

INTRODUCCIÓN A LAS TIC



**Universidad
Tecnológica
del Perú**

DBA(c) Jhon E. Monroy Barrios

Arequipa, 2022

¿Qué es el desarrollo de software?



¿Qué es el desarrollo de software?

Conjunto de actividades informáticas dedicadas al proceso de creación, diseño, despliegue y compatibilidad de software.

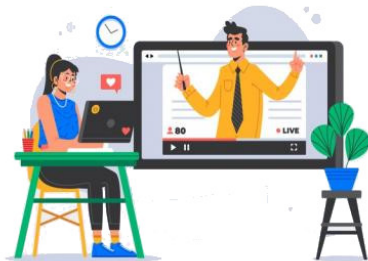


¿Qué es Git y GitHub?



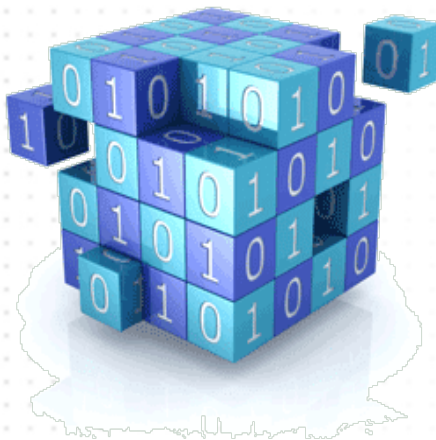
Logros

Al finalizar la sesión, el estudiante conoce la las principales características de una estructura de datos a través de diferentes ejemplos prácticos.



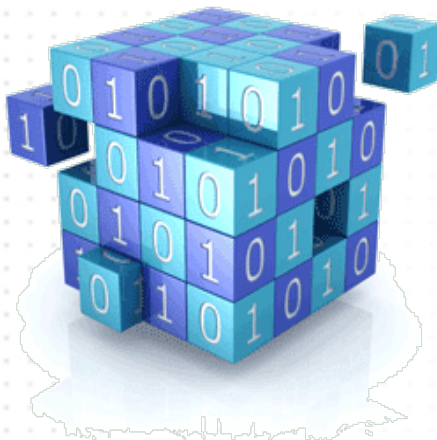
1 Estructura de datos

¿Qué es una estructura de datos?



¿Qué es una estructura de datos?

Es una representación interna de una colección de información.



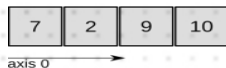
Array

Son estructuras de datos lineales que representan una colección de información, en informática los Arrays comienza en la posición 0 y termina en la posición N-1, siendo este N el tamaño del Array.



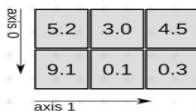
Tipos de Array

1D array



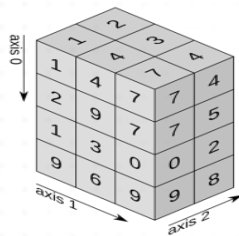
shape: (4,)

2D array



shape: (2, 3)

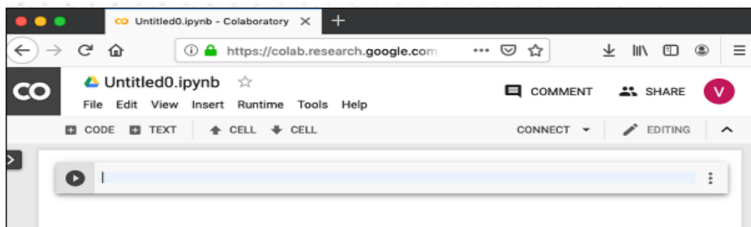
3D array



shape: (4, 3, 2)

¿Qué es Google Colab?

Es un entorno gratuito de Jupyter Notebook que no requiere configuración y que se ejecuta completamente en la nube.



¿Qué es Numpy?

Es una biblioteca para el lenguaje de programación Python que da soporte para crear vectores y matrices grandes multidimensionales, junto con una gran colección de funciones matemáticas de alto nivel para operar con ellas.

NumPy



Arreglos

```
#Importar la librería
import numpy as np
```

```
[ ] #Crear arreglos
np.array([10, 20, 24, 5, 15, 50])
```

```
array([10, 20, 24, 5, 15, 50])
```

```
[ ] #Asignar un arreglo a una variable y seleccionar el índice 4
a = np.array([10, 30, 20, 4, 30, 51, 7, 2, 4, 40, 100])
a[4]
```

```
30
```

```
[ ] #Conocer el tipo de dato
type(a[2])
```

```
numpy.int64
```

```
[ ] #Seleccionar el índice 3 hacia adelante
a[3:]
```

```
array([ 4, 30, 51, 7, 2, 4, 40, 100])
```

```
[ ] #Seleccionar desde el índice 3 hasta el índice 7
a[3:7]
```

```
array([ 4, 30, 51, 7])
```

```
[ ] #Seleccionar desde el índice 1 cada 4 índices
a[1::4]
```

```
array([30, 51, 40])
```

Arreglos

```
[ ] #Crear un arreglo de ceros
np.zeros(5)
```

```
array([0., 0., 0., 0., 0.])
```

```
[ ] #Crear un arreglo de unos de dos dimensiones
np.ones((4, 5))
```

```
array([[1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.],
       [1., 1., 1., 1., 1.]])
```

```
▶ #Crear un arreglo de dos dimensiones
b = np.array( [['x', 'y', 'z'], ['a', 'c', 'e']])
print(b)
```

```
[['x' 'y' 'z']
 ['a' 'c' 'e']]
```

```
[ ] #Conocer el número de dimensiones
b.ndim
```

```
2
```

```
[ ] #Ordenar de menor a mayor
c = [12, 4, 10, 40, 2]
np.sort(c)
```

```
▶ array([ 2,  4, 10, 12, 40])
```

```
[ ] #Crear un arreglo de ceros  
np.zeros(5)
```

```
array([0., 0., 0., 0., 0.])
```

```
[ ] #Crear un arreglo de unos de dos dimensiones  
np.ones((4, 5))
```

```
array([[1., 1., 1., 1., 1.],  
       [1., 1., 1., 1., 1.],  
       [1., 1., 1., 1., 1.],  
       [1., 1., 1., 1., 1.]])
```

```
▶ #Crear un arreglo de dos dimensiones  
b = np.array( [['x', 'y', 'z'], ['a', 'c', 'e']])  
print(b)
```

```
[['x' 'y' 'z']  
 ['a' 'c' 'e']]
```

```
[ ] #Conocer el número de dimensiones  
b.ndim
```

```
2
```

```
[ ] #Ordenar de menor a mayor  
c = [12, 4, 10, 40, 2]  
np.sort(c)
```

```
▶ array([ 2,  4, 10, 12, 40])
```

Arreglos

```
[ ] #Crear un arreglo dinámico de 25 elementos
np.arange(25)
```

```
array([ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24])
```

```
[ ] #Crear un arreglo dinámico con un límite inferior y un límite superior
np.arange(5,30)
```

```
array([ 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
       22, 23, 24, 25, 26, 27, 28, 29])
```

```
[ ] #Crear un arreglo dinámico con un límite inferior y un límite superior e incremento de 5
np.arange(5, 50, 5)
```

```
array([ 5, 10, 15, 20, 25, 30, 35, 40, 45])
```

```
[ ] #Crear un arreglo bidimensional de 3 filas y 5 columnas y el único valor 10
```

```
np.full( ( 3, 5 ), 10)
```

```
array([[10, 10, 10, 10, 10],
       [10, 10, 10, 10, 10],
       [10, 10, 10, 10, 10]])
```

```
[ ] #Crear un diagonal de arreglo tipo bidimensional
np.diag ( [0, 3, 9, 10] )
```

```
array([[ 0, 0, 0, 0],
       [ 0, 3, 0, 0],
       [ 0, 0, 9, 0],
       [ 0, 0, 0, 10]])
```


Estructuras de datos lineales

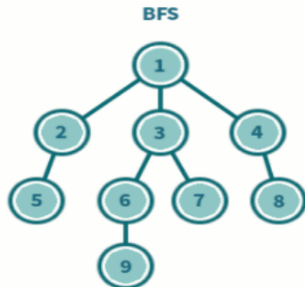
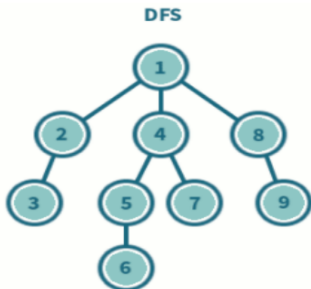
Son aquellas en las que los elementos ocupan lugares sucesivos en la estructura y cada uno de ellos tiene un único sucesor y un único predecesor.

- Pilas
- Colas



Los algoritmos de búsqueda

- BFS (Breadth First Search) se enfoca en atravesar el árbol capa por capa.
- DFS (Depth-first search), simplemente exploramos el camino hasta el final antes de regresar hacia arriba para explorar otro camino.



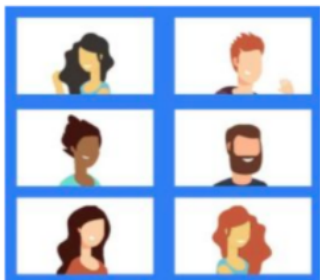
Trabajo en Equipo

Crear un arreglo de uno, dos y tres dimensiones

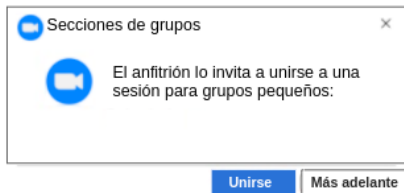


Equipos de Zoom

¿Listos para compartir en equipos de Zoom?



En tu pantalla aparecerá el nombre unirte a una sesión de grupos pequeños. Dale clic en “Unirse”.



Conclusiones

¿Qué aprendiste en esta sesión?

Te invitamos a compartir tus conclusiones en clase.



Conclusiones

Las estructuras de datos son una representación interna de una colección de información.