[Introducción 2](#_Toc528202117)

[Objetivo 2](#_Toc528202118)

[Los 4 paquetes 2](#_Toc528202119)

[Lógica 3](#_Toc528202120)

[Presentación 3](#_Toc528202121)

[Persistencia 5](#_Toc528202122)

[Fachada 5](#_Toc528202123)

[Motor de búsqueda 5](#_Toc528202124)

[Simulador 6](#_Toc528202125)

# Introducción

En este informe haremos una guía acerca del funcionamiento interno del programa. Lo acompañaremos con los diagramas correspondientes y una explicación de cada clase y la interacción entre ellas.

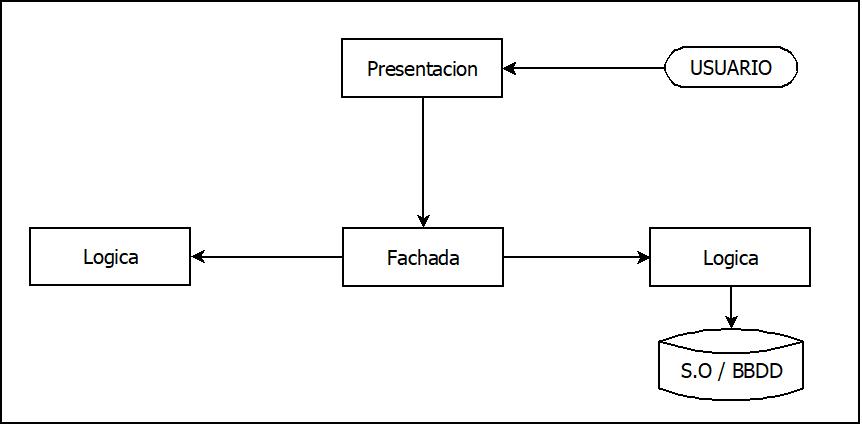
El programa se basa en el mundial Rusia 2018, contemplamos los aspectos más relevantes, como Jugadores, directores técnicos, jueces, selecciones, fases y partidos. Consiguiendo de esta forma un programa completo que nos permita abarcar todos los aspectos mas importantes del mundial otorgando una experiencia al usuario óptima.

Pasaremos a una explicación de cada paquete, la interacción entre ellos, su funcionalidad y composición.

# Objetivo

El objetivo de la practica es la correcta explicación del funcionamiento interno del programa, de esta forma el lector estará capacitado de poder entender y recrear dicho programa.

# Los 4 paquetes

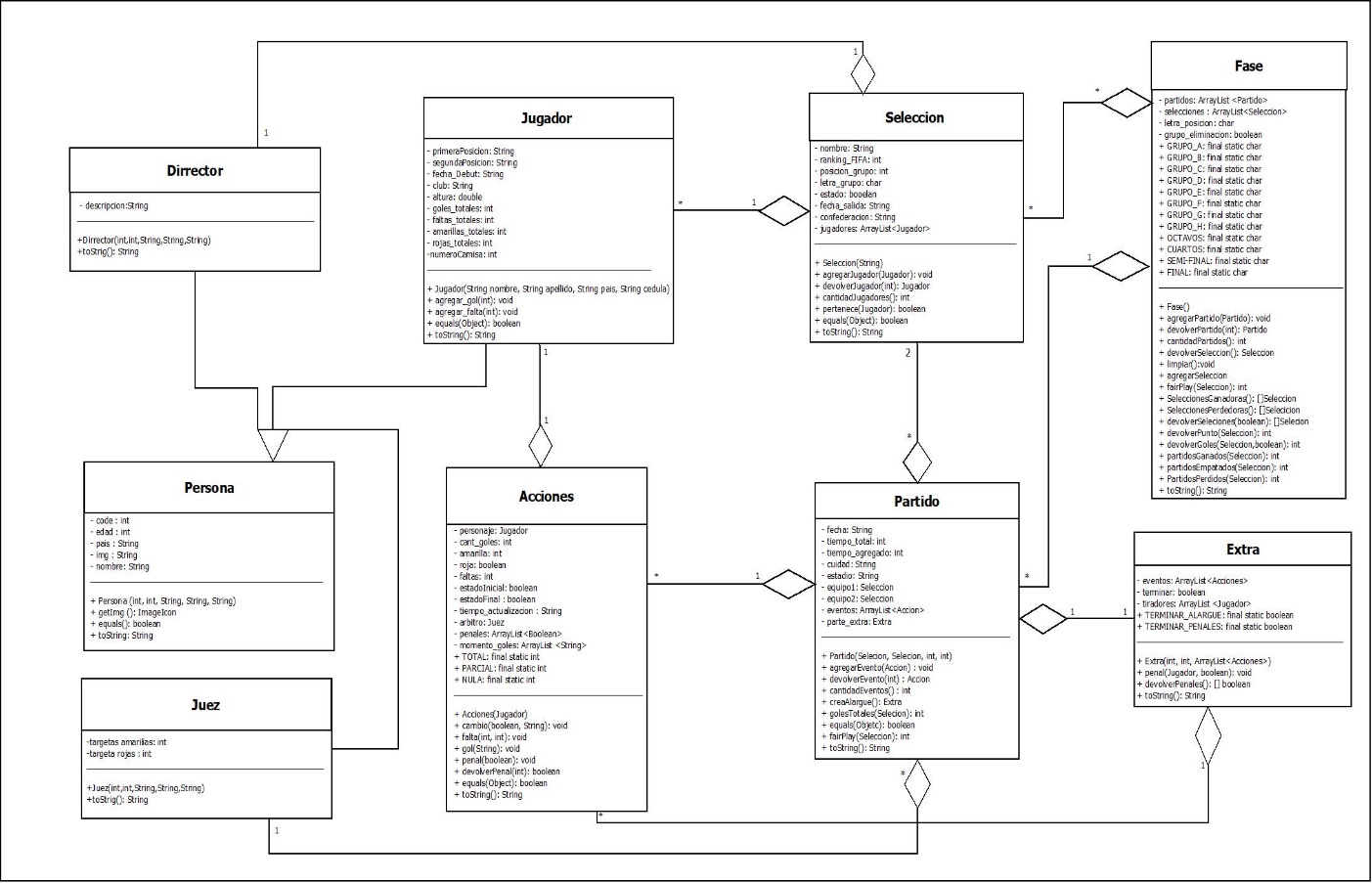
La función de los 4 paquetes ya fue mencionada en el curso por lo tanto no haremos hincapié. Solo mencionaremos que la clase **presentación** es la encargada de mostrar la GUI, **persistencia** de comunicarse con el S.O o la BBDD, **lógica** de darnos los tipos de datos y métodos con lo que trabajar y por último **fachada** conecta a los demás paquetes.

El paquete fachada posee una sola clase llamada Fachada, esta tiene dos funcionalidades, almacena todos los datos temporales en ArrayList y darnos métodos para operar con todos los datos. Existen métodos que para funcionar la clase necesita de disponer de todos los datos del mundial, ninguna de las clases de lógica cumple con dicho requisito, por lo tanto, fachada será la encargada.

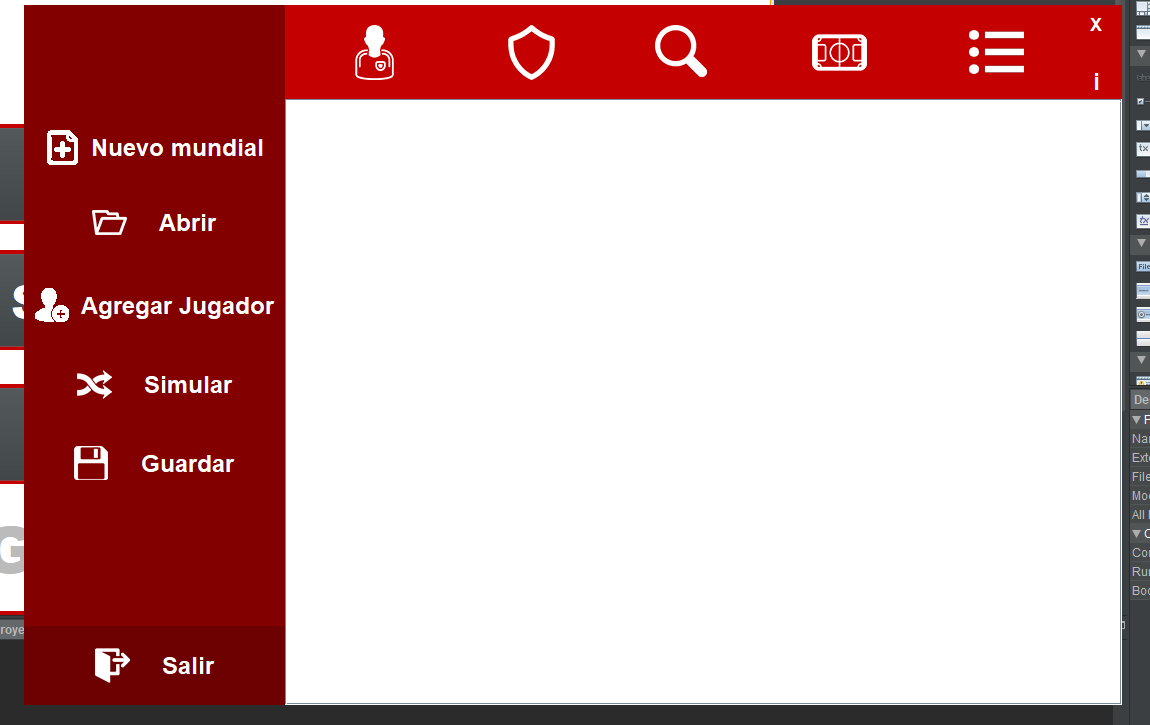
Ahora ya explicada la función de cada clase pasaremos a una descripción mas profunda.

# Lógica

El paquete lógico nos proporcionará las clases con las funcionaran nuestro programa como tipo de objeto, todos los datos que utilizaremos se encuentran empaquetados en los atributos de estas clases. Cada clase interactúa entre si a través de métodos y atributos dan doce casos de agregación y herencia. En el siguiente diagrama de implementación se explica todo a profundidad. (En el final del informe se encuentra una foto más grande)



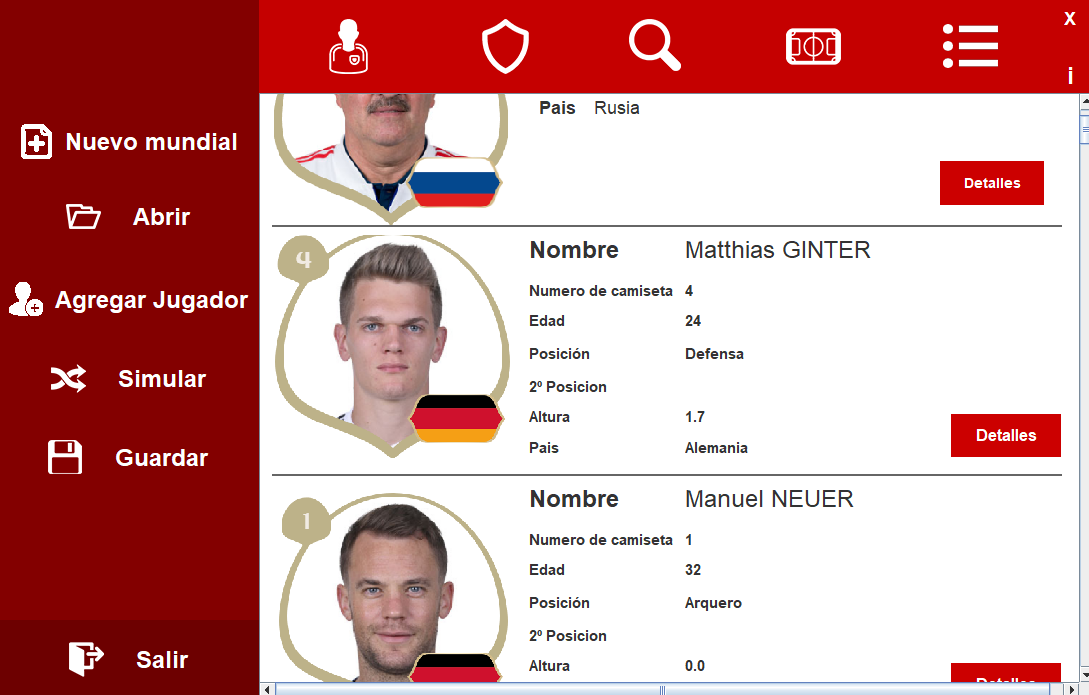
# Presentación

En este paquete podemos encontrar las clases que conforman la GUI. Además es el paquete con más número de clases, teniendo un número 16 clases. Comenzaremos por explicar la clase principal.

El bloque blanco es un JPanel llamado ***displayInterno***, este cumple una función muy importante ya que será el encargado de contener a los demás JPanel del paquete. Cada ves que despliega un nuevo panel el contendió de displayInterno es eliminado y luego sustituido por el nuevo panel, luego se repinta el panel. De esta tarea se encarga el método ***cargarPanel(JPanel)*** de la clase Principal. Es vital que cualquier clase puede desplegar una ventana, por lo tanto, se hiso a la clase principal un **Singleton**.

En otra ventana que tenemos que hacer hincapié es **Menujugador** ya que esta clase tiene dos peculiaridades, una de ellas es que fue escrita a mano (el motivo se explicara en breve) y la otra es que se construye con un conjunto de **ListaJugador** y **ListaDirector**. La función de esta clase es la de mostrar a los jugadores y directores técnicos del mundial, para eso se crea un **ListaJugador**  para cada jugador y un **ListaDirector** para cada director. Cada uno pide en el constructor su correspondiente objeto, con el cual modificará los textos del JPanel para sus mostrar sus datos.

El constructor del JPanel se encarga de ver el numero de jugador y de directores. Luego le da un tamaño al panel de 750 px, 260 px \* el número de jugadores y directores. Luego pone una disposición GridLayout (1, número de jugadores y directores) para que quede en forma de lista.



Por último, en presentación hablaremos de **menuBusqueda** ya que el funcionamiento es muy similar al **menuJugador**, este se compone de un JTextField para ingresar la búsqueda, un JCombobox para ingresar el criterio de búsqueda, un JButton para buscar y por último un JPanel que muestra los resultados de la búsqueda, el funcionamiento de ese panel es idéntico a cómo funciona **menuJugador**, solo que para mostrar los resultados utiliza **resultadoBusquedaPersona** y **resultadoBusquedaSeleccion**.

Lo demás de la capa de presentación no presenta dificultad que amerite una explicación adicional que el **javadoc** ya no haya explicado.

# Persistencia

Cuando usamos ObjectInput/outputStream este solo puede abrir/guardar en un solo archivo, para no tener que crear un archivo para cada ArrayList crearemos en lógica una clase llamada **Mundial**, que se encarga de almacenar cada uno de los ArrayList de fachada como sus atributos. En la única clase de persistencias llamada Archivo, a la hora de guardar o de abrir crea una instancia de tipo mundial. donde se vuelcan todos los ArrayList de fachada. A la hora de abrir se castea el object que nos devuelve readObject a Mundial y se carga en Fachada.

Por un tema estético se eligió el look and feel de Windows para el JFileChooser. Además se estableció como filtro de extensión para los archivos que se van a abrir “.mun”: este será la extensión para los archivos de nuestro programa.

# Fachada

La clase fachada es la encargada de conectar los 3 paquetes anteriormente descritos, teniendo para ello infinidad de métodos que simplemente sirven para conectar un paquete con otro.

La fachada cuenta con un conjunto de métodos que operan con los datos, estos no se pueden crear en las clases del paquete Lógica, ya que operan con la totalidad de los datos. La mayoría de estos métodos son utilizados para buscar información, aunque no profundizaremos en eso.

También tenemos una gama de métodos para trabajar con los ArrayList, como lo es limpiarArrays, quien elimina el contenido de todos los Arrays, también tenemos que mencionar a Sistema Base, que se encarga de cargar los datos predeterminados del programa.

**COMO A PODIDO OBSERVAR NO HEMOS MENCIONADO NI A SIMULAR DATOS NI AL MOTOR DE BÚSQUEDA, ESOS TIENE UN AREA APARTE QUE TRABAJAREMOS AHORA.**

# Motor de búsqueda

El motor búsqueda cuenta con 12 criterios de búsqueda, cada uno con su correspondiente método en Fachada. Cada criterio de búsqueda tiene una constante estática asociada que le permite recibir consultas externas por medio de otras interfaces, como puede ser el caso de descripción jugador.

Cada criterio de búsqueda ejecuta su correspondiente método en fachada, quien puede devolver un ArrayList de personas, jugadores o selecciones, los dos primeros se cargan en un JPanel llamado **resultadoBusquedaPersona**, el último en **resultadoBusquedaSeleccion**. Los paneles que representen los objetos que concordaron en la búsqueda son impresos en el JPanel.

# Simulador

Previamente para el mundial Rusia 2018 se generaron 4 bombos que se muestran en la siguiente imagen



Para generar un mundial debemos de mezclar los 4 bombos de tal forma que los grupos tengan uno de cada uno, siempre y cuando dos selecciones no pertenezcan a la misma confederación, por excepción de UEFA, ya que 2 selecciones de UEFA pueden pertenecer al mismo grupo.

Para eso en el programa cada vez que se hace una simulación cargamos los bombos y eliminamos el contenido de los grupos. El programa comienza a cargar los grupos y carga el resultado en los grupos correspondientes. Completando así la simulación.

**OBSERVACIÓN**: a la hora de generar los grupos existe la posibilidad que no lo logre a la primera, ya que cuando combina las selecciones puede ser que en dos bombos solo quede una selección que repetir y que ambas pertenezcan a una confederación que no sea UEFA, si eso ocurre se deberá repetir la simulación hasta que tenga éxito. Luego de haber mejorado el algoritmo logramos reducir el promedio de errores hasta 4 ± 2 .