



Disponemos de un conjunto de datos con información sobre el virus del zika. Los datos se encuentran almacenados en un fichero en formato CSV codificado en UTF-8. Cada registro del fichero ocupa una línea y contiene los datos correspondientes a la acción del virus zika sobre una zona: fecha de la detección, localización, tipo de división territorial (estado, país...), sintomatología detectada, código asignado al reporte, número de casos o de municipios afectados, y unidades del valor anterior (casos o municipios). Estas son las primeras líneas del fichero:

```
report_date,location,location_type,data_field,data_field_code,value,unit
02/07/2016,Brazil-Pernambuco,state,microcephaly_not,BR0003,1173,cases
02/07/2016,Brazil-Bahia,state,microcephaly_under_investigation,BR0001,659,cases
02/07/2016,Brazil-Pernambuco,state,microcephaly_under_investigation,BR0001,489,cases
02/07/2016,Brazil-Paraiba,state,microcephaly_not,BR0003,479,cases
```

El objetivo del ejercicio es leer estos datos y realizar distintas operaciones con ellos. Cada operación se implementará en una función distinta. Estas funciones son las siguientes:

1. **lee_fichero(fichero)**: lee el fichero de entrada y lo almacena en una lista de tuplas. (1 Punto)
2. **num_casos_entre(reports, fecha_inicial, fecha_final)**: obtiene el número total de casos detectados entre las fechas especificadas por el intervalo (fecha_inicial, fecha_final), ambas incluidas. Nótese que cada tupla representa un informe en el que el campo 'unit' hace referencia al dominio, pudiendo tomar como valores 'cases' y 'municipalities' (casos y municipios respectivamente), y el campo 'value' contiene el número de casos detectado en el informe que representa la tupla. (1 Punto)
3. **casos_zika_territorio_USA(reports)**: obtiene una lista con las localizaciones y el número de casos del virus Zika en el territorio USA. Note que el tipo de localización debe ser 'territory', las unidades deben ser 'cases' y la localización debe contener 'United_States'. (1,5 Puntos)
4. **filtra_por_estado(reports, anyo)**: obtiene una lista, ordenada por fechas, con tuplas distintas de la forma (fecha, localización) correspondientes a los informes cuyo tipo de localización sea 'state', y cuyos casos tuvieron lugar en el año que se pasa como parámetro de entrada. En la lista obtenida no debe haber tuplas duplicadas. (1,5 Puntos)
5. **anyo_mas_casos(reports)**: obtiene el año en el que se han producido más casos de Zika, en forma de tupla (total_casos, año). (1,5 Puntos)
6. **porcentaje_por_descrip(reports)**: crea un diccionario cuyas claves son la descripción del reporte ('data_field'), y cuyos valores son el porcentaje de informes con respecto al número total de informes con esa descripción. (1,5 Puntos)
7. **dibuja_casos_zika_por_pais(reports)**: dibuja un gráfico que muestra el número total de casos acumulados por país. El total de casos de un país se calcula acumulando el valor por país de aquellos reportes con unidad 'cases' y tipo de localización 'country'. Tenga en cuenta también que los Estados Unidos son una excepción a esta regla, ya que el número de casos de este país se debe calcular como la suma de los casos del territorio USA (al no haber informes que acumulen la totalidad del país, sino que la información está desglosada por territorio). (2 Puntos)

Escriba el código de cada función y ejecute el test correspondiente para probar su funcionamiento. Las soluciones deben ser genéricas y adaptarse a los datos que se reciben como parámetros, sin presuponer unos valores concretos para estos.

El resultado esperado de cada test lo tiene escrito en forma de comentario junto a la llamada al test correspondiente en el programa principal. El gráfico resultado de la función `dibuja_casos_zika_por_pais` es el siguiente:

