

*Proyecto Final: Análisis de Unidades Económicas del Sector Primario y de
Servicios en Aguascalientes*

Bases de datos para computo en la nube

Ruth Marcela Romero Rojas

Equipo:

Luis David Camarillo Leon

Juan Carlos Arias Camarena

Adrian Juarez Moreno

Francisco Fabian Rodriquez Sanchez

Desarrollo de software multiplataforma

03/08/25

Título

Exploración interactiva de datos del Directorio Estadístico Nacional de Unidades Económicas (DENUE) usando MongoDB y Python

Descripción del Proyecto

Este proyecto tiene como objetivo analizar y visualizar las características de las unidades económicas en Aguascalientes, extraídas del DENUE. Se implementó un flujo ETL (Extracción, Transformación y Carga) en Python y MongoDB para almacenar y consultar los datos. Posteriormente, se realizaron visualizaciones interactivas con Plotly para analizar aspectos como actividad económica, ubicación, y contacto.

Objetivos

Objetivo General:

Desarrollar un sistema interactivo para la consulta y análisis de unidades económicas en Aguascalientes mediante MongoDB, Python y datos del DENUE.

Objetivos Específicos:

- Extraer datos del DENUE proporcionados en formato CSV.
- Realizar limpieza y transformación de datos usando Pandas.
- Cargar los datos en MongoDB Atlas.
- Visualizar los datos mediante gráficas interactivas con Plotly.
- Desarrollar el flujo ETL y documentar el proceso completo.

Justificación

La visualización y análisis de unidades económicas locales es útil para comprender mejor la distribución de servicios, sectores y contactos de negocio en un territorio. Al usar tecnologías modernas como MongoDB y herramientas interactivas, se facilita el manejo de datos para gobiernos, emprendedores y ciudadanos interesados en oportunidades económicas locales.

Marco Teórico

Bases de Datos NoSQL: MongoDB es una base de datos orientada a documentos que facilita el almacenamiento de estructuras semiestructuradas como las provenientes del DENUE.

ETL:

- Extracción: Lectura de archivos CSV del DENUE.
- Transformación: Limpieza de columnas, normalización de textos y conversión a JSON.
- Carga: Uso de PyMongo para insertar los datos en MongoDB Atlas.

Librerías de Visualización:

- Plotly: Gráficas interactivas.
- Pandas: Análisis y transformación de datos.

Desarrollo del Proyecto

Proceso ETL:

- Extracción: Archivo CSV descargado del sitio del INEGI.
- Transformación: Pandas limpia columnas vacías, unifica campos como actividad económica y genera un DataFrame.
- Carga: Se convierte el DataFrame a JSON y se inserta en MongoDB Atlas.

Tecnologías y Herramientas:

MongoDB Atlas, Pandas, NumPy, Plotly, Google Colab

Base de Datos:

Colección: 'unidades_economicas'

Ejemplo:

```
{
  "Nombre": "CONTROL DE PLAGAS QUEZADA",
  "Actividad": "Servicios de fumigación agrícola",
  "Estrato": "0 a 5 personas",
  "Colonia": "VILLAS DE NUESTRA SEÑORA DE LA ASUNCIÓN",
  "Municipio": "AGUASCALIENTES",
  "Teléfono": "4492510135",
  "Latitud": 21.93982117,
  "Longitud": -102.25895324
}
```

Integración de Datos:

Los datos fueron limpiados y normalizados con Pandas, transformados a JSON y enviados a MongoDB Atlas para su consulta.

Backend / Conexiones:

```
client = pymongo.MongoClient(uri)
db = client['denue']
col = db['unidades_economicas']
col.insert_many(data_json)
```

Análisis y Resultados

- Se identificó que la mayoría de las unidades analizadas pertenecen al sector de fumigación agrícola.
- La mayoría operan con menos de 5 personas y están distribuidas en colonias urbanas.
- Algunas tienen sitio web o correo electrónico disponible para contacto.

Aprendizajes y Retos

- Manejo de datos incompletos o sin normalizar (ej. "NINGUNO" o nulos).
- Uso de PyMongo para manejar bases de datos en la nube.
- Aplicación de geolocalización para representar los negocios sobre mapas.

Futuro del Proyecto

- Ampliar a todo el estado o país.
- Incluir filtros por código postal, estrato de empleados, o sector económico.
- Crear una app web con Streamlit para búsqueda directa.

Estructura de los Documentos en MongoDB

Para este proyecto se utilizó MongoDB Atlas como sistema de almacenamiento de datos en la nube. Los datos provienen del Directorio Estadístico Nacional de Unidades Económicas (DENUE) del INEGI, específicamente para el estado de Aguascalientes. Tras aplicar el proceso ETL, cada unidad económica fue transformada en un documento en formato JSON e insertada en la base de datos.

Ejemplo de documento en la colección:

```
json
CopiarEditar
{
  "Nombre": "CONTROL DE PLAGAS QUEZADA",
  "Actividad": "Servicios de fumigación agrícola",
  "Estrato": "0 a 5 personas",
  "Colonia": "VILLAS DE NUESTRA SEÑORA DE LA ASUNCIÓN",
  "Municipio": "AGUASCALIENTES",
  "Teléfono": "4492510135",
  "Latitud": 21.93982117,
  "Longitud": -102.25895324
}
```

Cada documento representa una unidad económica con datos clave: nombre, actividad económica, número de empleados, colonia, municipio, contacto y coordenadas geográficas.

Colecciones Principales y Relaciones Lógicas

- **Base de datos:** `denue`
- **Colección principal:** `unidades_economicas`

Este proyecto maneja una sola colección (`unidades_economicas`) donde se almacenan todos los documentos. No hay múltiples colecciones ni relaciones como en bases de datos relacionales, ya que MongoDB es una base de datos NoSQL orientada a documentos. Sin embargo, los datos pueden ser consultados y filtrados por campos clave como `Municipio`, `Colonia`, `Actividad`, o `Estrato`.

Relaciones lógicas sugeridas (no implementadas en este proyecto, pero posibles para versiones futuras):

- Relación entre municipios y unidades económicas.
- Agrupación por sector económico (primario, secundario, terciario).
- Relación entre colonias y servicios disponibles.

Reglas de Seguridad (MongoDB Atlas)

Aunque el punto menciona Firebase, **este proyecto no utiliza Firebase** porque se trabaja directamente con MongoDB Atlas. Aun así, se aplicaron buenas prácticas de seguridad para proteger la base de datos en la nube:

- **Acceso mediante conexión segura (SSL)** desde Python usando la URI.
- **Autenticación mediante usuario y contraseña cifrada** en MongoDB Atlas.
- **Lista blanca de direcciones IP:** solo las IPs autorizadas pueden acceder a la base de datos (por ejemplo, la IP de Google Colab o la IP local del desarrollador).
- **Restricción de permisos:** las cuentas de usuario tienen permisos mínimos necesarios (lectura/escritura) sobre la base de datos `denuue`.

Estas configuraciones aseguran que solo los usuarios autorizados puedan acceder, modificar o consultar los datos, protegiendo así la confidencialidad y la integridad del proyecto.

Integración de Datos

Los datos fueron extraídos directamente del sitio web del INEGI, a través de los archivos CSV que conforman el Directorio Estadístico Nacional de Unidades Económicas (DENUE). En este caso, se utilizó el conjunto correspondiente al estado de Aguascalientes.

Una vez descargado el archivo, se utilizó la librería `pandas` en Python para abrir y procesar el contenido. Durante esta fase de transformación, se aplicaron varios pasos de limpieza:

- Eliminación de columnas innecesarias para el análisis.
- Renombrado de columnas para una mayor claridad (por ejemplo: `nom_estab` → `Nombre`, `tipoAct` → `Actividad`).
- Conversión de coordenadas a formato numérico.
- Eliminación de registros con valores nulos o marcados como "NINGUNO".
- Normalización de texto (mayúsculas y espacios) para estandarizar actividades y localidades.

Después del proceso de transformación, los datos se convirtieron a formato JSON y se insertaron en una base de datos MongoDB Atlas utilizando la librería `pymongo`. Cada registro se almacenó como un documento individual dentro de una colección llamada `unidades_economicas`.

Este proceso permitió tener una base de datos limpia, estructurada y lista para ser consultada mediante scripts Python y visualizaciones interactivas.

Backend / Conexiones

El backend del proyecto fue desarrollado en Python, aprovechando la conexión directa con MongoDB Atlas a través del cliente `pymongo`. Este backend permite ejecutar operaciones básicas como inserción, consulta y actualización sobre la colección `unidades_economicas`.

Conexión a MongoDB

```
python
CopiarEditar
from pymongo import MongoClient
import pandas as pd

# URI proporcionado por MongoDB Atlas (con usuario, contraseña y cluster)
uri =
"mongodb+srv://<usuario>:<contraseña>@<cluster>.mongodb.net/?retryWrites=true
&w=majority"
client = MongoClient(uri)

# Selección de base de datos y colección
db = client['denue']
col = db['unidades_economicas']
```

Inserción de documentos

```
python
CopiarEditar
# Supongamos que ya se limpió y transformó un DataFrame llamado df
data_json = df.to_dict(orient='records')
col.insert_many(data_json)
```

Ejemplos de consultas MongoDB

Buscar todos los negocios de fumigación agrícola:

```
python
CopiarEditar
col.find({"Actividad": {"$regex": "fumigación", "$options": "i"}})
```

Contar cuántas unidades hay por colonia:

```
python
CopiarEditar
col.aggregate([
    {"$group": {"_id": "$Colonia", "total": {"$sum": 1}}},
    {"$sort": {"total": -1}}
])
```

Ejemplo de actualización

Actualizar un teléfono en un registro específico:

```
python
CopiarEditar
col.update_one(
    {"Nombre": "CONTROL DE PLAGAS QUEZADA"},
    {"$set": {"Teléfono": "4499999999"}}
)
```

Referencias y Recursos

- INEGI: <https://www.inegi.org.mx>
- MongoDB Atlas: <https://www.mongodb.com/cloud/atlas>
- PyMongo Docs: <https://pymongo.readthedocs.io>
- Plotly Express: <https://plotly.com/python/plotly-express/>

Enlaces y Evidencia

- Google Colab Notebook: [enlace al notebook]
- Documento Word con capturas de pantalla adjunto al entregable.

Capturas de pantalla del proceso

Inserción de datos en Mongo db

```
# -----  
# Paso 3: Conexión a MongoDB Atlas y carga de datos  
# -----  
uri =  
"mongodb+srv://24102016:24102016@cluster0.z4kfcxb.mongodb.net/?retryWr  
ites=true&w=majority"  
client = MongoClient(uri)  
  
db = client["proyecto_datos"]  
collection = db["denue_aguscalientes"]  
  
# Convertir DataFrame a diccionario para insertar  
data_dict = df.to_dict('records')  
  
# Insertar datos en MongoDB  
try:  
    collection.insert_many(data_dict)  
    print("\n✅ Datos insertados en MongoDB correctamente.")  
except Exception as e:  
    print("\n❌ Error al insertar datos en MongoDB:", e)
```

evidencia de datos insertados en Mongo

The screenshot displays the MongoDB Atlas web interface. On the left sidebar, the 'Data Services' tab is selected, and the 'Collections' sub-tab is active. The main panel shows the 'proyecto_datos.denue_aguscalientes' collection. The 'Overview' sub-tab is selected, displaying a list of documents. The first document is expanded, showing its JSON structure. The document contains various fields including 'id', 'nombre_de_la_unidad_academica', 'descripcion', 'tipo_de_vialidad', 'nombre_de_la_vialidad', 'tipo_de_entrada_vialidad', 'nombre_de_entrada_vialidad', 'tipo_de_salida_vialidad', 'nombre_de_salida_vialidad', 'numero_exterior', 'tipo_de_asentamiento_humano', 'nombre_de_asentamiento_humano', 'tipo_centro_comercial', 'corredor_industrial', 'numero_de_local', 'codigo_postal', 'clave_entidad', 'entidad_federativa', and 'clave_municipio'.

Notebook Colab en ejecución

```
ProyectoWEB.pyynb 5 Guardado...
Archivos Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

except Exception as e:
    print("\n❌ Error al insertar datos en MongoDB", e)

# =====
# Paso 4: Visualizaciones con Plotly
# =====

# 4.1 Histograma: distribución por estrato personal ocupado
if 'descripcion_estrato_personal_ocupado' in df.columns:
    fig1 = px.histogram(df, x='descripcion_estrato_personal_ocupado',
                        title='Distribución de unidades económicas por estrato de personal ocupado',
                        labels={'descripcion_estrato_personal_ocupado': 'Estrato de personal ocupado'},
                        nbins=10)
    fig1.show()
else:
    print("Columna 'descripcion_estrato_personal_ocupado' no encontrada para gráfico.")

# 4.2 Mapa de puntos geográficos
if 'latitud' in df.columns and 'longitud' in df.columns:
    fig2 = px.scatter_mapbox(df, lat='latitud', lon='longitud',
                            hover_data=['municipio', 'entidad_federativa', 'descripcion_estrato_personal_ocupado'] if 'descripcion_estrato_personal_ocupado' in df.columns else [],
                            color='descripcion_estrato_personal_ocupado' if 'descripcion_estrato_personal_ocupado' in df.columns else None,
                            zoom=5, height=400)
    fig2.update_layout(mapbox_style='open-street-map')
    fig2.update_layout(title='Mapa de unidades económicas en Aguascalientes')
    fig2.show()
else:
    print("Columnas 'latitud' y/o 'longitud' no encontradas para mapa.")

# 4.3 Conteo por municipio y gráfico de barras
if 'municipio' in df.columns:
    conteo_municipio = df['municipio'].value_counts().reset_index()
    conteo_municipio.columns = ['municipio', 'cantidad']

    fig3 = px.bar(conteo_municipio, x='municipio', y='cantidad',
                  title='Cantidad de unidades económicas por municipio',
                  labels={'cantidad': 'Cantidad', 'municipio': 'Municipio'})
    fig3.show()
else:
    print("Columna 'municipio' no encontrada para gráfico de barras.")

--
1-A11-A 10 17-A 10/3 17-A 100-00
11-A111-A 10 17-A 10/3 17-A 100-00
```

Visualizaciones (barras y mapa)

