

Evaluación del módulo 3

Consigna del proyecto 

Evaluación del módulo

Proyecto: Sistema de gestión de datos

Situación inicial 📌

Unidad solicitante: Equipo de desarrollo de una empresa de tecnología

📌 En el ámbito del desarrollo de software, es fundamental contar con conocimientos sólidos en **Python** para resolver problemas de manera eficiente. La empresa ha identificado la necesidad de mejorar la automatización de sus procesos internos y ha solicitado la creación de un sistema en Python que implemente **estructuras de control, funciones y manejo de datos**.

El equipo de desarrollo será responsable de diseñar y construir un **sistema de gestión** que permita la manipulación de datos de manera eficiente, utilizando las mejores prácticas vistas en el módulo.

Nuestro objetivo 📋

El objetivo principal del proyecto es diseñar e implementar un **sistema de gestión basado en Python**, aplicando los conocimientos adquiridos en el módulo. Este sistema permitirá manejar información de manera estructurada, utilizando **condicionales, bucles, funciones, estructuras de datos y módulos** para optimizar el procesamiento de datos y la toma de decisiones.

Requerimientos 🤝

El sistema debe incluir las siguientes funcionalidades:

- ↳ **Entrada y salida de datos:**
 - Captura y visualización de información ingresada por el usuario.
 - Formateo de salida con `print()` y f-strings.
 - Validación de entradas para evitar errores en los datos.
- ↳ **Estructuras de control:**
 - Implementación de condicionales (`if`, `elif`, `else`) para la toma de decisiones.
 - Uso de bucles (`for`, `while`) para iteraciones eficientes.
 - Aplicación de `break` y `continue` para mejorar el flujo de control.
- ↳ **Funciones:**

- Creación de funciones personalizadas para modularizar el código.
- Uso de parámetros y valores de retorno (return).
- Implementación de funciones recursivas para cálculos específicos.

↳ Estructuras de datos:

- Uso de listas (list) para almacenar y manipular conjuntos de datos.
- Aplicación de diccionarios (dict) para gestionar información en pares clave-valor.
- Implementación de tuplas (tuple) para datos inmutables y organizados.
- Uso de conjuntos (set) para evitar elementos duplicados.

↳ Modularización:

- Creación de módulos para organizar el código de manera eficiente.
- Uso de import para reutilizar funciones y scripts en diferentes archivos.
- Organización del código en archivos .py separados para mejorar la mantenibilidad.

¿Qué vamos a validar?

Para garantizar la calidad del proyecto, se evaluarán los siguientes aspectos:

Aspectos técnicos:

- **Código limpio y bien estructurado:** Uso adecuado de indentación y nombres de variables según PEP 8.
- **Uso de estructuras de control:** Correcta implementación de condicionales y bucles.
- **Manejo eficiente de datos:** Implementación correcta de listas, diccionarios y otros tipos de datos.

Aspectos estructurales:

- **Modularización del código:** Uso de funciones y módulos para mejorar la organización del código.
- **Legibilidad y documentación:** Inclusión de comentarios explicativos y estructura clara.

Aspectos de performance:

- **Eficiencia en la ejecución:** Evitar redundancias y optimizar el uso de estructuras de datos.

- **Validaciones adecuadas:** Manejo correcto de excepciones y validaciones de entrada.

Referencias 🚒

- Sturtz, D. (s.f.). Python Data Structures. Real Python. <https://realpython.com/python-data-structures/>
- Sturtz, D. (s.f.). Python Modules and Packages: An Introduction. Real Python. <https://realpython.com/python-modules-packages/>
- Van Rossum, G., Warsaw, B., & Coghlan, N. (2001). PEP 8 – Guía de estilo para el código Python. Python Enhancement Proposals. <https://peps.python.org/pep-0008/>

Recursos 📁

<https://www.w3schools.com/python/>

Entregables ✅

- 1 **Código fuente** del proyecto con una estructura modular y comentarios explicativos.
- 2 **Documentación técnica** que incluya la descripción del sistema, estructuras de datos utilizadas y funcionalidades implementadas.
- 3 **Archivo de prueba (.txt o .csv)** con datos de entrada y salida generados por el sistema.
- 4 **Informe de validación** con capturas de pantalla y análisis del proceso de desarrollo.
- 5 **Presentación final** con una descripción del proyecto y las tecnologías utilizadas.

Portafolio 📁

¡Felicitaciones! 🎉 Este proyecto puede ser incluido en tu portafolio para demostrar tu capacidad de resolver problemas utilizando Python. Se recomienda documentarlo en GitHub y elaborar un README con detalles del proyecto, explicando los desafíos enfrentados y las soluciones implementadas.

Este proyecto no solo será una evidencia de tu aprendizaje en Python, sino también una excelente forma de mostrar tu capacidad de desarrollar soluciones efectivas y escalables. 🚀🐍

¡Éxitos!

Nos vemos más adelante

