1. **Código Fuente del Programa “digital-sign(primero-intermedios).py”: [122 lineas]**

# Importamos las librerías nativas de Python.

import os

import re

import socket

from datetime import datetime

from tkinter import filedialog

# Importamos las librerías de terceros en Python.

from docx import Document

from docxtpl import DocxTemplate

from docx2pdf import convert

# Definimos las funciones que serán usadas en la aplicación.

def obtener\_datos\_firma()->tuple [str]:

    """

    La función obtiene los datos de la persona que firma el documento (nombre, rubrica, fecha y hora,

    id del equipo, dirección ip del equipo) y retorna los valores.

    Returns:

        nombre\_persona (str): nombre de la persona que firma.

        rubrica\_persona (str): rubrica de la persona que firma.

        fecha\_hora (str): fecha y hora en la que se ejecuta la firma.

        id\_equipo (str): id del equipo donde se firma.

        direccion\_ip (str): ip del equipo donde se firma.

    """

    while True:

        nombre\_persona = input("Ingrese su nombre: ")

        if nombre\_persona == "":

            print ("Por favor ingrese su nombre.")

        else:

            break

    while True:

        rubrica\_persona = input("Ingrese su rubrica: ")

        if rubrica\_persona == "":

            print ("Por favor ingrese su nombre.")

        else:

            break

    fecha\_hora = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    id\_equipo = os.environ.get('COMPUTERNAME')

    direccion\_ip = socket.gethostbyname(socket.gethostname())

    return nombre\_persona, rubrica\_persona, fecha\_hora, id\_equipo, direccion\_ip

def renderizar\_plantilla (path\_plantilla:str, salida\_plantilla:str, contexto:dict)-> None:

    """

    Renderiza una plantilla de documento Word con el contexto dado y la guarda en el

    directorio de salida.

    Args:

        path\_plantilla (str): Ruta de la plantilla.

        salida\_plantilla (str): Ruta de salida del documento renderizado.

        contexto (dict): Contexto con los datos para renderizar la plantilla.

    """

    # Cargar la plantilla

    document = DocxTemplate(path\_plantilla)

    # Renderizar el documento con el contexto dado

    document.render(contexto)

    # Guardar el documento renderizado en la ruta especificada

    document.save(salida\_plantilla)

    pass

def convertir\_docx\_a\_pdf(archivo\_docx, archivo\_pdf):

    """

    Convierte un archivo .docx a un archivo .pdf.

    Args:

    - archivo\_docx (str): Ruta del archivo .docx a convertir.

    - archivo\_pdf (str): Ruta del archivo .pdf de salida.

    """

    # Convertir el archivo a PDF

    convert(archivo\_docx, archivo\_pdf)

    # Eliminar el archivo temporal

    os.remove(archivo\_docx)

    pass

def firmar\_documento(nombre\_archivo\_pdf: str) -> None:

    """

    El procedimiento recibe la ruta del archivo PDF y retorna el documento firmado.

    Args:

        nombre\_archivo\_pdf (str): Ruta del archivo PDF como una cadena de texto.

    """

    # Obtenemos los datos de la firma del documento.

    nombre\_persona, rubrica\_persona, fecha\_hora, id\_equipo, direccion\_ip = obtener\_datos\_firma ()

    # Creamos el contexto.

    contexto = {

        "sitio\_de\_firma" : f"{nombre\_persona}\n{rubrica\_persona}\n{fecha\_hora}\n{id\_equipo}\n{direccion\_ip}"

    }

    # Creamos la ruta de salida.

    nombre\_archivo\_firmado = f"{os.path.splitext(nombre\_archivo\_pdf)[0]}\_firmado.docx"

    # Renderizamos el documento.

    renderizar\_plantilla (nombre\_archivo\_pdf, nombre\_archivo\_firmado, contexto)

    nombre\_archivo\_pdf\_firmado = f"{os.path.splitext(nombre\_archivo\_pdf)[0]}\_firmado.pdf"

    pass

print ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print ("| Bienvenido a la firma digital de Documentos PDF para Laboratorios Cosmedilab SA. de CV.|")

print ("| Versión 1.0.0 desarrollado por L.Cortez                                                |")

print ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

while True:

    terminos\_condiciones = """

        El presente programa esta desarrollado para Laboratorios Cosmedilab S.A de C.V.

        La reproducción parcial o total de este sistema computacional queda totalmente

        prohibida sin la autorización escrita por parte de Laboratorios Cosmedilab.

        El presente sistema computacional tiene por objetivo el registro de firmas

        electrónicas en cumplimiento con la NOM-241-SSA1-2021 y CFR 21 parte 11.

        """

    print (terminos\_condiciones)

    if input ("\nAcepta los términos y condiciones de uso.\nY para Si\nN para no\n(Y/N): ") in ["y", "si", "Y", "Si"]:

        break

    else:

        exit ()

while True:

    print ("Por favor selecione el documento a firmar.")

    archivo\_path = filedialog.askopenfilename ()

    if archivo\_path == "":

        print ("Se ha cancelado la selección, desea terminar el programa?")

        terminar = input ("Desea terminar el programa (Y/N): ")

        if terminar in ["y", "Y", "Si", "si"]:

            break

    firmar\_documento (archivo\_path)

    if input ("Desea firmar otro documento? (Y continuar/ N terminar): ") in ["y", "Y", "Si", "si"]:

        continue

    break

# última linea de código.

1. **Código fuente del programa “digital-sign(final).py”: [123 lineas]**

# Importamos las librerías nativas de Python.

import os

import re

import socket

from datetime import datetime

from tkinter import filedialog

# Importamos las librerías de terceros en Python.

from docx import Document

from docxtpl import DocxTemplate

from docx2pdf import convert

# Definimos las funciones que serán usadas en la aplicación.

def obtener\_datos\_firma()->tuple [str]:

    """

    La función obtiene los datos de la persona que firma el documento (nombre, rubrica, fecha y hora,

    id del equipo, dirección ip del equipo) y retorna los valores.

    Returns:

        nombre\_persona (str): nombre de la persona que firma.

        rubrica\_persona (str): rubrica de la persona que firma.

        fecha\_hora (str): fecha y hora en la que se ejecuta la firma.

        id\_equipo (str): id del equipo donde se firma.

        direccion\_ip (str): ip del equipo donde se firma.

    """

    while True:

        nombre\_persona = input("Ingrese su nombre: ")

        if nombre\_persona == "":

            print ("Por favor ingrese su nombre.")

        else:

            break

    while True:

        rubrica\_persona = input("Ingrese su rubrica: ")

        if rubrica\_persona == "":

            print ("Por favor ingrese su nombre.")

        else:

            break

    fecha\_hora = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    id\_equipo = os.environ.get('COMPUTERNAME')

    direccion\_ip = socket.gethostbyname(socket.gethostname())

    return nombre\_persona, rubrica\_persona, fecha\_hora, id\_equipo, direccion\_ip

def renderizar\_plantilla (path\_plantilla:str, salida\_plantilla:str, contexto:dict)-> None:

    """

    Renderiza una plantilla de documento Word con el contexto dado y la guarda en el

    directorio de salida.

    Args:

        path\_plantilla (str): Ruta de la plantilla.

        salida\_plantilla (str): Ruta de salida del documento renderizado.

        contexto (dict): Contexto con los datos para renderizar la plantilla.

    """

    # Cargar la plantilla

    document = DocxTemplate(path\_plantilla)

    # Renderizar el documento con el contexto dado

    document.render(contexto)

    # Guardar el documento renderizado en la ruta especificada

    document.save(salida\_plantilla)

    pass

def convertir\_docx\_a\_pdf(archivo\_docx, archivo\_pdf):

    """

    Convierte un archivo .docx a un archivo .pdf.

    Args:

    - archivo\_docx (str): Ruta del archivo .docx a convertir.

    - archivo\_pdf (str): Ruta del archivo .pdf de salida.

    """

    # Convertir el archivo a PDF

    convert(archivo\_docx, archivo\_pdf)

    # Eliminar el archivo temporal

    os.remove(archivo\_docx)

    pass

def firmar\_documento(nombre\_archivo\_pdf: str) -> None:

    """

    El procedimiento recibe la ruta del archivo PDF y retorna el documento firmado.

    Args:

        nombre\_archivo\_pdf (str): Ruta del archivo PDF como una cadena de texto.

    """

    # Obtenemos los datos de la firma del documento.

    nombre\_persona, rubrica\_persona, fecha\_hora, id\_equipo, direccion\_ip = obtener\_datos\_firma ()

    # Creamos el contexto.

    contexto = {

        "sitio\_de\_firma" : f"{nombre\_persona}\n{rubrica\_persona}\n{fecha\_hora}\n{id\_equipo}\n{direccion\_ip}"

    }

    # Creamos la ruta de salida.

    nombre\_archivo\_firmado = f"{os.path.splitext(nombre\_archivo\_pdf)[0]}\_firmado.docx"

    # Renderizamos el documento.

    renderizar\_plantilla (nombre\_archivo\_pdf, nombre\_archivo\_firmado, contexto)

    nombre\_archivo\_pdf\_firmado = f"{os.path.splitext(nombre\_archivo\_pdf)[0]}\_firmado.pdf"

    convertir\_docx\_a\_pdf (nombre\_archivo\_firmado, nombre\_archivo\_pdf\_firmado)

    pass

print ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

print ("| Bienvenido a la firma digital de Documentos PDF para Laboratorios Cosmedilab SA. de CV.|")

print ("| Versión 1.0.0 desarrollado por L.Cortez                                                |")

print ("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

while True:

    terminos\_condiciones = """

        El presente programa esta desarrollado para Laboratorios Cosmedilab S.A de C.V.

        La reproducción parcial o total de este sistema computacional queda totalmente

        prohibida sin la autorización escrita por parte de Laboratorios Cosmedilab.

        El presente sistema computacional tiene por objetivo el registro de firmas

        electrónicas en cumplimiento con la NOM-241-SSA1-2021 y CFR 21 parte 11.

        """

    print (terminos\_condiciones)

    if input ("\nAcepta los términos y condiciones de uso.\nY para Si\nN para no\n(Y/N): ") in ["y", "si", "Y", "Si"]:

        break

    else:

        exit ()

while True:

    print ("Por favor selecione el documento a firmar.")

    archivo\_path = filedialog.askopenfilename ()

    if archivo\_path == "":

        print ("Se ha cancelado la selección, desea terminar el programa?")

        terminar = input ("Desea terminar el programa (Y/N): ")

        if terminar in ["y", "Y", "Si", "si"]:

            break

    firmar\_documento (archivo\_path)

    if input ("Desea firmar otro documento? (Y continuar/ N terminar): ") in ["y", "Y", "Si", "si"]:

        continue

    break

# última linea de código.

1. **Arquitectura del Programa:**

La arquitectura del programa es de tipo Monolítico, esta estructura se elige por la facilidad y desplegar que tienen los proyectos de programación con arquitectura Monolítica. Segundo racional por el cual se elige este tipo de arquitectura es que no se tiene previsto un escalamiento de este desarrollo de forma horizontal (añadir nuevas funciones) así como la aplicación es un desarrollo de pequeña escala. Estas características del proyecto conforme a las buenas practicas de programación encajan de mejor forma con un proyecto de arquitectura Monolítica.

La arquitectura del programa se desglosa de la siguiente forma:

|  |  |
| --- | --- |
| **Captura de Pantalla del Árbol** | **Descripción:** |
|  | En la carpeta general de proyecto (digital-sign 1.0.0) se encuentra la carpeta “app” con los programas compilados listos para para su distribución. A su vez dentro de la carpeta del proyecto se encuentra la carpeta “docs” con la documentación del proyecto (dependencias usadas en el programas, así como el ícono colocado en el programa). De la misma forma se encuentra la carpeta “env” carpeta donde se contiene el ambiente virtual donde se realizaron las pruebas de reto durante su desarrollo. Finalmente se ubica la carpera “src” donde se contiene los dos códigos fuente para las dos aplicaciones desarrolladas. |

1. **Secuencia de Operación de las Aplicaciones:**

|  |  |
| --- | --- |
| **Diagrama de Flujo** | **Descripción** |
|  | 1. Se importan las librerías necesarias para la manipulación de archivos dentro del sistema operativo (sys, os, filedialog), así como aquellas librerías para obtener los datos del equipo, la fecha y hora para el momento en que se realice la firma (socket, datetime). 2. Se importan las librerías para el manejo de documento de tipo Word (docx, docxtpl y docx2pdf). Necesarias para la firma, y conversión del archivo a un formato PDF. 3. Se desarrolla función que obtiene los datos del usuario y retorna estos valores. 4. Se desarrolla el procedimiento para insertar la firma dentro del documento a firmar. 5. Se desarrolla el procedimiento para convertir el documento de Word en un archivo con formato PDF. 6. Se genera el procedimiento que agrupa los procedimientos y funciones anteriores para realizar la firma del documento. 7. Se inicial el bucle para la operación del programa, el cual primero se deben acordar los términos y condiciones el programa previo a su uso. |

|  |  |  |
| --- | --- | --- |
| **ELABORÓ: L.Cortes** | **REVISÓ: J. Castella** | **AUTORIZÓ: C. Pedraza** |
| **FECHA: / /** | **FECHA: / /** | **FECHA: / /** |
| **PUESTO: Inspector de Calidad** | **PUESTO: Coordinador de Validación** | **PUESTO: Responsable Sanitario y Gerente de Aseguramiento de Calidad** |
| **FIRMA:** | **FIRMA:** | **FIRMA:** |