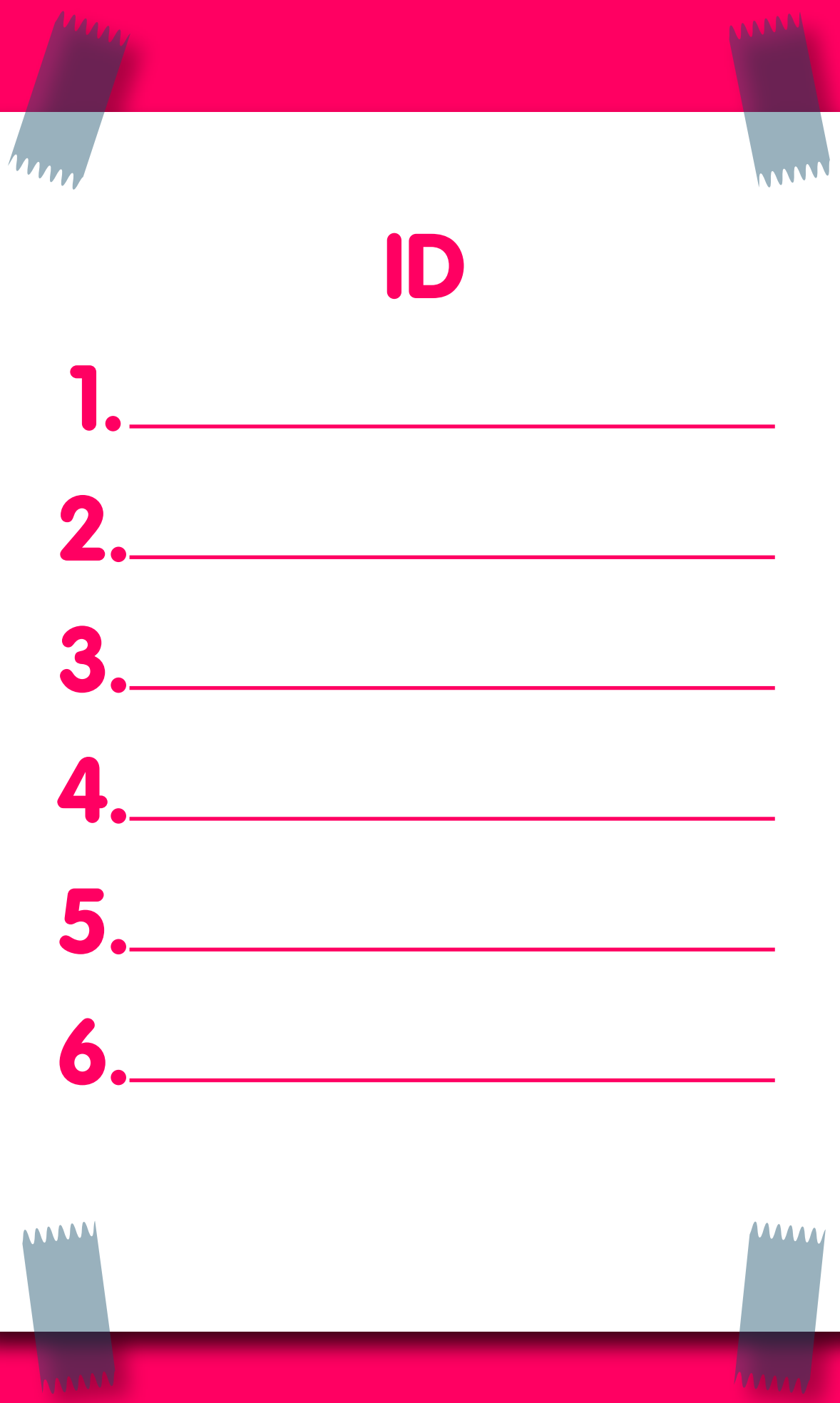
El futuro digital es de todos

MinTIC



Estructura de datos: Listas

Universidad de Caldas

¡Hola a todos!

Hemos dado un gran salto en los aspectos relacionados con el control de flujo, el manejo de ciclo nos abre mucho las posibilidades para resolver problemas más complejos, ahora vamos a dar un salto relacionado con las estructuras de datos; a partir de este momento vamos a iniciar con las listas; las listas junto con los ciclos repetitivos nos permiten hacer programas más dinámicos y cercanos a lo que vemos constantemente en el mundo real; el propósito es que entendamos este tema y lo apliquemos siempre consultando con nuestros formadores y tutores y explorando en el material bibliográfico,

Bienvenidos

Las listas, las cuales constituyen nuestro tema en este capítulo,se encuentran presentes en todas partes, no solamente desde el punto de vista computacional y de programación, ya que técnicamente hacemos aplicaciones para ese mundo real en el que nos movemos; es allí donde encontramos este concepto, en muchos elementos cotidianos, tales como, la lista de mercado, lista de amigos, lista de deudas, lista de notas, lista de dispositivos lista de estudiantes la lista de asistentes, lista de ingredientes, películas, coordenadas geográficas, y así podríamos aumentar esta lista.

Veamos detenidamente algunos ejemplos de estas listas. Tomemos como base una lista de mercado o una lista de regalos, en ellas podemos identificar elementos comunes.

LISTA DE MERCADO

1. Avena
2. Quínoa
3. Stevia en polvo
4. Miel de avejas
5. Frutas
6. Almendras

LISTA DE REGALOS

- [x] LISTA DE REGALOS
- [ ] Iphone 11
- [ ] Nintendo switch
- [ ] Lenovo yoga
- [ ] Cafetera Oster
- [ ] Nevera pequeña

En primer lugar se cuenta con el nombre de la lista, es claro que a la izquierda tenemos la lista del mercado, ese es su nombre, hablando computacionalmente ese es su Id y la otra lista se llama lista de regalos, vemos allí también nombres de objetos, con orden establecido.

Los anteriores son elementos importantes por entender ya que esto que tenemos en hojas de papel y que lo hacemos a mano, es necesario representarlo de alguna manera en el lenguaje computacional y el mismo concepto, el de lista, por lo tanto se hace conveniente identificar los componentes de una lista para que cuando lo veamos en su representación computacional lo entendamos y veamos que simplemente es otra manera de manejar lo que para nosotros ha sido natural por mucho tiempo.

Profundicemos en lo que tenemos hasta ahora, estamos hablando de estructura de datos, recordemos que hasta el momento, esas estructuras de datos tiene un nombre o ID, un valor y un tipo de dato y ocupan un espacio en la memoria del computador.

Salario = 50000

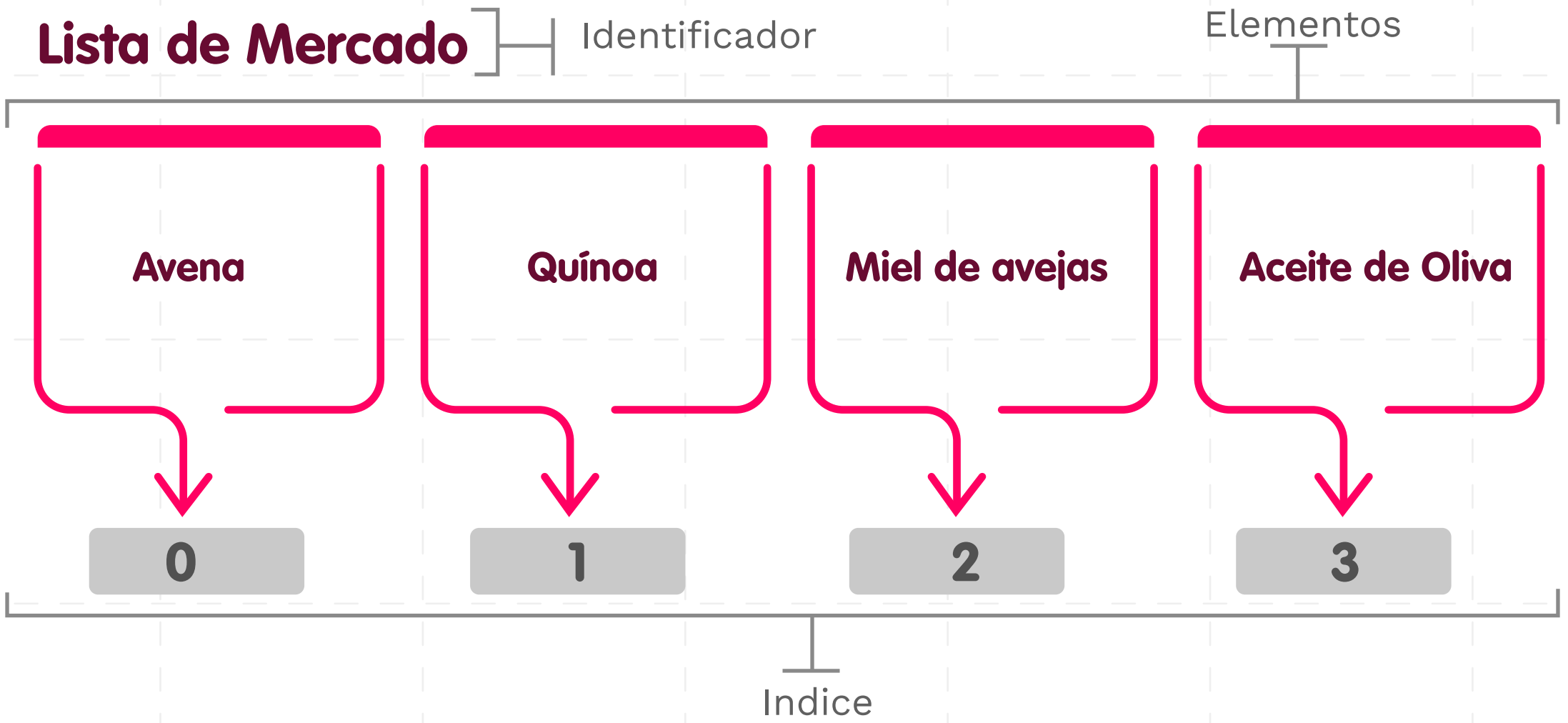
Promedio = 3.45

Respuesta = True

Mensaje = "Hola"

Estos son los tipos de datos que hemos visto, pero ¿será que nos sirven para almacenar una lista de compras, una lista de amigos o de ciudades? quizás las cadenas de caracteres o strings son los más parecidos porque representan literalmente una lista, sin embargo estas son listas donde cada uno de los elementos se representan por un solo carácter, no es una sola palabra, entonces lo primero que podemos concluir es que dichas listas no nos permiten manejar elementos que van a ser muy necesario para los problemas de programación, de esta manera, necesitamos un nuevo tipo de datos, ese nuevo tipo de datos nos debe permitir almacenar varios elementos.

En primer lugar, definiremos el concepto de lista; una lista es una secuencia o colección de valores del mismo o de diferente tipo, teniendo esto en cuenta analicemos el siguiente concepto, en el cual representaremos una lista. Analicemos las que serían técnicamente las partes de una lista



En Python y la mayoría de lenguajes de programación con respecto a este y otros ejemplos, las listas no empiezan en 1 sino en 0, para crear una lista en Python, definimos una lista entre corchetes y los elementos son separados por comas.

```
>>> [“Avena”, “Quínoa”, “Miel de abejas”, “Aceite de Oliva”]

>>> amigos= [ “John”, “Luis”, “Juan Pablo”, “Carolina”]

>>>ítems= list ()

>>>ítems= [ ]
```

En el ejemplo, tenemos 4 tipos de listas, la primera línea es una expresión la cual no queda almacenada en ninguna parte, la segunda una instrucción que está asignada, la lista ya instanciada es decir con elementos a una variable o también la podemos crear vacías como la 3 y 4 línea.
Al tener definida una lista, ésta se puede modificar, es posible adicionar elementos entre otras cosas, por ejemplo si tenemos la lista **“ítems”** (la cual estaba vacía), con la función append que tiene como parámetro el elemento que deseamos adicionar a la lista, quedaría de esta manera en el lenguaje de programación Python, Java y c# como se muestra a continuación.

```
>>> ítems.append (“Python”)
>>> ítems.append (“Java”)
>>> ítems.append (“C#”)
>>> ítems
[‘Python’, ‘Java’ , ‘C#’ ]
```

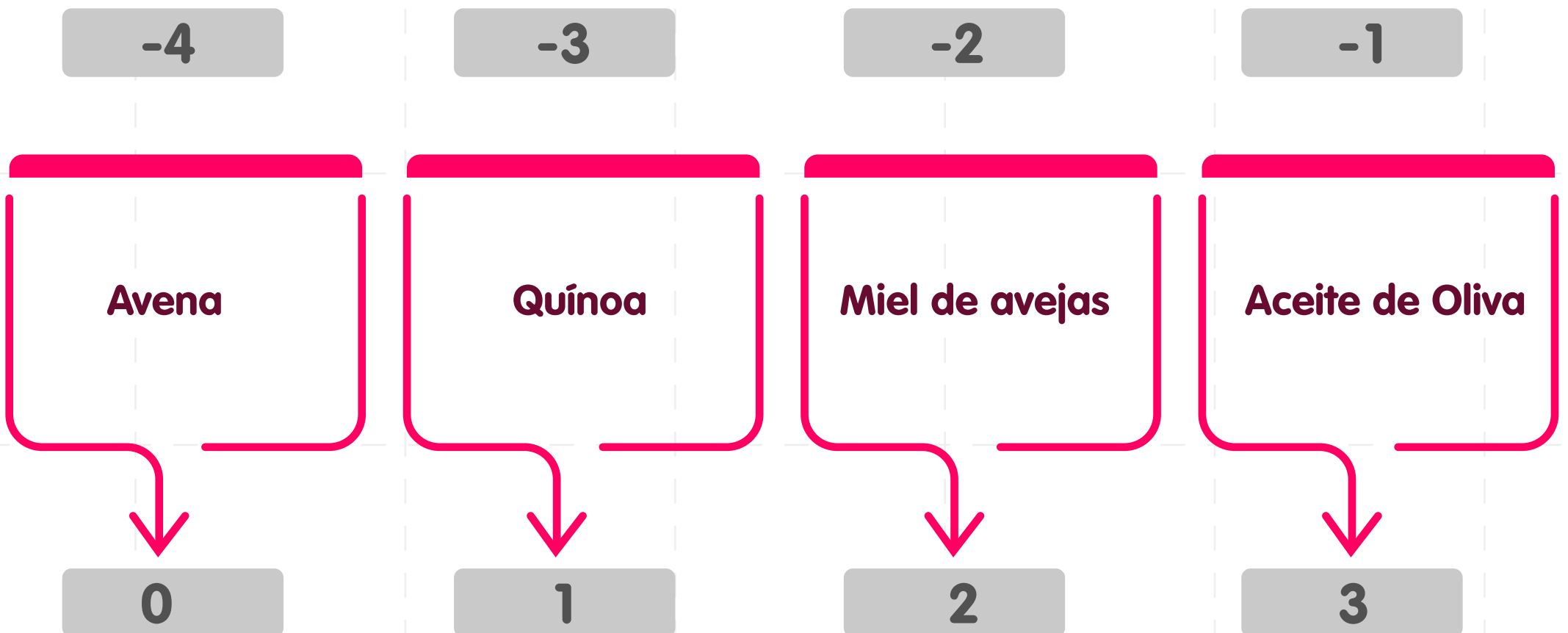
Una gran ventaja también, es que si deseamos adicionar no solamente uno por uno los elementos sino que se puede extender la lista, con otra lista de la siguiente manera

```
>>> ítems.append (“Python”)
>>> ítems.append (“Java”)
>>> ítems.append (“C#”)
>>> ítems
[‘Python’, ‘Java’ , ‘C#’ ]
>>>ítems.extend ([ “Ruby”, “Kotlin”])
>>>ítems
[‘Python’, ‘Java’, ‘C#’, ‘Ruby’, ‘kotlin’]
```

Además de entender el concepto de **lista**, saber cuáles son sus partes y cómo se crean, otro concepto clave es la indexación, es decir la manera en la cual se puede llegar a un elemento de la lista, como sabemos que Python se encuentra en la posición 0 de la lista, al escribir el nombre de la lista y entre corchetes 0, (normalmente se lee ítems sub 0), se obtendrá el valor del elemento en esa posición así:

```
>>>ítems
[‘Python’, ‘Java’, ‘C#’, ‘Ruby’,
‘kotlin’]
>>>ítems [0]
‘Python’
```

Es posible indexar de otra manera, con números negativos de atrás hacia adelante, esto quiere decir que una lista en Python tiene dos índices de recorrido el de izquierda a derecha y el de derecha a izquierda, en el siguiente ejemplo se pueden ver claramente las indexaciones.



Este tema es muy útil ya que podemos hacer muchos ejercicios con él.

También tenemos sublistas aquí estamos diciendo items[0: 2] lo que estoy diciendo es que a partir de la posición 0, se cuenten dos elementos así:

```
>>>ítems
[‘Python’, ‘Java’, ‘C#’, ‘Ruby’,
‘kotlin’]
>>>ítems [0:2]
[‘Python’, ‘Java’]
```

En resumen hemos visto las listas, las cuáles son estructuras de datos compuestas, secuencia o colección elementos que permiten almacenar elementos del mismo o diferentes tipos, de igual manera hemos conocido las partes de las listas, así como la manera de adicionar elementos e indexarlos, de todas esas operaciones es muy importante entender las sublistas que se pueden obtener.

Aumentemos cada vez más nuestras herramientas de programación las cuales van a permitir hacer programas más interesantes, no olviden hacer los laboratorios y talleres,

hasta el próximo video en revista.


Mision
TIC2022

Universidad de Caldas