









		RITE	RIOS DE EVALUACIÓN TRABAJO FI	NAL
ID SECCIÓN	SECCIÓN	ID	PUNTO A EVALUAR	AYUDA
1	VALIDACIONES TRANSVERSALES	1	MANEJO DE TYPING	Manejo de la libreria <i>typing</i> . Que cada variable, parámetro o retorno tenga tipo de dato, incluidas las estructuras complejas. Eje: <i>edad: int = 0 , List[int]</i>
		2	MANEJO DE LOGGIN	Contar con un apropiado uso de la librería <i>logger</i> para comunicar apropiadamente al usuario el comportamiento del script
		3	MANEJO DE DOCSTRINGS	Documentación apropiada del código, utilizando un estandar definido y conocido de <i>docstring</i>
		4	NO VALORES HARCODEADOS	Evitar valores <i>hardcoded</i> o directamente implícitos en el código
2	PROGRAMACIÓN ORIENTADA A OBJETOS	5	ENCAPSULAMIENTO	Correcto manejo del encapsulamiento en objetos y clases, mediante la clara definición de objetos públicos, semipúblicos y privados
		6	HERENCIA	Capacidad del código de ser reutilizable y extendible
		7	POLIMORFISMO	Hace referencia a que objetos de diferentes clases respondan de manera uniforme a un mismo mensaje o método
		8	ABSTRACCIÓN	Hace referencia a la simplificación de objetos o procesos complejos al enfocarse en los aspectos esenciales y ocultar los detalles innecesarios
		9	DATACLASSES O PYDENTIC	Hace referencia a la validación de atributos para reducir la complejidad de la lógica como condicionales o programas muy grandes
			PROPERTIES	Para manejo de <i>getters</i> y <i>setters</i> . En otras palabras, facilitan la asignación u obtención de valores a los atributos de una clase Python
			EXCEPCIONES	Manejo adecuado de excepciones dentro del código
		12	CONSTRUCTORES  MANEJO DE MÉTODOS	Manejo de inicializadores de clase
		13	ESPECIALES	Hace referencia a la inclusión de alguno de estos métodos especiales:rep ,str ,call , etc.
		14	DECORADORES	Usar decoradores como funciones o clases para extender alguna funcionalidad
		15	MANEJO DE STATIC METHOD	Hace referencia a funciones asociadas a una clase en lugar
		16	MANEJO DE CLASS METHOD	de una instancia específica de la clase  Hace referencia a funciones asociadas a una clase en lugar de una instancia específica de la clase. Pueden acceder y modificar atributos de la clase, pero no de instancias individuales
3	ESTRUCTURA DE PROYECTO	17	MANEJO DE PAQUETES Y MÓDULOS	Definición de la estructura del proyecto de código
		18	MANEJO DE ENTRY POINT	Hace referencia al punto de entrada de la aplicación para ejecutar el programa
		19	MANEJO DE AMBIENTES VIRTUALES	Poetry , Pipfile , pyenv o Conda
		20	MANEJO DE RUTAS MULTI OS	Uso de os ó pathlib
		21	MANEJO DE DEPENDENCIAS	Manejo adecuado de los <i>imports</i> , y limitación apropiada de cargar solamente los objetos necesarios de las librerías importadas
		22	MANEJO DE CONFIGURACIÓN DE APLICACIÓN	Uso de archivos de configuración, json , yaml , toml , etc.
		23	MANEJO DE CONFIGURACIÓN	Uso de archivos . <i>cfg</i> para manejo de librerías. Eje. <i>flake8</i> ,
			DE COMPONENTES  MANEJO DE PARSEARGS	manejo de coverage Entrada de parámetros al ejecutar el código
			DISTRIBUCIÓN ADECUADA O	Manejo adecuado de paquetes y módulos (evitar la
		25	CON ORDEN	sobrecarga en un solo paquete o módulo)
		26	README	Documentación de aplicación o información relevante del proyecto
	VALIDACIONES PROYECTO	27	COVERAGE TESTS	Validar si las pruebas unitarias cubren el 60% de coverage
		28	PRUEBAS UNITARIAS	Manejo de pruebas unitarias para garantizar el correcto funcionamiento de los componentes
4		29	FLAKE8	Verificar el código fuente contra el PEP8
			PREGUNTAS DISRUPTIVAS	Validar los problemas o puntos de dolor actuales del proyecto, en otras palabras, validar los peores escenarios
		31	GIT	Verificar que el proyecto se encuentre un repositorio público o privado
		32	PREGUNTAS ALEATORIAS PARA VALIDAR CONOCIMIENTO	Se refiere a hacer preguntas de forma aleatoria con el objetivo de evaluar la comprensión o el dominio del tema por parte de todos los integrantes del equipo