**🧠 Proyecto Lógica: Katas Python**

**📖 Descripción del Proyecto**

Este proyecto tiene como objetivo aplicar los conocimientos adquiridos en el módulo de Python a través de la resolución de una serie de ejercicios prácticos conocidos como katas. Las katas permiten perfeccionar habilidades de programación mediante la repetición y resolución de problemas concretos, lo cual resulta ideal para consolidar conceptos fundamentales.

Cada kata resuelta aborda una problemática diferente, utilizando funciones integradas de Python como map(), filter(), reduce(), y el manejo de excepciones, clases, funciones lambda y recursividad. El propósito principal ha sido desarrollar soluciones limpias, reutilizables y eficientes para mejorar la lógica de programación.

📁 **Estructura del Proyecto**

├ EnunciadoDataProjectPython.pdf # Documento con el listado de katas propuestas

├ proyecto.py # Archivo con todas las soluciones de las katas

├ README.md # Descripción completa del proyecto (este archivo)

**🛠 Instalación y Requisitos**

Este proyecto está desarrollado en Python desde Visual Studio Code (VSCode). Para ello, se requiere instalar la extensión de Python para ejecutar el código Python dentro del programa.

Para ejecutar el código, simplemente abre una terminal y ejecútalo como proyecto.py para que lo identifique como archivo Python.

No requiere de librerías externas, ya que todas las funciones se implementan utilizando únicamente módulos estándar del lenguaje.

**📊 Resultados y Conclusiones**

Durante el desarrollo del proyecto se resolvieron 40 katas diferentes. A continuación, se resumen los principales aprendizajes aplicados:

* **Funciones de orden superior:** Se usaron ampliamente map(), filter() y reduce() para transformar, filtrar y reducir estructuras de datos de forma concisa.
* **Expresiones lambda:** Se integraron funciones anónimas para operaciones rápidas dentro de llamadas a funciones como map.
* **Estructuras de control y condicionales:** La lógica condicional fue clave para gestionar validaciones, cálculos y decisiones en los programas.
* **Gestión de errores:** Se manejaron excepciones como ValueError y ZeroDivisionError para ofrecer un comportamiento seguro frente a entradas inválidas.
* **Programación orientada a objetos:** Se diseñaron clases como Arbol y UsuarioBanco que simulan escenarios reales de manera dinámica y estructurada.
* **Procesamiento de texto y listas:** Se aplicaron técnicas para analizar y modificar cadenas de texto, identificar patrones y filtrar contenido.

**Katas destacados:**

* **Kata 5**: Evaluación de una lista de calificaciones con resultado aprobado o suspenso.
* **Kata 17**: Conversión de una lista de dígitos a un número entero utilizando reduce().
* **Kata 18**: Filtrado de estudiantes con nota ≥ 90 usando filter() y estructuras de diccionario.
* **Kata 30**: Detección de anagramas comparando letras ordenadas de dos palabras.
* **Kata 36**: Simulación de transferencias bancarias y operaciones entre usuarios con control de errores.
* **Kata 37**: Procesador de texto con tres funcionalidades: contar palabras, reemplazar términos o eliminarlos dinámicamente.

**🔄 Próximos Pasos**

Aunque el proyecto está completo, hay ideas para seguir desarrollándolo:

* Implementar pruebas unitarias automáticas para validar cada solución.
* Crear un menú interactivo desde consola para seleccionar y ejecutar una kata.
* Añadir documentación técnica para cada función (docstrings).
* Separar las katas en módulos individuales para facilitar la organización.

**🤝 Contribuciones**

Si te interesa mejorar o extender este proyecto, las contribuciones son bienvenidas. Puedes sugerir nuevas katas, proponer mejoras en el código o ayudar a documentar mejor las funciones.

**✒ Autores y Agradecimientos**

Autor: Francisco Troyano Martínez.

Contacto: [troyano1406@gmail.com](mailto:troyano1406@gmail.com) .

GitHub: <https://github.com/trhoudini>

Proyecto realizado como parte del módulo de Python en el programa de formación de Data Analytics.