

## PIA – ETAPA 1

### Equipo:

- Jaime Sebastián González Ontiveros
- Mauro Jesús López Cerda
- Raziell Azarias Torres Garza

### Comparación de APIs exploradas:

Nombre de la API	Link	Descripción	Posible Enfoque	Posibles Datos a Obtener
<b>Directorio Estadístico Nacional de Unidades Económicas (DENUE) - INEGI</b>	<a href="https://www.inegi.org.mx/servicios/api_indicadores.html">https://www.inegi.org.mx/servicios/api_indicadores.html</a>	Proporciona indicadores estadísticos de México (económicos, demográficos, etc.) del INEGI.	Gráficas comparativas de estadísticas mexicanas con filtros por entidad.	Obtener indicadores como PIB, población, desempleo por estado o municipio.
<b>REST Countries</b>	<a href="https://restcountries.com">https://restcountries.com</a>	Información detallada de todos los países del mundo en formato JSON.	Gráfica comparativa de población, extensión del territorio, etc.	Consultar datos como capital, idioma, moneda, población, región, etc.
<b>NASA API</b>	<a href="https://api.nasa.gov">https://api.nasa.gov</a>	Acceso a datos y servicios públicos de la NASA (imágenes, meteorología espacial, asteroides, etc.).	Comparación de datos de misiones.	Mostrar imágenes del día, datos de misiones, meteoritos, eventos espaciales.

### Planteamiento del problema:

- El objetivo de este proyecto es analizar las empresas de “camiones” (transporte de carga, lavado, mantenimiento, etc.), utilizando datos del Directorio Estadístico Nacional de Unidades Económicas (DENUE) del INEGI. La finalidad es identificar zonas con mayor o menor concentración de estas empresas, entender su dispersión geográfica y visualizar patrones que podrían ser relevantes para estudios económicos, logísticos o de infraestructura.

- **¿Qué fenómeno o situación quieren analizar?**  
La distribución geográfica y características de las de “camiones” en distintas regiones de México. El enfoque está en identificar patrones de localización y estructura de estas empresas.
- **¿Por qué es relevante?**  
Este análisis es relevante porque el transporte de mercancías por camión representa una parte fundamental de la economía mexicana. Conocer dónde están ubicadas estas empresas puede ayudar a entender la logística, identificar posibles zonas de saturación o baja cobertura, y apoyar decisiones de inversión o desarrollo de infraestructura.
- **¿Qué datos necesitan obtener?**  
Se requiere información de la API del INEGI sobre empresas relacionadas con “camiones”, incluyendo nombre, actividad económica, ubicación geográfica (latitud y longitud), razón social y dirección. Esta información se obtiene consultando la API con diferentes coordenadas geográficas para cubrir diversas zonas del país.

#### Descripción de la API elegida

- La API utilizada es la **API del DENUE (INEGI)**, la cual permite consultar unidades económicas por nombre, actividad o ubicación geográfica.
- Se usa el endpoint:  
<https://www.inegi.org.mx/app/api/denue/v1/consulta/Buscar/{palabraClave}/{latitud},{longitud}/{radio}/{token}>
- **Parámetros principales:**
  - Palabra clave: término de búsqueda (en este caso, "camiones").
  - Latitud, longitud: coordenadas del punto de búsqueda.
  - Radio: distancia en metros (hasta 2500).
  - Token: clave personal para autenticación.
- Se eligieron varias coordenadas de distintas ciudades para ampliar la cobertura del análisis y obtener más diversidad de empresas.

#### Descripción de la estructura de datos utilizada

- Los datos extraídos se almacenan en una estructura compuesta por:
  - Una **lista de diccionarios** (transportes), donde cada diccionario representa una empresa con los campos:
    - 'nombre'
    - 'actividad'
    - 'latitud'
    - 'longitud'
    - 'razon\_social'
    - 'direccion'

- Esta estructura permite almacenar múltiples empresas de forma organizada y es compatible con la exportación a formatos como JSON y Excel, así como con bibliotecas de análisis como pandas.

#### Justificación del tratamiento de datos aplicado

- Se realiza **limpieza básica**, eliminando entradas sin nombre o con campos vacíos.
- Se aplican **expresiones regulares** para validar texto y evitar registros corruptos.
- Se estructura la información con nombres consistentes.
- Se calcula estadística descriptiva sobre las coordenadas geográficas.
- Se exportan los datos en formato “.json” y “.xlsx” para uso posterior.

Este tratamiento es fundamental para evitar errores en el análisis estadístico y para asegurar la integridad de los datos exportados.

#### Minutas de trabajo:

Fecha	Actividad realizada	Responsables	Observaciones
28-abr-2025	Revisión de documentación de la API del INEGI	Jaime Gonzalez, Mauro López, Raziél Torres	Se generó token de acceso
29-abr-2025	Pruebas iniciales con solicitudes a una coordenada	Jaime Gonzalez, Mauro López, Raziél Torres	Se confirmó funcionamiento
30-abr-2025	Implementación de múltiples coordenadas y limpieza	Jaime Gonzalez, Mauro López, Raziél Torres	Se mejoró cobertura de datos
01-may-2025	Análisis estadístico y visualización	Jaime Gonzalez, Mauro López, Raziél Torres	Se implementaron gráficos
02-may-2025	Exportación a Excel y validación final	Jaime Gonzalez, Mauro López, Raziél Torres	Documento final
10-may-2025	Grabación de video explicativo	Jaime Gonzalez, Mauro López, Raziél Torres	Documento final

#### Algoritmo:

Algoritmo: ObtenerDatosDesdeAPI

INICIO

1. Declarar las variables:

LATITUD, LONGITUD, TOKEN, RADIO, INTENTOS\_MAXIMOS, ESPERA\_SEGUNDOS  
URL, INTENTO, RESPUESTA, DATOS

2. Asignar el valor:

TOKEN ← 'ed827f5b-3bcc-4786-9289-311d219ea21e'

3. Asignar los valores de entrada (pueden ser leídos del usuario o fijos):

LATITUD ← <valor>

LONGITUD ← <valor>  
RADIO ← 250  
INTENTOS\_MAXIMOS ← 5  
ESPERA\_SEGUNDOS ← 2

4. Construir la URL:

URL ← 'https://www.inegi.org.mx/app/api/denue/v1/consulta/Buscar/camiones/' +  
LATITUD + ',' + LONGITUD + '/' + RADIO + '/' + TOKEN`

5. Para INTENTO desde 0 hasta INTENTOS\_MAXIMOS - 1 hacer:

6. Intentar:

7. Se manda llamar el módulo REQUESTS con la función GET con la URL → almacenar en  
RESPUESTA

8. Si RESPUESTA.status\_code = 200 entonces

9. Se manda llamar el módulo REQUESTS con la función JSON para extraer datos →  
almacenar en DATOS

10. Si DATOS está vacío o no es lista entonces

11. Escribir: "Error al obtener datos desde la API para la ubicación (LATITUD,  
LONGITUD)"

12. Retornar lista vacía

13. Retornar DATOS

Si no

14. Escribir: "Error: API no responde correctamente (Status code  
RESPUESTA.status\_code)"

15. Retornar lista vacía

Capturar excepción del tipo `RequestException` entonces

16. Escribir: "Error al realizar la solicitud: <mensaje de error>"

17. Si `INTENTO < INTENTOS\_MAXIMOS - 1` entonces

18. Escribir: "Reintentando... (Intento INTENTO + 1 de INTENTOS\_MAXIMOS)"

19. Se manda llamar el módulo TIME con la función SLEEP por ESPERA\_SEGUNDOS  
segundos

Si no

20. Escribir: "Se alcanzó el número máximo de reintentos."

21. Retornar lista vacía

22. Fin del ciclo

23. Fin

Algoritmo: ProcesarDatos

Inicio

1. Se mandan llamar los módulos:

JSON, STATISTICS, PANDAS, MATPLOTLIB.PYPILOT, OS

2. Se manda llamar el módulo `PIA\_Módulo` con las funciones `obtener\_datos\_api` y `TOKEN`

3. Declarar la lista `COORDENADAS` con los siguientes pares (latitud, longitud):

- (21.85717833, -102.28487238)

- (19.432608, -99.133209)

- (25.686614, -100.316113)

4. Declarar la lista vacía `TRANSPORTES`

5. Asignar TOKEN a la variable `token`

6. Para cada par LAT, LON en la lista `COORDENADAS` hacer:

7. Se manda llamar el módulo PIA\_Módulo\_Script con la función `obtener\_datos\_api(LAT, LON, token)` → almacenar en `DATOS`

8. Para cada `ESTABLECIMIENTO` en `DATOS` hacer:

9. Intentar:

10. Convertir `ESTABLECIMIENTO['Latitud']` y `ESTABLECIMIENTO['Longitud']` a tipo real → guardar en `LAT`, `LON`

11. Agregar un registro al final de la lista `TRANSPORTES` con los campos:

- `nombre`

- `actividad`

- `latitud`

- `longitud`

- `razon\_social`

- `direccion`

12. Capturar excepciones del tipo `ValueError`, `KeyError`, `TypeError`:

13. Continuar con el siguiente registro

14. Abrir el archivo 'transportes.json' para escritura en formato UTF-8

15. Se manda llamar el módulo JSON con la función `dump` para escribir los datos en el archivo
16. Filtrar los registros de TRANSPORTES donde el campo 'nombre' no esté vacío o con solo espacios → guardar en TRANSPORTES\_LIMPIOS
17. Crear listas LATITUDES y LONGITUDES con los valores respectivos de TRANSPORTES\_LIMPIOS
18. Definir función `CALCULAR\_VARIANZA(DATOS)`:
  - Si la longitud de DATOS > 1:
    - Retornar varianza de DATOS con la función `statistics.variance`
  - Si no:
    - Retornar NULO
19. Definir función CALCULAR\_MODA(DATOS):
  - Intentar retornar `statistics.mode(DATOS)`
  - Si lanza excepción `StatisticsError`, retornar NULO
20. Calcular e imprimir estadísticas de LATITUDES:
  - Media
  - Mediana
  - Moda (si existe)
  - Varianza (si hay suficientes datos)
  - Desviación estándar (si hay suficientes datos)
21. Calcular e imprimir estadísticas de LONGITUDES:
  - Media
  - Mediana
  - Moda
  - Varianza
  - Desviación estándar
22. Generar gráfico de histograma de LATITUDES:
  - Se manda llamar el módulo MATPLOTLIB.PYPILOT con la función `hist`
  - Título: 'Distribución Geográfica de Empresas de Transporte (Latitud)'
  - Guardar como 'grafico\_latitudes.png'
23. Generar gráfico de líneas con LATITUDES y LONGITUDES
24. Crear un DataFrame `DF` con TRANSPORTES\_LIMPIOS usando el módulo PANDAS
25. Intentar exportar `DF` a archivo Excel 'empresas\_transporte.xlsx':
  - Se manda llamar `to\_excel` con `engine='openpyxl'`
26. Capturar excepción e imprimir mensaje de error si ocurre
27. Fin

## **Diagrama de Flujo: (anexo al final)**

### **Guión:**

#### **Jaime:**

Buenos días, este es nuestro proyecto final. Somos tres integrantes en el equipo: Jaime Gonzalez, Mauro López y Raziell Torres. El proyecto consta de un código realizado en Python, donde se conecta a una API para realizar una investigación.

En este caso, una vez analizadas varias opciones, se eligió una API pública del Directorio Estadístico Nacional de Unidades Económicas (DENUE), en conjunto con la INEGI. El objetivo fue analizar empresas relacionadas con “camiones”, como transporte de carga, mantenimiento o lavado, para ver dónde están ubicadas en México y si hay zonas con más o menos concentración. Esto puede resultar importante para analizar u optimizar rutas de transporte de mercancía, y así apoyar la logística de una empresa o negocio.

Para empezar, usamos una API llamada DENUE, del INEGI. Esta API nos da acceso a información de unidades económicas de todo el país. En este caso, como ya mencionamos, nos centramos en empresas relacionadas a camiones y transportes de carga. Creamos una función llamada “obtener\_datos\_api”, que básicamente hace una consulta a esta API usando una palabra clave (en nuestro caso, “camiones”) y unas coordenadas geográficas (obtenidas a partir de búsqueda de información en Google Maps).

#### **Mauro:**

A parte de ello, agregamos una pequeña sección enfocada en el manejo de errores para que, si la API falla o no responde, el programa vuelva a intentarlo varias veces; decidimos implementar esto después de unas cuantas sesiones de prueba y error, pues nos percatamos que en ocasiones la conexión con la API no era la más estable o solía tardar. Esto lo hicimos con un ciclo for y el uso de try y except, con el fin de optimizar un poco el proceso. Ya para finalizar con esta parte, requerimos hacer uso de un token, que es como una clave especial que nos permitió el acceso a la API.

Una vez que ya teníamos la función, pasamos a recolectar algunos datos. Elegimos tres ciudades: Aguascalientes, Ciudad de México y Monterrey. Para cada ciudad, usamos sus coordenadas y consultamos la API. La información que nos dio la API la guardamos en una lista de diccionarios, donde cada diccionario representa una empresa. Ahí guardamos el nombre, actividad, latitud, longitud, razón social y dirección. Esto nos ayuda a organizar los datos de forma clara y reutilizable. Después hicimos una limpieza rápida, pues eliminamos empresas que no tenían nombre, y verificamos que los datos fueran válidos.

#### **Raziell:**

Ya con los datos limpios, pasamos al análisis y usamos la librería statistics para calcular cosas como:

- La media, o promedio de latitudes y longitudes.
- La mediana, o sea, el punto medio de los datos.

- La moda, si existía algún valor que se repitiera.
- Y también varianza y desviación estándar, que nos dicen qué tan dispersos están los datos.

Después hicimos unas gráficas con matplotlib. Primero, una que muestra cómo se distribuyen las empresas en cuanto a latitud, es decir, si están más al norte o más al sur, y luego una gráfica de líneas, donde comparamos latitudes y longitudes, para ver cómo cambian las ubicaciones de una empresa a otra. Estas gráficas nos ayudaron a visualizar los patrones geográficos.

Finalmente, exportamos todos los datos:

- Primero en formato .json, que es muy común para guardar información estructurada.
- Y también en un archivo de Excel (.xlsx), usando la librería pandas.

Eso es todo, gracias por su atención.

**LINK DE VIDEO:**

[https://drive.google.com/file/d/13LGHxI3J5YaejyF60CT8XKv\\_aCGn3xh5/view?usp=sharing](https://drive.google.com/file/d/13LGHxI3J5YaejyF60CT8XKv_aCGn3xh5/view?usp=sharing)





