




RETO GRUPAL

ESTADOS DE ESTADOS UNIDOS

PRIMERA PARTE

ESCUELA DE TALENTO | FUNDACIÓN NTTDATA



ÁREA I

FUNDAMENTOS DE PYTHON



METODOLOGÍA

En este documento presentamos nuestro trabajo correspondiente a la resolución de los ejercicios del primer Reto Grupal “Estados de Estados Unidos”.

Para la realización de las tareas descritas en el reto hemos realizado conferencias diarias para debatir cada detalle y compartir nuestras propuestas. Asimismo, hemos utilizado plataformas para compartir archivos para que en todo momento todos los integrantes pudiéramos tener acceso a la evolución del trabajo, aportar nuestras opiniones, anotar sugerencias etc, cuando no estábamos conectados.

El resultado final se resume en este trabajo con el que hemos aprendido tanto. Muchas gracias a nuestro profesor Roberto Doncel por habernos facilitado los conocimientos necesarios a través de sus clases y aclaraciones durante el Área I.



PARTICIPANTES

- LUISA
- ANA
- JAIME
- JESÉ
- SERGIO



INDICE

EJERCICIO 1: Diccionarios, listas y claves

- a. Creando listas que contengan un diccionario.
- b. Transformando listas en diccionarios.
- c. Actualizando datos incorrectos.
- d. Introduciendo el nombre de una nueva clave.
- e. Calculando una nueva clave.
- d. Desarrollando un programa.

EJERCICIO 2: Calcular ratios

EJERCICIO 3: Funciones

EJERCICIO 4: Librería Follium

EJERCICIO 1

Comenzamos el **apartado 1.a** creando una lista con los datos detallados en el reto. Corregimos el error indicado en la tabla correspondiente a “Población 2020” y “Población 2021”, ya que los datos correctos se corresponden con los años 2000 y 2001.

En este diccionario de forma unánime hemos elegido las claves “Estado”, “Población 2000”, “Población 2001”, “Residentes menores de 65 años 2000”, “Residentes menores de 65 años 2001”, “Muertes 2000”, “Muertes 2001”, “Fecha de fundación del estado”, “Latitud” y “Longitud”.

A continuación, mostramos el fragmento de código correspondiente a la creación de la lista una vez finalizado, realizamos un print para comprobar que estaba correcto.

```
lista_estados = [
    {
        'Estado': "Alabama",
        'Poblacion 2000': 4447100,
        'Poblacion 2001': 4451493,
        'Residentes menores de 65 años 2000': 3870598,
        'Residentes menores de 65 años 2001': 3880476,
        'Muertes 2000': 10622,
        'Muertes 2001': 15912,
        'Fecha de fundacion del Estado': '14-12-1819',
        'Latitud': 33.258882,
        'Longitud': -86.829534
    },
    {
        'Estado': "Florida",
        'Poblacion 2000': 15982378,
        'Poblacion 2001': 17054000,
        'Residentes menores de 65 años 2000': 13237167,
        'Residentes menores de 65 años 2001': 13548077,
        'Muertes 2000': 38103,
        'Muertes 2001': 166069,
        'Fecha de fundacion del Estado': '03-03-1845',
        'Latitud': 27.756767,
        'Longitud': -81.463983
    },
    {
        'Estado': "Georgia",
        'Poblacion 2000': 8186453,
        'Poblacion 2001': 8229823,
        'Residentes menores de 65 años 2000': 7440877,
        'Residentes menores de 65 años 2001': 7582146,
        'Muertes 2000': 14804,
        'Muertes 2001': 15000,
        'Fecha de fundacion del Estado': '12-02-1733',
        'Latitud': 32.329381,
        'Longitud': -83.113737
    },
    {
        'Estado': "South Carolina",
        'Poblacion 2000': 4012012,
        'Poblacion 2001': 4023438,
        'Residentes menores de 65 años 2000': 3535770,
        'Residentes menores de 65 años 2001': 3567172,
        'Muertes 2000': 8581,
        'Muertes 2001': 9500,
        'Fecha de fundacion del Estado': '26-03-1776',
        'Latitud': 33.687439,
        'Longitud': -80.436374
    }
]

print (lista_estados)
```

```
[{'Estado': 'Alabama', 'Poblacion 2000': 4447100, 'Poblacion 2001': 4451493, 'Residentes menores de 65 años 2000': 3870598, 'Residentes menores de 65 años 2001': 3880476, 'Muertes 2000': 10622, 'Muertes 2001': 15912, 'Fecha de fundacion del Estado': '14-12-1819', 'Latitud': 33.258882, 'Longitud': -86.829534}, {'Estado': 'Florida', 'Poblacion 2000': 15982378, 'Poblacion 2001': 17054000, 'Residentes menores de 65 años 2000': 13237167, 'Residentes menores de 65 años 2001': 13548077, 'Muertes 2000': 38103, 'Muertes 2001': 166069, 'Fecha de fundacion del Estado': '03-03-1845', 'Latitud': 27.756767, 'Longitud': -81.463983}, {'Estado': 'Georgia', 'Poblacion 2000': 8186453, 'Poblacion 2001': 8229823, 'Residentes menores de 65 años 2000': 7440877, 'Residentes menores de 65 años 2001': 7582146, 'Muertes 2000': 14804, 'Muertes 2001': 15000, 'Fecha de fundacion del Estado': '12-02-1733', 'Latitud': 32.329381, 'Longitud': -83.113737}, {'Estado': 'South Carolina', 'Poblacion 2000': 4012012, 'Poblacion 2001': 4023438, 'Residentes menores de 65 años 2000': 3535770, 'Residentes menores de 65 años 2001': 3567172, 'Muertes 2000': 8581, 'Muertes 2001': 9500, 'Fecha de fundacion del Estado': '26-03-1776', 'Latitud': 33.687439, 'Longitud': -80.436374}]
```

En el **apartado 1.b** se solicita que transformemos la lista inicial en tantos diccionarios como Estados tengamos con una estructura determinada en la que el nombre de estado sea el nombre dinámico de cada uno de los estados.

Ana propone estas opciones:

- Sin el uso del zip:

```
diccionarios_estados = {}
for estado in lista_estados:
    diccionario_estado = {}
    for clave, valor in estado.items():
        diccionario_estado[f"{clave}_{estado['Estado']}"] = valor
    diccionarios_estados[f"Diccionario_{estado['Estado']}"] = diccionario_estado
print(diccionarios_estados)
```

- Con zip:

```
diccionarios_estados = {}
for estado in lista_estados:
    diccionario_estado = dict(zip([f"{clave}_{estado['Estado']}" for clave in estado.keys]
    ))
    diccionarios_estados[f"Diccionario_{estado['Estado']}"] = diccionario_estado
print(diccionarios_estados)
```

El resultado del print:

```
{'Diccionario_Alabama': {'Estado_Alabama': 'Alabama', 'Poblacion 2000_Alabama': 4447100, 'Poblacion 2001_Alabama': 4451493, 'Residentes menores de 65 años 2000_Alabama': 3870598, 'Residentes menores de 65 años 2001_Alabama': 3880476, 'Muertes 2000_Alabama': 10622, 'Muertes 2001_Alabama': 15912, 'Fecha de fundacion del Estado_Alabama': '14-12-1819', 'Latitud_Alabama': 33.258882, 'Longitud_Alabama': -86.829534}, 'Diccionario_Florida': {'Estado_Florida': 'Florida', 'Poblacion 2000_Florida': 15982378, 'Poblacion 2001_Florida': 17054000, 'Residentes menores de 65 años 2000_Florida': 13237167, 'Residentes menores de 65 años 2001_Florida': 13548077, 'Muertes 2000_Florida': 38103, 'Muertes 2001_Florida': 166069, 'Fecha de fundacion del Estado_Florida': '03-03-1845', 'Latitud_Florida': 27.756767, 'Longitud_Florida': -81.463983}, 'Diccionario_Georgia': {'Estado_Georgia': 'Georgia', 'Poblacion 2000_Georgia': 8186453, 'Poblacion 2001_Georgia': 8229823, 'Residentes menores de 65 años 2000_Georgia': 7440877, 'Residentes menores de 65 años 2001_Georgia': 7582146, 'Muertes 2000_Georgia': 14804, 'Muertes 2001_Georgia': 15000, 'Fecha de fundacion del Estado_Georgia': '12-02-1733', 'Latitud_Georgia': 32.329381, 'Longitud_Georgia': -83.113737}, 'Diccionario_South Carolina': {'Estado_South Carolina': 'South Carolina', 'Poblacion 2000_South Carolina': 4012012, 'Poblacion 2001_South Carolina': 4023438, 'Residentes menores de 65 años 2000_South Carolina': 3535770, 'Residentes menores de 65 años 2001_South Carolina': 3567172, 'Muertes 2000_South Carolina': 8581, 'Muertes 2001_South Carolina': 9500, 'Fecha de fundacion del Estado_South Carolina': '26-03-1776', 'Latitud_South Carolina': 33.687439, 'Longitud_South Carolina': -80.436374}}
```

El grupo decide utilizar para el ejercicio la opción con el zip, ya que era la indicada en las instrucciones el reto.

En el **apartado 1.c** se nos indica que se ha detectado que los datos de población de Florida en 2001 eran incorrectos y que hay que corregirlo actualizando su valor a: "Población 2001: 16054328".

Luisa realiza la siguiente propuesta:

```
lista_estados[1]['Poblacion 2001'] = 16054328
print (lista_estados)
```



```
[{'Estado': 'Alabama', 'Poblacion 2000': 4447100, 'Poblacion 2001': 4451493, 'Residentes menores de 65 años 2000': 3870598, 'Residentes menores de 65 años 2001': 3880476, 'Muertes 2000': 10622, 'Muertes 2001': 15912, 'Fecha de fundacion del Estado': '14-12-1819', 'Latitud': 33.258882, 'Longitud': -86.829534}, {'Estado': 'Florida', 'Poblacion 2000': 15982378, 'Poblacion 2001': 16054328, 'Residentes menores de 65 años 2000': 13237167, 'Residentes menores de 65 años 2001': 13548077, 'Muertes 2000': 38103, 'Muertes 2001': 166069, 'Fecha de fundacion del Estado': '03-03-1845', 'Latitud': 27.756767, 'Longitud': -81.463983}, {'Estado': 'Georgia', 'Poblacion 2000': 8186453, 'Poblacion 2001': 8229823, 'Residentes menores de 65 años 2000': 7440877, 'Residentes menores de 65 años 2001': 7582146, 'Muertes 2000': 14804, 'Muertes 2001': 15000, 'Fecha de fundacion del Estado': '12-02-1733', 'Latitud': 32.329381, 'Longitud': -83.113737}, {'Estado': 'South Carolina', 'Poblacion 2000': 4012012, 'Poblacion 2001': 4023438, 'Residentes menores de 65 años 2000': 3535770, 'Residentes menores de 65 años 2001': 3567172, 'Muertes 2000': 8581, 'Muertes 2001': 9500, 'Fecha de fundacion del Estado': '26-03-1776', 'Latitud': 33.687439, 'Longitud': -80.436374}]
```

Se valora también la propuesta de Ana, poder realizar el cambio con:

```
for estado in lista_estados:
    if estado["Estado"] == "Florida":
        estado["Poblacion 2001"] = 16054328
```

Pero finalmente se elige la opción indicada por Luisa al ser más clara.

Pasamos al **apartado 1.d**, donde se nos indica que incluyamos una nueva clave “Días desde fundación nombre_estado” que sea el número de días que han pasado desde la fundación del Estado hasta la actualidad, y calculemos el número de meses que han transcurrido.

Para este caso utilizamos datetime, por lo que importamos el módulo e iteramos para calcular el número de días. En un inicio aplicamos el código propuesto por Luisa que es similar al final, pero modificando una parte dado que tanto Jaime como Ana indicaron que quizá no era necesario que se aportara información de días y horas para realizar el cálculo de meses, aunque podría ser más exacto. Tras consenso del grupo modificamos el código y añadimos la parte del cálculo de los meses haciendo una división del número de días obtenido entre 30.44 que es la media de días que tiene un mes:

```
from datetime import datetime

# Iteramos sobre la lista de estados
for estado in lista_estados:
    fecha_fundacion = datetime.strptime(estado['Fecha de fundacion del Estado'], '%d-%m-%Y')

    # Calculamos la cantidad de días
    dias_desde_fundacion = (datetime.today() - fecha_fundacion).days

    # Agregamos la nueva clave
    estado["Días desde fundación " + estado["Estado"]] = dias_desde_fundacion

    meses_desde_fundacion = int(dias_desde_fundacion / 30.44)
    print(f"El número de meses desde la fundación de {estado['Estado']} hasta hoy es: {meses_desde_fundacion}")
```

```
El número de meses desde la fundación de Alabama hasta hoy es: 2439
El número de meses desde la fundación de Florida hasta hoy es: 2137
El número de meses desde la fundación de Georgia hasta hoy es: 3481
El número de meses desde la fundación de South Carolina hasta hoy es: 2964
```

Pasamos al **apartado 1.e** en el que se pide crear una nueva clave del diccionario que sea “Porcentaje mayores de 65 años nombre_estado”, para realizarlo pensamos que sería buena opción calcular la población mayor de 65 años restando la población menor de 65 años al total de población del estado.

Luego, calculamos el porcentaje de población mayor de 65 años dividiendo la población mayor de 65 años por la población total y multiplicando por 100. Finalmente, agregamos la nueva clave al diccionario del estado con el porcentaje de población mayor de 65 años.

El código fue propuesto tanto por Luisa como por Ana y revisado por Jese y consensuado por el resto de compañeros, quedando de la siguiente manera.

```
for estado in lista_estados:
    # Obtenemos la población mayor y menor de 65 años
    poblacion_mayores_65 = estado["Poblacion 2001"] - estado["Residentes menores de 65 años 2001"]

    # Calculamos el porcentaje de población mayor de 65 años
    porcentaje_mayores_65 = (poblacion_mayores_65 / estado["Poblacion 2001"]) * 100

    # Agregamos la nueva clave al diccionario del estado
    estado["Porcentaje mayores de 65 años " + estado["Estado"]] = porcentaje_mayores_65

print (lista_estados)
```

```
[{'Estado': 'Alabama', 'Poblacion 2000': 4447100, 'Poblacion 2001': 4451493, 'Residentes menores de 65 años 2000': 3870598, 'Residentes menores de 65 años 2001': 3880476, 'Muertes 2000': 10622, 'Muertes 2001': 15912, 'Fecha de fundacion del Estado': '14-12-1819', 'Latitud': 33.258882, 'Longitud': -86.829534, 'Porcentaje mayores de 65 años Alabama': 12.82753898523484}, {'Estado': 'Florida', 'Poblacion 2000': 15982378, 'Poblacion 2001': 17054000, 'Residentes menores de 65 años 2000': 13237167, 'Residentes menores de 65 años 2001': 13548077, 'Muertes 2000': 38103, 'Muertes 2001': 166069, 'Fecha de fundacion del Estado': '03-03-1845', 'Latitud': 27.756767, 'Longitud': -81.463983, 'Porcentaje mayores de 65 años Florida': 20.55777530198194}, {'Estado': 'Georgia', 'Poblacion 2000': 8186453, 'Poblacion 2001': 8229823, 'Residentes menores de 65 años 2000': 7440877, 'Residentes menores de 65 años 2001': 7582146, 'Muertes 2000': 14804, 'Muertes 2001': 15000, 'Fecha de fundacion del Estado': '12-02-1733', 'Latitud': 32.329381, 'Longitud': -83.113737, 'Porcentaje mayores de 65 años Georgia': 7.869877638923704}, {'Estado': 'South Carolina', 'Poblacion 2000': 4012012, 'Poblacion 2001': 4023438, 'Residentes menores de 65 años 2000': 3535770, 'Residentes menores de 65 años 2001': 3567172, 'Muertes 2000': 8581, 'Muertes 2001': 9500, 'Fecha de fundacion del Estado': '26-03-1776', 'Latitud': 33.687439, 'Longitud': -80.436374, 'Porcentaje mayores de 65 años South Carolina': 11.340202085877799}]
```

En este punto Jese se dio cuenta de la importancia del redondeo y modificamos el código. Asimismo, realizamos un print para que únicamente el programa devolviera la información sobre los mayores de 65.

El código se corrigió de la siguiente forma:

```
for estado in lista_estados:
    # Obtenemos la población mayor y menor de 65 años
    poblacion_mayores_65 = estado["Poblacion 2001"] - estado["Residentes menores de 65 años 2001"]

    # Calculamos el porcentaje de población mayor de 65 años
    porcentaje_mayores_65 = (poblacion_mayores_65 / estado["Poblacion 2001"]) * 100
    porcentaje_mayores_65_redondeado = round(porcentaje_mayores_65, 1)

    # Agregamos la nueva clave
    estado["Porcentaje mayores de 65 años " + estado["Estado"]] = porcentaje_mayores_65_redondeado
    print("El porcentaje de mayores de 65 años en", estado["Estado"], "es:", porcentaje_mayores_65_redondeado)
```

Procedemos con el **apartado 1.f**, donde se nos pide desarrollar un programa que imprima por pantalla el nombre del Estado más antiguo y el más moderno, y que valga para cualquier número de estados. Asimismo, tendremos que calcular cuántos años de diferencia hay entre ellos.

Para realizar este ejercicio creamos un código propuesto por Luisa y Ana, y después consensuado por el grupo, inicializando las variables y que luego itere para ir comparando qué fechas de fundación son más antiguas y cuales son más modernas.

```
estado_mas_antiguo = lista_estados[0]["Estado"]
estado_mas_moderno = lista_estados[0]["Estado"]
fecha_mas_antigua = lista_estados[0]["Fecha de fundacion del Estado"]
fecha_mas_moderna = lista_estados[0]["Fecha de fundacion del Estado"]
```



```

for estado in lista_estados[1:]:
    # Comparamos las fechas de fundación
    if estado["Fecha de fundacion del Estado"] < fecha_mas_antigua:
        estado_mas_antiguo = estado["Estado"]
        fecha_mas_antigua = estado["Fecha de fundacion del Estado"]
    elif estado["Fecha de fundacion del Estado"] > fecha_mas_moderna:
        estado_mas_moderno = estado["Estado"]
        fecha_mas_moderna = estado["Fecha de fundacion del Estado"]

print("El estado más antiguo es:", estado_mas_antiguo)
print("El estado más moderno es:", estado_mas_moderno)

```

Para el cálculo de los años de diferencia añadimos al código una resta `diferencia = fecha_mas_moderna.year - fecha_mas_antigua.year` y un nuevo print, pero nos encontramos con que la resta no podía realizarse directamente, así que Ana propone convertir las fechas de fundación en objetos datetime usando la función `strptime` y el formato `%d-%m-%Y`, para luego comparar estas fechas de la misma manera que en el código original. Finalmente, para calcular la diferencia de años, se resta la fecha más antigua de la fecha más moderna y se divide el número de días resultante entre 365 usamos `//` para que el resultado sea entero.

```

import datetime

estado_mas_antiguo = lista_estados[0]["Estado"]
estado_mas_moderno = lista_estados[0]["Estado"]
fecha_mas_antigua = datetime.datetime.strptime(lista_estados[0]["Fecha de fundacion del Estado"], '%d-%m-%Y')
fecha_mas_moderna = datetime.datetime.strptime(lista_estados[0]["Fecha de fundacion del Estado"], '%d-%m-%Y')

for estado in lista_estados[1:]:
    # Comparamos las fechas de fundación
    fecha_estado = datetime.datetime.strptime(estado["Fecha de fundacion del Estado"], '%d-%m-%Y')
    if fecha_estado < fecha_mas_antigua:
        estado_mas_antiguo = estado["Estado"]
        fecha_mas_antigua = fecha_estado
    elif fecha_estado > fecha_mas_moderna:
        estado_mas_moderno = estado["Estado"]
        fecha_mas_moderna = fecha_estado

diferencia = (fecha_mas_moderna - fecha_mas_antigua).days // 365
print("El estado más antiguo es:", estado_mas_antiguo)
print("El estado más moderno es:", estado_mas_moderno)
print("La diferencia de años entre el estado más antiguo y el más moderno es:", diferencia)

```

```

El estado más antiguo es: Georgia
El estado más moderno es: Florida
La diferencia de años entre el estado más antiguo y el más moderno es: 112

```

EJERCICIO 2

El código correspondiente al ejercicio 2, fue propuesto y realizado casi en su totalidad por Luisa y Ana. Posteriormente fue revisado y aprobado por el resto del grupo. En él se nos pide la tasa de crecimiento de población para 2002 de los estados de Alabama y South Carolina son dos números aleatorios entre 0 y 0.1. Asignaremos a Alabama el ratio de crecimiento menor y a South Carolina el ratio de crecimiento mayor. Vamos a asumir también la población de 2001 de ambos estados descrita en los ejercicios anteriores, y responder a las siguientes preguntas:

- ¿Cuántos años tardará el estado de South Carolina en alcanzar en población a Alabama?
- ¿En qué año ocurrirá esto?
- Si al planteamiento anterior le añadimos la tasa de fallecidos de cada estado. ¿Cuántos años tardará South Carolina en alcanzar en población a Alabama? Vamos a asumir la tasa de fallecidos y la población correspondiente a 2021.

Para la consecución del mismo entendemos que tenemos que asignar tasas de crecimiento aleatorias para los estados indicados por lo que utilizamos la librería random.

En dos variables, guardamos dos números random que pueden ser desde el 0 hasta el 0,1.

En una sentencia if, colocaremos como condición que la variable que obtuvo el número random menor, se le asigne a la variable alabama, ya que se pide en el enunciado que alabama tenga el crecimiento menor, por consecuencia a Carolina del Sur se le asigna la variable que obtuvo el número mayor.

```
import random

alabama_crecimiento = random.uniform(0, 0.1)
south_carolina_crecimiento = random.uniform(0, 0.1)

if alabama_crecimiento > south_carolina_crecimiento:
    alabama_crecimiento, south_carolina_crecimiento = south_carolina_crecimiento, alabama_crecimiento
```

Como el enunciado lo dice, sumamos el porcentaje de crecimiento para el año 2002 (recordando que el porcentaje para cada estado lo obtuvimos en el párrafo anterior) y lo guardamos en dos variables para cada estado.

```
# Calcular población proyectada para 2002
alabama_poblacion_2002 = lista_estados[0]['Poblacion 2001'] * (1 + alabama_crecimiento)
south_carolina_poblacion_2002 = lista_estados[3]['Poblacion 2001'] * (1 + south_carolina_crecimiento)
```

Para calcular cuántos años tardará Carolina del Sur en alcanzar a Alabama, creamos tres variables, las cuales se inicializan con el crecimiento esperado para el año 2002 para cada estado con la tasa de crecimiento (variables: alabama_poblacion y south_carolina_poblacion), y la tercera variable llamada anho (por año), se inicializa con el número 2002 del año con que se inicializa el crecimiento de población desde ahora.

Para cumplir la condición de que el estado de Carolina del Sur alcance la población de Alabama, creamos un bucle While, este iterará hasta que se cumpla la condición, por cada iteración las variables alabama_poblacion y south_carolina_poblacion se irá actualizando, tomará la población calculada en estas variables y le irá sumando la tasa de crecimiento, además que la variable anho irá sumando un año, todo esto hasta que se cumpla la condición. Así obtendremos el año en que Carolina del Sur alcanzó a Alabama en población.

Para calcular cuantos años demoró en cumplirse la condición la variable anho que almacena el año hasta que se cumplió la condición se resta el año 2002, ya que en ese año se inicializó el conteo de años.

```
# Calcular en qué año South Carolina alcanzará Alabama
alabama_poblacion = alabama_poblacion_2002
south_carolina_poblacion = south_carolina_poblacion_2002
anho = 2002
while south_carolina_poblacion < alabama_poblacion:
    anho += 1
    alabama_poblacion *= (1 + alabama_crecimiento)
    south_carolina_poblacion *= (1 + south_carolina_crecimiento)

print(f"a. South Carolina tardará {anho - 2002} años en alcanzar la población de Alabama.")
print(f"b. Esto ocurrirá en el año {anho}.")
```

Se nos pide por último hacer el mismo procedimiento, pero agregando la información de la tasa de mortalidad, la calculamos en base a los datos de personas muertas durante el año 2001 y hacemos el mismo procedimiento anterior, esta vez, restando el porcentaje de mortalidad para cada operación.

```
alabama_mortalidad = lista_estados[0]['Muertes 2001'] / lista_estados[0]['Poblacion 2001']
south_carolina_mortalidad = lista_estados[3]['Muertes 2001'] / lista_estados[3]['Poblacion 2001']
alabama_poblacion = lista_estados[0]['Poblacion 2001']
south_carolina_poblacion = lista_estados[3]['Poblacion 2001']
anho = 2001
while south_carolina_poblacion < alabama_poblacion:
    anho += 1
    alabama_poblacion *= (1 + alabama_crecimiento) * (1 - alabama_mortalidad)
    south_carolina_poblacion *= (1 + south_carolina_crecimiento) * (1 - south_carolina_mortalidad)

print(f'''c. South Carolina tardará {anho - 2001} años en alcanzar la población de Alabama
    si se tiene en cuenta la tasa de mortalidad.''' )
```

EJERCICIO 3

En este ejercicio se nos propone crear una función que genere una proyección para el año 2002, utilizando como ratio la comparativa entre los años 2000 y 2001. De tal forma: Población Año 2002 = Población Año 2001 / Población Año 2000 x Población año 2001. Tras un estudio demográfico se ha determinado que el número de habitantes de cierta población, en los próximos años, vendrá dado por la función: $P(t) = 14500t + 7000/2t + 1$ donde t son los años transcurridos entre la fundación del Estado y 1900. En la misma función, devolved la población que tendría cada Estado bajo la estimación que devuelve ese estudio demográfico. Añadid los resultados a cada diccionario de la lista.

Sergio nos plantea el siguiente código en que itera para calcular la proyección de la población en 2002 y posteriormente y una nueva iteración para el cálculo que vendrá dado teniendo en cuenta los años transcurridos entre la fundación del estado y 1900. Posteriormente a través de update se actualizan los datos en la lista.

```
for i in lista_estados:
    proyeccion2002 = round((i["Población 2001"] / i["Población 2000"]) * i["Población 2001"])
    i.update({"proyeccion crecimiento": proyeccion2002 })
print(proyeccion2002)

for i in lista_estados:
    yrfundacion = int(i["Fecha de fundación del Estado"][-4:])
    tf = 1900
    t = tf - yrfundacion
    print(t)

    pt = round((14500 * t) + ( 7000 / (2 * t) ) + 1)
    print(pt)

for i in lista_estados:
    i.update({"estimacion": pt })
```

No obstante, dado que en el ejercicio se solicitaba el uso de funciones, Jesé plantea una alternativa que es la que es aplicada en el resultado final ya que define la función calculando los años transcurridos desde la fundación del estado hasta 1900. Realizamos la prueba de código con alguna modificación en el print, y Ana unifica variables con el resto del código utilizado hasta el momento.

```
for estado in lista_estados:
    def proyeccion_poblacion_2002(poblacion_2000, poblacion_2001):
        poblacion_2002 = ((poblacion_2001) * (poblacion_2001 / poblacion_2000))
        return (poblacion_2002)

    poblacion_2000 = estado["Poblacion 2000"]
    poblacion_2001 = estado["Poblacion 2001"]
    print("La poblacion para el año 2002 en el estado de", estado["Estado"], "será:", round(proyeccion_poblacion_2002(poblacion_2000, poblacion_2001)))

for estado in lista_estados:
    def proyeccion_poblacion_2002(poblacion_2000, poblacion_2001):
        poblacion_2002 = ((poblacion_2001) * (poblacion_2001 / poblacion_2000))
        return poblacion_2002

    poblacion_2000 = estado["Poblacion 2000"]
    poblacion_2001 = estado["Poblacion 2001"]
    poblacion_2002 = round(proyeccion_poblacion_2002(poblacion_2000, poblacion_2001))
    print("La población para el año 2002 en el estado de", estado["Estado"], "será:", poblacion_2002)

    yrfundacion = int(estado["Fecha de fundación del Estado"][-4:])
    tf = 1900
    t = tf - yrfundacion
    pt = round((14500 * t) + ( 7000 / (2 * t) ) + 1)
    print("La estimación de población en el estado de", estado["Estado"], "será:", pt)

    estado.update({"Estimacion": pt, "Poblacion 2002": poblacion_2002})
```

```

La población para el año 2002 en el estado de Alabama será: 4455890
La estimación de población en el estado de Alabama será: 1174544
La población para el año 2002 en el estado de Florida será: 16126602
La estimación de población en el estado de Florida será: 797565
La población para el año 2002 en el estado de Georgia será: 8273423
La estimación de población en el estado de Georgia será: 2421522
La población para el año 2002 en el estado de South Carolina será: 4034897
La estimación de población en el estado de South Carolina será: 1798029
[{'Estado': 'Alabama', 'Poblacion 2000': 4447100, 'Poblacion 2001': 4451493, 'Residentes menores de 65 años 2000': 3870598, 'Residentes menores de 65 años 2001': 3880476, 'Muertes 2000': 10622, 'Muertes 2001': 15912, 'Fecha de fundacion del Estado': '14-12-1819', 'Latitud': 33.258882, 'Longitud': -86.829534, 'Días desde fundación Alabama': 74269, 'Porcentaje mayores de 65 años Alabama': 12.8, 'Estimacion': 1174544, 'Poblacion 2002': 4455890}, {'Estado': 'Florida', 'Poblacion 2000': 15982378, 'Poblacion 2001': 16054328, 'Residentes menores de 65 años 2000': 13237167, 'Residentes menores de 65 años 2001': 13548077, 'Muertes 2000': 38103, 'Muertes 2001': 166069, 'Fecha de fundacion del Estado': '03-03-1845', 'Latitud': 27.756767, 'Longitud': -81.463983, 'Días desde fundación Florida': 65058, 'Porcentaje mayores de 65 años Florida': 15.6, 'Estimacion': 797565, 'Poblacion 2002': 16126602}, {'Estado': 'Georgia', 'Poblacion 2000': 8186453, 'Poblacion 2001': 8229823, 'Residentes menores de 65 años 2000': 7440877, 'Residentes menores de 65 años 2001': 7582146, 'Muertes 2000': 14804, 'Muertes 2001': 15000, 'Fecha de fundacion del Estado': '12-02-1733', 'Latitud': 32.329381, 'Longitud': -83.113737, 'Días desde fundación Georgia': 105984, 'Porcentaje mayores de 65 años Georgia': 7.9, 'Estimacion': 2421522, 'Poblacion 2002': 8273423}, {'Estado': 'South Carolina', 'Poblacion 2000': 4012012, 'Poblacion 2001': 4023438, 'Residentes menores de 65 años 2000': 3535770, 'Residentes menores de 65 años 2001': 3567172, 'Muertes 2000': 8581, 'Muertes 2001': 9500, 'Fecha de fundacion del Estado': '26-03-1776', 'Latitud': 33.687439, 'Longitud': -80.436374, 'Días desde fundación South Carolina': 90236, 'Porcentaje mayores de 65 años South Carolina': 11.3, 'Estimacion': 1798029, 'Poblacion 2002': 4034897}]

```

Una vez realizadas todas las comprobaciones, finalmente se verificó que no era una solución óptima para este apartado, por lo que revisamos de nuevo el código y se decidió modificarlo por el aportado por Luisa

```

def proyeccion_poblacion():
    for estado in lista_estados:

        # se hace el calculo de la poblacion para 2002 con la formula dada y se actualiza el diccionario
        poblacion_2002 = round(estado['Poblacion 2001'] / estado['Poblacion 2000'] * estado['Poblacion 2001'])
        estado.update({"Poblacion 2002": poblacion_2002 })

        print("La población para el año 2002 en el estado de", estado["Estado"], "será:", poblacion_2002)
        # se hace el calculo para sacar la proyeccion estimada para cda uno de los estados con la formula dada
        calculo_entre_fechas = 1900 - int(estado["Fecha de fundacion del Estado"][-4:])
        formula_de_proyeccion = round((14500 * calculo_entre_fechas + 7000 / (2 * calculo_entre_fechas) + 1))
        estado.update({"Estimacion": formula_de_proyeccion })
        print("La estimación de población en el estado de", estado["Estado"], "será:", formula_de_proyeccion)

```

```

La población para el año 2002 en el estado de Alabama será: 4455890
La estimación de población en el estado de Alabama será: 1174544
La población para el año 2002 en el estado de Florida será: 16126602
La estimación de población en el estado de Florida será: 797565
La población para el año 2002 en el estado de Georgia será: 8273423
La estimación de población en el estado de Georgia será: 2421522
La población para el año 2002 en el estado de South Carolina será: 4034897
La estimación de población en el estado de South Carolina será: 1798029

```


EJERCICIO 4

En el ejercicio se nos solicita que creemos un mapa utilizando la librería Folium y utilizando bucles, donde se representen con un marcador de color azul (color="blue") los datos de población del 2002 para cada uno de los estados, estando posicionados estos círculos en las coordenadas que se facilitan.

Ninguno de nosotros conocía Folium, y por temas organizativos y de reparto de tareas del grupo, el peso de la investigación del ejercicio 4 recayó sobre nuestro compañero Jaime, quien posteriormente nos facilitó toda la información relativa a la instalación y aplicación de Folium.

El código propuesto fue el siguiente y fue anexado al resto de código que ya habíamos comprobado previamente.

Las coordenadas y el zoom elegido se establecen antes de empezar a iterar los estados y se aplican las coordenadas y población calculada para 2002.

Para la creación del círculo se añade a folium Circle que es lo que determinará la forma de la zona marcada en el mapa, y se establece tanto el cálculo del radio que se obtiene de la Población de 2002 partido por 100, como las especificaciones de color del círculo elegido.

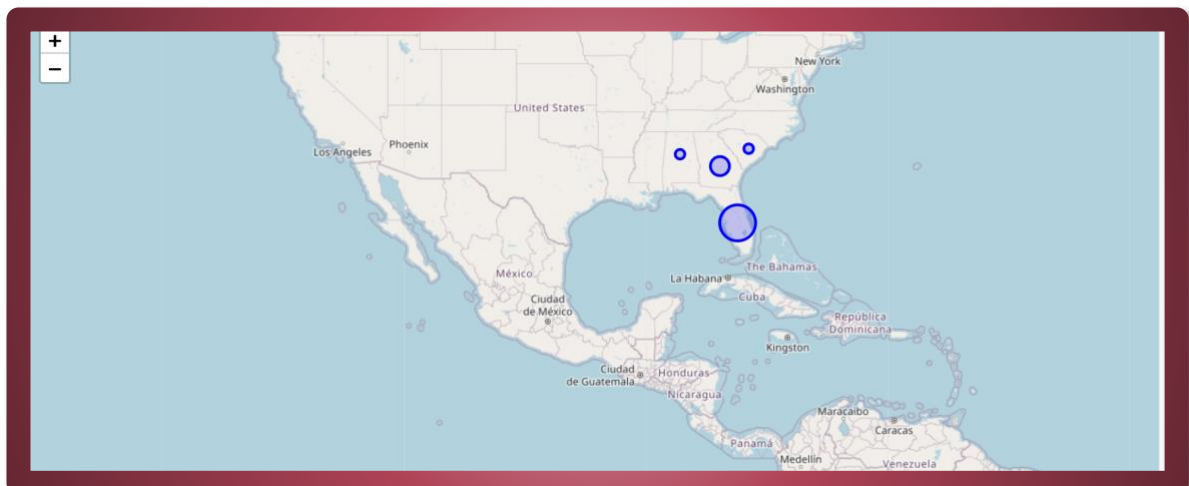
Dado que las pruebas se realizaron sin tener aún los datos de 2002, en las anotaciones del código y en las variables pone la población de 2001 que es con la que se realizaron las primeras pruebas de funcionamiento, aunque posteriormente al unir todo el código actualizamos toda la información.

```
mapa = folium.Map(location=[30.101271, -82.370146], zoom_start=6)

for estado in lista_estados:
    # Coordenadas y población del 2001 para cada estado
    latitud = estado['Latitud']
    longitud = estado['Longitud']
    poblacion = estado['Poblacion 2001']

    # Crear círculo azul de cada estado con el tamaño de radio dependiendo de la población
    folium.Circle(location=[latitud, longitud],
                  radius=estado['Poblacion 2001']/100,
                  color='blue',
                  fill_color='blue',
                  ).add_to(mapa)

mapa
```





ANEXO

Para finalizar, mostramos en este link unidas todas las partes del código explicado en este trabajo, donde hemos aplicado modificaciones en los prints, únicamente para que los resultados fueran más legibles, y constan todos los datos actualizados.

[CÓDIGO ACTUALIZADO](#)