**ACTIVIDAD #8: PYTHON.**

**ESTRUCTURAS DE DATOS:**

1. **¿Qué son las estructuras de datos?**

En programación, una estructura de datos es una forma de organizar un conjunto de datos elementales (un dato elemental es la mínima información que se tiene en el sistema) con el objetivo de facilitar la manipulación de estos datos como un todo y/o individualmente. Una estructura de datos define la organización e interrelacionamiento de estos, y un conjunto de operaciones que se pueden realizar sobre él. Las operaciones básicas son:

**-** Alta, adicionar un nuevo valor a la estructura.

**-** Baja, borrar un valor de la estructura.

**-** Búsqueda, encontrar un determinado valor en la estructura para ser realizar una operación con este valor, en forma SECUENCIAL o BINARIO (siempre y cuando los datos estén ordenados).

1. **Tipos de estructuras de datos.**

**Arreglos.** Un array es un tipo de dato estructurado que permite almacenar un conjunto de datos homogéneo y ordenado, es decir, todos ellos del mismo tipo y relacionados. Su condición de homogéneo, indica que sus elementos están compuestos por el mismo tipo de dato, y su condición de ordenado hace que se pueda identificar del primer al último elemento que lo compone.

**Listas enlazadas.** Las listas enlazadas se construyen con elementos que están ubicados en una secuencia. Aquí, cada elemento se conecta con el siguiente a través de un enlace que contiene la posición del siguiente elemento. De este modo, teniendo la referencia del principio de la lista podemos acceder a todos los elementos de la misma.

**Pila.** La pila es un tipo especial de lista lineal dentro de las estructuras de datos dinámicas que permite almacenar y recuperar datos, siendo el modo de acceso a sus elementos de tipo LIFO (del inglés Last In, First Out, es decir, último en entrar, primero en salir). ¿Cómo funciona? A través de dos operaciones básicas: apilar (push), que coloca un objeto en la pila, y su operación inversa, desapilar (pop), que retira el último elemento apilado.

**Cola.** Una cola (también llamada fila) es un tipo de dato abstracto, caracterizada por ser una secuencia de elementos en la que la operación de inserción push se realiza por un extremo y la operación de extracción pull por el otro. También se le llama estructura FIFO (del inglés First In First Out), debido a que el primer elemento en entrar será también el primero en salir.

**Árboles.** En estructura de datos, los árboles consisten en una estructura no lineal que se utiliza para representar datos con una relación jerárquica en la que cada elemento tiene un único antecesor y puede tener varios sucesores. Los mismos se encuentran clasificados en: árbol general, un árbol donde cada elemento puede tener un número ilimitado de sub árboles y árbol binario, que son una estructura de datos homogénea, dinámica y no lineal en donde a cada elemento le pueden seguir como máximo dos nodos.

**Grafos.** Otro tipo de no lineal de estructura de datos en programación, son los grafos. Se trata de una estructura matemática formada por un conjunto de puntos —una estructura de datos— y un conjunto de líneas, cada una de las cuales une un punto a otro. Los puntos se llaman nodos o vértices del grafo y las líneas se llaman aristas o arcos.

1. **Estructuras de datos en Python.**

Las estructuras de datos en Python se pueden entender como un tipo de dato compuesto, debido a que en una misma variable podemos almacenar una estructura completa con información. Dichas estructuras, pueden tener diferentes características y funcionalidades. De hecho, existen múltiples tipos de estructuras de datos en Python.

Las estructuras de datos más comunes en Python son las listas, las tablas y los diccionarios. Aunque tienen otro nombre, en otros lenguajes, son los arreglos o vectores, las matrices y los arreglos indexados, respectivamente. Son en esencia lo mismo, aunque como es habitual en Python, con varias facilidades y funcionalidades ya incluidas.

1. **Defina los tipos de estructuras de datos en Python.**

**Listas.**

Las listas en Python son un tipo contenedor, compuesto, que se usan para almacenar conjuntos de elementos relacionados del mismo tipo o de tipos distintos.

Junto a las clases tuple, range y str, son uno de los tipos de secuencia en Python, con la particularidad de que son mutables. Esto último quiere decir que su contenido se puede modificar después de haber sido creada.

Para crear una lista en Python, simplemente hay que encerrar una secuencia de elementos separados por comas entre paréntesis cuadrados [].

Por ejemplo, para crear una lista con los números del 1 al 10 se haría del siguiente modo:

>>> números = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

**Tuplas.**

Una tupla es una colección de objetos ordenados que encierra sus elementos con paréntesis () y los separa con comas. Las tuplas son muy similares a las listas, y pueden almacenar objetos de tipo distinto como enteros y Strings entre otros. Sin embargo, al contrario que las listas presentan la propiedad de inmutabilidad. Esto implica que los elementos de una tupla no pueden reasignarse. El siguiente ejemplo ilustra estos dos conceptos.

>>> t = (1,'dos',3)

>>> type(t)

<class 'tuple'>

>>> t[0] = 'uno'

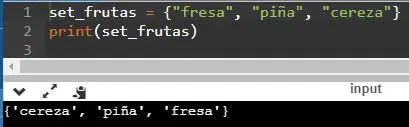
Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: 'tuple' object does not support item assignment

**Sets.**

Un set es una colección que no posee orden, y, por tanto, tampoco números de índex. Esto quiere decir que no podemos decidir el orden en el cual aparecerán sus elementos. En Python, los sets se escriben entre llaves.

****

**Strings.**

Las cadenas (o Strings) son un tipo de datos compuestos por secuencias de caracteres que representan texto. Estas cadenas de texto son de tipo str y se delimitan mediante el uso de comillas simples o dobles.

>>> cadena = "Programa en Python"

>>> type(cadena)

<class 'str'>

**Diccionario.**

Los diccionarios en Python son un tipo de estructuras de datos que permite guardar un conjunto no ordenado de pares clave-valor, siendo las claves únicas dentro de un mismo diccionario (es decir que no pueden existir dos elementos con una misma clave).

Veamos a continuación un ejemplo de cómo declararíamos un diccionario en Python en cuyos elementos tenemos los jugadores de la selección Española, siendo la clave su dorsal (clave única) y el valor su nombre.

# Defino la variable 'futbolistas' como un diccionario. No es necesario declarar que tipo de dato es

futbolistas = dict()

futbolistas = {

1 : "Casillas", 15 : "Ramos",

3 : "Pique", 5 : "Puyol",

11 : "Capdevila", 14 : "Xabi Alonso",

16 : "Busquets", 8 : "Xavi Hernandez",

18 : "Pedrito", 6 : "Iniesta",

7 : "Villa"

}

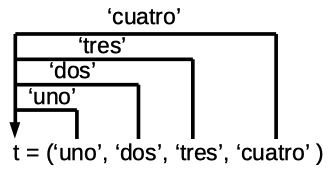
1. **Realice un gráfico o dibujo de los tipos de estructuras de datos.**

**Listas.**

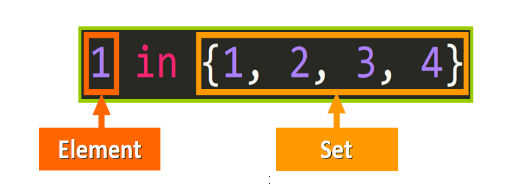
**Escala de tiempo

Descripción generada automáticamente con confianza media**

**Tuplas.**

****

**Sets.**



**Strings.**

Diagrama

Descripción generada automáticamente

**Diccionarios.**

****