# Tipos Futbol

Se dispone de los resultados de primera y segunda división desde la temporada 00-01 hasta la temporada 15-16. Los datos están organizados en dos carpetas (Primera y Segunda), y dentro de cada una de ellas hay un fichero CSV para cada temporada. Cada línea de estos ficheros se corresponde con un partido, y en ella se incluyen cinco datos:

* Fecha del partido
* Equipo que juega en casa
* Goles del equipo de casa
* Equipo visitante
* Goles del equipo visitante

Estas serían, por ejemplo, las primeras líneas del fichero de la temporada 15-16 en primera división:

21/08/2015,Málaga CF,0,Sevilla FC,0

21/08/2015,Atlético de Madrid,1,UD Las Palmas,0

21/08/2015,RC Deportivo,0,Real Sociedad,0

21/08/2015,RCD Espanyol,1,Getafe CF,0

21/08/2015,Rayo Vallecano,0,Valencia CF,0

21/08/2015,Athletic Club,0,FC Barcelona,1

21/08/2015,Real Betis,1,Villarreal CF,1

Los principales aspectos que tendremos que resolver a la hora de procesar estos datos de entrada serán: separar adecuadamente los campos mediante las comas, interpretar el formato de las fechas para extraer el día, el mes y el año, y leer todos los ficheros de una carpeta para obtener una lista con todos los partidos de una competición.

Tipos:

Resultado: enum {Empata,Gana}

## Equipo:

* Nombre: String
* Elo: Double

## Partido:

Propiedades

* Fecha: LocalDate
* EquipoLocal: Equipo
* GolesLocal: Integer
* EquipoVisitante: Equipo
* GolesVisitante: Integer
* HayEmpate: Boolean, derivada
* EquipoGanador: String, derivada, @pre !HayEmpate
* EquipoPerdedor: String, derivada, @pre !HayEmpate
* Equipos:Set<String>, derivada
* EloTrasPartido(K:integer): @pos actualiza los Elos de los participantes

Representación: Propiedades básicas

Factoría: parse de una línea del fichero

## Temporada

Propiedades

* Partidos: List<Partido>, ordenados por fechas
* Nombre: String

Representación: Un partido por línea

Factoría: parse de un fichero

## Liga:

Propiedades:

* Partidos: List<Partido>, ordenados por fechas
* Nombre: String

## Comun:

* EquiposParticipantes(List<Partido> partidos): Set<Partido>
* PartidosEntreFechas(List<Partido> partidos, LocalDate inicio, LocalDate fin):

List<Partido>

* PartidoJugadosPorEquipos(List<Partido> partidos,): Set<Partido>

Representación: Nombre de la Liga, un partido por línea, delante de cada temporada su nombre

Factoría: parse de una carpeta que contiene un fichero por temporada

## Notas

Puntuación Elo

La puntuación Elo de un jugador se calcula a partir de los resultados de sus enfrentamientos con otros jugadores. La idea básica es que la diferencia de puntuación de los contrincantes da, a priori, distintas probabilidades de victoria a cada uno de ellos. Es lo que se denomina puntuación esperada. Por ejemplo, dadas las puntuaciones Elo *Ea* y *Eb*de dos jugadores *A* y *B*, las puntuaciones esperadas *Pea* y *Peb* ante un enfrentamiento entre ellos, se calculan con la siguiente fórmula:

*Pea* y *Peb*  son probabilidades y, por tanto, valores están entre 0 y 1. Además, se cumple que *Pea* + *Peb  = 1.*

La puntuación *Elo* se incrementa, o disminuye, tras un enfrentamiento según si el resultado es mejor o peor de lo que era previsible en función de las puntuaciones esperadas de los contrincantes. Se aplica un ajuste lineal simple, proporcional a la diferencia entre la puntuación esperada y la obtenida por un contrincante. Los resultados normalizados *Ra, Rb* son:

* 1 en caso de victoria
* 0 en caso de derrota
* 0.5 en caso de empate

A partir de las puntuaciones esperadas y los resultados normalizados se calculan las nuevas puntuaciones Elo *Ea’* y *Eb’*de la forma:

Donde k es un parámetro que regula la estabilidad del ranking y su velocidad de convergencia. En ajedrez, por ejemplo, se usa k=16 para maestros y k=32 para jugadores de nivel menor. Dando así mayor velocidad a la competición entre aficionados que al ranking de los maestros