# TP 3: Algoritmo Goloso

## Universidad Nacional de General Sarmiento

## Materia: Programacion 3

Comision 1 (noche)

**Docentes:** Patricia Bagnes, Javier Marenco

**Integrantes:** Richter Alexis Valentin

# Problema:

Iimplementar un algoritmo goloso para asignar arbitros a partidos de un campeonato, intentando maximizar la equidad de la asignacion. Se debera tambien implementar una aplicacion visual para ejecutar este algoritmo y visualizar sus resultados.
Tenemos un campeonato con 2n equipos, de modo tal que en cada fecha se juegan n partidos (cada equipo juega exactamente un partido por fecha). Como parte de los datos de entrada tenemos el calendario de partidos. Tenemos ademas n arbitros. El problema consiste en determinar quien sera el arbitro de cada partido, con el siguiente criterio. Para cada equipo i, llamemos $m_i$ a la maxima cantidad de partidos de i con un mismo arbitro (es decir, si $m_i = 4$ entonces el equipo i tiene a un mismo arbitro en cuatro partidos, y no tiene a ningun otro arbitro en cinco o mas partidos). El objetivo es minimizar el promedio del peor arbitro

Se debera implementar una aplicacion visual con la siguiente funcionalidad.
1. Leer el calendario de partidos desde un archivo, cuyo formato queda a criterio del grupo. Puede ser un archivo JSON, XML o de texto plano.
2. Mostrar el calendario de partidos en controles visuales adecuados.
3. Ejecutar el algoritmo goloso para asignar los arbitros. Por una cuestion de transparencia, los arbitros se denominan con los numeros de 1 a n en lugar de tener sus nombres.
4. Mostrar el resultado de la asignacion (es decir, que numero de arbitro estara a cargo de cada partido)

# Implementacion:

El codigo esta divido en cuatro packages:

application: Solo se encuentra la clase Main que la cual ejecuta todo el programa

campeonato: La clase Calendario, la cual solo posee un array de fechas y sus respectivos metodos. La clase Equipo, que contiene un hashmap de arbitros y el nombre del equipo. La clase Fecha, que posee el nro de la fecha y un set de los partidos que se juegan en la fecha. La clase Partido, que tiene los dos equipos que juegan un partido y el arbitro. Cada una de estas clase tiene su test dentro de package tests y todas implementan serializable

model: La clase Instancia, la cual respresenta una instancia del problema con un Calendario y una cant de arbitros. La clase Model, donde se implenta la logica. La clase Solucion, la cual respresenta una solucion al problema con un array de fechas. La clase Solver, donde se implementa en algortimo goloso

view: La clase View que se usa para crear la interfaz grafica del programa

Explicacion basica del algoritmo:

Para cada partido del torneo se elige el mejor arbitro de hasta ese momento.

El criterio para el mejor arbitro es ver cual de todos aumenta menos el promedio del peor arbitro. Esto se hace con la func promedioPeorArbitro()

Asi recorre todos los partidos hasta llegar a una solucion

Aclaraciones:
El algoritmo esta mejor explicado paso a paso en la funcion resolver() de la clase Solver asi como tambien cada funcion.

Reconosco que la complejidad no es lo mejor (O n^7) por lo que pude calcular a ojo, sin embargo cuando me di cuenta ya era demasiado tarde. Creo que se puede hacer mucho mejor

El problema que me obligo a llegar a esa complejidad fue que el JSON me leia los equipos de cada fecha como un objeto diferente, por ej: el River que jugo en la fecha 2, era distinto de el de la fecha 1. Esto me obligo a hacer malabares en la funcion getEquipos

No se utilizo la metodologia MVP pero se logro separa en codigo de negocio de la interfaz visual