

# Taller de Ciencia de Datos para Gobierno

Desarrollo de competencias básicas en Python para análisis de datos gubernamentales.



1 Hora



Presencial / Virtual



Personal de Informática

## Preparación Previa

### Opción A: Instalación Local

- Instalar **Anaconda Individual**.
- Verificar la instalación de Python y JupyterLab.
- Instalar librerías clave como `folium`, `geopandas` y `streamlit`.

### Opción B: Google Colab

- Requiere una cuenta de Google y navegador actualizado.

- Acceso en la nube sin necesidad de instalación.
- Conexión a internet estable.

## Módulo 1: Fundamentos y Setup

### Temas Principales

- Introducción a la ciencia de datos en el sector público.
- Preparación del entorno de trabajo con JupyterLab.
- Fundamentos de Python: variables, listas y diccionarios.

### Ejemplo de Código

```
# Declaración de variables
nombre_taller = "Ciencia de Datos"
duracion_horas = 1

# Creación de una lista de participantes
participantes = ["Juan", "María", "Pedro"]

# Imprimir información
print(f"Taller: {nombre_taller}")
print(f"Duración: {duracion_horas} horas")
print(f"Total de participantes: {len(participantes)}")
```

## Módulo 2: Pandas y Manipulación de Datos

### Aprendizajes Clave

- Introducción a **Pandas** y los `DataFrames`.

- Carga y exploración inicial de datos (ejemplo de DENEUE).
- Operaciones de filtrado, selección y agrupación de datos.

## Ejemplo de Código

```
# Importar la librería Pandas
import pandas as pd

# Crear un DataFrame de ejemplo
datos = {
    'estado': ['CDMX', 'Jalisco', 'CDMX', 'Edo. de México'],
    'sector': ['Tecnología', 'Salud', 'Salud', 'Tecnología'],
    'ingresos': [15000, 22000, 18000, 30000]
}
df = pd.DataFrame(datos)

# Filtrar datos de CDMX y mostrar
ingresos_cdmx = df[df['estado'] == 'CDMX']['ingresos'].sum()
print(f"Ingresos totales en CDMX: {ingresos_cdmx}")
```

# Módulo 3: Visualización de Datos

## Temas Principales

- Creación de gráficos estáticos con **Matplotlib** y **Seaborn**.
- Visualización de distribuciones y relaciones entre variables.
- Creación de mapas interactivos con **Folium**.

## Ejemplo de Código

```
# Importar librerías de visualización
import matplotlib.pyplot as plt
import seaborn as sns

# Crear un DataFrame de ejemplo para visualización
datos_grafico = {
```

```

        'tipo': ['Escuelas', 'Hospitales', 'Oficinas'],
        'cantidad': [120, 45, 80]
    }
    df_grafico = pd.DataFrame(datos_grafico)

# Crear un gráfico de barras simple
plt.figure(figsize=(8, 6))
sns.barplot(x='tipo', y='cantidad', data=df_grafico, palette='viridis')
plt.title('Establecimientos por tipo')
plt.xlabel('Tipo de Establecimiento')
plt.ylabel('Cantidad')
plt.show()

```

## Módulo 4: API DENUE y Dashboards

### Aplicación Práctica

- Consumo de datos en tiempo real de la [API del INEGI-DENUE](#).
- Análisis de datos reales de establecimientos económicos.
- Construcción de un dashboard interactivo con [Streamlit](#).

### Ejemplo de Código

```

# Ejemplo de código de Streamlit para un dashboard básico
import streamlit as st

st.title('Dashboard de Datos del DENUE')

# La función para llamar a la API se definiría aquí.
# Por ejemplo: df = obtener_datos_denue('CDMX')

st.header('Análisis de establecimientos')
st.markdown('Aquí iría un análisis y visualizaciones interactivas.')

st.selectbox(
    "Selecciona un Sector:",

```

```
("Salud", "Educación", "Tecnología")
)
```

## Módulo 5: Proyecto Final y Cierre

### Proyecto Integrador

- Aplicación de todos los conocimientos en un análisis de **vocación productiva regional**.
- Identificación de clusters geográficos y sectores dominantes.
- Presentación y discusión de resultados.

### Ejemplo de Código

```
# Pseudocódigo del proyecto final que integra los módulos
# 1. Cargar datos de la API de DENUE
# df_denue = obtener_datos_denue()

# 2. Limpieza y preparación de datos
# df_denue_limpio = limpiar_datos(df_denue)

# 3. Análisis y visualización
# grafico_mapa_calor(df_denue_limpio)
# grafico_sectores_dominantes(df_denue_limpio)

# 4. Creación de un dashboard interactivo para compartir
# crear_dashboard_streamlit(df_denue_limpio)
```

## Recursos y Próximos Pasos

¡Gracias por su participación! El aprendizaje no termina aquí.



Acceso al código y materiales en GitHub.



Certificado de participación.



Sugerencias para continuar: APIs oficiales, otros datasets y machine learning.