

Fecha de entrega: 21 de Octubre de 2021, hasta las 09:00 pm

Leer atentamente las instrucciones de entrega.

1. Considere la carpeta adjunta **data_proof**, es una carpeta que contiene diferentes imágenes que por error se han replicado múltiples veces. Usted debe implementar un algoritmo en C que permita depurar las imágenes de forma automática. El objetivo es crear una nueva carpeta donde se guarde una copia única y sin repetición de cada escena mostrada en las imágenes.

Se debe ilustrar el proceso completo desde el diseño hasta la implementación. Considerar para esto los 4 pasos del Proceso Creativo: Analizar, Diseñar, Elaborar y Probar.

Deberán entregar un documento en PDF donde ilustren los primeros pasos mediante texto, gráficos, esquemas o diagramas de flujo, que consideren necesarios para dejar clara la estrategia de solución del problema.

Deben entregar el código fuente en lenguaje C que se probara con una carpeta que tiene imágenes en orden y con nombre diferente al que se les ha entregado.

Se puede realizar en parejas, teniendo en cuenta que su compañero debe tener la misma modalidad en que han matriculado el curso.

2. [Únicamente para los estudiantes de profundización] Existe un software que permite recibir datos de una tarjeta de Audio, sin embargo para garantizar que los datos se reciban y luego se guarden de forma correcta, el software debe tener unos parámetros de configuración. Dichos parámetros los puede definir el usuario usando un archivo de texto que tiene la siguiente información, no necesariamente en el orden que se presenta a continuación:

```
NUM_CHAN 2
SAMPLE_RATE 16000
BITS_SAMPLE 16
FORMAT wav
TYPE 1-PCM
ENCODING little-endian
```

El formato es: PARÁMETRO[ESPACIO]VALOR

NUM_CHAN indica el numero de canales que se grabaran, es un valor entero.

SAMPLE_RATE es un parámetro que indica la velocidad a la cual la tarjeta de audio debe capturar una muestra. Se mide en Hertz, y puede tomar valores como 8000, 16000, 44100, 48000.

BITS_SAMPLE es un parámetro que me permite definir cuantos bits se usaran por cada valor recibido de la tarjeta de Audio, mayor cantidad de bits implica también mejor calidad de audio, pero también indica archivos de mayor tamaño. Pueden ser 8, 16, 32 o 64 bits.

FORMAT indica el formato en que se guardara el archivo una vez se termine la grabación, típicamente se usa wav, ogg, mp3.

TYPE indica el tipo de modulación digital de las muestras de audio, se pueden tener los siguientes formatos: 1-PCM, 3-IEEE Float, 6-mulaw, 7-alaw, 257-IBM Mu-Law, 258-IBM A-Law, 259-ADPCM, 65534-Extensible

ENCODING Indica el tipo de codificación que se usa para cada valor recibido little-endian o big-endian. [continúa siguiente página]

Usted debe escribir un programa en C que permita leer dicha información de un archivo de texto llamado **config_file.txt** y definir una estructura para almacenar la información leída. La estructura debe tener en cuenta el tipo de dato que representa cada parámetro leído y debe tener nombres de variables relacionadas con la información guardada. Tener en cuenta que el orden en el que aparecen los parámetros en el archivo de texto puede cambiar. Posteriormente dicha estructura la debe guardar en un archivo binario que se llame **config_file.bin**, pues para el software es mas fácil leer un archivo binario. Para verificar que lo haya hecho bien, lea e imprima en pantalla la información de nuevo. Se adjunta un archivo de muestra.