

1. Importar los módulos necesarios: PIL.ExifTags, PIL.Image y os.
2. Definir la función `decode_gps_info` que toma el argumento `exif`:
 - 2.1. Inicializar un diccionario vacío `gpsinfo`.
 - 2.2. Si la clave `GPSInfo` está presente en `exif`:
 - 2.2.1. Analizar las referencias geográficas.
 - 2.2.2. Calcular las coordenadas de latitud y longitud.
 - 2.2.3. Actualizar el diccionario `exif['GPSInfo']` con las coordenadas calculadas.
3. Definir la función `get_exif_metadata` que toma el argumento `image_path`:
 - 3.1. Inicializar un diccionario vacío `ret`.
 - 3.2. Abrir la imagen con `Image.open(image_path)`.
 - 3.3. Si la imagen tiene el atributo `_getexif`:
 - 3.3.1. Obtener la información EXIF de la imagen.
 - 3.3.2. Decodificar cada etiqueta y valor de la información EXIF.
 - 3.3.3. Almacenar las etiquetas y valores decodificados en el diccionario `ret`.
 - 3.3.4. Llamar a la función `decode_gps_info` con el diccionario `ret`.
 - 3.4. Devolver el diccionario `ret`.
4. Definir la función `save_metadata_to_txt` que toma los argumentos `metadata` y `file_path`:
 - 4.1. Abrir el archivo `file_path` en modo `append`.
 - 4.2. Escribir los metadatos en el archivo.
5. Definir la función `print_meta` que toma el argumento `image_folders`:
 - 5.1. Iterar sobre cada carpeta de imágenes en `image_folders`:
 - 5.1.1. Verificar si la ruta de la carpeta es un directorio válido.
 - 5.1.2. Cambiar el directorio actual a la carpeta de imágenes.
 - 5.1.3. Crear una carpeta de salida para los metadatos.
 - 5.1.4. Recorrer todos los archivos en la carpeta de imágenes:
 - 5.1.4.1. Intentar:
 - 5.1.4.1.1. Obtener los metadatos EXIF de la imagen.

5.1.4.1.2. Guardar los metadatos específicos en el archivo metadatos.txt.

5.1.4.1.3. Si hay información GPS, guardar las coordenadas en el archivo gpsmetadatos.txt.

5.1.4.2. Capturar cualquier excepción e imprimir el error.

5.1.5. Imprimir un mensaje indicando que los metadatos se guardaron Correctamente.

6. Fin.