

Hoja de Trabajo no. 4

Instrucciones:

- Adjunto encontrará una carpeta comprimida llamada *dataset*, esta carpeta contiene mil imágenes de pokémones. Las imágenes cuentan con un formato PNG a color y dimensiones de 96x96.
- Deberá procesar todas las imágenes de la siguiente manera:
 - Cargar imagen a memoria (leerla).
 - Determinar el color dominante de la imagen. Idear un criterio y una función para indicar cuál es el color que domina en dicho pokémon. El resultado debe ser el código hexadecimal del color.
 - Determinar la cantidad de píxeles transparentes en la foto. Al tratarse de una imagen PNG, tenemos un cuarto canal que determina la opacidad. El tamaño del pokémon determina qué tantos píxeles de marco 96x96 abarca la ilustración. El resultado debe indicar el número exacto de píxeles que no se utilizan para representar al pokémon.
 - Calcular la cantidad de píxeles que sí se utilizan para representar al pokémon, utilizando como insumo el resultado del cálculo anterior.
 - Escribir en un archivo .csv el resultado del procesamiento, guardando los siguientes campos: número de pokémon, código hexadecimal del color dominante, cantidad de píxeles transparentes, cantidad píxeles a color.
- El procesamiento anteriormente descrito, realizarlo primero de forma *single-core* y *single-thread*. Medir el tiempo de ejecución.
- Luego, refactorizar el código para utilizar paralelismo (*multiprocessing*) y concurrencia (*multithreading*).
- Parametrizar el proceso paralelo/recurrente hasta tener una mejora significativa en rendimiento en comparación a la primera versión del código.

Entregable: Presentación en clase.

- 1) Hacer un demo de la versión paralela/recurrente del código.
- 2) Explicar su implementación de paralelismo y concurrencia.
- 3) Mostrar de forma gráfica los resultados de rendimiento.
- 4) Explicar las ganancias de rendimiento dadas por el paralelismo y las ganancias obtenidas por la concurrencia.

Se harán preguntas específicas del código como parte de la evaluación.