



CENTRO SAFA NUESTRA SEÑORA DE LOS REYES

DEPARTAMENTO DE INFORMÁTICA.

Proyecto Java 2 - Requisitos - V7

Profesor: Luis Javier López López

SEGUNDO PROYECTO JAVA

En esta versión del proyecto se seguirán trabajando los contenidos vistos anteriormente y además se introducirá y trabajará el concepto de interfaz, desarrollando y aplicándolo al proyecto.



Requisitos V7

1. Se ha de crear una clase llamada *UtilidadesPartida*.
2. En la carpeta modelos se han de crear los siguientes clases:

Jugador

- id -> int
- nickname-> String
- personajesFavoritos -> Set<Personajes>
- partidasGanadas -> Map<Personaje,Integer>

Partida

- id-> int
- codigo-> String
- server-> Double
- elecciones -> Map<Jugador,Personaje>
- jugadoresPorEquipo -> Map<Integer, Set<Jugador>>
- inicioPartida -> LocalDateTime
- finPartida -> LocalDateTime
- duracionPartida -> int (segundos)
- equipoVencedor ->Integer

3. Se ha de realizar una clase de tipo interfaz llamada IUtilidadesPartida. En ella se han de definir las cabeceras de los siguientes métodos, tal como se indica a continuación:
 - a. public void inicializarPartida(Partida partida, List<Jugadores> participantes, List<Personaje> personajesDisponibles)
 - b. public void finalizarPartida(Partida partida, Integer equipoVencedor)



4. En la clase UtilidadesPartida se ha de indicar que es una clase que implementa la interfaz IUtilidadesPartida, y hay que tener en cuenta que en dicha clase se implementarán los métodos de la interfaz , pero estos serán privados.

A continuación se detalla que debe de hacer cada uno de los métodos.

- a. `private void inicializarPartida(Partida partida, List<Jugadores> participantes, List<Personaje> personajesDisponibles)`
 - i. Este método inicializa los datos de la partida que se pasa como //parámetro de la siguiente manera:
 - ii. Inicializa el mapa de elecciones de la partida, asignando a cada jugador un personaje de entre los personajesDisponibles, eligiendo preferentemente uno de entre sus favoritos y si no uno aleatorio de los disponibles.
 - iii. Inicializa el mapa de jugadoresPorEquipo, asignando aleatoriamente cada jugador a uno de los equipos, sabiendo que el número de jugadores siempre es par y que siempre hay dos equipos.
 - iv. Establece la fecha de inicio de partida indicando el momento actual como el inicio de la partida.
- b. `private void finalizarPartida(Partida partida, Integer equipoVencedor)`
 - i. Este método establece los datos de la partida finalizada siguiendo las siguientes indicaciones.
 - ii. Establece la fecha fin de partida al momento actual .
 - iii. Establece la duración de la partida a la diferencia en segundos entre el inicio de partida y el fin de la misma.
 - iv. Establece el equipo vencedor al que se pasa como parámetro.
 - v. Para cada jugador que participa en dicha partida y es del equipo vencedor tiene que actualizarle el número de partidas ganadas en su atributo partidas ganadas teniendo en cuenta con el personaje con el que ha ganado la partida.