

# Actividad Intra-Aula 02

Gilio H. Linfati G. Autor, *Estudiante Ingeniería Civil Informática, Facultad de Ingeniería y Ciencias (Universidad de la Frontera, g.linfati02ufromail.cl)*

**Resumen—** Este documento se centrará en la explicación de un modelo UML relacionado con un proyecto web para la promoción del juego FIFA19, como también del porqué se tomó la decisión de las relaciones usadas entre clases.

## I. CONTEXTO PROBLEMA

EA sports, empresa creadora de los videojuegos FIFA, solicita el desarrollo de una web para la promoción del juego donde interesados y aficionados del mismo puedan acceder a los datos de todos los jugadores existentes en su versión 2019. Se requiere: (1) que la web permite listar todos los jugadores disponibles (fotografía, nombre, nacionalidad, rating, edad, club y potencial); (2) listar jugadores con nacionalidad indicada, (3) listar jugadores menores a edad indicada, (4) listar jugadores por club indicado, (5) listar jugadores con rating mayor al indicado, (6) listar por potencial mayor al indicado, (7) buscar un jugador por nombre, (8) desplegar una página con todos los datos del jugador al clicar sobre este, (9) comparar 2 jugadores (fotografía, nombre, nacionalidad, rating, edad, club, potencial, velocidad, control del balón, agilidad, resistencia) y (10) una página de bienvenida donde se muestran los datos del juego, el precio y dónde comprarlo.

## II. SOLUCIÓN PROPUESTA

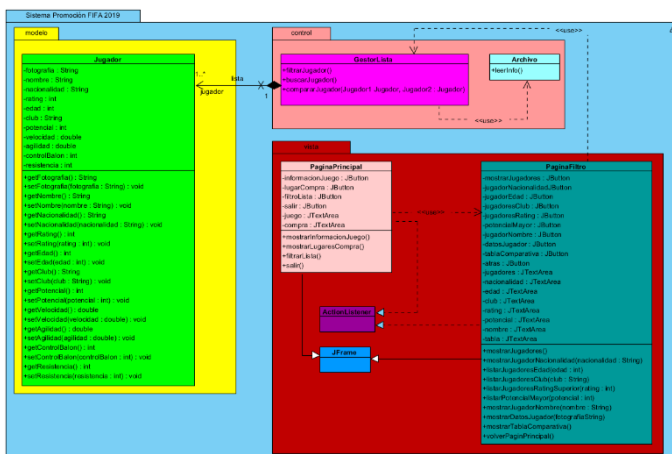


Fig. 1 Diagrama UML desarrollado.

La imagen adjunta de la figura 1 representa el modelo propuesto con el cuál se piensa dar solución a la problemática planteada.

En dicha solución que contemplan 3 package diferentes, modelo, control y vista, y las clases presentes son:

1. **Jugador**: esta clase se utiliza para almacenar la información de la lista de los jugadores.
2. **Archivo**: esta clase se encargará de permitirnos leer los archivos presentes en dónde se encontrará guardada la información de las listas.
3. **GestorLista**: esta clase de permitirnos buscar a un jugador en específico, buscar uno o más mediante un filtro y comparar a dos jugadores distintos. Se compone de la clase **Jugador**, pues almacena la información de ella para realizar las búsquedas pertinentes. También utiliza la clase **Archivo** para poder abrir los archivos a analizar.
4. **PaginaPrincipal**: esta clase se encargará de desplegar el primer menú de opciones con el cuál el usuario podrá interactuar, permitiéndole ver la información del juego, los lugares en dónde comprarlo, entre otras opciones más. Hereda de **JFrame** para usar las ventanas, implementa **ActionListener** para manejar eventos y utiliza **PaginaFiltro**.
5. **PaginaFiltro**: esta clase es la encargada de desplegar el segundo menú de opciones con el cual podrá interactuar el

usuario, permitiéndole realizar una búsqueda de uno o más jugadores mediante un filtro, como puede serlo la edad, nombre, club al cual pertenece, entre otros más. Hereda de **JFrame** para utilizar ventanas, implementa **ActionListener** para poder manejar los eventos y utiliza **GestorArchivo** para realizar las búsquedas de jugadores, mediante filtros o no.

6. JFrame: clase que nos permite generar ventanas sobre las cuales añadir distintos objetos con los que podrá interactuar o no el usuario.
7. ActionListener: interfaces de los grupos de los listener(escuchadores), que nos permitirá manejar los eventos implementados en los botones.

### III. CONCLUSIÓN

Uno de los factores más importantes a considerar al momento de crear software, es referente a la ingeniería detrás del mismo, pues nos permite establecer las relaciones y el comportamiento de cada una de las partes para así optimizar el rendimiento del mismo y de paso poder crear algo de calidad, disminuyendo al mínimo la posibilidad de errores.

Durante el desarrollo del documento se detalló el porqué de las decisiones tomadas al momento de establecer la relación entre las clases creadas para poder dar solución a la problemática que se intentaba resolver.

Cabe destacar qué, si bien esta no es la única forma de solucionar lo que se buscaba, para el momento en el cual se realizó el diagrama y el informe, fue la más conveniente que se pensó en dicho momento.

