

# Modelo de Diseño

PDF-reader

Versión 0.2.1

---

## Bitácora de Cambios

Fecha	Descripción del Cambio	Autor del Cambio
31-03-2024	Inicio	Mario Hernandez
09-04-2024	Release '0.1.0'	Mario Hernandez
16-04-2024	Release '0.2.0'	Mario Hernandez
16-04-2024	Release '0.2.1'	Mario Hernandez

# Tabla de contenido

<b>I. Introducción</b>	<b>4</b>
<b>II. Criterios</b>	<b>4</b>
Arquitectura:	4
Herramientas:	4
<b>III. Inventario de Artefactos</b>	<b>4</b>
<b>IV. Restricciones de la Arquitectura</b>	<b>5</b>
Restricciones:	5
<b>V. Criterios de Orden de Construcción</b>	<b>5</b>
Orden de construccion:	5
Construccion en base a la prioridad:	6
<b>VI. Definición de las Pantallas del Sistema</b>	<b>6</b>
Diagrama de Navegacion de pantallas:	6
Consola:	7
Finder (Explorador de archivos):	7
<b>VII. Definición de interfaces del sistema</b>	<b>8</b>
<b>Interfaz - FileManager</b>	<b>8</b>
select_file()	8
select_file_output()	9
<b>Interfaz - PDFManager</b>	<b>10</b>
extract_text_from_pdf()	11
write_file_pdf()	12
<b>Interfaz - TextAnalyzer</b>	<b>13</b>
create_dictionary()	13
count_repetitions()	14

# I. Introducción

- Leer y extraer el texto de dos (.pdf).
- Generar dos diccionarios, uno de frases y otro de palabras.
- Comparar cuantas veces se repiten las frases o palabras del diccionario en el otro (.pdf).
- Generar dos (.pdf) final que contenga la cantidad de repeticiones de frases / palabras.

## II. Criterios

### Arquitectura:

- Se opto por una arquitectura modular ya que es mas sencilla darle mantenimiento.
- Solo puede leer archivos (.pdf).
- Usar el subprocesso del Finder para abrir y guardar los archivos.
- El resultado generarlo en (.pdf).

### Herramientas:

- **Documentacion:** notion era buena opcion pero fue requisito hacerlo en word.
- **PyPDF2:** habia varias opciones pero esta era mas simple a las otras manteniendo una calidad buena de manipulacion de (.pdf)
- **Re:** es una librería nativa de python que es muy potente para manipular textos.
- **Tkinter:** es una librería nativa de python que te permite hacer GUI simples, usado para abrir archivos de manera dinamica y practica
- **FPDF:** librería que permite crear y manipular archivos (.pdf) es util si deseas crear un (.pdf) desde cero

## III. Inventario de Artefactos

Nombre	Tipo	Archivo	Descripción
CLI	consola	main.py	Abre el finder para seleccionar pdf
re	Libreria	Regex.py	Librería que te permite usar expresiones regulares para manipular textos
tkinter	Libreria	Tkinter.py	Librería que te permite generar GUI simples en python por medio de ventanas y widgets
PyPDF2	Libreria	PyPDF2.py	Librería que te permite leer (.pdf) de manera simple
FPDF	Libreria	FPDF.py	Librería que te permite crear y manipular (.pdf)
phrases_repetitions	archivo	phrases_repetitions.pdf	(.pdf) de un diccionario de frases
words_repetitions	archivo	words_repetitions.pdf	(.pdf) de un diccionario de palabras

select_file	Funcion	main.py	Abre el finder para seleccionar archivos
select_file_output	Funcion	main.py	Abre el finder para elegir ubicación para guardar archivos
extract_text_from_pdf	Funcion	main.py	Extrae el texto de un pdf
create_dictionary	Funcion	main.py	Crea dos diccionarios a travez de una lista
count_repetitions	Funcion	main.py	Cuenta las repeticiones de 2 diccionarios en un texto
write_file_pdf	funcion	main.py	Crea un archivo (.pdf) con las repeticiones de cada frase

## IV. Restricciones de la Arquitectura

### Restricciones:

- El software debe ser capaz de extraer el texto de un (.pdf).
- El software debe contemplar errores que puedan surgir.
- El software solo debe aceptar (.pdf).
- El resultado generarlo en (.pdf).
- Necesita tener python 3 instalado.
- El software funciona en PC's.

## V. Criterios de Orden de Construcción

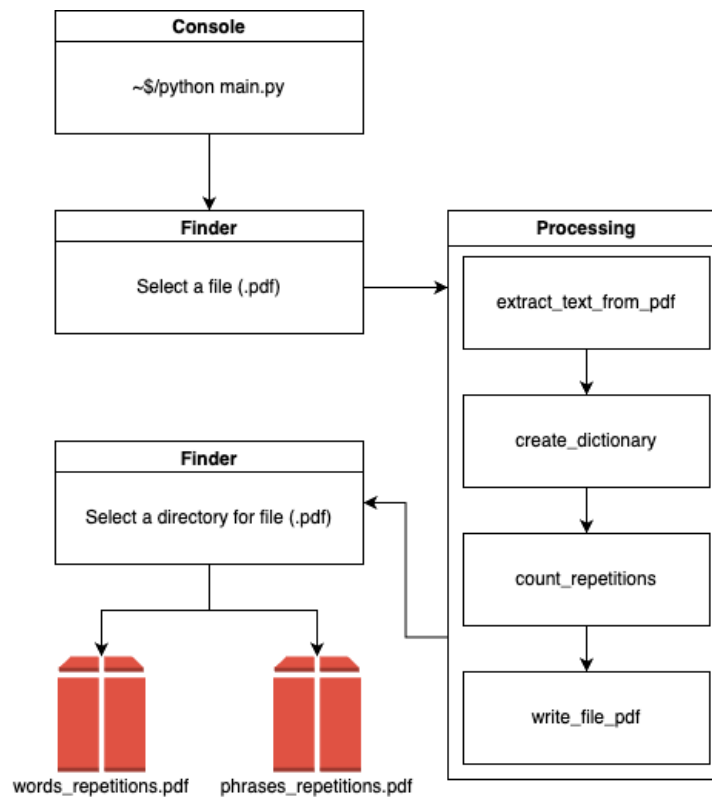
### Orden de construccion:

Prioridad	Componente
CLI	pythonX main.py
select_pdf	Abre finder para buscar tipo (.pdf), como pdf principal
select_pdf	Abre finder para buscar tipo (.pdf) como pdf diccionario
extract_text_from_pdf	Extrae el texto de un (.pdf) y lo guarda en una lista
extract_text_from_pdf	Extrae el texto de un (.pdf) y lo guarda en una lista
create_dictionary	Crea un diccionario de frases y otro de palabras
count_repetitions	Cuenta la cantidad de repeticiones de frases y palabras
select_file_output	Selecciona el destino del archivo de las repeticiones de frases
select_file_output	Selecciona el destino del archivo de las repeticiones de palabras
write_file_pdf	Crea un nuevo (.pdf) con las frases y sus repeticiones
write_file_pdf	Crea un nuevo (.pdf) con las palabras y sus repeticiones

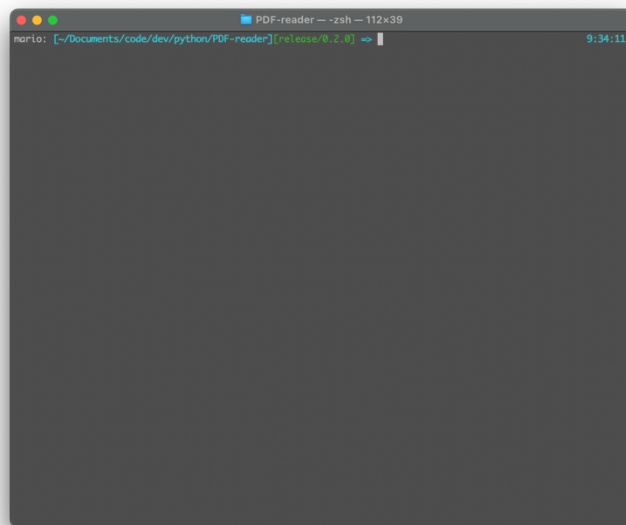
**Construccion en base a la prioridad:**

Prioridad	Componente
CLI	pythonX main.py
select_pdf	Abre finder para buscar tipo (.pdf), como pdf principal
select_pdf	Abre finder para buscar tipo (.pdf) como pdf diccionario
extract_text_from_pdf	Extrae el texto de un (.pdf) y lo guarda en una lista
extract_text_from_pdf	Extrae el texto de un (.pdf) y lo guarda en una lista
create_dictionary	Crea un diccionario de frases y otro de palabras
count_repetitions	Cuenta la cantidad de repeticiones de frases y palabras
select_file_output	Selecciona el destino del archivo de las repeticiones de frases
select_file_output	Selecciona el destino del archivo de las repeticiones de palabras
write_file_pdf	Crea un nuevo (.pdf) con las frases y sus repeticiones
write_file_pdf	Crea un nuevo (.pdf) con las palabras y sus repeticiones

## VI. Definición de las Pantallas del Sistema

**Diagrama de Navegacion de pantallas:**

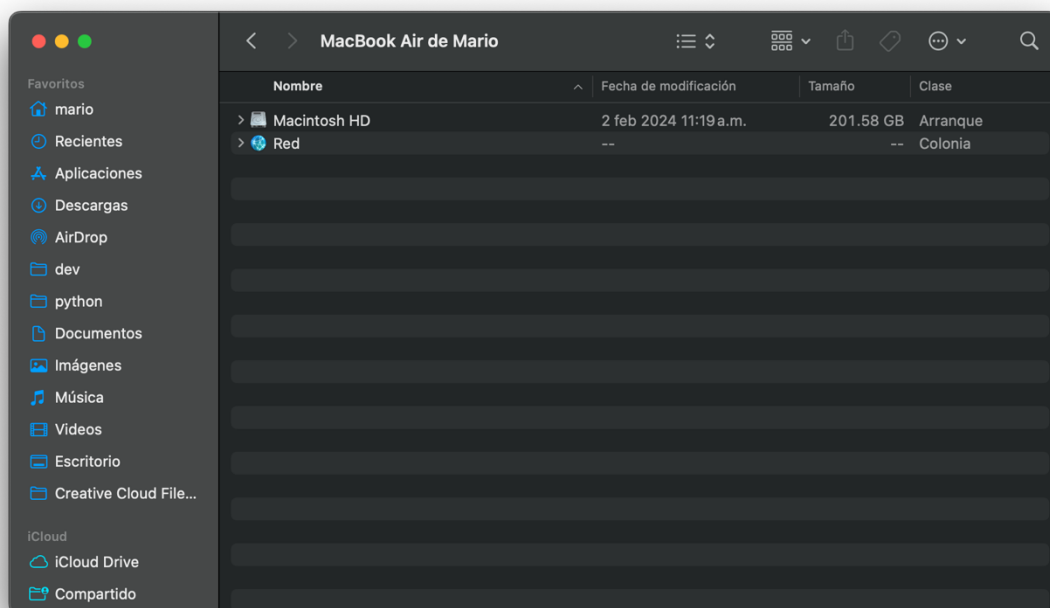
## Consola:



## Objetivo de la Consola:

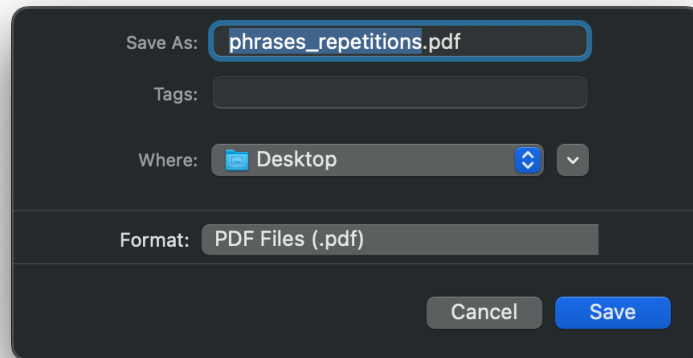
Pantalla principal del sistema para poder ejecutar nuestro programa con el comando “pythonX main.py” esto en el directorio en donde se encuentra nuestro archivo. Tambien es usada para darle indicaciones al usuario.

## Finder (Explorador de archivos):



**Objetivo del finder:**

Esta pantalla del finder le permite seleccionar al usuario la ubicación de donde se encuentran los (.pdf). Solo permite seleccionar (.pdf) para evitar errores.

**Objetivo de finder[save as]:**

Esta pantalla del finder le permite seleccionar al usuario la ubicación de donde se quiere guardar los resultados de los dos (.pdf). Solo permite (.pdf) para evitar errores.

## VII. Definición de interfaces del sistema

### Interfaz - FileManager

**Objetivo:**

Esta interfaz proporciona métodos para seleccionar archivos, tipo de archivos y ubicación de ellos en el dispositivo del usuario

**Origen:**

main.py

**Firma de operaciones:**

Operación: select\_file()

Operación: select\_file\_output()

**Propiedades:**

No hay propiedades publicas

**select\_file()**

**Descripcion:**

Permite al usuario la selección de un archivo mediante el finder (explorador de archivos)

**Precondiciones:**

- Se requiere el uso de tkinter

**Postcondiciones:**

- Devolver la ruta del archivo seleccionado por el usuario



```
pdf_file: str = select_file('PDF', ('PDF Files', '*.pdf'))
```

**Destino:**

pdf\_file, pdf\_dictionary

**Evento:**

Llamar a la funcion select\_file()

**Parametros:**

No.	Parámetro	E/S	Tipo de dato	Descripción
1	title	Entrada	str	El título para el cuadr diálogo.
2	filetypes	Entrada	tuple[str, str]	Tipos de archivos permitidos.
3	file_path	Salida	str	Ruta del archivo de salida

**Excepciones:**

No.	Tipo de excepción	Descripción	Causas de excepción
N/A	N/A	N/A	N/A

**Protocolo:**

Ninguna restriccion en especifico.

**Nivel de servicio:** Usabilidad

**Guia de uso:**

Llamar la funcion cuando se necesite seleccionar un archivo.

**Notas:**

En el parametro de filetypes se puede especificar si se requiere de un tipo de archivo en especifico.

**Pendientes:**

No hay pendientes por resolver.

**select\_file\_output()****Descripcion:**

Permite al usuario la proporsionar un nombre y seleccionar la ubicación de destino para guardar un archivo mediante el finder

**Precondiciones:**

- Se requiere el uso de tkinter

**Postcondiciones:**

- Devuelve la ruta y nombre del archivo seleccionado por el usuario

```
phrases_output_file: str = select_file_output('phrases_repetitions'('PDF Files', '*.pdf'))
```

**Destino:**

phrases\_output\_file, words\_output\_file

**Evento:**

Llamar a la funcion select\_file\_output()

**Parametros:**

No.	Parámetro	E/S	Tipo de dato	Descripción
1	title	Entrada	str	El nombre para el archivo de salida
2	filetypes	Entrada	tuple[str, str]	Tipos de archivos permitidos.
3	file_path	Salida	str	Ruta del archivo de salida

**Excepciones:**

No.	Tipo de excepción	Descripción	Causas de excepción
N/A	N/A	N/A	N/A

**Protocolo:**

Ninguna restriccion en especifico.

**Nivel de servicio:** Usabilidad

**Guia de uso:**

Llamar la funcion cuando se necesite seleccionar una ubicación para la salida de un archivo.

**Notas:**

En el parametro de filetypes se puede especificar si se requiere de un tipo de archivo en especifico,  
En el parametro de title se da un nombre default que el usuario puede elegir cambiar.

**Pendientes:**

No hay pendientes por resolver.

## Interfaz - PDFManager

**Objetivo:**

Esta interfaz proporciona métodos para la lectura y procesamiento de archivos (.pdf).

**Origen:**

main.py

**Firma de operaciones:**

Operación: extract\_text\_from\_pdf()

Operación: write\_file\_pdf()

**Propiedades:**

No hay propiedades publicas

**extract\_text\_from\_pdf()****Descripción:**

Esta funcion se encarga de extraer el texto que contiene un archivo (.pdf).

**Precondiciones:**

- Se requiere el uso de PyPDF2
- Se requiere de la ruta del archivo
- El archivo tiene que ser (.pdf)

**Postcondiciones:**

- Devuelve una lista de strings extraidas de cada pagina del (.pdf)

```
pdf_text: list[str] = extract_text_from_pdf(pdf_file)
```

**Destino:**

pdf\_text, dict\_text

**Evento:**

Llamar a la funcion extract\_text\_from\_pdf()

**Parametros:**

No.	Parámetro	E/S	Tipo de dato	Descripción
1	pdf_file	Entrada	str	Ruta del archivo pdf
2	pdf_text	Salida	list[str]	Lista de strings de todo el texto extraido en el (.pdf)

**Excepciones:**

No.	Tipo de excepción	Descripción	Causas de excepción
N/A	N/A	N/A	N/A

**Protocolo:**

Ninguna restriccion en especifico.

**Nivel de servicio:** **Funcionalidad****Guía de uso:**

Llamar la funcion para extraer el texto de un (.pdf).

**Notas:**

Esta operación es la principal del sistema ya que despues se usara para identificar repeteticiones con otros strings

**Pendientes:**

No hay pendientes por resolver.

**write\_file\_pdf()****Descripción:**

Esta función escribe los resultados de un diccionario en un archivo (.pdf)

**Precondiciones:**

- Se espera que el diccionario este en el formato de python
- Se requiere de la ruta del archivo sea válida
- Se requiere el uso de FPDF

**Postcondiciones:**

- Crea un archivo .txt donde cada línea es una frase / palabra con su valor

write\_file\_txt(phrases\_repetitions, phrases\_output\_file)

**Destino:**

phrases\_repetitions.pdf

**Evento:**

Llamar a la función write\_file\_txt()

**Parámetros:**

No.	Parámetro	E/S	Tipo de dato	Descripción
1	dictionary	Entrada	Dict[str, int]	El diccionario que se desea escribir en el archivo.
2	output_file	Entrada	str	La ruta del archivo de salida.
3	sort	Opcional	bool	Indica si se quiere clasificar por orden alfabético. Default = False
4	output_file.pdf	Salidas	archivo	Archivo con los resultados

**Excepciones:**

No.	Tipo de excepción	Descripción	Causas de excepción
N/A	N/A	N/A	N/A

**Protocolo:**

Ninguna restricción en específico.

**Nivel de servicio:** **Funcionalidad****Guía de uso:**

Llamar la función para crear un .txt por medio de un diccionario.

**Notas:**

Esta operación es bastante útil ya que escribe los resultados en un .txt para que el usuario los pueda manipular y visualizar con facilidad.

**Pendientes:**

No hay pendientes por resolver.

## Interfaz - TextAnalyzer

**Objetivo:**

Esta interfaz proporciona métodos para el análisis de texto y el conteo de repeticiones de frases o palabras

**Origen:**

main.py

**Firma de operaciones:**

Operación: create\_dictionary()

Operación: count\_repetitions()

**Propiedades:**

No hay propiedades públicas

### create\_dictionary()

**Descripción:**

Esta función crea dos diccionarios uno de frases y otro de palabras a partir de una lista de texto

**Precondiciones:**

- Se espera que el texto esté proporcionado en una lista
- Se requiere el uso de re.

**Postcondiciones:**

- Devuelve los dos diccionarios con cada frase y palabra con sus repeticiones iniciales en 0

```
dict_phrases, dict_words = create_dictionary(dict_text)
```

**Destino:**

dict\_phrases, dict\_words

**Evento:**

Llamar a la función create\_dictionary()

**Parámetros:**

No.	Parámetro	E/S	Tipo de dato	Descripción
1	pdf_file	Entrada	str	Ruta del archivo (.pdf)
2	phrases	Salida	dict[str, int]	Diccionario de frases con sus valores iniciales = 0

3	words	Salida	dict[str, int]	Diccionario de palabras con sus valores iniciales = 0
---	-------	--------	----------------	---

**Excepciones:**

No.	Tipo de excepción	Descripción	Causas de excepción
N/A	N/A	N/A	N/A

**Protocolo:**

Ninguna restriccion en especifico.

**Nivel de servicio:** Funcionalidad**Guia de uso:**

Llamar la funcion cuando se necesite hacer un diccionario de frases y palabras.

**Notas:**

Esta operación es de las mas importantes por que nos permite generar un diccionario que se usara para contar las repeticiones de cada frase / palabra. Por medio de las expresiones regulares para obtener ciertos patrones en el texto.

**Pendientes:**

No hay pendientes por resolver.

**count\_repetitions()****Descripcion:**

Cuenta la repeticion de los valores de los diccionarios en un texto.

**Precondiciones:**

- Se espera que el texto este proporcionado en una lista
- Se requiere de los diccionarios de frases y palabras

**Postcondiciones:**

- Devuelve los dos diccionarios actualizados con los valores de repeticion de cada frase / palabra

```
phrases_repetitions, words_repetitions = count_repetitions(pdf_text, dict_phrases, dict_words)
```

**Destino:**

phrases\_repetitions, words\_repetitions

**Evento:**

Llamar a la funcion count\_repetitions()

**Parametros:**

No.	Parámetro	E/S	Tipo de dato	Descripción
1	text	Entrada	list[str]	El texto en el cual se desea contar las repeticiones.

2	phrases_dict	Entrada	dict[str, int]	El diccionario de frases
3	words_dict	Entrada	dict[str, int]	El diccionario de palabras
4	phrase_repetition	Salida	dict[str, int]	El diccionario de frases actualizado con sus repeticiones en el texto
5	words_repetition	Salida	dict[str, int]	El diccionario de palabras actualizado con sus repeticiones en el texto

**Excepciones:**

No.	Tipo de excepción	Descripción	Causas de excepción
N/A	N/A	N/A	N/A

**Protocolo:**

Ninguna restriccion en especifico.

**Nivel de servicio:** **Funcionalidad**

**Guia de uso:**

Llamar la funcion cuando se necesite contar las repeticiones de frases / palabras en un texto.

**Notas:**

Esta operación regresa los diccionarios de frases y palabras con sus valores de repeticion en el texto

**Pendientes:**

No hay pendientes por resolver.