



## Objetivo de Clase:

- Crear algoritmo que utiliza Listas y Tuplas

# LISTAS

Secuencia de valores encerrados entre corchetes y separados por comas.

`edades=[]`

`edades=[1,2,3,4,"5"]`

`nombres = ["luis",2,"ana"]`

Programación	
Operación	Resultado
<code>x in s</code>	Indica si la variable <code>x</code> se encuentra en <code>s</code>
<code>s + t</code>	Concatena las secuencias <code>s</code> y <code>t</code>
<code>s * n</code>	Concatena <code>n</code> copias de <code>s</code>
<code>s[i]</code>	Elemento <code>i</code> de <code>s</code> , empezando por <code>0</code>
<code>s[i:j]</code>	Porción de la secuencia <code>s</code> desde <code>i</code> hasta <code>j</code> (no inclusive)
<code>s[i:j:k]</code>	Porción de la secuencia <code>s</code> desde <code>i</code> hasta <code>j</code> (no inclusive), con paso <code>k</code>
<code>len(s)</code>	Cantidad de elementos de la secuencia <code>s</code>
<code>min(s)</code>	Mínimo elemento de la secuencia <code>s</code>
<code>max(s)</code>	Máximo elemento de la secuencia <code>s</code>

Secuencia de valores encerrados entre corchetes y separados por comas.

```
edades=[1,2,3,4,"5"]
```

```
nombres = ["luis",2,"ana"]
```

```
len(edades)          ?
```

```
len(nombres)         ?
```

```
nombres[1:2]         ?
```

```
edades[3:5]          ?
```

```
edades.index(3)      ?
```

Secuencia de valores encerrados entre corchetes y separados por comas.

```
edades=[1,2,3,4,"5"]
```

```
nombres = ["luis",2,"ana"]
```

```
len(edades)          5
```

```
len(nombres)         3
```

```
Nombres[1:2]          [2]
```

```
edades[3:5]           [ 4,'5']
```

```
edades.index(3)       2
```

Secuencia de valores encerrados entre corchetes y separados por comas.

```
Edades=[1,2,3,4,"5"]
```

```
nombres = ["luis",2,"ana"]
```

```
edades[3]=99
```

```
print (edades[3])
```

?

```
print (edades[:])
```

```
edades.append(88)
```

?

```
print (edades[:])
```

```
edades.insert(2, 77)
```

?

```
print (edades[:])
```

```
edades.remove(77)
```

?

```
print (edades[:])
```

Secuencia de valores encerrados entre corchetes y separados por comas.

```
edades=[1,2,3,4,"5"]
```

```
nombres = ["luis",2,"ana"]
```

3 in edades ?

9 in edades ?

Recorrer lista edades utilizando FOR

Secuencia de valores encerrados entre corchetes y separados por comas.

Recorrer una lista utilizando FOR

```
edades=[1,2,3,4,"5"]  
for i in edades:  
    print ("numero : ",i)
```

Buscar el valor mayor de una lista numérica de valores ingresados por teclado.

```
edades=[1,7,3,4,9]
```

Secuencia de valores encerrados entre corchetes y separados por comas.

Recorrer una lista utilizando FOR

```
edades=[1,7,3,4,9]
```

```
edades.sort()
```

```
for i in edades:
```

```
    print ("numero : ",i)
```



Crear un algoritmo que permita:

- Al usuario ingresar los nombres de los estudiantes de la asignatura de Programación y almacenarlos en una Lista llamada ESTUDIANTES.

[ Matias, Micha, Benjamin, Oriana, Randall ]

Matias		Micha		Benjamin		Oriana		Randall
--------	--	-------	--	----------	--	--------	--	---------

0

1

2

3

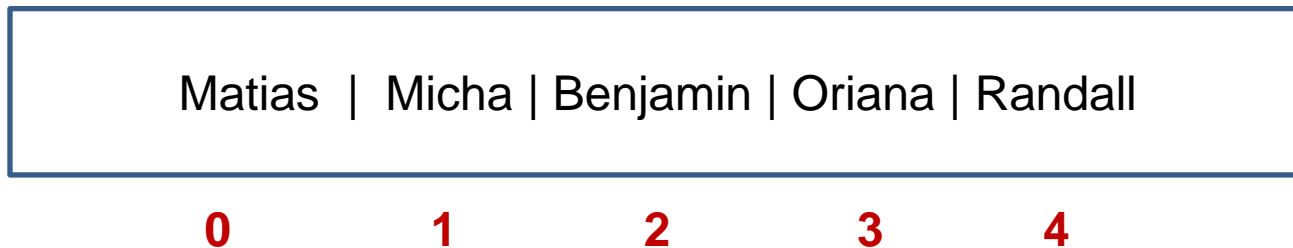
4

Después de finalizar el ingreso de datos a la lista debe mostrarla en pantalla

Crear un algoritmo que permita:

- Al usuario ingresar los nombres de los estudiantes de la asignatura de Programación y almacenarlos en una Lista llamada ESTUDIANTES.

[ Matias, Micha, Benjamin, Oriana, Randall ]



Después de finalizar el ingreso de datos a la lista debe:

- Mostrar la lista en pantalla
- BUSCAR: Permite al usuario ingresar un nombre y buscarlo en lista, indicando si existe o no. Si existe debe indicar en que posición de la lista se encuentra.

Crear un algoritmo que permita:

- Al usuario ingresar los nombres de los estudiantes de la asignatura de Programación y almacenarlos en una Lista llamada ESTUDIANTES.

[ Matias, Micha, Benjamin, Oriana, Randall ]

Matias		Micha		Benjamin		Oriana		Randall
--------	--	-------	--	----------	--	--------	--	---------

0

1

2

3

4

Después de finalizar el ingreso de datos a la lista debe:

- Mostrar la lista en pantalla
- BUSCAR: Permite al usuario ingresar un nombre y buscarlo en lista, indicando si existe o no. Si existe debe indicar en que posición de la lista se encuentra.
- ORDENAR: Ordenar la lista y mostrarla

Crear un algoritmo que permita:

- Al usuario ingresar los nombres de los estudiantes de la asignatura de Programación y almacenarlos en una Lista llamada ESTUDIANTES.

[ Matias, Micha, Benjamin, Oriana, Randall ]

Matias		Micha		Benjamin		Oriana		Randall
--------	--	-------	--	----------	--	--------	--	---------

0

1

2

3

4

Después de finalizar el ingreso de datos a la lista debe:

- Mostrar la lista en pantalla
- BUSCAR: Permite al usuario ingresar un nombre y buscarlo en lista, indicando si existe o no. Si existe debe indicar en que posición de la lista se encuentra.
- ORDENAR: Ordenar la lista y mostrarla
- Pedir al usuario el ingreso de un nombre y eliminarlo de la lista
- Mostrar la lista

Crear un algoritmo que permita:

- Al usuario ingresar los nombres de los estudiantes de la asignatura de Programación y almacenarlos en una Lista llamada ESTUDIANTES.

[ Matias, Micha, Benjamin, Oriana, Randall ]

Matias		Micha		Benjamin		Oriana		Randall
--------	--	-------	--	----------	--	--------	--	---------

0

1

2

3

4

Des pues de finalizar el ingreso de datos a la lista debe mostrar un menú con las siguientes opciones:

- BUSCAR: Permite al usuario ingresar un nombre y buscarlo en lista, indicando si existe o no. Si existe debe indicar en que posición de la lista se encuentra.
- MOSTRAR TODO: Permite Mostrar Todo el contenido de la lista .
- ORDENAR: Ordenar la lista
- ELIMINAR: Pedir al usuario el ingreso de un nombre y eliminarlo de la lista
- SALIR: Única forma de salir de la ejecución del menú

# Tuplas

**tupla= ( elemento 1, elemento 2, elemento 3, ... )**

Una tupla es una secuencia inmutable e indexable de elementos.

**Indexable:** Sus elementos se obtienen indicando su índice

**Inmutable:** Sus elementos NO pueden ser modificados

```
t0 = ()                # Tupla vacía
t1 = (3 ,)             # Tupla con un elemento
t2 = (5 ,3 ,9)         # Tupla de números
t3 = ('Juan ','Perez ') # Tupla de string
t4 = ('Juan ','P´e rez ', 18) # Tupla mixta
t5 = ( t1 , t2 , t3 )  # Tupla de tuplas
```

```
t1 = (5 ,3 ,9)
t1 [0] = 2      # Error
t1 [1] = t1 [1]+1  # Error !
```

```
lista=[1,2,3]
tupla=tuple(lista)
print(lista)
print(tupla)
```

```
t1 = (5 ,3 ,9)
t2 = ('Juan ','Perez ')
t3 = ('Juan ','P´e rez ', 18)
t4 = ( t1 , t2 , t3 )
```

```
print ( t4 [0][2])           # >>> 9
print ( t4 [1][ -1])         # >>> 'Perez '
print ( t4 [2][2])           # >>> 18
```

```
t = (5 ,3 ,9 ,8 ,6 ,5 ,34 ,2)
for i in t :
    print ( i )
```

```
t1 = (1 ,2 ,3)
t2 = (4 ,5 ,6)
print ( t1 + t2 )      # >>> (1 ,2 ,3 ,4 ,5 ,6)
print (2* t2 )         # >>> (4 ,5 ,6 ,4 ,5 ,6)
print (2 in t1 )       # >>> True
print (2 not in t2 )   # >>> True
```



```
t = (5 ,3 ,9 ,8 ,6 ,5 ,34 ,2)
print ( t [:4])  # >>> (5 ,3 ,9 ,8)
print ( t [6:])  # >>> (34 ,2)
print ( t [::-1]) # >>> (2 ,34 ,5 ,6 ,8 ,9 ,3 ,5)
```

```
t = (5 ,3 ,9 ,8 ,6 ,5 ,34 ,2)
print (len( t ) )      # >>> 8
print (max( t ) )      # >>> 34
print (min( t ) )      # >>> 2
```

```
p1 = (0 ,0 ,0) ;  
p2 = (0 ,0 ,1) ;  
p3 = (0 ,1 ,0)  
p4 = (0 ,1 ,1) ;  
p5 = (1 ,0 ,0) ;  
p6 = (1 ,0 ,1)  
p7 = (1 ,1 ,0) ;  
p8 = (1 ,1 ,1)
```

```
L = [ p1 , p2 , p3 , p4 , p5 , p6 , p7 , p8 ]
```

```
L[3]                >>>
```

```
L[3][1]            >>>
```

**Packing:** Reunir varias variables (o valores) en una tupla.

```
nombre = " margarita "  
apellido = " castro "  
correo = " mjcastro@uc .cl«
```

```
# packing  
usuario = nombre , apellido , correo
```

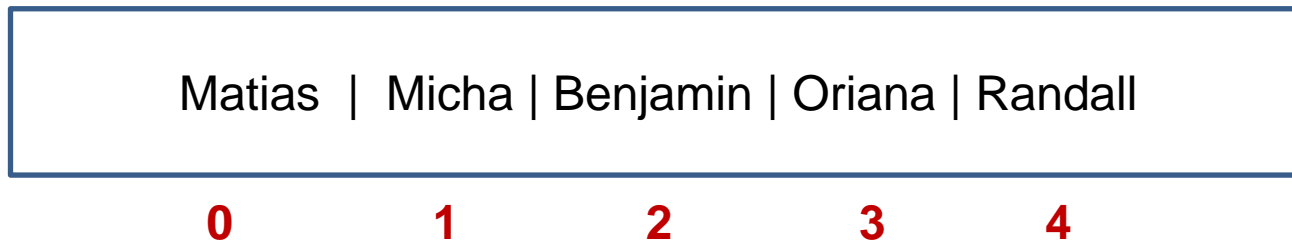
**Unpacking:** Pasar valores de una tupla a varias variables.

```
P = 3 ,6 ,5 3 4  
x ,y , z = P
```

Crear un algoritmo que permita:

- Al usuario ingresar los nombres de los estudiantes de la asignatura de Programación y almacenarlos en una Lista llamada ESTUDIANTES.
- **NO PUEDEN SER INGRESADOS NOMBRES CONTENIDOS EN LA TUPLA VETADOS ('Jennifer', 'Luis', 'Rady', 'Jimmy')**

[ Matias, Micha, Benjamin, Oriana, Randall ]



Des pues de finalizar el ingreso de datos a la lista debe mostrar un menú con las siguientes opciones:

- **BUSCAR:** Permite al usuario ingresar un nombre y buscarlo en lista, indicando si existe o no. Si existe debe indicar en que posición de la lista se encuentra.
- **MOSTRAR TODO:** Permite Mostrar Todo el contenido de la lista .
- **ORDENAR:** Ordenar la lista
- **ELIMINAR:** Pedir al usuario el ingreso de un nombre y eliminarlo de la lista
- **SALIR:** Única forma de salir de la ejecución del menú