

# PRUEBA TÉCNICA INGENIERO DE DATOS PYTHON

---

KPMG

JUAN SEBASTIÁN PEÑALOZA Q.  
@JSebastianDS



# 1. Conceptos teóricos

<b>Defina el concepto de escalabilidad en el contexto de Azure y detalle los métodos implementados para lograrla eficazmente</b>	Es la capacidad de un sistema o infraestructura para manejar los cambios en la carga de trabajo, sin comprometer el rendimiento o la eficiencia. La escalabilidad es fundamental para adaptarse a las variaciones en la demanda de los usuarios y maximizar los recursos sin desperdicio. Azure permite escalabilidad vertical (scale-up), escalabilidad horizontal (scale-out) y Autoscaling. Las operaciones de escalado pueden clasificarse como <b>estáticas</b> (ajustes programados diariamente para manejar patrones de carga típicos), <b>automáticas</b> (un proceso que se activa cuando se cumplen ciertas condiciones) o <b>manuales</b> (una acción de escalado puntual realizada por un operador en respuesta a un aumento de carga inesperado). Tanto el escalado vertical como el horizontal pueden lograrse utilizando cualquiera de estos métodos. Sin embargo, implementar el escalado vertical automático requiere consideraciones adicionales.
<b>¿Podrías proporcionar definiciones y ejemplos de datos estructurados, semiestructurados y no estructurados?</b>	Los datos estructurados están organizados en un formato fijo, como las <b>BD relacionales</b> . los datos semiestructurados no tienen una estructura tan rígida, pero cuentan con cierta organización, usualmente en formato de etiquetas o key-value, como <b>JSON</b> o <b>XML</b> files. Los datos no estructurados, como <b>textos</b> , <b>imágenes</b> , <b>videos</b> o audios, carecen de un formato o esquema predefinido
<b>¿Cuál es el propósito o caso de uso específico para el que están diseñados los servicios Azure Blob Storage y Azure Data Lake Storage Gen2?</b>	Azure Data Lake Storage Gen2 es un conjunto que agrega funcionalidades adicionales al servicio de Azure Blob storage. El objetivo es permitir la escalabilidad y la seguridad a nivel de archivo en un lago de datos; sin que se vea afectada la disponibilidad. El caso de uso es para una necesidad de un almacén masivo (petabytes) de datos semi-estructurados o no estructurados (llamados <b>blobs</b> en el contexto de Azure) pero que posea un tipo de jerarquía interna en el sistema de archivos (para alta disponibilidad y simplificada administración del Lago de Datos)

<p>Para un proyecto de desarrollo en Python, se necesita ejecutarlo en una VM que ya tiene otros desarrollos en funcionamiento. ¿Cuál sería la mejor manera de abordar esta situación? ¿Qué desafíos potenciales podrían surgir y cómo se podrían resolver?</p>	<p>Previo a añadir el nuevo desarrollo a la misma VM, es recomendable mover las aplicaciones a <b>Azure App Service</b>, donde se puede alojar código Python en entornos separados sin gestionar directamente la infraestructura. Esto reduce la complejidad de la administración y permite la escalabilidad. En caso de que se deba alojar todos los desarrollos se pueden presentar desafíos de <b>conflictos de dependencias, Gestión de seguridad y acceso</b> a las aplicaciones, <b>monitoreo</b>; además de testing y rendimiento de la VM</p>
<p>¿Cuál es la diferencia entre una base de datos relacional y una base de datos no relacional?</p>	<p>Las BD relacionales tienen un esquema estructurado y predefinido (tabular). No Relacional: No usa un esquema de tablas rígido, permitiendo mayor flexibilidad en la organización de los datos. Los datos se almacenan en formatos como documentos (JSON, BSON), key-value, grafos o columnas. Las bases de datos no relacionales pueden manejar grandes volúmenes de información de forma rápida y escalable. Ejemplos incluyen MongoDB (<b>documentos</b>), Redis (<b>key-value</b>) y Neo4j (<b>grafos</b>).</p>
<p>¿Sabe qué es la complejidad algorítmica y cómo se mide?</p>	<p>Sí. Es la cantidad de recursos computacionales necesarios para ejecutar un algoritmo en función del tamaño de su entrada. Existen dos tipos principales de complejidad: temporal (tiempo de ejecución) y espacial (uso de memoria). Para medir la complejidad algorítmica, se utiliza la notación Big O (O), que describe el comportamiento del tiempo o espacio de un algoritmo en función de su entrada. Por ejemplo, <b>O(1)</b> indica tiempo constante, <b>O(n)</b> tiempo lineal, <b>O(n<sup>2</sup>)</b> tiempo cuadrático, y así sucesivamente.</p>
<p>¿Qué significa ETL y cómo se aplica en situaciones reales? Proporcione un ejemplo práctico.</p>	<p>Extract, Transform, Load (Extracción, Transformación, Carga). Es un proceso en la gestión de datos que consiste en extraer datos de diversas fuentes, transformarlos en un formato adecuado y cargarlos en un sistema de almacenamiento o base de datos para su análisis. empresa de retail utiliza ETL para consolidar datos de ventas provenientes de tiendas físicas, su sitio web y una aplicación móvil. Primero, extrae los datos de cada plataforma, los transforma para unificar el formato (por ejemplo, unificando las monedas y eliminando duplicados) y luego los carga en un data warehouse. Este proceso permite a la empresa analizar las ventas en tiempo real y ajustar estrategias de marketing o inventario</p>

<p><b>¿Cuáles son las cuatro etapas comunes en el procesamiento de soluciones de Big Data, aplicables a todas las arquitecturas?</b></p>	<p><b>Ingesta de Datos:</b> Esta etapa implica la recopilación de datos provenientes de diversas fuentes (bases de datos, sensores, aplicaciones, redes sociales, etc.) y su integración en el sistema de procesamiento. Los datos pueden ser ingeridos en tiempo real o en lotes, dependiendo de los requisitos de la aplicación.</p> <p><b>Almacenamiento de Datos:</b> Una vez que los datos han sido capturados, deben almacenarse de manera segura y eficiente.</p> <p><b>Procesamiento y Análisis:</b> En esta fase, los datos son transformados, limpiados y analizados para extraer información útil. Aquí se aplican técnicas de procesamiento en paralelo, utilizando herramientas como Apache Spark o MapReduce.</p> <p><b>Visualización y Consumo de Datos:</b> Finalmente, los resultados procesados son presentados a los usuarios o integrados en sistemas de negocio. Esto puede incluir visualizaciones gráficas, reportes o dashboards, que permiten interpretar los datos y tomar decisiones informadas.</p>
--	---

## 2. Conocimientos basicos SQL

Ingresar al siguiente enlace <https://sqliteonline.com/>

Una vez dentro de la pagina, y utilizando el motor de SQLite, y con la tabla de ejemplo precargada en el sistema, llamada "**demo**", hacer lo siguiente:

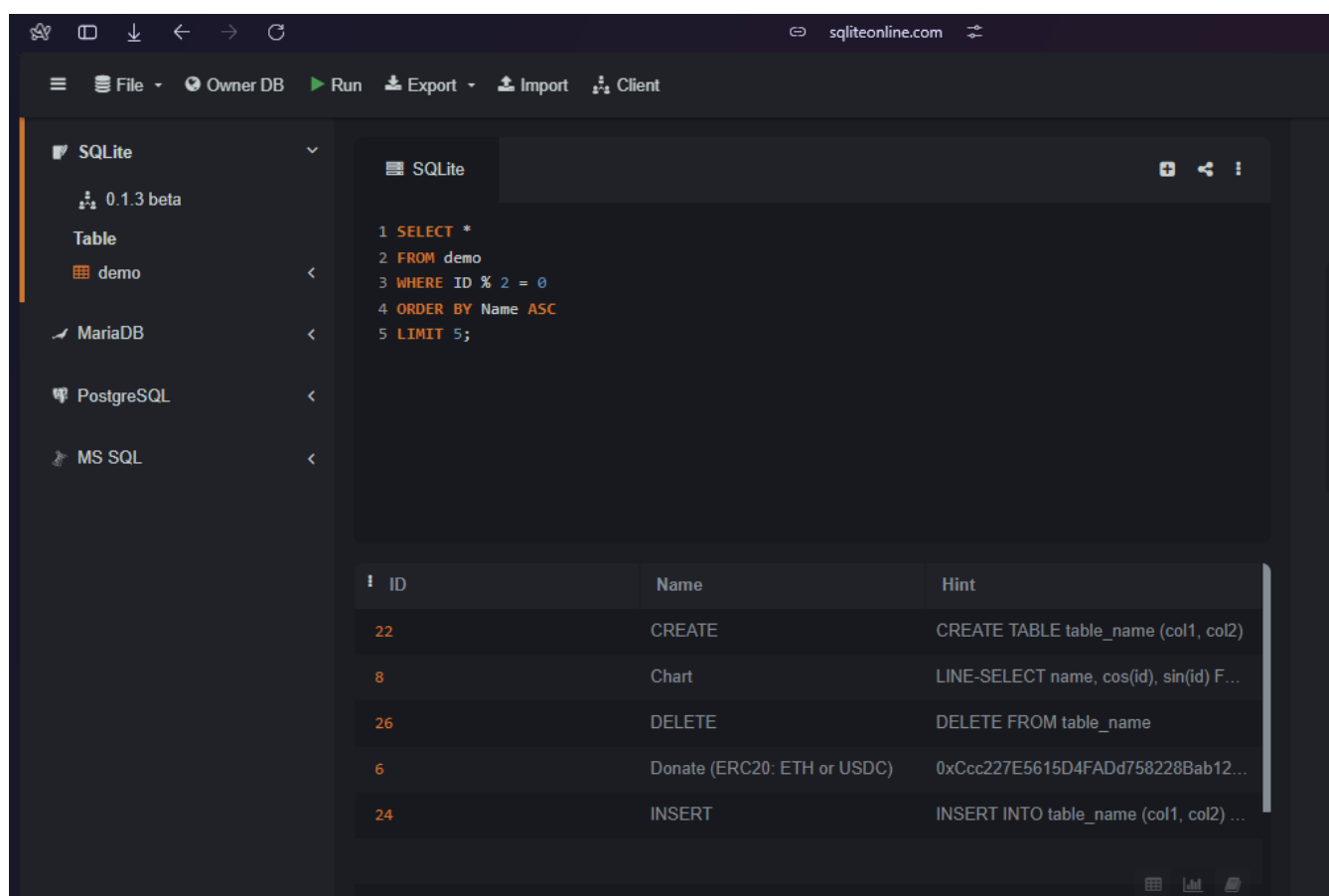
**Recuperar los primeros 5 registros con un "ID" que sea un múltiplo de 2. Mostrar estos registros en orden alfabético ascendente según su "Name".**

**Como evidencia adjunte en el correo el código y/o el pantallazo de la consulta ejecutada y la tabla de salida**

**Tenga presente que la respuesta optima que esperamos ver es la siguiente:**

ID	Name	Hint
8	Chart	LINE-SELECT name, cos(id), sin(id) FROM demo;
6	Donate (ERC20: ETH or USDC)	0xCcc227E5615D4FADd758228Bab12ceb465D4ED18
4	Kirill N.	<a href="https://www.linkedin.com/in/sqliteonlinecom">https://www.linkedin.com/in/sqliteonlinecom</a>
10	SQLite 3.41.0	SQL OnLine on JavaScript
2	<a href="https://SQL.BanD">https://SQL.BanD</a>	The most secure, fast, efficient web-sql client

*Figura 1. Ejemplo de resultado óptimo de la query.*



*Figura 2. Vista de la consulta en motor web.*

### 3. Conocimientos basicos Python

Se le proporcionará un archivo de Excel en el que los nombres de las columnas pueden contener espacios en blanco tanto en medio como al principio o final. Esto puede generar problemas al persistir o procesar la información en una base de datos o mediante SQL. Para solucionar esto, se necesita crear un script en Python que realice lo siguiente:

1. Cargue el archivo en un DataFrame.
2. Realice la limpieza de los nombres de las columnas. Si hay espacios en blanco al inicio o al final, estos deben eliminarse. Si hay espacios en blanco o caracteres especiales en medio de los nombres, deben reemplazarse con el carácter "\_".
3. Exporte el archivo a un formato CSV con los nombres de columnas corregidos.

Esta limpieza asegurará que los datos sean más manejables y coherentes para su persistencia y análisis.

Como evidencia adjunte en el correo el código ejecutado para su solución



**El archivo de origen (xlsx) tiene la siguiente vista previa:**

Archivo Editar Selección Ver Ir ... prueba.kpmg\_tecnica

Punto 3\_ Datos\_muestra\_prueba\_tecnica.xlsx

Punto 3\_ Datos\_muestra\_prueba\_tecnica.xlsx

	A	B	C	D	E	F	G	H	I
3	AS1001	Security Administration & Client Administration - Basis Security	IVONNE JULIETH MONROY SUAREZ	BS05	SCC5	BS10	PF0G		
4	AS1001	Security Administration & Client Administration - Basis Security	IVONNE JULIETH MONROY SUAREZ	BS05	SCC7	BS10	SM19		
5	AS1001	Security Administration & Client Administration - Basis Security	IVONNE JULIETH MONROY SUAREZ	BS05	SCC8	BS10	SU01		
6	AS1001	Security Administration & Client Administration - Basis Security	IVONNE JULIETH MONROY SUAREZ	BS05	SCC9	BS10	SU02		
7	AS1001	Security Administration & Client Administration - Basis Security	IVONNE JULIETH MONROY SUAREZ	BS05	SCC1	BS10	SU10		
8	AS1001	Security Administration & Client Administration - Basis Security	IVONNE JULIETH MONROY SUAREZ	BS05	SCC1	BS10	SU12		
9	AS1001	Security Administration & Client Administration - Basis Security	LEONARDO FABIAN ROJAS PEREZ	BS05	SCC1	BS10	LICENSE_ATTRIBUTES		
10	AS1001	Security Administration & Client Administration - Basis Security	LEONARDO FABIAN ROJAS PEREZ	BS05	SCC4	BS10	PF0G		
11	AS1001	Security Administration & Client Administration - Basis Security	LEONARDO FABIAN ROJAS PEREZ	BS05	SCC5	BS10	SM19		
12	AS1001	Security Administration & Client Administration - Basis Security	LEONARDO FABIAN ROJAS PEREZ	BS05	SCC7	BS10	SU01		
13	AS1001	Security Administration & Client Administration - Basis Security	LEONARDO FABIAN ROJAS PEREZ	BS05	SCC8	BS10	SU02		
14	AS1001	Security Administration & Client Administration - Basis Security	LEONARDO FABIAN ROJAS PEREZ	BS05	SCC9	BS10	SU10		
15	AS1001	Security Administration & Client Administration - Basis Security	LEONARDO FABIAN ROJAS PEREZ	BS05	SCC1	BS10	SU12		
16	AS1001	Security Administration & Client Administration - Basis Security	LEONARDO FABIAN ROJAS PEREZ	BS05	SCOT				
17	PTP2006	EBP / SRM Purchasing & Enter Counts - WM	OSCAR DE JESÚS BARRAZA PEDROZA	MM07	LI11	MM08	LI02	SR02	BAP1 - BD53
18	PTP2006	EBP / SRM Purchasing & Enter Counts - WM	OSCAR DE JESÚS BARRAZA PEDROZA	MM07	LI12	MM08	LI20	SR02	BAP1 - BD53
19	PTP2006	EBP / SRM Purchasing & Enter Counts - WM	OSCAR DE JESÚS BARRAZA PEDROZA	MM07	LI12	MM08	LI21		
20	PTP2006	EBP / SRM Purchasing & Enter Counts - WM	OSCAR DE JESÚS BARRAZA PEDROZA	MM07	LI14	MM08	LI20		
21	PTP2006	EBP / SRM Purchasing & Enter Counts - WM	OSCAR DE JESÚS BARRAZA PEDROZA	MM07	LI14	MM08	LI21		

Hojas: 1

*Figura 3. Estructura tabular inicial de los datos a tratar.*

El código que lleva a cabo la limpieza es incómodo de leer como texto plano, por lo tanto aquí adjunto una imagen del mismo pero dentro de editor VSCode:

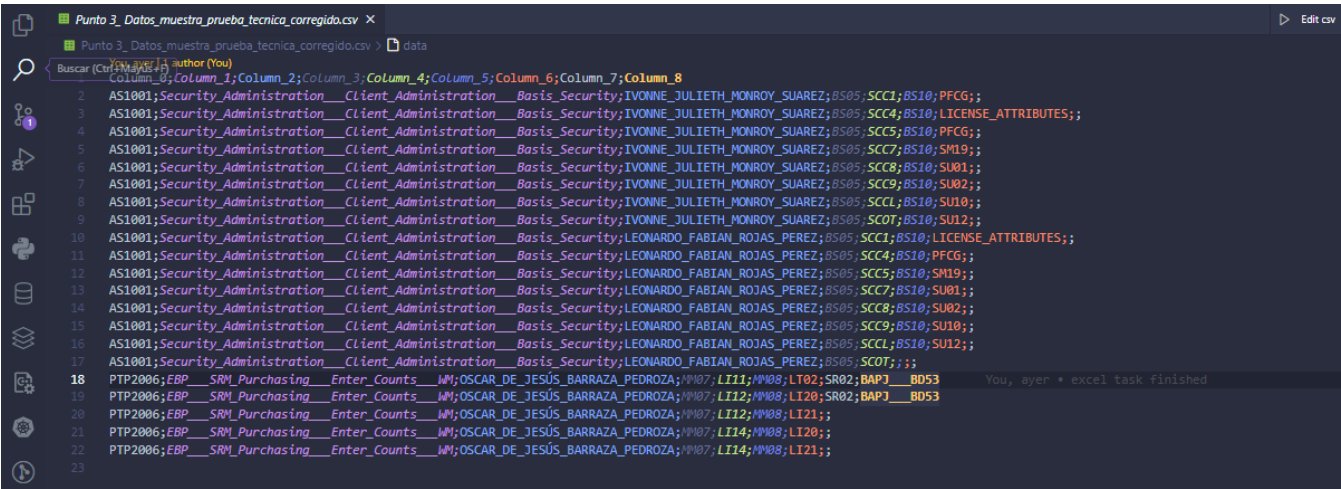
```

6 class ExcelCleaner:
35     def clean_values(self):
36         """Modifica los strings en todas las filas del DataFrame eliminando car
37         def clean_value(value):
38             if isinstance(value, str):
39                 value = value.strip()
40                 value = re.sub(r'(\W{10000000000})', '_', value)
41                 return value.strip()
42             return value
43
44         self.dataframe = self.dataframe.applymap(clean_value)
45         print("Valores de filas modificados exitosamente.")
46
47     def export_to_csv(self):
48         """El resultado de la limpieza se exporta a un archivo CSV
49         de la forma [nombre_archivo].corregido.csv
50         Se guarda en el mismo directorio del archivo Excel"""
51         # Definir la ruta para guardar el resultado
52         output_file = os.path.join(self.directory, f"{self.file_name}.corregido
53
54         # Exportar a CSV con punto y coma como delimitador
55         self.dataframe.to_csv(output_file, sep=";", index=False)
56
57 # Función principal
58 if __name__ == "__main__":
59
60     Excel_file = "Punto 3_Datos_muestra_prueba_tecnica.xlsx"
61
62     # Crear una instancia de la clase ExcelCleaner
63     cleaner = ExcelCleaner(Excel_file)
64
65     cleaner.load_excel()
66     cleaner.clean_column_names()
67     cleaner.clean_values()
68     cleaner.export_to_csv()
69     print("Ejecución finalizada. Revisa el archivo CSV corregido.")
70
71
72

```

Figura 4. Lógica, clases, dunciones e indentación del programa.

Por defecto usé el punto y coma de separador, y cada entrada se trabajó con métodos aplicables a strings. El resultado tiene encabezado para cada columna:



```
Punto 3_Datos_muestra_prueba_tecnica_corregido.csv X
Punto 3_Datos_muestra_prueba_tecnica_corregido.csv > data
Column_1;Column_2;Column_3;Column_4;Column_5;Column_6;Column_7;Column_8
2 AS1001;Security_Administration_Client_Administration_Basis_Security;IVONNE_JULIETH_MONROY_SUAREZ;BS05;SCC1;BS10;PFCG;;
3 AS1001;Security_Administration_Client_Administration_Basis_Security;IVONNE_JULIETH_MONROY_SUAREZ;BS05;SCC4;BS10;LICENSE_ATTRIBUTES;;
4 AS1001;Security_Administration_Client_Administration_Basis_Security;IVONNE_JULIETH_MONROY_SUAREZ;BS05;SCC5;BS10;PFCG;;
5 AS1001;Security_Administration_Client_Administration_Basis_Security;IVONNE_JULIETH_MONROY_SUAREZ;BS05;SCC7;BS10;SM19;;
6 AS1001;Security_Administration_Client_Administration_Basis_Security;IVONNE_JULIETH_MONROY_SUAREZ;BS05;SCC8;BS10;SU01;;
7 AS1001;Security_Administration_Client_Administration_Basis_Security;IVONNE_JULIETH_MONROY_SUAREZ;BS05;SCC9;BS10;SU02;;
8 AS1001;Security_Administration_Client_Administration_Basis_Security;IVONNE_JULIETH_MONROY_SUAREZ;BS05;SCC1;BS10;SU10;;
9 AS1001;Security_Administration_Client_Administration_Basis_Security;IVONNE_JULIETH_MONROY_SUAREZ;BS05;SCOT;BS10;SU12;;
10 AS1001;Security_Administration_Client_Administration_Basis_Security;LEONARDO_FABIAN_ROJAS_PEREZ;BS05;SCC1;BS10;LICENSE_ATTRIBUTES;;
11 AS1001;Security_Administration_Client_Administration_Basis_Security;LEONARDO_FABIAN_ROJAS_PEREZ;BS05;SCC4;BS10;PFCG;;
12 AS1001;Security_Administration_Client_Administration_Basis_Security;LEONARDO_FABIAN_ROJAS_PEREZ;BS05;SCC5;BS10;SM19;;
13 AS1001;Security_Administration_Client_Administration_Basis_Security;LEONARDO_FABIAN_ROJAS_PEREZ;BS05;SCC7;BS10;SU01;;
14 AS1001;Security_Administration_Client_Administration_Basis_Security;LEONARDO_FABIAN_ROJAS_PEREZ;BS05;SCC8;BS10;SU02;;
15 AS1001;Security_Administration_Client_Administration_Basis_Security;LEONARDO_FABIAN_ROJAS_PEREZ;BS05;SCC9;BS10;SU10;;
16 AS1001;Security_Administration_Client_Administration_Basis_Security;LEONARDO_FABIAN_ROJAS_PEREZ;BS05;SCC1;BS10;SU12;;
17 AS1001;Security_Administration_Client_Administration_Basis_Security;LEONARDO_FABIAN_ROJAS_PEREZ;BS05;SCOT;;
18 PTP2006;EBP_SRM_Purchasing_Enter_Counts;WM;OSCAR_DE_JESUS_BARRAZA_PEDROZA;MM07;LT11;MM08;LT02;SR02;BAP1_B053
19 PTP2006;EBP_SRM_Purchasing_Enter_Counts;WM;OSCAR_DE_JESUS_BARRAZA_PEDROZA;MM07;LT12;MM08;LT20;SR02;BAP1_B053
20 PTP2006;EBP_SRM_Purchasing_Enter_Counts;WM;OSCAR_DE_JESUS_BARRAZA_PEDROZA;MM07;LT12;MM08;LT21;;
21 PTP2006;EBP_SRM_Purchasing_Enter_Counts;WM;OSCAR_DE_JESUS_BARRAZA_PEDROZA;MM07;LT14;MM08;LT20;;
22 PTP2006;EBP_SRM_Purchasing_Enter_Counts;WM;OSCAR_DE_JESUS_BARRAZA_PEDROZA;MM07;LT14;MM08;LT21;;
23
```

Figura 5. Fichero resultante visto desde el editor de código

Decidí mantener las vocales con acentos en lugar de quitarle la tilde porque consideré que en el contexto del idioma español es posible trabajar con formato UNICODE ‘utf-8’; sin embargo, en un contexto de lengua inglesa sería optimo buscar reemplazar las vocales acentuadas por la misma vocal sin tilde.

Como Anexo encontrará el código como texto en lugar de imágenes. Lo más estético sería usar una funcionalidad como ‘verbatim’ de LaTeX, pero en microsoft Word no hay algo equivalente.

---

## Anexos

```
>>>

import os
import re
import pandas as pd

class ExcelCleaner:
    SPECIAL_CHARS_REGEX = r"^[A-Za-z0-9_]"

    def __init__(self, file_path):
        """
        Inicializa la clase con la ruta del archivo Excel.
        Verifica si el archivo existe y extrae el nombre y el directorio.
        """
        if not os.path.isfile(file_path):
            raise FileNotFoundError(f"El archivo {file_path} no se encontró.
Verifica la ruta y el nombre.")

        self.file_path = file_path
        self.directory = os.path.dirname(file_path)
        self.file_name = os.path.splitext(os.path.basename(file_path))[0]
        self.dataframe = None

    def load_excel(self):
        """Carga el archivo de Excel en un DataFrame"""
        self.dataframe = pd.read_excel(self.file_path, header=None)
        self.dataframe.columns = [f"Column_{i}" for i in
range(self.dataframe.shape[1])]
        print("Archivo de Excel cargado correctamente.")

    def clean_column_names(self):
        """Limpia los nombres de las columnas del DataFrame eliminando
caracteres especiales y espacios."""
        self.dataframe.columns = self.dataframe.columns.str.strip()
        self.dataframe.columns = self.dataframe.columns.str.replace(r"\s+",
"_", regex=True)
        self.dataframe.columns =
self.dataframe.columns.str.replace(self.SPECIAL_CHARS_REGEX, "", regex=True)
        print("Nombres de columnas modificados correctamente.")
```



---

```

def clean_values(self):
    """Modifica los strings en todas las filas del DataFrame eliminando
    caracteres especiales y espacios."""
    def clean_value(value):
        if isinstance(value, str):
            value = value.strip()
            value = re.sub(r'^\wÁÉÍÓÚáéíóúñÑ', '_', value)
            return value.strip()
        return value

    self.dataframe = self.dataframe.applymap(clean_value)
    print("Valores de filas modificados exitosamente.")

def export_to_csv(self):
    """El resultado de la limpieza se exporta a un archivo CSV
    de la forma {nombre_archivo}_corregido.csv
    Se guarda en el mismo directorio del archivo Excel"""
    # Definir la ruta para guardar el resultado
    output_file = os.path.join(self.directory,
    f"{self.file_name}_corregido.csv")

    # Exportar a CSV con punto y coma como delimitador
    self.dataframe.to_csv(output_file, sep=";", index=False)

# Función principal
if __name__ == "__main__":

    Excel_file = "Punto 3_ Datos_muestra_prueba_tecnica.xlsx"

    # Crear una instancia de la clase ExcelCleaner
    cleaner = ExcelCleaner(Excel_file)

    cleaner.load_excel()
    cleaner.clean_column_names()
    cleaner.clean_values()
    cleaner.export_to_csv()
    print("Ejecución finalizada. Revisa el archivo CSV corregido.")
<<<

```