

---

## Práctica 0:

# Tutorial Python y Procesamiento Básico de Imágenes

---

### 1. Objetivos

- Realizar la instalación y exploración inicial de los entornos de trabajo a utilizar en el curso.
- Repasar conceptos básicos sobre sintaxis, estructuras de datos y manipulación de paquetes en Python.
- Familiarizar al estudiante con las librerías de Python para el procesamiento de imágenes.

### 2. Parámetros de entrega

#### Entrega por Brightspace:

- Adjuntar un archivo *.ipynb* comprimido con el tutorial completo, con el siguiente nombre: *tutorial\_CódigoEstudiante*. **Nota:** Aceptaremos archivos comprimidos únicamente para esta práctica. Durante el resto del semestre, no se recibirán tales archivos/carpetas.
- Adjuntar un único archivo con el código de la segunda sección de la siguiente forma: *main\_CódigoEstudiante.py*. Para esta sección y en adelante, es **obligatoria la entrega de un archivo .py**. **NO** se recibirán archivos con otra extensión. Su entrega conllevará una penalización sobre la calificación final del proyecto.
  - Este código debe tener una longitud máxima de 500 líneas. Se generará una penalización sobre la calificación final de cada entrega por cada 10 líneas adicionales. Esto con el objetivo de promover el uso de estructuras iterativas y operaciones vectoriales, que aumenten la eficiencia de sus códigos (aspecto fundamental en tareas de programación).
- Adjuntar **TODO** el código que se entregó en el ítem anterior, en formato *.txt*. Llámelo de igual manera: *CódigoEstudiante.txt*. Esto con el fin de poder evaluar en Brightspace automáticamente cualquier intento de copia o similitud entre los algoritmos. **Cualquier intento de copia será tratado de acuerdo al reglamento de la Universidad. Si no incluye este ítem en su entrega tendrá una nota de 0 en el laboratorio.**

- Adjuntar un único archivo con el informe en PDF nombrado así: *CódigoEstudiante.pdf*. El informe debe presentarse con el formato CVPR (para más información del informe ver sección de 4).
- No se recibirá ningún archivo por algún medio diferente a Brightspace.
- Tendrán una semana para realizar esta práctica de forma individual. El plazo de entrega será hasta las **7:59 a.m. del domingo 31 de enero de 2021**. El vínculo para el envío de los documentos dejará de estar disponible luego de esta hora. **No se recibirán documentos o archivos después de la hora máxima de entrega** (tenga en cuenta que la hora queda registrada en el envío de Brightspace). Cada estudiante tiene varios intentos en Brightspace para cada entrega, pueden hacer entregas parciales y se tendrá en cuenta únicamente el último intento.
- Esta entrega hace parte del primer miniproyecto del curso. Su calificación será independiente de las siguientes entregas y tendrá un peso sobre la nota final del curso de 2 %.
- Esta entrega incluye el desarrollo del procedimiento descrito en el presente enunciado, junto con la firma del 'Contrato de Estudiantes'. Su aceptación y firma es obligatoria para comenzar el curso y debe ser parte de los documentos entregados en la fecha establecida. No entregar el documento implicará una calificación de cero (0) en esta primera práctica del semestre. Cualquier reclamación sobre el mismo debe realizarse antes de esta fecha de entrega.

### 3. Reglas Generales

- **La asistencia a la sección de laboratorio es obligatoria.** Acorde con el Reglamento General de Estudiantes de Pregrado de la Universidad de los Andes, la inasistencia a más del 20 % de las clases de laboratorio resultará en la reprobación de la materia completa (laboratorio y magistral).
- Aunque es válido discutir los problemas generales con sus compañeros, la solución y el código deben ser de su completa autoría. Está prohibido copiar literalmente el algoritmo y/o procedimiento desarrollado por otra persona (sea de Internet o por terceros).

### 4. Parámetros de calificación

#### Resultados:

1. Todos los códigos deben mantener orden y coherencia en la ejecución de comandos. Cada vez que se muestre una figura, el programa debe esperar para que se presione una tecla, para así continuar con la siguiente y así sucesivamente (para esto utilice en Python `input("Press Enter to continue...")`). Si quiere contrastar dos o más imágenes utilice un `subplot`.
2. Toda figura debe estar numerada y debe llevar su título y descripción en el informe.
3. El código debe estar debidamente comentado.
4. Asuma que dentro de la carpeta de ejecución del código se encuentran los archivos necesarios para el laboratorio.

5. No utilice rutas absolutas para leer o guardar archivos. Este es el error más común en la ejecución de los códigos.

Para generar rutas utilice `os.path.join` en Python, ya que puede que corramos los laboratorios usando Linux o Windows y los separadores de archivos cambian dependiendo del sistema operativo.

**Ejemplo:** Dentro del código principal, el estudiante quiere leer la imagen `im.png` que está dentro de una carpeta de imágenes en la misma ruta que el *main*.

- 5.1. Forma incorrecta:

```
skimage.io.imread('C:/Estudiante/docs/ElProyecto/ims/im.png')
```

- 5.2. Forma correcta:

```
skimage.io.imread(os.path.join('ims', 'im.png'))
```

### Informe:

Todos los laboratorios deben realizarse en formato *L<sup>A</sup>T<sub>E</sub>X* o Word, pueden obtener una plantilla del formato en el siguiente *vínculo*. Cabe resaltar que los informes no deben contener ninguna sección de artículo científico, esto significa que no deben incluir ninguna división como resultados, *abstract* o conclusiones. Por consiguiente, **deben responder únicamente a las preguntas del informe, de forma organizada**. Recuerden incluir imágenes de sus resultados y documentarlas debidamente. Por último, el informe tiene una longitud máxima de 3 páginas. Se pueden incluir imágenes y/o tablas en la sección de Anexos pero los resultados principales deben ser parte del informe.

### Bonos:

Cada persona ganará puntos que suben la nota por cumplir la siguiente característica:

1. Desarrollar el informe en *L<sup>A</sup>T<sub>E</sub>X*. Aquellas personas que lo desarrollen en *L<sup>A</sup>T<sub>E</sub>X*, deben escribir al final del informe **”Realizado en *L<sup>A</sup>T<sub>E</sub>X*”**. De lo contrario no se contará el bono. Los grupos que intenten reproducir la frase en un informe realizado en Word tendrán 0 en la nota de dicho proyecto. Para poder escribir el logo utilice el comando `”\LaTeX”` en su informe de Latex.

Estos puntos se asignarán de acuerdo al criterio de las profesoras.

## 5. Procedimiento

### 5.1. Tutorial de Python (50 %)

#### 5.1.1. Descargar archivos

Para realizar este tutorial, vamos a ejecutar Python dentro de archivos llamados Jupyter Notebooks (*.ipynb*). A diferencia de los scripts de Python (*.py*), los notebooks permiten ejecutar el código paso a paso, además de mostrar y guardar los resultados intermedios en el notebook.

1. Ingresen a Brightspace y descarguen el archivo IBIO3470\_202110\_Laboratorio0.zip.
2. Descompriman el archivo. En el archivo README encontrarán instrucciones y recomendaciones iniciales. En la carpeta Python-tutorial estará el archivo del tutorial organizado por temas.

#### 5.1.2. Iniciar Jupyter Notebook

1. En su computador inicie Anaconda Navigator. Seleccione y lance el Software *Jupyter*.
2. En la esquina superior izquierda, está el botón de *Upload* (flecha hacia arriba). Presiónelo y busque la carpeta del tutorial (*Python-tutorial*). Seleccione los archivos que están dentro.

#### 5.1.3. Tutorial de Python

El tutorial está organizado en un único notebook, con temas relevantes para que tengan las bases de programación con Python. El objetivo es correr todo el archivo, celda por celda, y realizar los ejercicios propuestos. Una vez termine de revisar (correr completamente) el notebook, asegúrese de guardar los cambios realizados. Al finalizar, debe comprimir el archivo revisado para su envío y asignar el nombre indicado. Los detalles de entrega se especificaron en la Sección 2.

El presente tutorial fue adaptado de aquel creado por *Python Bootcamp Uniandes*. Si desean profundizar en alguna de las temáticas vistas o adicionales pueden visitar el siguiente repositorio: <https://github.com/PythonBootcampUniandes/notas-de-clase>. Adicionalmente, los invitamos a que, por medio del *free trial* de 7 días, desarrollen el curso de Python3 de CodeAcademy (<https://www.codecademy.com/learn/learn-python-3>). Los tutoriales son interactivos y, con las herramientas que adquieren en el tutorial de esta práctica, se les facilitará desarrollarlo.

#### Notas:

- Los conceptos revisados en esta sección serán útiles para completar el resto de la práctica.
- No deben responder las preguntas propuestas por el tutorial en el informe. Estas son únicamente preguntas guías para una autoevaluación del estudiante a lo largo del estudiante.

## 5.2. Generalidades del Procesamiento de Imágenes (50 %)

### 5.2.1. Importación de librerías

Para la realización de esta práctica, necesitará las siguientes librerías de Python: `numpy`, `requests`, `skimage` y `matplotlib`. Por conveniencia, importe todo al comienzo del script. Importe los siguientes elementos, como se vio en el tutorial:

1. La librería `numpy` abreviada como `np`
2. La librería `requests`
3. La librería `skimage.io` abreviada como `io`
4. La función `rgb2gray` de la librería `skimage.color`
5. La librería `pyplot` de `matplotlib` abreviada como `plt`

### 5.2.2. Imágenes Digitales

1. Seleccione **una** de las imágenes disponibles en los siguientes vínculos:
  - *Elefantes*
  - *Jirafas*
  - *Cebras*
2. Descargue la imagen seleccionada de Internet utilizando el comando `requests.get()`. Su código debe realizar esta descarga de forma automática, de la siguiente forma:

```
r = requests.get(image_url)
with open(image_name, 'wb') as f:
    f.write(r.content)
```

En el informe, responda: ¿Cuál es la diferencia entre los formatos disponibles para guardar imágenes (JPEG, JPG, TIFF, PNG y DICOM)?

3. Cargue la imagen con el comando `io.imread()` y asígnela a una *variable*. Posteriormente, convierta la imagen a escala de grises y almacénela en una variable diferente. Responda: ¿Cómo funciona `rgb2gray`? ¿Cómo se refleja este cambio en las dimensiones y valores de la imagen luego de esta transformación?
4. Para visualizar las imágenes, utilice `plt.imshow()` acompañado de `plt.show()`. ¿Qué pasa si usa el comando `plt.plot()` para realizar la visualización? **No es necesario que reporte estas visualizaciones iniciales.**

5. Utilice el comando *variable*.shape para visualizar las dimensiones de la imagen. Luego, explore la naturaleza de la imagen en el computador, utilizando el nombre de su *variable* como comando. Con esto en mente, responda en el informe: ¿Cómo se representa una imagen a color en un computador? ¿De qué clase son estas estructuras? ¿Hay otras clases para representarlos? ¿Cuáles son las ventajas y desventajas de usar las clases *uint8*, *uint16*, *double* y *single*?
6. Visualice la imagen a color y cada uno de los canales en un subplot de  $2 \times 2$ , como se muestra en la Figura 1a. Utilice el comando `plt.subplot()`. Visualice los canales con la opción `cmap = 'gray'` de `plt.imshow()`. Guarde la imagen generada por el subplot mediante el comando `plt.savefig()`. ¿Qué representan los niveles de gris en las imágenes de los canales de color?
7. Obtenga el histograma de la imagen a color convirtiéndola en un vector y usando `plt.hist()`. Si tiene problemas con el número de dimensiones de la imagen, utilice la función `flatten()`.
8. Adicionalmente, obtenga el histograma de la imagen en escala de grises. Asegúrese de que la imagen sea un vector y que el rango del histograma corresponde al rango de valores de la imagen.
9. En un subplot de  $2 \times 2$ , visualice la imagen a color, la imagen en escala de grises y sus respectivos histogramas, como se muestra en la Figura 1b. Almacene esta nueva imagen y repórtela en el informe.
10. Responda: ¿Qué es el histograma de una imagen? ¿Qué diferencias observa entre el histograma de la imagen a color vectorizada y el de la imagen en escala de grises? ¿Tiene sentido el histograma de la imagen a color obtenido? (Recuerde que al vectorizar la imagen, se mezcla la información de los tres canales de color en un vector? ¿Es posible recuperar una imagen a partir de su(s) histograma(s)?

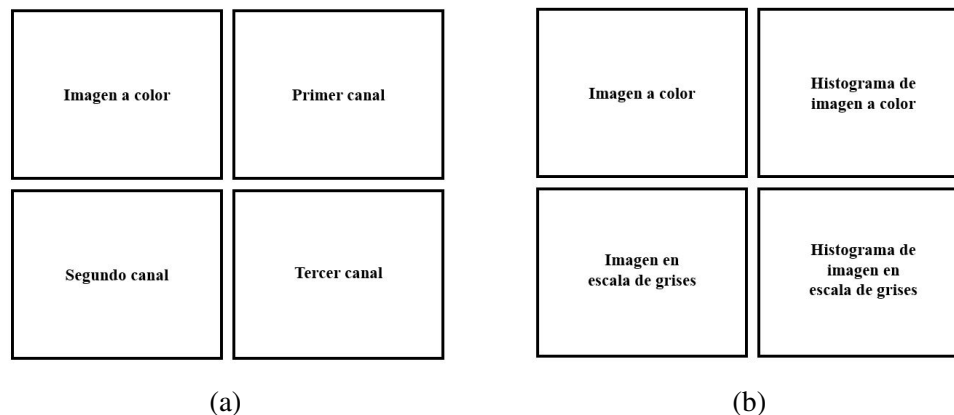


Figura 1: Estructuras de visualización. (a) Numeral 6, (b) Numeral 9.

**Nota:** No olvide incluir títulos y eliminar la numeración de los ejes donde corresponda (explore los métodos de `matplotlib.pyplot` para estos fines).