

TRABAJO PRÁCTICO INTEGRADOR

**HERRAMIENTAS DE PROCESAMIENTO DE
DATOS O AUTOMATIZACIÓN**

MATERIA: PROGRAMACIÓN AVANZADA

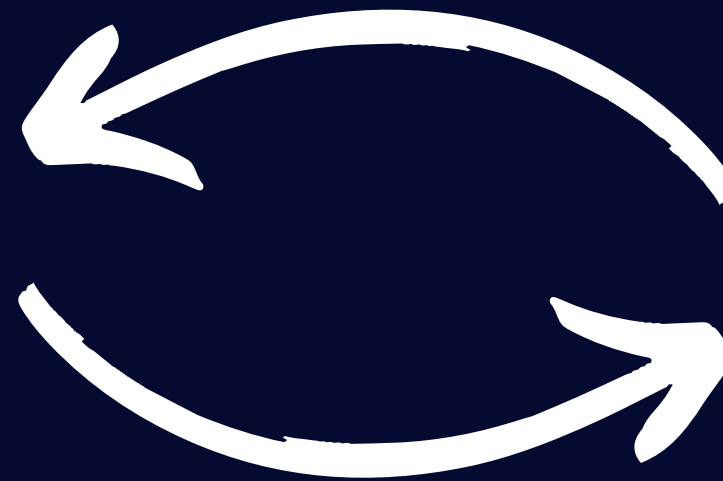
GRUPO: 17

PROFESOR: GIANLUCA PIRIZ

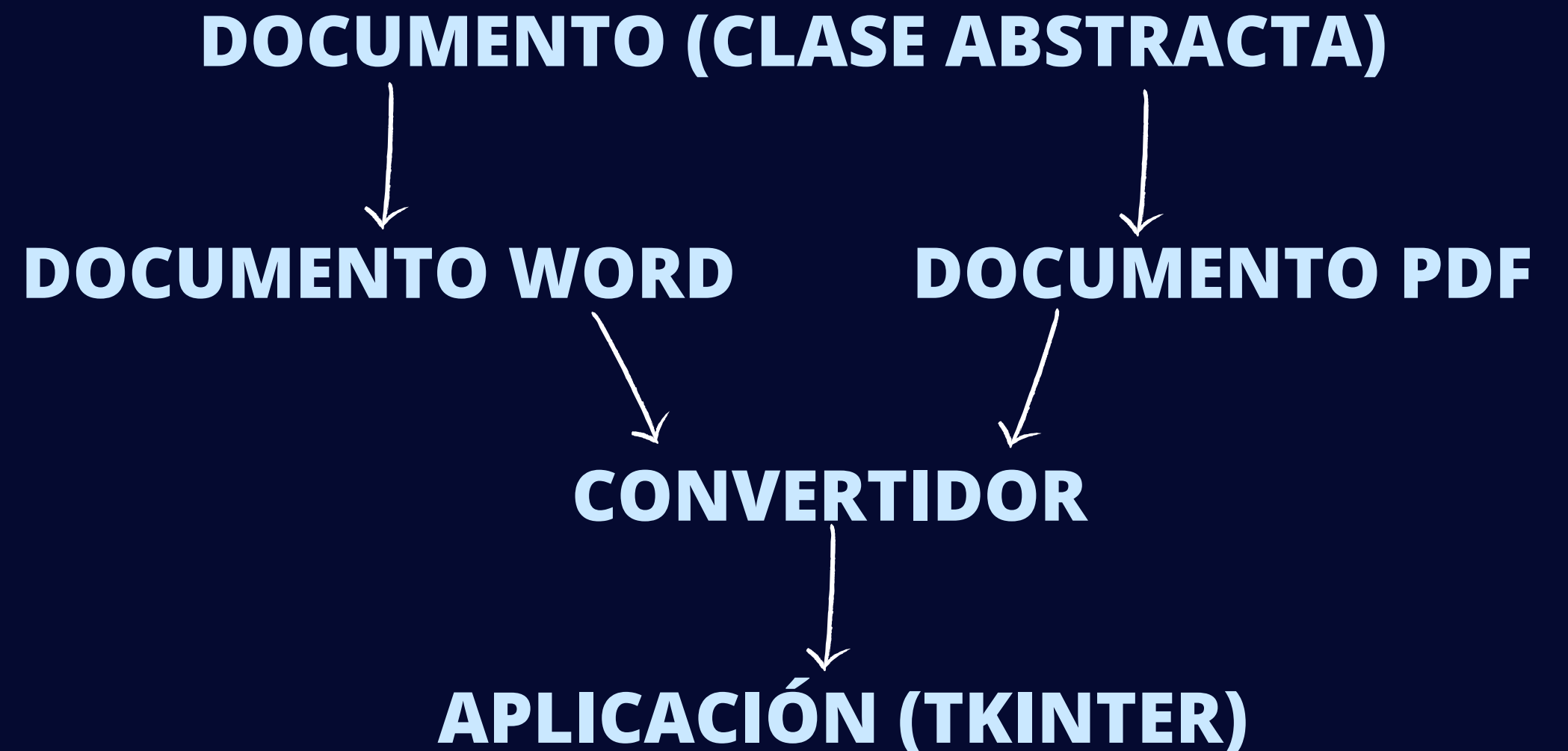
INTEGRANTES: MATIAS HAJNY - NICOLÁS LEGUNDA

NICOLE QUINTANA - JULIETA ROFRANO

CONVERTIDOR DE ARCHIVOS



ESTRUCTURA



COMPONENTES DEL SISTEMA

| Componente | Rol |
|---------------|--|
| Documento | Clase base abstracta con el método convertir(). |
| DocumentoWord | Subclase que representa un documento Word y su conversión a PDF. |
| DocumentoPDF | Subclase que representa un documento PDF y su conversión a Word. |
| Convertidor | Controla la lógica de conversión (usa composición). |
| Aplicación | Administra la interfaz gráfica y la interacción con el usuario. |

CLASES PRINCIPALES

| Clase | Descripción |
|---------------|--|
| Documento | Clase abstracta que define una interfaz común para todos los documentos. |
| DocumentoWord | Convierte archivos .docx a .pdf utilizando docx2pdf. |
| DocumentoPDF | Convierte archivos .pdf a .docx utilizando pdf2docx. |
| Convertidor | Ejecuta la conversión sin conocer el tipo de documento (polimorfismo). |
| Aplicación | Interfaz gráfica desarrollada con tkinter. |

PRINCIPIOS DE POO APLICADOS

| Concepto | Implementación |
|-----------------|--|
| Encapsulamiento | Cada clase gestiona internamente sus atributos y métodos. |
| Herencia | DocumentoWord y DocumentoPDF heredan de la clase abstracta Documento. |
| Polimorfismo | El método convertir() se comporta de forma distinta según la subclase. |
| Abstracción | La clase Documento no se puede instanciar directamente, solo se usa como base. |

INTERFAZ GRÁFICA (GUI)

CARACTERÍSTICAS PRINCIPALES DE LA GUI IMPLEMENTADA CON TKINTER:

- Permite seleccionar el tipo de conversión.
- Ofrece explorador de archivos para seleccionar documentos.
- Permite definir la ruta de salida.
- Muestra mensajes informativos ante errores o éxito.
- Fácil de usar y multiplataforma (aunque docx2pdf depende del SO).

```
...ca con tkinter
...ion:
def __init__(self, master):
    self.master = master
    self.master.title("Conversor Word + PDF")

    self.tipo_conversion = StringVar(value="word_to_pdf")
    self.ruta_archivo = ""

    Label(master, text="Seleccione el tipo de conversión:").pack()
    Radiobutton(master, text="Word + PDF", variable=self.tipo_conv).pack()
    Radiobutton(master, text="PDF + Word", variable=self.tipo_conv).pack()

    Button(master, text="Seleccionar archivo", command=self.seleccionar_archivo).pack()
    self.label_archivo = Label(master, text="Ningún archivo seleccionado")
    self.label_archivo.pack()

    Label(master, text="Ruta de salida:").pack()
    self.entry_salida = Entry(master, width=50)
    self.entry_salida.pack(pady=5)

    Button(master, text="Convertir", command=self.convertir).pack()

    def seleccionar_archivo(self):
        self.ruta_archivo = self.entry_salida.get()
```

Conversor Word + PDF

Seleccione el tipo de conversión:

☒ Word → PDF

☐ PDF → Word

Seleccionar archivo

Ningún archivo seleccionado

Ruta de salida:

Convertir

VENTAJAS DEL DISEÑO

ESCALABLE



Puede adaptarse fácilmente para incluir nuevos formatos como .txt, .html, etc.

MANTENIBLE



Cada clase tiene una única responsabilidad clara.

PORTATIL



Se puede utilizar tanto con GUI como desde la línea de comandos.

CONSIDERACIONES TÉCNICAS

- La librería docx2pdf requiere que Microsoft Word esté instalado (solo funciona correctamente en Windows y macOS).
- pdf2docx puede fallar con archivos escaneados o con gran cantidad de imágenes.
- La interfaz está pensada para ser utilizada por usuarios sin conocimientos técnicos.



CONCLUSIÓN

El proyecto logra cumplir con el objetivo propuesto de automatizar la conversión de documentos de manera eficiente y con una experiencia de usuario intuitiva. Además, se consolidaron conocimientos clave de POO y uso de librerías externas en Python, logrando un producto funcional, profesional y extendible.

POSIBLES MEJORAS FUTURAS

- Incorporar soporte para más tipos de archivos (HTML, TXT, ODT).
- Implementar arrastrar y soltar (drag & drop) para mejorar la UX.
- Añadir traducción automática de textos al momento de la conversión.
- Crear una versión web usando Flask o Django para mayor accesibilidad.



GRACIAS