



El futuro digital  
es de todos

MinTIC

«Misión  
TIC2022»

Fund. de Programación

Grupos 79,80,81



UNIVERSIDAD  
DE ANTIOQUIA

Facultad de Ingeniería

# Resumen sesión anterior





# PASOS PARA CONSTRUIR UN PROGRAMA

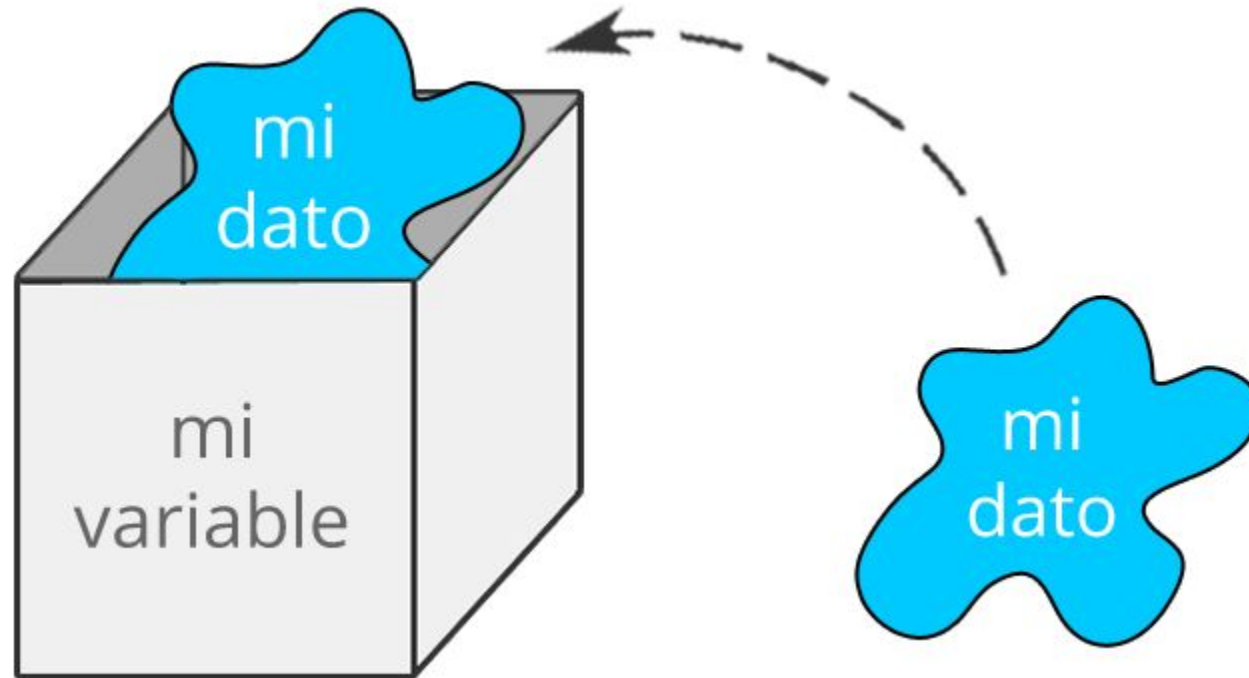
1. Analizar el problema.
2. Diseñar la solución.
3. Implementar la solución diseñada.
  - 3.1. Elaborar el algoritmo
  - 3.2. Codificar la solución en un lenguaje de programación.
  - 3.4. Compilación de la solución.
  - 3.5. Pruebas sistemáticas.
1. Pruebas con el usuario.
2. Despliegue.
3. Mantenimiento.



# CARACTERÍSTICAS DE PYTHON

- Fuertemente tipado
- Tipado dinámico
- Simple
- Multiplataforma
- Filosofía

# VARIABLES



# programación con Python





# ACTIVIDAD 1

**Problema:**

Comparar dos números e indicar cual de ellos es mayor

Implementar la solución descrita anteriormente usando python.





# TIPOS DE DATO

Categoría de tipo	Nombre	Descripción
<i>Números inmutables</i>	int	<u>entero</u>
	long	<u>entero long</u>
	float	<u>coma flotante</u>
	complex	<u>complejo</u>
	bool	<u>booleano</u>
<i>Secuencias inmutables</i>	str	<u>cadena de caracteres</u>
	unicode	<u>cadena de caracteres Unicode</u>
	tuple	<u>tupla</u>
	xrange	<u>rango inmutable</u>
<i>Secuencias mutables</i>	list	<u>lista</u>
	range	<u>rango mutable</u>
<i>Mapeos</i>	dict	<u>diccionario</u>
<i>Conjuntos mutables</i>	set	<u>conjunto mutable</u>
<i>Conjuntos inmutables</i>	frozenset	<u>conjunto inmutable</u>



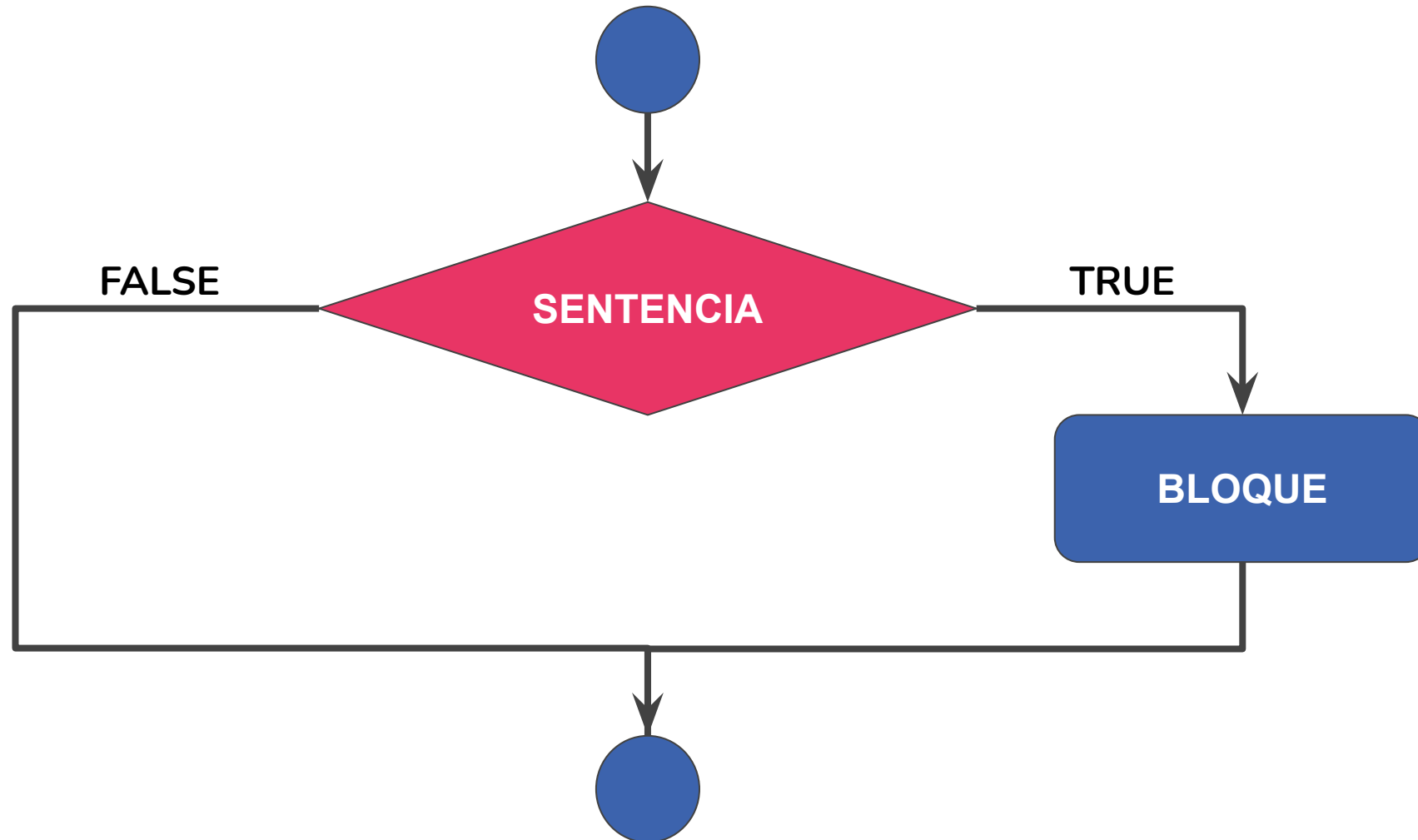


# OPERACIONES BOOLEANAS

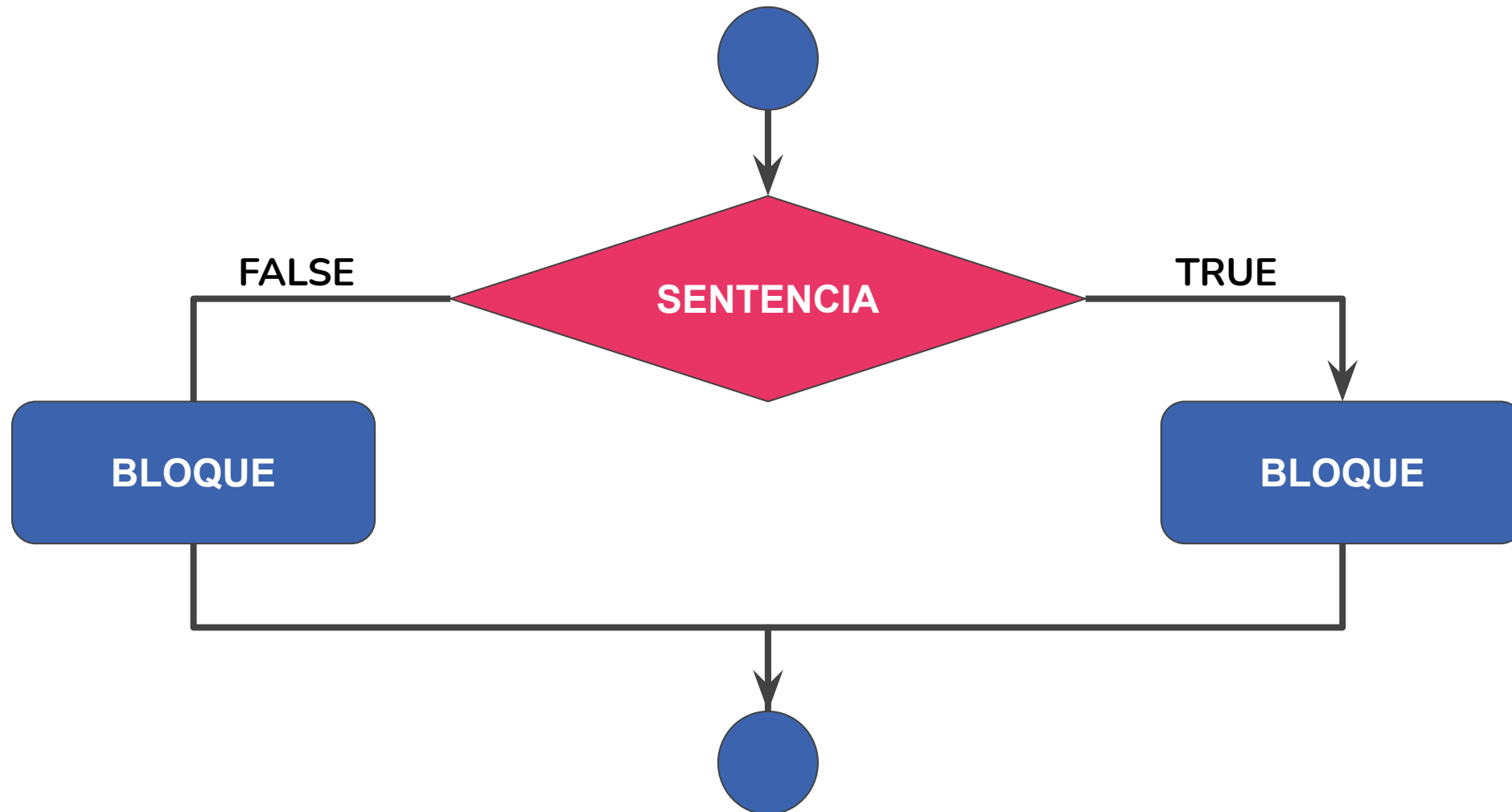
A continuación se enumeran algunas booleanas

$A > B$ $(A \geq B)$	A mayor que B (A mayor o igual que B)
$A < B$ $(A \leq B)$	A menor que B (A menor o igual que B)
$A == B$	A igual que B
$A != B$	A diferente que B
<b>not</b>	negación de un bool
<b>and</b>	Se deben cumplir ambas condiciones
<b>or</b>	Se cumple al menos una de las condiciones

# CONDICIONALES (IF)



# CONDICIONALES (IF ELSE)





# PASOS PARA CONSTRUIR UN PROGRAMA

1. Analizar el problema.
2. Diseñar la solución.
3. Implementar la solución diseñada.
  - 3.1. Elaborar el algoritmo
  - 3.2. Codificar la solución en un lenguaje de programación.
  - 3.4. Compilación de la solución.
  - 3.5. Pruebas sistemáticas.
1. Pruebas con el usuario.
2. Despliegue.
3. Mantenimiento.



## ACTIVIDAD 2

### **Problema:**

Diseñar un juego donde se genere un número aleatorio del 1 al 20, y el usuario tenga 3 oportunidades para adivinarlo. En cada intento se debe indicar si el numero ingresado es mayor o menor al número aleatorio.

Implementar la solución usando python.



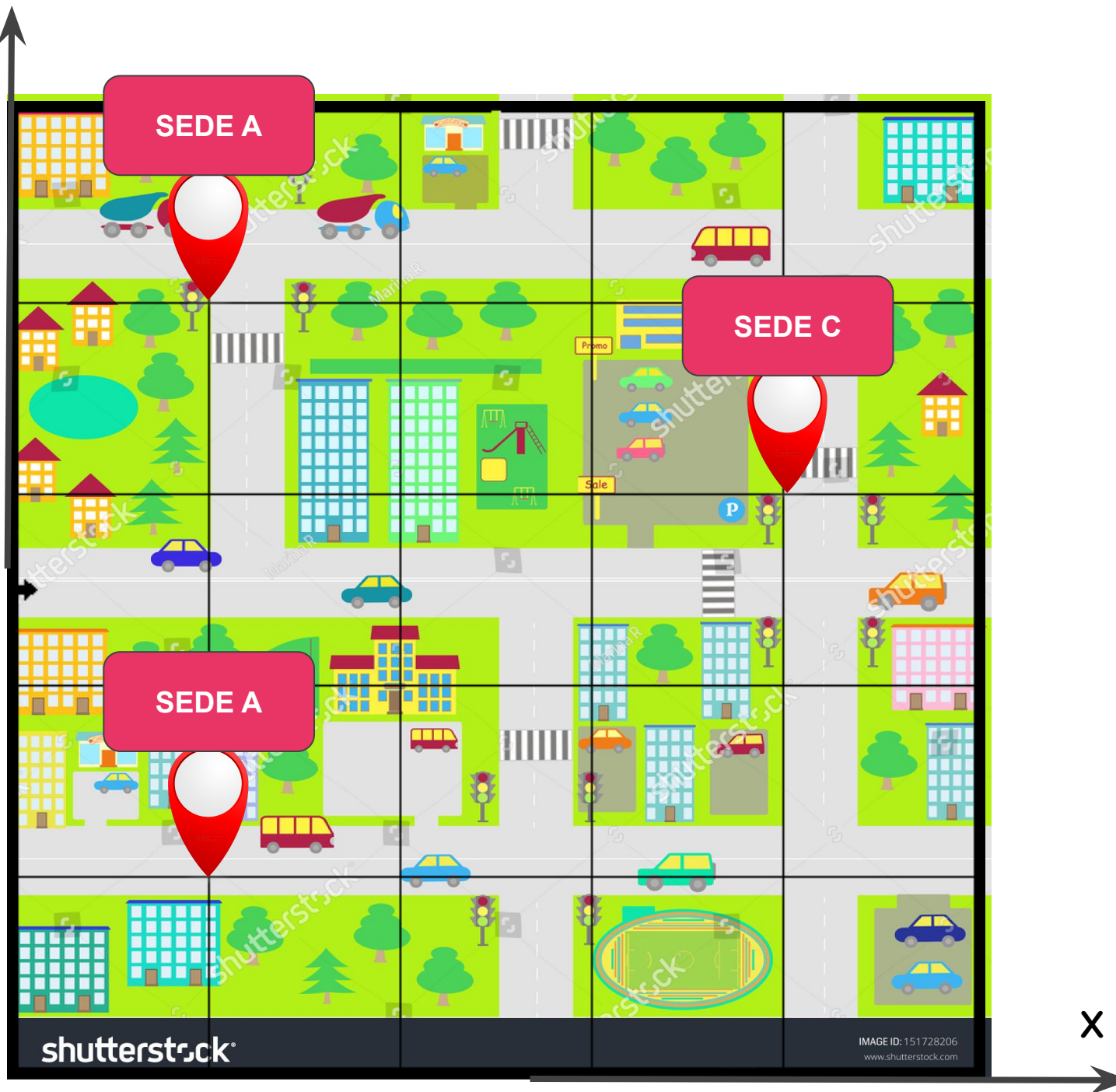
## ACTIVIDAD 3

### **Problema:**

El restaurante Menudencia express cuenta con 3 sedes ubicadas en distintos puntos de la ciudad, como se muestra en la siguiente diapositiva. Se debe crear una aplicación para seleccionar la sede más cercana en caso que un usuario pida un domicilio (Se ingresa la posición de un usuario)

Implementar la solución usando python.

Y



**SEDE A=(1,4)**  
**SEDE B=(1,1)**  
**SEDE C=(4,3)**

X