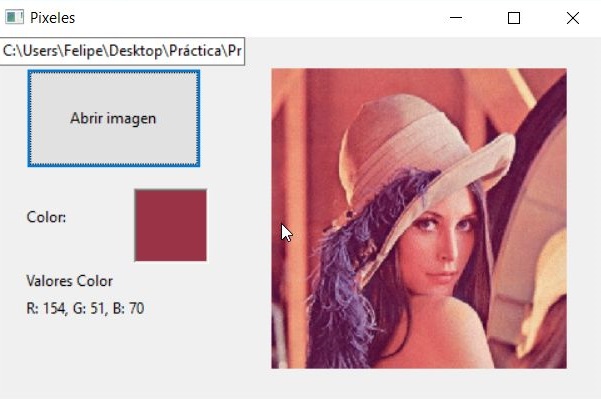
**Descripción:**

Una imagen digital es una representación bidimensional de una imagen a partir de una matriz numérica. De esta forma cada imagen digital está compuesta por pixeles. Estos pixeles almacenan la información en un registro numérico representativo, que el ser humano interpreta como color.

Una imagen normal RGB se compone de tres matrices, las cuales en su combinación numérica generan diversos colores que pueden ser percibidos por el sentido de la vista. De esta forma el color rojo estaría representado por la matriz para el rojo [255,0,0], verde [0,255,0] y azul [0,0,255].

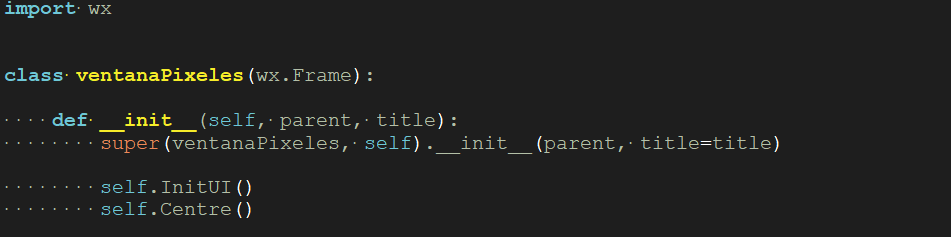
**Objetivo:**Explorar una imagen por pixeles, que al posicionar el cursor en la imagen nos muestre el color y los valores RGB del pixel señalado.

Descripción gráfica:



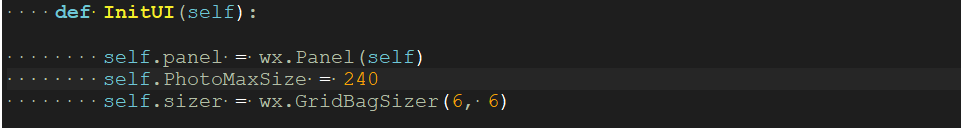
**Desarrollo**

1. Inicie Ninja o su IDE para programar en Python
2. Importe wx y cree la clase ventanaPixeles:



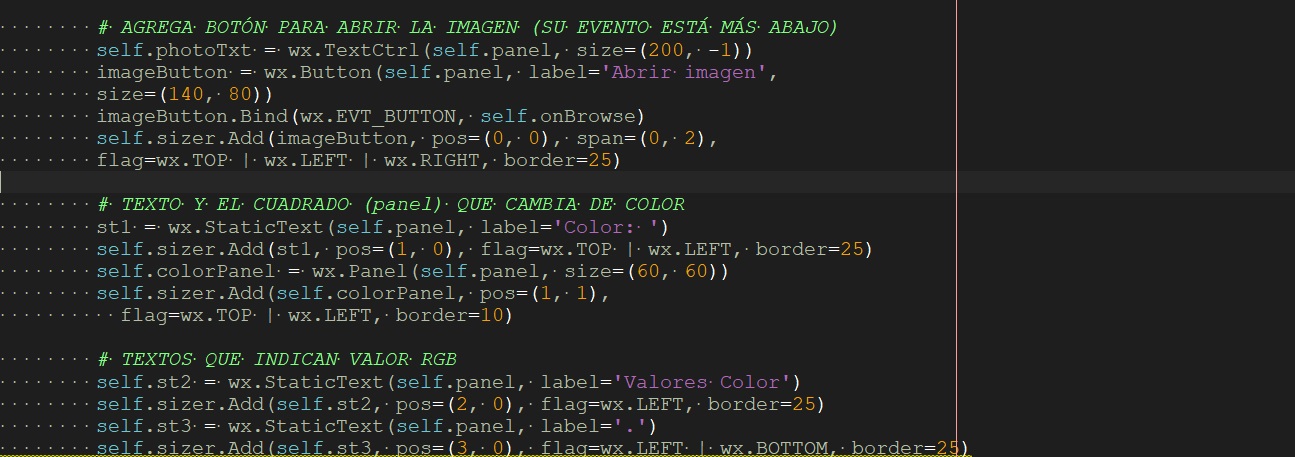
*self.InitUI() es una función que utilizaremos para agregar todos los demás objetos a la ventana, mientras que self.Centre() es simplemente para centrar la ventana en la pantalla.*

1. Cree el panel, el tamaño máximo de la imagen (después verá su utilidad) y por último un sizer de tipo GridBagSizer



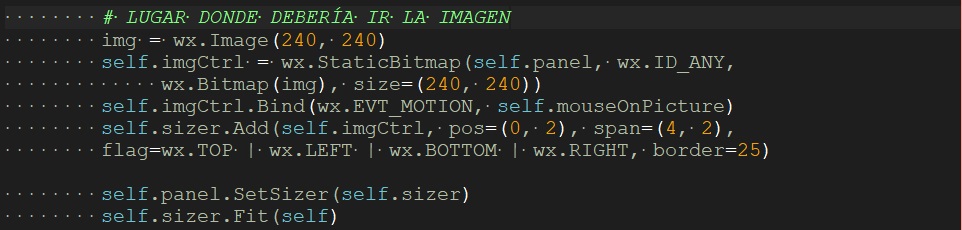
*Los sizers se utilizan para posicionar objetos en una ventana sin necesidad de coordenadas o posiciones absolutas, en este caso, GridBagSizer es versátil ya que permite poner objetos de distintos tamaños y envergadura. Link con más información:* [*Layout Management*](http://zetcode.com/wxpython/layout/)

1. Agregue botón, textos y cuadrado como se indican en la imagen de descripción gráfica



*Note que luego de agregar el botón se le asocia un evento que lleva a la función “onBrowse”, esta se utilizará para seleccionar y abrir la imagen; Luego de agregar cualquier objeto estos se añaden al sizer a través de la función Add donde se indica posición en filas y columnas. Para mayor información de clases de la librería wx visitar:* [*API wxPython*](https://wxpython.org/)

1. Agregue un marco en el cual irá la imagen

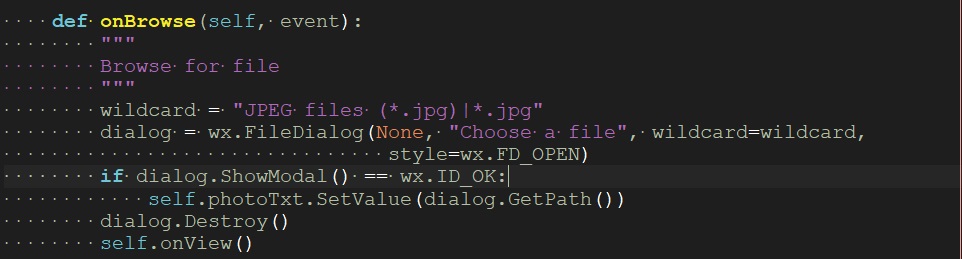


*En wx.Image usté está creando una “imagen vacía”, es decir, inicialmente aparecerá un cuadro en negro que será reemplazado cuando el usuario ingresé una imagen de su computador.*

*Use la clase StaticBitmap y luego asóciele el evento (self.imgCrtl.Bind) con el que toma la posición del mouse y llama a la función que le servirá para señalar los valores rgb del pixel seleccionado.*

*self.sizer.Fit(self) es una función que sirve para que la ventana tenga el tamaño necesario para contener a todos los objetos dentro del sizer.*

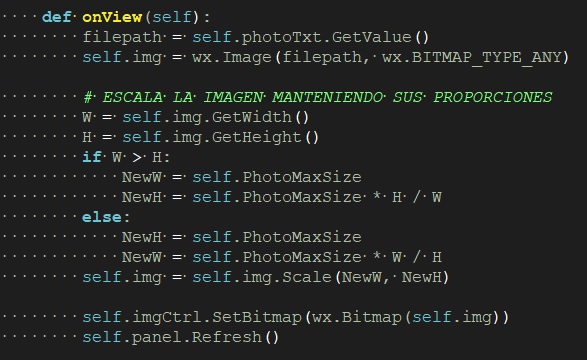
1. Cree la función onBrowse() activada con el evento del botón “Abrir imagen”



*Se usa la clase FileDialog para encontrar el archivo de imagen dentro del computador, en la API señalada anteriormente puede encontrar el funcionamiento de esta clase.*

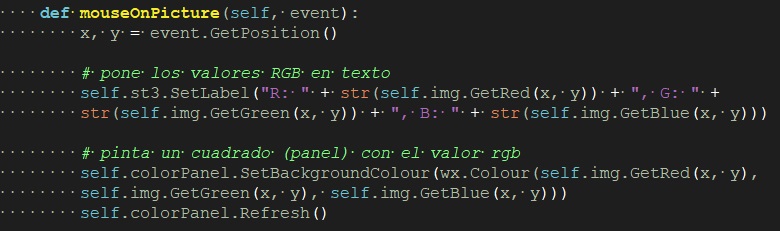
*Al final de la función se llama a self.onView() que se utiliza para mostrar la imagen seleccionada*

1. Cree la función para visualizar la imagen seleccionada



*Use el filepath obtenido anteriormente para señalar la imagen, pero esta no se muestra tal cual, sino que se escala al tamaño que la quiere, esta escalada deja la imagen con tamaños proporcionales a los originales.*

1. Cree la función asociada con el evento de mover el mouse dentro de la imagen



*Use coordenadas para indicar la posición del mouse, luego use las funciones GetGreen(), GetRed() y GetBlue() de la clase wx.Image para obtener los valores rgb, finalmente use estos valores para mostrarlos en un texto y cambiar el color del panel cuadrado.*

1. Defina la función main donde muestra la clase ventanaPixeles creada

