# Caso proyecto DataEngineer

## Planteamiento del Problema: De acuerdo a los datos de Covid19 , Población General y numero de Asegurados establecer criterios para conocer donde es posible aumentar la venta de seguros personales

Realizado por Gabriel Odreman

## Importar las librerias a utilizar

In [2]:
```
#librerias
import pandas as pd
import xlsxwriter
import sqlite3
import nbconvert
```

## 1. Establecer conexión con las datos.

### 1.1 Población del reporte de datos

### Datos de la poblacion estimada , bajamos los datos a una hoja de excel

https://censusreporter.org/data/table/?table=B01001&geo_ids=01000US,040|01000US&primary_geo_id=01000US#valueType|estimate

El archivo en excel debe ser modificado para eliminar los totales , luego añadir una columna nueva y colocar female y Male donde corresponda.

In [3]:
```
dfpoblacion = pd.read_excel('D:/d/2-ESTUDIO/Proyecto Data Engineer/poblacionbyedad-sex-state.xlsx')
dfpoblacion.head(10)
```

Out[3]:

| | AgeCategoria | genero | Alabama | Alaska | Arizona | Arkansas | California | Colorado | Connecticut | Delaware | ... | South Dakota | Tennessee | Texas |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Under 5 years | Male | 149579 | 26684 | 220346 | 96802 | 1233088 | 168650 | 93725 | 27809 | ... | 30408 | 208551 | 1020529 |
| 1 | 5 to 9 years | Male | 150937 | 26976 | 225723 | 98969 | 1242495 | 177258 | 97435 | 27824 | ... | 31359 | 208045 | 1026859 |
| 2 | 10 to 14 years | Male | 160287 | 25643 | 246866 | 103012 | 1328272 | 187579 | 113376 | 30047 | ... | 30555 | 222622 | 1088140 |
| 3 | 15 to 17 years | Male | 96832 | 14990 | 143488 | 60923 | 774841 | 110519 | 71199 | 17720 | ... | 17675 | 131806 | 629311 |
| 4 | 18 and 19 years | Male | 65459 | 9523 | 100233 | 41277 | 525740 | 76549 | 52105 | 12445 | ... | 12676 | 86714 | 415143 |
| 5 | 20 years | Male | 36705 | 5710 | 53533 | 24123 | 285907 | 41345 | 25215 | 6350 | ... | 6354 | 47706 | 211272 |
| 6 | 21 years | Male | 33089 | 6184 | 51827 | 20634 | 281350 | 37714 | 24644 | 6405 | ... | 6507 | 44634 | 210823 |
| 7 | 22 to 24 years | Male | 93871 | 18561 | 150589 | 58783 | 821103 | 118766 | 75457 | 17024 | ... | 17402 | 132010 | 608054 |
| 8 | 25 to 29 years | Male | 167179 | 32803 | 266718 | 102338 | 1594429 | 235076 | 114689 | 33274 | ... | 30689 | 243857 | 1086864 |
| 9 | 30 to 34 years | Male | 149890 | 29948 | 244743 | 95817 | 1505530 | 231592 | 110988 | 31190 | ... | 29794 | 219559 | 1054435 |

10 rows × 53 columns

### Ahora que tenemos los datos debe ser modificada para desnormaliza los estados en una sola columna y pivotear el genero y Categoria a Columnas

In [4]:
```
dfpoblacion1= pd.melt(dfpoblacion.reset_index(), id_vars=['AgeCategoria','genero']).sort_values(by=['genero'])
dfpoblacion1.head(10)
```

Out[4]:

| | AgeCategoria | genero | variable | value |
|---|---|---|---|---|
| **1195** | 85 years and over | Female | Mississippi | 34850 |
| **1331** | 75 to 79 years | Female | Nebraska | 27181 |
| **1330** | 70 to 74 years | Female | Nebraska | 37772 |
| **1329** | 67 to 69 years | Female | Nebraska | 30246 |
| **1328** | 65 and 66 years | Female | Nebraska | 21720 |
| **1327** | 62 to 64 years | Female | Nebraska | 36420 |
| **1326** | 60 and 61 years | Female | Nebraska | 25077 |
| **1325** | 55 to 59 years | Female | Nebraska | 61292 |
| **1324** | 50 to 54 years | Female | Nebraska | 55569 |
| **1323** | 45 to 49 years | Female | Nebraska | 53330 |

primero creamos dos tablas una de female y otra de Male luego a cada una las unpivot para que los valores de fila esten en columnas luego sumamos las columnas que necesitamos y unimos las tablas.

In [5]:
```
is_male = dfpoblacion1.loc[:, 'genero'] == 'Male'
df_male = dfpoblacion1.loc[is_male]
df_male.head()
```

Out[5]:

| | AgeCategoria | genero | variable | value |
|---|---|---|---|---|
| **1843** | 15 to 17 years | Male | Rhode Island | 18804 |
| **1842** | 10 to 14 years | Male | Rhode Island | 30918 |
| **1841** | 5 to 9 years | Male | Rhode Island | 27475 |
| **1840** | Under 5 years | Male | Rhode Island | 27985 |
| **278** | 10 to 14 years | Male | Colorado | 187579 |

In [6]:
```
is_female = dfpoblacion1.loc[:, 'genero'] == 'Female'
df_female = dfpoblacion1.loc[is_female]
df_female.head()
```

Out[6]:

| | AgeCategoria | genero | variable | value |
|---|---|---|---|---|
| **1195** | 85 years and over | Female | Mississippi | 34850 |
| **1331** | 75 to 79 years | Female | Nebraska | 27181 |
| **1330** | 70 to 74 years | Female | Nebraska | 37772 |
| **1329** | 67 to 69 years | Female | Nebraska | 30246 |
| **1328** | 65 and 66 years | Female | Nebraska | 21720 |

In [7]:
```
#filtramos los datos para tener solamente 3 columnas , la columna estado nos permite relacionarla con las otras
df_male1= df_male[['AgeCategoria','variable','value']]
#cambiamos el nombre de las columnas
df_male1.columns = ['AgeCategoria','state_name', 'PoblacionMale']
df_male1.head(10)
```

Out[7]:

| | AgeCategoria | state_name | PoblacionMale |
|---|---|---|---|
| **1843** | 15 to 17 years | Rhode Island | 18804 |
| **1842** | 10 to 14 years | Rhode Island | 30918 |
| **1841** | 5 to 9 years | Rhode Island | 27475 |
| **1840** | Under 5 years | Rhode Island | 27985 |
| **278** | 10 to 14 years | Colorado | 187579 |
| **277** | 5 to 9 years | Colorado | 177258 |
| **1844** | 18 and 19 years | Rhode Island | 16812 |
| **279** | 15 to 17 years | Colorado | 110519 |
| **280** | 18 and 19 years | Colorado | 76549 |
| **281** | 20 years | Colorado | 41345 |

In [8]:
```
# Debemos cabiarle el nombre a AgeCategoria para que sea el Index y el hacer el pivot quede una columna con ese
df_male1.columns = ['Indice','state_name','PoblacionMale']

df_male2 = df_male1.pivot_table('PoblacionMale', ['state_name'], 'Indice')
df_male2.head(5)
```

Out[8]:

| Indice | 10 to 14 years | 15 to 17 years | 18 and 19 years | 20 years | 21 years | 22 to 24 years | 25 to 29 years | 30 to 34 years | 35 to 39 years | 40 to 44 years | ... | 55 to 59 years | 60 and 61 years | 62 to 64 years | 65 and 66 years |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| state_name | | | | | | | | | | | | | | | |
| Alabama | 160287 | 96832 | 65459 | 36705 | 33089 | 93871 | 167179 | 149890 | 144863 | 142671 | ... | 158953 | 61732 | 90266 | 53730 |
| Alaska | 25643 | 14990 | 9523 | 5710 | 6184 | 18561 | 32803 | 29948 | 26780 | 21996 | ... | 25402 | 9691 | 13401 | 8263 |
| Arizona | 246866 | 143488 | 100233 | 53533 | 51827 | 150589 | 266718 | 244743 | 228667 | 217678 | ... | 209429 | 84679 | 117609 | 76146 |
| Arkansas | 103012 | 60923 | 41277 | 24123 | 20634 | 58783 | 102338 | 95817 | 94009 | 90127 | ... | 95723 | 37341 | 51855 | 32070 |
| California | 1328272 | 774841 | 525740 | 285907 | 281350 | 821103 | 1594429 | 1505530 | 1377089 | 1265725 | ... | 1216743 | 470895 | 618484 | 378972 |

5 rows × 23 columns

In [9]:
```
df_male2= pd.DataFrame(df_male2)
df_male2.reset_index(inplace=True, drop=False)
df_male2.head(5)
```

Out[9]:

| Indice | state_name | 10 to 14 years | 15 to 17 years | 18 and 19 years | 20 years | 21 years | 22 to 24 years | 25 to 29 years | 30 to 34 years | 35 to 39 years | ... | 55 to 59 years | 60 and 61 years | 62 to 64 years | 65 and 66 years |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Alabama | 160287 | 96832 | 65459 | 36705 | 33089 | 93871 | 167179 | 149890 | 144863 | ... | 158953 | 61732 | 90266 | 53730 |
| 1 | Alaska | 25643 | 14990 | 9523 | 5710 | 6184 | 18561 | 32803 | 29948 | 26780 | ... | 25402 | 9691 | 13401 | 8263 |
| 2 | Arizona | 246866 | 143488 | 100233 | 53533 | 51827 | 150589 | 266718 | 244743 | 228667 | ... | 209429 | 84679 | 117609 | 76146 |
| 3 | Arkansas | 103012 | 60923 | 41277 | 24123 | 20634 | 58783 | 102338 | 95817 | 94009 | ... | 95723 | 37341 | 51855 | 32070 |
| 4 | California | 1328272 | 774841 | 525740 | 285907 | 281350 | 821103 | 1594429 | 1505530 | 1377089 | ... | 1216743 | 470895 | 618484 | 378972 |

5 rows × 24 columns

## Debemos unir las columnas que se requiren sumar

In [10]:
```
columns_names = df_male2.columns.values
columns_names
```

Out[10]:
```
array(['state_name', '10 to 14 years', '15 to 17 years',
       '18 and 19 years', '20 years', '21 years', '22 to 24 years',
       '25 to 29 years', '30 to 34 years', '35 to 39 years',
       '40 to 44 years', '45 to 49 years', '5 to 9 years',
       '50 to 54 years', '55 to 59 years', '60 and 61 years',
       '62 to 64 years', '65 and 66 years', '67 to 69 years',
       '70 to 74 years', '75 to 79 years', '80 to 84 years',
       '85 years and over', 'Under 5 years'], dtype=object)
```

In [11]:
```
df_male2['totalAge19_64Male'] = df_male2['18 and 19 years'] + df_male2[ '20 years']+ df_male2[ '21 years']+df_m
df_male2.head(5)
```

Out[11]:

| Indice | state_name | 10 to 14 years | 15 to 17 years | 18 and 19 years | 20 years | 21 years | 22 to 24 years | 25 to 29 years | 30 to 34 years | 35 to 39 years | ... | 60 and 61 years | 62 to 64 years | 65 and 66 years | 67 to 69 years | 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Alabama | 160287 | 96832 | 65459 | 36705 | 33089 | 93871 | 167179 | 149890 | 144863 | ... | 61732 | 90266 | 53730 | 73695 | 98 |
| 1 | Alaska | 25643 | 14990 | 9523 | 5710 | 6184 | 18561 | 32803 | 29948 | 26780 | ... | 9691 | 13401 | 8263 | 10695 | 11 |
| 2 | Arizona | 246866 | 143488 | 100233 | 53533 | 51827 | 150589 | 266718 | 244743 | 228667 | ... | 84679 | 117609 | 76146 | 108746 | 155 |
| 3 | Arkansas | 103012 | 60923 | 41277 | 24123 | 20634 | 58783 | 102338 | 95817 | 94009 | ... | 37341 | 51855 | 32070 | 46614 | 59 |
| 4 | California | 1328272 | 774841 | 525740 | 285907 | 281350 | 821103 | 1594429 | 1505530 | 1377089 | ... | 470895 | 618484 | 378972 | 499096 | 643 |

5 rows × 25 columns

In [12]:
```
df_male3= df_male2[['state_name','totalAge19_64Male']]


df_male3.rename_axis(index='AgeCategoria', columns='attributes')
df_male3.head(5)
```

Out[12]:

| Indice | state_name | totalAge19_64Male |
|---|---|---|
| 0 | Alabama | 1447891 |
| 1 | Alaska | 246907 |
| 2 | Arizona | 2147201 |
| 3 | Arkansas | 892250 |
| 4 | California | 12472741 |

## Hacemos el mismo procedimiento pora la tabla Female

In [13]:

```
#filtramos los datos para tener solamente 3 columnas , la columna estado nos permite relacionarla con las otras
df_female1= df_female[['AgeCategoria','variable','value']]
#cambiamos el nombre de las columnas
df_female1.columns = ['AgeCategoria','state_name', 'PoblacionFemale']
#df_female1.head(10)
# Debemos cabiarle el nombre a AgeCategoria para que sea el Index y el hacer el pivot quede una columna con ese
df_female1.columns = ['Indice','state_name','PoblacionFemale']

df_female2 = df_female1.pivot_table('PoblacionFemale', ['state_name'], 'Indice')
#df_female2.head(5)
df_female2= pd.DataFrame(df_female2)
df_female2.reset_index(inplace=True, drop=False)
df_female2['totalAge19_64Female'] = df_female2['18 and 19 years'] + df_female2[ '20 years']+ df_female2[ '21 ye
#df_male2.head(5)
df_female3= df_female2[['state_name','totalAge19_64Female']]

df_female3.rename_axis(index='AgeCategoria', columns='attributes')
df_female3.head(5)
```

Out[13]:

| Indice | state_name | totalAge19_64Female |
|---|---|---|
| 0 | Alabama | 1524949 |
| 1 | Alaska | 219095 |
| 2 | Arizona | 2125014 |
| 3 | Arkansas | 906674 |
| 4 | California | 12272144 |

## Unimos las tablas Male y Female por la columna state_Name y tenemos la tabla poblcion /Male /Female por Estados

In [14]:

```
PoblacionMaleFemale19_64 = df_male3.merge(df_female3, on="state_name", how="left")
PoblacionMaleFemale19_64.head(10)
```

Out[14]:

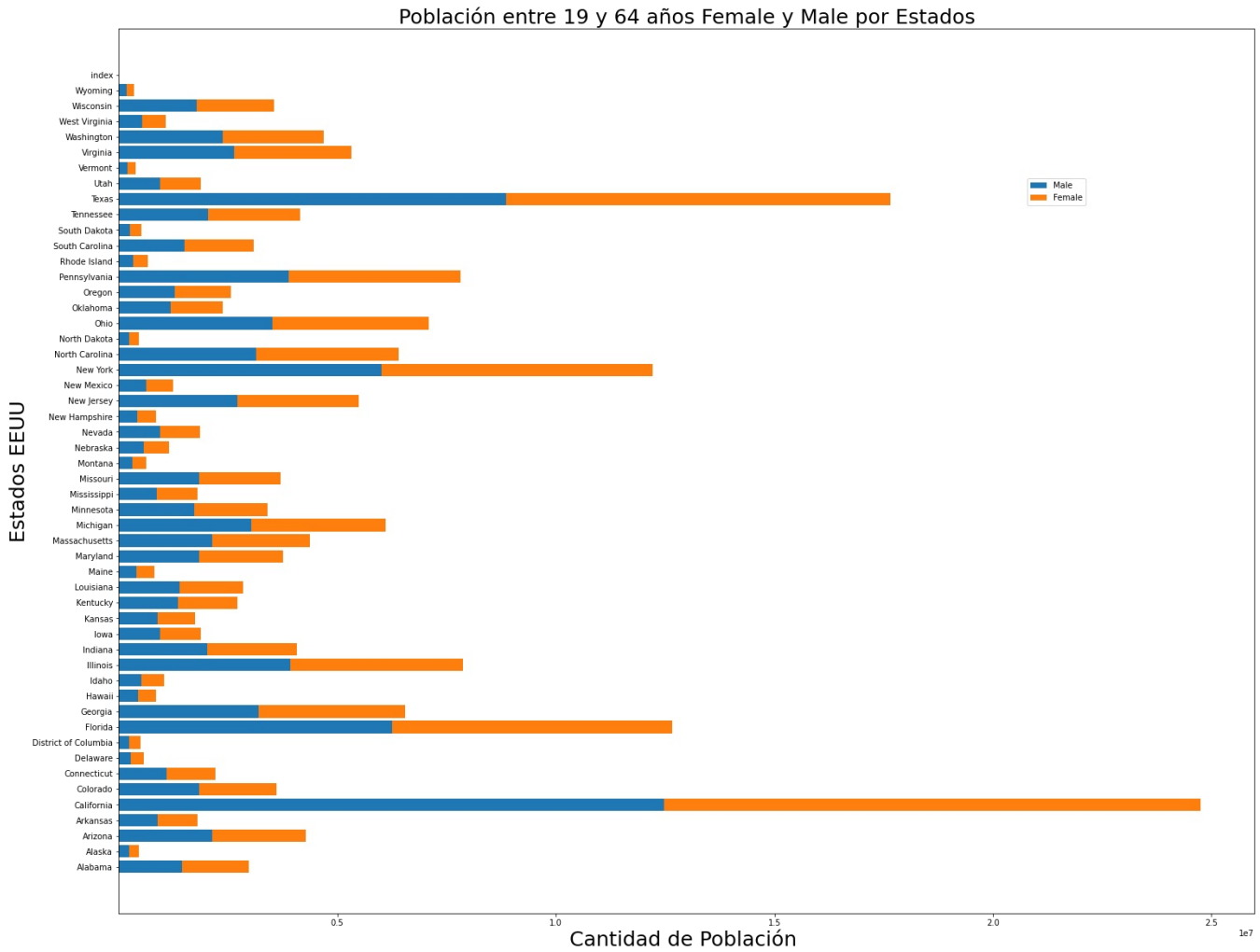
| Indice | state_name | totalAge19_64Male | totalAge19_64Female |
|---|---|---|---|
| 0 | Alabama | 1447891 | 1524949 |
| 1 | Alaska | 246907 | 219095 |
| 2 | Arizona | 2147201 | 2125014 |
| 3 | Arkansas | 892250 | 906674 |
| 4 | California | 12472741 | 12272144 |
| 5 | Colorado | 1848280 | 1769278 |
| 6 | Connecticut | 1097742 | 1121757 |
| 7 | Delaware | 283119 | 298329 |
| 8 | District of Columbia | 234225 | 255789 |
| 9 | Florida | 6261482 | 6393086 |

## Hacemos los graficos de Barra de la Población

In [29]:

```
import numpy as np
import matplotlib.pyplot as plt


plt.rcParams["figure.figsize"] = (25, 20)


y = PoblacionMaleFemale19_64.state_name
x1 = PoblacionMaleFemale19_64.totalAge19_64Male
x2 =PoblacionMaleFemale19_64.totalAge19_64Female


fig, ax = plt.subplots()
ax.barh(y, x1, label = "Male")
```

```
ax.barh(y,x2, left =x1, label= "Female")
#ax.barh(y,x3,left= x1+x2,label ="Muertes")
ax.legend(loc = (0.8, 0.8))

plt.xlabel("Cantidad de Población ",fontsize= 25)
plt.ylabel('Estados EEUU',fontsize= 25)
plt.title('Población entre 19 y 64 años Female y Male por Estados',fontsize=25)


plt.show()
```




## Buscamos la poblacion sin seguro

https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Faspe.hhs.gov%2Fsites%2Fdefault%2Ffiles%2Fmigrated_legacy_files%2F%uninsured-estimates-by-state.xlsx&wdOrigin=BROWSELINK

In [16]:

```
dfpoblacionsinseguro = pd.read_excel('D:/d/2-ESTUDIO/Proyecto Data Engineer/aspe-uninsured-estimates-by-state.x

dfpoblacionsinseguro.head(10)
```

Out[16]:

| | State Name | Total Non-Elderly Population (Excluding Undocumented) | Uninsured Population (Excluding Undocumented) | Percent Uninsured | % HIU Income < 100% FPL | % HIU Income 100-138% FPL | % HIU Income 139-249% FPL | % HIU Income 250-400% FPL | % HIU Income 400% FPL | % Age 0-18 | ... | % College Grad | % No English Speaking Adults in HH | % English Spoken in HH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Alabama | 3986500 | 456100 | 0.11 | 0.47 | 0.11 | 0.21 | 0.13 | 0.09 | 0.08 | ... | 0.08 | 0.02 | 0.97 |
| 1 | Alaska | 629300 | 80300 | 0.13 | 0.26 | 0.10 | 0.28 | 0.22 | 0.14 | 0.17 | ... | 0.13 | 0.01 | 0.97 |
| 2 | Arizona | 5720700 | 698800 | 0.12 | 0.32 | 0.09 | 0.28 | 0.19 | 0.13 | 0.21 | ... | 0.15 | 0.07 | 0.90 |
| 3 | Arkansas | 2443900 | 257200 | 0.11 | 0.34 | 0.11 | 0.31 | 0.15 | 0.08 | 0.15 | ... | 0.07 | 0.04 | 0.95 |
| 4 | California | 32060700 | 2397600 | 0.07 | 0.30 | 0.08 | 0.26 | 0.19 | 0.16 | 0.14 | ... | 0.17 | 0.14 | 0.75 |
| 5 | Colorado | 4758400 | 392300 | 0.08 | 0.23 | 0.07 | 0.26 | 0.23 | 0.21 | 0.16 | ... | 0.21 | 0.07 | 0.91 |
| 6 | Connecticut | 2795000 | 142200 | 0.05 | 0.22 | 0.07 | 0.25 | 0.21 | 0.25 | 0.15 | ... | 0.23 | 0.1 | 0.85 |
| 7 | Delaware | 757200 | 57800 | 0.08 | 0.32 | 0.07 | 0.28 | 0.18 | 0.15 | 0.15 | ... | 0.19 | 0.05 | 0.91 |
| 8 | District of Columbia | 601300 | 22100 | 0.04 | 0.41 | 0.07 | 0.20 | 0.12 | 0.20 | 0.12 | ... | 0.29 | 0.03 | 0.97 |
| 9 | Florida | 16240100 | 2519500 | 0.16 | 0.35 | 0.09 | 0.26 | 0.17 | 0.13 | 0.12 | ... | 0.16 | 0.11 | 0.84 |

10 rows × 47 columns

## Vemos todas las columnas para saber con cual nos quedaremos

In [17]:
```
columns_names = dfpoblacionsinseguro.columns.values
columns_names
```

Out[17]:
```
array(['State Name',
       'Total Non-Elderly Population (Excluding Undocumented)',
       'Uninsured Population (Excluding Undocumented)',
       'Percent Uninsured', '% HIU Income < 100% FPL',
       '% HIU Income 100-138% FPL', '% HIU Income 139-249% FPL',
       '% HIU Income 250-400% FPL', '% HIU Income 400% FPL', '% Age 0-18',
       '% Age 19-34', '% Age 35-49', '% Age 50-64', '% Male', '% Female',
       '% Married', '% Child in Family',
       '% Spanish/Hispanic/Latino Origin', '% White Non-Latino',
       '% Black Non-Latino', '% Asian / Native-Hawaiian / Pac Islander',
       '% American Indian / Alaska Native', '% Multi-racial or Other',
       '% SNAP Recipient', '% With a disability',
       '% Full-time Worker in Family',
       '% Employed in Agriculture Industry',
       '% Employed in Mining/Construction Industry',
       '% Employed in Manufacturing Industry',
       '% Employed in Trade Industry',
       '% Employed in Info/Finance Industry',
       '% Employed in Education/Health Industry',
       '% Employed in Entertainment Industry',
       '% Employed in Service Industry',
       '% Employed in Military/Public Industry',
       '% Less than High School', '% High School Diploma',
       '% College Grad', '% No English Speaking Adults in HH',
       '% English Spoken in HH', '% Spanish Spoken in HH',
       '% Chinese Spoken in HH', '% Korean Spoken in HH',
       '% Vietnamese Spoken in HH', '% Tagalog Spoken in HH',
       '% Russian Spoken in HH', '% Other Language Spoken in HH'],
      dtype=object)
```

In [18]:
```
dfpoblacionsinseguro=dfpoblacionsinseguro[['State Name','Uninsured Population (Excluding Undocumented)','% Age
dfpoblacionsinseguro.head(10)
```

Out[18]:

| | State Name | Uninsured Population (Excluding Undocumented) | % Age 19-34 | % Age 35-49 | % Age 50-64 | % Male | % Female |
|---|---|---|---|---|---|---|---|
| 0 | Alabama | 456100 | 0.41 | 0.28 | 0.22 | 0.56 | 0.44 |
| 1 | Alaska | 80300 | 0.47 | 0.20 | 0.16 | 0.62 | 0.38 |
| 2 | Arizona | 698800 | 0.36 | 0.24 | 0.19 | 0.57 | 0.43 |
| 3 | Arkansas | 257200 | 0.39 | 0.28 | 0.19 | 0.58 | 0.42 |
| 4 | California | 2397600 | 0.40 | 0.26 | 0.20 | 0.57 | 0.43 |
| 5 | Colorado | 392300 | 0.40 | 0.26 | 0.17 | 0.58 | 0.42 |
| 6 | Connecticut | 142200 | 0.37 | 0.26 | 0.23 | 0.61 | 0.39 |
| 7 | Delaware | 57800 | 0.43 | 0.21 | 0.22 | 0.63 | 0.37 |
| 8 | District of Columbia | 22100 | 0.50 | 0.24 | 0.14 | 0.70 | 0.30 |
| 9 | Florida | 2519500 | 0.36 | 0.28 | 0.23 | 0.56 | 0.44 |

In [46]:
```
dfpoblacionsinseguro['totalsinseguo19_64'] = dfpoblacionsinseguro['Uninsured Population (Excluding Undocumented

dfpoblacionsinseguro1= dfpoblacionsinseguro[['State Name','totalsinseguo19_64']]
```

```
dfpoblacionsinseguro1.head(10)
```

Out[46]:

| | State Name | totalsinseguo19_64 |
|---|---|---|
| 0 | Alabama | 4.150510e+05 |
| 1 | Alaska | 6.664900e+04 |
| 2 | Arizona | 5.520520e+05 |
| 3 | Arkansas | 2.211920e+05 |
| 4 | California | 2.061936e+06 |
| 5 | Colorado | 3.256090e+05 |
| 6 | Connecticut | 1.222920e+05 |
| 7 | Delaware | 4.970800e+04 |
| 8 | District of Columbia | 1.944800e+04 |
| 9 | Florida | 2.191965e+06 |

In [47]:

```
# Cambiamos el nombre de las columnas
dfpoblacionsinseguro1.columns = ['state_name', 'PoblacionsinSeguro19_64']
dfpoblacionsinseguro1['state_name'] = dfpoblacionsinseguro1['state_name'].str.upper()
dfpoblacionsinseguro1.head(10)
```

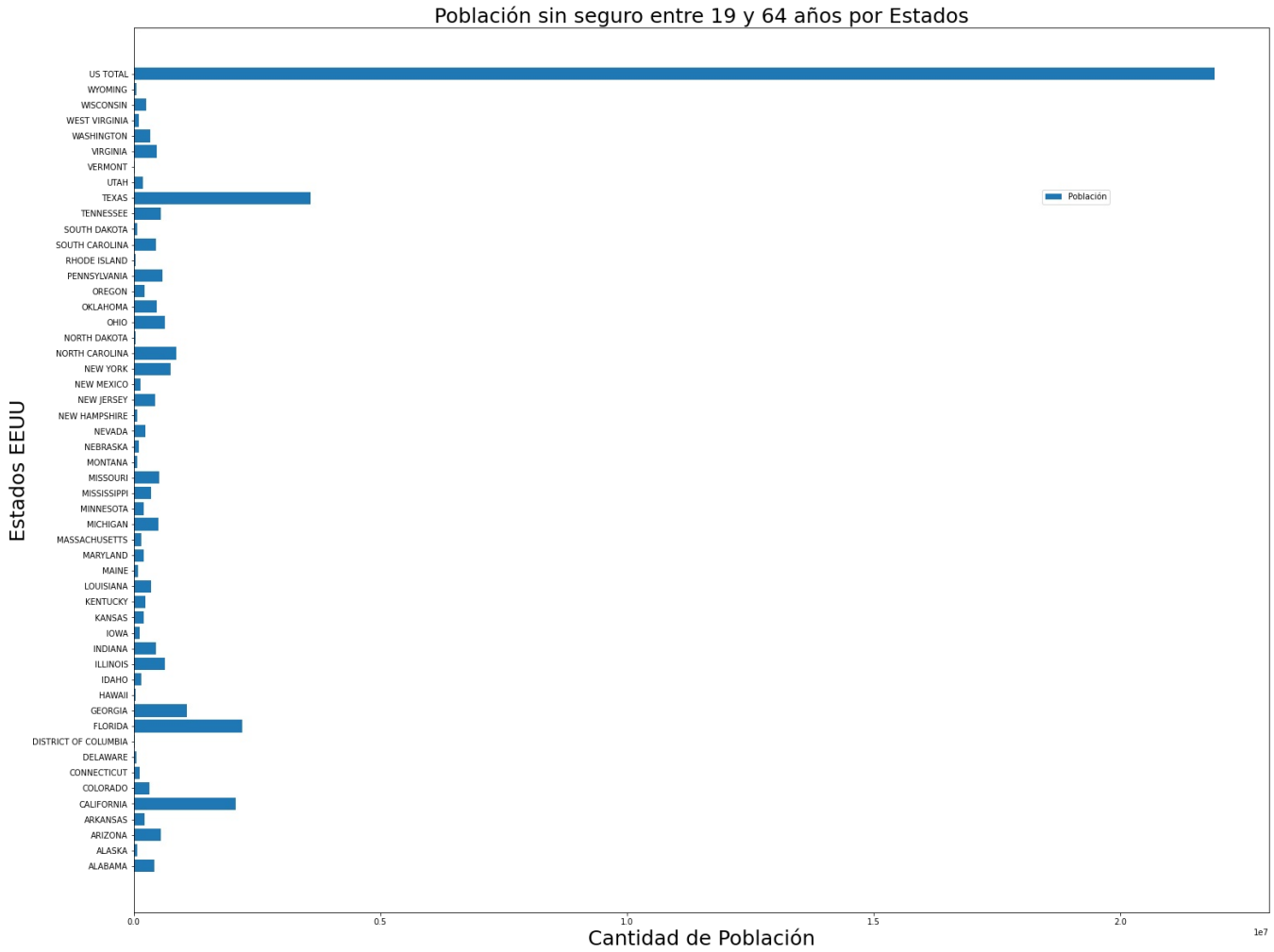
```
C:\Users\Gabriel\AppData\Local\Temp\ipykernel_3160\1155097750.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy
  dfpoblacionsinseguro1['state_name'] = dfpoblacionsinseguro1['state_name'].str.upper()
```

Out[47]:

| | state_name | PoblacionsinSeguro19_64 |
|---|---|---|
| 0 | ALABAMA | 4.150510e+05 |
| 1 | ALASKA | 6.664900e+04 |
| 2 | ARIZONA | 5.520520e+05 |
| 3 | ARKANSAS | 2.211920e+05 |
| 4 | CALIFORNIA | 2.061936e+06 |
| 5 | COLORADO | 3.256090e+05 |
| 6 | CONNECTICUT | 1.222920e+05 |
| 7 | DELAWARE | 4.970800e+04 |
| 8 | DISTRICT OF COLUMBIA | 1.944800e+04 |
| 9 | FLORIDA | 2.191965e+06 |

In [48]:

```
import numpy as np
import matplotlib.pyplot as plt


plt.rcParams["figure.figsize"] = (25, 20)


y = dfpoblacionsinseguro1.state_name
x1 = dfpoblacionsinseguro1.PoblacionsinSeguro19_64



fig, ax = plt.subplots()
ax.barh(y, x1, label = "Población")
#ax.barh(y,x2, left =x1, label= "Female")
#ax.barh(y,x3,left= x1+x2,label ="Muertes")
ax.legend(loc = (0.8, 0.8))

plt.xlabel("Cantidad de Población",fontsize= 25)
plt.ylabel('Estados EEUU',fontsize= 25)
plt.title('Población sin seguro entre 19 y 64 años por Estados',fontsize=25)


plt.show()
```




Poblacion de Covid 19 exctraemos los datos de cloud data publica de covid luego los pasamos a Data Estudio y esportamos a a csv . ( podemos utilizar otra tabla de preferencia )

In [31]:
```
# https://console.cloud.google.com/bigquery?p=bigquery-public-data&d=covid19_public_forecasts&page=table&_ga=2.

dfCovid = pd.read_csv('D:/d/2-ESTUDIO/Proyecto Data Engineer/forecast_STATE_28.csv')
dfCovid.head(10)
```

Out[31]:

| | state_fips_code | state_name | target_prediction_date | cumulative_confirmed | cumulative_confirmed_q0025 | cumulative_confirmed_q0975 | cumu |
|---|---|---|---|---|---|---|---|
| **0** | 1 | ALABAMA | 2022-02-06 | 1246468.250 | 1242645.500 | 1260607.375 | |
| **1** | 1 | ALABAMA | 2022-02-07 | 1250614.000 | 1243337.625 | 1277008.000 | |
| **2** | 1 | ALABAMA | 2022-02-08 | 1255275.375 | 1244869.750 | 1292367.125 | |
| **3** | 1 | ALABAMA | 2022-02-09 | 1259398.625 | 1246166.875 | 1305712.375 | |
| **4** | 1 | ALABAMA | 2022-02-10 | 1263124.000 | 1247341.375 | 1317269.875 | |
| **5** | 1 | ALABAMA | 2022-02-11 | 1266656.500 | 1248570.000 | 1327538.375 | |
| **6** | 1 | ALABAMA | 2022-02-12 | 1269949.875 | 1249783.625 | 1336528.875 | |
| **7** | 1 | ALABAMA | 2022-02-13 | 1272921.000 | 1250880.375 | 1344345.750 | |
| **8** | 1 | ALABAMA | 2022-02-14 | 1276072.375 | 1252330.875 | 1351725.000 | |
| **9** | 1 | ALABAMA | 2022-02-15 | 1280008.375 | 1254702.625 | 1359770.250 | |

10 rows × 32 columns

In [32]:
```
# selecionamos las columnas
dfCovid = dfCovid[['state_name', 'target_prediction_date','cumulative_confirmed', 'cumulative_deaths', 'hospita

dfCovid.head(10)
```

Out[32]:

| | state_name | target_prediction_date | cumulative_confirmed | cumulative_deaths | hospitalized_patients |
|---|---|---|---|---|---|
| 0 | ALABAMA | 2022-02-06 | 1246468.250 | 17403.625000 | 2674.573730 |
| 1 | ALABAMA | 2022-02-07 | 1250614.000 | 17440.130859 | 2648.798828 |
| 2 | ALABAMA | 2022-02-08 | 1255275.375 | 17481.675781 | 2619.628174 |
| 3 | ALABAMA | 2022-02-09 | 1259398.625 | 17523.759766 | 2580.830566 |
| 4 | ALABAMA | 2022-02-10 | 1263124.000 | 17564.027344 | 2537.801025 |
| 5 | ALABAMA | 2022-02-11 | 1266656.500 | 17597.248047 | 2492.255371 |
| 6 | ALABAMA | 2022-02-12 | 1269949.875 | 17626.558594 | 2454.183105 |
| 7 | ALABAMA | 2022-02-13 | 1272921.000 | 17652.808594 | 2432.121094 |
| 8 | ALABAMA | 2022-02-14 | 1276072.375 | 17681.896484 | 2408.961182 |
| 9 | ALABAMA | 2022-02-15 | 1280008.375 | 17714.740234 | 2382.829834 |

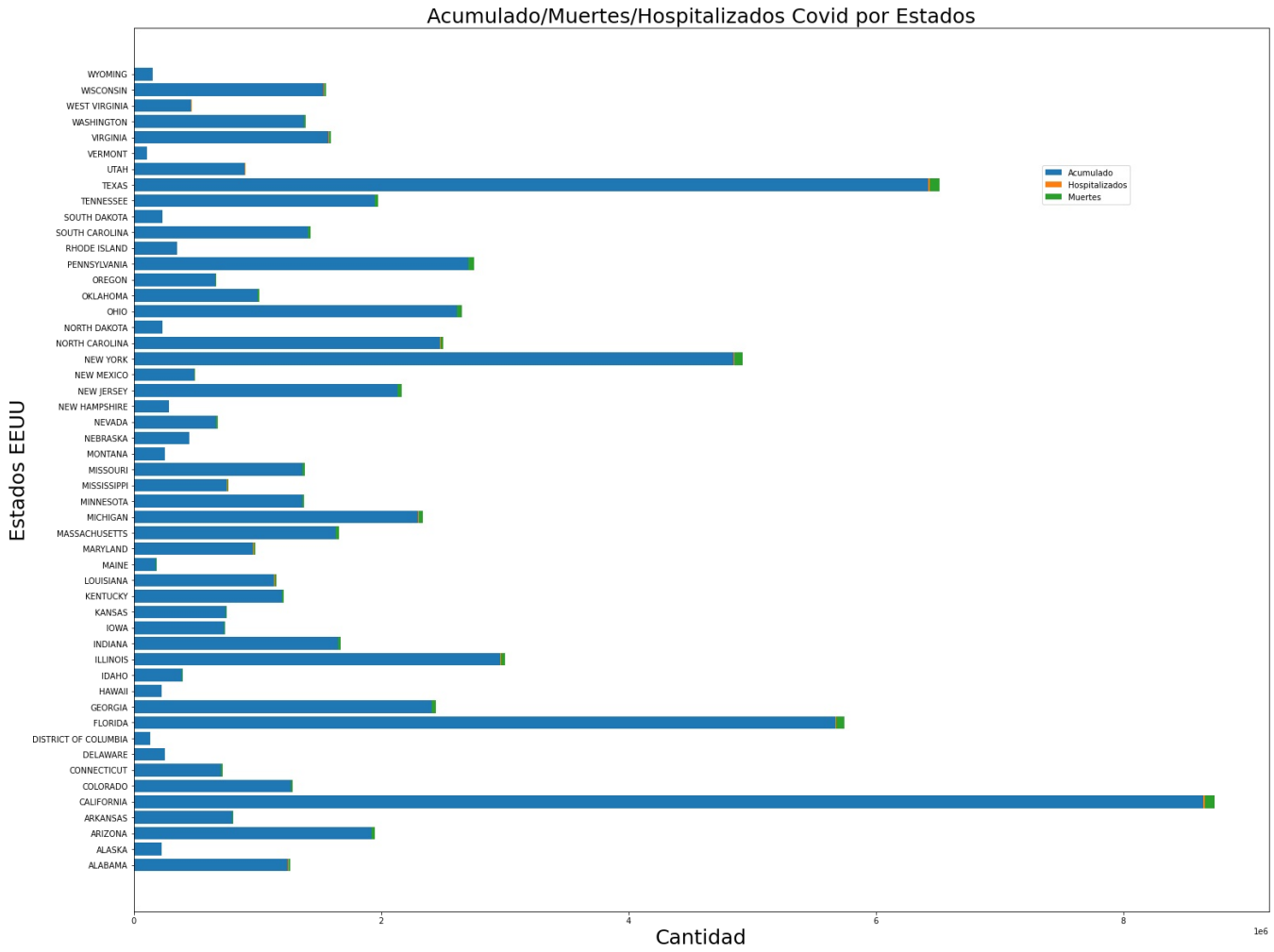
In [41]:
```
from datetime import datetime
import matplotlib.pyplot as plt


#dfdata= dfCovid.DataReader("state_name", 'cumulative_confirmed', 'cumulative_deaths', 'hospitalized_patients',

filtered_df=dfCovid.query("target_prediction_date == '2022-02-06'")
filtered_df.head(10)
```

Out[41]:

| | state_name | target_prediction_date | cumulative_confirmed | cumulative_deaths | hospitalized_patients |
|---|---|---|---|---|---|
| 0 | ALABAMA | 2022-02-06 | 1.246468e+06 | 17403.625000 | 2674.573730 |
| 28 | ALASKA | 2022-02-06 | 2.259794e+05 | 1095.842651 | 152.206497 |
| 56 | ARIZONA | 2022-02-06 | 1.919820e+06 | 26681.306641 | 2864.215332 |
| 84 | ARKANSAS | 2022-02-06 | 7.946482e+05 | 9809.074219 | 1566.971558 |
| 112 | CALIFORNIA | 2022-02-06 | 8.646250e+06 | 81083.695312 | 11633.527344 |
| 140 | COLORADO | 2022-02-06 | 1.272055e+06 | 11220.957031 | 1165.098267 |
| 168 | CONNECTICUT | 2022-02-06 | 7.070663e+05 | 10172.240234 | 718.194519 |
| 196 | DELAWARE | 2022-02-06 | 2.512354e+05 | 2579.324951 | 332.504211 |
| 224 | DISTRICT OF COLUMBIA | 2022-02-06 | 1.316892e+05 | 1295.201294 | 353.009064 |
| 252 | FLORIDA | 2022-02-06 | 5.672004e+06 | 66378.140625 | 7090.622070 |

## Graficamos los valores de Covid a la fecha 06-02-2022 por cada estado

In [43]:
```
import numpy as np
import matplotlib.pyplot as plt



y = filtered_df.state_name
x1 = filtered_df.cumulative_confirmed
x2 =filtered_df.hospitalized_patients
x3 = filtered_df.cumulative_deaths



fig, ax = plt.subplots()
ax.barh(y, x1, label = "Acumulado")
ax.barh(y,x2, left =x1, label= "Hospitalizados")
ax.barh(y,x3,left= x1+x2,label ="Muertes")
ax.legend(loc = (0.8, 0.8))

plt.xlabel("Cantidad",fontsize= 25)
plt.ylabel('Estados EEUU',fontsize= 25)
plt.title('Acumulado/Muertes/Hospitalizados Covid por Estados',fontsize=25)


plt.show()
```




## Unimos las Datos de Covid y de población noasegurada para su representación.

In [49]:
```
PoblacionSinSeguroyCovid = dfpoblacionsinseguro1.merge(filtered_df , on="state_name", how="left")
PoblacionSinSeguroyCovid.head(10)
```

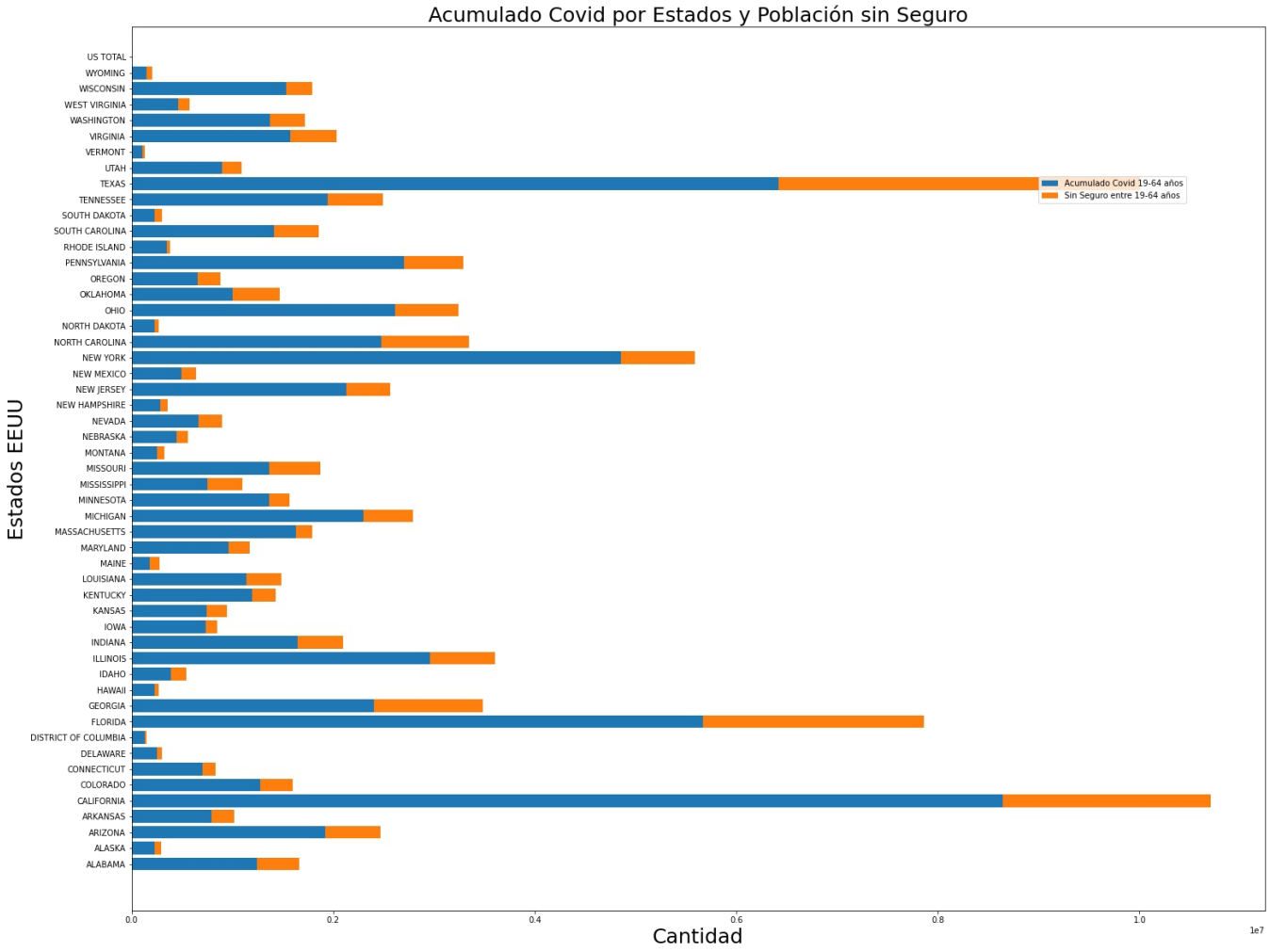
Out[49]:

| | state_name | PoblacionsinSeguro19_64 | target_prediction_date | cumulative_confirmed | cumulative_deaths | hospitalized_patients |
|---|---|---|---|---|---|---|
| 0 | ALABAMA | 4.150510e+05 | 2022-02-06 | 1.246468e+06 | 17403.625000 | 2674.573730 |
| 1 | ALASKA | 6.664900e+04 | 2022-02-06 | 2.259794e+05 | 1095.842651 | 152.206497 |
| 2 | ARIZONA | 5.520520e+05 | 2022-02-06 | 1.919820e+06 | 26681.306641 | 2864.215332 |
| 3 | ARKANSAS | 2.211920e+05 | 2022-02-06 | 7.946482e+05 | 9809.074219 | 1566.971558 |
| 4 | CALIFORNIA | 2.061936e+06 | 2022-02-06 | 8.646250e+06 | 81083.695312 | 11633.527344 |
| 5 | COLORADO | 3.256090e+05 | 2022-02-06 | 1.272055e+06 | 11220.957031 | 1165.098267 |
| 6 | CONNECTICUT | 1.222920e+05 | 2022-02-06 | 7.070663e+05 | 10172.240234 | 718.194519 |
| 7 | DELAWARE | 4.970800e+04 | 2022-02-06 | 2.512354e+05 | 2579.324951 | 332.504211 |
| 8 | DISTRICT OF COLUMBIA | 1.944800e+04 | 2022-02-06 | 1.316892e+05 | 1295.201294 | 353.009064 |
| 9 | FLORIDA | 2.191965e+06 | 2022-02-06 | 5.672004e+06 | 66378.140625 | 7090.622070 |

## Graficamos los valores de población sin Seguro y casos de Covid

In [51]:
```
y = PoblacionSinSeguroyCovid.state_name
x1 = PoblacionSinSeguroyCovid.cumulative_confirmed
x2 =PoblacionSinSeguroyCovid.PoblacionsinSeguro19_64
#x3 = filtered_df.cumulative_deaths



fig, ax = plt.subplots()
ax.barh(y, x1, label = "Acumulado Covid 19-64 años")
ax.barh(y,x2, left =x1, label= "Sin Seguro entre 19-64 años")
#ax.barh(y,x3,left= x1+x2,label ="Muertes")
ax.legend(loc = (0.8, 0.8))

plt.xlabel("Cantidad",fontsize= 25)
plt.ylabel('Estados EEUU',fontsize= 25)
plt.title('Acumulado Covid por Estados y Población sin Seguro',fontsize=25)
```

```
plt.show()
```




Los estados que se recomienda la venta de seguros por su alta población acumulada de pacientes de Covis y una gran cantidad de no asegurados son : Californis, Texas, Florida, New York

In [ ]: