|  |  |
| --- | --- |
| Nombre: David de Jesús Espinoza Meléndez | Matricula 03103614 |
| Nombre del Curso: Operaciones y Desarrollo | Nombre del Profesor: Viridiana Mares Rodriguez |
| Modulo: | Actividad: Actividad 7 |
| Fecha: 6/3/2025 | |
| Referencias Bibliografícas: | |

Subir a GitHub los archivos generados durante la clase

* act7\_1\_numeros.py
* act7\_2\_modulo.py
* act7\_3\_abstract\_class.py
* act7\_4\_unit\_test.py

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

https://github.com/David04eng/devops

2.- En Word realizar el reporte de la práctica describiendo para cada script que es lo que sucede de manera breve e incluyendo los conceptos que se vieron en cada script.

* act7\_1\_numeros.py (Herencia, polimorfismo y encapsulación)

En el primer código usamos números y racionales, en la clase de números esta print\_numbers que imprime n-1 si n es par o sus cuadrados si es impar mientras que la clase racionales hereda de números, hace que print\_numbers para que n sea fracción y agrega print\_hello.

Hace herencia porque racionales hereda atributos de números, hace polimorfismo porque print\_numbers cambie dependiendo de la clase y encapsulación porque cada clase tiene su salida

* act7\_2\_modulo.py (módulos en Python)

En el segundo código usamos los modulos al importar y reutilizar las clases de act7\_1\_numeros.py. Las clases números y racionales no se tienen que volver a definir. Los modulos facilitan la organización del código, la importación permite funcionalidades de otros scripts y el encapsulamiento hace que se mantenga la lógica

* act7\_3\_abstract\_class.py (Clase abstracta)

En el tercer código la clase abstracta absnumeros con el método abstracto print\_numeros es implementado por las clases que hereden de ella. La clase racionales\_concreto hereda absnumeros e implementa print\_numeros para el valor de n.

La clase abstracta no se puede instanciar directamente, el método abstracto debe ser implementada por las clases hijas. Racionales\_concreto hereda absnumeros mientras que print\_numeros varia según la clase

* act7\_4\_unit\_test.py (Pruebas unitarias)

En el último código el modulo unittest es para las pruebas unitarias. La clase TestStringMethods hereda unittest.TestCase, dentro de ella se implementa test\_upper verificando el método upper().

Las pruebas unitarias permiten que se pueda verificar las unidades individuales del codigo para asegurarnos de que funcionen bien, assertEqual compara el resultado contra lo que se obtuvo. La herencia hace que TestStringMethods herede unittest.TestCase

3.- Añade a la clase "Racionales" el método mostrado abajo y con base en el script "act7\_4\_unit\_test.py" crea un nuevo script llamado "act7\_test\_racionales.py" que contenga un método que pruebe la función "print\_hello()



