**ACTIVIDAD 1 – JESÚS DOMÍNGUEZ**

1. *Herencia y polimorfismo en Python*

La herencia es un procedimiento que nos permite transferir o heredar datos, propiedades y métodos de una clase a otra. La clase que transfiere o entrega una sus elementos suelen llamarse clase padre o super clase o clase base. Y la clase que recibe la herencia se le llama subclase o clase derivada.

En Python, para que una clase B herede los atributos y métodos de una clase A, se debe colocar la siguiente expresión:

class ClaseB(ClaseA):

Cualquier subclase puede heredar tantas clases como lo necesite, esto se conoce como herencia multiple, para que una clase D herede los atributos y métodos de una clase A, B y C, por ejemplo, se debe colocar la siguiente expresión:

class ClaseD(ClaseA, ClaseB, ClaseC):

Por otro lado, el polimorfismo o sobrecarga se podría definir como la capacidad que tienen los objetos de tomar distintas formas, dependiendo del tipo de dato que se ingrese en el constructor o en el método de la clase en la que pertenece dicho objeto

1. *Manejo de Archivos en Python (Texto, Binario)*

El manejo de archivos es un aspecto importante en cualquier lenguaje de programación, se trata de procedimientos que nos permiten leer y modificar archivos de textos, archivos binarios, etc. Los archivos de texto admiten cualquier tipo de texto plano sin formato, pero los archivos binarios pueden contener cualquier tipo de información codificada en binarios.

En Python, para acceder a un archivo de texto o binario es necesario utilizar el método open( ), este método puede recibir varios parámetros, pero siempre el primero indicará el nombre del archivo a abrir y el segundo indicará el modo para abrir dicho archivo. A continuación, se mostrará los algunos modos que podemos utilizar para abrir fichero o archivos de texto o binario:

* “r”, para lectura, posiciona el puntero al principio del archivo y nos devuelve error si no existe el fichero que queremos abrir.
* “a”, para añadir contenido, posiciona el puntero al final del archivo, si no existe el archivo lo crea.
* “w”, para escribir, posiciona el puntero al principio del archivo, si no existe el archivo lo crea.

1. *Persistencia de Objetos en Python*

La persistencia de datos es un procedimiento que nos permite almacenar la información después de finalizar un programa o algoritmo. La mayoría de los programas de la actualidad necesitan conservar datos después de la ejecución de dicho programa.

**Referencias**

*Herencia y Polimorfismo en Python (Ejemplo Simple)*. (2020, 22 diciembre). Mi Diario Python. Recuperado 20 de octubre de 2022, de <https://pythondiario.com/2016/10/herencia-y-polimorfismo-en-python.html>

F, G. (s. f.). *Polimorfismo POO*. Scribd. Recuperado 20 de octubre de 2022, de <https://es.scribd.com/document/490516691/Polimorfismo-POO>

Priya C, B. (2022, 14 marzo). *Cómo manejar archivos en Python*. Geekflare. Recuperado 20 de octubre de 2022, de <https://geekflare.com/es/handle-files-in-python/>

*Tutorial de python – Parte 4 (Manejo de ficheros)*. (2019, 12 diciembre). TecnoPS. Recuperado 20 de octubre de 2022, de <https://tecnops.es/tutorial-de-python-parte-4/>

*TXT*. (2020, 6 enero). Cómo abrir archivos. Recuperado 20 de octubre de 2022, de <https://comoabrirarchivos.com/txt/>

colaboradores de Wikipedia. (2022, 5 octubre). *Archivo binario*. Wikipedia, la enciclopedia libre. Recuperado 20 de octubre de 2022, de <https://es.wikipedia.org/wiki/Archivo_binario>

colaboradores de Wikipedia. (2021, 4 febrero). *Persistencia de datos*. Wikipedia, la enciclopedia libre. Recuperado 20 de octubre de 2022, de <https://es.wikipedia.org/wiki/Persistencia_de_datos>