

## Taller 1

**Fecha de entrega:** lunes 17 de agosto de 2020, 8 pm

**1. Cree una clase llamada `colorImage`:**

- a. Constructor:
  - i. recibe un string con la ruta de la imagen
  - ii. carga la imagen (via OpenCV) y la almacena en self
- b. Métodos:
  - i. **`displayProperties`**: visualiza en pantalla (via print) el ancho y alto de la imagen
  - ii. **`makeGray`**: devuelve una versión en grises de la imagen
  - iii. **`colorizeRGB`**: recibe un string correspondiente a un canal de color: 'red', 'green' o 'blue', y regresa una imagen colorizada: rojiza, verdoza, o azulosa. Respectivamente. La colorización se realiza, calculando la imagen en grises de la imagen original, copiando dicha imagen en el canal de color escogido\* y llevando a cero las otras dos componentes. Por ejemplo, si el método es llamado con parámetro de entrada 'green', la imagen colorizada debe verse verdoza.
  - iv. **`makeHue`**: devuelve una imagen que resalta los tonos (Hue) de la imagen original. Para generar esta imagen, transforme la imagen original al espacio HSV, lleve las componentes S y V al valor constante de 255 y deje H intacta. Finalmente, transforme la imagen obtenida a RGB y regrésela

\* En OpenCV el orden de las componentes de color es BGR

**2. Cree un script (`main.py`) de Python que:**

- a. Utilice la clase `colorImage`
- b. Pida al usuario la ruta de la imagen con la que desea trabajar
- c. Visualice el ancho y alto de la imagen
- d. Visualice (via OpenCV) la imagen en grises
- e. Visualice (via OpenCV) una versión rojiza de la imagen
- f. Visualice (via OpenCV) una versión de tonos/colores resaltados

**3. El código debe estar debidamente indentado y comentado.**

**4. Cree un repositorio de GitHub para su código y envíe el enlace antes de la fecha y hora de entrega.**