



En este ejercicio vamos a trabajar con datos de partidas del popular videojuego Counter-Strike, disponibles en <https://www.kaggle.com/thesiff/counterstrike>. El objetivo del ejercicio es leer esos datos y realizar distintas operaciones con ellos.

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 una partida, que están explicados en el documento Descripción del dataset.

Estas son las primeras líneas del fichero. La primera línea es una cabecera que contiene los nombres de los campos del registro:

1	"	Day	Month	Year	Date	Wait Time(s)	Match Time	Team A Rour	Team B Rour	Ping	Kills	Assists	Deaths	Mvp's	HS%	Points	Result	K/D Ratio	id
2	Mirage	3	8	2018	03/08/2018	327	2906	16	13	215	17	2	21	2	5	45	Win	True	ssgrave0
3	Mirage	2	8	2018	02/08/2018	336	2592	16	11	199	13	4	24	2	0	40	Lost	False	wostler1
4	Mirage	31	7	2018	31/07/2018	414	2731	16	14	85	15	3	18	3	26	37	Win	False	lledgard2
5	Mirage	31	7	2018	31/07/2018	317	2379	11	16	93	12	2	15	2	16	30	Lost	False	gdruitt3
6	Mirage	30	7	2018	30/07/2018	340	3467	15	15	94	33	5	20	5	30	83	Tie	True	hmdarden4
7	Mirage	29	7	2018	29/07/2018	391	1881	16	4	88	13	2	17	1	38	34	Lost	False	mseeds5
8	Dust II	28	7	2018	28/07/2018	274	3194	16	14	89	19	5	24	2	15	52	Lost	False	ecauson6
9	Mirage	27	7	2018	27/07/2018	291	2859	16	14	82	17	1	25	0	35	37	Lost	False	dgiscken7
10	Mirage	27	7	2018	27/07/2018	184	2300	16	8	112	25	10	12	4	16	70	Win	True	fpape8

Constructores:

- Un constructor sin parámetros.
- Un constructor que recibe parámetros de mapa, mvp, resultadoPartida e id iguales a "Dust II", 2, WIN, "Backewe216" respectivamente.
- Un constructor a partir de String.

Criterio de igualdad:

Dos partidas son iguales si tienen la misma fecha, el mismo mapa y la misma duración.

Criterio de ordenación:

Por mapa, a igualdad por fecha y por duración de la partida.

Representación como cadena:

Generada automáticamente con todas las propiedades básicas del tipo.

Restricciones:

- El año debe ser mayor o igual que 2015
- El mapa no puede ser vacío



- Los puntos deben ser positivos.
- La id del jugador debe tener más de 5 caracteres.
- La duración de la partida debe ser mayor que 0.

Otras operaciones:

- Double getKillsPorSegundo(): da las kills entre el matchTime de una partida.

Tipo FactoriaPartia

Operaciones:

- PartidaConetnedor leerPartidas (String ruta): lee un fichero de partidas y construye un objeto PartidaContenedor.
- Partida parsearPartida (String lineaCSV): crea un objeto de tipo Partida a partir de una cadena de caracteres. La cadena de caracteres debe tener el mismo formato que las líneas del fichero CSV.

Tipo PartidaContenedor

Propiedades:

- partidas** de tipo List<Partida>. Consultable. Lista de partidas.

Constructores:

- Un constructor sin parámetros.
- Un constructor a partir de un Stream <Partida>.

Criterio de igualdad:

Dos partidas son iguales si sus partidaContenedor son iguales.

Representación como cadena:

Generada automáticamente con todas las propiedades básicas del tipo.

Otras operaciones:



- List<Partida> getPartidaContenedor: obtiene una partida.
- void añadirPartida: añade una partida.
- Void eliminarPartida: elimina una partida, si no existe lanzar una excepción.
- boolean contienePartida: comprobar si partidaContenedor contiene una partida.
- void añadirPartidas: añadir un conjunto de partidas.
- boolean contienePartidas: ver si contiene todas las partidas de un conjunto.

Tratamientos secuenciales:

BLOQUE I

- Integer calcularPartidasPorMapa: filtra partidas en función del parámetro mapa y las cuenta.
- Boolean existePartidaMapaAño: filtra por mapa y año y devuelve si existe una partida en función de si hay al menos un valor que coincida de los dos a la vez.

BLOQUE II

- Integer sumarPuntosPartidaMapa: filtra por mapa y devuelve la suma de puntos de los primeros 5 resultados.
- Double killsMediasMapa: filtra por mapa y devuelve la media de kills por mapa.

BLOQUE III

- calcularMapaMayorPuntuacionDeAño: calcula el mapa con más puntos conseguidos en función del año parámetro.
- List<String> obtenerJugadoresMasKills: calcula los n jugadores con mayor cantidad de kills conseguidas.

BLOQUE IV

- Map<String, Set<Integer>> calcularRondasPorMapa: calcula las rondas ganadas por el equipo A en cada mapa sin repetirse.
- Map<String, Partida> calcularPartidaMasKillsMapa: calcula las kills máximas en todos los mapas.
- Map<Resultado, Long> calculaMediaMuertesMapa: calcula la media de muertes por mapa.
- Map<Integer, List<String>> calculaListaNPartidasPorAnyo: calcula la lista de n jugadores por año y puntos.
- String calcularMapasMasRondasEquipoBMayores: recibe un map de los n valores más altos de rondas B ganadas y devuelve el mapa con el número más alto.