

Definición Proyecto #2

UNAH 2020 - PAC I - AED - Proyecto #2

Buen día a todos. Favor hacer lectura de las indicaciones debajo para el desarrollo y entrega de su segundo y último proyecto. El texto es extenso sin embargo es necesario para entender, desarrollar y completar el proyecto. Para este tercer parcial se hace un cambio de la calificación del proyecto el cual ahora corresponde con el 65% de la nota del parcial (originalmente representaba el 40%), donde ahora la tarea #1 corresponde con el 5% de la nota del parcial, y donde el examen #3 ahora corresponde con el 30% de la nota del parcial.

Objetivo:

- Hacer un programa que permita encriptar/desencriptar un archivo, múltiples archivos o un directorio de archivos, generando un log con formato de tablas HTML, generando un log con el contenido del grafo, y mostrando un grafo con la información de los directorios y tipos de archivos procesados en el proceso.
- Cada equipo centralizará su desarrollo en el uso de TDA's, en el uso de algoritmos avanzados, en el uso de estructuras de datos, y el uso de algoritmos sobre esas estructuras de datos.
- Su proyecto es de investigación y desarrollo, aplicando las teorías y temas de la asignatura.
- La calificación de su proyecto será entregada de acuerdo con la calidad funcional, la calidad visual, la adecuada abstracción e implementación de Tipos de Datos Abstractos, la adecuada abstracción e implementación de las Estructuras de Datos, y sobre el cumplimiento de los requerimientos que se listan en este documento.

Requerimientos y restricciones:

- No se acepta el uso de librerías, frameworks o código adicional de terceros que no se mencionan en esta especificación, y se espera que su proyecto sea programado completamente por usted y su equipo de trabajo. No se aceptará plagio ni desconocimiento del código en ningún escenario. Existirá una entrevista donde se consultará a individuos del equipo sobre el funcionamiento del proyecto.
- Toda operación (recorrer, encriptar, etc) se debe hacer sobre TDA's puros para implementar los grafos y los algoritmos avanzados.
- El sistema debe ser funcional en Linux Ubuntu 16.04 con Python 3.7, PyQt5 y Matplotlib, aplicando las metodologías, nomenclaturas y técnicas enseñadas en clase.
- El sistema debe tener un aspecto visual aceptable en comparación con las imágenes de demostración del proyecto que se adjuntan en este documento.
- Debe tener evidencia en fotografías del análisis de toda la aplicación. Corresponde con incluir documentación escrita de los pasos de los procedimientos o algoritmos internos dentro de su proyecto. Cada acción (encriptar, guardar, etc) corresponde con un algoritmo. Esta evidencia debe ser imágenes png, jpg o archivos PDFs de calidad razonable.
- Documentación interna en el código mediante comentarios de múltiples líneas en cada función, clase o algoritmo.

- Equipo: mínimo 2, máximo 3 personas. Para promover el trabajo en equipo no se aceptarán trabajos individuales. Este cambio se aplica debido a que la cantidad de personas que parecen continuar en la clase (y que se conectan a las clases virtuales) se ha reducido considerablemente durante el periodo.
- Fecha y hora de Entrega: 30 de Abril de 2020, 11:55pm.
- Dentro de su codificación debe usar TDA's para abstraer de la realidad conceptos que resuelven el objetivo planteado.
- El proyecto debe entregarse como un archivo con compresión ".7z" en un enlace dentro del Campus Virtual de la asignatura. Dentro del archivo debe haber un directorio Integrantes, Algoritmos, Librería y un directorio Python, cada uno con el contenido respectivo. El directorio Librería debe poseer un archivo readme.txt el cual debe contener: el nombre de la
- , junto con los comandos de consola para instalar dicha librería en Linux (anteriormente se mencionó la versión de Linux que es compatible con el proyecto). Los integrantes deben ir como una fotografía de la "lista de integrantes" (similar a la que se usa a diario en clase con nombre de la clase, sección y fecha) con los nombres completos, números de cuenta, y firmas de cada integrante. El nombre del archivo 7z debe ser AED-[SECCION]-ProyectoNo2.7z donde debe reemplazarse [SECCIÓN] por la sección de su clase.
- El directorio Python debe tener dentro: un archivo main.py que ejecuta todo el proyecto, un directorio Núcleo que contenga todas sus clases, métodos de clase y algoritmos en código Python3, junto a un directorio LOGS que contendrá todos los archivos de resumen individuales de los encriptados y desencriptados que ha realizado el usuario.
- Su proyecto tiene que tener una evidencia de Video*** la cual deberá subirse como un archivo separado en la entrega del proyecto usando el Campus Virtual.
- Al entregar su proyecto debe incluir al menos algún archivo de prueba de LOG HTML y LOG Json.

Objetivo Secundario:

- Promover la investigación de componentes visuales GUI de python mediante Linux y Qt5.
- Integrar la operación de estructuras de datos (grafos) junto con otros aprendizajes de la materia, dentro del desarrollo de proyectos.
- Promover el trabajo colaborativo entre miembros del mismo equipo para alcanzar una meta en común.

***Su proyecto debe contener una evidencia de funcionamiento en video:

- Máximo 3 minutos de duración.
- Debe contener una demostración del funcionamiento visual de su proyecto (botones, acciones, compresión, etc).
- Una única persona en el video debe explicar de forma rápida, secuencial y superficial el flujo de programa.

- El video debe tener un máximo de 20MB de tamaño con calidad decentemente visible.
- El video debe ser grabado mediante "captura de pantalla" no mediante cámara de celular.
- El video puede tener la voz de una única persona, o en su lugar textos de explicación.
- Antes de enviar su proyecto verifique que el audio del proyecto sea de buena calidad y que se escuche claramente la voz y el contenido del expositor.

En Apoyo a Qt5:

- Qt5 es un Framework para ambiente gráfico, el mismo usado para el proyecto #1.
- Se debe instalar Qt5 en linux.
- Se debe instalar pip3 en linux,
- Se debe instalar Qt5 en Python3 usando Pip3.
- Usar el qt5 designer para generar rápidamente los componentes gráficos.

Pantalla Principal:

- Cada vez que se inicie el programa la pantalla principal se debe abrir en el centro de la pantalla.
- Debe existir un botón "acerca de" que abre una nueva ventana en el centro de la pantalla con la información de los integrantes del equipo autores del programa (nombres, números de cuenta de los autores junto con el nombre de la clase, sección, periodo académico, año y catedrático).
- Debe existir un campo para seleccionar un archivo, múltiples archivos o directorio origen (RutaOrigen de ahora en adelante) en el disco duro, mediante un dialogbox. El archivo o directorio que usted seleccione mediante el dialogbox será el que será procesado por el sistema.
- Debe existir un campo para seleccionar un directorio destino (RutaDestino de ahora en adelante) en el disco duro, mediante un dialogbox. El directorio seleccionado guardará todos los archivos encriptados guardando la forma original de la estructura de directorios del directorio origen.
- Debe existir un botón ejecutar encriptación que aplique todo el proceso de encriptado junto con los cálculos solicitados en este requerimiento (generación del log, generación del grafo, etc).
- Debe existir un botón ejecutar desencriptación que aplique todo el proceso de desencriptado junto con los mismos cálculos solicitados en la encriptación.
- La encriptación de los archivos se debe hacer mediante dos formas: la primera mediante una librería Python3 que encripte AES-256, y la segunda mediante un algoritmo propio desarrollado por los estudiantes.
- Una vez iniciada la operación de encriptado o desencriptado, el sistema debe mostrar una "ventana cargando" en el centro de la pantalla, ya que esta operación puede tardar vario tiempo. Al finalizar la operación se debe cerrar automáticamente esta ventana.
- Una vez finalizada la operación de encriptado o desencriptado, deberán aparecer 2 nuevas ventanas en la pantalla. La primer ventana debe contener la información del resumen de la operación (esta es la misma

información que se debe guardar en el archivo HTML de LOG) usando una tabla de Qt5. La segunda ventana debe contener un grafo con la información de la encriptación, donde el grafo debe mostrar los vértices directorios y una relación entre los directorios detectados contra los vértices “archivo” asociados con dicho directorio. Debe distinguir con colores y otras técnicas, los tipos de vértice (directorio, archivo). Cada vértice de un tipo específico de archivo debe mostrar la información de la cantidad de archivos procesados con ese tipo de extensión y el tamaño total en KB de la información encriptada o desencriptada.

- Debe existir un botón Mostrar Grafo desde archivo JSON que permita cargar un archivo JSON desde el campo RutaOrigen. Este botón sólo puede funcionar si RutaOrigen es un archivo de extensión json. Al presionar este botón se debe abrir la ventana de grafo mostrando la información cargada desde el disco duro.
- Debe existir un campo contraseña el cual es obligatorio para encriptar o desencriptar.

“Operación” Encriptar:

- Al encriptar la RutaOrigen, su programa deberá crear en la RutaDestino un directorio con el nombre AES_256, y un segundo directorio con el nombre OWNAlgorithm. Cada directorio debe tener internamente la misma estructura de la RutaOrigen.
- Cada archivo dentro de los directorios ya encriptados debe poseer el mismo nombre del archivo original pero ahora con extensión “enc” (e.g.: el archivo “Marvel - Avengers - End Game.mp4” se encriptará como “Marvel - Avengers - End Game.mp4.enc”; el archivo “NIN - GHOST V Together - Together.mp4.enc” se encriptará como “NIN - GHOST V Together - Together.mp4.enc.enc”).

“Operación” Desencriptar:

- Al desencriptar la RutaOrigen, su programa deberá crear en la RutaDestino un directorio con el nombre DESC_AES_256, y un segundo directorio con el nombre DESC_OWNAAlgorithm. Cada directorio debe tener internamente la misma estructura de la RutaOrigen.
- Cada archivo dentro de los directorios ya desencriptados debe regresar a su extensión original eliminando el “enc” (e.g.: el archivo “Marvel - Avengers - End Game.mp4.enc” se desencriptará como “Marvel - Avengers - End Game.mp4”; el archivo “NIN - GHOST V Together - Together.mp4.enc.enc” se desencriptará como “NIN - GHOST V Together - Together.mp4.enc”).

Operaciones en Disco:

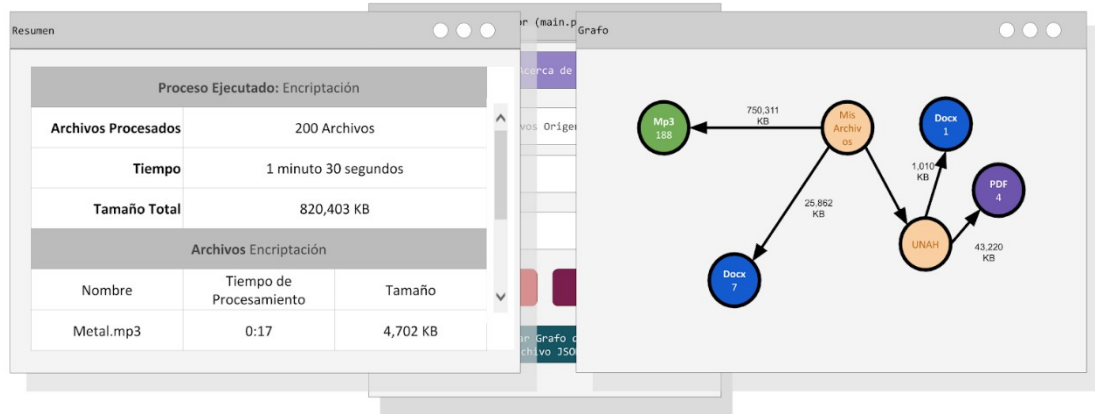
- Cada operación de encriptado o desencriptado debe generar un archivo LOG. En el disco duro, dentro de un directorio LOGS, se deben ir acumulando con un nombre de archivo único donde cada archivo único es un reporte en HTML de la información generada en cada proceso de encriptado o desencriptado. El reporte HTML debe tener la misma información que se mostró en la ventana de resumen. Tome en consideración que el reporte HTML debe tener internamente un formato de Tablas HTML. El nombre único de cada archivo de LOG debe ser:

“LOG_AÑO_MES_DÍA_HORA_MINUTO_SEGUNDO.html” (e.g.:
LOG_2020_03_28_07_12_25.html).

- Con cada operación de encriptado o desencriptado se debe generar un archivo LOG con el contenido Json del grafo mostrada en la pantalla grafo. El nombre único de cada archivo de log debe ser:
“LOG_AÑO_MES_DÍA_HORA_MINUTO_SEGUNDO.json” (e.g.:
LOG_2020_03_28_07_12_25.json). Este archivo LOG Json se debe poder cargar en el sistema mediante el botón Mostrar Grafo desde archivo JSON.

Interfaz Gráfica:

- Los tipos de botones, ventanas, campos de texto y otros componentes gráficos los deberá “imitar” desde las imágenes de muestra de este proyecto, sin embargo, usted tomará sus propias decisiones sobre la ubicación y estilo general del programa.



José Inestroza, PAC I 2020, AED

The screenshot shows the main interface of the 'Encriptador/Desencriptador (main.py)' application. It features several input fields and buttons for file encryption and decryption.

Encriptador/Desencriptador (main.py)

Acerca de

Ruta, Archivo o Archivos Origen **Seleccionar**

Ruta Destino **Seleccionar**

Contraseña

Ejecutar Encriptado **Ejecutar Desencriptado**

Mostrar Grafo desde archivo JSON