



En Sevilla se ha celebrado del 30 de mayo al 9 de junio la competición *"The Champions Burger"*. En este campeonato, varias hamburgueserías de toda España presentan una hamburguesa específica para que los asistentes puedan degustarla. Los asistentes prueban las hamburguesas y dan una puntuación a través de un cuestionario que se realiza a través de una aplicación móvil. Al final del certamen, se realiza un recuento de las valoraciones de los asistentes para determinar la hamburguesa ganadora. El objetivo de este examen va a ser realizar un análisis de los datos recogidos durante este certamen. Para cada valoración registrada, se tiene la siguiente información:

- **Email:** Correo electrónico de la persona que realizó la valoración.
- **Ciudad:** Ciudad de origen de donde proviene el evaluador.
- **Código Postal:** Código postal de donde proviene el evaluador.
- **Fecha y hora entrada:** Fecha y hora de entrada al recinto.
- **Temperatura:** Temperatura que hacía a la hora de la entrada al recinto del campeonato.
- **Fecha y hora salida:** Fecha y hora de salida del recinto.
- **Puntuaciones hamburgueserías:** Lista de evaluaciones para cada hamburguesería. Cada evaluación contiene el nombre de la hamburguesería y varias puntuaciones (presentación, punto de la carne, calidad de los ingredientes y calidad del pan).

Una línea ejemplo de del archivo csv en el que están registrados los datos es la siguiente:

```
usuario2@example.com;Sevilla;41006;04/06/2024 11:14;34;04/06/2024 14:14;[Brothers: (5, 7, 10, 8) - Umbrella: (6, 5, 7, 7)]
```

Esta línea indica que la persona con el email usuario2@example.com procede de Sevilla, específicamente de la zona con el código postal 41006. Esta persona visitó el evento el día 04/06/2024, cuando la temperatura era de 34°C, y estuvo en el evento desde las 11:14 hasta las 14:14. Durante su visita, probó un total de dos hamburguesas: la primera en Brothers, otorgándole una puntuación de 5 a la presentación, 7 al punto de la carne, 10 a la calidad de los ingredientes y 8 a la calidad del pan. La segunda hamburguesa fue en Umbrella, con una puntuación de 6 a la presentación, 5 al punto de la carne, 7 a la calidad de los ingredientes y 7 a la calidad del pan. Para analizar los datos, **implemente en Java** lo especificado en los siguientes ejercicios:

Ejercicio 1: Tipo Evaluacion (0,5 pts)

Implemente el tipo Evaluacion mediante **un record**, de acuerdo con la siguiente información:

Propiedades:

- **Hamburguesería**, de tipo String, consultable
- **Presentación**, de tipo Integer, consultable.
- **Punto de la carne**, de tipo Integer, consultable
- **Calidad de los ingredientes**, de tipo Integer, consultable
- **Calidad del pan**, de tipo Integer, consultable
- **Puntuación Final**, de tipo Double, consultable. La puntuación se calcula como la media de las puntuaciones individuales obtenidas en los diferentes criterios evaluados (presentación, punto de la carne, calidad de los ingredientes, calidad del pan). Por ejemplo, si la evaluación es *Brothers: (5, 7, 10, 8)*, la puntuación es 7.5



Constructores:

- C1: recibe un parámetro por cada propiedad básica del tipo, en el mismo orden en el que están definidas.

Representación como cadena: una cadena con todas las propiedades básicas del tipo.

Restricciones:

- R1: El valor de valoración de la presentación debe estar comprendido entre 0 y 10.
- R2: El valor de valoración del punto de la carne debe estar comprendido entre 0 y 10.
- R3: El valor de valoración de la calidad de los ingredientes debe estar comprendido entre 0 y 10.
- R4: El valor de valoración de la calidad del pan debe estar comprendido entre 0 y 10.

Ejercicio 2: tipo Visita (1,5 ptos)

Implemente el tipo Visita utilizando **una clase**, de acuerdo con la siguiente información:

Propiedades:

- **Email**, de tipo String, consultable
- **Ciudad**, de tipo String, consultable
- **Código Postal**, de tipo String, consultable
- **Entrada**, de tipo LocalDateTime, consultable
- **Salida**, de tipo LocalDateTime, consultable y modificable
- **Temperatura**, de tipo Double, consultable y modificable
- **Tiempo transcurrido**, de tipo Duration, consultable. Indica el tiempo que la persona estuvo en el recinto.
- **Evaluaciones**, de tipo List<Evaluacion>, consultable
- **Número de evaluaciones**, de tipo Integer, consultable. Indica el número de hamburguesas que la persona comió.
- **Paladar**, de tipo Paladar, consultable. Puede tomar los valores BAJO, MEDIO, ALTO. Se denomina que una persona tiene paladar BAJO si la media de las puntuaciones finales de las evaluaciones es mayor o igual a 9, MEDIO si está entre 6 y 9 y ALTO si es menor o igual que 6.

Constructores:

- C1: recibe un parámetro por cada propiedad básica del tipo.

Restricciones:

- R1: El momento de salida debe ser posterior al momento de entrada.
- R2: El email debe contener el carácter '@'.
- R3: La lista de evaluaciones debe contener al menos un elemento.
- R4: El día de entrada tiene que ser el mismo que el de salida.

Representación como cadena: una cadena con todas las propiedades básicas del tipo.

Criterio de igualdad: dos visitas son iguales si lo son su email, su momento de entrada y su momento de salida.

Criterio de ordenación: dos visitas se ordenan por su momento de entrada, su momento de salida, y a igualdad de ambas propiedades, por email.

**Ejercicio 3: Factoría (1 pto)**

En la clase **FactoriaVisitas**, que se le da parcialmente implementada, implemente el método:

Visita parseaVisita(String lineaCSV): crea un objeto de tipo *Visita* a partir de una cadena de caracteres. La cadena de caracteres debe tener el mismo formato que las líneas del fichero CSV.

Ejercicio 4: Tratamientos secuenciales (7 ptos)

El tipo **Competicion** tiene la siguiente descripción:

Propiedades:

- **visitas**: conjunto ordenado de las visitas recogidas durante la competición, de tipo *SortedSet*, consultable. Las visitas están ordenadas por código postal del usuario, y a igualdad de código postal se ordenarán por el orden natural de *Visita*. No debe ser posible añadir o eliminar elementos al conjunto desde fuera del tipo contenedor.

Constructores:

- C1: recibe un parámetro de tipo *Stream<Visita>*.

Representación como cadena: una cadena con todas las visitas. Cada visita aparecerá en una línea distinta.

Criterio de igualdad: dos objetos de tipo *Competicion* son iguales si lo son sus visitas.

Implemente el tipo **Competicion** y añada los siguientes tratamientos secuenciales. Debe resolver todos los métodos **mediante streams**, salvo que se le indique expresamente que debe utilizar bucles:

1. *SortedSet<String> getEmailsOrdenados(Duration d)*: devuelve un conjunto ordenado con los emails de aquellas visitas cuya duración fue mayor a *d* y cuya temperatura registrada fue menor a la temperatura media de todas las visitas realizadas. **(1 pto)**
2. *Integer getTotalVisitasComilones()*: devuelve un *Integer* con el número de visitas en las que el usuario ha realizado un número de evaluaciones mayor a la media de evaluaciones de todas las visitas realizadas en el campeonato. **(1 pto)**
3. *String getPeorHamburgueseriaPorCalidadIngredientes()*: devuelve el nombre de la hamburguesería con la peor puntuación promedio en la categoría de calidad de ingredientes. Si no se puede calcular, devuelve *null*. **(1.5 ptos)**
4. *Map<String, String> getTopComilonPorCPEnDia(LocalDate d)*: teniendo solamente en cuenta las visitas del día *d*, devuelve un *Map* en el que a cada código postal le hace corresponder el email de la persona que realizó la visita en la que se comieron más hamburguesas. Si hay dos visitas en las que se comió el mismo número de hamburguesas, se devolverá la que tiene una fecha de entrada más tardía. Tenga en cuenta que una persona hace como máximo una visita al día. **(1.5 ptos)**
5. *String getHamburgueseriaGanadora()*: devuelve el nombre de la hamburguesería ganadora. Se considera hamburguesería ganadora a aquella que ha obtenido la mayor puntuación promedio basada en las puntuaciones finales de las visitas registradas. **Implemente este método con bucles. (2 ptos)**

Escriba una clase **TestCompeticion**. En la clase se leerán los datos del fichero y se probarán todos los tratamientos secuenciales, definiendo un método de test por cada tratamiento secuencial a probar. No



se obtendrá la puntuación máxima del ejercicio si no se realiza el test y éste ejecuta. Los resultados esperados para el dataset proporcionado, con los valores indicados en los tests, son:

```
FACTORÍA = = = = =
Leidas 105 visitas
Primera visita: Visita [email=usuario33@example.com, ciudad=Cadiz, cp=10451,
entrada=2024-06-02T14:21, salida=2024-06-02T15:21, temperatura=34.0,
evaluaciones=[Evaluacion[hamburgueseria=Torriko, presentacion=6, punto=8,
ingredientes=4, pan=5], Evaluacion[hamburgueseria=El Barco, presentacion=3,
punto=10, ingredientes=6, pan=7], Evaluacion[hamburgueseria=Soul, presentacion=6,
punto=7, ingredientes=4, pan=4], Evaluacion[hamburgueseria=Bilios,
presentacion=10, punto=8, ingredientes=9, pan=4], Evaluacion[hamburgueseria=La
Muralla, presentacion=8, punto=3, ingredientes=10, pan=10]]]
```

```
Última visita: Visita [email=usuario9@example.com, ciudad=Cadiz, cp=94642,
entrada=2024-06-04T20:35, salida=2024-06-04T23:35, temperatura=30.0,
evaluaciones=[Evaluacion[hamburgueseria=El Barco, presentacion=7, punto=3,
ingredientes=6, pan=9], Evaluacion[hamburgueseria=XPecado, presentacion=10,
punto=9, ingredientes=7, pan=5], Evaluacion[hamburgueseria=Djanco,
presentacion=8, punto=6, ingredientes=3, pan=8], Evaluacion[hamburgueseria=Godeo,
presentacion=6, punto=5, ingredientes=8, pan=5]]]
```

```
EJ1 = = = = =
Los emails de las visitas con duración mayor a 240 minutos son:
[usuario13@example.com, usuario49@example.com]
Los emails de las visitas con duración mayor a 300 minutos son:
[]
```

```
EJ2 = = = = =
El total de visitas con un número de evaluaciones mayor a la media es:
55
```

```
EJ3 = = = = =
Si se tiene en cuenta solo la calidad de ingredientes, la peor hamburguesería
es:
Jenkins
```

```
EJ4 = = = = =
En el día 2024-06-02 la persona que más comió de cada CP es:
{41009=usuario49@example.com, 17602=usuario13@example.com,
10451=usuario33@example.com, 41002=usuario74@example.com,
21002=usuario91@example.com, 41003=usuario17@example.com,
41004=usuario48@example.com}
```

```
EJ5 = = = = =
La hamburguesería ganadora del campeonato es:
Godeo
```