

APRENDICES DE JOSUKA C.F

Generado por Doxygen 1.9.0

Capítulo 1

Indice de namespaces

1.1. Paquetes

Aquí van los paquetes con una breve descripción (si está disponible):

Aprendices_de_Josuka.Aprendices_de_Josuka	
Clases para testear	??
Aprendices_de_Josuka.LD	
Todos los objetos que tendrá la aplicación	??
Aprendices_de_Josuka.LP	
Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores	??
AprendicesDeJosukaPartidos	??
AprendicesDeJosukaSanciones	??
Controller	
Necesario para incorporar el patron de diseño del controller	??
DAO	
En este paquete estan incluidas todas las clases que tienen relacion con la base de datos	??
Fachada	
Necesario para incorporar el patrón de diseño de fachada. El servidor de la aplicación	??
Gateway	
La aplicación se conectará al servicio externo de la liga, de la cual obtendrá los partidos y las sanciones	??
python	??
python.Partido	??
python.Partido.Equipo	??
python.Partido.Partido	??
python.Partido.Sancion	??
python.Partidos	??
python.Partidos.Interface_Partidos	??
python.Partidos.Interface_Sanciones	??
Remote	
Necesario para incorporar el patrón de diseño de serice locator	??
ServiciosExternos	
Clases para hacer posible la conexión con la raspberry y el parseo de la información	??

Capítulo 2

Índice jerárquico

2.1. Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

Aprendices_de_Josuka.LD.Administrador	??
Aprendices_de_Josuka.LD.Categoria	??
Controller.Controller	??
DAO.DAO	??
ServiciosExternos.DF_Printable	??
ServiciosExternos.Partidos_parameters	??
ServiciosExternos.Sanciones_parameters	??
python.Partido.Equipo.Equipo	??
Gateway.itfGateway	??
Gateway.Gateway	??
python.Partido.Partido.Partido	??
ServiciosExternos.Partido_JSON	??
ServiciosExternos.RestClient< T extends DF_Printable >	??
ServiciosExternos.RestClient< ServiciosExternos.Partidos_parameters >	??
python.Partido.Sancion.Sancion	??
ServiciosExternos.Sancion_JSON	??
Remote.ServiceLocator	??
Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO	??
Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada	??
Aprendices_de_Josuka.Aprendices_de_Josuka.testPartidos	??
Aprendices_de_Josuka.LD.Tipo_Material	??
Aprendices_de_Josuka.LD.Tipo_Sancion	??
ABC	
python.Partidos.Interface_Partidos.Interface_Partidos	??
python.Partidos.Interface_Sanciones.Interface_Sanciones	??
Interface_Partidos	
AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos	??
Interface_Sanciones	
AprendicesDeJosukaSanciones.AprendicesDeJosukaSanciones	??
JFrame	
Aprendices_de_Josuka.LP.Anyadir_Inventario	??
Aprendices_de_Josuka.LP.Clasificacion	??
Aprendices_de_Josuka.LP.Clasificacion_Entrenador	??
Aprendices_de_Josuka.LP.Clasificacion_Jugador	??
Aprendices_de_Josuka.LP.EditarDatos	??

Aprendices_de_Josuka.LP.EditarDatosEntrenador	??
Aprendices_de_Josuka.LP.EditarEntrenador	??
Aprendices_de_Josuka.LP.EditarEquipo	??
Aprendices_de_Josuka.LP.EditarJugador	??
Aprendices_de_Josuka.LP.Estadisticas	??
Aprendices_de_Josuka.LP.EstadisticasEntrenador	??
Aprendices_de_Josuka.LP.FichaEntrenador	??
Aprendices_de_Josuka.LP.FichaJugador	??
Aprendices_de_Josuka.LP.IniciarSesion	??
Aprendices_de_Josuka.LP.IniciarSesion_Admin	??
Aprendices_de_Josuka.LP.Pantalla_Principal	??
Aprendices_de_Josuka.LP.Principal_Administrador	??
Aprendices_de_Josuka.LP.Principal_Entrenador	??
Aprendices_de_Josuka.LP.Principal_Jugador	??
Aprendices_de_Josuka.LP.RegistrarAdmin	??
Aprendices_de_Josuka.LP.RegistrarEntrenador	??
Aprendices_de_Josuka.LP.RegistrarEquipo	??
Aprendices_de_Josuka.LP.RegistrarJugador	??
Aprendices_de_Josuka.LP.RegistrarMaterial	??
Aprendices_de_Josuka.LP.Visualizar_Entrenadores	??
Aprendices_de_Josuka.LP.Visualizar_Equipos	??
Aprendices_de_Josuka.LP.Visualizar_Jugadores	??
Aprendices_de_Josuka.LP.Visualizar_Mis_Jugadores	??
JPanel	
Aprendices_de_Josuka.LP.Panel_Entrenador	??
Aprendices_de_Josuka.LP.Panel_Equipos	??
Aprendices_de_Josuka.LP.Panel_Jugador	??
Aprendices_de_Josuka.LP.Panel_Partidos	??
Aprendices_de_Josuka.LP.Panel_Sanciones	??
Remote	
Fachada.itfFachada	??
Fachada.ServidorPrincipal	??
Serializable	
Aprendices_de_Josuka.LD.Entrenador	??
Aprendices_de_Josuka.LD.Equipo	??
Aprendices_de_Josuka.LD.Equipos_Ext	??
Aprendices_de_Josuka.LD.Jugador	??
Aprendices_de_Josuka.LD.Material	??
Aprendices_de_Josuka.LD.Partido	??
Aprendices_de_Josuka.LD.Persona	??
Aprendices_de_Josuka.LD.Sancion	??
UnicastRemoteObject	
Fachada.ServidorPrincipal	??

Capítulo 3

Índice de clases

3.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Aprendices_de_Josuka.LD.Administrador	??
Esta clase define objetos de tipo administrador	??
Aprendices_de_Josuka.LP.Anyadir_Inventario	??
Esta clase define la GUI de la ventana de anyadir inventario del administrador	??
AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos	??
AprendicesDeJosukaSanciones.AprendicesDeJosukaSanciones	??
Aprendices_de_Josuka.LD.Categoria	??
Esta clase define objetos de tipo Categoria mediante un enumerado	??
Aprendices_de_Josuka.LP.Clasificacion	??
En esta clase se desarrolla la GUI donde se muestra la clasificacion al administrador	??
Aprendices_de_Josuka.LP.Clasificacion_Entrenador	??
En esta clase se desarrolla la GUI donde se muestra la clasificacion al entrenador	??
Aprendices_de_Josuka.LP.Clasificacion_Jugador	??
En esta clase se desarrolla la GUI donde se muestra la clasificacion al jugador	??
Controller.Controller	??
DAO.DAO	??
ServiciosExternos.DF_Printable	??
Interfaz con método print que será implementado por Partidos_parameters y Sanciones_parameters para visualizar el contenido de las listas	??
Aprendices_de_Josuka.LP.EditarDatos	??
En esta clase se desarrolla la GUI para que el jugador pueda modificar su correo o su contraseña	??
Aprendices_de_Josuka.LP.EditarDatosEntrenador	??
En esta clase se desarrolla la GUI para que el entrenador pueda modificar su correo o su contraseña	??
Aprendices_de_Josuka.LP.EditarEntrenador	??
En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del entrenador	??
Aprendices_de_Josuka.LP.EditarEquipo	??
En esta clase se desarrolla la GUI para que el administrador pueda parte de la informacion del equipo	??
Aprendices_de_Josuka.LP.EditarJugador	??
En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del jugador	??
Aprendices_de_Josuka.LD.Entrenador	??
Esta clase define objetos de tipo Entrenador	??

Aprendices_de_Josuka.LD.Equipo	
Esta clase define objetos de tipo equipo	??
python.Partido.Equipo.Equipo	??
Aprendices_de_Josuka.LD.Equipos_Ext	
Esta clase define objetos de tipo equipo que no están guardados en la base de datos y se obtienen de la raspberry	??
Aprendices_de_Josuka.LP.Estadísticas	
En esta clase se desarrolla la GUI donde se muestran las estadísticas del jugador	??
Aprendices_de_Josuka.LP.EstadísticasEntrenador	
En esta clase se desarrolla la GUI donde se muestran las estadísticas del entrenador	??
Aprendices_de_Josuka.LP.FichaEntrenador	
En esta clase se desarrolla la GUI donde se muestra al entrenador su ficha	??
Aprendices_de_Josuka.LP.FichaJugador	
En esta clase se desarrolla la GUI donde se muestra al jugador su ficha	??
Gateway.Gateway	??
Aprendices_de_Josuka.LP.IniciarSesion	
En esta clase se desarrolla la GUI para el inicio de sesion de los jugadores y entrenadores ya registrados	??
Aprendices_de_Josuka.LP.IniciarSesion_Admin	
En esta clase se desarrolla la GUI para el inicio de sesion del administrador	??
python.Partidos.Interface_Partidos.Interface_Partidos	??
python.Partidos.Interface_Sanciones.Interface_Sanciones	??
Fachada.itfFachada	
Interfaz que contiene todos los métodos que implementará el servidor	??
Gateway.itfGateway	
Interfaz del Gateway donde se establecen todos los métodos que tendrá el Gateway	??
Aprendices_de_Josuka.LD.Jugador	
Esta clase define los objetos de tipo jugador	??
Aprendices_de_Josuka.LD.Material	
Esta clase define los objetos de tipo material, es decir, del inventario	??
Aprendices_de_Josuka.LP.Panel_Entrenador	
En esta clase se crea un panel con la informacion del entrenador, para luego poder introducirlo en otras ventanas	??
Aprendices_de_Josuka.LP.Panel_Equipos	
En esta clase se crea un panel para mostrar por pantalla cada equipo, para luego poder introducirlo en otras ventanas	??
Aprendices_de_Josuka.LP.Panel_Jugador	
En esta clase se crea un panel para mostrar por pantalla cada jugador, para luego poder introducirlo en otras ventanas	??
Aprendices_de_Josuka.LP.Panel_Partidos	
En esta clase se crea un panel para mostrar por pantalla cada partido, para luego poder introducirlo en otras ventanas	??
Aprendices_de_Josuka.LP.Panel_Sanciones	??
Aprendices_de_Josuka.LP.Pantalla_Principal	
En esta clase se desarrolla la GUI para la pantalla principal de la aplicacion.	
??	
Aprendices_de_Josuka.LD.Partido	
Esta clase define objetos de tipo partido que no están guardados en la base de datos y se obtienen de la raspberry	??
python.Partido.Partido.Partido	??
ServiciosExternos.Partido_JSON	
Clase para parsear los partidos de la raspberry	??
ServiciosExternos.Partidos_parameters	
Clase que contiene los parametros de los partidos y sus getters y setters	??
Aprendices_de_Josuka.LD.Persona	
Esta clase define objetos de tipo persona (clase padre), de la que después heredarán las clases entrenador y jugador (las clases hijas)	??

Aprendices_de_Josuka.LP.Principal_Administrador	
En esta clase se desarrolla la GUI que visualiza el administrador cuando abre la aplicacion despues de haber iniciado sesion	??
Aprendices_de_Josuka.LP.Principal_Entrenador	
En esta clase se desarrolla la GUI que visualiza el entrenador cuando abre la aplicacion despues de haber iniciado sesion	??
Aprendices_de_Josuka.LP.Principal_Jugador	
En esta clase se desarrolla la GUI que visualiza el jugador cuando abre la aplicacion despues de haber iniciado sesion	??
Aprendices_de_Josuka.LP.RegistrarAdmin	
En esta clase se desarrolla la GUI para el registro del administrador	??
Aprendices_de_Josuka.LP.RegistrarEntrenador	
En esta clase se desarrolla la GUI para el registro del entrenador	??
Aprendices_de_Josuka.LP.RegistrarEquipo	
En esta clase se desarrolla la GUI para cuando un administrador quiera registrar un equipo . .	??
Aprendices_de_Josuka.LP.RegistrarJugador	
En esta clase se desarrolla la GUI para el registro del jugador	??
Aprendices_de_Josuka.LP.RegistrarMaterial	
En esta clase se desarrolla la GUI para que el administrador pueda registrar el material	??
ServiciosExternos.RestClient< T extends DF_Printable >	
Sirve para conectar dos aplicaciones. En este caso la raspberry y la aplicación de Aprendices de Josuka	??
Aprendices_de_Josuka.LD.Sancion	
Esta clase define objetos de tipo sancion que no están guardados en la base de datos y se obtienen de la raspberry	??
python.Partido.Sancion.Sancion	??
ServiciosExternos.Sancion_JSON	
Clase para parsear las sanciones de la raspberry	??
ServiciosExternos.Sanciones_parameters	
Clase que contiene los atributos de las sanciones, con sus getters y setters mediante el plugin de Lombok. Implementa DF_Printable para imprimir el contenido de cada sancion	??
Remote.ServiceLocator	
Esta clase establece la conexion entre la fachada y controller	??
Fachada.ServidorPrincipal	
En esta clase se ejecutarán o delegarán todas las funciones de la aplicación	??
Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO	
Clase donde se hara el testeo de la base de datos	??
Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada	
Clase donde se hara el testeo de la fachada (gestor)	??
Aprendices_de_Josuka.Aprendices_de_Josuka.testPartidos	
Clase donde se hara el testeo con mockito	??
Aprendices_de_Josuka.LD.Tipo_Material	
Esta clase define un enumerado para el atributo tipo de la clase Material	??
Aprendices_de_Josuka.LD.Tipo_Sancion	
Esta clase define un enumerado para el atributo tipo de la clase Sancion	??
Aprendices_de_Josuka.LP.Visualizar_Entrenadores	
En esta clase se desarrolla la GUI para que el administrador pueda visualizar los entrenadores	??
Aprendices_de_Josuka.LP.Visualizar_Equipos	
En esta clase se desarrolla la GUI para que el administrador pueda visualizar los equipos . . .	??
Aprendices_de_Josuka.LP.Visualizar_Jugadores	
En esta clase se desarrolla la GUI para que los administradores puedan ver los jugadores . .	??
Aprendices_de_Josuka.LP.Visualizar_Mis_Jugadores	
En esta clase se desarrolla la GUI que se encarga de mostrar a cada entrenador sus jugadores	??

Capítulo 4

Indice de archivos

4.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

src/main/java/Aprendices_de_Josuka/LD/Administrador.java	??
src/main/java/Aprendices_de_Josuka/LD/Categoria.java	??
src/main/java/Aprendices_de_Josuka/LD/Entrenador.java	??
src/main/java/Aprendices_de_Josuka/LD/Equipo.java	??
src/main/java/Aprendices_de_Josuka/LD/Equipos_Ext.java	??
src/main/java/Aprendices_de_Josuka/LD/Jugador.java	??
src/main/java/Aprendices_de_Josuka/LD/Material.java	??
src/main/java/Aprendices_de_Josuka/LD/Partido.java	??
src/main/java/Aprendices_de_Josuka/LD/Persona.java	??
src/main/java/Aprendices_de_Josuka/LD/Sancion.java	??
src/main/java/Aprendices_de_Josuka/LD/Tipo_Material.java	??
src/main/java/Aprendices_de_Josuka/LD/Tipo_Sancion.java	??
src/main/java/Aprendices_de_Josuka/LP/Anyadir_Inventario.java	??
src/main/java/Aprendices_de_Josuka/LP/Clasificacion.java	??
src/main/java/Aprendices_de_Josuka/LP/Clasificacion_Entrenador.java	??
src/main/java/Aprendices_de_Josuka/LP/Clasificacion_Jugador.java	??
src/main/java/Aprendices_de_Josuka/LP/EditarDatos.java	??
src/main/java/Aprendices_de_Josuka/LP/EditarDatosEntrenador.java	??
src/main/java/Aprendices_de_Josuka/LP/EditarEntrenador.java	??
src/main/java/Aprendices_de_Josuka/LP/EditarEquipo.java	??
src/main/java/Aprendices_de_Josuka/LP/EditarJugador.java	??
src/main/java/Aprendices_de_Josuka/LP/Estadisticas.java	??
src/main/java/Aprendices_de_Josuka/LP/EstadisticasEntrenador.java	??
src/main/java/Aprendices_de_Josuka/LP/FichaEntrenador.java	??
src/main/java/Aprendices_de_Josuka/LP/FichaJugador.java	??
src/main/java/Aprendices_de_Josuka/LP/IniciarSesion.java	??
src/main/java/Aprendices_de_Josuka/LP/IniciarSesion_Admin.java	??
src/main/java/Aprendices_de_Josuka/LP/Panel_Entrenador.java	??
src/main/java/Aprendices_de_Josuka/LP/Panel_Equipos.java	??
src/main/java/Aprendices_de_Josuka/LP/Panel_Jugador.java	??
src/main/java/Aprendices_de_Josuka/LP/Panel_Partidos.java	??
src/main/java/Aprendices_de_Josuka/LP/Panel_Sanciones.java	??
src/main/java/Aprendices_de_Josuka/LP/Pantalla_Principal.java	??
src/main/java/Aprendices_de_Josuka/LP/Principal_Administrador.java	??
src/main/java/Aprendices_de_Josuka/LP/Principal_Entrenador.java	??

src/main/java/Aprendices_de_Josuka/LP/Principal_Jugador.java	??
src/main/java/Aprendices_de_Josuka/LP/RegistrarAdmin.java	??
src/main/java/Aprendices_de_Josuka/LP/RegistrarEntrenador.java	??
src/main/java/Aprendices_de_Josuka/LP/RegistrarEquipo.java	??
src/main/java/Aprendices_de_Josuka/LP/RegistrarJugador.java	??
src/main/java/Aprendices_de_Josuka/LP/RegistrarMaterial.java	??
src/main/java/Aprendices_de_Josuka/LP/Visualizar_Entrenadores.java	??
src/main/java/Aprendices_de_Josuka/LP/Visualizar_Equipos.java	??
src/main/java/Aprendices_de_Josuka/LP/Visualizar_Jugadores.java	??
src/main/java/Aprendices_de_Josuka/LP/Visualizar_Mis_Jugadores.java	??
src/main/java/Controller/Controller.java	??
src/main/java/DAO/DAO.java	??
src/main/java/Fachada/itfFachada.java	??
src/main/java/Fachada/ServidorPrincipal.java	??
src/main/java/Gateway/Gateway.java	??
src/main/java/Gateway/itfGateway.java	??
src/main/java/Remote/ServiceLocator.java	??
src/main/java/ServiciosExternos/DF_Printable.java	??
src/main/java/ServiciosExternos/Partido_JSON.java	??
src/main/java/ServiciosExternos/Partidos_parameters.java	??
src/main/java/ServiciosExternos/RestClient.java	??
src/main/java/ServiciosExternos/Sancion_JSON.java	??
src/main/java/ServiciosExternos/Sanciones_parameters.java	??
src/main/python/__init__.py	??
src/main/python/Partido/__init__.py	??
src/main/python/Partido/Equipo.py	??
src/main/python/Partido/Partido.py	??
src/main/python/Partido/Sancion.py	??
src/main/python/Partidos/__init__.py	??
src/main/python/Partidos/Interface_Partidos.py	??
src/main/python/Partidos/Interface_Sanciones.py	??
src/main/python/Partidos/ADJPartidos/AprendicesDeJosukaPartidos.py	??
src/main/python/Partidos/ADJPartidos/AprendicesDeJosukaSanciones.py	??
src/test/java/Aprendices_de_Josuka/Aprendices_de_Josuka/testDAO.java	??
src/test/java/Aprendices_de_Josuka/Aprendices_de_Josuka/testFachada.java	??
src/test/java/Aprendices_de_Josuka/Aprendices_de_Josuka/testPartidos.java	??

Capítulo 5

Documentación de namespaces

5.1. Paquetes Aprendices_de_Josuka.Aprendices_de_Josuka

Clases para testear.

Clases

- class `testDAO`
Clase donde se hara el testeo de la base de datos.
- class `testFachada`
Clase donde se hara el testeo de la fachada (gestor)
- class `testPartidos`
Clase donde se hara el testeo con mockito.

5.1.1. Descripción detallada

Clases para testear.

5.2. Paquetes Aprendices_de_Josuka.LD

Todos los objetos que tendrá la aplicación.

Clases

- class [Administrador](#)
Esta clase define objetos de tipo administrador.
- enum [Categoria](#)
Esta clase define objetos de tipo [Categoria](#) mediante un enumerado.
- class [Entrenador](#)
Esta clase define objetos de tipo [Entrenador](#).
- class [Equipo](#)
Esta clase define objetos de tipo equipo.
- class [Equipos_Ext](#)
Esta clase define objetos de tipo equipo que no están guardados en la base de datos y se obtienen de la raspberry.
- class [Jugador](#)
Esta clase define los objetos de tipo jugador.
- class [Material](#)
Esta clase define los objetos de tipo material, es decir, del inventario.
- class [Partido](#)
Esta clase define objetos de tipo partido que no están guardados en la base de datos y se obtienen de la raspberry.
- class [Persona](#)
Esta clase define objetos de tipo persona (clase padre), de la que después heredarán las clases entrenador y jugador (las clases hijas)
- class [Sancion](#)
Esta clase define objetos de tipo sancion que no están guardados en la base de datos y se obtienen de la raspberry.
- enum [Tipo_Material](#)
Esta clase define un enumerado para el atributo tipo de la clase [Material](#).
- enum [Tipo_Sancion](#)
Esta clase define un enumerado para el atributo tipo de la clase [Sancion](#).

5.2.1. Descripción detallada

Todos los objetos que tendrá la aplicación.

5.3. Paquetes Aprendices_de_Josuka.LP

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

Clases

- class [Anyadir_Inventario](#)
Esta clase define la GUI de la ventana de anyadir inventario del administrador.
- class [Clasificacion](#)
En esta clase se desarrolla la GUI donde se muestra la clasificacion al administrador.
- class [Clasificacion_Entrenador](#)
En esta clase se desarrolla la GUI donde se muestra la clasificacion al entrenador.
- class [Clasificacion_Jugador](#)
En esta clase se desarrolla la GUI donde se muestra la clasificacion al jugador.
- class [EditarDatos](#)

- En esta clase se desarrolla la GUI para que el jugador pueda modificar su correo o su contraseña.*
- class [EditarDatosEntrenador](#)
 - En esta clase se desarrolla la GUI para que el entrenador pueda modificar su correo o su contraseña.*
- class [EditarEntrenador](#)
 - En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del entrenador.*
- class [EditarEquipo](#)
 - En esta clase se desarrolla la GUI para que el administrador pueda parte de la informacion del equipo.*
- class [EditarJugador](#)
 - En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del jugador.*
- class [Estadisticas](#)
 - En esta clase se desarrolla la GUI donde se muestran las estadisticas del jugador.*
- class [EstadisticasEntrenador](#)
 - En esta clase se desarrolla la GUI donde se muestran las estadisticas del entrenador.*
- class [FichaEntrenador](#)
 - En esta clase se desarrolla la GUI donde se muestra al entrenador su ficha.*
- class [FichaJugador](#)
 - En esta clase se desarrolla la GUI donde se muestra al jugador su ficha.*
- class [IniciarSesion](#)
 - En esta clase se desarrolla la GUI para el inicio de sesion de los jugadores y entrenadores ya registrados.*
- class [IniciarSesion_Admin](#)
 - En esta clase se desarrolla la GUI para el inicio de sesion del administrador.*
- class [Panel_Entrenador](#)
 - En esta clase se crea un panel con la informacion del entrenador, para luego poder introducirlo en otras ventanas.*
- class [Panel_Equipos](#)
 - En esta clase se crea un panel para mostrar por pantalla cada equipo, para luego poder introducirlo en otras ventanas.*
- class [Panel_Jugador](#)
 - En esta clase se crea un panel para mostrar por pantalla cada jugador, para luego poder introducirlo en otras ventanas.*
- class [Panel_Partidos](#)
 - En esta clase se crea un panel para mostrar por pantalla cada partido, para luego poder introducirlo en otras ventanas.*
- class [Panel_Sanciones](#)
- class [Pantalla_Principal](#)
 - En esta clase se desarrolla la GUI para la pantalla principal de la aplicacion.*
- class [Principal_Administrador](#)
 - En esta clase se desarrolla la GUI que visualiza el administrador cuando abre la aplicacion despues de haber iniciado sesion.*
- class [Principal_Entrenador](#)
 - En esta clase se desarrolla la GUI que visualiza el entrenador cuando abre la aplicacion despues de haber iniciado sesion.*
- class [Principal_Jugador](#)
 - En esta clase se desarrolla la GUI que visualiza el jugador cuando abre la aplicacion despues de haber iniciado sesion.*
- class [RegistrarAdmin](#)
 - En esta clase se desarrolla la GUI para el registro del administrador.*
- class [RegistrarEntrenador](#)
 - En esta clase se desarrolla la GUI para el registro del entrenador.*
- class [RegistrarEquipo](#)
 - En esta clase se desarrolla la GUI para cuando un administrador quiera registrar un equipo.*
- class [RegistrarJugador](#)
 - En esta clase se desarrolla la GUI para el registro del jugador.*

- class [RegistrarMaterial](#)

En esta clase se desarrolla la GUI para que el administrador pueda registrar el material.

- class [Visualizar_Entrenadores](#)

En esta clase se desarrolla la GUI para que el administrador pueda visualizar los entrenadores.

- class [Visualizar_Equipos](#)

En esta clase se desarrolla la GUI para que el administrador pueda visualizar los equipos.

- class [Visualizar_Jugadores](#)

En esta clase se desarrolla la GUI para que los administradores puedan ver los jugadores.

- class [Visualizar_Mis_Jugadores](#)

En esta clase se desarrolla la GUI que se encarga de mostrar a cada entrenador sus jugadores.

5.3.1. Descripción detallada

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

5.4. Referencia del Namespace AprendicesDeJosukaPartidos

Clases

- class [AprendicesDeJosukaPartidos](#)

5.5. Referencia del Namespace AprendicesDeJosukaSanciones

Clases

- class [AprendicesDeJosukaSanciones](#)

5.6. Paquetes Controller

Necesario para incorporar el patron de diseño del controller.

Clases

- class [Controller](#)

5.6.1. Descripción detallada

Necesario para incorporar el patron de diseño del controller.

5.7. Paquetes DAO

En este paquete estan incluidas todas las clases que tienen relacion con la base de datos.

Clases

- class [DAO](#)

5.7.1. Descripción detallada

En este paquete estan incluidas todas las clases que tienen relacion con la base de datos.

Autor

Alumno

5.8. Paquetes Fachada

Necesario para incorporar el patrón de diseño de fachada. El servidor de la aplicación.

Clases

- interface [itfFachada](#)
interfaz que contiene todos los métodos que implementará el servidor
- class [ServidorPrincipal](#)
En esta clase se ejecutarán o delegarán todas las funciones de la aplicación.

5.8.1. Descripción detallada

Necesario para incorporar el patrón de diseño de fachada. El servidor de la aplicación.

5.9. Paquetes Gateway

La aplicación se conectará al servicio externo de la liga, de la cual obtendrá los partidos y las sanciones.

Clases

- class [Gateway](#)
- class [itfGateway](#)
Interfaz del [Gateway](#) donde se establecen todos los métodos que tendrá el [Gateway](#).

5.9.1. Descripción detallada

La aplicación se conectará al servicio externo de la liga, de la cual obtendrá los partidos y las sanciones.

5.10. Referencia del Namespace python

Namespaces

- [Partido](#)
- [Partidos](#)

5.11. Referencia del Namespace python.Partido

Namespaces

- [Equipo](#)
- [Partido](#)
- [Sancion](#)

5.12. Referencia del Namespace python.Partido.Equipo

Clases

- class [Equipo](#)

5.13. Referencia del Namespace python.Partido.Partido

Clases

- class [Partido](#)

5.14. Referencia del Namespace python.Partido.Sancion

Clases

- class [Sancion](#)

5.15. Referencia del Namespace python.Partidos

Namespaces

- [Interface_Partidos](#)
- [Interface_Sanciones](#)

5.16. Referencia del Namespace python.Partidos.Interface_Partidos

Clases

- class [Interface_Partidos](#)

5.17. Referencia del Namespace python.Partidos.Interface_Sanciones

Clases

- class [Interface_Sanciones](#)

5.18. Paquetes Remote

Necesario para incorporar el patrón de diseño de service locator.

Clases

- class [ServiceLocator](#)

Esta clase establece la conexión entre la fachada y controller.

5.18.1. Descripción detallada

Necesario para incorporar el patrón de diseño de service locator.

5.19. Paquetes ServiciosExternos

Clases para hacer posible la conexión con la raspberry y el parseo de la información.

Clases

- class [DF_Printable](#)

Interfaz con método print que será implementado por [Partidos_parameters](#) y [Sanciones_parameters](#) para visualizar el contenido de las listas.

- class [Partido_JSON](#)

Clase para parsear los partidos de la raspberry.

- class [Partidos_parameters](#)

Clase que contiene los parametros de los partidos y sus getters y setters.

- class [RestClient](#)

Sirve para conectar dos aplicaciones. En este caso la raspberry y la aplicación de Aprendices de Josuka.

- class [Sancion_JSON](#)

Clase para parsear las sanciones de la raspberry.

- class [Sanciones_parameters](#)

Clase que contiene los atributos de las sanciones, con sus getters y setters mediante el plugin de Lombok. Implementa [DF_Printable](#) para imprimir el contenido de cada sancion.

5.19.1. Descripción detallada

Clases para hacer posible la conexión con la raspberry y el parseo de la información.

Capítulo 6

Documentación de las clases

6.1. Referencia de la Clase Aprendices_de_Josuka.LD.Administrador

Esta clase define objetos de tipo administrador.

Métodos públicos

- [Administrador](#) (String email, String psw)

6.1.1. Descripción detallada

Esta clase define objetos de tipo administrador.

Autor

Alumno

Definición en la línea 21 del archivo Administrador.java.

6.1.2. Documentación del constructor y destructor

6.1.2.1. Administrador()

```
Aprendices_de_Josuka.LD.Administrador.Administrador (
    String email,
    String psw )
```

Definición en la línea 27 del archivo Administrador.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

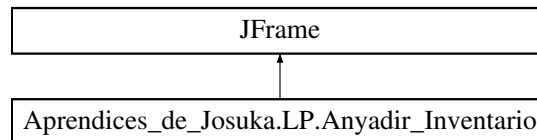
- src/main/java/Aprendices_de_Josuka/LD/[Administrador.java](#)

6.2. Referencia de la Clase

Aprendices_de_Josuka.LP.Anyadir_Inventario

Esta clase define la GUI de la ventana de anyadir inventario del administrador.

Diagrama de herencias de Aprendices_de_Josuka.LP.Anyadir_Inventario



Métodos públicos

- `Anyadir_Inventario` (`Controller` controller, `RegistrarEquipo` r, `HashMap`< `Material`, `Integer` > inventario)
- void `initComponents` ()
- void `RellenarCombo` ()
- void `Anyadir` () throws `RemoteException`

6.2.1. Descripción detallada

Esta clase define la GUI de la ventana de anyadir inventario del administrador.

Autor

Alumno

Definición en la línea 37 del archivo `Anyadir_Inventario.java`.

6.2.2. Documentación del constructor y destructor

6.2.2.1. Anyadir_Inventario()

```

Aprendices_de_Josuka.LP.Anyadir_Inventario.Anyadir_Inventario (
    Controller controller,
    RegistrarEquipo r,
    HashMap< Material, Integer > inventario )
  
```

Constructor

Parámetros

<i>controller</i>	
<i>r</i>	
<i>inventario</i>	

Definición en la línea 67 del archivo Anyadir_Inventario.java.

6.2.3. Documentación de las funciones miembro

6.2.3.1. Anyadir()

```
void Aprendices_de_Josuka.LP.Anyadir_Inventario.Anyadir ( ) throws RemoteException
```

Se asigna el material seleccionado al equipo

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 352 del archivo Anyadir_Inventario.java.

6.2.3.2. initComponents()

```
void Aprendices_de_Josuka.LP.Anyadir_Inventario.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 78 del archivo Anyadir_Inventario.java.

6.2.3.3. RellenarCombo()

```
void Aprendices_de_Josuka.LP.Anyadir_Inventario.RellenarCombo ( )
```

Se introduce la lista de materiales disponibles en el comboBox

Definición en la línea 333 del archivo Anyadir_Inventario.java.

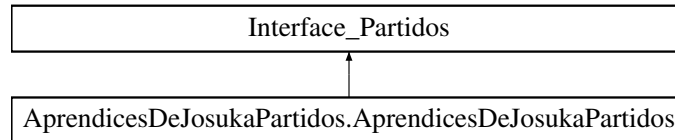
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/Anyadir_Inventario.java](#)

6.3. Referencia de la Clase

AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos

Diagrama de herencias de AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos



Métodos públicos

- def `__init__` (self)
- [Partido] `search_partidos` (self, **kwargs)
- def `with_time` (self, partido, kwargs)
- def `toJSON` (self, partido_array)
- def `generate_partido` (self)

6.3.1. Descripción detallada

Definición en la línea 15 del archivo AprendicesDeJosukaPartidos.py.

6.3.2. Documentación del constructor y destructor

6.3.2.1. `__init__()`

```
def AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos.__init__ (
    self )
```

Definición en la línea 17 del archivo AprendicesDeJosukaPartidos.py.

6.3.3. Documentación de las funciones miembro

6.3.3.1. `generate_partido()`

```
def AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos.generate_partido (
    self )
```

Definición en la línea 68 del archivo AprendicesDeJosukaPartidos.py.

6.3.3.2. search_partidos()

```
[Partido] AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos.search_partidos (
    self,
    ** kwargs )
```

Definición en la línea 29 del archivo AprendicesDeJosukaPartidos.py.

6.3.3.3. toJSON()

```
def AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos.toJSON (
    self,
    partido_array )
```

Definición en la línea 61 del archivo AprendicesDeJosukaPartidos.py.

6.3.3.4. with_time()

```
def AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos.with_time (
    self,
    partido,
    kwargs )
```

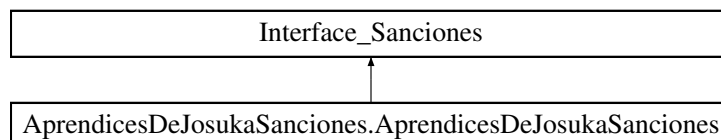
Definición en la línea 52 del archivo AprendicesDeJosukaPartidos.py.

La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/python/Partidos/ADJPartidos/AprendicesDeJosukaPartidos.py](#)

6.4. Referencia de la Clase AprendicesDeJosukaSanciones.AprendicesDeJosukaSanciones

Diagrama de herencias de AprendicesDeJosukaSanciones.AprendicesDeJosukaSanciones



Métodos públicos

- `def __init__ (self)`
- `[Sancion] search_sanciones (self, **kwargs)`
- `def toJSON (self, sancion_array)`
- `def generate_sancion (self)`

6.4.1. Descripción detallada

Definición en la línea 11 del archivo `AprendicesDeJosukaSanciones.py`.

6.4.2. Documentación del constructor y destructor

6.4.2.1. `__init__()`

```
def AprendicesDeJosukaSanciones.AprendicesDeJosukaSanciones.__init__ (
    self )
```

Definición en la línea 13 del archivo `AprendicesDeJosukaSanciones.py`.

6.4.3. Documentación de las funciones miembro

6.4.3.1. `generate_sancion()`

```
def AprendicesDeJosukaSanciones.AprendicesDeJosukaSanciones.generate_sancion (
    self )
```

Definición en la línea 46 del archivo `AprendicesDeJosukaSanciones.py`.

6.4.3.2. `search_sanciones()`

```
[Sancion] AprendicesDeJosukaSanciones.AprendicesDeJosukaSanciones.search_sanciones (
    self,
    ** kwargs )
```

Definición en la línea 18 del archivo `AprendicesDeJosukaSanciones.py`.

6.4.3.3. `toJSON()`

```
def AprendicesDeJosukaSanciones.AprendicesDeJosukaSanciones.toJSON (
    self,
    sancion_array )
```

Definición en la línea 40 del archivo `AprendicesDeJosukaSanciones.py`.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/python/Partidos/ADJPartidos/AprendicesDeJosukaSanciones.py`

6.5. Referencia del enum Aprendices_de_Josuka.LD.Categoria

Esta clase define objetos de tipo [Categoria](#) mediante un enumerado.

Atributos públicos

- [ALEVIN](#)
- [INFANTIL](#)
- [CADETE](#)
- [JUVENIL](#)
- [SENIOR](#)

6.5.1. Descripción detallada

Esta clase define objetos de tipo [Categoria](#) mediante un enumerado.

Autor

Alumno

Definición en la línea 8 del archivo Categoria.java.

6.5.2. Documentación de los datos miembro

6.5.2.1. ALEVIN

```
Aprendices_de_Josuka.LD.Categoria.ALEVIN
```

Definición en la línea 10 del archivo Categoria.java.

6.5.2.2. CADETE

```
Aprendices_de_Josuka.LD.Categoria.CADETE
```

Definición en la línea 10 del archivo Categoria.java.

6.5.2.3. INFANTIL

```
Aprendices_de_Josuka.LD.Categoria.INFANTIL
```

Definición en la línea 10 del archivo Categoria.java.

6.5.2.4. JUVENIL

`Aprendices_de_Josuka.LD.Categoria.JUVENIL`

Definición en la línea 10 del archivo `Categoria.java`.

6.5.2.5. SENIOR

`Aprendices_de_Josuka.LD.Categoria.SENIOR`

Definición en la línea 11 del archivo `Categoria.java`.

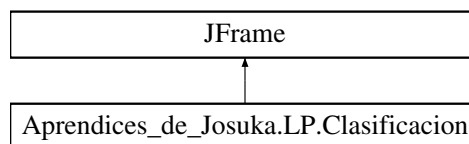
La documentación para este enum ha sido generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LD/Categoria.java`

6.6. Referencia de la Clase `Aprendices_de_Josuka.LP.Clasificacion`

En esta clase se desarrolla la GUI donde se muestra la clasificacion al administrador.

Diagrama de herencias de `Aprendices_de_Josuka.LP.Clasificacion`



Métodos públicos

- `Clasificacion` (`Controller` controller) throws `ParseException`
- void `initComponents` () throws `ParseException`
- void `InsertarJPanel` () throws `ParseException`

6.6.1. Descripción detallada

En esta clase se desarrolla la GUI donde se muestra la clasificacion al administrador.

Autor

Alumno

Definición en la línea 32 del archivo `Clasificacion.java`.

6.6.2. Documentación del constructor y destructor

6.6.2.1. `Clasificacion()`

`Aprendices_de_Josuka.LP.Clasificacion.Clasificacion (`
 `Controller controller) throws ParseException`

Create the frame.

Parámetros

<i>controller</i>	
-------------------	--

Excepciones

<i>ParseException</i>	
-----------------------	--

Definición en la línea 67 del archivo Clasificacion.java.

6.6.3. Documentación de las funciones miembro

6.6.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.Clasificacion.initComponents ( ) throws ParseException
```

Se inicializan los componentes

Excepciones

<i>ParseException</i>	
-----------------------	--

Definición en la línea 77 del archivo Clasificacion.java.

6.6.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Clasificacion.InsertarJPanel ( ) throws ParseException
```

Se insertan las sanciones en un JPanel

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	

Definición en la línea 353 del archivo Clasificacion.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

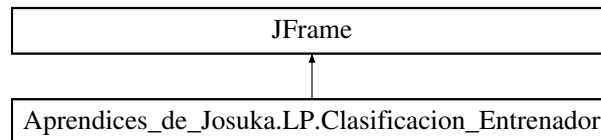
- [src/main/java/Aprendices_de_Josuka/LP/Clasificacion.java](#)

6.7. Referencia de la Clase

Aprendices_de_Josuka.LP.Clasificacion_Entrenador

En esta clase se desarrolla la GUI donde se muestra la clasificacion al entrenador.

Diagrama de herencias de Aprendices_de_Josuka.LP.Clasificacion_Entrenador



Métodos públicos

- `Clasificacion_Entrenador (Controller controller, Entrenador j)` throws `RemoteException`, `ParseException`
- void `initComponents ()` throws `ParseException`
- void `InsertarJPanel ()` throws `ParseException`

6.7.1. Descripción detallada

En esta clase se desarrolla la GUI donde se muestra la clasificacion al entrenador.

Autor

Alumno

Definición en la línea 35 del archivo `Clasificacion_Entrenador.java`.

6.7.2. Documentación del constructor y destructor

6.7.2.1. Clasificacion_Entrenador()

```

Aprendices_de_Josuka.LP.Clasificacion_Entrenador.Clasificacion_Entrenador (
    Controller controller,
    Entrenador j ) throws RemoteException, ParseException
  
```

Constructor

Parámetros

<i>controller</i>	
<i>j</i>	entrenador dentro de la aplicación

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 63 del archivo Clasificacion_Entrenador.java.

6.7.3. Documentación de las funciones miembro

6.7.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.Clasificacion_Entrenador.initComponents ( ) throws ParseException
```

Se inicializan los componentes

Excepciones

<i>ParseException</i>	
-----------------------	--

Definición en la línea 74 del archivo Clasificacion_Entrenador.java.

6.7.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Clasificacion_Entrenador.InsertarJPanel ( ) throws ParseException
```

Se introduce un panel con los equipos externos ordenados según el ranking

Excepciones

<i>ParseException</i>	
-----------------------	--

Definición en la línea 307 del archivo Clasificacion_Entrenador.java.

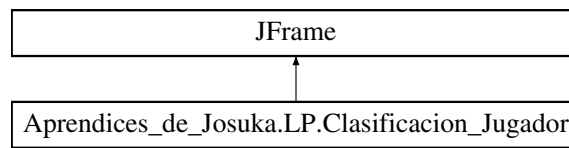
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Aprendices_de_Josuka/LP/[Clasificacion_Entrenador.java](#)

6.8. Referencia de la Clase Aprendices_de_Josuka.LP.Clasificacion_Jugador

En esta clase se desarrolla la GUI donde se muestra la clasificacion al jugador.

Diagrama de herencias de Aprendices_de_Josuka.LP.Clasificacion_Jugador



Métodos públicos

- `Clasificacion_Jugador` (`Controller` controller, `Jugador` j) throws `RemoteException`, `ParseException`
- void `initComponents` () throws `ParseException`
- void `InsertarJPanel` () throws `ParseException`

6.8.1. Descripción detallada

En esta clase se desarrolla la GUI donde se muestra la clasificacion al jugador.

Autor

Alumno

Definición en la línea 34 del archivo `Clasificacion_Jugador.java`.

6.8.2. Documentación del constructor y destructor

6.8.2.1. Clasificacion_Jugador()

```

Aprendices_de_Josuka.LP.Clasificacion_Jugador.Clasificacion_Jugador (
    Controller controller,
    Jugador j ) throws RemoteException, ParseException
  
```

Constructor

Parámetros

<i>controller</i>	
<i>j</i>	jugador que esta dentro de la aplicacion

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 65 del archivo Clasificacion_Jugador.java.

6.8.3. Documentación de las funciones miembro

6.8.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.Clasificacion_Jugador.initComponents ( ) throws ParseException
```

Se inicializan los componentes

Excepciones

<i>ParseException</i>	
-----------------------	--

Definición en la línea 77 del archivo Clasificacion_Jugador.java.

6.8.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Clasificacion_Jugador.InsertarJPanel ( ) throws ParseException
```

Inserta los equipos ordenados por ranking segun clasificacion

Excepciones

<i>ParseException</i>	
-----------------------	--

Definición en la línea 275 del archivo Clasificacion_Jugador.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/Clasificacion_Jugador.java](#)

6.9. Referencia de la Clase Controller.Controller

Métodos públicos

- [Controller](#) () throws RemoteException
- List< [Partido](#) > [getPartidos](#) () throws RemoteException, ParseException
- boolean [EntrarJugador](#) (String email, String psw) throws RemoteException
- List< [Equipos_Ext](#) > [Clasificacion](#) ([Categoria](#) cat) throws RemoteException, ParseException
- boolean [EntrarEntrenador](#) (String email, String psw) throws RemoteException

- boolean [EntrarAdministrador](#) (String email, String psw) throws RemoteException
- boolean [RegistrarJugador](#) (String nombre, String apellido, String fecha_S, String DNI, int telefono, String correo, String psw) throws RemoteException
- void [RegistrarEntrenador](#) (String nombre, String apellido, String fecha_S, String DNI, int telefono, String correo, String psw, boolean isAsignado) throws RemoteException
- void [RegistrarAdmin](#) (String correo, String psw) throws RemoteException
- boolean [RegistrarEquipo2](#) (String nombre, [Categoria](#) cat, [Entrenador](#) entrenador, List< [Jugador](#) > lista_Jugadores, HashMap< [Material](#), Integer > inventario) throws RemoteException
- void [RegistrarInventario](#) ([Tipo_Material](#) mat, int cantidad, long precio) throws RemoteException
- int [partidosGanados](#) (String DNI) throws RemoteException, ParseException
- List< [Sancion](#) > [sancionesPersona](#) (String DNI) throws RemoteException, ParseException
- List< [Jugador](#) > [misJugadores](#) (String ent) throws RemoteException
- boolean [modificarCorreo](#) ([Jugador](#) j, String correo, String psw) throws RemoteException
- boolean [modificarCorreoEntrenador](#) ([Entrenador](#) j, String correo, String psw) throws RemoteException
- [Jugador](#) [getJugador](#) (String correo, String psw) throws RemoteException
- [Entrenador](#) [getEntrenador](#) (String correo, String psw) throws RemoteException
- List< [Jugador](#) > [MostrarJugadores](#) ([Categoria](#) c) throws RemoteException
- int [getRanking](#) (String dni) throws RemoteException, ParseException
- List< [Jugador](#) > [equipoJugadores](#) ([Equipo](#) e) throws RemoteException
- [Entrenador](#) [equipoEntrenador](#) ([Equipo](#) e) throws RemoteException
- [Partido](#) [sancionPartido](#) (String cod) throws RemoteException, ParseException
- List< [Partido](#) > [jugadorPartido](#) (List< [Partido](#) > partidos, String DNI) throws RemoteException, ParseException
- List< [Entrenador](#) > [getEntrenador](#) () throws RemoteException
- List< [Material](#) > [getMaterial](#) () throws RemoteException
- List< [Equipo](#) > [getEquipos](#) () throws RemoteException
- String [ObtenerEquipoJugador](#) ([Jugador](#) jugador) throws RemoteException
- List< [Equipo](#) > [getEquiposFilter](#) ([Categoria](#) cat) throws RemoteException
- String [ObtenerEquipoEntrenador](#) (String entrenador) throws RemoteException
- void [ActualizarJugadorEquipo](#) (List< [Jugador](#) > lista_Jugadores) throws RemoteException
- void [ActualizarJugador](#) ([Jugador](#) e, boolean reconocimiento, boolean lesionado, boolean cuota) throws RemoteException
- void [ActualizarEntrenador](#) ([Entrenador](#) e, long salario) throws RemoteException
- void [ActualizarEquipo](#) ([Equipo](#) equipo, HashMap< [Material](#), Integer > inventario, List< [Jugador](#) > jugadores) throws RemoteException
- void [ActualizarEquipoEntrenador](#) ([Entrenador](#) e) throws RemoteException
- void [AsignarInventario](#) ([Material](#) m) throws RemoteException

6.9.1. Descripción detallada

Definición en la línea 26 del archivo Controller.java.

6.9.2. Documentación del constructor y destructor

6.9.2.1. Controller()

```
Controller.Controller.Controller ( ) throws RemoteException
```

Constructor

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 32 del archivo Controller.java.

6.9.3. Documentación de las funciones miembro

6.9.3.1. ActualizarEntrenador()

```
void Controller.Controller.ActualizarEntrenador (
    Entrenador e,
    long salario ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de modificar la informacion del entrenador que ha editado el administrador

Parámetros

<i>e</i>	
<i>salario</i>	

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 376 del archivo Controller.java.

6.9.3.2. ActualizarEquipo()

```
void Controller.Controller.ActualizarEquipo (
    Equipo equipo,
    HashMap< Material, Integer > inventario,
    List< Jugador > jugadores ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de modificar la informacion del equipo que ha editado el administrador

Parámetros

<i>equipo</i>	
<i>inventario</i>	
<i>jugadores</i>	

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 387 del archivo Controller.java.

6.9.3.3. ActualizarEquipoEntrenador()

```
void Controller.Controller.ActualizarEquipoEntrenador (
    Entrenador e ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de modificar el entrenador de un equipo

Parámetros

<i>e</i>	
----------	--

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 396 del archivo Controller.java.

6.9.3.4. ActualizarJugador()

```
void Controller.Controller.ActualizarJugador (
    Jugador e,
    boolean reconocimiento,
    boolean lesionado,
    boolean cuota ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de modificar la informacion del jugador que ha editado el administrador

Parámetros

<i>e</i>	
<i>reconocimiento</i>	
<i>lesionado</i>	
<i>cuota</i>	

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 366 del archivo Controller.java.

6.9.3.5. ActualizarJugadorEquipo()

```
void Controller.Controller.ActualizarJugadorEquipo (
    List< Jugador > lista_Jugadores ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de actualizar la lista de jugadores de un equipo

Parámetros

<i>lista_Jugadores</i>	
------------------------	--

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 355 del archivo Controller.java.

6.9.3.6. AsignarInventario()

```
void Controller.Controller.AsignarInventario (
    Material m ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de modificar el inventario asignado a un equipo

Parámetros

<i>m</i>	
----------	--

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 405 del archivo Controller.java.

6.9.3.7. Clasificacion()

```
List<Equipos_Ext> Controller.Controller.Clasificacion (
    Categoria cat ) throws RemoteException, ParseException
```

Metodo que hace la llamada al metodo de la fachada que obtiene la lista de los equipos obtenidos de la raspberry

Parámetros

<i>cat</i>	
------------	--

Devuelve

la lista de los equipos obtenidos de la raspberry

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 65 del archivo Controller.java.

6.9.3.8. EntrarAdministrador()

```
boolean Controller.Controller.EntrarAdministrador (
    String email,
    String psw ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de inicio de sesion del entrenador administrador

Parámetros

<i>email</i>	
<i>psw</i>	

Devuelve

un boolean que informa sobre si el administrador ha entrado bien en la aplicacion (true) o no (false)

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 85 del archivo Controller.java.

6.9.3.9. EntrarEntrenador()

```
boolean Controller.Controller.EntrarEntrenador (
    String email,
    String psw ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de inicio de sesion del entrenador

Parámetros

<i>email</i>	
<i>psw</i>	

Devuelve

un boolean que informa sobre si el entrenador ha entrado bien en la aplicación (true) o no (false)

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 75 del archivo Controller.java.

6.9.3.10. EntrarJugador()

```
boolean Controller.Controller.EntrarJugador (
    String email,
    String psw ) throws RemoteException
```

Método que hace la llamada al método de la fachada de inicio de sesión del jugador

Parámetros

<i>email</i>	
<i>psw</i>	

Devuelve

un boolean que informa sobre si el jugador ha entrado bien en la aplicación (true) o no (false)

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 55 del archivo Controller.java.

6.9.3.11. equipoEntrenador()

```
Entrenador Controller.Controller.equipoEntrenador (
    Equipo e ) throws RemoteException
```

Método que hace la llamada al método de la fachada de obtener el entrenador de un equipo

Parámetros

<i>e</i>	
----------	--

Devuelve

el entrenador de un equipo

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 273 del archivo Controller.java.

6.9.3.12. equipoJugadores()

```
List<Jugador> Controller.Controller.equipoJugadores (
    Equipo e ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de obtener la lista de jugadores de un equipo

Parámetros

<i>e</i>	
----------	--

Devuelve

la lista de jugadores de un equipo

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 264 del archivo Controller.java.

6.9.3.13. getEntrenador() [1/2]

```
List<Entrenador> Controller.Controller.getEntrenador ( ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de obtener una lista de entrenadores

Devuelve

una lista de entrenadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 302 del archivo Controller.java.

6.9.3.14. getEntrenador() [2/2]

```
Entrenador Controller.Controller.getEntrenador (
    String correo,
    String psw ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de la fachada de leer un entrenador

Parámetros

<i>correo</i>	
<i>psw</i>	

Devuelve

el entrenador

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 234 del archivo Controller.java.

6.9.3.15. getEquipos()

```
List<Equipo> Controller.Controller.getEquipos ( ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de obtener equipos

Devuelve

lista de equipos

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 318 del archivo Controller.java.

6.9.3.16. getEquiposFilter()

```
List<Equipo> Controller.Controller.getEquiposFilter (
    Categoria cat ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de filtrar los equipos por categoria

Parámetros

<i>cat</i>	
------------	--

Devuelve

la lista de equipos filtrada

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 336 del archivo Controller.java.

6.9.3.17. getJugador()

```
Jugador Controller.Controller.getJugador (
    String correo,
    String psw ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de leer un jugador

Parámetros

<i>correo</i>	
<i>psw</i>	

Devuelve

el jugador

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 223 del archivo Controller.java.

6.9.3.18. getMaterial()

```
List<Material> Controller.Controller.getMaterial ( ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de obtener inventario

Devuelve

una lista de inventario

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 310 del archivo Controller.java.

6.9.3.19. getPartidos()

```
List<Partido> Controller.Controller.getPartidos ( ) throws RemoteException, ParseException
```

Metodo que devuelve la lista de partidos de la fachada

Devuelve

lista de partidos

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 44 del archivo Controller.java.

6.9.3.20. getRanking()

```
int Controller.Controller.getRanking (
    String dni ) throws RemoteException, ParseException
```

Metodo que hace la llamada al metodo de la fachada de obtener el puesto del ranking

Parámetros

<i>dni</i>	
------------	--

Devuelve

un int con el puesto del ranking

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 254 del archivo Controller.java.

6.9.3.21. jugadorPartido()

```
List<Partido> Controller.Controller.jugadorPartido (  
    List< Partido > partidos,  
    String DNI ) throws RemoteException, ParseException
```

Metodo que hace la llamada al metodo de la fachada de obtener una lista de partidos jugados por un jugador

Parámetros

<i>partidos</i>	
<i>DNI</i>	

Devuelve

una lista de partidos

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 294 del archivo Controller.java.

6.9.3.22. misJugadores()

```
List<Jugador> Controller.Controller.misJugadores (  
    String ent ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de obtener la lista de los jugadores de un entrenador

Parámetros

<i>ent</i>	
------------	--

Devuelve

lista de los jugadores de un entrenador

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 188 del archivo Controller.java.

6.9.3.23. modificarCorreo()

```
boolean Controller.Controller.modificarCorreo (
    Jugador j,
    String correo,
    String psw ) throws RemoteException
```

Método que hace la llamada al método de la fachada de modificar el correo y la contraseña de un jugador

Parámetros

<i>j</i>	
<i>correo</i>	
<i>psw</i>	

Devuelve

un boolean que informa si se ha modificado correctamente (true) o no (false)

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 200 del archivo Controller.java.

6.9.3.24. modificarCorreoEntrenador()

```
boolean Controller.Controller.modificarCorreoEntrenador (
    Entrenador j,
```

```
String correo,
String psw ) throws RemoteException
```

Método que hace la llamada al método de la fachada de modificar el correo y la contraseña de un jugador

Parámetros

<i>j</i>	
<i>correo</i>	
<i>psw</i>	

Devuelve

un boolean que informa si se ha modificado correctamente (true) o no (false)

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 212 del archivo Controller.java.

6.9.3.25. MostrarJugadores()

```
List<Jugador> Controller.Controller.MostrarJugadores (
    Categoria c ) throws RemoteException
```

Método que hace la llamada al método de la fachada de leer jugadores de una categoría

Parámetros

<i>c</i>	
----------	--

Devuelve

la lista de jugadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 244 del archivo Controller.java.

6.9.3.26. ObtenerEquipoEntrenador()

```
String Controller.Controller.ObtenerEquipoEntrenador (
    String entrenador ) throws RemoteException
```

Método que hace la llamada al método de la fachada de obtener el nombre del equipo de un entrenador

Parámetros

<i>entrenador</i>	
-------------------	--

Devuelve

un string con el nombre del equipo de un entrenador

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 346 del archivo Controller.java.

6.9.3.27. ObtenerEquipoJugador()

```
String Controller.Controller.ObtenerEquipoJugador (
    Jugador jugador ) throws RemoteException
```

Método que hace la llamada al método de la fachada de obtener el nombre del equipo de un jugador

Parámetros

<i>jugador</i>	
----------------	--

Devuelve

un String con el nombre de un equipo

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 327 del archivo Controller.java.

6.9.3.28. partidosGanados()

```
int Controller.Controller.partidosGanados (
    String DNI ) throws RemoteException, ParseException
```

Método que hace la llamada al método de la fachada de obtener los partidos ganados

Parámetros

<i>DNI</i>	
------------	--

Devuelve**Excepciones**

<i>RemoteException</i>	
<i>ParseException</i>	Metodo que hace la llamada al metodo de la fachada de obtener los partidos ganados

Parámetros

<i>DNI</i>	
------------	--

Devuelve**Excepciones**

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 169 del archivo Controller.java.

6.9.3.29. RegistrarAdmin()

```
void Controller.Controller.RegistrarAdmin (
    String correo,
    String psw ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de registrarse del administrador

Parámetros

<i>correo</i>	
<i>psw</i>	

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 128 del archivo Controller.java.

6.9.3.30. RegistrarEntrenador()

```
void Controller.Controller.RegistrarEntrenador (
    String nombre,
    String apellido,
    String fecha_S,
    String DNI,
    int telefono,
    String correo,
    String psw,
    boolean isAsignado ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de registrarse del entrenador

Parámetros

<i>nombre</i>	
<i>apellido</i>	
<i>fecha_S</i>	
<i>DNI</i>	
<i>telefono</i>	
<i>correo</i>	
<i>psw</i>	
<i>isAsignado</i>	

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 118 del archivo Controller.java.

6.9.3.31. RegistrarEquipo2()

```
boolean Controller.Controller.RegistrarEquipo2 (
    String nombre,
    Categoria cat,
    Entrenador entrenador,
    List< Jugador > lista_Jugadores,
    HashMap< Material, Integer > inventario ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de registro de equipo

Parámetros

<i>nombre</i>	
---------------	--

Parámetros

<i>cat</i>	
<i>entrenador</i>	
<i>lista_Jugadores</i>	
<i>inventario</i>	

Devuelve

boolean que informa sobre si el equipo se ha guardado bien (true) o no (false)

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 141 del archivo Controller.java.

6.9.3.32. RegistrarInventario()

```
void Controller.Controller.RegistrarInventario (
    Tipo_Material mat,
    int cantidad,
    long precio ) throws RemoteException
```

Metodo que hace la llamada al metodo de la fachada de registro de inventario

Parámetros

<i>mat</i>	
<i>cantidad</i>	
<i>precio</i>	

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 152 del archivo Controller.java.

6.9.3.33. RegistrarJugador()

```
boolean Controller.Controller.RegistrarJugador (
    String nombre,
    String apellido,
```

```
String fecha_S,
String DNI,
int telefono,
String correo,
String psw ) throws RemoteException
```

Método que hace la llamada al método de la fachada de registrarse del jugador

Parámetros

<i>nombre</i>	
<i>apellido</i>	
<i>fecha_S</i>	
<i>DNI</i>	
<i>telefono</i>	
<i>correo</i>	
<i>psw</i>	

Devuelve

un boolean que informa sobre si el jugador se ha registrado bien en la aplicación (true) o no (false)

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 100 del archivo Controller.java.

6.9.3.34. sancionesPersona()

```
List<Sancion> Controller.Controller.sancionesPersona (
String DNI ) throws RemoteException, ParseException
```

Método que hace la llamada al método de la fachada de obtener la lista de las sanciones

Parámetros

<i>DNI</i>	
------------	--

Devuelve

lista de sanciones

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 179 del archivo Controller.java.

6.9.3.35. sancionPartido()

```
Partido Controller.Controller.sancionPartido (
    String cod ) throws RemoteException, ParseException
```

Metodo que hace la llamada al metodo de la fachada de obtener en que partido se ha realizado una sancion

Parámetros

<i>cod</i>	
------------	--

Devuelve

un partido en el que se haya realizado una sancion

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 283 del archivo Controller.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Controller/[Controller.java](#)

6.10. Referencia de la Clase DAO.DAO

Métodos públicos

- boolean [guardarObjeto](#) (Object objeto)
- List< [Material](#) > [getMaterial](#) ()
- boolean [ModificarMaterial](#) ([Tipo_Material](#) tipo, int cantidad, long precio)
- List< [Administrador](#) > [getAdmin](#) ()
- List< [Jugador](#) > [getJugador](#) ()
- List< [Entrenador](#) > [getEntrenador](#) ()
- List< [Equipo](#) > [getEquipo](#) ()
- boolean [ActualizarEntrenador](#) (Long salario, [Entrenador](#) entrenador)
- boolean [ActualizarJugador](#) ([Jugador](#) jugador, boolean reconocimiento, boolean lesionado, boolean cuota)
- boolean [modificarCorreo](#) ([Jugador](#) jugador, String correo, String psw)
- boolean [modificarCorreoEntrenador](#) ([Entrenador](#) e, String correo, String psw)
- void [ActualizarEquipoJugador](#) (List< [Jugador](#) > jugadores)
- void [ActualizarEquipoEntrenador](#) ([Entrenador](#) ent)
- void [ActualizarEquipo](#) ([Equipo](#) equipo, HashMap inventario, List< String > jugadores)

- void [ActualizarMaterial](#) (List< [Material](#) > lista_material)
- boolean [EliminarAdmin](#) ([Administrador](#) a)
- boolean [EliminarJugador](#) ([Jugador](#) j)
- boolean [EliminarEntrenador](#) ([Entrenador](#) e)
- boolean [EliminarEquipo](#) ([Equipo](#) e)
- boolean [EliminarMaterial](#) ([Material](#) e)
- void [cerrarConexion](#) ()

Métodos públicos estáticos

- static [DAO getInstance](#) ()

6.10.1. Descripción detallada

Definición en la línea 30 del archivo DAO.java.

6.10.2. Documentación de las funciones miembro

6.10.2.1. ActualizarEntrenador()

```
boolean DAO.DAO.ActualizarEntrenador (
    Long salario,
    Entrenador entrenador )
```

Método para modificar los entrenadores (de la parte de administrador)

Parámetros

<i>salario</i>	
<i>entrenador</i>	

Devuelve

un boolean que informa sobre si el entrenador se ha modificado correctamente (true) o no (false)

Definición en la línea 219 del archivo DAO.java.

6.10.2.2. ActualizarEquipo()

```
void DAO.DAO.ActualizarEquipo (
    Equipo equipo,
    HashMap inventario,
    List< String > jugadores )
```

Método para modificar un equipo

Parámetros

<i>equipo</i>	
<i>inventario</i>	
<i>jugadores</i>	

Definición en la línea 372 del archivo DAO.java.

6.10.2.3. ActualizarEquipoEntrenador()

```
void DAO.DAO.ActualizarEquipoEntrenador (
    Entrenador ent )
```

Metodo para modificar el boolean asignado a un entrenador

Parámetros

<i>ent</i>	
------------	--

Definición en la línea 348 del archivo DAO.java.

6.10.2.4. ActualizarEquipoJugador()

```
void DAO.DAO.ActualizarEquipoJugador (
    List< Jugador > jugadores )
```

Metodo para modificar los jugadores de un equipo

Parámetros

<i>jugadores</i>	
------------------	--

Definición en la línea 325 del archivo DAO.java.

6.10.2.5. ActualizarJugador()

```
boolean DAO.DAO.ActualizarJugador (
    Jugador jugador,
    boolean reconocimiento,
    boolean lesionado,
    boolean cuota )
```

Metodo para modificar los jugadores (de la parte de administrador)

Parámetros

<i>jugador</i>	
<i>reconocimiento</i>	
<i>lesionado</i>	
<i>cuota</i>	

Devuelve

un boolean que informa sobre si el jugador se ha modificado correctamente (true) o no (false)

Definición en la línea 246 del archivo DAO.java.

6.10.2.6. ActualizarMaterial()

```
void DAO.DAO.ActualizarMaterial (
    List< Material > lista_material )
```

Metodo para modificar el inventario

Parámetros

<i>lista_material</i>	
-----------------------	--

Definición en la línea 394 del archivo DAO.java.

6.10.2.7. cerrarConexion()

```
void DAO.DAO.cerrarConexion ( )
```

Metodo para cerrar la conexion con la base de datos

Definición en la línea 592 del archivo DAO.java.

6.10.2.8. EliminarAdmin()

```
boolean DAO.DAO.EliminarAdmin (
    Administrador a )
```

Metodo para eliminar un administrador registrado

Parámetros

<i>a</i>	
----------	--

Devuelve

un boolean que informa sobre si el administrador se ha eliminado correctamente (true) o no (false)

Definición en la línea 418 del archivo DAO.java.

6.10.2.9. EliminarEntrenador()

```
boolean DAO.DAO.EliminarEntrenador (
    Entrenador e )
```

Metodo para eliminar un entrenador registrado

Parámetros

<i>e</i>	
----------	--

Devuelve

un boolean que informa sobre si el entrenador se ha eliminado correctamente (true) o no (false)

Definición en la línea 489 del archivo DAO.java.

6.10.2.10. EliminarEquipo()

```
boolean DAO.DAO.EliminarEquipo (
    Equipo e )
```

Metodo para eliminar un equipo

Parámetros

<i>e</i>	
----------	--

Devuelve

un boolean que informa sobre si el equipo se ha eliminado correctamente (true) o no (false)

Definición en la línea 524 del archivo DAO.java.

6.10.2.11. EliminarJugador()

```
boolean DAO.DAO.EliminarJugador (
    Jugador j )
```

Método para eliminar un jugador registrado

Parámetros

<i>j</i>	
----------	--

Devuelve

un boolean que informa sobre si el jugador se ha eliminado correctamente (true) o no (false)

Definición en la línea 453 del archivo DAO.java.

6.10.2.12. EliminarMaterial()

```
boolean DAO.DAO.EliminarMaterial (
    Material e )
```

Método para eliminar el material registrado

Parámetros

<i>e</i>	
----------	--

Devuelve

un boolean que informa sobre si el material se ha eliminado correctamente (true) o no (false)

Definición en la línea 559 del archivo DAO.java.

6.10.2.13. getAdmin()

```
List<Administrador> DAO.DAO.getAdmin ( )
```

Método para leer los administradores registrados

Devuelve

la lista de administradores

Definición en la línea 153 del archivo DAO.java.

6.10.2.14. getEntrenador()

```
List<Entrenador> DAO.DAO.getEntrenador ( )
```

Metodo para leer los entrenadores registrados

Devuelve

la lista de entrenadores

Definición en la línea 186 del archivo DAO.java.

6.10.2.15. getEquipo()

```
List<Equipo> DAO.DAO.getEquipo ( )
```

Metodo para leer los equipos registrados

Devuelve

la lista de equipos

Definición en la línea 202 del archivo DAO.java.

6.10.2.16. getInstance()

```
static DAO DAO.DAO.getInstance ( ) [static]
```

Patrón singleton

Devuelve

un new del DAO

Definición en la línea 48 del archivo DAO.java.

6.10.2.17. getJugador()

```
List<Jugador> DAO.DAO.getJugador ( )
```

Metodo para leer los jugadores registrados

Devuelve

la lista de jugadores

Definición en la línea 169 del archivo DAO.java.

6.10.2.18. getMaterial()

```
List<Material> DAO.DAO.getMaterial ( )
```

Metodo para obtener la lista completa del inventario registrada

Devuelve

el inventario registrado en la base de datos

Definición en la línea 114 del archivo DAO.java.

6.10.2.19. guardarObjeto()

```
boolean DAO.DAO.guardarObjeto (
    Object objeto )
```

Se guardan los objetos en la base de datos en funcion de su tipo

Parámetros

<i>objeto</i>	
---------------	--

Devuelve

un boolean que informa sobre si el objeto se ha guardado correctamente (true) o no (false)

Definición en la línea 57 del archivo DAO.java.

6.10.2.20. modificarCorreo()

```
boolean DAO.DAO.modificarCorreo (
    Jugador jugador,
    String correo,
    String psw )
```

Metodo para modificar los jugadores (parte jugador)

Parámetros

<i>jugador</i>	
<i>correo</i>	
<i>psw</i>	

Devuelve

un boolean que informa sobre si el jugador se ha modificado correctamente (true) o no (false)

Definición en la línea 274 del archivo DAO.java.

6.10.2.21. modificarCorreoEntrenador()

```
boolean DAO.DAO.modificarCorreoEntrenador (
    Entrenador e,
    String correo,
    String psw )
```

Metodo para modificar los entrenadores (parte entrenador)

Parámetros

<i>e</i>	
<i>correo</i>	
<i>psw</i>	

Devuelve

un boolean que informa sobre si el entrenador se ha modificado correctamente (true) o no (false)

Definición en la línea 301 del archivo DAO.java.

6.10.2.22. ModificarMaterial()

```
boolean DAO.DAO.ModificarMaterial (
    Tipo_Material tipo,
    int cantidad,
    long precio )
```

Metodo para modificar el inventario

Parámetros

<i>tipo</i>	
<i>cantidad</i>	
<i>precio</i>	

Devuelve

un boolean que informa sobre si el material se ha modificado correctamente (true) o no (false)

Definición en la línea 131 del archivo DAO.java.

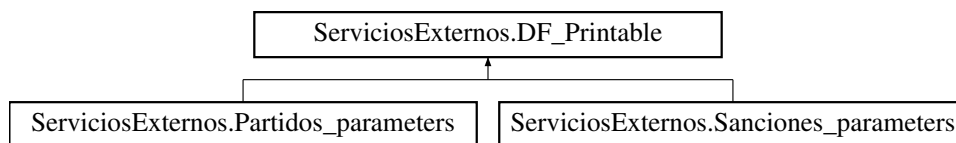
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/DAO/DAO.java`

6.11. Referencia de la Clase ServiciosExternos.DF_Printable

Interfaz con método print que será implementado por [Partidos_parameters](#) y [Sanciones_parameters](#) para visualizar el contenido de las listas.

Diagrama de herencias de ServiciosExternos.DF_Printable



Métodos públicos

- void [print](#) ()

6.11.1. Descripción detallada

Interfaz con método print que será implementado por [Partidos_parameters](#) y [Sanciones_parameters](#) para visualizar el contenido de las listas.

Autor

Alumno

Definición en la línea 12 del archivo DF_Printable.java.

6.11.2. Documentación de las funciones miembro

6.11.2.1. print()

```
void ServiciosExternos.DF_Printable.print ( )
```

Reimplementado en [ServiciosExternos.Sanciones_parameters](#) y [ServiciosExternos.Partidos_parameters](#).

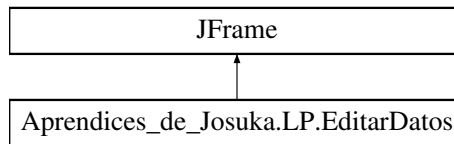
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/ServiciosExternos/DF_Printable.java`

6.12. Referencia de la Clase Aprendices_de_Josuka.LP.EditarDatos

En esta clase se desarrolla la GUI para que el jugador pueda modificar su correo o su contraseña.

Diagrama de herencias de Aprendices_de_Josuka.LP.EditarDatos



Métodos públicos

- `EditarDatos` (`Controller` controller, `Jugador` j) throws `RemoteException`, `ParseException`
- void `initComponents` ()
- void `modificar` () throws `RemoteException`, `ParseException`

6.12.1. Descripción detallada

En esta clase se desarrolla la GUI para que el jugador pueda modificar su correo o su contraseña.

Autor

Alumno

Definición en la línea 34 del archivo EditarDatos.java.

6.12.2. Documentación del constructor y destructor

6.12.2.1. EditarDatos()

```

Aprendices_de_Josuka.LP.EditarDatos.EditarDatos (
    Controller controller,
    Jugador j ) throws RemoteException, ParseException
  
```

Constructor

Parámetros

<i>controller</i>	
<i>j</i>	jugador que esta dentro de la aplicacion

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 62 del archivo EditarDatos.java.

6.12.3. Documentación de las funciones miembro

6.12.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.EditarDatos.initComponents ( )
```

Inicia los componentes

Definición en la línea 72 del archivo EditarDatos.java.

6.12.3.2. modificar()

```
void Aprendices_de_Josuka.LP.EditarDatos.modificar ( ) throws RemoteException, ParseException
```

Método para guardar los cambios

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 271 del archivo EditarDatos.java.

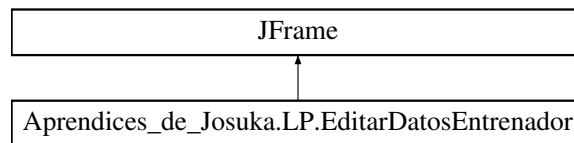
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Aprendices_de_Josuka/LP/[EditarDatos.java](#)

6.13. Referencia de la Clase Aprendices_de_Josuka.LP.EditarDatosEntrenador

En esta clase se desarrolla la GUI para que el entrenador pueda modificar su correo o su contraseña.

Diagrama de herencias de Aprendices_de_Josuka.LP.EditarDatosEntrenador



Métodos públicos

- `EditarDatosEntrenador (Controller controller, Entrenador e)` throws `RemoteException`, `ParseException`
- void `initComponents ()`
- void `modificar ()` throws `RemoteException`, `ParseException`

6.13.1. Descripción detallada

En esta clase se desarrolla la GUI para que el entrenador pueda modificar su correo o su contraseña.

Autor

Alumno

Definición en la línea 34 del archivo `EditardatosEntrenador.java`.

6.13.2. Documentación del constructor y destructor

6.13.2.1. EditarDatosEntrenador()

```

Aprendices_de_Josuka.LP.EditarDatosEntrenador.EditarDatosEntrenador (
    Controller controller,
    Entrenador e ) throws RemoteException, ParseException
  
```

Constructor

Parámetros

<i>controller</i>	
<i>e</i>	entrenador que esta dentro de la aplicacion

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 62 del archivo `EditardatosEntrenador.java`.

6.13.3. Documentación de las funciones miembro

6.13.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.EditarDatosEntrenador.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 72 del archivo EditarDatosEntrenador.java.

6.13.3.2. modificar()

```
void Aprendices_de_Josuka.LP.EditarDatosEntrenador.modificar ( ) throws RemoteException, ParseException
```

Se guardan los datos a modificar

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 275 del archivo EditarDatosEntrenador.java.

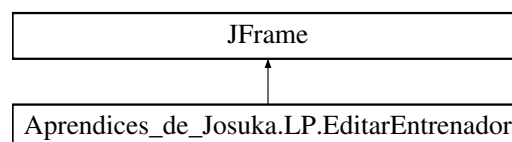
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Aprendices_de_Josuka/LP/[EditarDatosEntrenador.java](#)

6.14. Referencia de la Clase Aprendices_de_Josuka.LP.EditarEntrenador

En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del entrenador.

Diagrama de herencias de Aprendices_de_Josuka.LP.EditarEntrenador



Métodos públicos

- [EditarEntrenador](#) (Controller c)
- void [initComponents](#) ()
- void [IntroducirEntrenadores](#) ()
- void [VisualizarEntrenadores](#) ()
- void [Guardar](#) ()

6.14.1. Descripción detallada

En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del entrenador.

Autor

Alumno

Definición en la línea 32 del archivo EditarEntrenador.java.

6.14.2. Documentación del constructor y destructor

6.14.2.1. EditarEntrenador()

```
Aprendices_de_Josuka.LP.EditarEntrenador.EditarEntrenador (
    Controller c )
```

Constructor

Parámetros

c	
---	--

Definición en la línea 67 del archivo EditarEntrenador.java.

6.14.3. Documentación de las funciones miembro

6.14.3.1. Guardar()

```
void Aprendices_de_Josuka.LP.EditarEntrenador.Guardar ( )
```

Se guardan los cambios efectuados

Definición en la línea 417 del archivo EditarEntrenador.java.

6.14.3.2. initComponents()

```
void Aprendices_de_Josuka.LP.EditarEntrenador.initComponents ( )
```

Inicializar componentes

Definición en la línea 76 del archivo EditarEntrenador.java.

6.14.3.3. IntroducirEntrenadores()

```
void Aprendices_de_Josuka.LP.EditarEntrenador.IntroducirEntrenadores ( )
```

Se introduce la lista de entrenadores en el ComboBox

Definición en la línea 379 del archivo EditarEntrenador.java.

6.14.3.4. VisualizarEntrenadores()

```
void Aprendices_de_Josuka.LP.EditarEntrenador.VisualizarEntrenadores ( )
```

Se visualizan las características del entrenador seleccionado

Definición en la línea 396 del archivo EditarEntrenador.java.

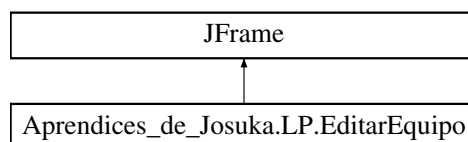
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/EditarEntrenador.java`

6.15. Referencia de la Clase Aprendices_de_Josuka.LP.EditarEquipo

En esta clase se desarrolla la GUI para que el administrador pueda parte de la información del equipo.

Diagrama de herencias de Aprendices_de_Josuka.LP.EditarEquipo

**Métodos públicos**

- [EditarEquipo](#) ([Controller](#) c)
- void [initComponents](#) ()
- void [VisualizarEquipos](#) ()
- void [MostrarInformacion](#) () throws RemoteException
- void [AnyadirJugadores](#) ()
- void [EliminarJugadores](#) ()
- void [Guardar](#) () throws RemoteException
- List< Integer > [ActualizarMaterial](#) () throws RemoteException

6.15.1. Descripción detallada

En esta clase se desarrolla la GUI para que el administrador pueda parte de la informacion del equipo.

Autor

Alumno

Definición en la línea 40 del archivo EditarEquipo.java.

6.15.2. Documentación del constructor y destructor

6.15.2.1. EditarEquipo()

```
Aprendices_de_Josuka.LP.EditarEquipo.EditarEquipo (
    Controller c )
```

Constructor

Parámetros

<i>c</i>	
----------	--

Definición en la línea 97 del archivo EditarEquipo.java.

6.15.3. Documentación de las funciones miembro

6.15.3.1. ActualizarMaterial()

```
List<Integer> Aprendices_de_Josuka.LP.EditarEquipo.ActualizarMaterial ( ) throws RemoteException
```

Se actualiza el material del equipo con los cambios efectuados

Devuelve

la lista del material actualizada

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 765 del archivo EditarEquipo.java.

6.15.3.2. AnyadirJugadores()

```
void Aprendices_de_Josuka.LP.EditarEquipo.AnyadirJugadores ( )
```

Se anyaden los jugadores seleccionados

Definición en la línea 658 del archivo EditarEquipo.java.

6.15.3.3. EliminarJugadores()

```
void Aprendices_de_Josuka.LP.EditarEquipo.EliminarJugadores ( )
```

Se eliminan los jugadores seleccionados

Definición en la línea 674 del archivo EditarEquipo.java.

6.15.3.4. Guardar()

```
void Aprendices_de_Josuka.LP.EditarEquipo.Guardar ( ) throws RemoteException
```

Se guardan los cambios efectuados

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 691 del archivo EditarEquipo.java.

6.15.3.5. initComponents()

```
void Aprendices_de_Josuka.LP.EditarEquipo.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 105 del archivo EditarEquipo.java.

6.15.3.6. MostrarInformacion()

```
void Aprendices_de_Josuka.LP.EditarEquipo.MostrarInformacion ( ) throws RemoteException
```

Se muestra la informacion del equipo

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 568 del archivo EditarEquipo.java.

6.15.3.7. VisualizarEquipos()

```
void Aprendices_de_Josuka.LP.EditarEquipo.VisualizarEquipos ( )
```

Se visualizan los equipos disponibles

Definición en la línea 549 del archivo EditarEquipo.java.

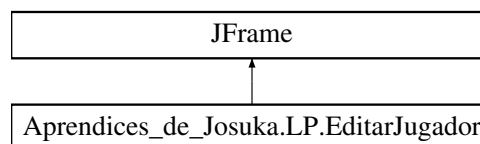
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Aprendices_de_Josuka/LP/[EditarEquipo.java](#)

6.16. Referencia de la Clase Aprendices_de_Josuka.LP.EditarJugador

En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del jugador.

Diagrama de herencias de Aprendices_de_Josuka.LP.EditarJugador

**Métodos públicos**

- [EditarJugador](#) ([Controller](#) controller)
- void [initComponents](#) ()
- void [VisualizarJugadores](#) ()
- void [MostrarInformacion](#) ()
- void [Guardar](#) ()

Métodos públicos estáticos

- static void [main](#) (String[] args)

6.16.1. Descripción detallada

En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del jugador.

Autor

Alumno

Definición en la línea 33 del archivo EditarJugador.java.

6.16.2. Documentación del constructor y destructor

6.16.2.1. EditarJugador()

```
Aprendices_de_Josuka.LP.EditarJugador.EditarJugador (
    Controller controller )
```

Constructor

Parámetros

controller	
------------	--

Definición en la línea 92 del archivo EditarJugador.java.

6.16.3. Documentación de las funciones miembro

6.16.3.1. Guardar()

```
void Aprendices_de_Josuka.LP.EditarJugador.Guardar ( )
```

Se guardan los cambios efectuados del jugador

Definición en la línea 506 del archivo EditarJugador.java.

6.16.3.2. initComponents()

```
void Aprendices_de_Josuka.LP.EditarJugador.initComponents ( )
```

Definición en la línea 98 del archivo EditarJugador.java.

6.16.3.3. main()

```
static void Aprendices_de_Josuka.LP.EditarJugador.main (
    String[] args ) [static]
```

Launch the application.

Definición en la línea 74 del archivo EditarJugador.java.

6.16.3.4. MostrarInformacion()

```
void Aprendices_de_Josuka.LP.EditarJugador.MostrarInformacion ( )
```

Se muestra la informacion del jugador seleccionado

Definición en la línea 480 del archivo EditarJugador.java.

6.16.3.5. VisualizarJugadores()

```
void Aprendices_de_Josuka.LP.EditarJugador.VisualizarJugadores ( )
```

Se visualizan los jugadores disponibles por categoria

Definición en la línea 463 del archivo EditarJugador.java.

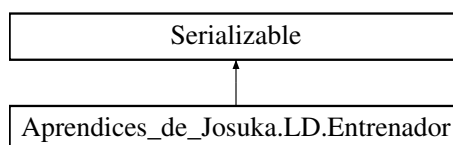
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/EditarJugador.java](#)

6.17. Referencia de la Clase Aprendices_de_Josuka.LD.Entrenador

Esta clase define objetos de tipo [Entrenador](#).

Diagrama de herencias de Aprendices_de_Josuka.LD.Entrenador



Métodos públicos

- [Entrenador](#) (String nombre, String apellido, String fecha_nacimiento, String DNI, int telefono, String correo, String psw, long salario, boolean asignado_equipo)
- String [toString](#) ()

6.17.1. Descripción detallada

Esta clase define objetos de tipo [Entrenador](#).

Autor

Alumno

Definición en la línea 28 del archivo Entrenador.java.

6.17.2. Documentación del constructor y destructor

6.17.2.1. Entrenador()

```
Aprendices_de_Josuka.LD.Entrenador.Entrenador (
    String nombre,
    String apellido,
    String fecha_nacimiento,
    String DNI,
    int telefono,
    String correo,
    String psw,
    long salario,
    boolean asignado_equipo )
```

Definición en la línea 41 del archivo Entrenador.java.

6.17.3. Documentación de las funciones miembro

6.17.3.1. toString()

```
String Aprendices_de_Josuka.LD.Entrenador.toString ( )
```

Definición en la línea 56 del archivo Entrenador.java.

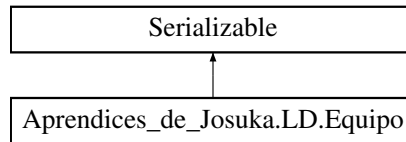
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LD/Entrenador.java`

6.18. Referencia de la Clase Aprendices_de_Josuka.LD.Equipo

Esta clase define objetos de tipo equipo.

Diagrama de herencias de Aprendices_de_Josuka.LD.Equipo



Métodos públicos

- **Equipo** (String nombre, **Categoria** categoria, String entrenador, List< String >lista_jugador, HashMap< String, Integer > inventario)
- String **toString** ()

6.18.1. Descripción detallada

Esta clase define objetos de tipo equipo.

Autor

Alumno

Definición en la línea 26 del archivo Equipo.java.

6.18.2. Documentación del constructor y destructor

6.18.2.1. Equipo()

```
Aprendices_de_Josuka.LD.Equipo.Equipo (
    String nombre,
    Categoria categoria,
    String entrenador,
    List< String > lista_jugador,
    HashMap< String, Integer > inventario )
```

Definición en la línea 37 del archivo Equipo.java.

6.18.3. Documentación de las funciones miembro

6.18.3.1. toString()

```
String Aprendices_de_Josuka.LD.Equipo.toString ( )
```

Definición en la línea 47 del archivo Equipo.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LD/Equipo.java`

6.19. Referencia de la Clase python.Partido.Equipo.Equipo

Métodos públicos

- `def __init__ (self, str code, str nombre, int puntuacion, str categoria)`
- `str get_code (self)`
- `str get_nombre (self)`
- `int get_puntuacion (self)`
- `str get_categoria (self)`
- `None print (self)`

6.19.1. Descripción detallada

Definición en la línea 5 del archivo Equipo.py.

6.19.2. Documentación del constructor y destructor

6.19.2.1. __init__()

```
def python.Partido.Equipo.Equipo.__init__ (
    self,
    str code,
    str nombre,
    int puntuacion,
    str categoria )
```

Definición en la línea 9 del archivo Equipo.py.

6.19.3. Documentación de las funciones miembro

6.19.3.1. get_categoria()

```
str python.Partido.Equipo.Equipo.get_categoria (  
    self )
```

Definición en la línea 31 del archivo Equipo.py.

6.19.3.2. get_code()

```
str python.Partido.Equipo.Equipo.get_code (  
    self )
```

Definición en la línea 18 del archivo Equipo.py.

6.19.3.3. get_nombre()

```
str python.Partido.Equipo.Equipo.get_nombre (  
    self )
```

Definición en la línea 23 del archivo Equipo.py.

6.19.3.4. get_puntuacion()

```
int python.Partido.Equipo.Equipo.get_puntuacion (  
    self )
```

Definición en la línea 27 del archivo Equipo.py.

6.19.3.5. print()

```
None python.Partido.Equipo.Equipo.print (  
    self )
```

Definición en la línea 36 del archivo Equipo.py.

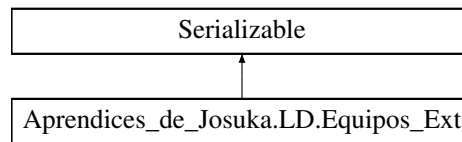
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/python/Partido/Equipo.py](#)

6.20. Referencia de la Clase Aprendices_de_Josuka.LD.Equipos_Ext

Esta clase define objetos de tipo equipo que no están guardados en la base de datos y se obtienen de la raspberry.

Diagrama de herencias de Aprendices_de_Josuka.LD.Equipos_Ext



Métodos públicos

- [Equipos_Ext](#) (String cod, String nombre, long puntuacion, String categoria)
- boolean [equals](#) (Object equipo)
- int [hashCode](#) ()

6.20.1. Descripción detallada

Esta clase define objetos de tipo equipo que no están guardados en la base de datos y se obtienen de la raspberry.

Autor

Alumno

Definición en la línea 16 del archivo Equipos_Ext.java.

6.20.2. Documentación del constructor y destructor

6.20.2.1. Equipos_Ext()

```

Aprendices_de_Josuka.LD.Equipos_Ext.Equipos_Ext (
    String cod,
    String nombre,
    long puntuacion,
    String categoria )
  
```

Definición en la línea 22 del archivo Equipos_Ext.java.

6.20.3. Documentación de las funciones miembro

6.20.3.1. equals()

```
boolean Aprendices_de_Josuka.LD.Equipos_Ext.equals (
    Object equipo )
```

Definición en la línea 30 del archivo Equipos_Ext.java.

6.20.3.2. hashCode()

```
int Aprendices_de_Josuka.LD.Equipos_Ext.hashCode ( )
```

Definición en la línea 41 del archivo Equipos_Ext.java.

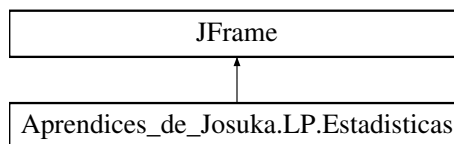
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LD/Equipos_Ext.java](#)

6.21. Referencia de la Clase Aprendices_de_Josuka.LP.Estadisticas

En esta clase se desarrolla la GUI donde se muestran las estadísticas del jugador.

Diagrama de herencias de Aprendices_de_Josuka.LP.Estadisticas



Métodos públicos

- [Estadisticas](#) ([Controller](#) controller, [Jugador](#) j) throws RemoteException, ParseException
- void [initComponents](#) () throws ParseException, RemoteException
- void [InsertarJPanel](#) () throws ParseException, RemoteException

6.21.1. Descripción detallada

En esta clase se desarrolla la GUI donde se muestran las estadísticas del jugador.

Autor

Alumno

Definición en la línea 35 del archivo Estadisticas.java.

6.21.2. Documentación del constructor y destructor

6.21.2.1. Estadisticas()

```
Aprendices_de_Josuka.LP.Estadisticas.Estadisticas (
    Controller controller,
    Jugador j ) throws RemoteException, ParseException
```

Constructor

Parámetros

<i>controller</i>	
<i>j</i>	jugador que esta dentro de la aplicación

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 68 del archivo Estadisticas.java.

6.21.3. Documentación de las funciones miembro

6.21.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.Estadisticas.initComponents ( ) throws ParseException, RemoteException
```

Se inicializan los componentes

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	

Definición en la línea 83 del archivo Estadisticas.java.

6.21.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Estadisticas.InsertarJPanel ( ) throws ParseException, RemoteException
```

Se insertan las sanciones en un JPanel

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	

Definición en la línea 286 del archivo Estadisticas.java.

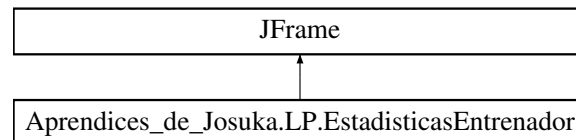
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/Estadisticas.java`

6.22. Referencia de la Clase Aprendices_de_Josuka.LP.EstadisticasEntrenador

En esta clase se desarrolla la GUI donde se muestran las estadísticas del entrenador.

Diagrama de herencias de `Aprendices_de_Josuka.LP.EstadisticasEntrenador`



Métodos públicos

- `EstadisticasEntrenador (Controller controller, Entrenador j)` throws `RemoteException`, `ParseException`
- `void initComponents ()` throws `ParseException`, `RemoteException`
- `void InsertarJPanel ()` throws `ParseException`, `RemoteException`

6.22.1. Descripción detallada

En esta clase se desarrolla la GUI donde se muestran las estadísticas del entrenador.

Autor

Alumno

Definición en la línea 36 del archivo `EstadisticasEntrenador.java`.

6.22.2. Documentación del constructor y destructor

6.22.2.1. EstadisticasEntrenador()

```

Aprendices_de_Josuka.LP.EstadisticasEntrenador.EstadisticasEntrenador (
    Controller controller,
    Entrenador j ) throws RemoteException, ParseException
  
```

Constructor

Parámetros

<i>controller</i>	
<i>j</i>	entrenador que esta dentro de la aplicacion

Generado por Doxygen

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 69 del archivo EstadisticasEntrenador.java.

6.22.3. Documentación de las funciones miembro

6.22.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.EstadisticasEntrenador.initComponents ( ) throws ParseException,  
RemoteException
```

Se inicializan los componentes

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	

Definición en la línea 84 del archivo EstadisticasEntrenador.java.

6.22.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.EstadisticasEntrenador.InsertarJPanel ( ) throws ParseException,  
RemoteException
```

Se insertan todas las sanciones en un JPanel

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	

Definición en la línea 310 del archivo EstadisticasEntrenador.java.

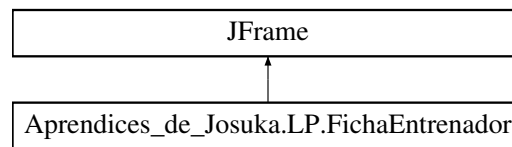
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Aprendices_de_Josuka/LP/[EstadisticasEntrenador.java](#)

6.23. Referencia de la Clase Aprendices_de_Josuka.LP.FichaEntrenador

En esta clase se desarrolla la GUI donde se muestra al entrenador su ficha.

Diagrama de herencias de Aprendices_de_Josuka.LP.FichaEntrenador



Métodos públicos

- `FichaEntrenador (Controller controller, Entrenador ent)` throws `RemoteException`, `ParseException`
- void `initComponents ()`

6.23.1. Descripción detallada

En esta clase se desarrolla la GUI donde se muestra al entrenador su ficha.

Autor

Alumno

Definición en la línea 32 del archivo FichaEntrenador.java.

6.23.2. Documentación del constructor y destructor

6.23.2.1. FichaEntrenador()

```

Aprendices_de_Josuka.LP.FichaEntrenador.FichaEntrenador (
    Controller controller,
    Entrenador ent ) throws RemoteException, ParseException
  
```

Constructor de la clase

Parámetros

<i>controller</i>	
<i>ent</i>	entrenador que se encuentra dentro de la aplicación

Excepciones

<i>RemoteException</i>	
------------------------	--

Excepciones

<i>ParseException</i>	
-----------------------	--

Definición en la línea 55 del archivo FichaEntrenador.java.

6.23.3. Documentación de las funciones miembro

6.23.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.FichaEntrenador.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 65 del archivo FichaEntrenador.java.

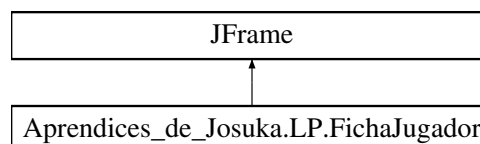
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/FichaEntrenador.java`

6.24. Referencia de la Clase Aprendices_de_Josuka.LP.FichaJugador

En esta clase se desarrolla la GUI donde se muestra al jugador su ficha.

Diagrama de herencias de Aprendices_de_Josuka.LP.FichaJugador



Métodos públicos

- `FichaJugador (Controller controller, Jugador j)` throws `RemoteException`, `ParseException`
- `void initComponents ()`

6.24.1. Descripción detallada

En esta clase se desarrolla la GUI donde se muestra al jugador su ficha.

Autor

Alumno

Definición en la línea 31 del archivo FichaJugador.java.

6.24.2. Documentación del constructor y destructor

6.24.2.1. FichaJugador()

```
Aprendices_de_Josuka.LP.FichaJugador.FichaJugador (
    Controller controller,
    Jugador j ) throws RemoteException, ParseException
```

Constructor

Parámetros

<i>controller</i>	
<i>j</i>	el jugador que esta dentro de la aplicación

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 56 del archivo FichaJugador.java.

6.24.3. Documentación de las funciones miembro

6.24.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.FichaJugador.initComponents ( )
```

Se inicializan los componentes

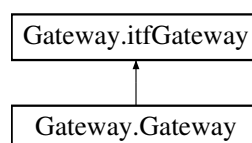
Definición en la línea 66 del archivo FichaJugador.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/FichaJugador.java`

6.25. Referencia de la Clase Gateway.Gateway

Diagrama de herencias de Gateway.Gateway



Métodos públicos

- List< Partido > [search_partidos](#) () throws ParseException
- boolean [partidos](#) () throws ParseException
- boolean [sanciones](#) () throws ParseException
- ArrayList< Partido > [convertir_partidos](#) (List< Partido_JSON > json)
- HashSet< Equipos_Ext > [getEquipos](#) () throws ParseException
- List< Sancion > [search_sanciones](#) () throws ParseException
- ArrayList< Sancion > [convertir_sanciones](#) (List< Sancion_JSON > json)

Métodos públicos estáticos

- static [Gateway getInstance](#) ()

6.25.1. Descripción detallada

Definición en la línea 30 del archivo Gateway.java.

6.25.2. Documentación de las funciones miembro

6.25.2.1. [convertir_partidos\(\)](#)

```
ArrayList<Partido> Gateway.Gateway.convertir_partidos (
    List< Partido_JSON > json )
```

Parsea el contenido de la raspberry (Partido_JSON) al objeto Partido de la aplicación

Parámetros

<i>json</i>	lista de partidos a parsear
-------------	-----------------------------

Devuelve

lista de partidos parseado

Reimplementado de [Gateway.ifGateway](#).

Definición en la línea 101 del archivo Gateway.java.

6.25.2.2. [convertir_sanciones\(\)](#)

```
ArrayList<Sancion> Gateway.Gateway.convertir_sanciones (
    List< Sancion_JSON > json )
```

Parsea el contenido de la raspberry (Sancion_JSON) al objeto Sancion de la aplicación

Parámetros

<i>json</i>	lista de sanciones a parsear
-------------	------------------------------

Devuelve

lista de sanciones parseado

Reimplementado de [Gateway.itfGateway](#).

Definición en la línea 163 del archivo Gateway.java.

6.25.2.3. getEquipos()

```
HashSet<Equipos_Ext> Gateway.Gateway.getEquipos ( ) throws ParseException
```

Los partidos tienen dos equipos, mediante este método se obtiene la lista de todos los equipos externos. Donde algunos corresponderán con los equipos del club de futbol. Así, se podrá calcular la clasificación

Devuelve

devuelve una lista de los equipos

Excepciones

<i>ParseException</i>	
-----------------------	--

Reimplementado de [Gateway.itfGateway](#).

Definición en la línea 116 del archivo Gateway.java.

6.25.2.4. getInstance()

```
static Gateway Gateway.Gateway.getInstance ( ) [static]
```

Mediante este método se incorpora el patrón Singleton, para instanciar el objeto una unica vez

Devuelve

devolverá el objeto [Gateway](#)

Definición en la línea 46 del archivo Gateway.java.

6.25.2.5. partidos()

```
boolean Gateway.Gateway.partidos ( ) throws ParseException
```

Método para ver si las listas de partidos de la raspberry está vacía o no

Devuelve

true si la lista está llena, false si no

Excepciones

<i>ParseException</i>	
-----------------------	--

Reimplementado de [Gateway.itfGateway](#).

Definición en la línea 80 del archivo Gateway.java.

6.25.2.6. sanciones()

```
boolean Gateway.Gateway.sanciones ( ) throws ParseException
```

Método para ver si las listas de sanciones de la raspberry está vacía o no

Devuelve

true si la lista está llena, false si no

Excepciones

<i>ParseException</i>	
-----------------------	--

Reimplementado de [Gateway.itfGateway](#).

Definición en la línea 91 del archivo Gateway.java.

6.25.2.7. search_partidos()

```
List<Partido> Gateway.Gateway.search_partidos ( ) throws ParseException
```

Método para obtener la lista de partidos de la raspberry

Devuelve

devuelve la lista

Excepciones

<i>ParseException</i>	
-----------------------	--

Reimplementado de [Gateway.itfGateway](#).

Definición en la línea 51 del archivo Gateway.java.

6.25.2.8. search_sanciones()

```
List<Sancion> Gateway.Gateway.search_sanciones ( ) throws ParseException
```

Método para obtener la lista de sanciones de la raspberry

Devuelve

devuelve la lista

Excepciones

<i>ParseException</i>	
-----------------------	--

Reimplementado de [Gateway.itfGateway](#).

Definición en la línea 129 del archivo Gateway.java.

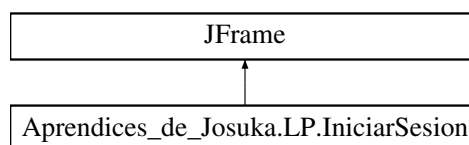
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Gateway/[Gateway.java](#)

6.26. Referencia de la Clase Aprendices_de_Josuka.LP.IniciarSesion

En esta clase se desarrolla la GUI para el inicio de sesion de los jugadores y entrenadores ya registrados.

Diagrama de herencias de Aprendices_de_Josuka.LP.IniciarSesion

**Métodos públicos**

- [IniciarSesion](#) ([Controller](#) c)
- void [initComponents](#) ()
- void [Entrar](#) (String email, String psw) throws `ParseException`

Métodos públicos estáticos

- static void `main` (String[] args)

6.26.1. Descripción detallada

En esta clase se desarrolla la GUI para el inicio de sesión de los jugadores y entrenadores ya registrados.

Autor

Alumno

Definición en la línea 28 del archivo `IniciarSesion.java`.

6.26.2. Documentación del constructor y destructor

6.26.2.1. `IniciarSesion()`

```
Aprendices_de_Josuka.LP.IniciarSesion.IniciarSesion (
    Controller c )
```

Constructor de la ventana

Parámetros

<code>c</code>	el controller para utilizar el mismo todo el rato
----------------	---------------------------------------------------

Definición en la línea 55 del archivo `IniciarSesion.java`.

6.26.3. Documentación de las funciones miembro

6.26.3.1. `Entrar()`

```
void Aprendices_de_Josuka.LP.IniciarSesion.Entrar (
    String email,
    String psw ) throws ParseException
```

Cuando pulsa "Entrar" se ejecuta este método. Verifica la existencia de esos datos en la base de datos, y en caso de existir, le da paso a la aplicación

Parámetros

<i>email</i>	
<i>psw</i>	

Excepciones

<i>ParseException</i>	
-----------------------	--

Definición en la línea 185 del archivo IniciarSesion.java.

6.26.3.2. initComponents()

```
void Aprendices_de_Josuka.LP.IniciarSesion.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 64 del archivo IniciarSesion.java.

6.26.3.3. main()

```
static void Aprendices_de_Josuka.LP.IniciarSesion.main (
    String[] args ) [static]
```

La primera ventana en lanzar, establece la conexión con el servidor.

Definición en la línea 39 del archivo IniciarSesion.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

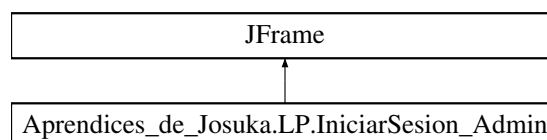
- `src/main/java/Aprendices_de_Josuka/LP/IniciarSesion.java`

6.27. Referencia de la Clase

Aprendices_de_Josuka.LP.IniciarSesion_Admin

En esta clase se desarrolla la GUI para el inicio de sesion del administrador.

Diagrama de herencias de Aprendices_de_Josuka.LP.IniciarSesion_Admin



Métodos públicos

- `IniciarSesion_Admin` (`Controller` controller)
- `void initComponents` ()

Métodos públicos estáticos

- `static void main` (`String[]` args)

6.27.1. Descripción detallada

En esta clase se desarrolla la GUI para el inicio de sesion del administrador.

Autor

Alumno

Definición en la línea 33 del archivo `IniciarSesion_Admin.java`.

6.27.2. Documentación del constructor y destructor

6.27.2.1. `IniciarSesion_Admin()`

```
Aprendices_de_Josuka.LP.IniciarSesion_Admin.IniciarSesion_Admin (
    Controller controller )
```

Constructor

Parámetros

<code>controller</code>	
-------------------------	--

Definición en la línea 57 del archivo `IniciarSesion_Admin.java`.

6.27.3. Documentación de las funciones miembro

6.27.3.1. `initComponents()`

```
void Aprendices_de_Josuka.LP.IniciarSesion_Admin.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 66 del archivo `IniciarSesion_Admin.java`.

6.27.3.2. main()

```
static void Aprendices_de_Josuka.LP.IniciarSesion_Admin.main (
    String[] args ) [static]
```

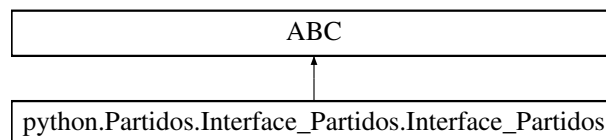
Definición en la línea 41 del archivo IniciarSesion_Admin.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/IniciarSesion_Admin.java](#)

6.28. Referencia de la Clase python.Partidos.Interface_Partidos.Interface_Partidos

Diagrama de herencias de python.Partidos.Interface_Partidos.Interface_Partidos



Métodos públicos

- [Partido] [search_partidos](#) (self, **kwargs)

6.28.1. Descripción detallada

Definición en la línea 6 del archivo Interface_Partidos.py.

6.28.2. Documentación de las funciones miembro

6.28.2.1. search_partidos()

```
[ Partido ] python.Partidos.Interface_Partidos.Interface_Partidos.search_partidos (
    self,
    ** kwargs )
```

Definición en la línea 9 del archivo Interface_Partidos.py.

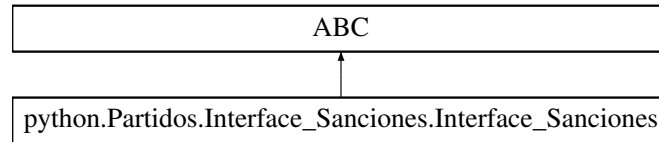
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/python/Partidos/Interface_Partidos.py](#)

6.29. Referencia de la Clase

python.Partidos.Interface_Sanciones.Interface_Sanciones

Diagrama de herencias de python.Partidos.Interface_Sanciones.Interface_Sanciones



Métodos públicos

- [Sancion] [search_sanciones](#) (self, **kwargs)

6.29.1. Descripción detallada

Definición en la línea 4 del archivo Interface_Sanciones.py.

6.29.2. Documentación de las funciones miembro

6.29.2.1. search_sanciones()

```
[ Sancion ] python.Partidos.Interface_Sanciones.Interface_Sanciones.search_sanciones (
    self,
    ** kwargs )
```

Definición en la línea 7 del archivo Interface_Sanciones.py.

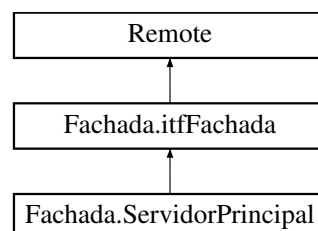
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/python/Partidos/Interface_Sanciones.py](#)

6.30. Referencia de la Interfaz Fachada.itfFachada

interfaz que contiene todos los métodos que implementará el servidor

Diagrama de herencias de Fachada.itfFachada



Métodos públicos

- boolean [EntrarJugador](#) (String email, String psw) throws RemoteException
- boolean [EntrarEntrenador](#) (String email, String psw) throws RemoteException
- boolean [EntrarAdministrador](#) (String email, String psw) throws RemoteException
- boolean [RegistrarJugador](#) (String nombre, String apellido, String fecha_S, String DNI, int telefono, String correo, String psw) throws RemoteException
- boolean [RegistrarAdmin](#) (String correo, String psw) throws RemoteException
- boolean [RegistrarEntrenador](#) (String nombre, String apellido, String fecha_S, String DNI, int telefono, String correo, String psw) throws RemoteException
- boolean [RegistrarEquipo2](#) (String nombre, [Categoria](#) cat, [Entrenador](#) entrenador, List< [Jugador](#) > lista_↵ Jugadores, HashMap< [Material](#), Integer > inventario) throws RemoteException
- boolean [RegistrarInventario](#) ([Tipo_Material](#) mat, int cantidad, long precio) throws RemoteException
- List< [Jugador](#) > [getJugador](#) () throws RemoteException
- [Jugador](#) [getJug](#) (String correo, String psw) throws RemoteException
- [Entrenador](#) [getEnt](#) (String correo, String psw) throws RemoteException
- [Partido](#) [sancionPartido](#) (String codPartido) throws ParseException, RemoteException
- int [partidosGanados](#) (String DNI) throws RemoteException, ParseException
- boolean [modificarCorreo](#) ([Jugador](#) j, String correo, String psw) throws RemoteException
- List< [Sancion](#) > [sancionesPersona](#) (String DNI) throws RemoteException, ParseException
- List< [Jugador](#) > [misJugadores](#) (String ent) throws RemoteException
- List< [Partido](#) > [getPartidos](#) () throws RemoteException, ParseException
- List< [Jugador](#) > [MostrarJugadores](#) ([Categoria](#) c) throws RemoteException
- List< [Entrenador](#) > [getEntrenador](#) () throws RemoteException
- List< [Equipo](#) > [getEquipos](#) () throws RemoteException
- int [getRanking](#) (String dni) throws ParseException, RemoteException
- boolean [modificarCorreoEntrenador](#) ([Entrenador](#) j, String correo, String psw) throws RemoteException
- List< [Partido](#) > [jugadorPartido](#) (List< [Partido](#) > partidos, String DNI) throws RemoteException, Parse↵ Exception
- void [ActualizarEquipoEntrenador](#) ([Entrenador](#) ent) throws RemoteException
- List< [Equipo](#) > [getEquiposFilter](#) ([Categoria](#) cat) throws RemoteException
- List< [Material](#) > [getMaterial](#) () throws RemoteException
- String [ObtenerEquipoEntrenador](#) (String entrenador) throws RemoteException
- int [getEdad](#) (String date) throws RemoteException
- String [ObtenerEquipoJugador](#) ([Jugador](#) jugador) throws RemoteException
- boolean [ActualizarJugador](#) ([Jugador](#) e, boolean reconocimiento, boolean lesionado, boolean cuota) throws RemoteException
- boolean [ActualizarEntrenador](#) ([Entrenador](#) e, long salario) throws RemoteException
- void [ActualizarJugadorEquipo](#) (List< [Jugador](#) > lista_Jugadores) throws RemoteException
- void [ActualizarEquipo](#) ([Equipo](#) equipo, HashMap< [Material](#), Integer > inventario, List< [Jugador](#) > jugadores) throws RemoteException
- void [ActualizarMaterial](#) ([Tipo_Material](#) tipo, int cantidad, long precio) throws RemoteException
- void [AsignarInventario](#) ([Material](#) m) throws RemoteException
- [Entrenador](#) [equipoEntrenador](#) ([Equipo](#) e) throws RemoteException
- List< [Jugador](#) > [equipoJugadores](#) ([Equipo](#) e) throws RemoteException
- List< [Equipos_Ext](#) > [clasificacion](#) ([Categoria](#) cat) throws RemoteException, ParseException

6.30.1. Descripción detallada

interfaz que contiene todos los métodos que implementará el servidor

Autor

Alumno

Definición en la línea 25 del archivo itfFachada.java.

6.30.2. Documentación de las funciones miembro

6.30.2.1. ActualizarEntrenador()

```
boolean Fachada.itfFachada.ActualizarEntrenador (
    Entrenador e,
    long salario ) throws RemoteException
```

Mediante este método se actualizará el salario de un entrenador

Parámetros

<i>e</i>	entrenador al que se le actualizarán los datos
<i>salario</i>	nuevo salario

Devuelve

devuelve true si se ha modificado correctamente

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.2. ActualizarEquipo()

```
void Fachada.itfFachada.ActualizarEquipo (
    Equipo equipo,
    HashMap< Material, Integer > inventario,
    List< Jugador > jugadores ) throws RemoteException
```

Método mediante la cual se actualizan los datos de un equipo

Parámetros

<i>equipo</i>	al cual se le modificarán los datos
<i>inventario</i>	nuevo inventario
<i>jugadores</i>	nuevos jugadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.3. ActualizarEquipoEntrenador()

```
void Fachada.itfFachada.ActualizarEquipoEntrenador (
    Entrenador ent ) throws RemoteException
```

Mediante este método se actualizan los entrenadores, estableciendo así su asignación

Parámetros

<i>ent</i>	
------------	--

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.4. ActualizarJugador()

```
boolean Fachada.itfFachada.ActualizarJugador (
    Jugador e,
    boolean reconocimiento,
    boolean lesionado,
    boolean cuota ) throws RemoteException
```

Método para actualizar el estado del jugador, tanto médico como de su cuota

Parámetros

<i>e</i>	jugador al que se le actualizarán los datos
<i>reconocimiento</i>	si ha superado el reconocimiento médico o no
<i>lesionado</i>	si esta lesionado o no
<i>cuota</i>	si ha pagado la cuota o no

Devuelve

devuelve true si se ha modificado correctamente

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.5. ActualizarJugadorEquipo()

```
void Fachada.itfFachada.ActualizarJugadorEquipo (
    List< Jugador > lista_Jugadores ) throws RemoteException
```

Cuando un equipo se crea y se añaden jugadores a este, se ejecuta este metodo para actualizar el estado de los jugadores a ASIGNADO

Parámetros

<i>lista_Jugadores</i>	jugadores a los que se les actualizará el estado
------------------------	--------------------------------------------------

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.6. ActualizarMaterial()

```
void Fachada.itfFachada.ActualizarMaterial (
    Tipo_Material tipo,
    int cantidad,
    long precio ) throws RemoteException
```

Este método sirve para modificar el material

Parámetros

<i>tipo</i>	de material
<i>cantidad</i>	
<i>precio</i>	

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.7. AsignarInventario()

```
void Fachada.itfFachada.AsignarInventario (
    Material m ) throws RemoteException
```

Cuando se asigna un material a un equipo, hay que reducir el inventario del club. Es lo que se hace en este método mediante la conexión a la base de datos

Parámetros

<i>m</i>	material a modificar
----------	----------------------

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.8. clasificacion()

```
List<Equipos_Ext> Fachada.itfFachada.clasificacion (
    Categoria cat ) throws RemoteException, ParseException
```

Devuelve una lista ordenada de los equipos de la liga según la categoría para hacer clasificación

Parámetros

<i>cat</i>	categoría
------------	-----------

Devuelve

lista ordenada

Excepciones

<i>RemoteException</i>	
<i>org.json.simple.parser.ParseException</i>	

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.9. EntrarAdministrador()

```
boolean Fachada.itfFachada.EntrarAdministrador (
    String email,
    String psw ) throws RemoteException
```

Verifica si el administrador existe en la base de datos para permitir el acceso a la aplicación

Parámetros

<i>email</i>	
<i>psw</i>	

Devuelve

true si existe

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.10. EntrarEntrenador()

```
boolean Fachada.itfFachada.EntrarEntrenador (
    String email,
    String psw ) throws RemoteException
```

Verifica si el entrenador existe en la base de datos para permitir el acceso a la aplicación

Parámetros

<i>email</i>	
<i>psw</i>	

Devuelve

true si existe

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.11. EntrarJugador()

```
boolean Fachada.itfFachada.EntrarJugador (
    String email,
    String psw ) throws RemoteException
```

Verifica si el jugador existe en la base de datos para permitir el acceso a la aplicación

Parámetros

<i>email</i>	
<i>psw</i>	

Devuelve

true si existe

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.12. equipoEntrenador()

```
Entrenador Fachada.itfFachada.equipoEntrenador (
    Equipo e ) throws RemoteException
```

Mediante este método se obtiene el entrenador asignado a un equipo concreto

Parámetros

<i>e</i>	el equipo en cuestión
----------	-----------------------

Devuelve

devuelve el entrenador

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.13. equipoJugadores()

```
List<Jugador> Fachada.itfFachada.equipoJugadores (
    Equipo e ) throws RemoteException
```

Este método sirve para obtener los DNIs de los jugadores asignados a un equipo

Parámetros

e	el equipo
---	-----------

Devuelve

devuelve la lista de jugadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.14. getEdad()

```
int Fachada.itfFachada.getEdad (
    String date ) throws RemoteException
```

Calcula la edad en base a una fecha, para así clasificar los jugadores en categorías

Parámetros

<i>date</i>	fecha de nacimiento
-------------	---------------------

Devuelve

devuelve años

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.15. getEnt()

```
Entrenador Fachada.itfFachada.getEnt (
    String correo,
    String psw ) throws RemoteException
```

Devuelve todos los datos de un entrenador segun correo y contraseña

Parámetros

<i>correo</i>	
<i>psw</i>	

Devuelve

entrenador

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.16. getEntrenador()

```
List<Entrenador> Fachada.itfFachada.getEntrenador ( ) throws RemoteException
```

Método para obtener todos los entrenadores almacenados en la base de datos

Devuelve

devuelve lista de entrenadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.17. getEquipos()

```
List<Equipo> Fachada.itfFachada.getEquipos ( ) throws RemoteException
```

Mediante este método se obtienen todos los equipos de la base de datos que pertenecen al club

Devuelve

lista de equipos

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.18. getEquiposFilter()

```
List<Equipo> Fachada.itfFachada.getEquiposFilter (
    Categoria cat ) throws RemoteException
```

Es un método mediante la cual se filtran los equipos dependiendo de la categoria

Parámetros

<i>cat</i>	categoria por la cual se filtrará
------------	-----------------------------------

Devuelve

lista filtrada

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.19. getJug()

```
Jugador Fachada.itfFachada.getJug (
    String correo,
    String psw ) throws RemoteException
```

Obtiene todos los datos del jugador segun su correo y contraseña

Parámetros

<i>correo</i>	
<i>psw</i>	

Devuelve

jugador

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.20. `getJugador()`

```
List<Jugador> Fachada.itfFachada.getJugador ( ) throws RemoteException
```

Devuelve la lista de todos los jugadores de la base de datos

Devuelve

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.21. `getMaterial()`

```
List<Material> Fachada.itfFachada.getMaterial ( ) throws RemoteException
```

Obtiene la lista de materiales desde la base de datos

Devuelve

devuelve la lista

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.22. `getPartidos()`

```
List<Partido> Fachada.itfFachada.getPartidos ( ) throws RemoteException, ParseException
```

Coge todos los partidos de la raspberry

Devuelve

lista de partidos

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.23. getRanking()

```
int Fachada.itfFachada.getRanking (
    String dni ) throws ParseException, RemoteException
```

Método para calcular el puesto en el que se encuentra un equipo

Parámetros

<i>dni</i>	del jugador/entrenador del cual se quiere conocer el ranking
------------	--------------------------------------------------------------

Devuelve

devuelve el número del puesto

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.24. jugadorPartido()

```
List<Partido> Fachada.itfFachada.jugadorPartido (
    List< Partido > partidos,
    String DNI ) throws RemoteException, ParseException
```

Con este método se obtienen los partidos de un jugador

Parámetros

<i>partidos</i>	lista de todos los partidos
<i>DNI</i>	del jugador del cual se quieren conocer sus partidos

Devuelve

devuelve lista de partidos

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.25. misJugadores()

```
List<Jugador> Fachada.itfFachada.misJugadores (
    String ent ) throws RemoteException
```

Método para obtener los jugadores relacionados con un entrenador

Parámetros

<i>ent</i>	DNI del entrenador
------------	--------------------

Devuelve

lista de jugadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.26. modificarCorreo()

```
boolean Fachada.itfFachada.modificarCorreo (
    Jugador j,
    String correo,
    String psw ) throws RemoteException
```

Modifica el correo y la contraseña de un jugador

Parámetros

<i>j</i>	jugador del cual los datos se modificaran
<i>correo</i>	
<i>psw</i>	

Devuelve

true si se ha realizado la modificación con éxito

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.27. modificarCorreoEntrenador()

```
boolean Fachada.itfFachada.modificarCorreoEntrenador (
    Entrenador j,
    String correo,
    String psw ) throws RemoteException
```

Mediante este método se modifican el correo y la contraseña del entrenador

Parámetros

<i>j</i>	el objeto entrenador a modificar
<i>correo</i>	
<i>psw</i>	

Devuelve

devuelve un true si la modificación se realiza con éxito

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.28. MostrarJugadores()

```
List<Jugador> Fachada.itfFachada.MostrarJugadores (
    Categoria c ) throws RemoteException
```

Este método sirve para mostrar los jugadores según categoría

Parámetros

<i>c</i>	categoría
----------	-----------

Devuelve

lista de jugadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.29. ObtenerEquipoEntrenador()

```
String Fachada.itfFachada.ObtenerEquipoEntrenador (
    String entrenador ) throws RemoteException
```

Mediante este método se obtiene el equipo asignado al entrenador

Parámetros

<i>entrenador</i>	dni del entrenador
-------------------	--------------------

Devuelve

devuelve el nombre del equipo

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.30. ObtenerEquipoJugador()

```
String Fachada.itfFachada.ObtenerEquipoJugador (
    Jugador jugador ) throws RemoteException
```

Obtiene, mediante este metodo, el equipo en el que juega un jugador

Parámetros

<i>jugador</i>	el jugador en cuestión
----------------	------------------------

Devuelve

devuelve el nombre del equipo

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.31. partidosGanados()

```
int Fachada.itfFachada.partidosGanados (
    String DNI ) throws RemoteException, ParseException
```

Calcula el número total de partidos ganados del equipo de un jugador o entrenador

Parámetros

<i>DNI</i>	del jugador o del entrenador
------------	------------------------------

Devuelve

suma de todos los partidos ganados

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.32. RegistrarAdmin()

```
boolean Fachada.itfFachada.RegistrarAdmin (
    String correo,
    String psw ) throws RemoteException
```

Registra un administrador en la base de datos

Parámetros

<i>correo</i>	
<i>psw</i>	

Devuelve

true si se registra bien

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.33. RegistrarEntrenador()

```
boolean Fachada.itfFachada.RegistrarEntrenador (
    String nombre,
    String apellido,
    String fecha_S,
    String DNI,
    int telefono,
    String correo,
    String psw ) throws RemoteException
```

Registra un entrenador

Parámetros

<i>nombre</i>	
<i>apellido</i>	
<i>fecha</i> ↔ <i>_S</i>	
<i>DNI</i>	
<i>telefono</i>	
<i>correo</i>	
<i>psw</i>	

Devuelve

true si se registra bien

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.34. RegistrarEquipo2()

```
boolean Fachada.itfFachada.RegistrarEquipo2 (
```



```
String nombre,
Categoria cat,
Entrenador entrenador,
List< Jugador > lista_Jugadores,
HashMap< Material, Integer > inventario ) throws RemoteException
```

Registra un nuevo equipo en la base de datos

Parámetros

<i>nombre</i>	
<i>cat</i>	
<i>entrenador</i>	
<i>lista_Jugadores</i>	
<i>inventario</i>	

Devuelve

true si se registra bien

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.35. RegistrarInventario()

```
boolean Fachada.itfFachada.RegistrarInventario (
    Tipo_Material mat,
    int cantidad,
    long precio ) throws RemoteException
```

Registro de un nuevo inventario

Parámetros

<i>mat</i>	tipo material
<i>cantidad</i>	
<i>precio</i>	

Devuelve

true si se registra bien

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.36. RegistrarJugador()

```
boolean Fachada.itfFachada.RegistrarJugador (
    String nombre,
    String apellido,
    String fecha_S,
    String DNI,
    int telefono,
    String correo,
    String psw ) throws RemoteException
```

Crea un nuevo jugador en la base de datos

Parámetros

<i>nombre</i>	
<i>apellido</i>	
<i>fecha</i> ↔ <i>_S</i>	
<i>DNI</i>	
<i>telefono</i>	
<i>correo</i>	
<i>psw</i>	

Devuelve

true si se crea correctamente

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.37. sancionesPersona()

```
List<Sancion> Fachada.itfFachada.sancionesPersona (
    String DNI ) throws RemoteException, ParseException
```

Mediante este método se obtiene una lista de todas las sanciones de una persona

Parámetros

<i>DNI</i>	del jugador o entrenador
------------	--------------------------

Devuelve

lista de sanciones

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Implementado en [Fachada.ServidorPrincipal](#).

6.30.2.38. sancionPartido()

```
Partido Fachada.itfFachada.sancionPartido (
    String codPartido ) throws ParseException, RemoteException
```

Para obtener los datos del partido en el cual se cometieron una sanción

Parámetros

<i>codPartido</i>	
-------------------	--

Devuelve

partido

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	

Implementado en [Fachada.ServidorPrincipal](#).

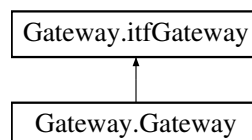
La documentación para este interfaz fue generada a partir del siguiente fichero:

- `src/main/java/Fachada/itfFachada.java`

6.31. Referencia de la Clase Gateway.itfGateway

Interfaz del [Gateway](#) donde se establecen todos los métodos que tendrá el [Gateway](#).

Diagrama de herencias de Gateway.itfGateway



Métodos públicos

- List< Partido > [search_partidos](#) () throws ParseException
- List< Sancion > [search_sanciones](#) () throws ParseException
- ArrayList< Partido > [convertir_partidos](#) (List< Partido_JSON > json)
- ArrayList< Sancion > [convertir_sanciones](#) (List< Sancion_JSON > json)
- HashSet< Equipos_Ext > [getEquipos](#) () throws ParseException
- boolean [partidos](#) () throws ParseException
- boolean [sanciones](#) () throws ParseException

6.31.1. Descripción detallada

Interfaz del [Gateway](#) donde se establecen todos los métodos que tendrá el [Gateway](#).

Autor

Alumno

Definición en la línea 19 del archivo itfGateway.java.

6.31.2. Documentación de las funciones miembro

6.31.2.1. [convertir_partidos\(\)](#)

```
ArrayList<Partido> Gateway.itfGateway.convertir_partidos (
    List< Partido_JSON > json )
```

Parsea el contenido de la raspberry (Partido_JSON) al objeto Partido de la aplicación

Parámetros

<i>json</i>	lista de partidos a parsear
-------------	-----------------------------

Devuelve

lista de partidos parseado

Reimplementado en [Gateway.Gateway](#).

6.31.2.2. [convertir_sanciones\(\)](#)

```
ArrayList<Sancion> Gateway.itfGateway.convertir_sanciones (
    List< Sancion_JSON > json )
```

Parsea el contenido de la raspberry (Sancion_JSON) al objeto Sancion de la aplicación

Parámetros

<i>json</i>	lista de sanciones a parsear
-------------	------------------------------

Devuelve

lista de sanciones parseado

Reimplementado en [Gateway.Gateway](#).

6.31.2.3. getEquipos()

```
HashSet<Equipos_Ext> Gateway.itfGateway.getEquipos ( ) throws ParseException
```

Los partidos tienen dos equipos, mediante este método se obtiene la lista de todos los equipos externos. Donde algunos corresponderán con los equipos del club de fútbol. Así, se podrá calcular la clasificación

Devuelve

devuelve una lista de los equipos

Excepciones

<i>ParseException</i>	
-----------------------	--

Reimplementado en [Gateway.Gateway](#).

6.31.2.4. partidos()

```
boolean Gateway.itfGateway.partidos ( ) throws ParseException
```

Método para ver si las listas de partidos de la raspberry está vacía o no

Devuelve

true si la lista está llena, false si no

Excepciones

<i>ParseException</i>	
-----------------------	--

Reimplementado en [Gateway.Gateway](#).

6.31.2.5. sanciones()

```
boolean Gateway.itfGateway.sanciones ( ) throws ParseException
```

Método para ver si las listas de sanciones de la raspberry está vacía o no

Devuelve

true si la lista está llena, false si no

Excepciones

<i>ParseException</i>	
-----------------------	--

Reimplementado en [Gateway.Gateway](#).

6.31.2.6. search_partidos()

```
List<Partido> Gateway.itfGateway.search_partidos ( ) throws ParseException
```

Método para obtener la lista de partidos de la raspberry

Devuelve

devuelve la lista

Excepciones

<i>ParseException</i>	
-----------------------	--

Reimplementado en [Gateway.Gateway](#).

6.31.2.7. search_sanciones()

```
List<Sancion> Gateway.itfGateway.search_sanciones ( ) throws ParseException
```

Método para obtener la lista de sanciones de la raspberry

Devuelve

devuelve la lista

Excepciones

<code>ParseException</code>	
-----------------------------	--

Reimplementado en [Gateway.Gateway](#).

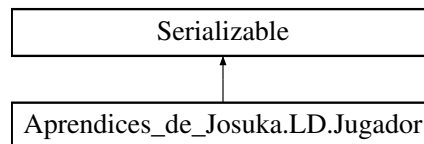
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Gateway/itfGateway.java`

6.32. Referencia de la Clase Aprendices_de_Josuka.LD.Jugador

Esta clase define los objetos de tipo jugador.

Diagrama de herencias de `Aprendices_de_Josuka.LD.Jugador`



Métodos públicos

- [Jugador](#) (String nombre, String apellido, String fecha_nacimiento, String DNI, boolean reconocimiento_medico, boolean estado, int telefono, String correo, String password, boolean cuota_pagada, boolean asignado)
- String [toString](#) ()

6.32.1. Descripción detallada

Esta clase define los objetos de tipo jugador.

Autor

Alumno

Definición en la línea 26 del archivo `Jugador.java`.

6.32.2. Documentación del constructor y destructor

6.32.2.1. Jugador()

```
Aprendices_de_Josuka.LD.Jugador.Jugador (
    String nombre,
    String apellido,
    String fecha_nacimiento,
    String DNI,
    boolean reconocimiento_medico,
    boolean estado,
    int telefono,
    String correo,
    String password,
    boolean cuota_pagada,
    boolean asignado )
```

Definición en la línea 43 del archivo Jugador.java.

6.32.3. Documentación de las funciones miembro

6.32.3.1. toString()

```
String Aprendices_de_Josuka.LD.Jugador.toString ( )
```

Definición en la línea 62 del archivo Jugador.java.

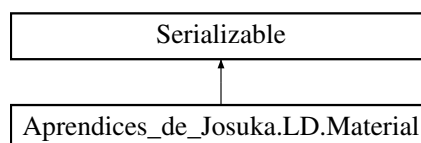
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LD/Jugador.java](#)

6.33. Referencia de la Clase Aprendices_de_Josuka.LD.Material

Esta clase define los objetos de tipo material, es decir, del inventario.

Diagrama de herencias de Aprendices_de_Josuka.LD.Material



Métodos públicos

- [Material](#) ([Tipo_Material](#) tipo, int cantidad, long precio)
- String [toString](#) ()

6.33.1. Descripción detallada

Esta clase define los objetos de tipo material, es decir, del inventario.

Autor

Alumno

Definición en la línea 23 del archivo Material.java.

6.33.2. Documentación del constructor y destructor

6.33.2.1. Material()

```
Aprendices_de_Josuka.LD.Material.Material (
    Tipo_Material tipo,
    int cantidad,
    long precio )
```

Definición en la línea 31 del archivo Material.java.

6.33.3. Documentación de las funciones miembro

6.33.3.1. toString()

```
String Aprendices_de_Josuka.LD.Material.toString ( )
```

Definición en la línea 40 del archivo Material.java.

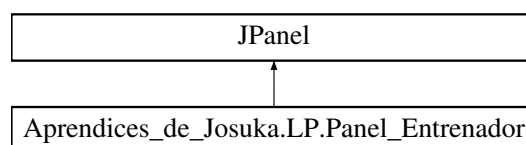
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LD/Material.java](#)

6.34. Referencia de la Clase Aprendices_de_Josuka.LP.Panel_Entrenador

En esta clase se crea un panel con la informacion del entrenador, para luego poder introducirlo en otras ventanas.

Diagrama de herencias de Aprendices_de_Josuka.LP.Panel_Entrenador



Métodos públicos

- [Panel_Entrenador](#) ([Entrenador](#) e)
- void [Introducir_Datos](#) ()

6.34.1. Descripción detallada

En esta clase se crea un panel con la información del entrenador, para luego poder introducirlo en otras ventanas.

Autor

Alumno

Definición en la línea 18 del archivo `Panel_Entrenador.java`.

6.34.2. Documentación del constructor y destructor

6.34.2.1. `Panel_Entrenador()`

```
Aprendices_de_Josuka.LP.Panel_Entrenador.Panel_Entrenador (
    Entrenador e )
```

Constructor

Parámetros

e	
---	--

Definición en la línea 26 del archivo `Panel_Entrenador.java`.

6.34.3. Documentación de las funciones miembro

6.34.3.1. `Introducir_Datos()`

```
void Aprendices_de_Josuka.LP.Panel_Entrenador.Introducir_Datos ( )
```

Se inicializan los componentes del panel y se introducen los datos correspondientes

Definición en la línea 33 del archivo `Panel_Entrenador.java`.

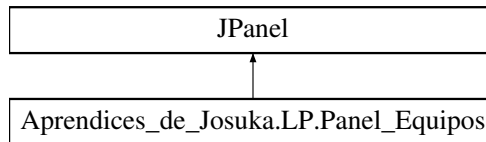
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/Panel_Entrenador.java`

6.35. Referencia de la Clase Aprendices_de_Josuka.LP.Panel_Equipos

En esta clase se crea un panel para mostrar por pantalla cada equipo, para luego poder introducirlo en otras ventanas.

Diagrama de herencias de Aprendices_de_Josuka.LP.Panel_Equipos



Métodos públicos

- [Panel_Equipos](#) ([Equipos_Ext](#) e, int i)
- void [Introducir_Datos](#) ()

6.35.1. Descripción detallada

En esta clase se crea un panel para mostrar por pantalla cada equipo, para luego poder introducirlo en otras ventanas.

Autor

Alumno

Definición en la línea 22 del archivo Panel_Equipos.java.

6.35.2. Documentación del constructor y destructor

6.35.2.1. Panel_Equipos()

```
Aprendices_de_Josuka.LP.Panel_Equipos.Panel_Equipos (  
    Equipos\_Ext e,  
    int i )
```

Constructor

Parámetros

<i>e</i>	
<i>i</i>	

Definición en la línea 32 del archivo Panel_Equipos.java.

6.35.3. Documentación de las funciones miembro

6.35.3.1. Introducir_Datos()

```
void Aprendices_de_Josuka.LP.Panel_Equipos.Introducir_Datos ( )
```

Se inicializan los componentes del panel y se introducen los datos correspondientes

Definición en la línea 43 del archivo Panel_Equipos.java.

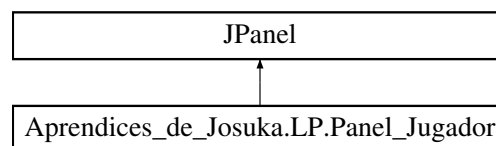
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/Panel_Equipos.java`

6.36. Referencia de la Clase Aprendices_de_Josuka.LP.Panel_Jugador

En esta clase se crea un panel para mostrar por pantalla cada jugador, para luego poder introducirlo en otras ventanas.

Diagrama de herencias de Aprendices_de_Josuka.LP.Panel_Jugador



Métodos públicos

- `Panel_Jugador (Jugador j)`
- `void Introducir_Datos ()`

6.36.1. Descripción detallada

En esta clase se crea un panel para mostrar por pantalla cada jugador, para luego poder introducirlo en otras ventanas.

Autor

Alumno

Definición en la línea 25 del archivo Panel_Jugador.java.

6.36.2. Documentación del constructor y destructor

6.36.2.1. Panel_Jugador()

```
Aprendices_de_Josuka.LP.Panel_Jugador.Panel_Jugador (
    Jugador j )
```

Constructor

Parámetros

<i>j</i>	
----------	--

Definición en la línea 32 del archivo Panel_Jugador.java.

6.36.3. Documentación de las funciones miembro

6.36.3.1. Introducir_Datos()

```
void Aprendices_de_Josuka.LP.Panel_Jugador.Introducir_Datos ( )
```

Se inicializan los componentes del panel y se introducen los datos correspondientes

Definición en la línea 40 del archivo Panel_Jugador.java.

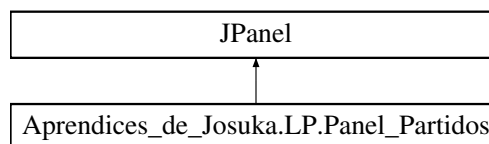
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/Panel_Jugador.java`

6.37. Referencia de la Clase Aprendices_de_Josuka.LP.Panel_Partidos

En esta clase se crea un panel para mostrar por pantalla cada partido, para luego poder introducirlo en otras ventanas.

Diagrama de herencias de Aprendices_de_Josuka.LP.Panel_Partidos



Métodos públicos

- `Panel_Partidos (Partido p)`
- `void Introducir_Datos ()`

6.37.1. Descripción detallada

En esta clase se crea un panel para mostrar por pantalla cada partido, para luego poder introducirlo en otras ventanas.

Autor

Alumno

Definición en la línea 28 del archivo Panel_Partidos.java.

6.37.2. Documentación del constructor y destructor

6.37.2.1. Panel_Partidos()

```
Aprendices_de_Josuka.LP.Panel_Partidos.Panel_Partidos (
    Partido p )
```

Constructor

Parámetros

<i>p</i>	
----------	--

Definición en la línea 34 del archivo Panel_Partidos.java.

6.37.3. Documentación de las funciones miembro

6.37.3.1. Introducir_Datos()

```
void Aprendices_de_Josuka.LP.Panel_Partidos.Introducir_Datos ( )
```

Se inicializan los componentes del panel y se introducen los datos correspondientes

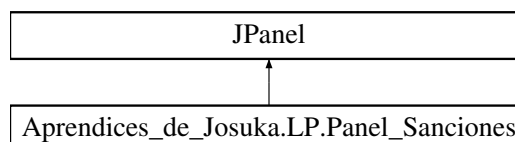
Definición en la línea 42 del archivo Panel_Partidos.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/Panel_Partidos.java`

6.38. Referencia de la Clase Aprendices_de_Josuka.LP.Panel_Sanciones

Diagrama de herencias de Aprendices_de_Josuka.LP.Panel_Sanciones



Métodos públicos

- `Panel_Sanciones (Sancion p, Partido partido)`
- `void Introducir_Datos ()`

6.38.1. Descripción detallada

Definición en la línea 29 del archivo Panel_Sanciones.java.

6.38.2. Documentación del constructor y destructor

6.38.2.1. Panel_Sanciones()

```
Aprendices_de_Josuka.LP.Panel_Sanciones.Panel_Sanciones (
    Sancion p,
    Partido partido )
```

Constructor

Parámetros

<i>p</i>	
<i>partido</i>	

Definición en la línea 37 del archivo Panel_Sanciones.java.

6.38.3. Documentación de las funciones miembro

6.38.3.1. Introducir_Datos()

```
void Aprendices_de_Josuka.LP.Panel_Sanciones.Introducir_Datos ( )
```

Se inicializan los componentes del panel y se introducen los datos correspondientes

Definición en la línea 45 del archivo Panel_Sanciones.java.

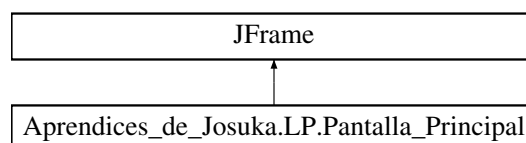
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Aprendices_de_Josuka/LP/[Panel_Sanciones.java](#)

6.39. Referencia de la Clase Aprendices_de_Josuka.LP.Pantalla_Principal

En esta clase se desarrolla la GUI para la pantalla principal de la aplicacion.

Diagrama de herencias de Aprendices_de_Josuka.LP.Pantalla_Principal



Métodos públicos

- `Pantalla_Principal` (`Controller c`)
- `void initComponents` (`Controller c`)

Métodos públicos estáticos

- `static void main` (`String[] args`)

6.39.1. Descripción detallada

En esta clase se desarrolla la GUI para la pantalla principal de la aplicacion.

Autor

Alumno

Definición en la línea 26 del archivo `Pantalla_Principal.java`.

6.39.2. Documentación del constructor y destructor

6.39.2.1. `Pantalla_Principal()`

```
Aprendices_de_Josuka.LP.Pantalla_Principal.Pantalla_Principal (
    Controller c )
```

Constructor

Parámetros

<code>c</code>	
----------------	--

Definición en la línea 65 del archivo `Pantalla_Principal.java`.

6.39.3. Documentación de las funciones miembro

6.39.3.1. `initComponents()`

```
void Aprendices_de_Josuka.LP.Pantalla_Principal.initComponents (
    Controller c )
```

Se inicializan los componentes

Parámetros

<i>c</i>	
----------	--

Definición en la línea 73 del archivo Pantalla_Principal.java.

6.39.3.2. main()

```
static void Aprendices_de_Josuka.LP.Pantalla_Principal.main (
    String[] args ) [static]
```

Se ejecuta la aplicacion

Parámetros

<i>args</i>	
-------------	--

Definición en la línea 49 del archivo Pantalla_Principal.java.

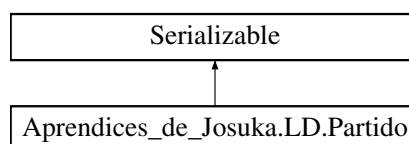
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/Pantalla_Principal.java](#)

6.40. Referencia de la Clase Aprendices_de_Josuka.LD.Partido

Esta clase define objetos de tipo partido que no están guardados en la base de datos y se obtienen de la raspberry.

Diagrama de herencias de Aprendices_de_Josuka.LD.Partido

**Métodos públicos**

- [Partido](#) (String cod_partido, LocalDateTime fecha, long resultado_e1, long resultado_e2, [Equipos_Ext](#) equipo_1, [Equipos_Ext](#) equipo_2)

6.40.1. Descripción detallada

Esta clase define objetos de tipo partido que no están guardados en la base de datos y se obtienen de la raspberry.

Autor

Alumno

Definición en la línea 22 del archivo Partido.java.

6.40.2. Documentación del constructor y destructor

6.40.2.1. Partido()

```
Aprendices_de_Josuka.LD.Partido.Partido (
    String cod_partido,
    LocalDateTime fecha,
    long resultado_e1,
    long resultado_e2,
    Equipos_Ext equipo_1,
    Equipos_Ext equipo_2 )
```

Definición en la línea 34 del archivo Partido.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Aprendices_de_Josuka/LD/[Partido.java](#)

6.41. Referencia de la Clase python.Partido.Partido.Partido

Métodos públicos

- def `__init__` (self, str code, Equipo equipo_1, Equipo equipo_2, int resultado_e1, int resultado_e2, datetime date)
- str `get_code` (self)
- Equipo `get_equipo_1` (self)
- Equipo `get_equipo_2` (self)
- int `get_resultado_e1` (self)
- int `get_resultado_e2` (self)
- datetime `get_date` (self)
- None `print` (self)

6.41.1. Descripción detallada

Definición en la línea 11 del archivo Partido.py.

6.41.2. Documentación del constructor y destructor

6.41.2.1. `__init__()`

```
def python.Partido.Partido.Partido.__init__ (
    self,
    str code,
    Equipo equipo_1,
    Equipo equipo_2,
    int resultado_e1,
    int resultado_e2,
    datetime date )
```

Definición en la línea 13 del archivo Partido.py.

6.41.3. Documentación de las funciones miembro

6.41.3.1. `get_code()`

```
str python.Partido.Partido.Partido.get_code (
    self )
```

Definición en la línea 23 del archivo Partido.py.

6.41.3.2. `get_date()`

```
datetime python.Partido.Partido.Partido.get_date (
    self )
```

Definición en la línea 43 del archivo Partido.py.

6.41.3.3. `get_equipo_1()`

```
Equipo python.Partido.Partido.Partido.get_equipo_1 (
    self )
```

Definición en la línea 27 del archivo Partido.py.

6.41.3.4. `get_equipo_2()`

```
Equipo python.Partido.Partido.Partido.get_equipo_2 (
    self )
```

Definición en la línea 31 del archivo Partido.py.

6.41.3.5. `get_resultado_e1()`

```
int python.Partido.Partido.Partido.get_resultado_e1 (  
    self )
```

Definición en la línea 35 del archivo Partido.py.

6.41.3.6. `get_resultado_e2()`

```
int python.Partido.Partido.Partido.get_resultado_e2 (  
    self )
```

Definición en la línea 39 del archivo Partido.py.

6.41.3.7. `print()`

```
None python.Partido.Partido.Partido.print (  
    self )
```

Definición en la línea 46 del archivo Partido.py.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/python/Partido/Partido.py`

6.42. Referencia de la Clase `ServiciosExternos.Partido_JSON`

Clase para parsear los partidos de la raspberry.

Métodos públicos

- `Partido_JSON` (Object json_encoded)
- String `getCode` ()
- long `getResE1` ()
- long `getResE2` ()
- String `getEquipo1Nombre` ()
- String `getEquipo2Nombre` ()
- String `getEquipo1Categoria` ()
- String `getEquipo2Categoria` ()
- String `getEquipo1Code` ()
- String `getEquipo2Code` ()
- long `getEquipo1Puntuacion` ()
- long `getEquipo2Puntuacion` ()
- String `getDate` ()
- LocalDateTime `getDate` (boolean withHours)

6.42.1. Descripción detallada

Clase para parsear los partidos de la raspberry.

Autor

Alumno

Definición en la línea 13 del archivo Partido_JSON.java.

6.42.2. Documentación del constructor y destructor

6.42.2.1. Partido_JSON()

```
ServiciosExternos.Partido_JSON.Partido_JSON (
    Object json_encoded )
```

Constructor de la clase

Parámetros

<i>json_encoded</i>	objeto que se va a parsear
---------------------	----------------------------

Definición en la línea 20 del archivo Partido_JSON.java.

6.42.3. Documentación de las funciones miembro

6.42.3.1. getCode()

```
String ServiciosExternos.Partido_JSON.getCode ( )
```

Método para crear correspondencia del código del partido

Devuelve

el código del partido

Definición en la línea 28 del archivo Partido_JSON.java.

6.42.3.2. getDate() [1/2]

```
String ServiciosExternos.Partido_JSON.getDate ( )
```

Método para crear correspondencia de la fecha del partido

Devuelve

la fecha del partido

Definición en la línea 116 del archivo Partido_JSON.java.

6.42.3.3. getDate() [2/2]

```
LocalDateTime ServiciosExternos.Partido_JSON.getDate (
    boolean withHours )
```

Este método sirve para parsear las fechas que se consiguen de la raspberry, pues son string y hay que convertirlos en LocalDateTime

Parámetros

<i>withHours</i>	booleano para coger horas
------------------	---------------------------

Devuelve

devuelve la fecha formateada

Definición en la línea 125 del archivo Partido_JSON.java.

6.42.3.4. getEquipo1Categoria()

```
String ServiciosExternos.Partido_JSON.getEquipo1Categoria ( )
```

Método para parsear la categoría del primer equipo

Devuelve

la categoría del primer equipo

Definición en la línea 68 del archivo Partido_JSON.java.

6.42.3.5. getEquipo1Code()

```
String ServiciosExternos.Partido_JSON.getEquipo1Code ( )
```

Método para parsear el código del primer equipo

Devuelve

el código del primer equipo

Definición en la línea 84 del archivo Partido_JSON.java.

6.42.3.6. getEquipo1Nombre()

```
String ServiciosExternos.Partido_JSON.getEquipo1Nombre ( )
```

Método para crear correspondencia del nombre del primer equipo

Devuelve

el nombre del primer equipo

Definición en la línea 52 del archivo Partido_JSON.java.

6.42.3.7. getEquipo1Puntuacion()

```
long ServiciosExternos.Partido_JSON.getEquipo1Puntuacion ( )
```

Método para parsear la puntuación del primer equipo

Devuelve

la puntuación del equipo uno

Definición en la línea 100 del archivo Partido_JSON.java.

6.42.3.8. getEquipo2Categoria()

```
String ServiciosExternos.Partido_JSON.getEquipo2Categoria ( )
```

Método para parsear la categoría del segundo equipo

Devuelve

la categoría del segundo equipo

Definición en la línea 76 del archivo Partido_JSON.java.

6.42.3.9. `getEquipo2Code()`

```
String ServiciosExternos.Partido_JSON.getEquipo2Code ( )
```

Método para parsear el código del segundo equipo

Devuelve

el código del segundo equipo

Definición en la línea 92 del archivo Partido_JSON.java.

6.42.3.10. `getEquipo2Nombre()`

```
String ServiciosExternos.Partido_JSON.getEquipo2Nombre ( )
```

Método para crear correspondencia del nombre del segundo equipo

Devuelve

el nombre del segundo equipo

Definición en la línea 60 del archivo Partido_JSON.java.

6.42.3.11. `getEquipo2Puntuacion()`

```
long ServiciosExternos.Partido_JSON.getEquipo2Puntuacion ( )
```

Método para parsear la puntuación del segundo equipo

Devuelve

la puntuación del equipo dos

Definición en la línea 108 del archivo Partido_JSON.java.

6.42.3.12. `getResE1()`

```
long ServiciosExternos.Partido_JSON.getResE1 ( )
```

Método para crear correspondencia del resultado del primer equipo

Devuelve

el resultado del primer equipo

Definición en la línea 36 del archivo Partido_JSON.java.

6.42.3.13. getResE2()

```
long ServiciosExternos.Partido_JSON.getResE2 ( )
```

Método para crear correspondencia del resultado del segundo equipo

Devuelve

el resultado del segundo equipo

Definición en la línea 44 del archivo Partido_JSON.java.

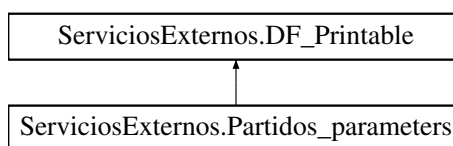
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/ServiciosExternos/[Partido_JSON.java](#)

6.43. Referencia de la Clase ServiciosExternos.Partidos_parameters

Clase que contiene los parametros de los partidos y sus getters y setters.

Diagrama de herencias de ServiciosExternos.Partidos_parameters



Métodos públicos

- void [print](#) ()

6.43.1. Descripción detallada

Clase que contiene los parametros de los partidos y sus getters y setters.

Autor

Alumno

Definición en la línea 13 del archivo Partidos_parameters.java.

6.43.2. Documentación de las funciones miembro

6.43.2.1. print()

```
void ServiciosExternos.Partidos_parameters.print ( )
```

Imprime los parametros de los partidos

Reimplementado de [ServiciosExternos.DF_Printable](#).

Definición en la línea 24 del archivo Partidos_parameters.java.

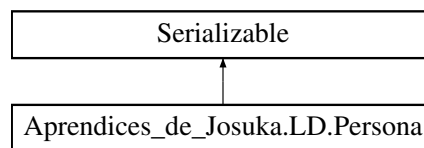
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/ServiciosExternos/Partidos_parameters.java`

6.44. Referencia de la Clase Aprendices_de_Josuka.LD.Persona

Esta clase define objetos de tipo persona (clase padre), de la que después heredarán las clases entrenador y jugador (las clases hijas)

Diagrama de herencias de Aprendices_de_Josuka.LD.Persona



Métodos públicos

- [Persona](#) (String dNI, String nombre, String apellido, String fecha_nacimiento, int telefono, String correo, String psw)
- String [toString](#) ()

6.44.1. Descripción detallada

Esta clase define objetos de tipo persona (clase padre), de la que después heredarán las clases entrenador y jugador (las clases hijas)

Autor

Alumno

Definición en la línea 19 del archivo Persona.java.

6.44.2. Documentación del constructor y destructor

6.44.2.1. Persona()

```
Aprendices_de_Josuka.LD.Persona.Persona (
    String dni,
    String nombre,
    String apellido,
    String fecha_nacimiento,
    int telefono,
    String correo,
    String psw )
```

Definición en la línea 34 del archivo Persona.java.

6.44.3. Documentación de las funciones miembro

6.44.3.1. toString()

```
String Aprendices_de_Josuka.LD.Persona.toString ( )
```

Definición en la línea 49 del archivo Persona.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

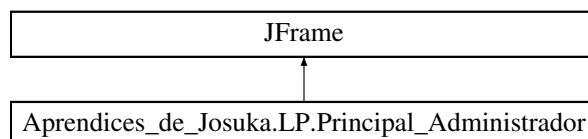
- `src/main/java/Aprendices_de_Josuka/LD/Persona.java`

6.45. Referencia de la Clase

Aprendices_de_Josuka.LP.Principal_Administrador

En esta clase se desarrolla la GUI que visualiza el administrador cuando abre la aplicacion despues de haber iniciado sesion.

Diagrama de herencias de Aprendices_de_Josuka.LP.Principal_Administrador



Métodos públicos

- `Principal_Administrador` (`Controller` controller) throws `RemoteException`, `ParseException`
- void `initComponents` ()
- void `InsertarJPanel` ()

6.45.1. Descripción detallada

En esta clase se desarrolla la GUI que visualiza el administrador cuando abre la aplicación después de haber iniciado sesión.

Autor

Alumno

Definición en la línea 32 del archivo `Principal_Administrador.java`.

6.45.2. Documentación del constructor y destructor

6.45.2.1. `Principal_Administrador()`

```
Aprendices_de_Josuka.LP.Principal_Administrador.Principal_Administrador (
    Controller controller ) throws RemoteException, ParseException
```

Constructor

Parámetros

<i>controller</i>	
-------------------	--

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 62 del archivo `Principal_Administrador.java`.

6.45.3. Documentación de las funciones miembro

6.45.3.1. `initComponents()`

```
void Aprendices_de_Josuka.LP.Principal_Administrador.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 72 del archivo `Principal_Administrador.java`.

6.45.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Principal_Administrador.InsertarJPanel ( )
```

Se inserta un JPanel donde se visualizan los partidos segun la categoria

Definición en la línea 343 del archivo Principal_Administrador.java.

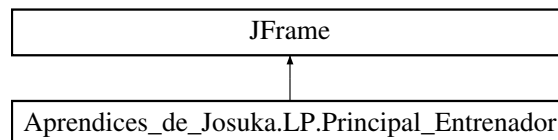
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/Principal_Administrador.java](#)

6.46. Referencia de la Clase Aprendices_de_Josuka.LP.Principal_Entrenador

En esta clase se desarrolla la GUI que visualiza el entrenador cuando abre la aplicacion despues de haber iniciado sesion.

Diagrama de herencias de Aprendices_de_Josuka.LP.Principal_Entrenador



Métodos públicos

- [Principal_Entrenador](#) ([Controller](#) controller, [Entrenador](#) j) throws RemoteException, ParseException
- void [initComponents](#) ()
- void [InsertarJPanel](#) ()

6.46.1. Descripción detallada

En esta clase se desarrolla la GUI que visualiza el entrenador cuando abre la aplicacion despues de haber iniciado sesion.

Autor

Alumno

Definición en la línea 37 del archivo Principal_Entrenador.java.

6.46.2. Documentación del constructor y destructor

6.46.2.1. Principal_Entrenador()

```
Aprendices_de_Josuka.LP.Principal_Entrenador.Principal_Entrenador (
    Controller controller,
    Entrenador j ) throws RemoteException, ParseException
```

Constructor

Parámetros

<i>controller</i>	el objeto controller que está conectado al servidor
<i>j</i>	el entrenador que está usando la aplicación

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 65 del archivo Principal_Entrenador.java.

6.46.3. Documentación de las funciones miembro**6.46.3.1. initComponents()**

```
void Aprendices_de_Josuka.LP.Principal_Entrenador.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 76 del archivo Principal_Entrenador.java.

6.46.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Principal_Entrenador.InsertarJPanel ( )
```

Se inserta un JPanel donde se visualizan los partidos segun la categoria

Definición en la línea 307 del archivo Principal_Entrenador.java.

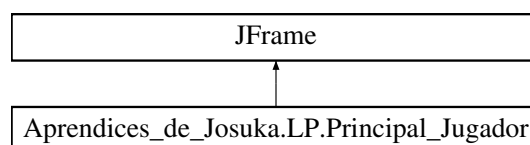
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/Principal_Entrenador.java`

6.47. Referencia de la Clase**Aprendices_de_Josuka.LP.Principal_Jugador**

En esta clase se desarrolla la GUI que visualiza el jugador cuando abre la aplicacion despues de haber iniciado sesion.

Diagrama de herencias de Aprendices_de_Josuka.LP.Principal_Jugador



Métodos públicos

- `Principal_Jugador` (`Controller` controller, `Jugador` j) throws `RemoteException`, `ParseException`
- void `initComponents` ()
- void `InsertarJPanel` ()
- void `InsertarMisPartidos` () throws `RemoteException`, `ParseException`

6.47.1. Descripción detallada

En esta clase se desarrolla la GUI que visualiza el jugador cuando abre la aplicacion despues de haber iniciado sesion.

Autor

Alumno

Definición en la línea 34 del archivo `Principal_Jugador.java`.

6.47.2. Documentación del constructor y destructor

6.47.2.1. `Principal_Jugador()`

```
Aprendices_de_Josuka.LP.Principal_Jugador.Principal_Jugador (  
    Controller controller,  
    Jugador j ) throws RemoteException, ParseException
```

Constructor

Parámetros

<i>controller</i>	el objeto controller que está conectado al servidor
<i>j</i>	el jugador que está usando la aplicación

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 67 del archivo `Principal_Jugador.java`.

6.47.3. Documentación de las funciones miembro

6.47.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.Principal_Jugador.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 78 del archivo Principal_Jugador.java.

6.47.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Principal_Jugador.InsertarJPanel ( )
```

Se inserta un JPanel donde se visualizan los partidos segun la categoria

Definición en la línea 303 del archivo Principal_Jugador.java.

6.47.3.3. InsertarMisPartidos()

```
void Aprendices_de_Josuka.LP.Principal_Jugador.InsertarMisPartidos ( ) throws RemoteException,
ParseException
```

En caso de pulsar el botón MIS PARTIDOS, se ejecutara este metodo donde se muestran los partidos del jugador en particular

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Definición en la línea 339 del archivo Principal_Jugador.java.

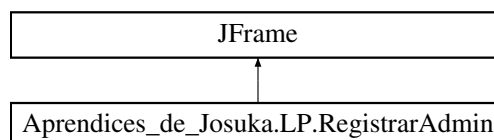
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/Principal_Jugador.java](#)

6.48. Referencia de la Clase Aprendices_de_Josuka.LP.RegistrarAdmin

En esta clase se desarrolla la GUI para el registro del administrador.

Diagrama de herencias de Aprendices_de_Josuka.LP.RegistrarAdmin



Métodos públicos

- [RegistrarAdmin](#) ([Controller](#) controller)
- void [initComponents](#) ()

6.48.1. Descripción detallada

En esta clase se desarrolla la GUI para el registro del administrador.

Autor

Alumno

Definición en la línea 26 del archivo RegistrarAdmin.java.

6.48.2. Documentación del constructor y destructor

6.48.2.1. RegistrarAdmin()

```
Aprendices_de_Josuka.LP.RegistrarAdmin.RegistrarAdmin (
    Controller controller )
```

Constructor

Parámetros

controller	
----------------------------	--

Definición en la línea 48 del archivo RegistrarAdmin.java.

6.48.3. Documentación de las funciones miembro

6.48.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.RegistrarAdmin.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 57 del archivo RegistrarAdmin.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

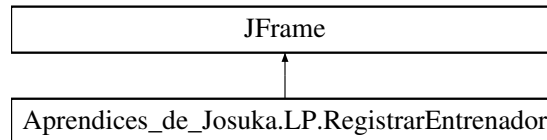
- src/main/java/Aprendices_de_Josuka/LP/[RegistrarAdmin.java](#)

6.49. Referencia de la Clase

Aprendices_de_Josuka.LP.RegistrarEntrenador

En esta clase se desarrolla la GUI para el registro del entrenador.

Diagrama de herencias de Aprendices_de_Josuka.LP.RegistrarEntrenador



Métodos públicos

- `RegistrarEntrenador` (`Controller` controller)
- `void initComponents` ()

6.49.1. Descripción detallada

En esta clase se desarrolla la GUI para el registro del entrenador.

Autor

Alumno

Definición en la línea 32 del archivo `RegistrarEntrenador.java`.

6.49.2. Documentación del constructor y destructor

6.49.2.1. RegistrarEntrenador()

```
Aprendices_de_Josuka.LP.RegistrarEntrenador.RegistrarEntrenador (
    Controller controller )
```

Constructor

Parámetros

<code>controller</code>	
-------------------------	--

Definición en la línea 53 del archivo `RegistrarEntrenador.java`.

6.49.3. Documentación de las funciones miembro

6.49.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.RegistrarEntrenador.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 62 del archivo RegistrarEntrenador.java.

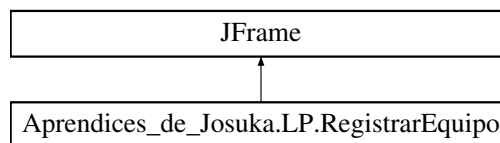
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/RegistrarEntrenador.java](#)

6.50. Referencia de la Clase Aprendices_de_Josuka.LP.RegistrarEquipo

En esta clase se desarrolla la GUI para cuando un administrador quiera registrar un equipo.

Diagrama de herencias de Aprendices_de_Josuka.LP.RegistrarEquipo



Métodos públicos

- [RegistrarEquipo](#) ([Controller](#) controller)
- void [initComponents](#) ()
- void [MostrarJugadores](#) ()
- void [PasarJugadores](#) ()
- void [EliminarJugadores](#) ()
- void [vaciarJList](#) ()
- void [Anyadir_Material](#) ()
- void [Anyadir_Equipo](#) () throws RemoteException
- void [RellenarEntrenadores](#) ()
- void [AsignarMaterial](#) ([Material](#) m, int i)

6.50.1. Descripción detallada

En esta clase se desarrolla la GUI para cuando un administrador quiera registrar un equipo.

Autor

Alumno

Definición en la línea 41 del archivo RegistrarEquipo.java.

6.50.2. Documentación del constructor y destructor

6.50.2.1. RegistrarEquipo()

```
Aprendices_de_Josuka.LP.RegistrarEquipo.RegistrarEquipo (
    Controller controller )
```

Constructor

Parámetros

<i>controller</i>	
-------------------	--

Definición en la línea 85 del archivo RegistrarEquipo.java.

6.50.3. Documentación de las funciones miembro

6.50.3.1. Anyadir_Equipo()

```
void Aprendices_de_Josuka.LP.RegistrarEquipo.Anyadir_Equipo ( ) throws RemoteException
```

Método para guardar el equipo introducido

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 544 del archivo RegistrarEquipo.java.

6.50.3.2. Anyadir_Material()

```
void Aprendices_de_Josuka.LP.RegistrarEquipo.Anyadir_Material ( )
```

Método para que se abra la ventana de anyadir inventario

Definición en la línea 534 del archivo RegistrarEquipo.java.

6.50.3.3. AsignarMaterial()

```
void Aprendices_de_Josuka.LP.RegistrarEquipo.AsignarMaterial (
    Material m,
    int i )
```

Metodo para asignar el material seleccionado al equipo

Parámetros

<i>m</i>	
<i>i</i>	

Definición en la línea 606 del archivo RegistrarEquipo.java.

6.50.3.4. EliminarJugadores()

```
void Aprendices_de_Josuka.LP.RegistrarEquipo.EliminarJugadores ( )
```

Metodo para eliminar alguno de los jugadores seleccionados, en caso de que no se haya seleccionado el adecuado

Definición en la línea 512 del archivo RegistrarEquipo.java.

6.50.3.5. initComponents()

```
void Aprendices_de_Josuka.LP.RegistrarEquipo.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 95 del archivo RegistrarEquipo.java.

6.50.3.6. MostrarJugadores()

```
void Aprendices_de_Josuka.LP.RegistrarEquipo.MostrarJugadores ( )
```

Se muestran los jugadores disponibles

Definición en la línea 467 del archivo RegistrarEquipo.java.

6.50.3.7. PasarJugadores()

```
void Aprendices_de_Josuka.LP.RegistrarEquipo.PasarJugadores ( )
```

Metodo para pasar los jugadores seleccionados al nuevo equipo

Definición en la línea 497 del archivo RegistrarEquipo.java.

6.50.3.8. RellenarEntrenadores()

```
void Aprendices_de_Josuka.LP.RegistrarEquipo.RellenarEntrenadores ( )
```

Metodo para rellenar el ComboBox de los entrenadores

Definición en la línea 588 del archivo RegistrarEquipo.java.

6.50.3.9. vaciarJList()

```
void Aprendices_de_Josuka.LP.RegistrarEquipo.vaciarJList ( )
```

Metodo para vaciar la lista de los jugadores

Definición en la línea 523 del archivo RegistrarEquipo.java.

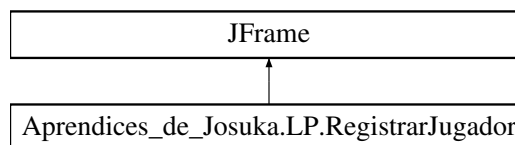
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/RegistrarEquipo.java`

6.51. Referencia de la Clase Aprendices_de_Josuka.LP.RegistrarJugador

En esta clase se desarrolla la GUI para el registro del jugador.

Diagrama de herencias de Aprendices_de_Josuka.LP.RegistrarJugador



Métodos públicos

- `RegistrarJugador` (Controller controller)
- `void initComponents` ()
- `void guardar` () throws ParseException

6.51.1. Descripción detallada

En esta clase se desarrolla la GUI para el registro del jugador.

Autor

Alumno

Definición en la línea 32 del archivo RegistrarJugador.java.

6.51.2. Documentación del constructor y destructor

6.51.2.1. RegistrarJugador()

```
Aprendices_de_Josuka.LP.RegistrarJugador.RegistrarJugador (
    Controller controller )
```

Constructor

Parámetros

<i>controller</i>	
-------------------	--

Definición en la línea 57 del archivo RegistrarJugador.java.

6.51.3. Documentación de las funciones miembro

6.51.3.1. guardar()

```
void Aprendices_de_Josuka.LP.RegistrarJugador.guardar ( ) throws ParseException
```

Guarda los datos del jugador en la base de datos y entra en la aplicación

Excepciones

<i>ParseException</i>	
-----------------------	--

Definición en la línea 209 del archivo RegistrarJugador.java.

6.51.3.2. initComponents()

```
void Aprendices_de_Josuka.LP.RegistrarJugador.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 66 del archivo RegistrarJugador.java.

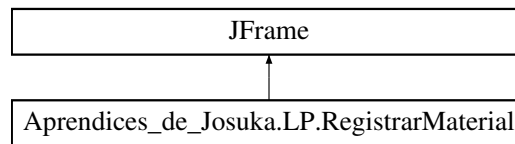
La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LP/RegistrarJugador.java`

6.52. Referencia de la Clase Aprendices_de_Josuka.LP.RegistrarMaterial

En esta clase se desarrolla la GUI para que el administrador pueda registrar el material.

Diagrama de herencias de Aprendices_de_Josuka.LP.RegistrarMaterial



Métodos públicos

- `RegistrarMaterial` (`Controller` controller)
- `void initComponents` ()
- `void crear_Material` ()

6.52.1. Descripción detallada

En esta clase se desarrolla la GUI para que el administrador pueda registrar el material.

Autor

Alumno

Definición en la línea 30 del archivo RegistrarMaterial.java.

6.52.2. Documentación del constructor y destructor

6.52.2.1. RegistrarMaterial()

```
Aprendices_de_Josuka.LP.RegistrarMaterial.RegistrarMaterial (
    Controller controller )
```

Launch the application. Constructor

Parámetros

<code>controller</code>	
-------------------------	--

Definición en la línea 70 del archivo RegistrarMaterial.java.

6.52.3. Documentación de las funciones miembro

6.52.3.1. crear_Material()

```
void Aprendices_de_Josuka.LP.RegistrarMaterial.crear_Material ( )
```

Método para guardar el material introducido

Definición en la línea 354 del archivo RegistrarMaterial.java.

6.52.3.2. initComponents()

```
void Aprendices_de_Josuka.LP.RegistrarMaterial.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 79 del archivo RegistrarMaterial.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/RegistrarMaterial.java](#)

6.53. Referencia de la plantilla de la Clase

ServiciosExternos.RestClient< T extends DF_Printable >

Sirve para conectar dos aplicaciones. En este caso la raspberry y la aplicación de Aprendices de Josuka.

Métodos públicos

- boolean [request_OK](#) (int response_status)
- [RestClient](#) (String hostname, String port)
- Invocation.Builder [createInvocationBuilder](#) (String resource_URL) throws Exception
- Response [makeGetRequest](#) (Invocation.Builder invocationBuilder) throws Exception
- Response [makePostRequest](#) (Invocation.Builder invocationBuilder, T objectToSend) throws Exception
- Response [makePutRequest](#) (Invocation.Builder invocationBuilder, T objectToSend) throws Exception
- void [simplePrint](#) (Response response)

Atributos protegidos

- Client [client](#)
- WebTarget [webTarget](#)

6.53.1. Descripción detallada

Sirve para conectar dos aplicaciones. En este caso la raspberry y la aplicación de Aprendices de Josuka.

Autor

Alumno

Parámetros

< T >	objeto del que se hara POST o PUT
-------	-----------------------------------

Definición en la línea 14 del archivo RestClient.java.

6.53.2. Documentación del constructor y destructor**6.53.2.1. RestClient()**

```
ServiciosExternos.RestClient< T extends DF_Printable >.RestClient (
    String hostname,
    String port )
```

Constructor de la clase

Parámetros

<i>hostname</i>	el host con el que se realiza la conexión
<i>port</i>	puerto en el que se realiza la conexión

Definición en la línea 34 del archivo RestClient.java.

6.53.3. Documentación de las funciones miembro**6.53.3.1. createInvocationBuilder()**

```
Invocation.Builder ServiciosExternos.RestClient< T extends DF_Printable >.createInvocationBuilder (
    String resource_URL ) throws Exception
```

Definición en la línea 41 del archivo RestClient.java.

6.53.3.2. makeGetRequest()

```
Response ServiciosExternos.RestClient< T extends DF_Printable >.makeGetRequest (
    Invocation.Builder invocationBuilder ) throws Exception
```

Método donde se realiza una petición GET

Parámetros

<i>invocationBuilder</i>	una interfaz que proporciona métodos para preparar la petición del cliente y también para invocarla.
--------------------------	------------------------------------------------------------------------------------------------------

Devuelve

devuelve la respuesta de la petición

Excepciones

<i>Exception</i>	
------------------	--

Definición en la línea 63 del archivo RestClient.java.

6.53.3.3. makePostRequest()

```
Response ServiciosExternos.RestClient< T extends DF_Printable >.makePostRequest (
    Invocation.Builder invocationBuilder,
    T objectToSend ) throws Exception
```

Método donde se realiza una petición POST

Parámetros

<i>invocationBuilder</i>	una interfaz que proporciona métodos para preparar la petición del cliente y también para invocarla.
<i>objectToSend</i>	la entidad que se va a mandar

Devuelve

devuelve la respuesta de la petición

Excepciones

<i>Exception</i>	
------------------	--

Definición en la línea 75 del archivo RestClient.java.

6.53.3.4. makePutRequest()

```
Response ServiciosExternos.RestClient< T extends DF_Printable >.makePutRequest (
    Invocation.Builder invocationBuilder,
    T objectToSend ) throws Exception
```

Método donde se realiza una petición PUT

Parámetros

<i>invocationBuilder</i>	una interfaz que proporciona métodos para preparar la petición del cliente y también para invocarla.
<i>objectToSend</i>	la entidad que se va a mandar

Devuelve

devuelve la respuesta de la petición

Excepciones

<i>Exception</i>	
------------------	--

Definición en la línea 88 del archivo RestClient.java.

6.53.3.5. request_OK()

```
boolean ServiciosExternos.RestClient< T extends DF_Printable >.request_OK (
    int response_status )
```

Parsea la respuesta de la solicitud

Parámetros

<i>response_status</i>	código de la respuesta
------------------------	------------------------

Devuelve

respuesta parseada a OK, CREADA o ACEPTADA

Definición en la línea 23 del archivo RestClient.java.

6.53.3.6. simplePrint()

```
void ServiciosExternos.RestClient< T extends DF_Printable >.simplePrint (
    Response response )
```

Se imprime la respuesta como un string

Parámetros

<i>response</i>	la respuesta que debe imprimir
-----------------	--------------------------------

Definición en la línea 98 del archivo RestClient.java.

6.53.4. Documentación de los datos miembro

6.53.4.1. client

```
Client ServiciosExternos.RestClient< T extends DF\_Printable >.client [protected]
```

Definición en la línea 16 del archivo RestClient.java.

6.53.4.2. webTarget

```
WebTarget ServiciosExternos.RestClient< T extends DF\_Printable >.webTarget [protected]
```

Definición en la línea 17 del archivo RestClient.java.

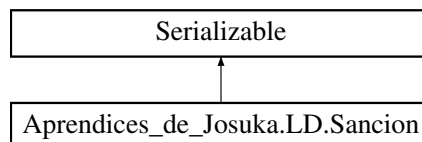
La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/ServiciosExternos/RestClient.java](#)

6.54. Referencia de la Clase Aprendices_de_Josuka.LD.Sancion

Esta clase define objetos de tipo sancion que no están guardados en la base de datos y se obtienen de la raspberry.

Diagrama de herencias de Aprendices_de_Josuka.LD.Sancion



Métodos públicos

- [Sancion](#) (String codigo, [Tipo_Sancion](#) tipo, String persona, String cod_partido)

6.54.1. Descripción detallada

Esta clase define objetos de tipo sancion que no están guardados en la base de datos y se obtienen de la raspberry.

Autor

Alumno

Definición en la línea 19 del archivo Sancion.java.

6.54.2. Documentación del constructor y destructor

6.54.2.1. Sancion()

```
Aprendices_de_Josuka.LD.Sancion.Sancion (
    String codigo,
    Tipo_Sancion tipo,
    String persona,
    String cod_partido )
```

Definición en la línea 28 del archivo Sancion.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Aprendices_de_Josuka/LD/[Sancion.java](#)

6.55. Referencia de la Clase python.Partido.Sancion.Sancion

Métodos públicos

- def [__init__](#) (self, str code, str tipo, str dni, str cod_partido)
- str [get_code](#) (self)
- str [get_tipo](#) (self)
- str [get_dni](#) (self)
- str [get_cod_partido](#) (self)
- None [print](#) (self)

6.55.1. Descripción detallada

Definición en la línea 5 del archivo Sancion.py.

6.55.2. Documentación del constructor y destructor

6.55.2.1. __init__()

```
def python.Partido.Sancion.Sancion.__init__ (
    self,
    str code,
    str tipo,
    str dni,
    str cod_partido )
```

Definición en la línea 8 del archivo Sancion.py.

6.55.3. Documentación de las funciones miembro

6.55.3.1. `get_cod_partido()`

```
str python.Partido.Sancion.Sancion.get_cod_partido (
    self )
```

Definición en la línea 26 del archivo Sancion.py.

6.55.3.2. `get_code()`

```
str python.Partido.Sancion.Sancion.get_code (
    self )
```

Definición en la línea 16 del archivo Sancion.py.

6.55.3.3. `get_dni()`

```
str python.Partido.Sancion.Sancion.get_dni (
    self )
```

Definición en la línea 23 del archivo Sancion.py.

6.55.3.4. `get_tipo()`

```
str python.Partido.Sancion.Sancion.get_tipo (
    self )
```

Definición en la línea 20 del archivo Sancion.py.

6.55.3.5. `print()`

```
None python.Partido.Sancion.Sancion.print (
    self )
```

Definición en la línea 30 del archivo Sancion.py.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/python/Partido/Sancion.py`

6.56. Referencia de la Clase ServiciosExternos.Sancion_JSON

Clase para parsear las sanciones de la raspberry.

Métodos públicos

- [Sancion_JSON](#) (Object json_encoded)
- String [getCode](#) ()
- String [getTipo](#) ()
- String [getDNI](#) ()
- String [getCodPartido](#) ()

6.56.1. Descripción detallada

Clase para parsear las sanciones de la raspberry.

Autor

Alumno

Definición en la línea 9 del archivo Sancion_JSON.java.

6.56.2. Documentación del constructor y destructor

6.56.2.1. Sancion_JSON()

```
ServiciosExternos.Sancion_JSON.Sancion_JSON (
    Object json_encoded )
```

Constructor de la clase

Parámetros

<i>json_encoded</i>	objeto a parsear
---------------------	------------------

Definición en la línea 16 del archivo Sancion_JSON.java.

6.56.3. Documentación de las funciones miembro

6.56.3.1. `getCode()`

```
String ServiciosExternos.Sancion_JSON.getCode ( )
```

Método para parsear el código de la sancion

Devuelve

el código de la sancion

Definición en la línea 24 del archivo Sancion_JSON.java.

6.56.3.2. `getCodPartido()`

```
String ServiciosExternos.Sancion_JSON.getCodPartido ( )
```

Método para parsear el código del partido donde se tuvo lugar la sanción

Devuelve

el código del partido

Definición en la línea 48 del archivo Sancion_JSON.java.

6.56.3.3. `getDNI()`

```
String ServiciosExternos.Sancion_JSON.getDNI ( )
```

Parsea el DNI de la persona que cometió la sanción, puede ser tanto entrenador como jugador

Devuelve

el dni de la persona

Definición en la línea 40 del archivo Sancion_JSON.java.

6.56.3.4. `getTipo()`

```
String ServiciosExternos.Sancion_JSON.getTipo ( )
```

Método para parsear el tipo de la sanción, puede ser FALTA, AGRESION o INSULTO

Devuelve

tipo de la sanción

Definición en la línea 32 del archivo Sancion_JSON.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- `src/main/java/ServiciosExternos/Sancion_JSON.java`

6.57. Referencia de la Clase ServiciosExternos.Sanciones_parameters

Clase que contiene los atributos de las sanciones, con sus getters y setters mediante el plugin de Lombok. Implementa [DF_Printable](#) para imprimir el contenido de cada sancion.

Diagrama de herencias de ServiciosExternos.Sanciones_parameters



Métodos públicos

- void [print](#) ()

6.57.1. Descripción detallada

Clase que contiene los atributos de las sanciones, con sus getters y setters mediante el plugin de Lombok. Implementa [DF_Printable](#) para imprimir el contenido de cada sancion.

Autor

Alumno

Definición en la línea 10 del archivo Sanciones_parameters.java.

6.57.2. Documentación de las funciones miembro

6.57.2.1. print()

```
void ServiciosExternos.Sanciones_parameters.print ( )
```

Imprime el contenido de las sanciones de la raspberry

Reimplementado de [ServiciosExternos.DF_Printable](#).

Definición en la línea 20 del archivo Sanciones_parameters.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/ServiciosExternos/[Sanciones_parameters.java](#)

6.58. Referencia de la Clase Remote.ServiceLocator

Esta clase establece la conexión entre la fachada y controller.

Métodos públicos

- void `setService` () throws RemoteException
- `itfFachada` `getService` ()

6.58.1. Descripción detallada

Esta clase establece la conexión entre la fachada y controller.

Autor

Alumno

Definición en la línea 17 del archivo ServiceLocator.java.

6.58.2. Documentación de las funciones miembro

6.58.2.1. `getService()`

```
itfFachada Remote.ServiceLocator.getService ( )
```

Devuelve la referencia del servicio externo, es decir, de la fachada

Devuelve

la referencia de la fachada

Definición en la línea 49 del archivo ServiceLocator.java.

6.58.2.2. `setService()`

```
void Remote.ServiceLocator.setService ( ) throws RemoteException
```

Establece la conexión con el servidor mediante el Set Service

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 23 del archivo ServiceLocator.java.

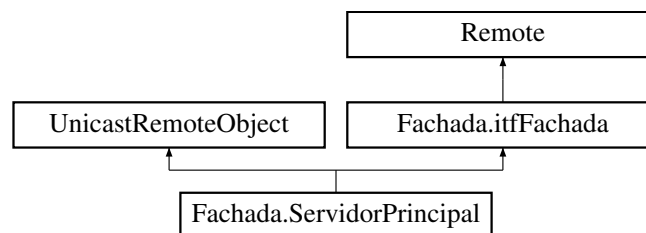
La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Remote/ServiceLocator.java

6.59. Referencia de la Clase Fachada.ServidorPrincipal

En esta clase se ejecutarán o delegarán todas las funciones de la aplicación.

Diagrama de herencias de Fachada.ServidorPrincipal



Métodos públicos

- boolean [EntrarJugador](#) (String email, String psw) throws RemoteException
- boolean [EntrarEntrenador](#) (String email, String psw) throws RemoteException
- boolean [EntrarAdministrador](#) (String email, String psw) throws RemoteException
- boolean [RegistrarJugador](#) (String nombre, String apellido, String fecha_S, String DNI, int telefono, String correo, String psw) throws RemoteException
- boolean [RegistrarEntrenador](#) (String nombre, String apellido, String fecha_S, String DNI, int telefono, String correo, String psw) throws RemoteException
- boolean [modificarCorreo](#) ([Jugador](#) j, String correo, String psw) throws RemoteException
- boolean [modificarCorreoEntrenador](#) ([Entrenador](#) j, String correo, String psw) throws RemoteException
- boolean [RegistrarAdmin](#) (String correo, String psw) throws RemoteException
- List< [Partido](#) > [getPartidos](#) () throws ParseException, RemoteException
- boolean [RegistrarEquipo2](#) (String nombre, [Categoria](#) cat, [Entrenador](#) entrenador, List< [Jugador](#) > lista_↵ Jugadores, HashMap< [Material](#), Integer > inventario) throws RemoteException
- boolean [RegistrarInventario](#) ([Tipo_Material](#) tipo, int cantidad, long precio) throws RemoteException
- void [AsignarInventario](#) ([Material](#) m) throws RemoteException
- List< [Jugador](#) > [getJugador](#) () throws RemoteException
- int [partidosGanados](#) (String DNI) throws RemoteException, ParseException
- List< [Partido](#) > [jugadorPartido](#) (List< [Partido](#) > partidos, String DNI) throws RemoteException, Parse↵ Exception
- List< [Sancion](#) > [sancionesPersona](#) (String DNI) throws RemoteException, ParseException
- List< [Jugador](#) > [misJugadores](#) (String ent) throws RemoteException
- [Jugador](#) [getJug](#) (String correo, String psw) throws RemoteException
- [Entrenador](#) [getEnt](#) (String correo, String psw) throws RemoteException
- List< [Jugador](#) > [MostrarJugadores](#) ([Categoria](#) c) throws RemoteException
- [Entrenador](#) [equipoEntrenador](#) ([Equipo](#) e) throws RemoteException
- List< [Jugador](#) > [equipoJugadores](#) ([Equipo](#) e) throws RemoteException
- int [getEdad](#) (String date) throws RemoteException
- [Partido](#) [sancionPartido](#) (String codPartido) throws ParseException, RemoteException
- int [getRanking](#) (String dni) throws ParseException, RemoteException

- void `ActualizarEquipoEntrenador` (`Entrenador` ent) throws `RemoteException`
- List< `Entrenador` > `getEntrenador` () throws `RemoteException`
- List< `Equipo` > `getEquipos` () throws `RemoteException`
- List< `Equipo` > `getEquiposFilter` (`Categoria` cat) throws `RemoteException`
- List< `Material` > `getMaterial` () throws `RemoteException`
- boolean `ActualizarJugador` (`Jugador` e, boolean reconocimiento, boolean lesionado, boolean cuota) throws `RemoteException`
- boolean `ActualizarEntrenador` (`Entrenador` e, long salario) throws `RemoteException`
- void `ActualizarJugadorEquipo` (List< `Jugador` > lista_Jugadores) throws `RemoteException`
- void `ActualizarEquipo` (`Equipo` equipo, HashMap< `Material`, Integer > inventario, List< `Jugador` > jugadores) throws `RemoteException`
- void `ActualizarMaterial` (`Tipo_Material` tipo, int cantidad, long precio) throws `RemoteException`
- String `ObtenerEquipoEntrenador` (String entrenador) throws `RemoteException`
- String `ObtenerEquipoJugador` (`Jugador` jugador) throws `RemoteException`
- List< `Equipos_Ext` > `clasificacion` (`Categoria` cat) throws `ParseException`

Métodos públicos estáticos

- static `ServidorPrincipal` `getInstance` () throws `RemoteException`
- static void `main` (String[] args)

6.59.1. Descripción detallada

En esta clase se ejecutarán o delegarán todas las funciones de la aplicación.

Autor

Alumno

Definición en la línea 41 del archivo `ServidorPrincipal.java`.

6.59.2. Documentación de las funciones miembro

6.59.2.1. ActualizarEntrenador()

```
boolean Fachada.ServidorPrincipal.ActualizarEntrenador (
    Entrenador e,
    long salario ) throws RemoteException
```

Mediante este método se actualizará el salario de un entrenador

Parámetros

<i>e</i>	entrenador al que se le actualizarán los datos
<i>salario</i>	nuevo salario

Devuelve

devuelve true si se ha modificado correctamente

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 461 del archivo ServidorPrincipal.java.

6.59.2.2. ActualizarEquipo()

```
void Fachada.ServidorPrincipal.ActualizarEquipo (
    Equipo equipo,
    HashMap< Material, Integer > inventario,
    List< Jugador > jugadores ) throws RemoteException
```

Método mediante la cual se actualizan los datos de un equipo

Parámetros

<i>equipo</i>	al cual se le modificarán los datos
<i>inventario</i>	nuevo inventario
<i>jugadores</i>	nuevos jugadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 469 del archivo ServidorPrincipal.java.

6.59.2.3. ActualizarEquipoEntrenador()

```
void Fachada.ServidorPrincipal.ActualizarEquipoEntrenador (
    Entrenador ent ) throws RemoteException
```

Mediante este método se actualizan los entrenadores, estableciendo así su asignación

Parámetros

<i>ent</i>	
------------	--

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 438 del archivo ServidorPrincipal.java.

6.59.2.4. ActualizarJugador()

```
boolean Fachada.ServidorPrincipal.ActualizarJugador (
    Jugador e,
    boolean reconocimiento,
    boolean lesionado,
    boolean cuota ) throws RemoteException
```

Método para actualizar el estado del jugador, tanto médico como de su cuota

Parámetros

<i>e</i>	jugador al que se le actualizarán los datos
<i>reconocimiento</i>	si ha superado el reconocimiento médico o no
<i>lesionado</i>	si esta lesionado o no
<i>cuota</i>	si ha pagado la cuota o no

Devuelve

devuelve true si se ha modificado correctamente

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 457 del archivo ServidorPrincipal.java.

6.59.2.5. ActualizarJugadorEquipo()

```
void Fachada.ServidorPrincipal.ActualizarJugadorEquipo (
    List< Jugador > lista_Jugadores ) throws RemoteException
```

Cuando un equipo se crea y se añaden jugadores a este, se ejecuta este metodo para actualizar el estado de los jugadores a ASIGNADO

Parámetros

<i>lista_Jugadores</i>	jugadores a los que se les actualizará el estado
------------------------	--------------------------------------------------

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 465 del archivo ServidorPrincipal.java.

6.59.2.6. ActualizarMaterial()

```
void Fachada.ServidorPrincipal.ActualizarMaterial (
    Tipo_Material tipo,
    int cantidad,
    long precio ) throws RemoteException
```

Este método sirve para modificar el material

Parámetros

<i>tipo</i>	de material
<i>cantidad</i>	
<i>precio</i>	

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 483 del archivo ServidorPrincipal.java.

6.59.2.7. AsignarInventario()

```
void Fachada.ServidorPrincipal.AsignarInventario (
    Material m ) throws RemoteException
```

Cuando se asigna un material a un equipo, hay que reducir el inventario del club. Es lo que se hace en este método mediante la conexión a la base de datos

Parámetros

<i>m</i>	material a modificar
----------	----------------------

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 184 del archivo ServidorPrincipal.java.

6.59.2.8. clasificacion()

```
List<Equipos_Ext> Fachada.ServidorPrincipal.clasificacion (
    Categoria cat ) throws ParseException
```

Devuelve una lista ordenada de los equipos de la liga según la categoría para hacer clasificación

Parámetros

<i>cat</i>	categoría
------------	-----------

Devuelve

lista ordenada

Excepciones

<i>RemoteException</i>	
<i>org.json.simple.parser.ParseException</i>	

Implementa [Fachada.itfFachada](#).

Definición en la línea 504 del archivo ServidorPrincipal.java.

6.59.2.9. EntrarAdministrador()

```
boolean Fachada.ServidorPrincipal.EntrarAdministrador (
    String email,
    String psw ) throws RemoteException
```

Verifica si el administrador existe en la base de datos para permitir el acceso a la aplicación

Parámetros

<i>email</i>	
<i>psw</i>	

Devuelve

true si existe

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 112 del archivo ServidorPrincipal.java.

6.59.2.10. EntrarEntrenador()

```
boolean Fachada.ServidorPrincipal.EntrarEntrenador (
    String email,
    String psw ) throws RemoteException
```

Verifica si el entrenador existe en la base de datos para permitir el acceso a la aplicación

Parámetros

<i>email</i>	
<i>psw</i>	

Devuelve

true si existe

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 106 del archivo ServidorPrincipal.java.

6.59.2.11. EntrarJugador()

```
boolean Fachada.ServidorPrincipal.EntrarJugador (
    String email,
    String psw ) throws RemoteException
```

Verifica si el jugador existe en la base de datos para permitir el acceso a la aplicación

Parámetros

<i>email</i>	
<i>psw</i>	

Devuelve

true si existe

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 98 del archivo ServidorPrincipal.java.

6.59.2.12. equipoEntrenador()

```
Entrenador Fachada.ServidorPrincipal.equipoEntrenador (
    Equipo e ) throws RemoteException
```

Mediante este método se obtiene el entrenador asignado a un equipo concreto

Parámetros

<i>e</i>	el equipo en cuestión
----------	-----------------------

Devuelve

devuelve el entrenador

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 346 del archivo ServidorPrincipal.java.

6.59.2.13. equipoJugadores()

```
List<Jugador> Fachada.ServidorPrincipal.equipoJugadores (
    Equipo e ) throws RemoteException
```

Este método sirve para obtener los DNIs de los jugadores asignados a un equipo

Parámetros

e	el equipo
---	-----------

Devuelve

devuelve la lista de jugadores

Excepciones

RemoteException	
-----------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 358 del archivo ServidorPrincipal.java.

6.59.2.14. getEdad()

```
int Fachada.ServidorPrincipal.getEdad (
    String date ) throws RemoteException
```

Calcula la edad en base a una fecha, para así clasificar los jugadores en categorías

Parámetros

date	fecha de nacimiento
------	---------------------

Devuelve

devuelve años

Excepciones

RemoteException	
-----------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 374 del archivo ServidorPrincipal.java.

6.59.2.15. getEnt()

```
Entrenador Fachada.ServidorPrincipal.getEnt (
    String correo,
    String psw ) throws RemoteException
```

Devuelve todos los datos de un entrenador segun correo y contraseña

Parámetros

<i>correo</i>	
<i>psw</i>	

Devuelve

entrenador

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 305 del archivo ServidorPrincipal.java.

6.59.2.16. getEntrenador()

```
List<Entrenador> Fachada.ServidorPrincipal.getEntrenador ( ) throws RemoteException
```

Método para obtener todos los entrenadores almacenados en la base de datos

Devuelve

devuelve lista de entrenadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 442 del archivo ServidorPrincipal.java.

6.59.2.17. getEquipos()

```
List<Equipo> Fachada.ServidorPrincipal.getEquipos ( ) throws RemoteException
```

Mediante este método se obtienen todos los equipos de la base de datos que pertenecen al club

Devuelve

lista de equipos

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 446 del archivo ServidorPrincipal.java.

6.59.2.18. getEquiposFilter()

```
List<Equipo> Fachada.ServidorPrincipal.getEquiposFilter (
    Categoria cat ) throws RemoteException
```

Es un método mediante la cual se filtran los equipos dependiendo de la categoría

Parámetros

<i>cat</i>	categoría por la cual se filtrará
------------	-----------------------------------

Devuelve

lista filtrada

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 449 del archivo ServidorPrincipal.java.

6.59.2.19. getInstance()

```
static ServidorPrincipal Fachada.ServidorPrincipal.getInstance ( ) throws RemoteException
[static]
```

Método en la cual se implementa el patrón singleton. Así, solo se inicializará el servidor una única vez

Devuelve

la instancia del servidor

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 56 del archivo ServidorPrincipal.java.

6.59.2.20. getJug()

```
Jugador Fachada.ServidorPrincipal.getJug (
    String correo,
    String psw ) throws RemoteException
```

Obtiene todos los datos del jugador segun su correo y contraseña

Parámetros

<i>correo</i>	
<i>psw</i>	

Devuelve

jugador

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 289 del archivo ServidorPrincipal.java.

6.59.2.21. getJugador()

```
List<Jugador> Fachada.ServidorPrincipal.getJugador ( ) throws RemoteException
```

Devuelve la lista de todos los jugadores de la base de datos

Devuelve

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 188 del archivo ServidorPrincipal.java.

6.59.2.22. getMaterial()

```
List<Material> Fachada.ServidorPrincipal.getMaterial ( ) throws RemoteException
```

Obtiene la lista de materiales desde la base de datos

Devuelve

devuelve la lista

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 453 del archivo ServidorPrincipal.java.

6.59.2.23. getPartidos()

```
List<Partido> Fachada.ServidorPrincipal.getPartidos ( ) throws ParseException, RemoteException
```

Coge todos los partidos de la raspberry

Devuelve

lista de partidos

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Implementa [Fachada.itfFachada](#).

Definición en la línea 138 del archivo ServidorPrincipal.java.

6.59.2.24. getRanking()

```
int Fachada.ServidorPrincipal.getRanking (
    String dni ) throws ParseException, RemoteException
```

Método para calcular el puesto en el que se encuentra un equipo

Parámetros

<i>dni</i>	del jugador/entrenador del cual se quiere conocer el ranking
------------	--------------------------------------------------------------

Devuelve

devuelve el número del puesto

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	

Implementa [Fachada.itfFachada](#).

Definición en la línea 404 del archivo ServidorPrincipal.java.

6.59.2.25. jugadorPartido()

```
List<Partido> Fachada.ServidorPrincipal.jugadorPartido (
    List< Partido > partidos,
    String DNI ) throws RemoteException, ParseException
```

Con este método se obtienen los partidos de un jugador

Parámetros

<i>partidos</i>	lista de todos los partidos
<i>DNI</i>	del jugador del cual se quieren conocer sus partidos

Devuelve

devuelve lista de partidos

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Implementa [Fachada.itfFachada](#).

Definición en la línea 230 del archivo ServidorPrincipal.java.

6.59.2.26. main()

```
static void Fachada.ServidorPrincipal.main (
    String[] args ) [static]
```

Pone el servidor en marcha

Parámetros

<i>args</i>	
-------------	--

Definición en la línea 68 del archivo ServidorPrincipal.java.

6.59.2.27. misJugadores()

```
List<Jugador> Fachada.ServidorPrincipal.misJugadores (
    String ent ) throws RemoteException
```

Método para obtener los jugadores relacionados con un entrenador

Parámetros

<i>ent</i>	DNI del entrenador
------------	--------------------

Devuelve

lista de jugadores

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 268 del archivo ServidorPrincipal.java.

6.59.2.28. modificarCorreo()

```
boolean Fachada.ServidorPrincipal.modificarCorreo (
    Jugador j,
```

```
String correo,  
String psw ) throws RemoteException
```

Modifica el correo y la contraseña de un jugador

Parámetros

<i>j</i>	jugador del cual los datos se modificaran
<i>correo</i>	
<i>psw</i>	

Devuelve

true si se ha realizado la modificación con éxito

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 128 del archivo ServidorPrincipal.java.

6.59.2.29. modificarCorreoEntrenador()

```
boolean Fachada.ServidorPrincipal.modificarCorreoEntrenador (  
    Entrenador j,  
    String correo,  
    String psw ) throws RemoteException
```

Mediante este método se modifican el correo y la contraseña del entrenador

Parámetros

<i>j</i>	el objeto entrenador a modificar
<i>correo</i>	
<i>psw</i>	

Devuelve

devuelve un true si la modificación se realiza con éxito

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 131 del archivo ServidorPrincipal.java.

6.59.2.30. MostrarJugadores()

```
List<Jugador> Fachada.ServidorPrincipal.MostrarJugadores (
    Categoria c ) throws RemoteException
```

Este método sirve para mostrar los jugadores segun categoria

Parámetros

c	categoria
---	-----------

Devuelve

lista de jugadores

Excepciones

RemoteException	
-----------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 321 del archivo ServidorPrincipal.java.

6.59.2.31. ObtenerEquipoEntrenador()

```
String Fachada.ServidorPrincipal.ObtenerEquipoEntrenador (
    String entrenador ) throws RemoteException
```

Mediante este método se obtiene el equipo asignado al entrenador

Parámetros

entrenador	dni del entrenador
------------	--------------------

Devuelve

devuelve el nombre del equipo

Excepciones

RemoteException	
-----------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 487 del archivo ServidorPrincipal.java.

6.59.2.32. **ObtenerEquipoJugador()**

```
String Fachada.ServidorPrincipal.ObtenerEquipoJugador (
    Jugador jugador ) throws RemoteException
```

Obtiene, mediante este metodo, el equipo en el que juega un jugador

Parámetros

<i>jugador</i>	el jugador en cuestión
----------------	------------------------

Devuelve

devuelve el nombre del equipo

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 494 del archivo ServidorPrincipal.java.

6.59.2.33. **partidosGanados()**

```
int Fachada.ServidorPrincipal.partidosGanados (
    String DNI ) throws RemoteException, ParseException
```

Calcula el número total de partidos ganados del equipo de un jugador o entrenador

Parámetros

<i>DNI</i>	del jugador o del entrenador
------------	------------------------------

Devuelve

suma de todos los partidos ganados

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Implementa [Fachada.itfFachada](#).

Definición en la línea 192 del archivo ServidorPrincipal.java.

6.59.2.34. RegistrarAdmin()

```
boolean Fachada.ServidorPrincipal.RegistrarAdmin (
    String correo,
    String psw ) throws RemoteException
```

Registra un administrador en la base de datos

Parámetros

<i>correo</i>	
<i>psw</i>	

Devuelve

true si se registra bien

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 135 del archivo ServidorPrincipal.java.

6.59.2.35. RegistrarEntrenador()

```
boolean Fachada.ServidorPrincipal.RegistrarEntrenador (
    String nombre,
    String apellido,
    String fecha_S,
    String DNI,
    int telefono,
    String correo,
    String psw ) throws RemoteException
```

Registra un entrenador

Parámetros

<i>nombre</i>	
<i>apellido</i>	
<i>fecha↔</i> <i>_S</i>	
<i>DNI</i>	
<i>telefono</i>	
<i>correo</i>	
<i>psw</i>	

Devuelve

true si se registra bien

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 123 del archivo ServidorPrincipal.java.

6.59.2.36. RegistrarEquipo2()

```
boolean Fachada.ServidorPrincipal.RegistrarEquipo2 (
    String nombre,
    Categoria cat,
    Entrenador entrenador,
    List< Jugador > lista_Jugadores,
    HashMap< Material, Integer > inventario ) throws RemoteException
```

Registra un nuevo equipo en la base de datos

Parámetros

<i>nombre</i>	
<i>cat</i>	
<i>entrenador</i>	
<i>lista_Jugadores</i>	
<i>inventario</i>	

Devuelve

true si se registra bien

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 143 del archivo ServidorPrincipal.java.

6.59.2.37. RegistrarInventario()

```
boolean Fachada.ServidorPrincipal.RegistrarInventario (
    Tipo_Material mat,
    int cantidad,
    long precio ) throws RemoteException
```

Registro de un nuevo inventario

Parámetros

<i>mat</i>	tipo material
<i>cantidad</i>	
<i>precio</i>	

Devuelve

true si se registra bien

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 168 del archivo ServidorPrincipal.java.

6.59.2.38. RegistrarJugador()

```
boolean Fachada.ServidorPrincipal.RegistrarJugador (
    String nombre,
    String apellido,
    String fecha_S,
    String DNI,
    int telefono,
    String correo,
    String psw ) throws RemoteException
```

Crea un nuevo jugador en la base de datos

Parámetros

<i>nombre</i>	
<i>apellido</i>	
<i>fecha↔ _S</i>	
<i>DNI</i>	
<i>telefono</i>	
<i>correo</i>	
<i>psw</i>	

Devuelve

true si se crea correctamente

Excepciones

<i>RemoteException</i>	
------------------------	--

Implementa [Fachada.itfFachada](#).

Definición en la línea 118 del archivo ServidorPrincipal.java.

6.59.2.39. sancionesPersona()

```
List<Sancion> Fachada.ServidorPrincipal.sancionesPersona (  
    String DNI ) throws RemoteException, ParseException
```

Mediante este método se obtiene una lista de todas las sanciones de una persona

Parámetros

<i>DNI</i>	del jugador o entrenador
------------	--------------------------

Devuelve

lista de sanciones

Excepciones

<i>RemoteException</i>	
<i>ParseException</i>	

Implementa [Fachada.itfFachada](#).

Definición en la línea 254 del archivo ServidorPrincipal.java.

6.59.2.40. sancionPartido()

```
Partido Fachada.ServidorPrincipal.sancionPartido (
    String codPartido ) throws ParseException, RemoteException
```

Para obtener los datos del partido en el cual se cometieron una sanción

Parámetros

<i>codPartido</i>	
-------------------	--

Devuelve

partido

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	

Implementa [Fachada.itfFachada](#).

Definición en la línea 390 del archivo ServidorPrincipal.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/main/java/Fachada/[ServidorPrincipal.java](#)

6.60. Referencia de la Clase Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO

Clase donde se hara el testeo de la base de datos.

Métodos públicos

- void [inicializar](#) ()
- void [a_escribirJugadores](#) () throws RemoteException, NotBoundException
- void [a_escribirMaterial](#) () throws RemoteException
- void [a_escribirAdmin](#) () throws RemoteException, NotBoundException
- void [a_escribirEntrenador](#) () throws RemoteException, NotBoundException
- void [leerJugador](#) () throws RemoteException
- void [leerAdmin](#) () throws RemoteException
- void [leerEntrenador](#) () throws RemoteException
- void [leerMaterial](#) () throws RemoteException
- void [modificarEntrenador](#) () throws RemoteException, AlreadyBoundException, NotBoundException
- void [modificarJugador](#) () throws RemoteException, AlreadyBoundException, NotBoundException
- void [v_eliminar](#) () throws RemoteException

6.60.1. Descripción detallada

Clase donde se hara el testeo de la base de datos.

Autor

Alumno

Definición en la línea 43 del archivo testDAO.java.

6.60.2. Documentación de las funciones miembro

6.60.2.1. a_escribirAdmin()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.a_escribirAdmin ( ) throws RemoteException, NotBoundException
```

Registra un administrador

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 96 del archivo testDAO.java.

6.60.2.2. a_escribirEntrenador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.a_escribirEntrenador ( ) throws RemoteException, NotBoundException
```

Registra un entrenador

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 110 del archivo testDAO.java.

6.60.2.3. a_escribirJugadores()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.a_escribirJugadores ( ) throws RemoteException, NotBoundException
```

Registra un jugador

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 72 del archivo testDAO.java.

6.60.2.4. a_escribirMaterial()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.a_escribirMaterial ( ) throws RemoteException
```

Registra una compra de material

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 85 del archivo testDAO.java.

6.60.2.5. inicializar()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.inicializar ( )
```

El primer método que se va a ejecutar, inicializa los objetos

Definición en la línea 53 del archivo testDAO.java.

6.60.2.6. leerAdmin()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.leerAdmin ( ) throws RemoteException
```

Lee los administradores

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 143 del archivo testDAO.java.

6.60.2.7. leerEntrenador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.leerEntrenador ( ) throws Remote←  
Exception
```

Lee los entrenadores

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 163 del archivo testDAO.java.

6.60.2.8. leerJugador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.leerJugador ( ) throws RemoteException
```

Lee los jugadores

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 123 del archivo testDAO.java.

6.60.2.9. leerMaterial()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.leerMaterial ( ) throws RemoteException
```

Lee el inventario

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 184 del archivo testDAO.java.

6.60.2.10. modificarEntrenador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.modificarEntrenador ( ) throws RemoteException, AlreadyBoundException, NotBoundException
```

Modifica el salario de un entrenador

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 204 del archivo testDAO.java.

6.60.2.11. modificarJugador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.modificarJugador ( ) throws RemoteException, AlreadyBoundException, NotBoundException
```

Modifica el estado de un jugador

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 228 del archivo testDAO.java.

6.60.2.12. v_eliminar()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO.v_eliminar ( ) throws RemoteException
```

Elimina todos los objetos creados

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 250 del archivo testDAO.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/Aprendices_de_Josuka/Aprendices_de_Josuka/testDAO.java

6.61. Referencia de la Clase

Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada

Clase donde se hara el testeo de la fachada (gestor)

Métodos públicos

- void [setup](#) ()
- void [partidos](#) () throws ParseException, RemoteException, NotBoundException
- void [material](#) () throws RemoteException, NotBoundException
- void [sanciones](#) () throws ParseException, RemoteException, NotBoundException
- void [a_jugador](#) () throws RemoteException, AlreadyBoundException, NotBoundException
- void [entrarJugador](#) () throws RemoteException, AlreadyBoundException, NotBoundException
- void [v_eliminar](#) () throws RemoteException
- void [getEdad](#) () throws RemoteException, NotBoundException, AlreadyBoundException
- void [a_admin](#) () throws RemoteException, NotBoundException
- void [entrarAdministrador](#) () throws RemoteException, AlreadyBoundException, NotBoundException
- void [v_eliminarAdmin](#) () throws RemoteException
- void [v_eliminarMaterial](#) () throws RemoteException
- void [a_entrenador](#) () throws RemoteException, AlreadyBoundException, NotBoundException
- void [entrarEntrenador](#) () throws RemoteException, AlreadyBoundException, NotBoundException
- void [v_eliminarEntrenador](#) () throws RemoteException
- void [v_actualizarEntrenador](#) () throws RemoteException, NotBoundException
- void [v_actualizarJugador](#) () throws RemoteException, NotBoundException

Métodos públicos estáticos

- static void [setupClass](#) () throws RemoteException, AlreadyBoundException

6.61.1. Descripción detallada

Clase donde se hara el testeo de la fachada (gestor)

Autor

Alumno

Definición en la línea 43 del archivo testFachada.java.

6.61.2. Documentación de las funciones miembro

6.61.2.1. a_admin()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.a_admin ( ) throws RemoteException,  
NotBoundException
```

Registra un administrador

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 181 del archivo testFachada.java.

6.61.2.2. a_entrenador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.a_entrenador ( ) throws Remote↵  
Exception, AlreadyBoundException, NotBoundException
```

Registra un nuevo entrenador

Excepciones

<i>RemoteException</i>	
<i>AlreadyBoundException</i>	
<i>NotBoundException</i>	

Definición en la línea 226 del archivo testFachada.java.

6.61.2.3. a_jugador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.a_jugador ( ) throws RemoteException,  
AlreadyBoundException, NotBoundException
```

Simula el registro de un jugador

Excepciones

<i>RemoteException</i>	
<i>AlreadyBoundException</i>	
<i>NotBoundException</i>	

Definición en la línea 131 del archivo testFachada.java.

6.61.2.4. entrarAdministrador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.entrarAdministrador ( ) throws  
RemoteException, AlreadyBoundException, NotBoundException
```

Simula la entrada del admin registrado en la aplicación

Excepciones

<i>RemoteException</i>	
<i>AlreadyBoundException</i>	
<i>NotBoundException</i>	

Definición en la línea 193 del archivo testFachada.java.

6.61.2.5. entrarEntrenador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.entrarEntrenador ( ) throws Remote←  
Exception, AlreadyBoundException, NotBoundException
```

El entrenador nuevo entra a la aplicación

Excepciones

<i>RemoteException</i>	
<i>AlreadyBoundException</i>	
<i>NotBoundException</i>	

Definición en la línea 238 del archivo testFachada.java.

6.61.2.6. entrarJugador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.entrarJugador ( ) throws Remote←  
Exception, AlreadyBoundException, NotBoundException
```

Prueba si el jugador creado puede entrar bien a la aplicación

Excepciones

<i>RemoteException</i>	
<i>AlreadyBoundException</i>	
<i>NotBoundException</i>	

Definición en la línea 143 del archivo testFachada.java.

6.61.2.7. getEdad()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.getEdad ( ) throws RemoteException,  
NotBoundException,AlreadyBoundException
```

Comprueba que el método devuelva la edad correctamente

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	
<i>AlreadyBoundException</i>	

Definición en la línea 167 del archivo testFachada.java.

6.61.2.8. material()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.material ( ) throws RemoteException,  
NotBoundException
```

Testea el registro de nuevo material

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 103 del archivo testFachada.java.

6.61.2.9. partidos()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.partidos ( ) throws ParseException,  
RemoteException, NotBoundException
```

Hace pruebas con los partidos del servicio externo

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 86 del archivo testFachada.java.

6.61.2.10. sanciones()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.sanciones ( ) throws ParseException,  
RemoteException, NotBoundException
```

Hace pruebas con las sanciones del servicio externo

Excepciones

<i>ParseException</i>	
<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 117 del archivo testFachada.java.

6.61.2.11. setup()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.setup ( )
```

Inicializa los objetos

Definición en la línea 68 del archivo testFachada.java.

6.61.2.12. setupClass()

```
static void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.setupClass ( ) throws Remote↵  
Exception, AlreadyBoundException [static]
```

Método que permite establecer la conexión con el servidor

Excepciones

<i>RemoteException</i>	
<i>AlreadyBoundException</i>	

Definición en la línea 59 del archivo testFachada.java.

6.61.2.13. v_actualizarEntrenador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.v_actualizarEntrenador ( ) throws  
RemoteException, NotBoundException
```

Se actualizan los datos del entrenador

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 263 del archivo testFachada.java.

6.61.2.14. v_actualizarJugador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.v_actualizarJugador ( ) throws  
RemoteException, NotBoundException
```

Se actualizan los datos del jugador

Excepciones

<i>RemoteException</i>	
<i>NotBoundException</i>	

Definición en la línea 279 del archivo testFachada.java.

6.61.2.15. v_eliminar()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.v_eliminar ( ) throws Remote←  
Exception
```

Elimina el jugador de la base de datos

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 155 del archivo testFachada.java.

6.61.2.16. v_eliminarAdmin()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.v_eliminarAdmin ( ) throws RemoteException
```

Elimina el admin de la base de datos

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 204 del archivo testFachada.java.

6.61.2.17. v_eliminarEntrenador()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.v_eliminarEntrenador ( ) throws RemoteException
```

Se elimina el entrenador de la base de datos

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 251 del archivo testFachada.java.

6.61.2.18. v_eliminarMaterial()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada.v_eliminarMaterial ( ) throws RemoteException
```

Elimina el material creado

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 214 del archivo testFachada.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- src/test/java/Aprendices_de_Josuka/Aprendices_de_Josuka/testFachada.java

6.62. Referencia de la Clase

Aprendices_de_Josuka.Aprendices_de_Josuka.testPartidos

Clase donde se hara el testeo con mockito.

Métodos públicos

- void [setUp](#) ()
- void [test](#) ()

6.62.1. Descripción detallada

Clase donde se hara el testeo con mockito.

Autor

Alumno

Definición en la línea 27 del archivo testPartidos.java.

6.62.2. Documentación de las funciones miembro

6.62.2.1. setUp()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testPartidos.setUp ( )
```

Primer método en ejecutar, inicializa los objetos

Definición en la línea 42 del archivo testPartidos.java.

6.62.2.2. test()

```
void Aprendices_de_Josuka.Aprendices_de_Josuka.testPartidos.test ( )
```

Comprueba si los objetos se han creado bien con Mockito

Definición en la línea 53 del archivo testPartidos.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/test/java/Aprendices_de_Josuka/Aprendices_de_Josuka/testPartidos.java](#)

6.63. Referencia del enum `Aprendices_de_Josuka.LD.Tipo_Material`

Esta clase define un enumerado para el atributo tipo de la clase [Material](#).

Atributos públicos

- [Balones](#)
- [Botellas](#)
- [Petos](#)
- [Conos](#)
- [Picas](#)
- [Barreras](#)
- [Vallas](#)

6.63.1. Descripción detallada

Esta clase define un enumerado para el atributo tipo de la clase [Material](#).

Autor

Alumno

Definición en la línea 9 del archivo `Tipo_Material.java`.

6.63.2. Documentación de los datos miembro

6.63.2.1. Balones

`Aprendices_de_Josuka.LD.Tipo_Material.Balones`

Definición en la línea 11 del archivo `Tipo_Material.java`.

6.63.2.2. Barreras

`Aprendices_de_Josuka.LD.Tipo_Material.Barreras`

Definición en la línea 11 del archivo `Tipo_Material.java`.

6.63.2.3. Botellas

`Aprendices_de_Josuka.LD.Tipo_Material.Botellas`

Definición en la línea 11 del archivo `Tipo_Material.java`.

6.63.2.4. Conos

`Aprendices_de_Josuka.LD.Tipo_Material.Conos`

Definición en la línea 11 del archivo `Tipo_Material.java`.

6.63.2.5. Petos

`Aprendices_de_Josuka.LD.Tipo_Material.Petos`

Definición en la línea 11 del archivo `Tipo_Material.java`.

6.63.2.6. Picas

`Aprendices_de_Josuka.LD.Tipo_Material.Picas`

Definición en la línea 11 del archivo `Tipo_Material.java`.

6.63.2.7. Vallas

`Aprendices_de_Josuka.LD.Tipo_Material.Vallas`

Definición en la línea 12 del archivo `Tipo_Material.java`.

La documentación para este enum ha sido generada a partir del siguiente fichero:

- `src/main/java/Aprendices_de_Josuka/LD/Tipo_Material.java`

6.64. Referencia del enum `Aprendices_de_Josuka.LD.Tipo_Sancion`

Esta clase define un enumerado para el atributo tipo de la clase `Sancion`.

Atributos públicos

- [FALTA](#)
- [AGRESION](#)
- [INSULTO](#)

6.64.1. Descripción detallada

Esta clase define un enumerado para el atributo tipo de la clase [Sancion](#).

Autor

Alumno

Definición en la línea 9 del archivo Tipo_Sancion.java.

6.64.2. Documentación de los datos miembro

6.64.2.1. AGRESION

```
Aprendices_de_Josuka.LD.Tipo_Sancion.AGRESION
```

Definición en la línea 10 del archivo Tipo_Sancion.java.

6.64.2.2. FALTA

```
Aprendices_de_Josuka.LD.Tipo_Sancion.FALTA
```

Definición en la línea 10 del archivo Tipo_Sancion.java.

6.64.2.3. INSULTO

```
Aprendices_de_Josuka.LD.Tipo_Sancion.INSULTO
```

Definición en la línea 11 del archivo Tipo_Sancion.java.

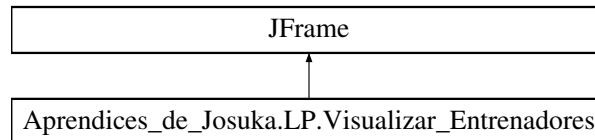
La documentación para este enum ha sido generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LD/Tipo_Sancion.java](#)

6.65. Referencia de la Clase Aprendices_de_Josuka.LP.Visualizar_Entrenadores

En esta clase se desarrolla la GUI para que el administrador pueda visualizar los entrenadores.

Diagrama de herencias de Aprendices_de_Josuka.LP.Visualizar_Entrenadores



Métodos públicos

- `Visualizar_Entrenadores` (`Controller` controller)
- void `initComponents` ()
- void `InsertarJPanel` ()

6.65.1. Descripción detallada

En esta clase se desarrolla la GUI para que el administrador pueda visualizar los entrenadores.

Autor

Alumno

Definición en la línea 33 del archivo `Visualizar_Entrenadores.java`.

6.65.2. Documentación del constructor y destructor

6.65.2.1. Visualizar_Entrenadores()

```
Aprendices_de_Josuka.LP.Visualizar_Entrenadores.Visualizar_Entrenadores (
    Controller controller )
```

Constructor

Parámetros

<code>controller</code>	
-------------------------	--

Definición en la línea 59 del archivo `Visualizar_Entrenadores.java`.

6.65.3. Documentación de las funciones miembro

6.65.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.Visualizar_Entrenadores.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 68 del archivo Visualizar_Entrenadores.java.

6.65.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Visualizar_Entrenadores.InsertarJPanel ( )
```

Se insertan JPanels con la informacion de cada entrenador

Definición en la línea 310 del archivo Visualizar_Entrenadores.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

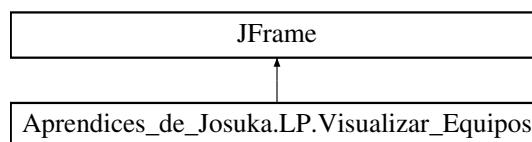
- [src/main/java/Aprendices_de_Josuka/LP/Visualizar_Entrenadores.java](#)

6.66. Referencia de la Clase

Aprendices_de_Josuka.LP.Visualizar_Equipos

En esta clase se desarrolla la GUI para que el administrador pueda visualizar los equipos.

Diagrama de herencias de Aprendices_de_Josuka.LP.Visualizar_Equipos



Métodos públicos

- [Visualizar_Equipos](#) ([Controller](#) controller)
- void [initComponents](#) ()
- void [RellenarCombo](#) ()
- void [MostrarJugadores](#) () throws RemoteException
- void [MostrarInventario](#) ()

6.66.1. Descripción detallada

En esta clase se desarrolla la GUI para que el administrador pueda visualizar los equipos.

Autor

Alumno

Definición en la línea 37 del archivo Visualizar_Equipos.java.

6.66.2. Documentación del constructor y destructor

6.66.2.1. Visualizar_Equipos()

```
Aprendices_de_Josuka.LP.Visualizar_Equipos.Visualizar_Equipos(  
    Controller controller )
```

Constructor

Parámetros

controller	
------------	--

Definición en la línea 70 del archivo Visualizar_Equipos.java.

6.66.3. Documentación de las funciones miembro

6.66.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.Visualizar_Equipos.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 79 del archivo Visualizar_Equipos.java.

6.66.3.2. MostrarInventario()

```
void Aprendices_de_Josuka.LP.Visualizar_Equipos.MostrarInventario ( )
```

Metodo para mostrar el inventario del equipo

Definición en la línea 429 del archivo Visualizar_Equipos.java.

6.66.3.3. MostrarJugadores()

```
void Aprendices_de_Josuka.LP.Visualizar_Equipos.MostrarJugadores ( ) throws RemoteException
```

Metodo para mostrar los jugadores del equipo seleccionado

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 404 del archivo Visualizar_Equipos.java.

6.66.3.4. RellenarCombo()

```
void Aprendices_de_Josuka.LP.Visualizar_Equipos.RellenarCombo ( )
```

Metodo para introducir los equipos en el ComboBox

Definición en la línea 386 del archivo Visualizar_Equipos.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

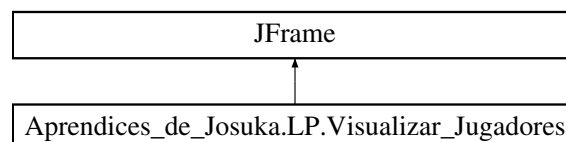
- [src/main/java/Aprendices_de_Josuka/LP/Visualizar_Equipos.java](#)

6.67. Referencia de la Clase

Aprendices_de_Josuka.LP.Visualizar_Jugadores

En esta clase se desarrolla la GUI para que los administradores puedan ver los jugadores.

Diagrama de herencias de Aprendices_de_Josuka.LP.Visualizar_Jugadores



Métodos públicos

- [Visualizar_Jugadores](#) ([Controller](#) controller)
- void [initComponents](#) ()
- void [InsertarJPanel](#) ()

6.67.1. Descripción detallada

En esta clase se desarrolla la GUI para que los administradores puedan ver los jugadores.

Autor

Alumno

Definición en la línea 36 del archivo Visualizar_Jugadores.java.

6.67.2. Documentación del constructor y destructor

6.67.2.1. Visualizar_Jugadores()

```
Aprendices_de_Josuka.LP.Visualizar_Jugadores.Visualizar_Jugadores (
    Controller controller )
```

Constructor

Parámetros

<i>controller</i>	
-------------------	--

Definición en la línea 78 del archivo Visualizar_Jugadores.java.

6.67.3. Documentación de las funciones miembro

6.67.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.Visualizar_Jugadores.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 87 del archivo Visualizar_Jugadores.java.

6.67.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Visualizar_Jugadores.InsertarJPanel ( )
```

Metodo para insertar JPanels con la informacion de cada jugador

Definición en la línea 350 del archivo Visualizar_Jugadores.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

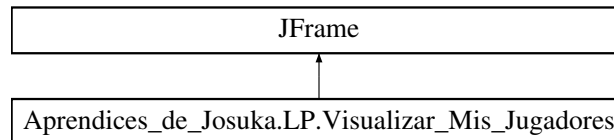
- [src/main/java/Aprendices_de_Josuka/LP/Visualizar_Jugadores.java](#)

6.68. Referencia de la Clase

Aprendices_de_Josuka.LP.Visualizar_Mis_Jugadores

En esta clase se desarrolla la GUI que se encarga de mostrar a cada entrenador sus jugadores.

Diagrama de herencias de Aprendices_de_Josuka.LP.Visualizar_Mis_Jugadores



Métodos públicos

- `Visualizar_Mis_Jugadores` (`Controller` controller, `Entrenador` j) throws `RemoteException`
- void `initComponents` ()
- void `InsertarJPanel` ()

6.68.1. Descripción detallada

En esta clase se desarrolla la GUI que se encarga de mostrar a cada entrenador sus jugadores.

Autor

Alumno

Definición en la línea 32 del archivo `Visualizar_Mis_Jugadores.java`.

6.68.2. Documentación del constructor y destructor

6.68.2.1. Visualizar_Mis_Jugadores()

```

Aprendices_de_Josuka.LP.Visualizar_Mis_Jugadores.Visualizar_Mis_Jugadores (
    Controller controller,
    Entrenador j ) throws RemoteException
  
```

Constructor de la clase

Parámetros

<i>controller</i>	para la conexión RMI con el servidor
<i>j</i>	el entrenador que está utilizando la aplicación

Excepciones

<i>RemoteException</i>	
------------------------	--

Definición en la línea 56 del archivo Visualizar_Mis_Jugadores.java.

6.68.3. Documentación de las funciones miembro

6.68.3.1. initComponents()

```
void Aprendices_de_Josuka.LP.Visualizar_Mis_Jugadores.initComponents ( )
```

Se inicializan los componentes

Definición en la línea 67 del archivo Visualizar_Mis_Jugadores.java.

6.68.3.2. InsertarJPanel()

```
void Aprendices_de_Josuka.LP.Visualizar_Mis_Jugadores.InsertarJPanel ( )
```

Inserta un panel para mostrar todos los jugadores relacionados al entrenador en cuestion

Definición en la línea 260 del archivo Visualizar_Mis_Jugadores.java.

La documentación para esta clase fue generada a partir del siguiente fichero:

- [src/main/java/Aprendices_de_Josuka/LP/Visualizar_Mis_Jugadores.java](#)

Capítulo 7

Documentación de archivos

7.1. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LD/Administrador.java

Clases

- class [Aprendices_de_Josuka.LD.Administrador](#)

Esta clase define objetos de tipo administrador.

Paquetes

- package [Aprendices_de_Josuka.LD](#)

Todos los objetos que tendrá la aplicación.

7.2. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LD/Categoria.java

Clases

- enum [Aprendices_de_Josuka.LD.Categoria](#)

Esta clase define objetos de tipo [Categoria](#) mediante un enumerado.

Paquetes

- package [Aprendices_de_Josuka.LD](#)

Todos los objetos que tendrá la aplicación.

7.3. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LD/Entrenador.java

Clases

- class [Aprendices_de_Josuka.LD.Entrenador](#)

Esta clase define objetos de tipo [Entrenador](#).

Paquetes

- package [Aprendices_de_Josuka.LD](#)

Todos los objetos que tendrá la aplicación.

7.4. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LD/Equipo.java

Clases

- class [Aprendices_de_Josuka.LD.Equipo](#)

Esta clase define objetos de tipo equipo.

Paquetes

- package [Aprendices_de_Josuka.LD](#)

Todos los objetos que tendrá la aplicación.

7.5. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LD/Equipos_Ext.java

Clases

- class [Aprendices_de_Josuka.LD.Equipos_Ext](#)

Esta clase define objetos de tipo equipo que no están guardados en la base de datos y se obtienen de la raspberry.

Paquetes

- package [Aprendices_de_Josuka.LD](#)

Todos los objetos que tendrá la aplicación.

7.6. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LD/Jugador.java

Clases

- class [Aprendices_de_Josuka.LD.Jugador](#)
Esta clase define los objetos de tipo jugador.

Paquetes

- package [Aprendices_de_Josuka.LD](#)
Todos los objetos que tendrá la aplicación.

7.7. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LD/Material.java

Clases

- class [Aprendices_de_Josuka.LD.Material](#)
Esta clase define los objetos de tipo material, es decir, del inventario.

Paquetes

- package [Aprendices_de_Josuka.LD](#)
Todos los objetos que tendrá la aplicación.

7.8. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LD/Partido.java

Clases

- class [Aprendices_de_Josuka.LD.Partido](#)
Esta clase define objetos de tipo partido que no están guardados en la base de datos y se obtienen de la raspberry.

Paquetes

- package [Aprendices_de_Josuka.LD](#)
Todos los objetos que tendrá la aplicación.

7.9. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LD/Persona.java

Clases

- class [Aprendices_de_Josuka.LD.Persona](#)

Esta clase define objetos de tipo persona (clase padre), de la que después heredarán las clases entrenador y jugador (las clases hijas)

Paquetes

- package [Aprendices_de_Josuka.LD](#)

Todos los objetos que tendrá la aplicación.

7.10. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LD/Sancion.java

Clases

- class [Aprendices_de_Josuka.LD.Sancion](#)

Esta clase define objetos de tipo sancion que no están guardados en la base de datos y se obtienen de la raspberry.

Paquetes

- package [Aprendices_de_Josuka.LD](#)

Todos los objetos que tendrá la aplicación.

7.11. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LD/Tipo_Material.java

Clases

- enum [Aprendices_de_Josuka.LD.Tipo_Material](#)

Esta clase define un enumerado para el atributo tipo de la clase [Material](#).

Paquetes

- package [Aprendices_de_Josuka.LD](#)

Todos los objetos que tendrá la aplicación.

7.12. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LD/Tipo_Sancion.java

Clases

- enum [Aprendices_de_Josuka.LD.Tipo_Sancion](#)

Esta clase define un enumerado para el atributo tipo de la clase [Sancion](#).

Paquetes

- package [Aprendices_de_Josuka.LD](#)

Todos los objetos que tendrá la aplicación.

7.13. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Anyadir_Inventario.java

Clases

- class [Aprendices_de_Josuka.LP.Anyadir_Inventario](#)

Esta clase define la GUI de la ventana de anyadir inventario del administrador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.14. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Clasificacion.java

Clases

- class [Aprendices_de_Josuka.LP.Clasificacion](#)

En esta clase se desarrolla la GUI donde se muestra la clasificacion al administrador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.15. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/↵ Clasificacion_Entrenador.java

Clases

- class [Aprendices_de_Josuka.LP.Clasificacion_Entrenador](#)

En esta clase se desarrolla la GUI donde se muestra la clasificacion al entrenador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.16. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/↵ Clasificacion_Jugador.java

Clases

- class [Aprendices_de_Josuka.LP.Clasificacion_Jugador](#)

En esta clase se desarrolla la GUI donde se muestra la clasificacion al jugador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.17. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/EditarDatos.java

Clases

- class [Aprendices_de_Josuka.LP.EditarDatos](#)

En esta clase se desarrolla la GUI para que el jugador pueda modificar su correo o su contraseña.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.18. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/↵ EditarDatosEntrenador.java

Clases

- class [Aprendices_de_Josuka.LP.EditarDatosEntrenador](#)

En esta clase se desarrolla la GUI para que el entrenador pueda modificar su correo o su contraseña.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.19. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/EditarEntrenador.java

Clases

- class [Aprendices_de_Josuka.LP.EditarEntrenador](#)

En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del entrenador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.20. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/EditarEquipo.java

Clases

- class [Aprendices_de_Josuka.LP.EditarEquipo](#)

En esta clase se desarrolla la GUI para que el administrador pueda parte de la informacion del equipo.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.21. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/EditarJugador.java

Clases

- class [Aprendices_de_Josuka.LP.EditarJugador](#)

En esta clase se desarrolla la GUI para que el administrador pueda editar parte de la informacion del jugador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.22. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Estadisticas.java

Clases

- class [Aprendices_de_Josuka.LP.Estadisticas](#)

En esta clase se desarrolla la GUI donde se muestran las estadisticas del jugador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.23. Referencia del Archivo **src/main/java/Aprendices_de_Josuka/LP/↵ EstadisticasEntrenador.java**

Clases

- class [Aprendices_de_Josuka.LP.EstadisticasEntrenador](#)

En esta clase se desarrolla la GUI donde se muestran las estadisticas del entrenador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.24. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/FichaEntrenador.java

Clases

- class [Aprendices_de_Josuka.LP.FichaEntrenador](#)

En esta clase se desarrolla la GUI donde se muestra al entrenador su ficha.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.25. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/FichaJugador.java

Clases

- class [Aprendices_de_Josuka.LP.FichaJugador](#)

En esta clase se desarrolla la GUI donde se muestra al jugador su ficha.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.26. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/IniciarSesion.java

Clases

- class [Aprendices_de_Josuka.LP.IniciarSesion](#)

En esta clase se desarrolla la GUI para el inicio de sesion de los jugadores y entrenadores ya registrados.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.27. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/IniciarSesion_Admin.java

Clases

- class [Aprendices_de_Josuka.LP.IniciarSesion_Admin](#)

En esta clase se desarrolla la GUI para el inicio de sesion del administrador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.28. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Panel_Entrenador.java

Clases

- class [Aprendices_de_Josuka.LP.Panel_Entrenador](#)

En esta clase se crea un panel con la informacion del entrenador, para luego poder introducirlo en otras ventanas.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.29. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Panel_Equipos.java

Clases

- class [Aprendices_de_Josuka.LP.Panel_Equipos](#)

En esta clase se crea un panel para mostrar por pantalla cada equipo, para luego poder introducirlo en otras ventanas.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.30. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Panel_Jugador.java

Clases

- class [Aprendices_de_Josuka.LP.Panel_Jugador](#)

En esta clase se crea un panel para mostrar por pantalla cada jugador, para luego poder introducirlo en otras ventanas.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.31. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Panel_Partidos.java

Clases

- class [Aprendices_de_Josuka.LP.Panel_Partidos](#)

En esta clase se crea un panel para mostrar por pantalla cada partido, para luego poder introducirlo en otras ventanas.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.32. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Panel_Sanciones.java

Clases

- class [Aprendices_de_Josuka.LP.Panel_Sanciones](#)

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.33. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Pantalla_Principal.java

Clases

- class [Aprendices_de_Josuka.LP.Pantalla_Principal](#)

En esta clase se desarrolla la GUI para la pantalla principal de la aplicacion.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.34. Referencia del Archivo **src/main/java/Aprendices_de_Josuka/LP/Principal_Administrador.java**

Clases

- class [Aprendices_de_Josuka.LP.Principal_Administrador](#)

En esta clase se desarrolla la GUI que visualiza el administrador cuando abre la aplicacion despues de haber iniciado sesion.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.35. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Principal_Entrenador.java

Clases

- class [Aprendices_de_Josuka.LP.Principal_Entrenador](#)

En esta clase se desarrolla la GUI que visualiza el entrenador cuando abre la aplicacion despues de haber iniciado sesion.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.36. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/Principal_Jugador.java

Clases

- class [Aprendices_de_Josuka.LP.Principal_Jugador](#)

En esta clase se desarrolla la GUI que visualiza el jugador cuando abre la aplicacion despues de haber iniciado sesion.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.37. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/RegistrarAdmin.java

Clases

- class [Aprendices_de_Josuka.LP.RegistrarAdmin](#)

En esta clase se desarrolla la GUI para el registro del administrador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.38. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/RegistrarEntrenador.java

Clases

- class [Aprendices_de_Josuka.LP.RegistrarEntrenador](#)

En esta clase se desarrolla la GUI para el registro del entrenador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.39. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/RegistrarEquipo.java

Clases

- class [Aprendices_de_Josuka.LP.RegistrarEquipo](#)

En esta clase se desarrolla la GUI para cuando un administrador quiera registrar un equipo.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.40. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/RegistrarJugador.java

Clases

- class [Aprendices_de_Josuka.LP.RegistrarJugador](#)

En esta clase se desarrolla la GUI para el registro del jugador.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.41. Referencia del Archivo

src/main/java/Aprendices_de_Josuka/LP/RegistrarMaterial.java

Clases

- class [Aprendices_de_Josuka.LP.RegistrarMaterial](#)

En esta clase se desarrolla la GUI para que el administrador pueda registrar el material.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.42. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/↵ Visualizar_Entrenadores.java

Clases

- class [Aprendices_de_Josuka.LP.Visualizar_Entrenadores](#)

En esta clase se desarrolla la GUI para que el administrador pueda visualizar los entrenadores.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.43. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/Visualizar_Equipos.java

Clases

- class [Aprendices_de_Josuka.LP.Visualizar_Equipos](#)

En esta clase se desarrolla la GUI para que el administrador pueda visualizar los equipos.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.44. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/↵ Visualizar_Jugadores.java

Clases

- class [Aprendices_de_Josuka.LP.Visualizar_Jugadores](#)

En esta clase se desarrolla la GUI para que los administradores puedan ver los jugadores.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.45. Referencia del Archivo src/main/java/Aprendices_de_Josuka/LP/Visualizar_Mis_Jugadores.java

Clases

- class [Aprendices_de_Josuka.LP.Visualizar_Mis_Jugadores](#)

En esta clase se desarrolla la GUI que se encarga de mostrar a cada entrenador sus jugadores.

Paquetes

- package [Aprendices_de_Josuka.LP](#)

Todas las ventanas de la aplicación, tanto de la parte de administrador como la correspondiente a entrenadores y jugadores.

7.46. Referencia del Archivo src/main/java/Controller/Controller.java

Clases

- class [Controller.Controller](#)

Paquetes

- package [Controller](#)

Necesario para incorporar el patron de diseño del controller.

7.47. Referencia del Archivo src/main/java/DAO/DAO.java

Clases

- class [DAO.DAO](#)

Paquetes

- package [DAO](#)

En este paquete estan incluidas todas las clases que tienen relacion con la base de datos.

7.48. Referencia del Archivo src/main/java/Fachada/itfFachada.java

Clases

- interface [Fachada.itfFachada](#)

interfaz que contiene todos los métodos que implementará el servidor

Paquetes

- package [Fachada](#)

Necesario para incorporar el patrón de diseño de fachada. El servidor de la aplicación.

7.49. Referencia del Archivo src/main/java/Fachada/ServidorPrincipal.java

Clases

- class [Fachada.ServidorPrincipal](#)

En esta clase se ejecutarán o delegarán todas las funciones de la aplicación.

Paquetes

- package [Fachada](#)

Necesario para incorporar el patrón de diseño de fachada. El servidor de la aplicación.

7.50. Referencia del Archivo src/main/java/Gateway/Gateway.java

Clases

- class [Gateway.Gateway](#)

Paquetes

- package [Gateway](#)

La aplicación se conectará al servicio externo de la liga, de la cual obtendrá los partidos y las sanciones.

7.51. Referencia del Archivo src/main/java/Gateway/itfGateway.java

Clases

- class [Gateway.itfGateway](#)

Interfaz del [Gateway](#) donde se establecen todos los métodos que tendrá el [Gateway](#).

Paquetes

- package [Gateway](#)

La aplicación se conectará al servicio externo de la liga, de la cual obtendrá los partidos y las sanciones.

7.52. Referencia del Archivo src/main/java/Remote/ServiceLocator.java

Clases

- class [Remote.ServiceLocator](#)

Esta clase establece la conexión entre la fachada y controller.

Paquetes

- package [Remote](#)

Necesario para incorporar el patrón de diseño de service locator.

7.53. Referencia del Archivo src/main/java/ServiciosExternos/DF_Printable.java

Clases

- class [ServiciosExternos.DF_Printable](#)

Interfaz con método print que será implementado por [Partidos_parameters](#) y [Sanciones_parameters](#) para visualizar el contenido de las listas.

Paquetes

- package [ServiciosExternos](#)

Clases para hacer posible la conexión con la raspberry y el parseo de la información.

7.54. Referencia del Archivo src/main/java/ServiciosExternos/Partido_JSON.java

Clases

- class [ServiciosExternos.Partido_JSON](#)

Clase para parsear los partidos de la raspberry.

Paquetes

- package [ServiciosExternos](#)

Clases para hacer posible la conexión con la raspberry y el parseo de la información.

7.55. Referencia del Archivo src/main/java/ServiciosExternos/Partidos_parameters.java

Clases

- class [ServiciosExternos.Partidos_parameters](#)

Clase que contiene los parametros de los partidos y sus getters y setters.

Paquetes

- package [ServiciosExternos](#)

Clases para hacer posible la conexión con la raspberry y el parseo de la información.

7.56. Referencia del Archivo src/main/java/ServiciosExternos/RestClient.java

Clases

- class [ServiciosExternos.RestClient](#)< T extends [DF_Printable](#) >

Sirve para conectar dos aplicaciones. En este caso la raspberry y la aplicación de Aprendices de Josuka.

Paquetes

- package [ServiciosExternos](#)

Clases para hacer posible la conexión con la raspberry y el parseo de la información.

7.57. Referencia del Archivo src/main/java/ServiciosExternos/Sancion_JSON.java

Clases

- class [ServiciosExternos.Sancion_JSON](#)

Clase para parsear las sanciones de la raspberry.

Paquetes

- package [ServiciosExternos](#)

Clases para hacer posible la conexión con la raspberry y el parseo de la información.

7.58. Referencia del Archivo src/main/java/ServiciosExternos/Sanciones_parameters.java

Clases

- class [ServiciosExternos.Sanciones_parameters](#)

Clase que contiene los atributos de las sanciones, con sus getters y setters mediante el plugin de Lombok. Implementa [DF_Printable](#) para imprimir el contenido de cada sancion.

Paquetes

- package [ServiciosExternos](#)

Clases para hacer posible la conexión con la raspberry y el parseo de la información.

7.59. Referencia del Archivo src/main/python/__init__.py

Namespaces

- [python](#)

7.60. Referencia del Archivo src/main/python/Partido/__init__.py

Namespaces

- [python.Partido](#)

7.61. Referencia del Archivo src/main/python/Partidos/__init__.py

Namespaces

- [python.Partidos](#)

7.62. Referencia del Archivo src/main/python/Partido/Equipo.py

Clases

- class [python.Partido.Equipo.Equipo](#)

Namespaces

- [python.Partido.Equipo](#)

7.63. Referencia del Archivo src/main/python/Partido/Partido.py

Clases

- class [python.Partido.Partido.Partido](#)

Namespaces

- [python.Partido.Partido](#)

7.64. Referencia del Archivo src/main/python/Partido/Sancion.py

Clases

- class [python.Partido.Sancion.Sancion](#)

Namespaces

- [python.Partido.Sancion](#)

7.65. Referencia del Archivo src/main/python/Partidos/ADJPartidos/AprendicesDeJosukaPartidos.py ↩

Clases

- class [AprendicesDeJosukaPartidos.AprendicesDeJosukaPartidos](#)

Namespaces

- [AprendicesDeJosukaPartidos](#)

7.66. Referencia del Archivo src/main/python/Partidos/ADJPartidos/AprendicesDeJosukaSanciones.py ↩

Clases

- class [AprendicesDeJosukaSanciones.AprendicesDeJosukaSanciones](#)

Namespaces

- [AprendicesDeJosukaSanciones](#)

7.67. Referencia del Archivo src/main/python/Partidos/Interface_Partidos.py

Clases

- class [python.Partidos.Interface_Partidos.Interface_Partidos](#)

Namespaces

- [python.Partidos.Interface_Partidos](#)

7.68. Referencia del Archivo src/main/python/Partidos/Interface_Sanciones.py

Clases

- class [python.Partidos.Interface_Sanciones.Interface_Sanciones](#)

Namespaces

- [python.Partidos.Interface_Sanciones](#)

7.69. Referencia del Archivo `src/test/java/Aprendices_de_Josuka/Aprendices_de_Josuka/testDAO.java`

Clases

- class [Aprendices_de_Josuka.Aprendices_de_Josuka.testDAO](#)
Clase donde se hara el testeo de la base de datos.

Paquetes

- package [Aprendices_de_Josuka.Aprendices_de_Josuka](#)
Clases para testear.

7.70. Referencia del Archivo `src/test/java/Aprendices_de_Josuka/Aprendices_de_Josuka/testFachada.java`

Clases

- class [Aprendices_de_Josuka.Aprendices_de_Josuka.testFachada](#)
Clase donde se hara el testeo de la fachada (gestor)

Paquetes

- package [Aprendices_de_Josuka.Aprendices_de_Josuka](#)
Clases para testear.

7.71. Referencia del Archivo `src/test/java/Aprendices_de_Josuka/Aprendices_de_Josuka/testPartidos.java`

Clases

- class [Aprendices_de_Josuka.Aprendices_de_Josuka.testPartidos](#)
Clase donde se hara el testeo con mockito.

Paquetes

- package [Aprendices_de_Josuka.Aprendices_de_Josuka](#)
Clases para testear.