

Python

Actividad Individual: Programación estructurada

Trabajo Práctico Integrador

Nro: 01

Consideraciones: Leer los orientadores de las clases

En este trabajo integrador, deberás aplicar todos los conceptos planteados en cada una de las clases. en este sentido, debes buscar el modo de aplicar cada concepto publicado y sus variantes. Utilizaras los principios de diseño de algoritmos aplicados a una solución general con Python.

Material para entregar (Excluyente)

- 1- Carpeta comprimida con el código fuente completo
- 2- documento (.doc) con informe o documentación que describa el problema abordado, los algoritmos diseñados, la implementación del sistema y los resultados obtenidos.
- 3- Video con corrida del código y descripción del proyecto. el video no debe tener más de 3 minutos y detallará los aspectos positivos que considera pertinentes.

Instrucciones:

1. Selecciona un problema real o ficticio que te interese resolver utilizando programación.
2. Analiza el problema y descomponerlo en tareas más pequeñas y manejables. Identifica los requisitos y funcionalidades clave que debe incluir tu sistema para resolver el problema.
3. Implementa tu sistema utilizando programación estructurada en el lenguaje de programación que hayamos estado utilizando en clase (por ejemplo, Python).
4. Divide tu código en funciones o procedimientos lógicos que aborden cada tarea identificada. Mantén tu código modular y legible.
5. Utiliza comentarios adecuados para documentar tu código y explicar el propósito de cada función, así como cualquier decisión de diseño importante.
6. Realiza pruebas exhaustivas de tu sistema para garantizar su funcionamiento correcto y la validación de los resultados.
7. Prepara un informe o documentación que describa el problema abordado, los algoritmos diseñados, la implementación del sistema y los resultados obtenidos.

8. Presenta tu trabajo práctico en la fecha establecida, explicando tu solución y respondiendo cualquier pregunta que pueda surgir.

Recursos:

- Material de clase sobre programación estructurada y diseño de algoritmos.
- Documentación y ejemplos del lenguaje de programación utilizado.
- Herramientas de desarrollo y depuración de código.

Criterios de evaluación

- Originalidad de la Propuesta:
 - a. La solución propuesta aborda el problema de una manera novedosa o creativa
 - b. El enfoque utilizado para resolver el problema demuestra originalidad y pensamiento crítico y no se encuentran similitudes con otros proyectos del curso
- Integridad Intelectual del Contenido:
 - a. El trabajo muestra evidencia de investigación y comprensión del problema y su contexto
 - b. No se evidencia copias o plagios en la presentación
- Eficiencia y Eficacia de la Solución:
 - a. La solución propuesta es eficiente en términos de uso de recursos computacionales y tiempo de ejecución y respeta las buenas practicas planteadas en el curso
- Claridad y Legibilidad del Código:
 - a. El código es claro, legible y está bien organizado
 - b. Se han utilizado nombres de variables y funciones descriptivos y significativos
 - c. El código incluye comentarios adecuados que explican su funcionalidad y decisiones de diseño
- Documentación y Presentación:
 - a. El informe o la documentación proporcionada son claros, completos y bien estructurados
 - b. El estudiante ha explicado adecuadamente el problema abordado, los algoritmos diseñados y la implementación del sistema