

# Reporte Quincenal

17/05/2024

---

Ismael Montoro Peñasco  
Fundación Goodlob

<b>Acciones llevadas a cabo.....</b>	<b>2</b>
<b>Avances realizados.....</b>	<b>2</b>
Tarea 1 - Introducción a PySide.....	2
Tarea 2 - Moduladores de Voz.....	3
Tarea 3 - Transcripciones de Audio.....	4
Tarea 4 - Conversores de formatos de archivo.....	6
Tarea 5 - Apuntes.....	8
Apunte 1.....	9
Apunte 2.....	10
Apunte 3.....	11
<b>Planes para el próximo mes.....</b>	<b>11</b>

## Acciones llevadas a cabo

Dentro de unas semanas recibiré la titulación del curso con los vouchers para presentarme a los exámenes del PCAP y el PCPP. Me centraré en estudiar para el examen del PCAP y el PCPP.

## Avances realizados

### Tarea 1 - Introducción a PySide

He realizado una introducción a la librería PySide, creando mi primera ventana, y mis primeros frames.

```
1 from PySide6.QtWidgets import *
2 import sys
3
4 class mi_primera_ventana(QMainWindow):
5
6     def setupUi(self):
7         # Ventana
8         self.setFixedSize(1280, 720) # Ancho y Alto
9
10        # Frame
11        self.frame_1 = QFrame(self)
12        self.frame_1.setGeometry(20, 10, 500, 200) # Posicion Eje X, Posicion Eje Y, Ancho, Alto
13        self.frame_1.setStyleSheet("background: black;")
14
15        self.frame_2 = QFrame(self)
16        self.frame_2.setGeometry(20, 250, 100, 200)
17        self.frame_2.setStyleSheet("background: white;")
18
19 # Fuente: https://www.youtube.com/watch?v=JjNgU1ntdT0
20
```

## Tarea 2 - Moduladores de Voz

He realizado moduladores de texto a voz básicos con Python, con dos alternativas que son pytsx3 y gTTS.

```
Investigaciones > Moduladores de Voz > Src > Text-to-Voice > 1_prueba_pytsx3.py > ...
1 import pytsx3
2
3 # Crearemos el motor para generar audios
4 motor = pytsx3.init()
5
6 # Crearemos una coleccion con los audios
7 # necesarios para generar un audio solo
8 # hay dos de hombre y mujer
9 voces = motor.getProperty('voices')
10
11 # Seleccionaremos la voz de hombre indicando
12 # el indice 0, si quisieras la voz de mujer
13 # tienes que pedir el indice 1
14 motor.setProperty('voice', voces[0].id)
15
16
17 from gtts import gTTS # pip install gTTS
18 from IPython.display import Audio # pip install IPython
19
20 # FUENTE: https://www.youtube.com/watch?v=4WU-CUht4sk
21
22 # Creamos el mensaje
23 mensaje = "Hola, ¿Como estás?"
24
25 # Instanciamos el objeto gTTS, usando los parametros
26 # text para indicar el mensaje, y lang para indicar
27 # el idioma en el que esta escrito el mensaje
28 tts = gTTS(text=mensaje, lang="es")
29
30 archivo_salida = 'mi_mensaje.mp3'
31 tts.save(archivo_salida)
32
33 Audio(archivo_salida)
```

## Tarea 3 - Transcripciones de Audio

He realizado un programa de transcripciones de audio a texto de manera básica.

```
1  import speech_recognition as sr # pip install SpeechRecognition
2
3  # FUENTE: https://www.youtube.com/watch?v=P51RHwue4aw
4
5  # Inciaremos un reconocedor de voz
6  reconocedor = sr.Recognizer()
7
8  # Capturar el audio guardado. Necesitaremos el archivo de
9  # audio en formato wav, no vale mp3 u otros formatos
10 archivo = 'Archivo.wav'
11 audio = sr.AudioFile(archivo)
12
13 # Abriremos el objeto con with
14 with audio as source:
15     audio = reconocedor.record(source)
16
17 # Convertiremos el audio a texto
18 texto: str = reconocedor.recognize_google(audio)
19
20 # Lo imprimimos
21 print(texto.lower())
22
```

Incluso he aprendido a realizar dichas transcripciones grabandose con el micrófono en stream.

```
1 import speech_recognition as sr # pip install SpeechRecognition
2 import pyaudio
3
4 # FUENTE: https://www.youtube.com/watch?v=P5lRHwue4aw
5
6 # Inciaremos un reconocedor de voz
7 reconocedor = sr.Recognizer()
8
9 # Abriremos el objeto con with
10 with sr.Microphone() as micro:
11     # Ajustaremos el script para utilice el microfono por defecto,
12     # y que la duracion de la grabacion sea de 0.4
13     reconocedor.adjust_for_ambient_noise(micro, duration=0.4)
14     # Grabaremos el audio con el microfono por defecto
15     audio = reconocedor.listen(micro)
16     # Convertiremos el audio a texto
17     texto: str = reconocedor.recognize_google(audio)
18     # Lo imprimimos
19     print(texto.lower())
20
```

## Tarea 4 - Conversores de formatos de archivo

He visto como crear ficheros ODT de LibreOffice con la librería `odfdo`

```
1  from odfdo import Document, Paragraph
2
3  documento = Document('text')
4
5  texto = "Hola mundo!"
6  parrafo = Paragraph(texto)
7  documento.body.append(parrafo)
8
9  archivo = "1_Crear_documentos.odt"
10 documento.save(archivo)
11
```

También he aprendido a crear documentos vacíos.

```
1  from odfdo import Document, Paragraph
2
3  documento = Document('text')
4
5  archivo = "2_Crear_documentos_vacios.odt"
6  documento.save(archivo)
7
```

He aprendido a generar cabeceras en documento ODT.

```
15 cabecera = Header(1, "Titulo 1.1")
16 documento.body.append(cabecera)
17
18 cabecera = Header(2, "Titulo 2")
19 documento.body.append(cabecera)
20
21 parrafo = Paragraph(texto)
22 documento.body.append(parrafo)
23
24 cabecera = Header(3, "Titulo 3")
25 documento.body.append(cabecera)
26
27 cabecera = Header(4, "Titulo 4")
28 documento.body.append(cabecera)
29
30 cabecera = Header(5, "Titulo 5")
31 documento.body.append(cabecera)
32
33 cabecera = Header(6, "Titulo 6")
34 documento.body.append(cabecera)
35
36 cabecera = Header(7, "Titulo 7")
37 documento.body.append(cabecera)
```



He aprendido a crear estilos para formatear textos.

```
50  ## Creas los estilos
51  nombre_primer_estilo = "mi_primer_estilo" # Defines el nombre del estilo
52  estilo_1 = Style( # Creas el estilo
53      family="text",
54      name=nombre_primer_estilo,
55      display_name=nombre_primer_estilo,
56      bold=True,
57      color="darkgoldenrod",
58  )
59  documento.insert_style(estilo_1) # Insertado el estilo en el documento
60
61  nombre_segundo_estilo = "mi_segundo_estilo"
62  estilo_2 = Style(
63      family="text",
64      name=nombre_segundo_estilo,
65      display_name=nombre_segundo_estilo,
66      italic=True,
67      size="120%",
68      color="lime",
69  )
70  documento.insert_style(estilo_2)
```

He aprendido a insertar estilos dentro de un parrafo.

```
85  ## Insertar los estilos en el parrafo, indicando el texto a formatear
86  parrafo.set_span(nombre_primer_estilo, regex="dolor") # Editara el primer texto "dolor"
87  parrafo.set_span(nombre_primer_estilo, regex="Vestibulum ullamcorper, libero")
88  parrafo.set_span(nombre_segundo_estilo, regex=r"\w+ing") # Puedes indicar los textos a formatear con regex
```

## Tarea 5 - Apuntes

En esta sección comentaré sobre mis avances que he realizado con mis estudios de PCAP y PCPP.

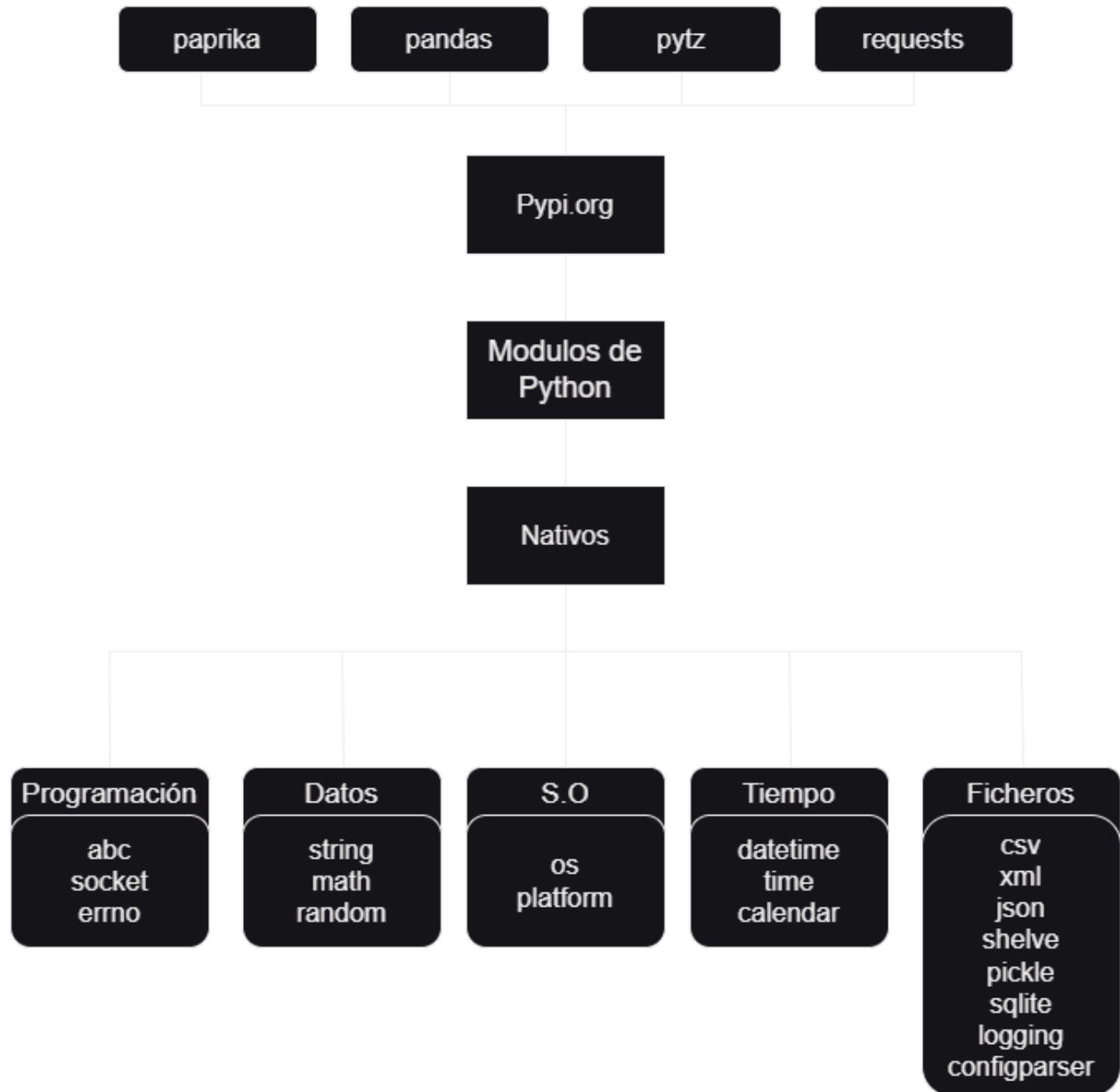
## Apunte 1

He realizado apuntes de algunos errores que surgen al interactuar con archivos.

```
match ex.errno: # A partir de Python 3.7
    case errno.EACCES:
        print("No tienes permisos para abrir ese fichero.")
    case errno.ENOENT:
        print("Ese archivo o directorio no existe.")
    case errno.ENOSPC:
        print("No queda espacio en el dispositivo.")
    case errno.EFBIG:
        print("El archivo es demasiado grande para abrirlo.")
    case errno.EMFILE:
        print("Demasiados archivos abiertos.")
    case errno.EEXIST:
        print("El archivo ya existe de antes.")
    case errno.EISDIR:
        print("Es un directorio.")
    case errno.EBADF:
        print("Numero de archivo incorrecto.")
    case _: # default
        print("Otro error")
```

## Apunte 2

He realizado algunas modificaciones en el esquema del diagrama de librerías de Python.



### Apunte 3

He realizado los apuntes de los permisos que se utilizan para acceder e interactuar con archivos.

Permisos de Fichero		
Permiso	Agregado	Tipo de archivo
<b>r</b> (Lectura) <b>w</b> (Crear o Escribir con sobrescritura) <b>x</b> (Crear sin sobrescritura) <b>a</b> (Escribir por el final del archivo)	<b>+</b> (Agregar permiso de lectura o escritura al permiso)	<b>t</b> (Archivo de texto) <b>b</b> (Archivo binario)

### Planes para el próximo mes

Me centraré en estudiar para el examen del PCAP y el PCPP.