Jesus Alejandro Valencia Valadez

3013480

Fundamentos Big Data

In [23]:
```
import pandas as pd
import matplotlib.pyplot as plt
#import numpy as np

pd.set_option('display.max_colwidth',None)
pd.options.display.float_format = '{:,.2f}'.format
```

In [24]:
```
df = pd.read_csv("Casos_Diarios_Estado_Nacional_Confirmados_20230308.csv", par
df.head()
```

Out[24]:

	cve_ent	poblacion	nombre	26-02-2020	28-02-2020	29-02-2020	01-03-2020	02-03-2020	03-03-2020	04-03-2020	...	26-02-202
27-02-2020												
0	1	1434635	AGUASCALIENTES	0	0	0	0	0	0	0	...	
0	2	3634868	BAJA CALIFORNIA	0	0	0	0	0	0	0	...	
0	3	804708	BAJA CALIFORNIA SUR	0	0	0	0	0	0	0	...	1
0	4	1000617	CAMPECHE	0	0	0	0	0	0	0	...	
0	7	5730367	CHIAPAS	0	0	1	0	0	0	0	...	

5 rows × 1108 columns

In [25]:
```
columnas = list(df.columns)
dias = columnas[3:]
min_dia = dias[0]
max_dias = dias[-1]
```

In [26]:
```
meses = []
for dia in dias:
    mes = dia[3:]
    if mes not in meses:
        meses.append(mes)

print(meses)
```

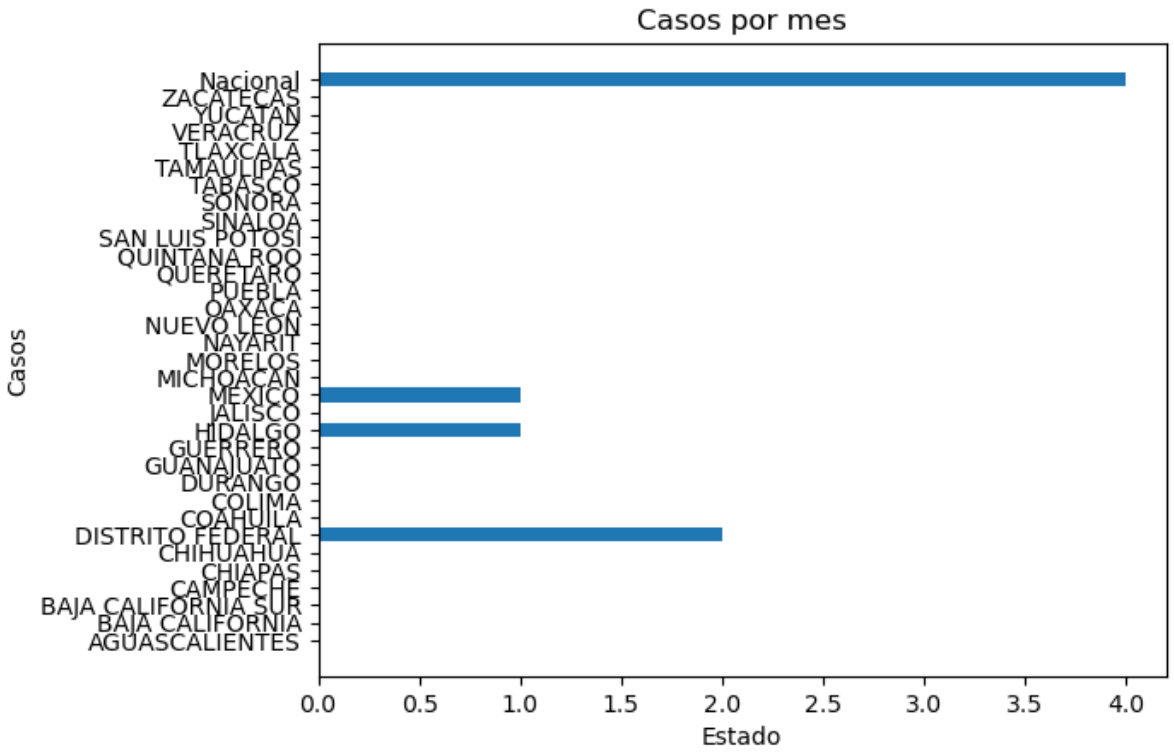
```
['02-2020', '03-2020', '04-2020', '05-2020', '06-2020', '07-2020', '08-2020',
'09-2020', '10-2020', '11-2020', '12-2020', '01-2021', '02-2021', '03-2021',
'04-2021', '05-2021', '06-2021', '07-2021', '08-2021', '09-2021', '10-2021',
'11-2021', '12-2021', '01-2022', '02-2022', '03-2022', '04-2022', '05-2022',
'06-2022', '07-2022', '08-2022', '09-2022', '10-2022', '11-2022', '12-2022',
'01-2023', '02-2023', '03-2023']
```

Números de casos confirmados por cada mes

In [12]:
```
zeros = []
for i in range(0, len(meses)):
    zeros.append(0)
zeros1 = [x for x in range(len(meses))]
print(zeros)
print(zeros1)
```

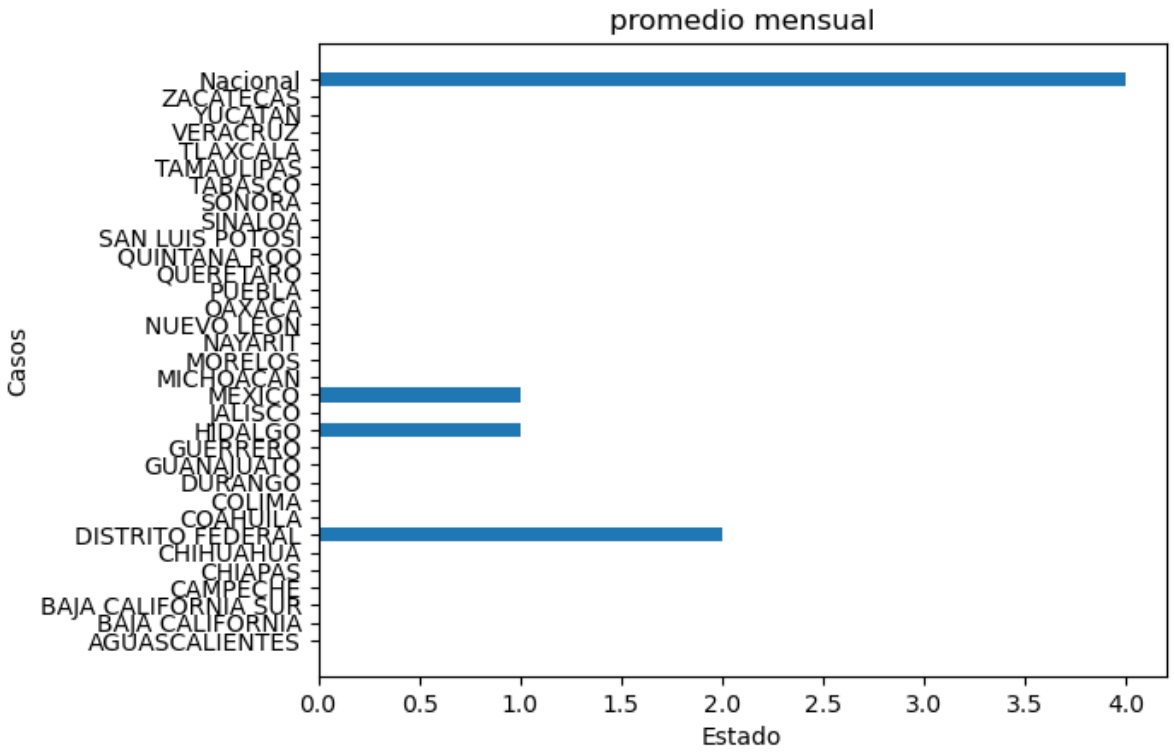
```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 2
1, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37]
```

In [47]:
```
def casos_por_mes():
    resultados = dict(zip(meses,[0 for x in range(len(meses))]))
    for dia in dias:
        mes = dia[3:]
        resultados[mes] += df[dia]
    return resultados


res = casos_por_mes()
res_df = pd.DataFrame(res)
resul = pd.concat([df["nombre"], res_df], axis = 'columns')

# graficar los resultados
plt.barh(resul["nombre"],resul.index)
plt.title('Casos por mes')
plt.xlabel('Estado')
plt.ylabel('Casos')
plt.show()
```



Promedio Mensual

In [46]:
```
def prom_casos_por_mes():
    resultados = dict(zip(meses,[0 for x in range(len(meses))]))
    num_dias = dict(zip(meses,[0 for x in range(len(meses))]))

    for dia in dias:
        mes = dia[3:]
        resultados[mes] += df[dia]
        num_dias[mes] += 1
    for key in resultados:
        resultados[key] //= num_dias[key]
    return resultados


res2 = prom_casos_por_mes()
res_df2 = pd.DataFrame(res2)
resu = pd.concat([df["nombre"], res_df2], axis = 'columns')

# graficar los resultados
plt.barh(resu["nombre"],resu.index)
plt.title('promedio mensual')
plt.xlabel('Estado')
plt.ylabel('Casos')
plt.show()
```



Porcentaje poblacion por mes

In [8]:
```
def porcentaje_por_mes():
    resultados = dict(zip(meses,[0 for x in range(len(meses))]))
    num_dias = dict(zip(meses,[0 for x in range(len(meses))]))

    for dia in dias:
        mes = dia[3:]
        resultados[mes] += df[dia]
        num_dias[mes] += 1
    for key in resultados:
        resultados[key] /= df["poblacion"]
    return resultados

res3 = porcentaje_por_mes()
res_df3 = pd.DataFrame(res3)
pd.concat([df["nombre"], res_df3], axis = 'columns')
```

Out[8]:

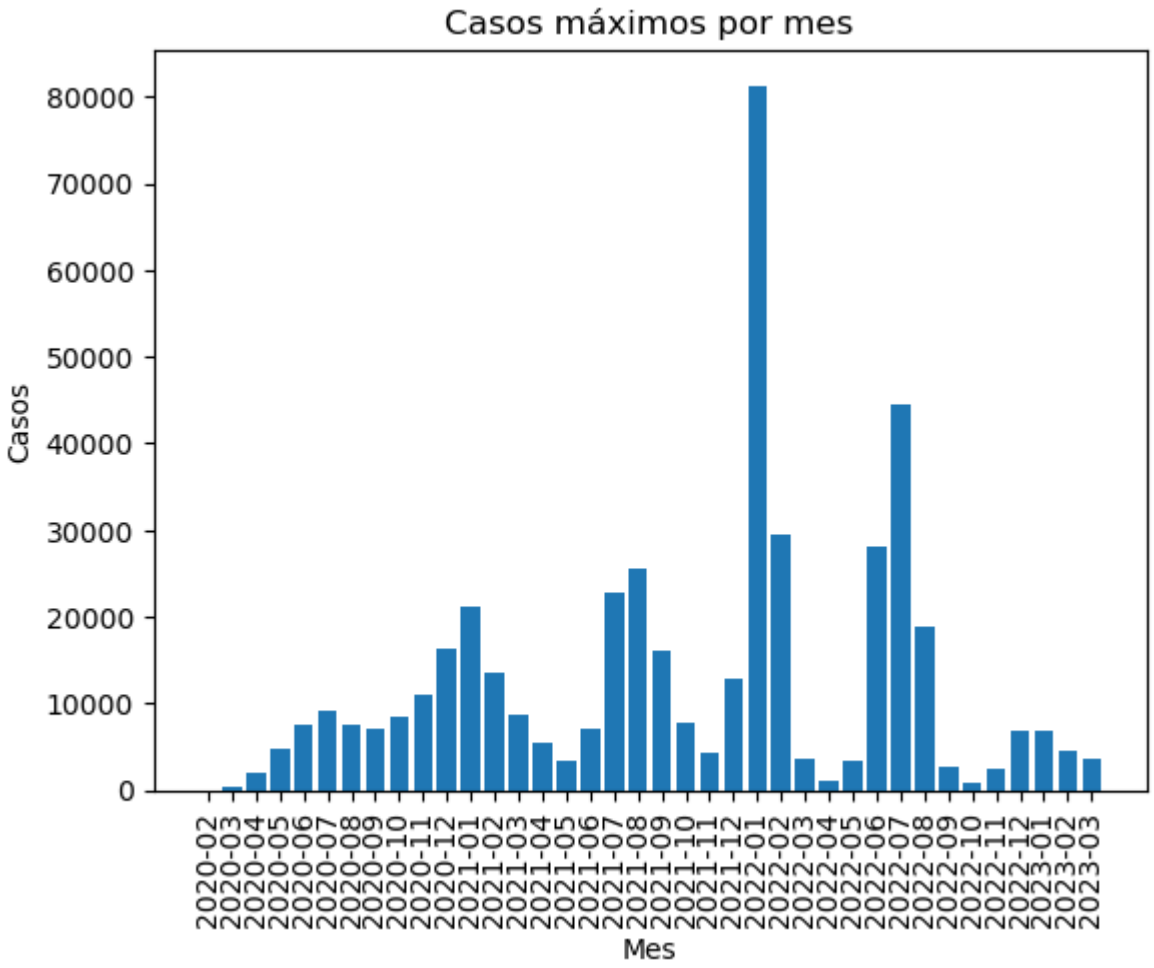
27-02-2020	nombre	02-2020	03-2020	04-2020	05-2020	06-2020	07-2020	08-2020	09-2020	10-2020	...	06-2022	07-2022
0	AGUASCALIENTES	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	BAJA CALIFORNIA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	BAJA CALIFORNIA SUR	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.01	0.01
0	CAMPECHE	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	CHIAPAS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	CHIHUAHUA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
2	DISTRITO FEDERAL	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.01	0.02
0	COAHUILA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	COLIMA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	DURANGO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	GUANAJUATO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	GUERRERO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
1	HIDALGO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	JALISCO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
1	MEXICO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	MICHOACAN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	MORELOS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	NAYARIT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	NUEVO LEON	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	OAXACA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	PUEBLA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	QUERETARO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	QUINTANA ROO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.01	0.00
0	SAN LUIS POTOSI	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	SINALOA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.01	0.01
0	SONORA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	TABASCO	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	TAMAULIPAS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	TLAXCALA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	VERACRUZ	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00
0	YUCATAN	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01
0	ZACATECAS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00

	nombre	02-2020	03-2020	04-2020	05-2020	06-2020	07-2020	08-2020	09-2020	10-2020	...	06-2022	07-2022
27-02-2020													
4	Nacional	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.01

33 rows × 39 columns

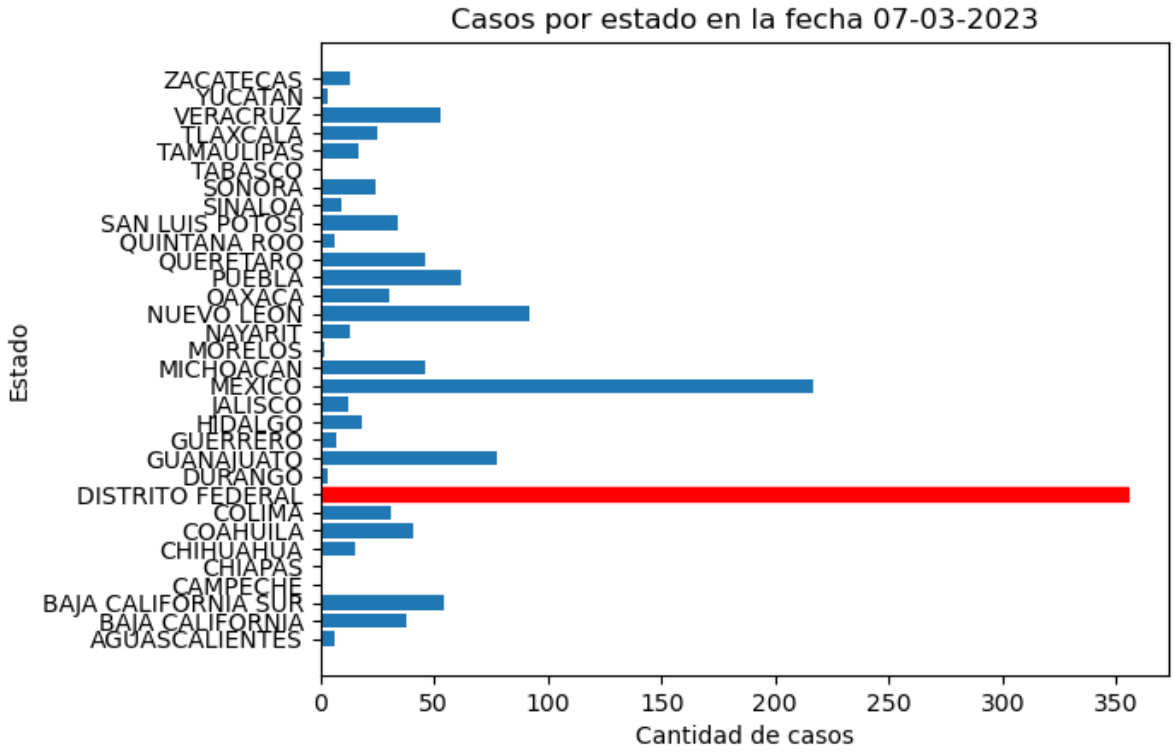
Día del mes en el que se presento el mayor numero de casos confirmados

In [49]:
```
def casos_maximos_por_mes():
    resultados = []

    for dia in dias:
        fecha = pd.to_datetime(dia, format='%d-%m-%Y')
        year, mes, _ = fecha.strftime("%Y-%m-%d").split("-")
        casos_mes = df.loc[df["cve_ent"] == 0, dia].sum()

        nuevo_resultado = (year, mes, dia, casos_mes)

        if not resultados:
            resultados.append(nuevo_resultado)
        else:
            encontrado = False
            for i, resultado in enumerate(resultados):
                if resultado[:2] == (year, mes):
                    if casos_mes > resultado[3]:
                        resultados[i] = nuevo_resultado
                    encontrado = True
                    break

            if not encontrado:
                resultados.append(nuevo_resultado)

    return resultados

resultados = casos_maximos_por_mes()

# Crear una lista de los valores máximos por mes
maximos_mes = [resultado[3] for resultado in resultados]

# Crear una lista de los meses en formato "YYYY-MM"
meses = [f"{resultado[0]}-{resultado[1]}" for resultado in resultados]

# Crear el gráfico de barras
plt.bar(meses, maximos_mes)

# Personalizar el gráfico
plt.title('Casos máximos por mes')
plt.xlabel('Mes')
plt.ylabel('Casos')
plt.xticks(rotation=90)

# Mostrar el gráfico
plt.show()
```


Casos máximos por mes
Casos
0
10000
20000
30000
40000
50000
60000
70000
80000
2020-02
2020-03
2020-04
2020-05
2020-06
2020-07
2020-08
2020-09
2020-10
2020-11
2020-12
2021-01
2021-02
2021-03
2021-04
2021-05
2021-06
2021-07
2021-08
2021-09
2021-10
2021-11
2021-12
2022-01
2022-02
2022-03
2022-04
2022-05
2022-06
2022-07
2022-08
2022-09
2022-10
2022-11
2022-12
2023-01
2023-02
2023-03
Mes

Funcion personalizada, Estado con mas casos

In [54]:
```
df = df.drop(df[df['nombre'] == 'Nacional'].index)
# Obtener la cantidad de casos por estado en la fecha '07-03-2023'
datos_por_estado = df.groupby('nombre').sum()
casos_por_estado = datos_por_estado.loc[:, '07-03-2023']

# Obtener el estado con mayor cantidad de casos
estado_mas_casos = casos_por_estado.idxmax()

# Crear el gráfico de barras horizontales
plt.barh(casos_por_estado.index, casos_por_estado.values)
plt.title('Casos por estado en la fecha 07-03-2023')
plt.xlabel('Cantidad de casos')
plt.ylabel('Estado')

# Resaltar el estado con mayor cantidad de casos
index_estado_mas_casos = casos_por_estado.index.get_loc(estado_mas_casos)
plt.gca().get_children()[index_estado_mas_casos].set_color('red')

plt.show()
```



In [ ]: