

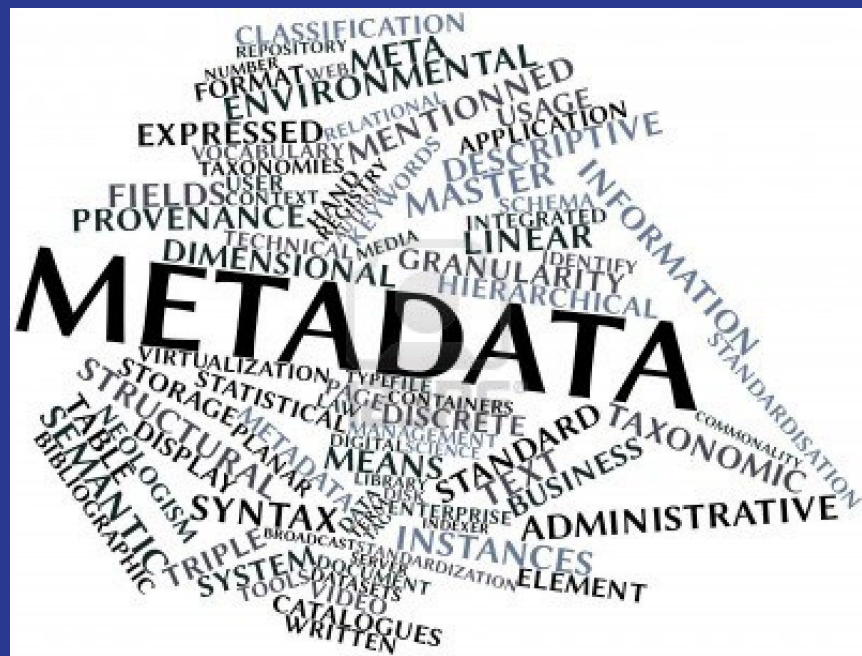
# Proyecto en Python: Metadata.

Javier Ponferrada López, Miguel Ángel Gavilán Merino,  
Guillermo Boquizo Sánchez

Lenguaje de Marcas y Sistemas de Gestión de Información  
1º de Desarrollo de Aplicaciones Web  
I.E.S Gran Capitán, Junio 2017

# ÍNDICE

1. Presentación de motivos
2. ¿Qué son los metadatos y cuál es su utilidad?
3. Ventajas de la gestión de metadatos
4. Escáner de metadata
5. Ejemplos de ejecución
6. Conclusiones
7. Bibliografía



## 1. Presentación de motivos:

La presente documentación se encarga de dar cuenta del propósito del proyecto, de los objetivos que se han perseguido, así como trata de explicar la funcionalidad de la aplicación resultante tras la elaboración de la misma en Python.

Este proyecto surge como resultado del trabajo colaborativo de las personas citadas, con fines pedagógicos.



## 2. ¿Qué son los metadatos y cuál es su utilidad?

Los metadatos refieren a datos que describen otros datos o "**datos sobre datos**". De forma general, el concepto de metadatos se refiere a aquellos datos que describen el contenido de los archivos o la información de los mismos.



Los metadatos se caracterizan por ser datos altamente estructurados que describen características de los datos, como el contenido, calidad, información y otras circunstancias o atributos.

Las diferencias entre tipos de metadatos dependerán, en esencia, de las reglas incluidas en las aplicaciones para determinar la estructura interna de los esquemas de datos.



En el actual contexto de Big Data, Internet de las cosas y cloud computing los metadatos han adquirido una relevancia inédita.

Frente a ingentes cantidades de información que crecen de forma exponencial, la gestión de los metadatos puede ser de gran ayuda para ganar en eficiencia operativa y/o tomar mejores decisiones para la obtención de ventajas competitivas.



### 3. **Ventaja de la gestión de metadatos**

Esta "información sobre información" engloba todos los atributos de los datos que describe cada archivo y, de forma conjunta, proporciona una información de interés con muy distintas utilidades a la hora de mejorar la gestión de los datos.



Algunas de las funciones más ventajosas de una buena administración de los metadatos a la hora de mejorar los procesos de gestión y gobierno de datos:

- *Facilitan búsquedas y análisis.*
- *Facilitan la estandarización.*
- *Ayudan a la integración.*
- *Gestión del cambio.*
- *Más seguridad.*
- *Mejoran los informes.*
- *Desarrollos más ágiles.*
- *Mejor gobernanza de datos.*



# 4. Escáner de metadata

## 4.1 Describiendo el proyecto

El presente proyecto trata de construir, en torno al comando linux exiftool, un escáner de metadatos para cualquier tipo de archivo.

No obstante, el proyecto tiene cuatro prerequisites, de momento ineludibles por funcionalidad:

1. Requiere la instalación de exiftool con `sudo apt-get install exiftool`.
2. Requiere la instalación del módulo tk con `sudo apt-get install python-tk`.
3. La ruta desde la que se escanee no puede contener espacios, pues los interpreta como partes del nombre del archivo. Esto es, espacios ni en carpetas ni en archivos.
4. Es operativo en sistemas Linux, no se ha logrado portabilidad aún a sistemas basados en NT como Windows, al no interpretar la línea de comandos dicho comando linux, y no lograr operatividad de algún port de dicho comando.

## 4. Escáner de metadata

```
import os
from Tkinter import *
from tkFileDialog import askopenfilename

def scanMetadataFile(ruta):
    try:
        if ruta != "":
            print "\n-----METADATA-----\n"
            os.system("exiftool " + ruta)
        else:
            print "Ruta incorrecta."
    except:
        print "Fichero no valido."
```

### 4.2 Funciones.py

Éste es un script muy básico, donde sencillamente se realizan las importaciones necesarias para el funcionamiento del programa, y se define la función `scanMetadataFile()` que será la encargada de realizar, previo control, la búsqueda.

## 4. Escáner de metadata

```
main.py x
1  |# -*- coding:utf-8 -*-
2  |from funciones import *
3
4  |def limpiarPantalla():
5  |    if os.name == "nt":
6  |        os.system("cls")
7  |    elif os.name == "posix":
8  |        os.system("clear")
9
10 |limpiarPantalla()
11
12 |print '\n>>>Pulse INTRO para abrir un documento...'
13 |raw_input()
14 |entrada = 's'
```

### 4.3 Main.py

En el script del main, encontramos diferenciadas dos secciones. La que se muestra a la izquierda permite visualizar la definición de la función `limpiarPantalla()`, así como su invocación, la importación desde `funciones.py` y el inicio de la segunda sección.

## 4. Escáner de metadata

```
while entrada == "s":  
    root = Tk()  
    root.withdraw()  
    ruta = askopenfilename()  
    scanMetadataFile(ruta)  
    entrada = raw_input("\n¿Desea abrir otro documento? s/n: ")  
    while entrada != 's' and entrada != 'n':  
        entrada = raw_input("\n¿Desea abrir otro documento? s/n: ")  
    limpiarPantalla()
```

En esta segunda sección encontramos la invocación de los métodos de librerías Tk que permitirán al usuario indicar, mediante el gestor de archivos, la ruta del fichero en la que se desea analizar los metadatos.

El programa solicitará la respuesta del usuario para seguir analizando otros ficheros o directamente, cerrar el programa.

## 5. Ejemplos de ejecución

A continuación, pasaremos a mostrar algunos ejemplos de ejecución del script en algunos de los tipos de archivo más conocidos.

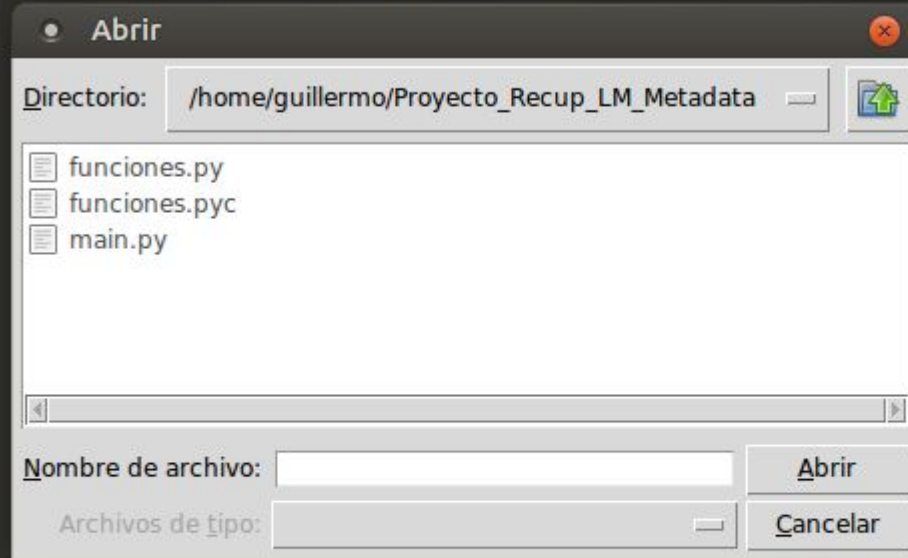
Como se indicó previamente, la única restricción es que la ruta no contenga ningún espacio como carácter

## 5. Ejemplos de ejecución

```
i:~/Proyecto_Recup_LM_Metadata$ python main.py
```

## 5. Ejemplos de ejecución

>>>Pulse INTRO para abrir un documento...



## 5. Ejemplos de ejecución

```
guillermo@ubuntu: ~/Proyecto_Recup_LM_Metadata
Archivo Editar Ver Buscar Terminal Ayuda

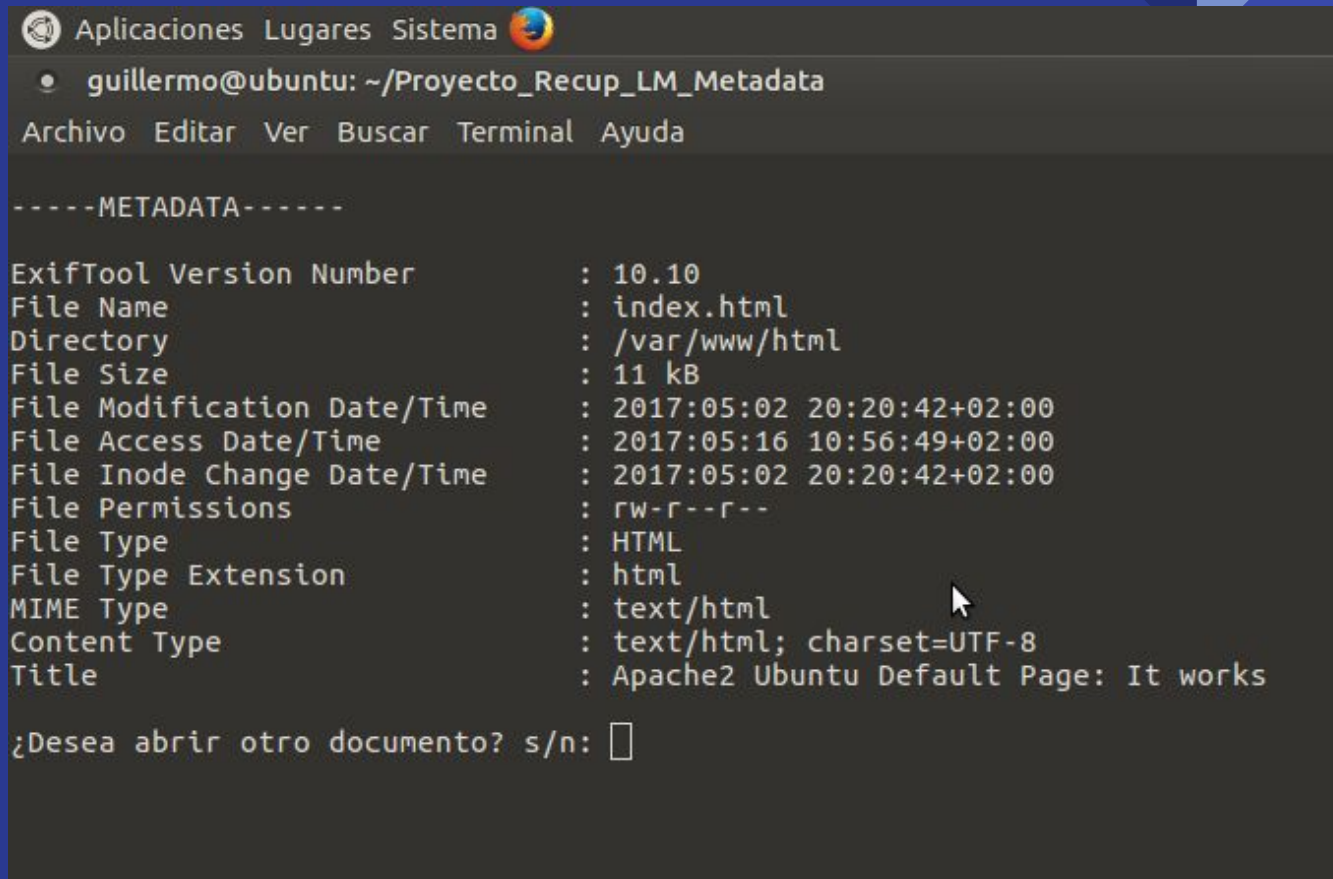
-----METADATA-----

ExifTool Version Number      : 10.10
File Name                    : guillermoboquijosanchez.tar
Directory                   : /home/guillermo/escritorio
File Size                   : 235 bytes
File Modification Date/Time  : 2017:05:16 12:10:41+02:00
File Access Date/Time       : 2017:06:05 10:29:04+02:00
File Inode Change Date/Time  : 2017:05:16 12:10:41+02:00
File Permissions             : rw-r--r--
File Type                   : GZIP
File Type Extension         : gz
MIME Type                   : application/x-gzip
Compression                 : Deflated
Flags                       : (none)
Modify Date                 : 2017:05:16 12:10:41+02:00
Extra Flags                 : (none)
Operating System            : Unix

¿Desea abrir otro documento? s/n: s
```



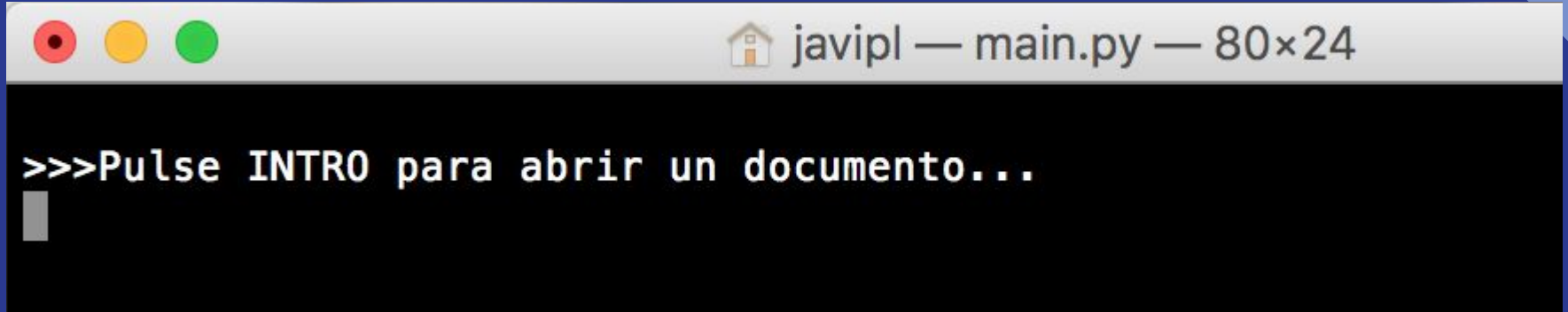
# 5. Ejemplos de ejecución



```
-----METADATA-----
ExifTool Version Number      : 10.10
File Name                    : index.html
Directory                    : /var/www/html
File Size                    : 11 kB
File Modification Date/Time   : 2017:05:02 20:20:42+02:00
File Access Date/Time        : 2017:05:16 10:56:49+02:00
File Inode Change Date/Time   : 2017:05:02 20:20:42+02:00
File Permissions              : rw-r--r--
File Type                    : HTML
File Type Extension          : html
MIME Type                    : text/html
Content Type                  : text/html; charset=UTF-8
Title                        : Apache2 Ubuntu Default Page: It works

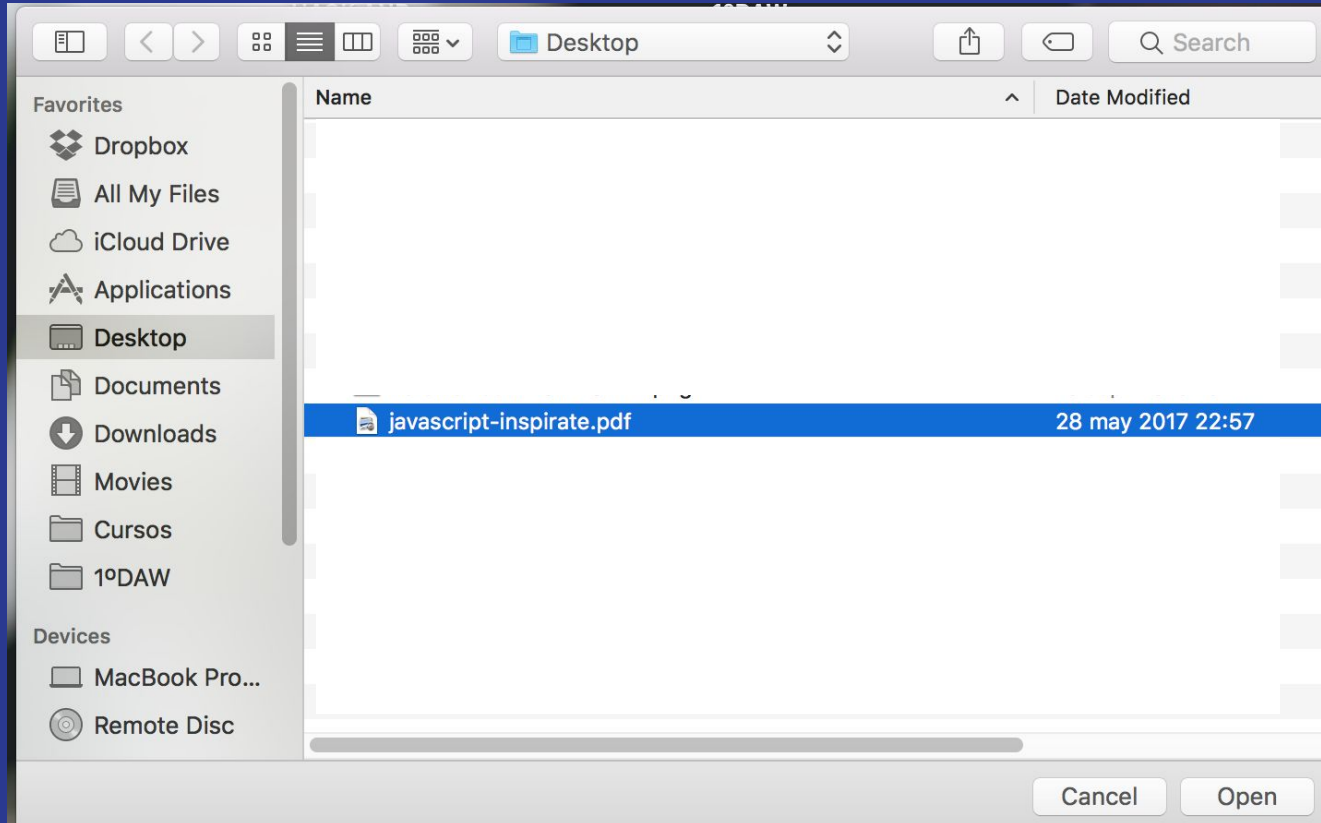
¿Desea abrir otro documento? s/n: ☐
```

## 5. Ejemplos de ejecución

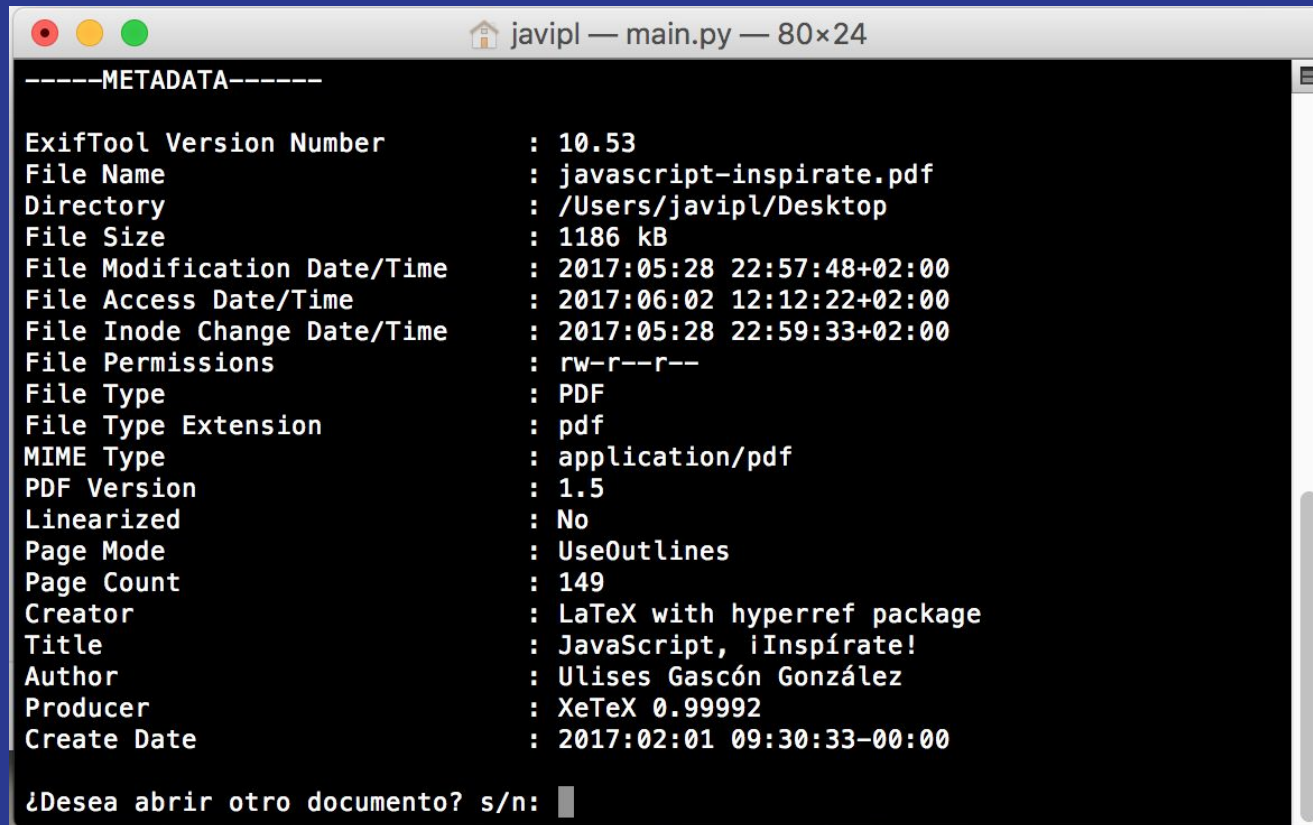


```
>>>Pulse INTRO para abrir un documento...
█
```

## 5. Ejemplos de ejecución



## 5. Ejemplos de ejecución

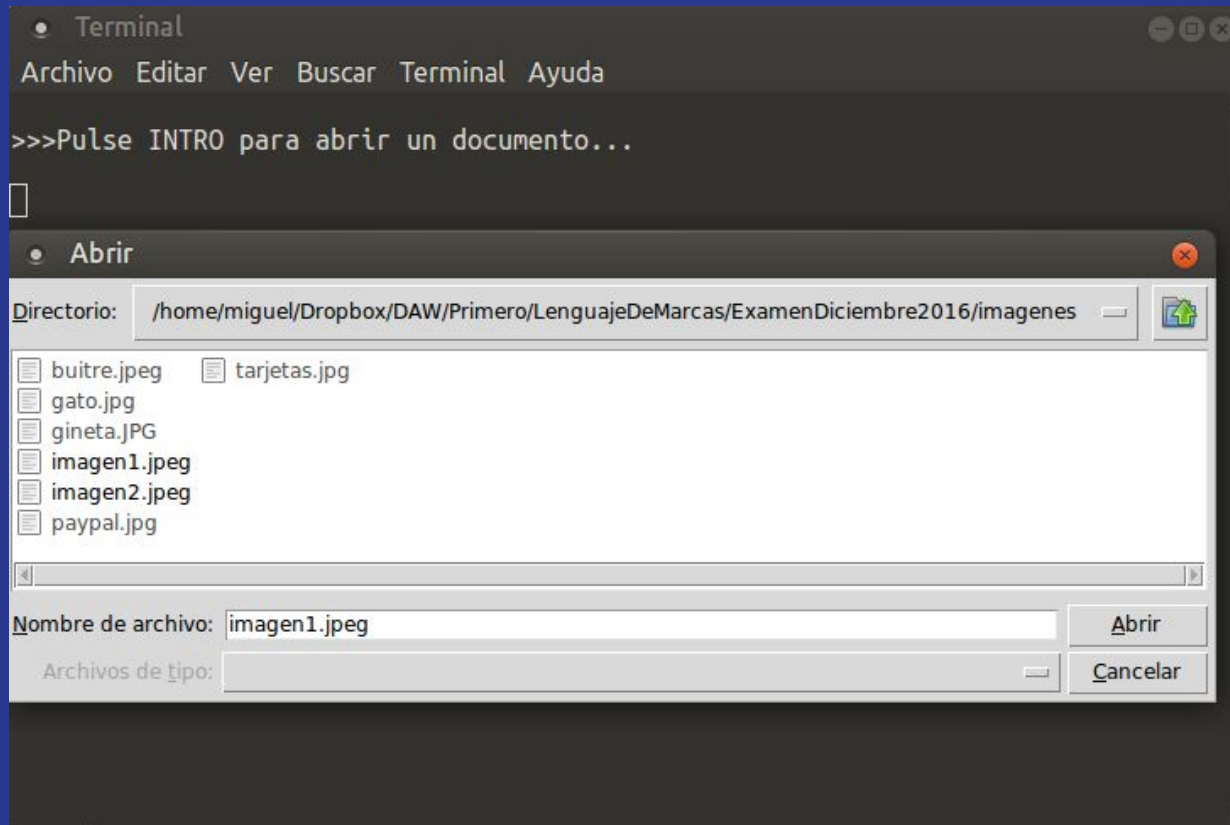


A screenshot of a terminal window titled "javipl — main.py — 80x24". The terminal displays the output of the ExifTool command, showing metadata for a PDF file named "javascript-inspirate.pdf". The output is organized into a table-like structure with labels and values. At the bottom, there is a prompt asking if the user wants to open another document.

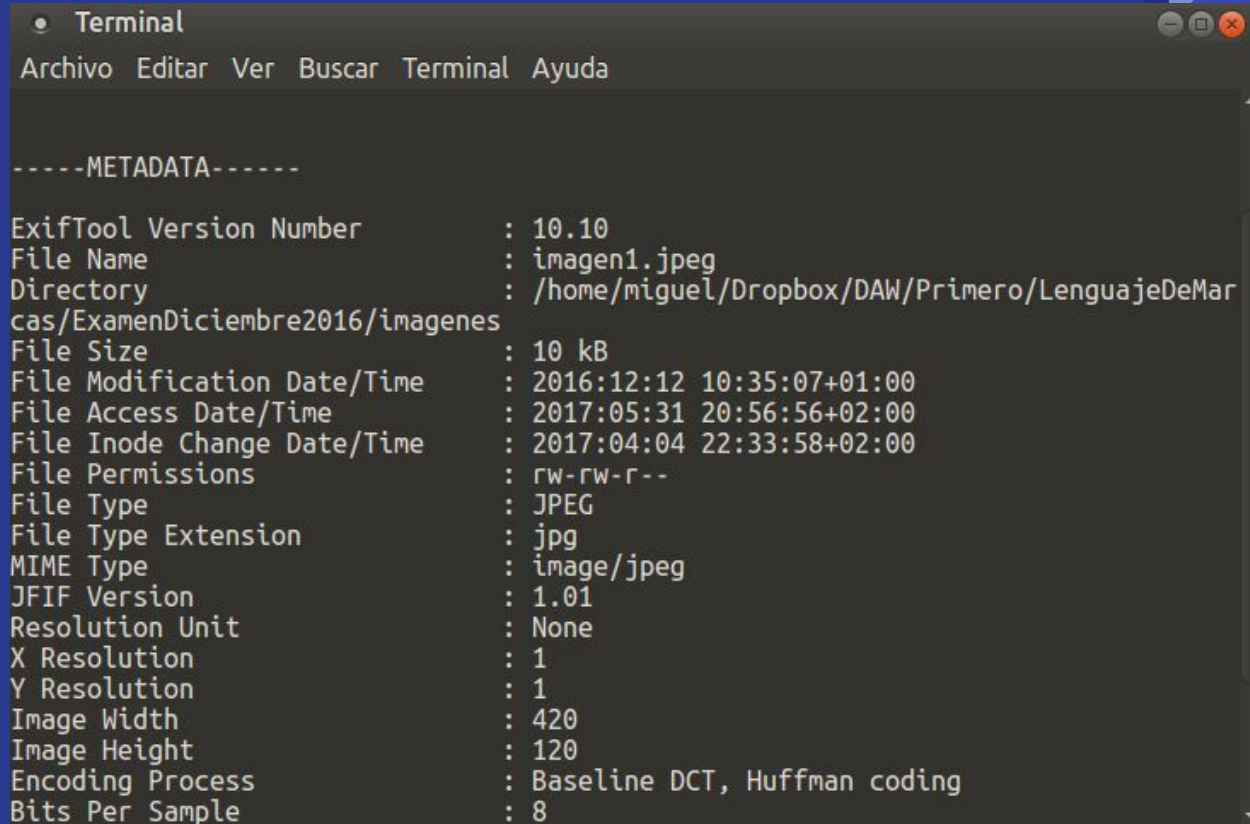
```
-----METADATA-----
ExifTool Version Number      : 10.53
File Name                    : javascript-inspirate.pdf
Directory                   : /Users/javipl/Desktop
File Size                    : 1186 kB
File Modification Date/Time  : 2017:05:28 22:57:48+02:00
File Access Date/Time       : 2017:06:02 12:12:22+02:00
File Inode Change Date/Time  : 2017:05:28 22:59:33+02:00
File Permissions             : rw-r--r--
File Type                    : PDF
File Type Extension          : pdf
MIME Type                    : application/pdf
PDF Version                  : 1.5
Linearized                   : No
Page Mode                    : UseOutlines
Page Count                   : 149
Creator                      : LaTeX with hyperref package
Title                       : JavaScript, ¡Inspírate!
Author                       : Ulises Gascón González
Producer                     : XeTeX 0.99992
Create Date                  : 2017:02:01 09:30:33-00:00

¿Desea abrir otro documento? s/n: █
```

## 5. Ejemplos de ejecución



# 5. Ejemplos de ejecución



```
Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda

-----METADATA-----
ExifTool Version Number      : 10.10
File Name                    : imagen1.jpeg
Directory                   : /home/miguel/Dropbox/DAW/Primero/LenguajeDeMar
cas/ExamenDiciembre2016/imagenes
File Size                    : 10 kB
File Modification Date/Time  : 2016:12:12 10:35:07+01:00
File Access Date/Time       : 2017:05:31 20:56:56+02:00
File Inode Change Date/Time  : 2017:04:04 22:33:58+02:00
File Permissions             : rw-rw-r--
File Type                    : JPEG
File Type Extension         : jpg
MIME Type                    : image/jpeg
JFIF Version                 : 1.01
Resolution Unit              : None
X Resolution                 : 1
Y Resolution                 : 1
Image Width                  : 420
Image Height                 : 120
Encoding Process             : Baseline DCT, Huffman coding
Bits Per Sample              : 8
```

## 6. Conclusiones

Mediante el presente proyecto, hemos aprendido a desarrollar, mediante el lenguaje de programación Python, y empleando una implementación de código muy sencilla, una aplicación que nos permita encontrar los metadatos en diversos formatos de archivo.

El programa es eficiente y ahorra en la tarea de importación de excesivas librerías externas, lo cual es una ventaja en la usabilidad para el usuario.

No obstante, como contrapartida, aún cabría mejorar la portabilidad del mismo en ulteriores desarrollos. El programa ha sido testeado y verificado en entornos Linux convenientemente preparados.

# 7. Bibliografía

Algunos elementos de bibliografía y webgrafía empleados:

- <http://blog.powerdata.es/el-valor-de-la-gestion-de-datos/que-son-los-metadatos-y-cual-es-su-utilidad>
- <https://es.wikipedia.org/wiki/Metadato>
- [https://es.wikipedia.org/wiki/Big\\_data](https://es.wikipedia.org/wiki/Big_data)
- Violent Python, a CookBook for Hackers, Forensic Analyst, Penetration Testers and Search Engineers, O'Connor, T.J, Elsevier, USA, 2013,