

Ejercicio – Cifrador Vigenere

Empleando la metodología de desarrollo TDD, y trabajando en los grupos de trabajo previamente conformados, resuelva:

- Genere un reporte de seguimiento de la metodología de TDD en el desarrollo de un cifrador Vigenere compatible con textos en el idioma inglés, que contengan múltiples párrafos de extensión, mayúsculas, minúsculas, números, espacios, y signos de puntuación.
- Para la selección de los cuatro (4) textos, se emplearán como fuente dos discursos disponibles en <https://highspark.co/famous-persuasive-speeches/>
- El cifrador debe funcionar en convencional y en modo generación de llave.
 - En el modo convencional, se suministran como entradas un texto (a escoger entre las cuatro opciones), y se suministra un archivo que contiene la llave conocida por el usuario. Como salidas, el cifrador producirá el texto encriptado (cifrado), y el texto desencriptado empleando la llave suministrada. En esta fase del desarrollo - aunque sea una vulnerabilidad evidente-, se asume que la llave está en texto plano, y que, de hecho, esta ha sido generada por el propio producto de software y almacenada en el archivo, cuando este trabaja en modo generación de llave.
 - En el modo de generación de llave, se le pregunta al usuario por la longitud (cantidad) de símbolos que contendrá la llave. Para este ejercicio se usará una longitud de llave que sea potencia de 2, y cuya longitud esté en el siguiente intervalo [8, 128]. El producto de software debe verificar la idoneidad del dato suministrado por el usuario.
- **Entregables:**
 - Adicional a lo aquí mencionado, se espera que se sigan los lineamientos de entrega discutidos en la clase.
 - Un (1) producto de software, desarrollado en un (1) lenguaje bajo el paradigma orientado a objetos, que satisfaga la funcionalidad anteriormente mencionada.
 - Descripción de la estrategia de pruebas diseñada y empleada.
 - Reporte del seguimiento de la metodología TDD, acorde con la estrategia anteriormente descrita.
 - Breve manual o instrucciones para el usuario sobre cómo ejecutar el producto y descripción de la estrategia de pruebas.
 - Diagramas de clases y de secuencia de la solución y el código que genera los diagramas (empleando *plantuml* o *mermaid*).
 - Evidencia del uso de analizadores estáticos de código (e.g., cppcheck, pylint) y de cómo su uso sirvió para mejorar la calidad del código (e.g., mediante capturas de pantalla durante la transición de VERDE a AZUL).
 - Archivo PDF conteniendo los elementos anteriormente mencionados.
 - Carpeta con código fuente que incluye, por lo menos tres (3) clases, enforzando SRP.
 - Archivo .ZIP con documentos anteriormente mencionados.
- **Medio y plazo de entrega:** Sólo se reciben entregas en el Campus Virtual del curso, antes de las 00:00 horas del miércoles 01 de mayo de 2024. El archivo .zip enviado debe pesar menos de 10MB, y debe contener sólo lo solicitado.