https://doi.org/10.69639/arandu.v12i3.1528

Uso de Python para la generación de una rosa de los vientos con datos locales de viento

Use of Python for Wind Rose Generation with Local Wind Data

Gabriel Eduardo Macías Tello
gabriel.macias@unach.edu.ec
https://orcid.org/0009-0008-2394-3024
Universidad Nacional de Chimborazo
Riobamba – Ecuador

Leonardo Fabián Rentería Bustamante
leonardo.renteria@unah.edu.ec
https://orcid.org/0000-0003-0056-5602
Universidad Nacional de Chimborazo
Riobamba – Ecuador

Artículo recibido: 18 julio 2025 - Aceptado para publicación: 28 agosto 2025
Conflictos de intereses: Ninguno que declarar.

RESUMEN

Este trabajo presenta el desarrollo de una herramienta automatizada para la generación de rosas de los vientos a partir de datos reales de velocidad y dirección del viento, recopilados en la zona industrial norte de Guayaquil durante una jornada de monitoreo. Las mediciones se realizaron con un equipo meteorológico portátil certificado, registrando variables como velocidad del viento, dirección, temperatura y presión atmosférica. A partir de los datos en formato CSV, se diseñó un flujo automatizado que integra Microsoft Excel con macros en VBA y un script en Python, permitiendo importar los datos, procesarlos y generar un reporte técnico con tablas de frecuencia, metadatos y la gráfica de la rosa de los vientos. La integración con Excel responde a la necesidad de ofrecer una interfaz operativa más accesible para técnicos ambientales sin conocimientos avanzados de programación, facilitando su aplicación en contextos reales de monitoreo. Los resultados evidenciaron una predominancia de vientos desde el noroeste, lo cual permite identificar posibles rutas de dispersión de contaminantes hacia zonas residenciales aledañas. La metodología propuesta es replicable, de bajo costo y adaptable a distintos entornos, promoviendo el uso de herramientas digitales abiertas para estudios de calidad del aire.

Palabras clave: rosa de los vientos, monitoreo ambiental, dispersión de contaminantes, Excel, Python

ABSTRACT

This work presents the development of an automated tool for generating wind roses from real wind speed and direction data collected in the northern industrial zone of Guayaquil during a

monitoring session. Measurements were carried out using a certified portable meteorological device, recording variables such as wind speed, direction, temperature, and atmospheric pressure. Based on CSV-format data, an automated workflow was created that integrates Microsoft Excel with VBA macros and a Python script, enabling data import, processing, and the generation of a technical report including frequency tables, metadata, and the wind rose chart. The integration with Excel addresses the need to provide a more user-friendly interface for environmental technicians without advanced programming skills, facilitating its use in real-world monitoring scenarios. The results showed a predominance of winds from the northwest, allowing the identification of potential pollutant dispersion routes toward nearby residential areas. The proposed methodology is replicable, cost-effective, and adaptable to different contexts, promoting the use of open digital tools for air quality studies.

Keywords: wind rose, environmental monitoring, pollutant dispersion, Excel, Python

Todo el contenido de la Revista Científica Internacional Arandu UTIC publicado en este sitio está disponible bajo licencia Creative Commons Atribution 4.0 International.

INTRODUCCIÓN

La calidad del aire en entornos urbanos e industriales representa un componente esencial para la salud ambiental y el bienestar de las poblaciones. En ciudades con alta actividad industrial, como Guayaquil, la caracterización de los patrones del viento resulta fundamental para comprender la dispersión de contaminantes atmosféricos y sus posibles impactos en zonas habitadas o sensibles. La dirección y velocidad del viento influyen directamente en el transporte y la dilución de emisiones provenientes de fuentes fijas, lo que hace indispensable su consideración en estudios de monitoreo y evaluación de la calidad del aire (Ministerio del Ambiente, Agua y Transición Ecológica, 2020; WHO, 2006).

En este contexto, herramientas de visualización como la rosa de los vientos se han consolidado como recursos clave. Esta representación gráfica muestra la frecuencia con la que sopla el viento desde diferentes direcciones, permitiendo identificar patrones dominantes, estimar rutas de dispersión y orientar la ubicación de estaciones de monitoreo. Su utilidad se ha extendido al diseño de estrategias de mitigación de emisiones, la evaluación de riesgos ambientales y la planificación territorial (OMS, 2018).

Tradicionalmente, la generación de rosas de los vientos ha requerido software especializado, lo que puede implicar elevados costos o conocimientos técnicos avanzados. Sin embargo, gracias a la evolución de herramientas de código abierto como Python, es posible automatizar este tipo de análisis de manera eficiente. Python se ha posicionado como uno de los lenguajes más utilizados en entornos científicos y técnicos debido a su versatilidad, amplia comunidad de desarrollo y disponibilidad de bibliotecas como pandas, matplotlib y windrose, que permiten el procesamiento, análisis y visualización de datos meteorológicos.

Con el propósito de ofrecer una alternativa accesible, el presente trabajo desarrolla una herramienta automatizada que integra Microsoft Excel con programación en Python. La elección de Excel responde a su carácter ampliamente conocido y utilizado por técnicos ambientales, lo que permite contar con una interfaz más amigable para el usuario final. De esta forma, es posible ejecutar el análisis de datos con un solo clic desde una plataforma familiar, sin requerir conocimientos avanzados en programación, ampliando así su aplicabilidad práctica.

La investigación se basa en datos recolectados en la zona industrial norte de Guayaquil, durante una jornada de monitoreo meteorológico. A través de un enfoque aplicado, se demuestra la utilidad de la herramienta desarrollada y se evalúan los patrones de viento

registrados. Este análisis permite valorar su potencial aplicación en informes técnicos, especialmente en aquellos relacionados con la normativa ambiental vigente en el país (Ministerio del Ambiente del Ecuador, 2015).

La hipótesis planteada en este estudio sostiene que los patrones de viento registrados en la zona industrial inciden significativamente en la dispersión de contaminantes hacia áreas

circundantes. En consecuencia, su análisis es clave para comprender posibles riesgos ambientales y orientar futuras estrategias de gestión.

MATERIALES Y MÉTODOS

La presente investigación adopta un enfoque cuantitativo, con alcance descriptivo y aplicado, orientado a la automatización del análisis de datos meteorológicos para la caracterización de patrones de viento en contextos industriales. El diseño metodológico es no experimental y de tipo transversal, ya que los datos fueron recolectados durante un único periodo de observación, sin manipular variables ni intervenir en las condiciones del entorno.

El estudio se desarrolló en la zona industrial norte de Guayaquil, en las coordenadas UTM 616681.00 m E y 9772007.00 m N, durante una jornada de monitoreo ambiental realizada el 16 de octubre de 2024. Las mediciones se efectuaron con un equipo meteorológico portátil certificado, que registró variables como velocidad y dirección del viento, temperatura y humedad relativa, a intervalos regulares durante aproximadamente 48 horas.

Los datos fueron exportados en formato CSV y procesados mediante una herramienta automatizada desarrollada por el autor, que integra Microsoft Excel con macros en VBA (Visual Basic for Applications) y un script en Python 3.11. El script, ejecutado desde la propia hoja de cálculo, emplea bibliotecas como pandas, matplotlib, windrose y openpyxl para importar los datos, generar la gráfica de la rosa de los vientos, construir la tabla de frecuencias por sector y exportar un reporte técnico final.

La elección de Excel como entorno operativo responde a su amplia adopción en entornos técnicos y a su facilidad de uso por parte de personal no especializado en programación. Este enfoque permite realizar el análisis completo con un solo clic, sin necesidad de manipular directamente código fuente o utilizar software con licencias comerciales.

Para su funcionamiento adecuado, fue necesario habilitar las macros en Excel e instalar las dependencias de Python mediante el gestor de paquetes pip. La metodología propuesta fue

diseñada con un enfoque replicable, accesible y de bajo costo, lo que favorece su aplicación en futuras campañas de monitoreo ambiental por parte de técnicos, instituciones educativas o consultoras especializadas.

La medición realizada tuvo como propósito validar el funcionamiento operativo de la herramienta automatizada. No obstante, el sistema puede ser aplicado a campañas más prolongadas o a múltiples puntos de monitoreo, permitiendo la comparación de patrones de viento entre distintos días, semanas o estaciones del año. Esto amplía su utilidad potencial en estudios ambientales más complejos y análisis comparativos de calidad del aire.

Este procedimiento se desarrolló tomando como referencia estudios previos sobre la aplicación de rosas de los vientos en entornos urbanos e industriales, así como las normas técnicas de calidad del aire emitidas por la Organización Mundial de la Salud (OMS, 2018) y lineamientos

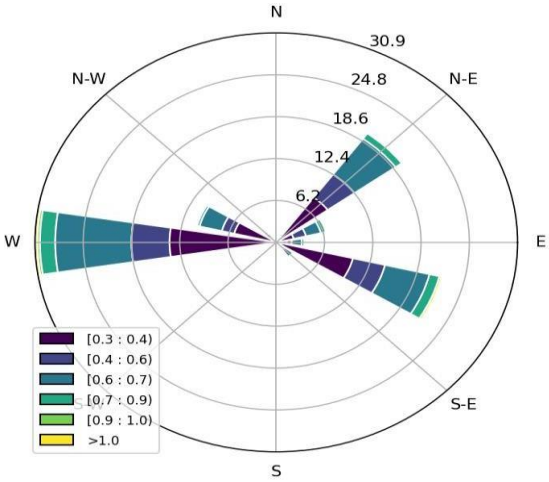
metodológicos para el procesamiento de datos meteorológicos (Python Software Foundation, 2023; Microsoft, 2024).

RESULTADOS Y DISCUSIÓN

La automatización desarrollada mediante la integración de Python con Excel permitió la generación de un reporte técnico detallado sobre el comportamiento del viento en la zona industrial analizada. A partir del procesamiento de los datos obtenidos con el equipo meteorológico portátil, se generó una rosa de los vientos y una tabla de frecuencias clasificadas por dirección y rango de velocidad.

Figura 1

Rosa de los vientos generada para el periodo de medición en la zona industrial norte de Guayaquil



La figura 1 muestra la rosa de los vientos obtenida, donde se observa claramente que las direcciones predominantes durante el periodo de medición fueron el Oeste (W) con 118 registros y el Noreste (NE) con 82 registros, ambos en el rango de velocidad de 0–1 m/s. Estas direcciones concentran la mayoría de los eventos registrados, mientras que otras direcciones como S, SW y N presentan frecuencias significativamente menores, lo que permite identificar un patrón direccional dominante.

Tabla 1

Frecuencia de registros de viento clasificados por dirección y rango de velocidad

Dirección	Rango	frecuencia
W	0–1	118
NE	0–1	82
E	0–1	54
SE	0–1	31
NW	0–1	15
SW	0–1	3

N	0–1	2
E	1–2	1
W	1–2	1

Por su parte, la tabla 1 presenta los datos numéricos agrupados por dirección y rangos de velocidad. Se confirma que la mayoría de las velocidades se mantienen por debajo de los 1 m/s, evidenciando que el viento en la zona es predominantemente suave, con una baja energía de transporte, lo cual tiene implicaciones directas en la dispersión de contaminantes y partículas.

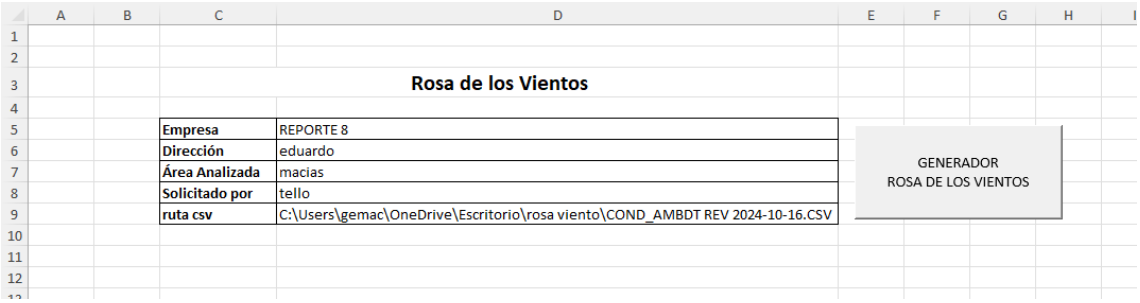
Figura 2

Foto de la medición

Desde el punto de vista operativo, el sistema automatizado permitió obtener estos resultados sin intervención manual significativa, garantizando la trazabilidad y estandarización del proceso. El script en Python extrae automáticamente los datos desde un archivo CSV generado por el equipo de monitoreo, procesa la información, genera los gráficos y las tablas, y finalmente estructura un reporte técnico en Excel, con cabecera editable, logotipo institucional y pie de página profesional.

Figura 3

F*ormulario de ingreso en Excel para la generación automática de rosas de los vientos*



Estructura del sistema automatizado

El flujo de trabajo está compuesto por tres módulos principales:

1. **Formulario en Excel**

Incluye campos para ingresar los datos del sitio de estudio, nombre de la empresa, área analizada, solicitante y ruta al archivo CSV. Este archivo contiene los registros de velocidad y dirección del viento

2. **Macro en VBA (Visual Basic for Applications)**

La macro automatiza la importación de datos desde el archivo CSV hacia la hoja "Datos", y ejecuta el script Python que realiza el análisis completo.

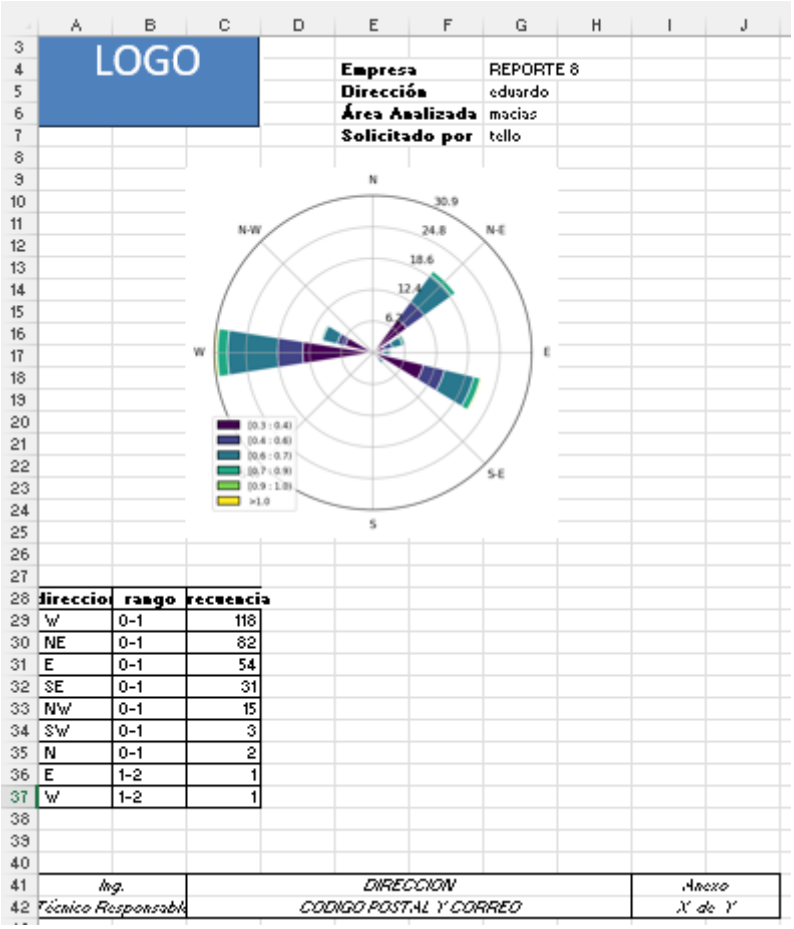
3. **Script en Python**

Ejecutado en Python 3.13, este script realiza:

- Lectura y preprocesamiento del archivo CSV con pandas.
- Clasificación de los datos por rangos de velocidad y sectores direccionales.
- Generación del gráfico tipo rosa de los vientos con windrose.
- Creación de un reporte Excel profesional con openpyxl, que incluye encabezado, gráfico, tabla de frecuencias, pie de página y logotipo institucional.

Figura 4

Reporte final en Excel



Herramientas y configuración

- **Lenguaje de programación**: Python 3.13
- **Bibliotecas utilizadas:** pandas, matplotlib, windrose, openpyxl, win32com (para exportación a PDF, opcional)
- **Plataforma:** Microsoft Excel 365 con macros habilitadas

Estos resultados no solo permiten visualizar la información de forma clara, sino que también sientan las bases para futuras aplicaciones, como estudios de modelado de dispersión de contaminantes, definición de zonas de muestreo estratégicas (U.S. EPA, 2017). o validación de simulaciones con herramientas tipo AERMOD o CALPUFF (U.S. EPA, 2017; Scire et al., 2000)

Cabe destacar que, aunque el presente análisis se realizó con datos de un solo punto de monitoreo, la herramienta desarrollada es completamente escalable, permitiendo su adaptación a múltiples puntos de recolección, distintos periodos de tiempo, e incluso con ajustes en el código, se puede hacer conexión con plataformas de sensores en tiempo real.

CONCLUSIONES

La implementación de una herramienta automatizada para la generación de rosas de los vientos, desarrollada en Python e integrada con Microsoft Excel, demostró ser una solución eficaz, accesible y replicable para el análisis meteorológico en entornos industriales. El sistema permitió construir reportes técnicos con estructura profesional, integrando visualizaciones, tablas de frecuencia, encabezados institucionales y formato personalizado, sin requerir intervención manual significativa.

El análisis de los datos recolectados en la zona industrial norte de Guayaquil evidenció una predominancia de viento desde el Oeste y el Noreste, con velocidades mayoritariamente inferiores a 1 m/s. Este comportamiento sugiere condiciones de baja capacidad de dispersión atmosférica, lo cual representa un factor crítico a considerar en la evaluación de la calidad del aire en zonas con actividad emisora constante.

Desde el punto de vista operativo, la metodología propuesta fortalece la trazabilidad de los análisis, minimiza el error humano y reduce el tiempo requerido para la elaboración de reportes. Su diseño centrado en Excel como interfaz facilita su adopción por usuarios sin experiencia en programación, ampliando su aplicabilidad en contextos reales de monitoreo técnico o académico.

Además de su utilidad para estudios exploratorios, la herramienta constituye una base sólida para alimentar modelos de simulación ambiental como AERMOD o CALPUFF, así como para validar campañas de campo en evaluaciones de impacto ambiental. Su arquitectura escalable permite adaptarla fácilmente a múltiples puntos de monitoreo, periodos prolongados o integración con sensores en tiempo real.

Cabe destacar que, con mejoras progresivas al código y ajustes en la interfaz, esta

solución puede evolucionar hacia una plataforma completamente operativa desde Excel, donde los usuarios —sin necesidad de conocimientos en Python— puedan importar sus datos, generar rosas de los vientos y exportar reportes técnicos de forma automática. Esta adaptabilidad la convierte en una herramienta de gran valor para técnicos ambientales, estudiantes y profesionales que requieren soluciones ágiles, reproducibles y visualmente claras para el análisis meteorológico aplicado.

En definitiva, esta automatización representa apenas una muestra del potencial que ofrece la integración de Python con herramientas de uso cotidiano. Existe un mundo de posibilidades para escalar soluciones similares hacia otras áreas técnicas —como salud, educación, industria o gestión ambiental—, democratizando el acceso a tecnologías avanzadas. Python, al combinarse con plataformas familiares como Excel, permite transformar procesos rutinarios en sistemas inteligentes, eficientes y al alcance de todos.

REFERENCIAS

Agencia de Protección Ambiental de los Estados Unidos. (1995). Compilation of air pollutant emission factors AP-42. Washington D. C.

Agencia de Regulación y Control de Electricidad (ARCONEL). (2018). Estadística anual y multianual del sector eléctrico ecuatoriano. Quito, Ecuador.

Ministerio del Ambiente del Ecuador. (2015). Anexo 4: Norma de calidad del aire ambiente o nivel de inmisión. Libro VI del Texto Unificado de Legislación Secundaria del Ministerio del Ambiente.

Ministerio del Ambiente, Agua y Transición Ecológica. (2020, agosto 14). La calidad del aire de las ciudades una prioridad nacional. Gobierno del Ecuador. https://www.ambiente.gob.ec/la-calidad-del-aire-de-las-ciudades-una-prioridad-nacional/

Organización Mundial de la Salud (OMS). (2018, mayo 2). Ambient (outdoor) air quality and health. https://www.who.int/es/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health

Python Software Foundation. (2023). The Python tutorial.https://docs.python.org/es/3/tutorial/

U.S. Environmental Protection Agency. (2004). AERMOD: Description of model formulation. Office of Air Quality Planning and Standards. https://www.epa.gov/scram/air-quality-dispersion-modeling-preferred-and-recommended-models#aermod

APÉNDICES Y MATERIALES COMPLEMENTARIOS

Apéndice A: Código VBA Excel

```
Sub GenerarRosa()
    Dim rutaCSV As String Dim
    carpeta  As  String  Dim
    rutaPython As String Dim
    rutaScript As String Dim
    cmd    As String
    Dim shell   As Object
    Dim wb         As Workbook
    Dim wsForm     As Worksheet
    Dim wsDatos    As Worksheet
    Dim rutaExcel As String

    Set wb = ThisWorkbook
    Set wsForm = wb.Sheets("Formulario")
    Set wsDatos = wb.Sheets("Datos")

    ' Guardar antes de continuar
    ThisWorkbook.Save
    ' Obtener ruta del archivo CSV desde D9
    rutaCSV = wsForm.Range("D9").Value
    If Len(Dir(rutaCSV, vbNormal)) = 0 Then
        MsgBox "? La celda D9 no contiene una ruta válida al archivo
CSV.", vbExclamation
        Exit Sub
    End If

    ' Limpiar hoja Datos
    wsDatos.Cells.Clear

    ' Importar el CSV a hoja Datos
    With wsDatos.QueryTables.Add(Connection:="TEXT;" & rutaCSV,
Destination:=wsDatos.Range("A1"))
        .TextFileParseType = xlDelimited
        .TextFileCommaDelimiter = True
        .TextFilePlatform = xlWindows
        .TextFileTrailingMinusNumbers = True
        .Refresh BackgroundQuery:=False
    End With

    ' Obtener ruta completa del Excel que está abierto
    rutaExcel = wb.FullName

    ' Configurar Python
    carpeta = wb.Path
    rutaPython = "python" ' Usa la ruta completa si es necesario
    rutaScript = carpeta & "\genera_reporte.py"

    ' Construir comando para ejecutar Python con dos argumentos: CSV y
Excel
    cmd = "cmd.exe /k " & rutaPython & " "" & rutaScript & "" "" &
rutaCSV & "" "" & rutaExcel & "" & pause"
```

```
    ' Ejecutar script
    Set shell = CreateObject("WScript.Shell")
    shell.Run cmd, 1, True

    MsgBox "? Python terminó. Se ha generado un nuevo archivo con el
reporte.", vbInformation
End Sub
```

Nota técnica: Para ejecutar el script Python correctamente, es necesario tener instalado **Python 3.11 o superior** y contar con las siguientes bibliotecas: pandas, matplotlib, windrose, openpyxl,.

Estas pueden instalarse con el siguiente comando en la terminal o consola:

-pip install pandas matplotlib windrose openpyxl

Apéndice B. Script Python para la generación del reporte de rosa de los vientos

```
import sys

import os

import pandas as pd

import matplotlib.pyplot as plt

from windrose import WindroseAxes

from openpyxl import Workbook, load_workbook

from openpyxl.drawing.image import Image as ExcelImage

from openpyxl.styles import Font, Alignment, Border, Side


def generar_rosa_vientos(direccion, velocidad,
nombre_imagen='rosa_vientos.png'):

    ax = WindroseAxes.from_ax()

    ax.bar(direccion, velocidad, normed=True, opening=0.8, edgecolor='white')

    ax.set_legend()

    fig = plt.gcf()

    fig.set_size_inches(5, 5)

    plt.savefig(nombre_imagen, dpi=150, bbox_inches='tight')

    plt.close()


def crear_reporte_excel(nombre_archivo, imagen_path, datos_df,
datos_formulario, logo_path=None):

    wb = Workbook()

    ws = wb.active

    ws.title = "REPORTE"

    for col in range(ord('A'), ord('K')):

        ws.column_dimensions[chr(col)].width = 10

    ws["E2"] = "Rosa de los Vientos"

    ws["E2"].font = Font(size=12, bold=True)

    etiquetas = ["Empresa", "Dirección", "Área Analizada", "Solicitado por"]

    for i, etiqueta in enumerate(etiquetas):

        fila = 4 + i

        ws[f"E{fila}"] = etiqueta

        ws[f"G{fila}"] = datos_formulario.get(etiqueta.lower(), "")

        ws[f"E{fila}"].font = Font(bold=True)

    if logo_path and os.path.exists(logo_path):

        try:
```

```
            logo = ExcelImage(logo_path)
            logo.anchor = 'A1'
            ws.add_image(logo)
        except:
            pass
    if os.path.exists(imagen_path):
        img = ExcelImage(imagen_path)
        img.width = 350
        img.height = 350
        img.anchor = 'C9'
        ws.add_image(img)
    bins = [0, 1, 2, 3, 4, 5, 10, 100]
    labels_rango = [f"{bins[i]}-{bins[i+1]}" for i in range(len(bins) - 1)]
    datos_df['rango'] = pd.cut(datos_df['ws (m/s)'], bins=bins,
labels=labels_rango, right=False)
    labels_dir = ["N", "NE", "E", "SE", "S", "SW", "W", "NW", "N2"]
    datos_df['direccion_label'] =
        pd.cut( datos_df['wd (deg)'],
        bins=[-0.1, 22.5, 67.5, 112.5, 157.5, 202.5, 247.5, 292.5, 337.5,
360],
        labels=labels_dir,
        ordered=False
    ).astype(str).replace("N2", "N")
    tabla = (
        datos_df
        .groupby(['direccion_label', 'rango'], observed=False)
        .size()
        .reset_index(name='frecuencia')
    )
    tabla = tabla[tabla['frecuencia'] > 0]
    tabla = tabla.sort_values(by='frecuencia',
ascending=False).reset_index(drop=True)
    tabla = tabla.head(36)
    borde =
        Border( left=Side(style='t
        hin'),
        right=Side(style='thin'),
        top=Side(style='thin'),
        bottom=Side(style='thin')
```

```
    }
    inicio_fila = 28
    columnas_inicio = [1 + 3 * i for i in range((len(tabla) + 10) // 11)]
    for b, i in enumerate(range(0, len(tabla), 11)):
        bloque = tabla.iloc[i:i+11]
        col = columnas_inicio[b]
        headers = ["direccion", "rango", "frecuencia"]
        for j, encabezado in enumerate(headers):
            ws.cell(row=inicio_fila, column=col + j, value=encabezado).font =
Font(bold=True)
        for fila_offset, fila in enumerate(bloque.itertuples(index=False),
start=1):
            ws.cell(row=inicio_fila + fila_offset, column=col,
value=fila.direccion_label)
            ws.cell(row=inicio_fila + fila_offset, column=col + 1,
value=fila.rango)
            ws.cell(row=inicio_fila + fila_offset, column=col + 2,
value=int(fila.frecuencia))
    # PIE DE PÁGINA
    fila_pie = inicio_fila + 13
    contenido = [
        ["Ing. XXXXXXX", "DIRECCION, Teléf.: 593-0XXXXXXXXXXXXXX", "Anexo 2"],
        ["Técnico Responsable", "INFORMACION DEL LABORATORIO", "1 de 1"]
    ]
    bloques = [(1, 2), (3, 8), (9, 10)]
    for i in range(2):
        row = fila_pie + i
        for c_ini, c_fin in bloques:
            ws.merge_cells(start_row=row, start_column=c_ini, end_row=row,
end_column=c_fin)
    wb.save(nombre_archivo)


def main():
    if len(sys.argv) < 3:
        print("+Uso: genera_reporte.py ruta_csv archivo_excel_origen")
        return
    ruta_csv = sys.argv[1]
    archivo_excel = sys.argv[2]
    if not os.path.isfile(ruta_csv) or not os.path.isfile(archivo_excel):
```

```
                    p
                    r
                    i
                    n
                    t
                    (
                    "
                    +
                    R
"",                 e
                    v
                    i
                    s
                    a
                    l
                    a
                    s
                    r
                    u
                    t
                    a
                    s
                    "
                    )
                    r
                    e
                    t
                    u
                    r
                    n
                try:
                    wb =
                    load_workboo
                    k(archivo_ex
                    cel,
                    data_only=Tr
                    ue) ws =
                    wb["Formular
                    io"]
```

```
    datos_formulario = {                                    str(ws["D6"].value).strip() if
                                                            ws["D6"].value else
        "empresa":
        str(ws["D5"].value).strip() if
        ws["D5"].value else "",                             "área analizada":
                                                            str(ws["D7"].value).strip() if
        "dirección":                                        ws["D7"].value
else "",

                                                "solicitado por":
else ""                                         str(ws["D8"].value).strip() if
                                                ws["D8"].value
                                            }


        nombre_empresa = datos_formulario["empresa"] or "Empresa"

        wb.close()

        df = pd.read_csv(ruta_csv, sep=",", engine="python", skiprows=5)

        df.columns = df.columns.str.strip().str.lower()

        if 'ws (m/s)' not in df.columns or 'wd (deg)' not in df.columns:

            print("+Columnas de viento no válidas.")

            return

        generar_rosa_vientos(df['wd (deg)'], df['ws (m/s)'])

        carpeta = os.path.dirname(archivo_excel)

        ruta_logo = os.path.join(carpeta, "logo.png")

        nombre_reporte = os.path.join(carpeta,
f"REPORTE_{nombre_empresa}.xlsx")

        crear_reporte_excel(nombre_reporte, "rosa_vientos.png", df,
datos_formulario, ruta_logo)

        print(f"   Reporte generado: {nombre_reporte}")

    except Exception as e:

        print("+Error:", e)


if _name_== "_main_": main()
```