El futuro digital es de todos

MinTIC

'Mision TIC 2022'

Fund. de Programación

Grupos 79,80,81

UNIVERSIDAD DE ANTIOQUIA

1803

Facultad de Ingeniería

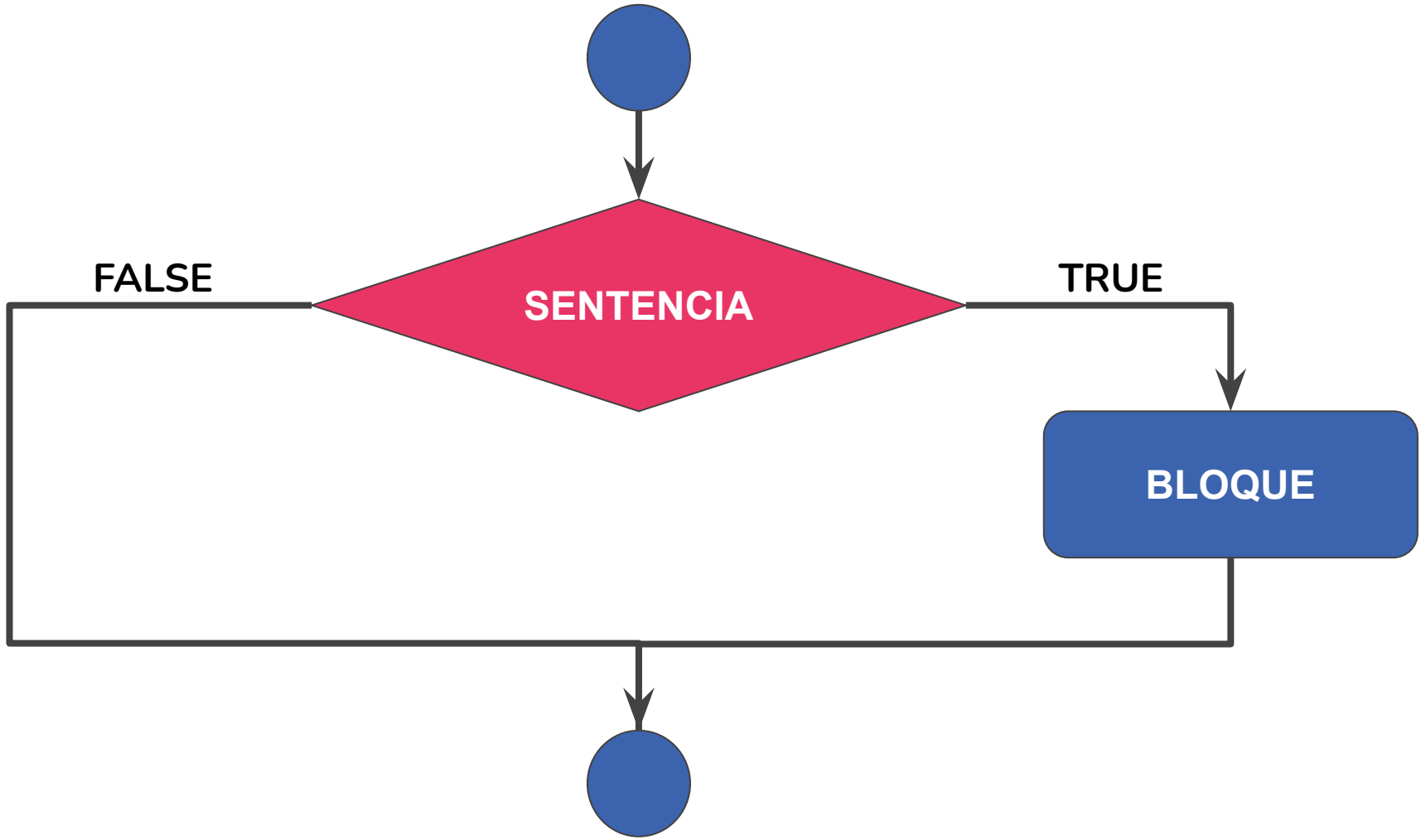
Resumen sesión anterior

OPERACIONES BOOLEANAS

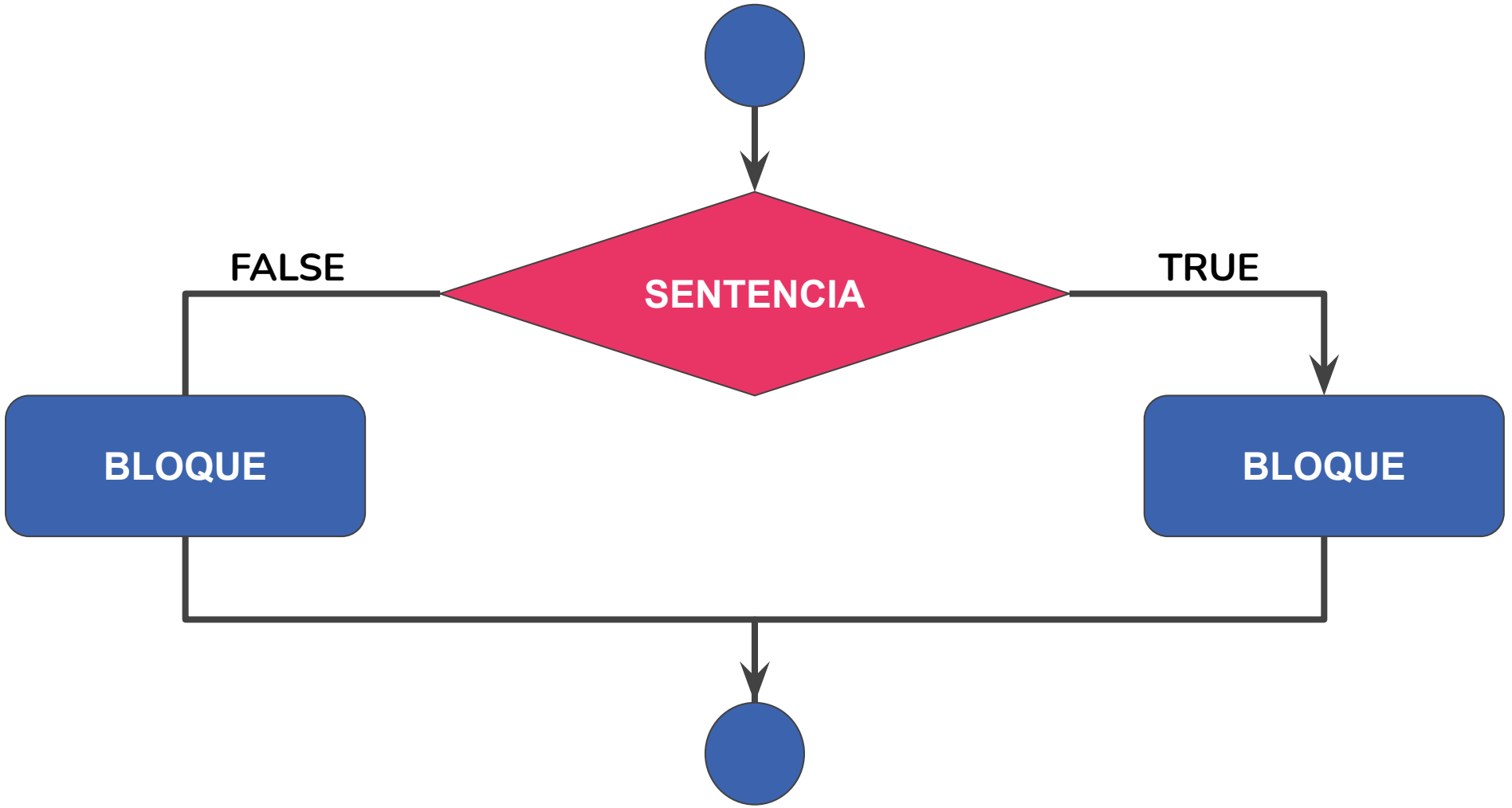
A continuación se enumeran algunas booleanas

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

CONDICIONALES (IF)



CONDICIONALES (ELSE y ELIF)



programación con Python

JUAN FERNANDO GONZÁLEZ
GRUPOS 79,80,81
SESIÓN 3

SEMANA 2

Temas a tratar:

- **Condicionales**
- **Ciclos**
- **requisitos funcionales**

DATOS ORDENADOS

- **Tuplas**
- **Listas**
- **Diccionarios**

TUPLAS

En Python, una tupla es un conjunto ordenado e inmutable de elementos del mismo o diferente tipo

- Los elementos están encerrados en un paréntesis y separados por coma ***tupla=('Ele_1', 'Ele_2', 'Ele_3')***
- Pueden almacenar distintos tipos de datos
- Se pueden concatenar
- ***tupla[<posición>]*:** Retorna el elemento en la posición dada.
- ***tupla.count(<elemento>)***: cuenta cuantas veces aparece un elemento en la tupla
- **tupla.index(<elemento>):** retorna la posición en la que está el elemento.

LISTAS

Las listas son conjuntos ordenados de elementos (números, cadenas, listas, etc). Las listas se delimitan por corchetes ([ ]) y los elementos se separan por comas.

- Puede almacenar cualquier tipo de elemento
- Se pueden concatenar
- Tienen más opciones que una tupla para manejar los datos

MÉTODOS PARA LISTAS

- **append(<elemento>)**: Agrega un elemento a una lista
- **extend(<[lista]>)**: Expande la lista con varios elementos.
- **remove(<elemento>)**: Elimina el elemen
- **pop(<indice>)**: Elimina un elemento por su índice
- **clear():** Elimina todos los elementos de una lista
- aplica **index()** y **count()**
- **insert(pos, eleme):** Agrega un elemento en la posición dada

DICCIONARIOS

Son estructuras de datos que permiten guardar información ordenada por medio de un mapeo key:valor como se muestra a continuación

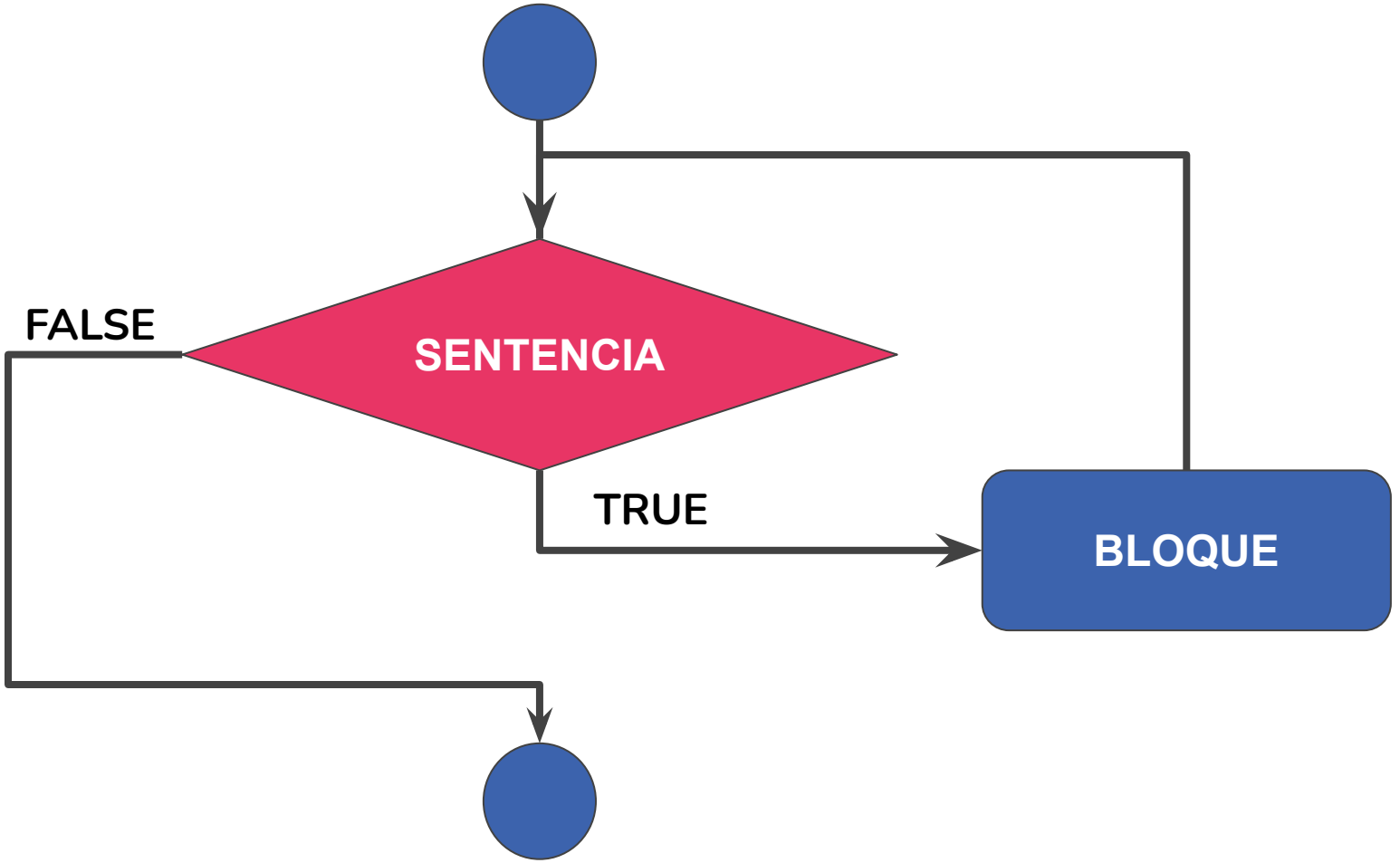
midiccionario={‘key_1’:valor_1, “key_2”:”valor_2”}

- Para acceder a un valor **minidiccionario[key]**

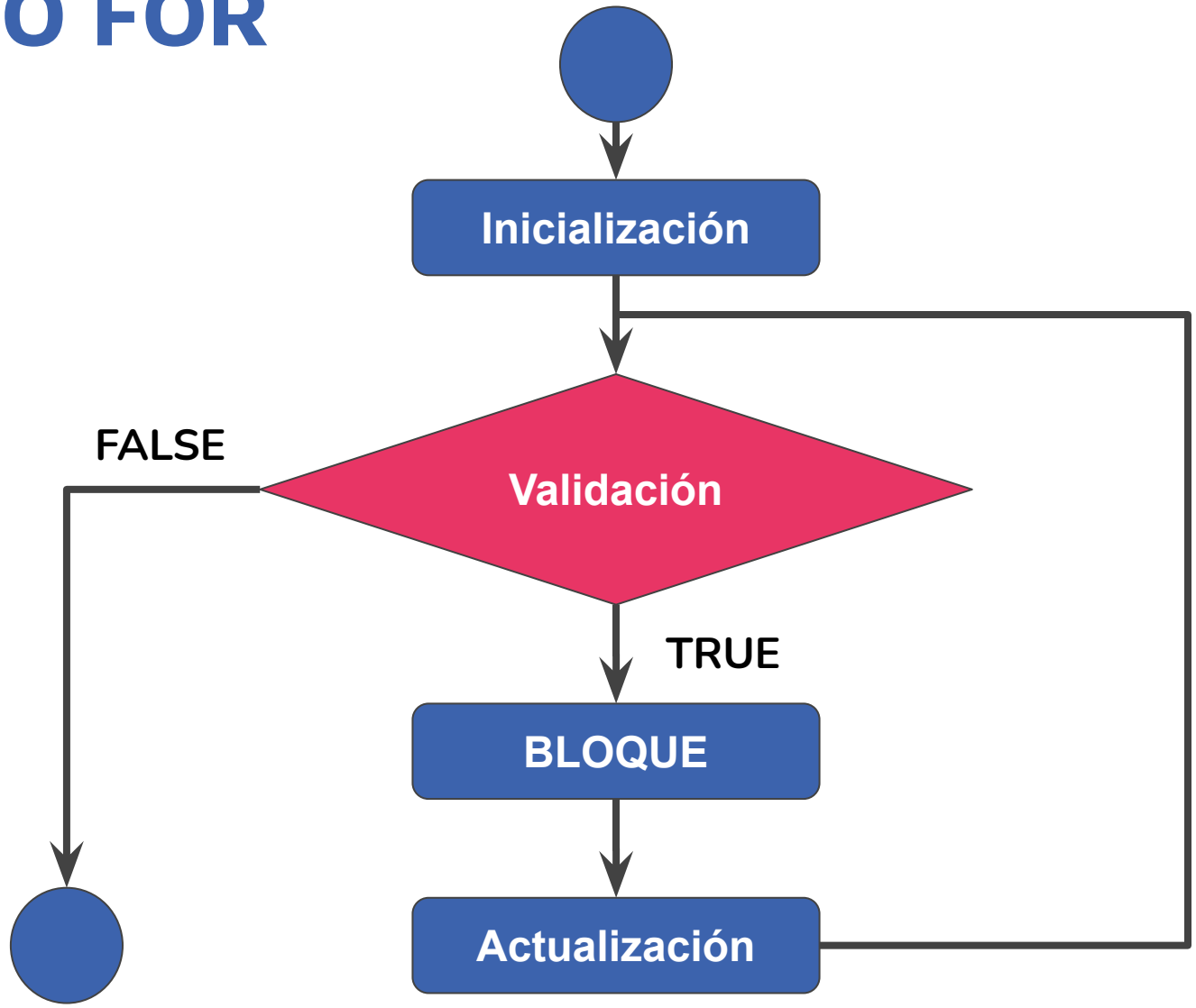
MÉTODOS PARA DICCIONARIOS

- **dict([(key, valor), (key, valor)]):** Define un dict
- **keys():** Retorna las keys del diccionario
- **items():** Retorna los elementos del dict
- **values():** Retorna los valores asociados a las keys
- **clear():** Elimina todos los elementos del dict
- **get(keay):** Retorna el valor asociado a la clave
- **pop(key):** Elimina el registro con esa clave y retorna el valor que tenía almacenado.

CICLO WHILE



CICLO FOR


Inicialización
Validación
FALSE
TRUE
BLOQUE
Actualización

ACTIVIDAD 2

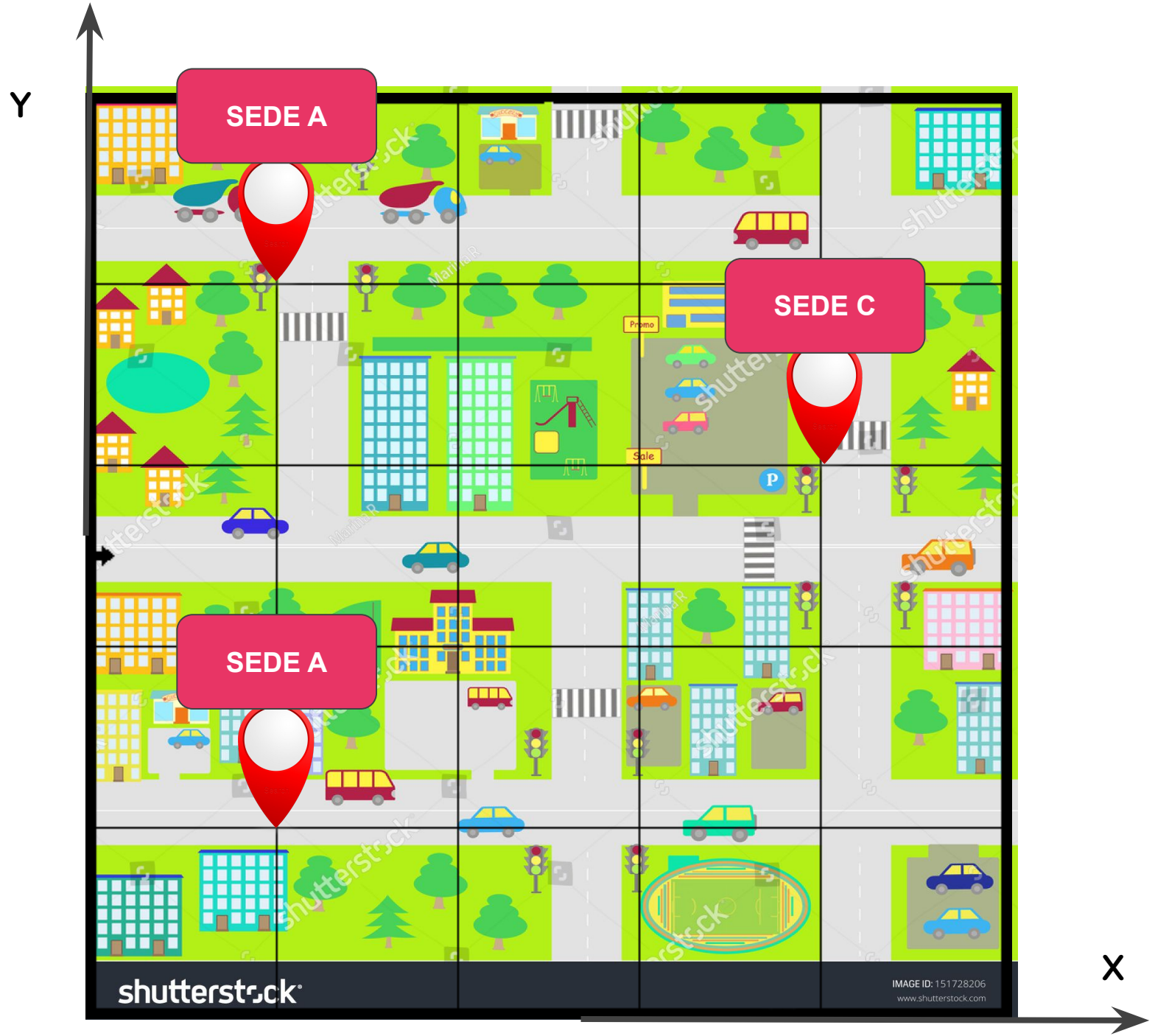
Problema:
Diseñar un juego donde se genere un número aleatorio del 1 al 20, y el usuario tenga 3 oportunidades para adivinarlo. En cada intento se debe indicar si el numero ingresado es mayor o menor al número aleatorio.

Implementar la solución usando ciclo while 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.



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