**Marco teórico**

Una imagen de color RGB se representa por tres matrices bidimensionales, correspondientes a los planos R, G y B. La descomposición se asocia a canales cuyos valores se encuentran en general entre 0 y 1, indicando el 0 la falta total de ese canal y 1 la intensidad máxima de ese canal.   
Por ejemplo, en la representación RGB tener todos los canales en 1 representan el color blanco, y tener todos los canales en 0 representan el negro.

Para obtener los planos RGB se ejecutan los comandos:

>> Im\_R = f(:,:,1)

>> Im\_G = f(:,:,2)

>> Im\_B = f(:,:,3)

Que son los mapas de bits en de un plano de color y porque en este caso son 8 mapas de bits por cada plano de color.

Es común encontrar que los valores sean entre 0 y 255 ya que se almacenan en la computadora con 8 bits o un byte. Y en general 256 valores son suficientes ya que son un total de 16.7 millones de tonalidades de colores.

La forma más tradicional para trabajar con Imágenes es interpretando que son matrices de 2 + 1 dimensiones.

En Octave las operaciones de lectura y escritura de imágenes son imread y imwrite.

Para imread el nombre del archivo es suficiente para devolver una matriz de la cantidad de pixeles y canales que almacena el archivo. Para almacenar una matriz como una imagen hay que indicar en que formato se debe almacenar.

**imshow**  
Para ver la imagen se usa el comando imshow cuyo parámetro de entrada es la matriz a mostrar como una imagen.

**subplot**

Se puede decir que el comando subplot( ) tiene como prototipo subplot( filas, columnas, fig\_actual ), tanto ‘filas’, ‘columnas’ y ‘fig\_actual’ serán números enteros positivos, donde:

\*filas: Es un número entero que corresponde al número de filas en que se ha dividido la ventana de gráficos.

\* columnas: Es un número entero que corresponde al número de columnas en que se ha dividido la ventana de gráficos.

\* fig\_actual: Es un número entero positivo que hace referencia al plot o figura a la cual queremos hacer cambios, podemos decir que este valor hace las veces de subíndice dentro de esta matriz de gráficos.

La función BITAND (a, b) tiene como propósito anidar bits por medio de AND de bits de los números enteros a y b