

Contenido

Pdf a imagen .jpg con Python	2
Descripción.....	2
Preparando el entorno de desarrollo:	2
Python	2
Instalando Poppler.	2
Instalando ImageMagick	4
Instalando Pdf2image	5
Instalando Pillow	6
Instalando Pyinstaller.....	6
Implementación	6
Cómo ejecutar el archivo test_PDFtoImage.exe	9

Pdf a imagen .jpg con Python

Descripción

El programa tiene la función de convertir un archivo .pdf a una imagen .jpg, recibe dos argumentos, el primer argumento es la ruta del directorio y archivo pdf que va a ser convertido a imagen, el segundo argumento debe ser la ruta del directorio donde la imagen procesada va a guardarse, el programa entonces retorna una imagen .jpg, que es el archivo pdf ya convertido en imagen.

Preparando el entorno de desarrollo:

Para la implementación de esta tarea vamos a utilizar algunas utilidades y bibliotecas:

1. Python 3.8.0.
2. Poppler.
3. ImageMagik.
4. Pdf2image.
5. Pillow.
6. Pyinstaller

Python

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional; para lograr la conversión de archivos pdf a imagen .jpg utilizaremos este lenguaje de programación en el proceso de desarrollo.

Instalando Poppler.

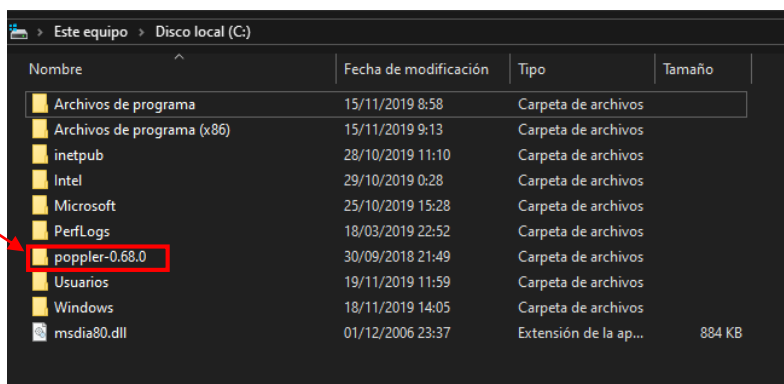
Poppler es una biblioteca de manipulación de PDF que se basa en la base de código xpdf-3.0. Para instalarlo se deben descargar los binarios de Windows consultando <http://blog.alivate.com.au/poppler-windows/>

Descargar

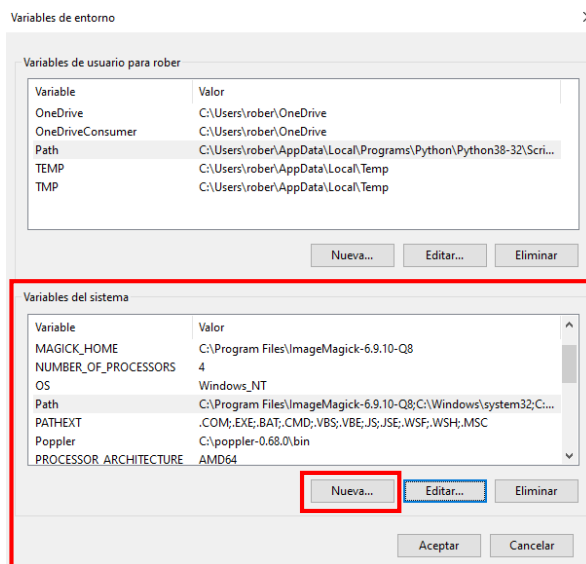
Último binario: [poppler-0.68.0_x86](#)

Teniendo ya descargados los archivos necesarios ubicamos la carpeta poppler-0.68.0 en la unidad C:

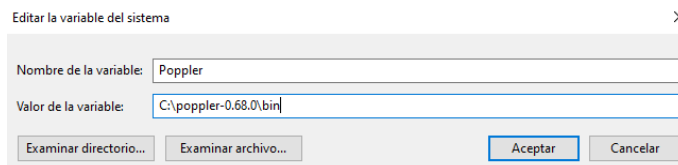
Carpeta con los
archivos para usar
poppler en la
unidad C:



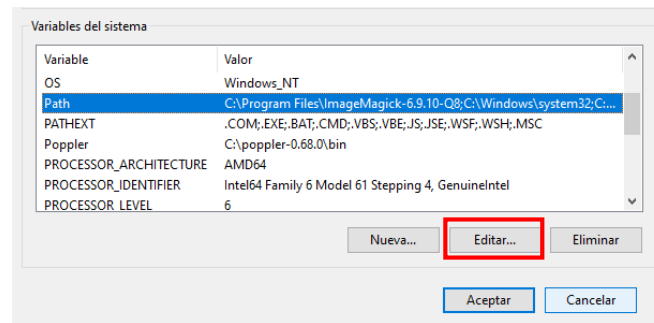
A continuación, agregaremos poppler como una variable del sistema, para conseguirlo iremos al panel de control, click en Configuración avanzada, click en Variables de entorno, en variables del sistema click en nuevo.



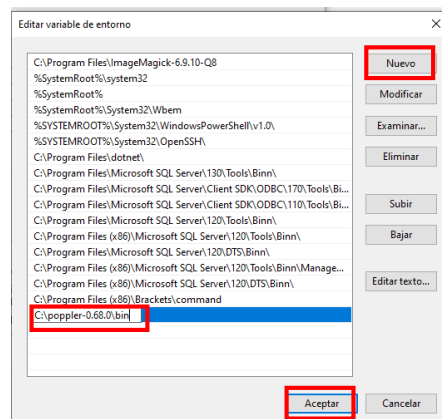
Agregamos la nueva variable y click en aceptar.



Buscamos la variable Path y click en editar.

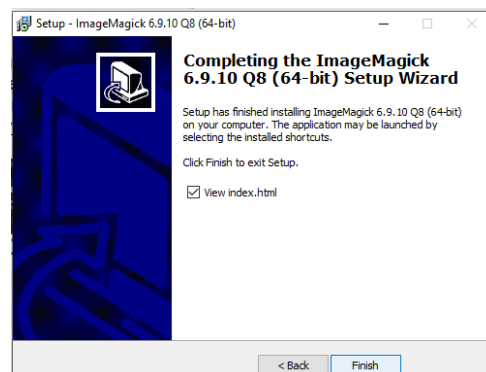


Click en nuevo, agregar C:\poppler-0.68.0\bin y click en aceptar hasta cerrar todas las ventanas.



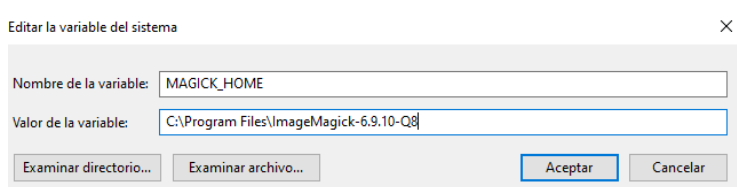
Instalando ImageMagick

ImageMagick es un conjunto de utilidades de código abierto para mostrar, manipular y convertir imágenes, capaz de leer y escribir más de 200 formatos. Para poder usarlo descargar el instalador en <https://imagemagick.org/script/download.php>. Luego de descargar el instalador lo ejecutamos y seguimos el asistente de instalación hasta finalizar la instalación.

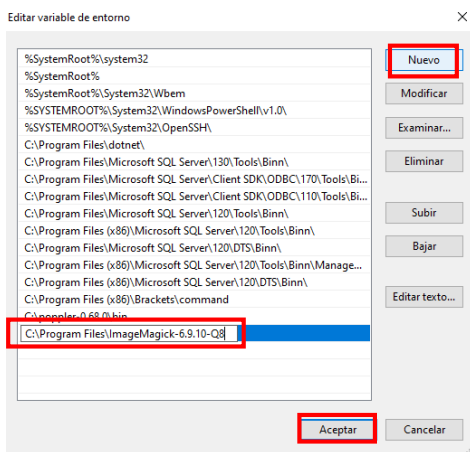


El siguiente paso es agregar ImageMagick como variable del sistema, para hacerlo seguimos los mismos pasos que se hicieron para agregar poppler como variable del sistema.

Iremos al panel de control, click en Configuración avanzada, click en Variables de entorno, en variables del sistema click en nuevo. Agregamos como nombre de variable MAGICK_HOME y en valor de la variable C:\Program Files\ImageMagick-6.9.10-Q8, y click en aceptar.



Ahora buscamos la variable Path, click en editar, click en nuevo, agregamos C:\Program Files\ImageMagick-6.9.10-Q8 y click en aceptar hasta cerrar todas las ventanas.



Instalando Pdf2image

Pdf2image es una biblioteca de Python que llama a la biblioteca pdftoppm para convertir un pdf en una secuencia de objetos de imagen PIL. La biblioteca pdftoppm utiliza poppler para ejecutar la conversión. Para instalar la biblioteca Pdf2image puede hacerlo a través de la consola de comandos usando:

```
Pip install pdf2image
```

Instalando Pillow

Pillow es una biblioteca de imágenes de Python que devuelve una lista de objetos de imagen para un PDF determinado según el formato elegido. Estos objetos de imagen se pueden convertir a formatos de archivo png o jpg utilizando la biblioteca Pillow. Para instalar la biblioteca pillow en python, puede hacerlo desde la consola de comandos ejecutando el comando:

```
Pip install Pillow
```

Instalando Pyinstaller.

PyInstaller agrupa una aplicación Python y todas sus dependencias en un solo paquete para poder ejecutar la aplicación empaquetada sin instalar un intérprete de Python; usaremos Pyinstaller para generar un único archivo ejecutable. Para instalar la versión de desarrollo actual desde la consola de comandos use:

```
pip install https://github.com/pyinstaller/pyinstaller/tarball/develop
```

Implementación

Para comenzar a crear el programa es necesario el intérprete de Python, y cualquier editor de texto. En el editor de texto de nuestra preferencia crearemos un archivo de nombre test_PDFtoImage.py donde escribiremos el código para que el programa pueda funcionar. Cuando tengamos el archivo creado, ya podemos comenzar a escribir código.

Importamos las bibliotecas que instalamos con anterioridad, adicionalmente la biblioteca sys nos servirá para enviar argumentos de línea de comando al script de Python.

```
import pdf2image
from PIL import Image
import time
import sys
```

Realizamos todas las declaraciones necesarias, y hacemos uso de la biblioteca sys.argv, donde sys.argv[1] es el primer argumento que el script va a recibir, es decir, el documento pdf, y sys.argv[2] el segundo argumento, es decir, la ruta del directorio donde guardaremos la imagen que retorna el programa.

```
#DECLARE CONSTANTS
PDF_PATH = sys.argv[1]
DPI = 200
OUTPUT_FOLDER = sys.argv[2]
FIRST_PAGE = None
LAST_PAGE = None
FORMAT = 'jpg'
THREAD_COUNT = 1
USERPWD = None
USE_CROBOX = False
STRICT = False
```

Crearemos dos funciones encargadas del proceso de conversión y guardado y las aplicamos.

```
def pdftopil(PDF_PATH,OUTPUT_FOLDER):
    start_time = time.time()
    pil_images = pdf2image.convert_from_path(
        PDF_PATH, dpi=DPI, output_folder=OUTPUT_FOLDER, first_page=FIRST_PAGE
        last_page=LAST_PAGE, fmt=FORMAT, thread_count=THREAD_COUNT,
        userpw=USERPWD, use_crobox=USE_CROBOX, strict=STRICT)

    print ("Time taken : " + str(time.time() - start_time))
    return pil_images

def save_images(pil_images):
    index = 1
    for image in pil_images:
        image.save("page_" + str(index) + ".jpg")
        index += 1

if __name__ == "__main__":
    pil_images = pdftopil(PDF_PATH,OUTPUT_FOLDER)
    save_images(pil_images)
```

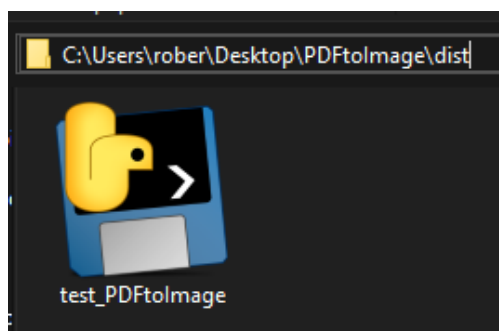
Por ultimo empaquetaremos test_PDFtoImage.py en un archivo ejecutable (.exe) usando la biblioteca Pyinstaller previamente instalada. Para lograrlo, realizaremos un proceso sencillo, como es un programa para terminal, habrá que abrir la terminal en ese directorio

```
C:\Users\rober\Desktop\PDFtoImage
```

Cuando ya estemos ubicados en el directorio donde se encuentra test_PDFtoImage.py en la terminal, ejecutaremos el comando Pyinstaller --onefile seguido del nombre del archivo que vamos a convertir a .exe:

```
Pyinstaller --onefile test_PDFtoImage.py
```

Después de ejecutar este comando se crearán una serie de directorios, en el directorio dist se encontrará el archivo test_PDFtoImage.exe que ahora está listo para ser ejecutado.



Cómo ejecutar el archivo test_PDFTolImage.exe

Antes de poder ejecutar este archivo es necesario instalar algunas utilidades y bibliotecas.

1. Poppler.
2. ImageMagick.

Para Instalar estos componentes consulte la sección **Preparando el entorno de desarrollo** en este documento.

El archivo ejecutable test_PDFTolImage.exe retorna una imagen a partir de un pdf, tiene la función de convertir un archivo .pdf a una imagen .jpg, recibe dos argumentos, el primer argumento es la ruta donde se encuentra el archivo pdf que va a ser convertido a imagen y el segundo argumento debe ser la ruta donde la imagen procesada va a guardarse, por eso es necesario tener listos los directorios donde se encuentra el documento pdf que vamos a convertir y el directorio donde queremos guardar la imagen resultante de la conversión.

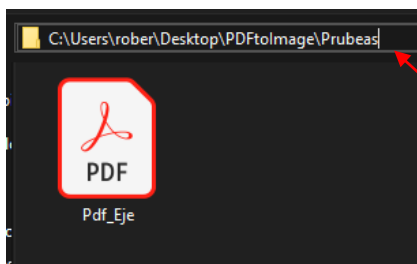
En este caso guardaremos la imagen en el mismo directorio donde se encuentra el documento pdf que usaremos de ejemplo. Cuando tengamos listo el documento y los directorios procederemos a identificar las rutas que usaremos como argumentos.

Argumento 1:

```
C:\Users\rober\Desktop\PDFtoImage\Prubeas\Pdf_Eje.pdf
```

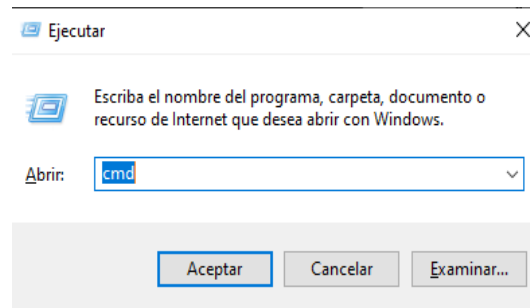
Argumento 2:

```
C:\Users\rober\Desktop\PDFtoImage\Prubeas
```

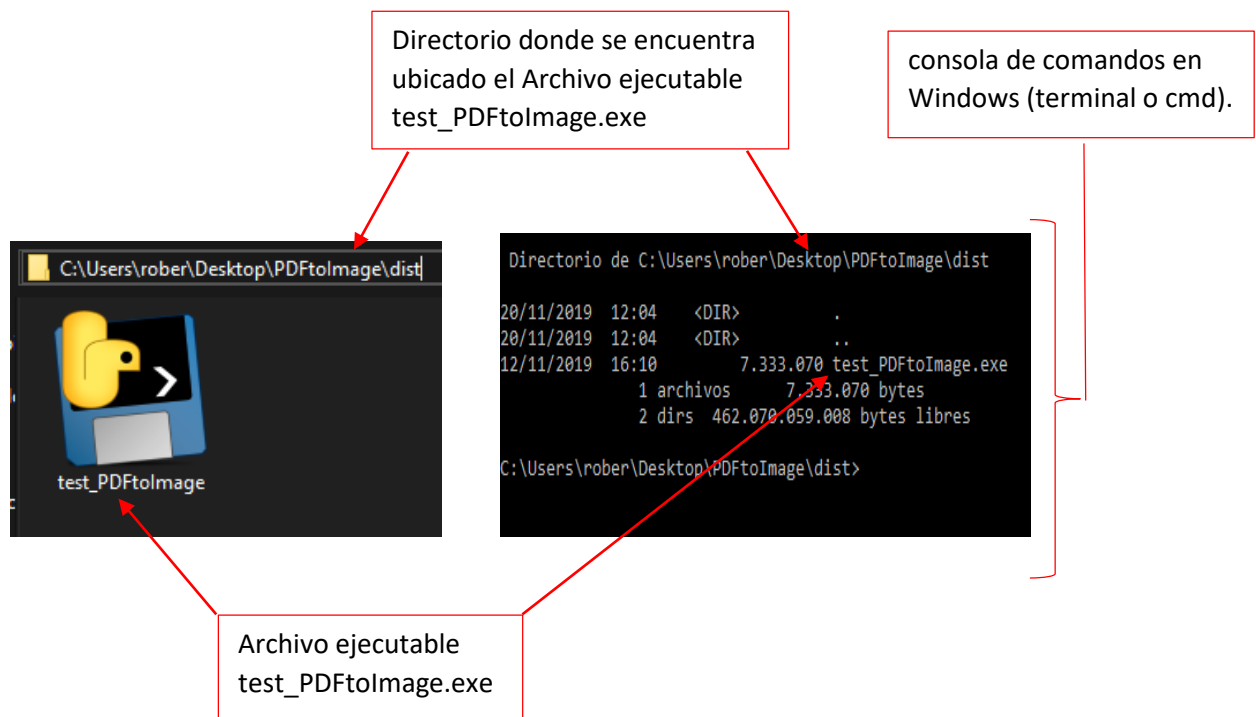


Ruta del directorio que usaremos como argumentos para realizar la conversión a imagen del documento pdf.

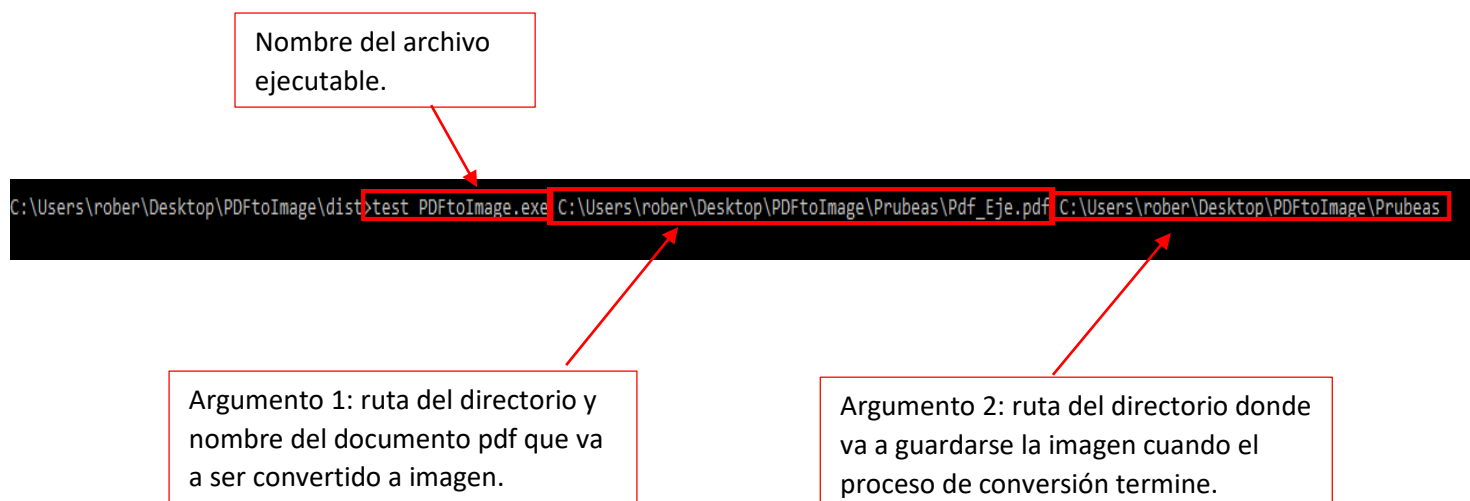
Para ejecutar el archivo test_PDfToImage.exe correctamente, como es un programa para terminal, es necesario hacerlo desde la consola de comandos en Windows (terminal o cmd). Para acceder y abrir la consola lo más cómodo es pulsar la combinación de teclas Windows + R, y en la pantalla que aparece escribir cmd, luego presionar Enter.



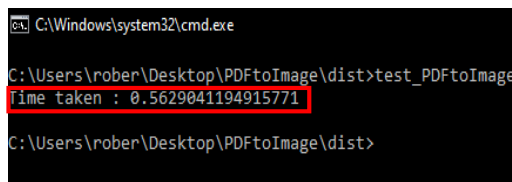
Una vez abierta la consola de comandos nos dirigiremos al directorio donde se encuentre ubicado el archivo ejecutable test_PDfToImage.exe



Cuando estemos en el directorio donde se encuentre el archivo ejecutable test_PDFtoImage.exe procederemos a ejecutarlo, para lograrlo escribiremos en la consola el nombre del archivo test_PDFtoImage.exe seguido de los argumentos necesarios, primero la ruta y nombre del documento pdf que va a ser convertido a imagen seguido de la ruta donde va a guardarse la imagen cuando termine el proceso de conversión, y pulsamos enter.



Podremos confirmar desde la consola de Windows que se realizó el proceso de conversión de forma correcta viendo el tiempo de ejecución del archivo test_PDFtoImage.exe.



A continuación, podremos ver que la imagen resultante de la conversión ya está en el directorio indicado.

