



## Documento de descripción del dataset

DATOS DEL ALUMNO			
Nombre y apellidos	Pablo Fernández Alcántara		
Titulación	<input checked="" type="checkbox"/> IC <input type="checkbox"/> IS <input type="checkbox"/> TI <input type="checkbox"/> DG		
Grupo	<input type="checkbox"/> 1 <input checked="" type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	Subgrupo de laboratorio	<input type="checkbox"/> 1 <input checked="" type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4

INFORMACIÓN DEL DATASET	
Nombre	CounterStrike
URL	<a href="https://www.kaggle.com/thesiff/counterstrike">https://www.kaggle.com/thesiff/counterstrike</a>
¿El dataset contiene varios ficheros csv? <input type="checkbox"/> Sí <input checked="" type="checkbox"/> No	
En caso de responder sí, indique el que ha elegido:	
Descripción de los datos (lo más detallada posible): El dataset recoge los datos de una serie de jugadores en varias partidas del videojuego CounterStrike así como varios datos generales de la partida en general.	

DESCRIPCIÓN DE LAS COLUMNAS	
Nº de columnas	12
Para cada columna, indique lo que representa y el tipo de dato que cree que se adapta mejor a ella.	
Columna 1	Map (String): El mapa en el que se jugó la partida.
Columna 2	Year (Integer): El año de la partida.
Columna 3	Date (LocalDate): La fecha de la partida.
Columna 4	Match Time (Integer): La duración de la partida en segundos.



Columna 5	Team A Rounds(Integer): Rondas ganadas por el equipo A.
Columna 6	Team B rounds(Integer): Rondas ganadas por el equipo B.
Columna 7	Kills(Integer): Bajas logradas por el jugador en la partida.
Columna 8	Deaths(Integer): Número de veces que el jugador ha muerto.
Columna 9	Mvp's(Integer): Cantidad de títulos de Mvp que ha recibido el equipo del jugador en la partida.
Columna 10	Points(Integer): Puntos ganados por el jugador en la partida.
Columna 11	Result(Enum): Resultado de la partida como victoria, derrota o empate.
Columna 12	K/D Ratio(Boolean): Devuelve True en caso de que el ratio de kills y deaths sea mayor o igual a 0.
Columna 13	Id (String): Nick del jugador

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:

- 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.
- 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.