

En la unidad 3 de nuestro curso hemos aplicado mecanismos de reutilización como la **herencia**, el **polimorfismo** y el uso de **interfaces**. Estas técnicas enriquecen el comportamiento de las clases modeladas como parte de una solución en el diagrama de clases.

En la unidad 4, aprenderemos a utilizar el concepto de **matriz** como elemento de modelaje, pudiendo así agrupar elementos del modelo de la solución en una estructura contenedora de dos dimensiones de tamaño fijo; y hemos empleado **estructuras contenedoras lineales de tamaño variable** como elementos de modelado que permiten manejar una secuencia de objetos.

Esta tarea integradora presenta una actividad en la cual se requiere aplicar todos los conocimientos adquiridos hasta el momento. Por tanto, esta tarea es un instrumento para verificar el cumplimiento de los objetivos planteados para las unidades 3 y 4 descritos en el programa del curso. Para llevar a cabo este ejercicio es necesario realizar las actividades listadas a continuación:

## Actividades

Lleve a cabo las siguientes actividades de cada una de las etapas de desarrollo de software:

1. **Análisis del problema.** [Tabla de especificación y Lista de requerimientos funcionales](#) en el formato visto en la clase de Ingeniería de Software I.
2. **Diseño de la solución.**
  - a. Elabore un **diagrama de clases** que modele la solución del problema de acuerdo con las buenas prácticas y los patrones de diseño revisados hasta el momento en el curso. Su diagrama debe incluir el paquete modelo (model) y el de interfaz de usuario (ui).
  - b. **Trazabilidad del Análisis al Diseño.** Una [tabla a tres columnas](#) en la que se relaciona cada requerimiento con las clases y métodos que permiten satisfacer dicho requerimiento.
  - c. Recuerde que TODOS los artefactos generados en la fase de diseño deben **escribirse en inglés** y ubicarse en la carpeta doc de su proyecto
3. **Implementación en Java.** Incluya en la implementación, los comentarios descriptivos sobre los atributos y métodos de cada clase. Recuerde que TODOS los artefactos generados en la fase de implementación deben **escribirse en inglés**.
4. **Documentación en javadoc** (Debe entregarse el javadoc generado y ubicarlo en la carpeta API dentro de la carpeta doc).
5. **Usar GitHub como repositorio de código fuente y documentación** utilizando la estructura de carpetas aprendida en clase. Se **debe poder identificar commits en diferentes fechas**.

Recuerde que puede encontrar la **Rúbrica de la tarea integradora** en el siguiente [enlace](#).

**Nota:**

- Usted debe entregar la URL de su repositorio GitHub donde se deben encontrar los archivos de codificación en sus respectivos paquetes.
- Tenga en cuenta que su repositorio GitHub debe presentar una estructura base como por ejemplo:

**TorneoFutbol/**

**src/**

**bin/**

**doc/**

- Dentro de los directorios **src/** y **bin/** estarán presentes estos directorios (representando cada uno de sus paquetes):

**ui/**

**model/**

- El directorio **src** (source code) contiene sus clases .java dentro del directorio **ui/** y **model/**. Por otro lado, el directorio **bin** (binary files) contiene los archivos .class en el directorio **ui/** y **model/**. El directorio **doc** tendrá toda la documentación de análisis y diseño
- Su código debería compilar de acuerdo con lo explicado en la diapositiva 15 de esta presentación: <http://tinyurl.com/y3bd9bg2>

## Enunciado

Un torneo de fútbol es una competencia organizada en la que varios equipos de fútbol compiten entre sí para determinar un campeón. Un profesor de la Universidad que además es árbitro y le encanta el fútbol requiere diseñar un sistema que permita gestionar un campeonato de fútbol de clubes internacional (en un formato similar a la Copa América), dicho sistema deberá permitir registrar equipos y generar el fixture del torneo (calendario de juego), llevar registro de puntuación de equipos, goles, tarjetas amarillas y rojas, y designaciones arbitrales. En esta tarea integradora se desarrollará el sistema para gestionar la fase de grupos para un torneo de fútbol que consta de 8 equipos.

El fixture se utiliza para planificar el calendario de partidos de una temporada o torneo, y para determinar quiénes jugarán en cada ronda. También se utiliza para informar a los aficionados y a los medios de comunicación sobre los próximos partidos y para promocionar los eventos deportivos. En un torneo, se realizan diferentes rondas dependiendo de la cantidad de equipos, en este caso serán 4 rondas: fase de grupos, cuartos de final, semifinal y final. Al iniciar la ronda, se genera el calendario de los partidos de esa ronda. Una ronda se da por finalizada cuando se han ingresado los marcadores de todos sus partidos.

El torneo tiene fecha de inicio y de finalización para la programación de los partidos, y tendrá 8 equipos que se distribuirán de manera aleatoria (no como le tocó a Argentina en la Copa América 2024) en 2 grupos (cada grupo con 4 equipos), cada equipo deberá jugar una vez contra cada uno de los otros tres equipos del grupo, a esto se le conoce como fase de grupos. El sistema deberá realizar automáticamente el sorteo de los partidos a jugar en cada grupo por cada fecha, teniendo en cuenta que se deben jugar partidos intercalados en los grupos, cada equipo contará como mínimo con dos días de descanso entre partido. La última fecha de cada grupo (los dos partidos) se deben jugar a la vez.

De los equipos se debe registrar nombre, país al que pertenece, lista de 20 jugadores y nombre del director técnico. De los jugadores se debe registrar su número, nombre, país de origen y su posición (arquero, defensa, volante o delantero).

De los árbitros se necesita registrar un id, nombre, país de origen y tipo de árbitro (Central, Asistente), se requieren 4 árbitros centrales y 8 árbitros asistentes; por lo que el sistema debe garantizar el almacenamiento de esta información.

Para tener un control de los juegos y las estadísticas, el sistema debe permitir registrar lo siguiente:





A cada partido se asignarán 3 árbitros distribuidos como: 1 central y 2 asistentes. La asignación cumplirá la condición de que la nacionalidad de los árbitros no puede ser la misma del país al que pertenecen los equipos del partido al que se asignan.

Previo a cada partido, los delegados de los equipos deberán entregar la lista de los titulares para el partido compuesta por 11 jugadores, los demás podrán ser suplentes.

Posterior a cada partido se deben registrar las estadísticas del mismo, entre las cuales es necesario tener el marcador del partido con los autores de los goles y el asistidor, la cantidad de tarjetas recibidas por equipo; si es amarilla, es una amonestación; y si es roja, es una expulsión.

Tenga en cuenta que la información de la fase de grupos debe poder ser consultada en cualquier momento de la realización del torneo. Se debe emplear el siguiente formato matricial para cada Grupo.

Fig. 1. Información fase de grupos

Grupo A								
Club	PJ	G	E	P	GF	GC	DG	Pts
1  Argentina	3	3	0	0	5	0	5	9
2  Canadá	3	1	1	1	1	2	-1	4
3  Chile	3	0	2	1	0	1	-1	2
4  Perú	3	0	1	2	0	3	-3	1

**Convenciones:** PJ: partidos jugados, G: partidos ganados, E: partidos empatados, P: partidos perdidos, GF: goles a favor, GC: goles en contra, DG: diferencia de gol, Pts: puntos

En la fase de grupos, las posiciones de la tabla se calculan según el número de puntos alcanzados y éste se define de la siguiente manera: una victoria son 3 puntos, un empate es 1 punto y una derrota no da puntuación. En caso de haber empate en puntos, el orden de la clasificación se define con los siguientes ítems: primero el equipo con mayor diferencia de goles (se calcula como goles a favor menos goles en contra) si persiste el empate, se considerará goles a favor y sino, goles en contra.

Al finalizar el campeonato, se premiará al goleador (botín de oro) y al equipo con Fair Play (menor número de amonestaciones y expulsiones), se requiere que el sistema pueda mostrar esta información.

Luego de finalizar el campeonato, se requiere generar un indicador para los equipos, jugadores y árbitros. Para el equipo se calcula, de los partidos jugados porcentaje de los que ha ganado. Para el jugador, porcentaje de goles realizados versus partidos jugados. Para el árbitro, se calcula la cantidad de tarjetas colocadas dividido en la cantidad de partidos arbitrados (solo para centrales). Se requiere poder mostrar este indicador para algún equipo, jugador o arbitro particular.

En resumen, el prototipo a desarrollar debe ofrecer las siguientes funcionalidades:

**Respecto al registro de información base:**

1. Registrar un equipo.
2. Registrar un jugador.
3. Registrar un árbitro.
4. Precargar la información necesaria para iniciar un torneo: equipos, jugadores y árbitros.

**Respecto al fixture:**

5. Calcular y mostrar el sorteo (partidos) de la fase de grupos y las fechas respectivas de esta fase.

**Respecto a la gestión de los partidos:**

6. Asignar el equipo arbitral a un partido según su disponibilidad (nacionalidad).
7. Registrar el marcador de un partido.
8. Registrar jugador que marca gol y jugador que asiste.
9. Registrar tarjetas a jugadores.

**Respecto a las estadísticas del torneo:**

10. Consultar en cualquier momento la información de la fase de grupos (tabla de posiciones de todos los grupos).
11. Calcular y mostrar el máximo goleador del torneo.
12. Calcular y mostrar el equipo con Fair Play.
13. Calcular y mostrar la eficiencia de un equipo (número de partidos ganados respecto a jugados).
14. Calcular y mostrar la eficiencia de un jugador (número de goles respecto a partidos jugados).
15. Calcular y mostrar el índice de tarjetas de un árbitro central (número de tarjetas respecto a partidos arbitrados).

## Entregas

La entrega de la tarea integradora se encontrará dividida en dos partes:

**Primera entrega: Hasta el 26 de octubre de 2024 a las 21:59.**

- Su entrega debe contener el análisis y diseño del programa a desarrollar:
  - **Análisis:** Primera versión de la tabla de especificación del problema y el análisis y especificación de cada requerimiento.
  - **Diseño:** Primera versión del diagrama de clases y tabla de trazabilidad análisis-diseño.
  - **Implementación:** Todas las clases y métodos necesarios para solucionar los primeros 7 requerimientos funcionales.
  - **Documentación:** Javadoc generado de los métodos necesarios para solucionar los primeros 7 requerimientos funcionales.

**Segunda entrega: Hasta el 16 de noviembre de 2023 a las 21:59.**

- Su entrega debe contener el análisis y diseño del programa a desarrollar:
  - **Análisis:** Versión final de la tabla de especificación del problema y el análisis y especificación de cada requerimiento.
  - **Diseño:** Versión final del diagrama de clases y la tabla de trazabilidad análisis-diseño.
  - **Implementación:** Todas las clases y métodos necesarios para solucionar TODOS los requerimientos funcionales.
  - **Documentación:** Javadoc generado de los métodos necesarios para solucionar TODOS los requerimientos funcionales.

## Sustentación

Se realizará una sustentación individual ante sus respectivos profesores de Algoritmos y Programación I.