)
<class 'str'>
>>> n1=int(n1)
>>> n1
5
>>> type(n1)
<class 'int'>
>>> n2=34.7
>>> type(n2)
<class 'float'>
>>> n2=int(n2)
>>> n2
34
>>> n5="Juan"
>>> n5=int(n5)
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    n5=int(n5)
ValueError: invalid literal for int() with base 10: 'Juan'
>>>
```



Ejercicios



Metodología -> Pensamiento lógico estructurado





2.2. ALGORITMOS

Se define como la solución en lenguaje natural a un problema de la vida cotidiana y para ello se utiliza una serie de órdenes ó acciones, en un orden lógico y cronológico.

Un algoritmo se puede representar mediante:

- ✓ **Diagrama de Flujo**: Es una representación gráfica de la solución a un problema haciendo uso de símbolos y expresiones.
- ✓ **Pseudocódigo**: Las órdenes del algoritmo son expresadas mediante un lenguaje natural. Esta solución, no es gráfica, sino textual y persigue visualizar en comandos la secuencia lógica con la cual se soluciona el problema al usuario

Ejercicios



Fundamentos de Programación - Python



El futuro digital
es de todos

MinTIC



Programa: Solución final o definitiva a un problema de manejo de información
Conjunto o secuencia lógica de **instrucciones (orden, comando)** para resolver un problema de manejo de información

Lenguaje de Programación: Conjunto de instrucciones o comandos => Comunicación con el computador => Solución problema

Lenguaje de Programación Python



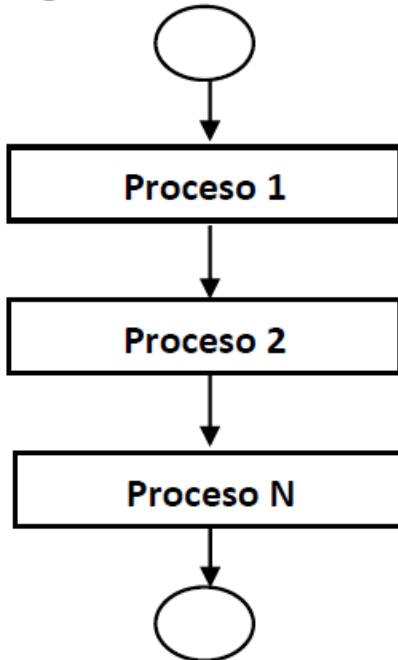
Ejercicios



Estructura lógica Secuencia



Figura 5. Diagrama con estructura secuencial



En pseudocódigo se presenta cuando se utilizan órdenes ó procesos en forma consecutiva, Así:

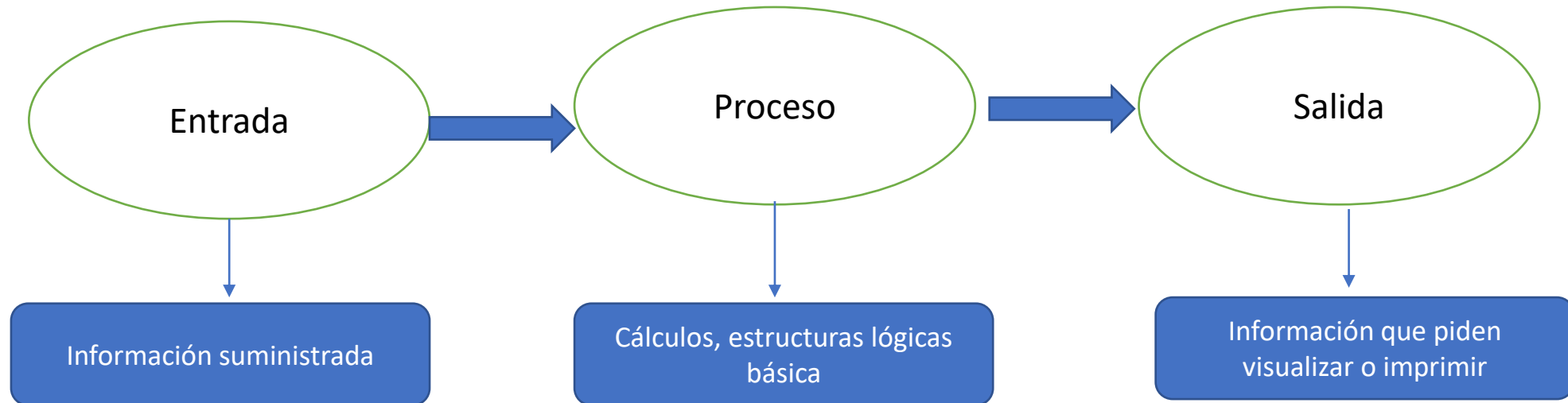
```
LEA A,B  
C=A+B  
ESCRIBA C
```



Ejercicios



Análisis → Método Entrada-Proceso-Salida





Ejercicios



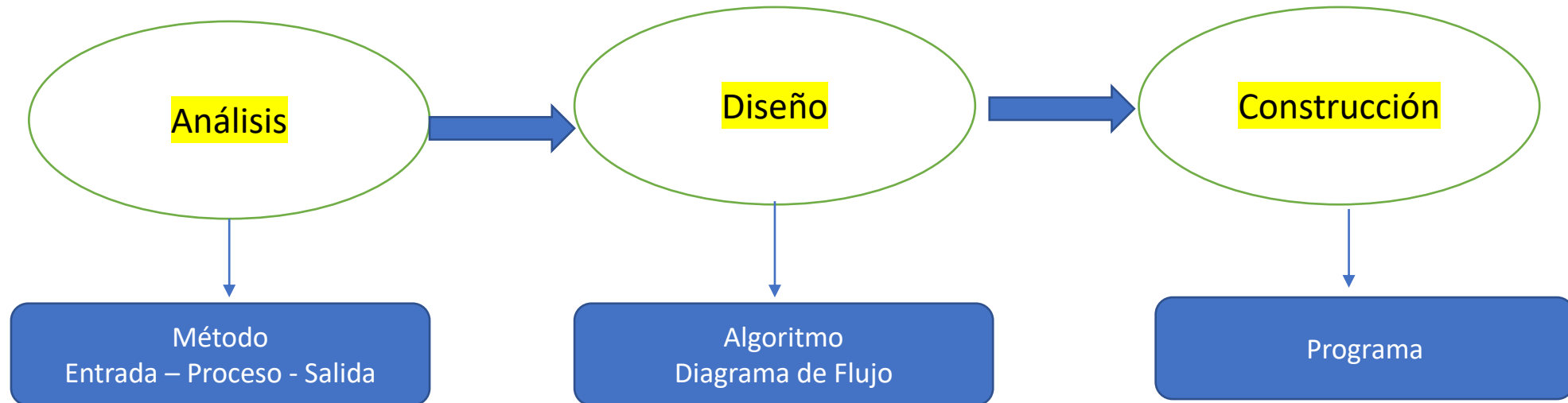
Dado la **base** y la **altura** de un triángulo, calcular y mostrar su **área**, a través de la **fórmula área = $(base * altura) / 2$**



Ejercicios



Metodología -> Pensamiento lógico estructurado





Ejercicios



Análisis → Método Entrada-Proceso-Salida

Se debe **LEER**



Entrada

base, altura

1

Proceso

$\text{area} = (\text{base} * \text{altura}) / 2$

2

Se debe **Visualizar o imprimir**



Salida

area

3



Variables

Entrada → Se debe leer → **input**

Salida → Se debe visualizar o imprimir → **print**



Diseño → Algoritmo

Ejercicios



```
1  Algoritmo area_triangulo
2      Escribir "Ingrese Base: "
3      Leer base
4      Escribir "Ingrese Altura: "
5      Leer altura
6      area<-(base*altura)/2
7      Escribir "Area: ",area
8      |
9  FinAlgoritmo
```

▶ PSeInt - Ejecutando proceso AREA_TRIAN...

*** Ejecución Iniciada. ***

Ingrese Base:

> 10.4

Ingrese Altura:

> 21.4

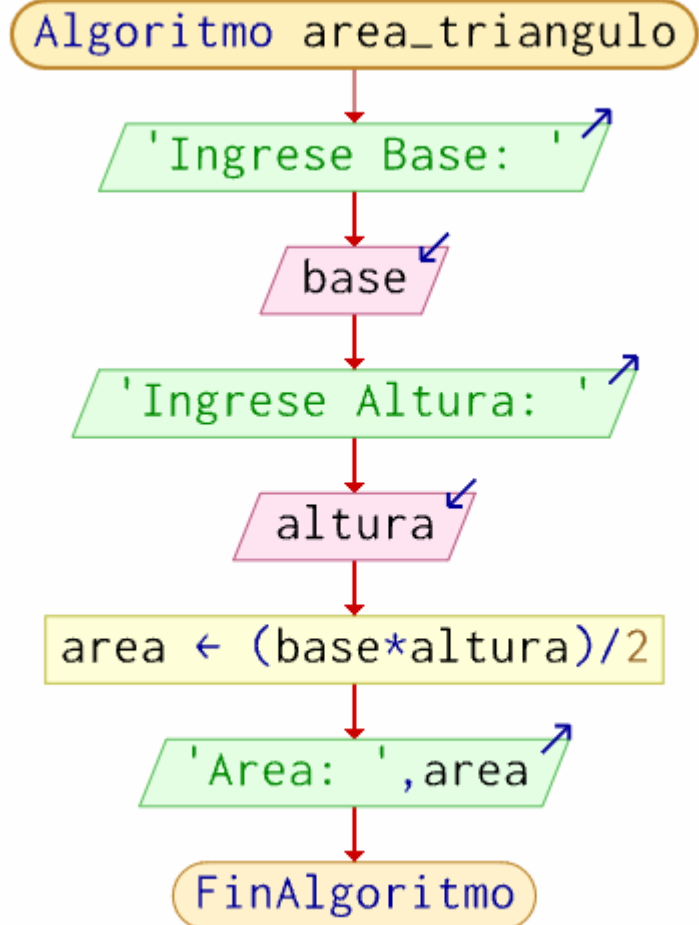
Area: 111.28

*** Ejecución Finalizada. ***



Diseño → Diagrama de flujo

Ejercicios





Construcción → Programa en Python

Ejercicios



```
#Programa para calcular el área de un triángulo
#Autor: Sergio Medina
#Fecha: 20/04/2022

#Entrada
base=float(input("Ingrese base: "))
altura=float(input("Ingrese altura: "))
#Proceso
area=(base*altura)/2
#Salida
print("El área es: ",area)
```



El futuro digital
es de todos

MinTIC



Fundamentos de Programación - Python

x
x
x



Formateo de Salidas

Función de Formateo de las salida

formareo_num.py - C:\Users\SERGIO\AppData\Local\Programs\Python\Python39\formareo_num.py (3.9.2)

File Edit Format Run Options Window Help

```
#Formateo de int
numero=4512342341
print('{:,}'.format(numero))

#Formateo de float
numero=123456.8987454
print('{:,.2f}'.format(numero))
```

Visualizar la puntuación de miles, millones, etc.

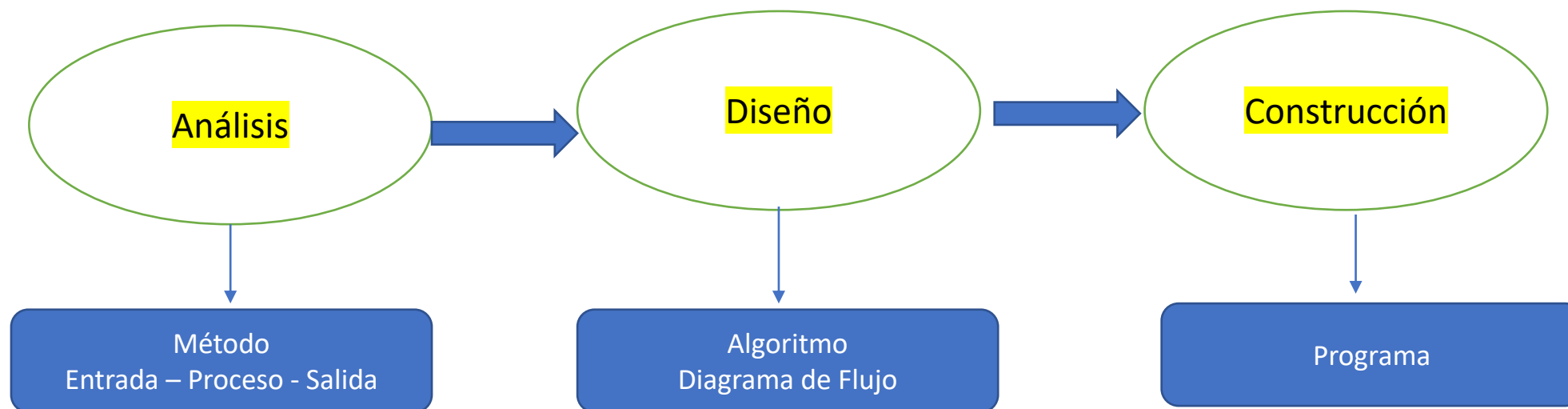
Cantidad de decimales



Sobre un tripulante conocemos las calificaciones parciales obtenidas los retos, **nota reto 1**, **nota reto 2**, **nota reto 3**, **nota reto 4**, **nota reto 5** y **nota de inglés**. Si los porcentajes para cada reto e inglés son **10%, 10%, 20%, 20%, 20% y 20%** respectivamente, calcular su **calificación definitiva**

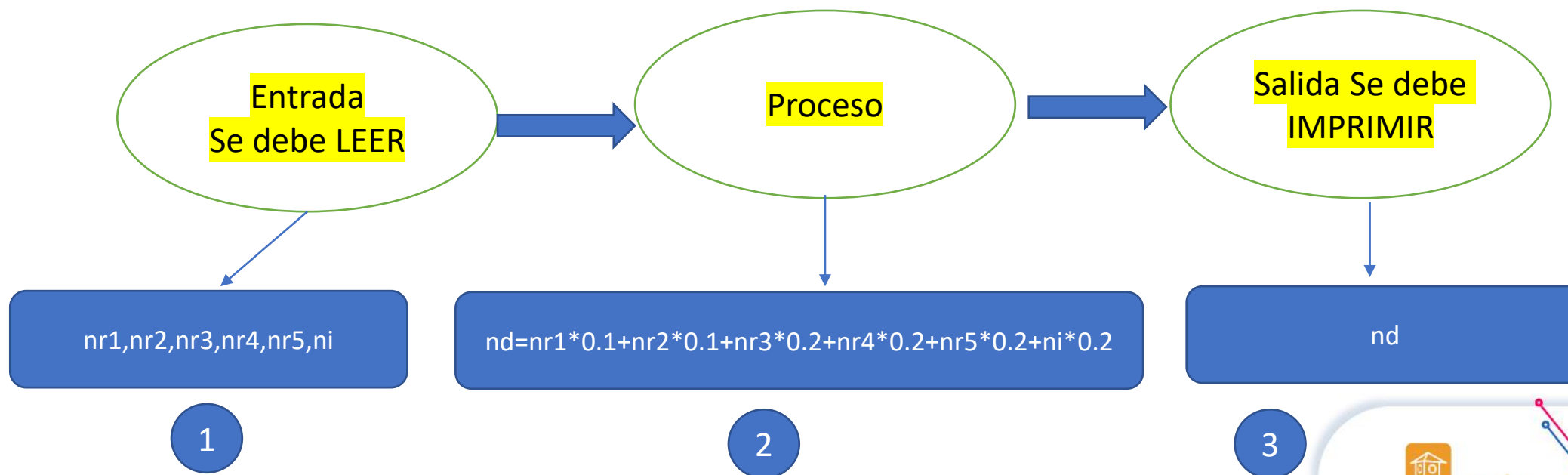


Metodología -> Pensamiento lógico estructurado





Análisis → Nota Definitiva del tripulante





Introducción a la Programación

Diseño → Algoritmo

Algoritmo nota_definitiva_tripulante

 Escribir "Nota reto 1: "

 Leer n1

 Escribir "Nota reto 2: "

 Leer n2

 Escribir "Nota reto 3: "

 Leer n3

 Escribir "Nota reto 4: "

 Leer n4

 Escribir "Nota reto 5: "

 Leer n5

 Escribir "Nota Inglés: "

 Leer ni

$nd = n1 * 0.1 + n2 * 0.1 + n3 * 0.2 + n4 * 0.2 + n5 * 0.2 + ni * 0.2$

 Escribir "Nota Definitiva: ",nd

FinAlgoritmo

Ejercicios





El futuro dig
es de todos

Introducción a la Programación

Ejercicios



Diseño → Diagrama de flujo





Construcción → Programa en Python

Ejercicios



```
# Programa para calcular nota definitiva de un tripulante  
# Autor: Sergio Medina  
# Fecha: 21/04/2022
```

```
n1=float(input("Nota reto 1: "))  
n2=float(input("Nota reto 2: "))  
n3=float(input("Nota reto 3: "))  
n4=float(input("Nota reto 4: "))  
n5=float(input("Nota reto 5: "))  
ni=float(input("Nota Inglés: "))  
nd=n1*0.1+n2*0.1+n3*0.2+n4*0.2+n5*0.2+ni*0.2  
print("Nota definitiva: ", '{:,.2f}'.format(nd))
```



El futuro digital
es de todos

MinTIC

Introducción A la Programación

»
Misión TIC 2022

xxx



Misión
TIC 2022