

# Dashboard de Análisis Exploratorio de Datos (EDA) - Taxis de NYC

## 1. Resumen del Proyecto

Este proyecto presenta un dashboard web interactivo para el análisis exploratorio de un dataset de viajes de taxi en la ciudad de Nueva York.



La aplicación, desarrollada con **Dash** y **Plotly**, permite a los usuarios visualizar patrones, identificar tendencias y extraer insights a través de una interfaz dinámica y responsiva.

## 2. Stack Tecnológico

**Lenguaje:** Python 3.13

**Análisis de Datos:** Pandas

**Visualización y Dashboard:** Plotly & Dash

## 3. Guía de Inicio Rápido

Sigue estos pasos para ejecutar el dashboard en tu entorno local.

### Paso 1: Prerrequisitos

Asegúrate de tener **Python 3.13** instalado en tu sistema.

Confirma que el archivo `nyc_taxi.csv` se encuentra en la carpeta `datos/` del proyecto principal.

### Paso 2: Configuración del Entorno Virtual

Desde la terminal, en la **raíz del proyecto** (`ejercicios-bigdata`), ejecuta los siguientes comandos para crear y configurar el entorno virtual:

```
# Crea un nuevo entorno virtual llamado .venv
```

```
python -m venv .venv
.venv\Scripts\Activate.ps1
```

### Paso 3: Instalación de Dependencias

Este dashboard tiene sus propias dependencias. Para no interferir con el archivo `requirements.txt` principal del repositorio, se ha creado un archivo específico llamado `requirements_2.txt`.

Instala las dependencias usando este archivo:

```
pip install -r requirements_2.txt
```

### Paso 4: Ejecutar la Aplicación

Una vez configurado el entorno, inicia el servidor de Dash. Asegúrate de ejecutar el siguiente comando también desde la **raíz del proyecto**:

```
python dashboards/fernando-ramos-dashboard/app_dash.py
```

El proceso puede tardar unos segundos mientras se cargan y procesan los datos.

### Paso 5: Acceder al Dashboard

Cuando la terminal muestre un mensaje como `Dash is running on http://127.0.0.1:8050/`, abre tu navegador web y ve a esa dirección:

<http://127.0.0.1:8050/>

Para detener el servidor, vuelve a la terminal y presiona `Ctrl + C`.

#### 4. Hallazgos y Conclusiones del Análisis

El análisis exploratorio a través del dashboard revela varias tendencias clave sobre la dinámica de los viajes de taxi en NYC:

**El Ritmo de la Ciudad: Patrones Semanales.** Se observa un claro patrón en la actividad a lo largo de la semana. El número de viajes aumenta progresivamente desde el lunes, alcanza su pico los **viernes**, y desciende durante el fin de semana, siendo el domingo el día de menor actividad.

1. **El Dominio del Plástico: Métodos de Pago.** El pago con **tarjeta de crédito** es el método predominante, superando ampliamente al pago en **efectivo**. Esto sugiere una alta bancarización de los usuarios y una preferencia por la comodidad del pago digital.

2. **La Generosidad Digital: Propinas y Tipo de Pago.** El análisis de propinas es concluyente: se registran casi exclusivamente en los pagos con **tarjeta**. Esto indica que las propinas en efectivo no se digitalizan en el sistema o son menos comunes, y que la funcionalidad de propina en los terminales de pago es un factor clave en los ingresos de los conductores.

3. **A Mayor Coste, Mayor Propina.** Existe una clara correlación positiva entre el **importe total** del viaje y la **propina**. A mayor coste del viaje, los pasajeros tienden a dejar una propina proporcionalmente mayor, especialmente en los pagos con tarjeta.

5. **La Lógica del Taxímetro: Distancia vs. Tarifa.** Como es de esperar, hay una fuerte correlación positiva entre la **distancia** del viaje y la **tarifa total**. El modelo de precios es consistente y se basa fundamentalmente en la distancia recorrida.

#### 5. Futuras Mejoras

- **Filtros Avanzados:** Añadir más filtros, como por hora del día o RatecodeID, para un análisis más granular.
- **Análisis Geográfico:** Incorporar un mapa interactivo (Choropleth) para visualizar las zonas de origen (PULocationID) y destino (DOLocationID) más populares, utilizando un archivo GeoJSON.
- **Despliegue:** Empaquetar la aplicación en un contenedor Docker y desplegarla en un servicio en la nube (como Heroku, Render o AWS) para hacerla accesible públicamente.