

# TFC. Memoria 2013

---

## **Diseño e implementación de un SGBD para todo lo relacionado a jugadores, partidos y equipos de baloncesto.**

---

Estudiante:

**Ángel Fernández Gutierrez**

Ingeniería Técnica de Informática de Gestión

[afernandezg@uoc.edu](mailto:afernandezg@uoc.edu)

Consultor

**Alex Caminals Sánchez de la Campa**

---

*2013 – Primer Trimestre*

---

# Contenido

Dedicatoria y agradecimientos.....	5
Resumen.....	6
1. Descripción del Proyecto.....	6
1.1. Objetivos del proyecto .....	6
1.2. Planificación del Proyecto .....	6
- PAC1 (Planificación del TFC).....	6
- PAC2: Análisis y Diseño.....	6
- PAC3: Implementación y Pruebas .....	6
- Entrega Final.....	7
1.3. Entregas Parciales.....	8
PAC 1 – Planificación del TFC.....	8
PAC 2 – Análisis y Diseño.....	8
PAC 3 – Implementación .....	8
Entrega Final (Producto, Memoria y Presentación) .....	9
1.4. Recursos Humanos, técnicos y materiales. ....	9
1.5. Plan de contingencias.....	9
2. Análisis y diseño .....	10
2.1. Análisis de requisitos .....	10
2.1.1. Requisitos funcionales.....	10
2.1.2. Requisitos no funcionales.....	13
2.2. Diseño.....	14
2.2.1. Diseño Conceptual.....	14
2.2.1.1. Identificación de las entidades.....	15
2.2.1.2. Identificación de las relaciones. ....	16
2.2.1.3. Atributos de las entidades.....	17
2.2.2. Diseño Lógico.....	19
2.2.3. Diseño Físico .....	21

3.	Implementación.....	33
3.1.	Instalación SGBD Oracle.....	33
3.2.	Configuración Oracle y herramientas.....	35
3.3.	Implementación Diseño E/R.....	36
3.3.1.	Diseño Tablas.....	36
3.3.2.	Diseño de índices y disparadores.....	36
3.4.	Implementación de funcionalidades.....	37
3.4.1.	Implementación procedimientos ABM.....	37
	Procedimientos ABM.....	37
	Alta Jugador: p_alta_jugador.....	37
	Modificar Jugador: p_modificar_jugador.....	37
	Modificar Jugador Asignación Representante: p_modjug_repre.....	38
	Modificar Jugador Baja Representante: p_modjug_bajarepre.....	39
	Modificar Jugador nueva baja: p_modjug_nuevabaja.....	39
	Modificar Jugador añade diagnóstico a baja: p_modjug_adddiagnostico.....	40
	Modificar Jugador elimina diagnóstico a baja: p_modjug_deldiagnostico.....	40
	Modificar Jugador modificar baja: p_modjug_modificarbaja.....	41
	Baja jugador: p_baja_jugador.....	41
	Baja jugador: p_activo_jugador.....	41
	Alta Contrato: p_alta_contrato.....	42
	Modificar Contrato: p_modificar_contrato.....	42
	Alta Partido: p_alta_partido.....	43
	Modificación Partido: p_modif_partido.....	43
	Modificación Partido asignación arbitro: p_asignarb_partido.....	44
	Modificación Partido desasignar árbitro: p_desasignarb_partido.....	44
	Modificación Partido insertar convocados equipo: p_convequi_partido.....	44
	Modificación Partido insertar estadísticas jugador: p_estjug_partido.....	45
	Procedimientos de consulta.....	46
	Procedimiento 1: p_jug_competicion.....	46
	Procedimiento 2: p_equip_tpunts_comp.....	47
	Procedimiento 3: p_5mej_jug_posicion.....	47
	Procedimiento 4: p_contratos_any_repre.....	47
	Procedimiento 5: p_10equip_gastadores.....	48
	Procedimiento 6: p_jug_final_contrato.....	48

3.4.2.	Implementación módulo estadístico.....	49
-	EST_JUG_ACTIVOS.....	49
-	EST_TEMP_MAX_ANNOTADOR.....	49
-	EST_JUG_MEJOR_PAGADO.....	50
-	EST_TOTAL_GANANCIAS.....	50
-	EST_PTGM_SALARIO_MEDIO_ANUAL.....	50
-	EST_CT_PUNTOS_EQ.....	51
-	EST_GM_MJUG_ANY.....	51
<b>3.5.</b>	<b>Juego de pruebas.....</b>	<b>52</b>
3.5.1.	Diseño del juego de pruebas.....	52
3.5.2.	Script del juego de pruebas.....	52
•	Juego de pruebas de ABM de jugadores.....	52
•	Juego de pruebas de ABM de contratos.....	52
•	Juego de pruebas de ABM de partidos.....	52
•	Juego de pruebas de los procedimientos de consultas.....	52
3.5.3.	Ejecución y depuración del juego de pruebas.....	52
	Valoración Económica y Recursos.....	52
	Conclusiones.....	54
	Glosario.....	54
	Bibliografía.....	55
	Anexos.....	55

## Dedicatoria y agradecimientos.

Me gustaría agradecer a todos mis compañeros y compañeras con los que he compartido alguna aportación y que nos hemos estado dando soporte para solucionar dudas y compartir preocupaciones.

Agradecer a todos los consultores que he “sufrido” en todos los semestres cursados a lo largo de esta carrera.

Y por supuesto, a mi pareja, a los dos peques y a todos mis amigos que me han estado apoyando sobre todo a mi gran amigo Marc, que han estado a mi lado y apoyándome en todo lo necesario.

## Resumen

La asociación mundial de jugadores de básquet y con la colaboración con la UOC nos solicita como proyecto de final de carrera el diseño y la implementación de una base de datos para la gestión de todo lo que implica y rodea a los jugadores de básquet a nivel mundial. A continuación se describirá todo lo que se realizará en dicho proyecto.

### 1. Descripción del Proyecto

Tenemos que realizar un desarrollo para la implementación de una gestión de jugadores de baloncesto. Estos datos se almacenarán en un sistema gestión de base de datos y concretamente en Oracle. Se guardará la información referente a los jugadores, equipos, contratos, competiciones, partidos, representantes, técnicos y datos para las estadísticas.

Con esta información el sistema podrá permitir a los gestores de la aplicación a realizar el mantenimiento de todas las entidades, controlar partidos de un fin de semana, estadísticas de cada jugador, clasificaciones por jornadas y por tipo de competición.

Además de esas funcionalidades el cliente nos ha solicitado que también necesita disponer de un módulo estadístico cuyos datos se puedan consultar en directo. Para esto tendremos que configurar un módulo estadístico.

Para esto es de suma importancia que los procedimientos almacenados implementados estén suficientemente especificados con el fin de facilitar el trabajo a futuros desarrollos.

Para ello deberemos de realizar las siguientes tareas:

- Diseño conceptual, lógico y físico de la base de datos
- Generación de los scripts según el diseño físico.
- Generar script para rellenar con datos las tablas maestras para el inicio de la aplicación.
- Implementar los procedimientos que sean necesarios para el control de excepciones y tener un control de logs.
- Diseñar un juego de pruebas integral, para comprobar la funcionalidad correcta del sistema.

#### 1.1. Objetivos del proyecto

El objetivo general del TFC de la UOC es el de proporcionar a nivel mundial, el diseño e implementación de una base de datos que cubra las necesidades para el registro de toda la información referente a los jugadores de baloncesto con todo su entorno. Para que en un futuro se pueda realizar la implementación.

El producto a entregar deberá cumplir con todos los requisitos básicos especificados en la documentación entregada por el cliente. Estos puntos se detallarán más en el apartado que se desarrollará en entregas sucesivas de "Requisitos Funcionales".

#### 1.2. Planificación del Proyecto

Para ello realizaré una planificación en base a una metodología en cascada, que nos define el ciclo de vida del Proyecto. No se empezará una tarea si no se ha realizado la anterior. Este se ajustará a las fechas propuestas por la UOC para el término de las siguientes entregas:

- **PAC1 (Planificación del TFC):** Diseñar y desarrollar un plan de trabajo. Para ello se realizará una planificación lo más detallado y realista posible. Adjuntando los requisitos necesarios para su desarrollo, análisis de riesgos y coste del proyecto.
- **PAC2: Análisis y Diseño.** En este apartado se realizará el análisis detallado de los requisitos del sistema que se nos ha pedido y a continuación en base a este análisis se diseñará la Base de Datos.
- **PAC3: Implementación y Pruebas.** Aquí es donde se realizará el desarrollo y la implementación de lo definido en el punto anterior. Aquí se realizarán los scripts de tablas,

índices, procedimientos, disparadores... Todo ello se implantará en el Sistema Gestor de Base de Datos escogido, que es Oracle.

- **Entrega Final:** Esta entrega se compone de diversos documentos:
  - o Producto: Se compondrá de varios scripts que tendrán el diseño y las pruebas para la implementación en Oracle.
  - o Memoria: Documento de síntesis final.
  - o Presentación: Un documento que resuma el trabajo realizado y los resultados obtenidos.

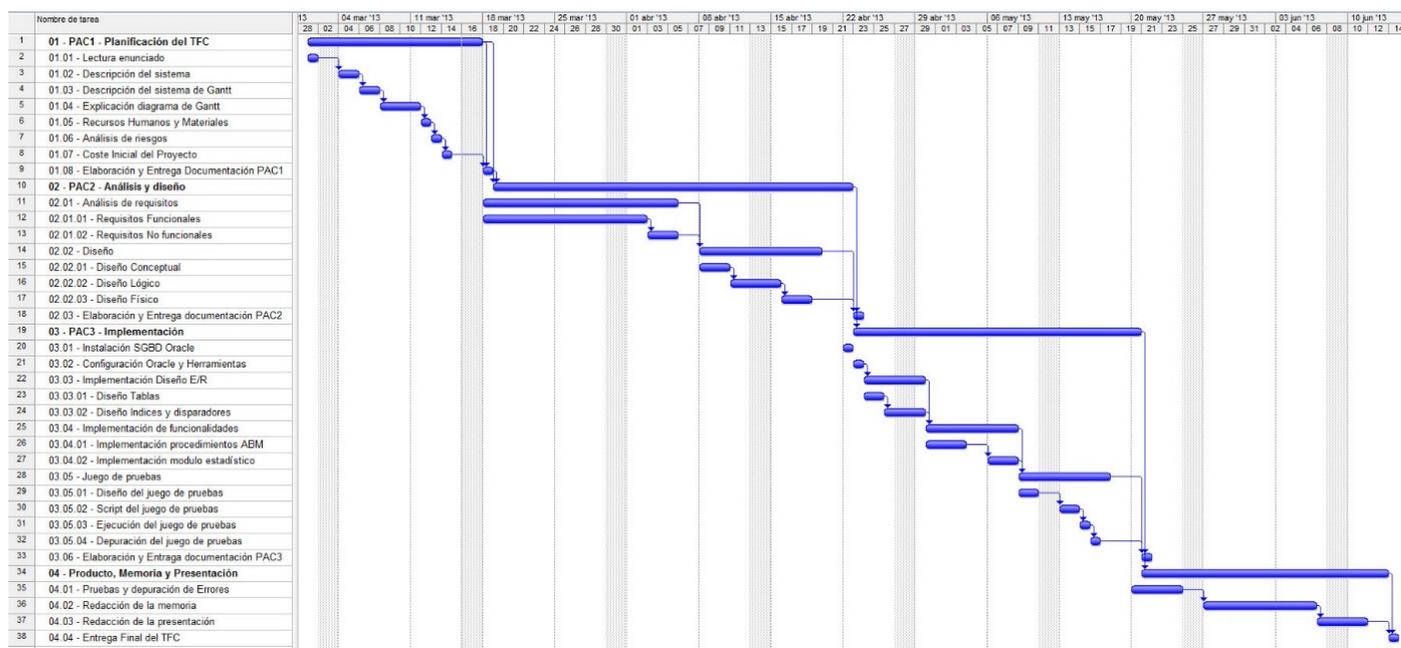
Para realizar la planificación se tendrá en cuenta los siguientes puntos:

- o La semana laboral será de 5 días.
- o Cada día se le dedicara el total de 3 horas.
- o Según la UOC la mayoría de entregas se realizarán el domingo. Para realizar correctamente la planificación, según nuestras especificaciones la fecha de entrega se reflejará el viernes aunque dicha tarea se podrá realizar el domingo, al ser el último día propuesto de entrega.
- o Los días festivos que caigan en días laborales se contabilizarán como laborales para la realización del proyecto.

A continuación se detalla el diagrama de Gantt, según especificaciones anteriores:

TAREA	DIAS	INICIO	FINAL
<b>01 - PAC1 – Planificación del TFC</b>	<b>11 días</b>	<b>01/03/2013</b>	<b>17/03/2013</b>
01.01 - Lectura del enunciado	1 día	01/03/2013	01/03/2013
01.02 - Descripción del sistema	2 días	04/03/2013	05/03/2013
01.03 - Descripción del sistema de Gantt	2 días	06/03/2013	07/03/2013
01.04 - Explicación diagrama de Gantt	2 días	08/03/2013	11/03/2013
01.05 - Recursos Humanos y Materiales	1 día	12/03/2013	12/03/2013
01.06 - Análisis de riesgos	1 día	13/03/2013	13/03/2013
01.07 - Coste Inicial del Proyecto	1 día	14/03/2013	14/03/2013
01.08 - Elaboración y Entrega Documentación PAC1	1 día	15/03/2013	15/03/2013
<b>02 - PAC2 - Análisis y diseño</b>	<b>25 días</b>	<b>18/03/2013</b>	<b>20/04/2013</b>
02.01 - Análisis de requisitos	15 días	18/03/2013	05/04/2013
02.01.01 - Requisitos Funcionales	12 días	18/03/2013	02/04/2013
02.01.02 - Requisitos No funcionales	3 días	03/04/2013	05/04/2013
02.02 - Diseño	10 días	08/04/2013	19/04/2013
02.02.01 - Diseño Conceptual	3 días	08/04/2013	10/04/2013
02.02.02 - Diseño Lógico	3 días	11/04/2013	15/04/2013
02.02.03 - Diseño Físico	3 días	16/04/2013	18/04/2013
02.03 - Elaboración y Entrega documentación PAC2	1 día	19/04/2013	19/04/2013
<b>03 - PAC3 - Implementación</b>	<b>20 días</b>	<b>22/04/2013</b>	<b>19/05/2013</b>
03.01 - Instalación SGBD Oracle	1 día	22/04/2013	22/04/2013
03.02 - Configuración Oracle y Herramientas	1 día	23/04/2013	23/04/2013
03.03 - Implementación Diseño E/R	4 días	24/04/2013	29/04/2013
03.03.01 - Diseño Tablas	2 días	24/04/2013	25/04/2013
03.03.02 - Diseño Índices y disparadores	2 días	26/04/2013	29/04/2013
03.04 - Implementación de funcionalidades	7 días	30/04/2013	08/05/2013
03.04.01 - Implementación procedimientos ABM	4 días	30/04/2013	03/05/2013
03.04.02 - Implementación modulo estadístico	3 días	06/05/2013	08/05/2013
03.05 - Juego de pruebas	7 días	09/05/2013	17/05/2013
03.05.01 - Diseño del juego de pruebas	2 días	09/05/2013	10/05/2013
03.05.02 - Script del juego de pruebas	2 días	13/05/2013	14/05/2013
03.05.03 - Ejecución del juego de pruebas	1 día	15/05/2013	15/05/2013
03.05.04 - Depuración del juego de pruebas	1 día	16/05/2013	16/05/2013
03.06 - Elaboración y Entrega documentación PAC3	1 día	17/05/2013	17/05/2013
<b>04 - Producto, Memoria y Presentación</b>	<b>18 días</b>	<b>20/05/2013</b>	<b>12/06/2013</b>
04.01 - Pruebas y depuración de Errores	5 días	20/05/2013	24/05/2013
04.02 - Redacción de la memoria	9 días	27/05/2013	06/06/2013
04.03 - Redacción de la presentación	3 días	07/06/2013	11/06/2013
04.04 - Entrega Final del TFC	1 día	12/06/2013	12/06/2013

A continuación se mostrara en diagrama de Gantt gráficamente.



### 1.3. Entregas Parciales

Estas entregas se realizarán en base a unas fechas propuestas por la UOC que servirán para la evaluación continuada del TFC. A continuación describiré lo que entregaré en dichas fechas.

#### PAC 1 – Planificación del TFC

En esta primera fase se definirá el plan de trabajo a desarrollar. Para ello utilizaré las metodologías de planificación aprendidas para alcanzar el objetivo pretendido. Junto con el plan de trabajo incluiré los recursos humanos, técnicos y materiales necesarios para el desarrollo del proyecto. Definiré que riesgos supondría dicha implantación y que controles debería aplicar para reducirlo. Y como último punto la valoración inicial en horas y el coste total que supondrá dicho proyecto en base a esta planificación.

#### PAC 2 – Análisis y Diseño

Esta fase comprenderá las siguientes actividades:

- Análisis de los requerimientos proporcionados en el enunciado. Desarrollo del alcance y mejoras propuestas.
- Elaboración del diseño conceptual de la Base de Datos. Para ello utilizaré los diagramas de E/R (entity relationship model).
- Se transforma el esquema conceptual al lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basará el SGBD.
- Se realizará el diseño físico en base al lógico. En este sistema se tendrá en cuenta que el SGBD a la que se va a aplicar será Oracle.

#### PAC 3 – Implementación

- Se instalará y configurará el SGBD Oracle en los equipos asignados al proyecto, para que esté disponible al inicio del desarrollo. Junto con las herramientas a utilizar para el uso y gestión de la Base de datos.
- Elaboración del diseño físico a partir del diagrama E/R realizado en la etapa del diseño físico. Este punto incluye el diseño de las tablas, índices y disparadores.
- Se implementará la funcionalidad para los procedimientos de ABM (alta, baja y modificaciones) y la gestión del módulo estadístico.
- Análisis y diseño de un juego de pruebas. Creación del script y ejecución.

## Entrega Final (Producto, Memoria y Presentación)

Esta etapa comprenderá los siguientes 3 documentos:

- Producto: Este contendrá lo siguiente:
  - Script de creación de las tablas de la Base de datos.
  - Script de relleno de datos de las tablas maestras.
  - Script de creación de procedimientos.
  - Script de testeo de la base de datos para comprobar el correcto funcionamiento.
- Memoria: Debe de sintetizar el trabajo realizado y mostrar que los objetivos propuestos se han alcanzado. Esta debe de contener información relevante que permita entender el problema planteado por la UOC para el desarrollo del TFC, la metodología utilizada para la resolución y la resolución planteada para dicho problema.
- Presentación: Documento sintetizado de forma clara y concisa de todo el trabajo realizado a lo largo de todo el trimestre, junto con los resultados obtenidos.

### 1.4. Recursos Humanos, técnicos y materiales.

En este proyecto tendremos diferentes tipos de perfiles que entrarán en juego, estos serán los siguientes:

- Jefe del proyecto
- Analista
- Programador
- Administrador de la Base de Datos

Estos serán de gran utilidad para nuestra vida profesional, ya que podemos realizar en un mismo proyecto, tareas de diferentes perfiles.

A nivel de hardware se utilizará un ordenador portátil ASUS U36J y de soporte uno de sobremesa, ambos con la capacidad para la ejecución de las herramientas a continuación descritas.

En estos ordenadores se instalarán los siguientes programas para la realización del proyecto:

- Microsoft Word: para realizar la documentación oportuna
- Adobe Acrobat profesional: Para la transformación de cualquier documento a un documento con extensión pdf.
- Microsoft Excel: Para la realización de tablas.
- Photoshop: Para el tratamiento de algunas imágenes.
- Ms Project: Para hacer el diagrama de Gantt.
- Oracle xe: El sistema gestor de base de datos.
- Toad for Oracle: Para realizar los scripts y ejecutarlos.
- UltraEdit: Para gestionar ficheros planos.
- Ms Visio: Para hacer diagramas.

### 1.5. Plan de contingencias.

Existen diversos riesgos o factores que pueden afectar de alguna forma el desarrollo del proyecto y así no tener tiempo de reaccionar para el cumplimiento de las fechas acordadas. Para prevenir estos contratiempos tendré en cuenta estos puntos:

- Para cuando haya un problema de error de datos, error del sistema, problemas del hardware o errores en el SGBD, tendré una copia diaria de información tanto en local como en la nube. En este caso utilizaré dos tipos la de dropbox y la de google drive.
- Cuando por motivos personales (compromisos o enfermedades) o laborales (ampliación del horario laboral) no se pueda mantener con el plan de trabajo, se podrá utilizar los días de fines de semana para ponerse al día.

## 2. Análisis y diseño

### 2.1. Análisis de requisitos

En este apartado se incluye todos los requisitos y funcionalidades que tengan que cumplir el diseño de la base de datos. Además de las nuevas propuestas que mejorarán y amplían su funcionalidad.

#### 2.1.1. Requisitos funcionales.

Los requisitos funcionales establecidos por el cliente y las mejoras propuestas para el modelo de datos a diseñar son los siguientes:

**R1.** El modelo tiene que permitir guardar los datos de los jugadores, estos datos son los siguientes:

- Cada jugador tendrá un número identificativo único.
- Se guardarán los datos personales como el nombre, apellidos, fecha de nacimiento, altura, peso, nacionalidad, país al que pertenece o la nacionalidad, opcionalmente la web personal del jugador o una url donde tenga un video promocional.
- Un jugador tiene asignado un representante, no puede tener más de uno al mismo tiempo.
- Posición donde juega habitualmente. Siendo esta una referencia a un código donde cada número significará una posición. Un jugador solo puede jugar en una posición y no en varias.
- Se tendrá en cuenta que se guardará un historial de equipos en los cuales ha jugado o ha sido contratado y en el que está contratado actualmente.
- Un jugador tiene dos estados principales activo o retirado. Solo podrá tener un estado. No se propone un historial de los estados en los que ha estado, ya que se cree que es poco útil.
- Un jugador que este activo puede tener los siguientes subestados en una fecha:
  - Activo
  - Baja médica
  - Baja por motivos personales
- Solo en caso que un jugador este de baja médica se guardará el diagnóstico. Para que un jugador esté de baja debe estar en activo, no puede darse el caso que esté retirado y se le asigne una baja. Para que un jugador esté de baja se guardará la siguiente información:
  - Disponibilidad estimada en número de días para que el jugador vuelva a estar activo.
  - Fecha de alta de la baja, que no tiene por qué coincidir con la fecha estimada calculada. Esta se calcula a partir de la suma de los días estimados más la fecha de inicio de la baja.

Mejoras:

- Se guardará un historial de todos los representantes que ha tenido este jugador, en donde también se recogerá el representante actual. Al mismo tiempo un jugador no puede tener dos representantes.
- Todas las bajas se guardarán en un historial, donde se reflejará la fecha de inicio de la baja o lesión, cuando se le dió de alta y la disponibilidad estimada. En el mismo tiempo un jugador solo puede tener una baja y hasta que no se le dé el alta no puede tener otra baja.
- Una baja médica puede tener varios diagnósticos, en donde es el diagnóstico el que define los días estimados de la baja. El diagnóstico de los días estimados más alto, nos proporcionan los días estimados de una baja. Normalmente cada baja tiene un diagnóstico pero se da la posibilidad de que se puedan sufrir varios. Se podría dar el caso que una baja pueda ser por dos causas como por ejemplo un esguince en el tobillo y en la muñeca a causa de sufrir una caída.

**R2.-** Tiene que permitir guardar los datos sobre los equipos de baloncesto teniendo en cuenta lo siguiente:

- Cada equipo se identificará por un número y será único.
- En cada club o sociedad se guardará el nombre del representante legal, dirección de las oficinas centrales, municipio, país, teléfono, url de la página web.
- El equipo tiene asignado un pabellón donde juega los partidos habitualmente.
- Un equipo pertenece a un club o sociedad (anónima, limitada, ...). En el caso que sea un club se guardará el número de socios.
- Cada equipo tiene un equipo técnico, este lo compondrán diferentes personas con cargos diferentes. Se definirá quien es el primer entrenador, el segundo, el fisioterapeuta, etc. Todos los tipos de técnicos estarán registrados. Una tipología de técnicos al mismo tiempo solo la puede ejercer un técnico, es decir, que sólo puede haber una persona que realice las funciones de primer entrenador en un intervalo de tiempo.

Mejoras:

- Un club puede tener varios equipos de baloncesto pero en diferentes categorías. Se diferencia entre la terminología de club o sociedad y un equipo.
- Como añadido se tendrá un historial de que funciones ha realizado un técnico y en que equipos.

**R3.-** El modelo permitirá guardar los datos de los contratos de los jugadores con los equipos. A continuación mostraré las restricciones y necesidades:

- Se guardará la fecha del contrato a partir del cual pasa a jugar con el nuevo equipo.
- Para poder hacerlo más real la duración de los contratos la define dos fechas, la de inicio y la de final del contrato. En principio también guardaremos un campo con la fecha prevista de fin del contrato, ya que un jugador puede rescindir su contrato para ser contratado por otro equipo, en ese caso cambiaremos la fecha final del contrato pero guardaremos la fecha de fin del contrato que se había firmado.
- El identificador del jugador al que hace referencia el contrato. Cuando se produzca un alta en una fecha determinada, dicho jugador no puede tener contrato en esa fecha con otro equipo. Se tendrá que dar de baja el contrato anterior a una fecha de un día anterior.
- El identificador del equipo que realiza el contrato y el identificador del equipo que vende en el caso que dicho jugador pertenezca a un equipo y lo venda a otro. Se podría dar el caso que un jugador no tenga contrato con nadie y un equipo le haga contrato.
- En caso que el contrato se haya realizado por una venta de un equipo se guardará la compensación económica que ha recibido el equipo que ha vendido. Dicha operación se guardará en todas las monedas que se dispongan.
- Se guardará también el representante del jugador cuando se firmó el contrato.
- Para el salario y el traspaso se guardarán en la moneda que se realice la operación y después tenemos una tabla de equivalencias hacia el euro que es la moneda principal.
- El salario que se guardará es lo que cobrará bruto el jugador anualmente. Para saber lo que cobra un mes se tendrá que dividir el salario bruto anual entre 12 meses. Si decidimos que un mes tiene 30 días, para saber lo que cobra un día tendremos que dividir el salario bruto mensual entre 30.

**R4.-** Tiene que permitir el almacenamiento de información referente a los partidos para poder realizar las estadísticas. Para cada partido se guardará los datos concretos de cada jugador que ha participado en dicho encuentro, este jugador antes ha tenido que ser convocado para poderlo jugar. El jugador convocado y el que ha participado se guardará en la misma tabla, la única diferencia es que el que ha participado tiene rellenado los datos estadísticos como el de minutos jugados. La información que se registrará será la siguiente:

- Los minutos jugados por un jugador (MI).
- Los puntos realizados en ese partido (PT).
- Los lanzamientos realizados de tiro libre (T1) y los encestandos (T1E).
- Lanzamientos realizados de dos puntos (T2) y los encestandos (T2E).
- Lanzamientos realizados de tres puntos (T3) y los encestandos (T3E).
- Los rebotes defensivos (RD) y ofensivos (RO).
- Las asistencias realizadas (AS).
- Los tapones realizados (TF) y los tapones que le han hecho (TC).

- El número de pelotas recuperadas (PR) y perdidas (PP).
- Número de faltas realizadas (FC) y recibidas (FR).

La valoración del jugador se calculará de la siguiente forma:

$$V = PT + T1E + T2E + T3E + RD + RO + AS + TF + PR + FR - T1 - T2 - T3 - TC - PP - FC$$

La valoración ponderada de cada jugador se calculará aplicando un factor (p) que se multiplica a la valoración del partido.  $VP = V * p$ .

El factor p está asociado a cada competición, este factor es para diferenciar la importancia de la competición a mayor factor más importante es la competición. Por ejemplo la Euroliga tendrá un valor de 10 y una liga de un país tendrá un factor de 9. Esto se realiza para que tenga más peso las ligas o competiciones más importantes.

**R5.-** El modelo permitirá guardar información sobre todas las competiciones de baloncesto. A continuación se detallará lo que contendrá:

- Cada competición tendrá un identificador único.
- Tendrá un nombre.
- Ámbito en que se realizara la competición (Nacional, Continental, Mundial)
- Género (masculino o femenino).
- Modalidad (normal o silla de ruedas).
- País o continente donde se realiza la competición. Esto dependerá del ámbito seleccionado anteriormente. En el caso que sea mundial no se pondrá nada ni en país ni en continente.
- Número de árbitros que serán designados para cada partido de la competición.
- Se guardará el número de categoría, en el caso de España tendremos que la primera categoría es la ACB y la segunda es la LEB.
- Cada competición tiene muchas temporadas, las que tienen unos partidos definidos. Dicha competición dura un intervalo de fechas. En el caso de competiciones nacionales suelen empezar a mediados de un año y acabar a mediados del año próximo, caso contrario lo de los mundiales o europeos o juegos olímpicos que se definen en un año en concreto.
- Aquí se definirá el factor p que definirá el nivel de la competición. Este se utilizará para el módulo de estadísticas, para el cálculo de la valoración del jugador ponderada.

Mejoras:

- Se guardará un parámetro definiendo si la competición tiene un formato de liga o no, en caso que tenga formato de liga se guardará el número de jornadas tanto en la temporada como en los partidos.
- En caso que no sea tipo liga se guardará el número de partidos que un equipo tiene que realizar para llegar a la final de la competición.

**R6.-** Se guardará información sobre todos los partidos de todas las competiciones. Estos almacenarán la siguiente información:

- Día y hora de inicio del partido.
- Pabellón donde tendrá lugar el encuentro.
- Equipo local y equipo visitante.
- El identificador de la temporada a la que se refiere dicho encuentro. A partir de la temporada se identificará también la competición a la que se refiere el encuentro.
- En caso que la competición sea del tipo liga, se guardará el número de jornada a la que se refiere el partido. En caso que no sea tipo liga se guardará el número de partido disputados esa categoría para esa temporada hasta entonces contando dicho partido.
- Se registrarán los datos de los árbitros que arbitrarán dicho encuentro. En base al tipo de competición el número de árbitros variará. Tendremos varios tipos de árbitros el principal, auxiliar principal, auxiliar secundario. En un futuro también se pueden incluir más tipos de árbitros como son los de mesa o cualquier otro.
- Jugadores convocados para el partido, como máximo se podrán convocar a 12.

**R7.-** Se realizará las siguientes funcionalidades en base a los requisitos antes descritos:

- Se implementarán los procedimientos de Alta – Baja – Modificación (ABM) de los jugadores, contratos y partidos.
- No se implementarán los procedimientos de Alta – Baja – Modificación (ABM) de las tablas generadas y que no tenga que ver con la funcionalidad anterior, como ejemplo de estas tablas encontramos: países, representantes de jugadores, equipos, clubs, estados, diagnósticos de lesiones, monedas, etc.

- Se implementará y describirá los procedimientos para la consulta de información. Estos procedimientos serán los siguientes:
  - Listado de todos los jugadores de una competición con todos los datos junto con datos del contrato y sobre todo el de la fecha de finalización.
  - Listado de todos los equipos de una competición ordenados por número de puntos totales a favor en la temporada actual.
  - A partir de un año y un representante de jugadores, obtener el número de contratos de jugadores firmados y el valor económico total de cada uno de ellos.
  - Dado un año concreto el listado de los 10 equipos que más se han gastado en la obtención de jugadores, ordenados de quien ha gastado más a quien ha gastado menos.
  - Dado un país, un género y una modalidad, se mostrarán todos los jugadores que acaben contrato a final de la temporada presente o que están en activo pero sin equipo.

Mejoras:

- Como mejora se podrá realizar un listado de los puntos según la jornada. Solo en el caso de las competiciones que son con formato liga.
- De una competición y temporada, clasificación por jornadas, a partir de los partidos ganados en jornadas anteriores. Sólo en competiciones con formato liga.

**R8.-** Módulo estadístico. Este se tiene que alimentar a partir de los procedimientos que implementan las funcionalidades descritas. Este tiene que dar los siguientes resultados:

- Número total de jugadores en activo en todos los géneros y modalidades.
- Dada una competición, quien es el máximo anotador en la temporada en curso o bien la última temporada en el caso que no haya empezado la nueva.
- Que jugador es el mejor pagado de cada modalidad y género.
- El jugador que más ha ganado a lo largo de toda su carrera. Para ello tendremos que definir como se calcula lo que ha ganado un jugador. Para ello los meses del contrato que se han trabajado enteros se contabilizan como si tuvieran 30 días y el pago se realizará en base al contrato y concretamente a lo que cobra bruto por 1 día. Para los meses que no los ha trabajado enteros que normalmente son al inicio del contrato y al final se realizará la equivalencia de los días en base al mes. Es decir si empieza su contrato el 15 de enero, ha trabajado 16 días, a partir de ahora se hace una regla de tres para realizar la equivalencia en base a los 30 días. En este caso la equivalencia sería la siguiente:  $(16 * 30) / 31$ .
- Dado un país, una temporada, un género y una modalidad, el salario medio anual de los jugadores.
- Dada una competición y una temporada, los mejores equipos ofensivos y defensivos.
- Para cada género y modalidad, el mejor jugador del mundo en el último año, este es el que mejor media de todas sus valoraciones ponderadas tenga.

### 2.1.2. Requisitos no funcionales.

En este apartado se desarrollara que requisitos no son funcionales:

- El sistema de gestión de base de datos será Oracle.
- En el apartado del sistema operativo no se desarrollará en ninguno en concreto ya que Oracle puede ser instalado en cualquier sistema operativo.
- Política de backups: Este apartado no se contemplará en este desarrollo, ya que solo se nos solicita para desarrollar un sistema para el almacenamiento de dicha información. Este tema se debería de tratar una vez realizada la implementación.
- Los procedimientos dispondrán como mínimo de un parámetro de salida, llamado RSP de tipo String. Este indicará si la ejecución ha finalizado con éxito 'OK' o si ha fracasado 'ERROR + Tipo de Error'. Todas las llamadas a estos procedimientos se guardarán en una tabla de logs, junto con todos sus datos, parámetros de entrada y salida.
- Carga inicial del sistema: Se tiene que tener en cuenta que en un inicio se tendrá que realizar la carga de datos de tablas auxiliares.





Principales	Intermedias	Auxiliares
CONTINENTES	EQUIPO_TECNICO	EST_CT_PUNTOS_EQ
PAISES	PARTIDOS	EST_GM_MJUG_ANY
MUNICIPIOS	EQUIPOS_TEMPORADAS	EST_JUG_ACTIVOS
CLUBS	PARTIDOS_ARBITROS	EST_JUG_MEJOR_PAGADO
PERSONES	PARTIDOS_JUGADORES	EST_PTGM_SALARIO_MEDIO_ANUAL
TECNICOS	JUG_REPRESENTANTES	EST_TEMP_MAX_ANOTADOR
TIPO_TECNICOS	CONTRATOS_JUGADORES	EST_TOTAL_GANANCIAS
PABELLON	BAJAS_DIAGNOSTICOS	LOGS
EQUIPOS		
COMPETICIONES		
TEMPORADAS		
AMBITOS		
GENEROS		
MODALIDADES		
TIPO_ARBITROS		
ARBITROS		
JUGADORES		
JUG_POSICIONES		
REPRESENTANTES		
MONEDAS		
TIPOS_BAJA		
DIAGNOSTICOS		
JUG_BAJA		
TIPO_CLUBS		

### 2.2.1.2. Identificación de las relaciones.

Entidad Origen	Entidad Destino	Descripción
AMBITOS	COMPETICIONES	Las competiciones tienen asignado un ámbito.
ARBITROS	PARTIDOS_ARBITROS	La entidad partidos_arbitros tiene asignado un árbitro.
CLUBS	EQUIPOS	Los equipos están asignados a un club.
COMPETICIONES	TEMPORADAS	Las temporadas pertenecen a una competición.
CONTINENTES	COMPETICIONES	Las competiciones pueden pertenecer al ámbito de continente.
CONTINENTES	PAISES	Las competiciones pueden pertenecer al ámbito de un país.
DIAGNOSTICOS	BAJAS_DIAGONOSTICOS	Las bajas de los jugadores tiene asignados un diagnóstico
EQUIPOS	PARTIDOS	Los partidos tienen un equipo local que lo juega.
EQUIPOS	PARTIDOS_JUGADORES	Los jugadores que han disputado un partido son de un equipo.
EQUIPOS	PARTIDOS	Los partidos tienen un equipo visitante que lo juega.
EQUIPOS	EQUIPO_TECNICO	Los técnicos tienen contrato con un equipo durante una fecha.
EQUIPOS	CONTRATOS_JUGADORES	Los jugadores tienen contrato con un equipo durante una fecha.
EQUIPOS	CONTRATOS_JUGADORES	Los contratos de los jugadores se puede dar el caso que haya un equipo vendedor.
GENEROS	COMPETICIONES	Las competiciones tienen asignado un género.
JUGADORES	CONTRATOS_JUGADORES	Los contratos tienen asignado un jugador que lo firma.
JUGADORES	PARTIDOS_JUGADORES	Las estadísticas de los jugadores están ligados con el jugador que las realiza.
JUGADORES	JUG_REPRESENTANTES	Los representantes están asignados durante un tiempo a un jugador.
JUGADORES	JUG_BAJA	Una baja está asignada a un jugador concreto.
JUG_BAJA	BAJAS_DIAGONOSTICOS	Unos diagnósticos tienen asociado a una baja de un jugador concreto.
JUG_POSICIONES	JUGADORES	Los jugadores tienen asignado una posición de juego.
MODALIDADES	COMPETICIONES	Las competiciones tienen unas modalidades definidas.
MONEDAS	CONTRATOS_JUGADORES	Los contratos tienen un salario bruto anual en una moneda determinada.
MONEDAS	CONTRATOS_JUGADORES	Los contratos tienen un traspaso en una moneda determinada.
MUNICIPIOS	CLUBS	Los clubs están en un municipio.
MUNICIPIOS	PABELLON	Los pabellones están en un municipio.
PABELLON	EQUIPOS	Los equipos tienen un pabellón por defecto de juego.
PAISES	MUNICIPIOS	Los municipios pertenecen a un país.
PAISES	JUGADORES	Los jugadores están nacionalizados en un país.

PAISES	JUGADORES	Los jugadores están federados en un país.
PAISES	COMPETICIONES	Las competiciones son de un país.
PARTIDOS	PARTIDOS_ARBITROS	Un árbitro esta designado para pitar un partido.
PARTIDOS	PARTIDOS_JUGADORES	Un jugador está convocado para jugar un partido.
PERSONES	JUGADORES	Un jugador es una persona concreta.
PERSONES	CLUBS	Un club tiene un representante que es una persona.
PERSONES	ARBITROS	Un árbitro es una persona.
PERSONES	TECNICOS	Un técnico es una persona.
PERSONES	REPRESENTANTES	Un representante es una persona.
REPRESENTANTES	CONTRATOS_JUGADORES	Los contratos de los jugadores los firma también un representante.
REPRESENTANTES	JUG_REPRESENTANTES	Los jugadores tienen en una fecha un representante asignado.
TECNICOS	EQUIPO_TECNICO	Los equipos tienen durante unas fechas a un técnico asignado con unas funciones.
TEMPORADAS	EQUIPOS_TEMPORADAS	Las temporadas tienen unos equipos que la jugarán.
TEMPORADAS	PARTIDOS	Los partidos se disputan en una temporada.
TIPO_ARBITROS	PARTIDOS_ARBITROS	Los árbitros de los partidos son de una tipología.
TIPO_CLUBS	CLUBS	Los clubs son de un tipo.
TIPOS_BAJA	JUG_BAJA	Las bajas de los jugadores son de un tipo.
TIPO_TECNICOS	EQUIPO_TECNICO	El cuerpo técnico de un equipo tiene un tipo definido que será su cargo.
EQUIPOS	EST_CT_PUNTOS_EQ	El mejor equipo defensivo.
EQUIPOS	EST_CT_PUNTOS_EQ	El mejor equipo ofensivo.
EQUIPOS	EQUIPOS_TEMPORADAS	El identificador del equipo que está jugando esa temporada
GENEROS	EST_JUG_MEJOR_PAGADO	Género del jugador mejor pagado.
GENEROS	EST_PTGM_SALARIO_MEDIO_ANUAL	Género del salario medio anual.
GENEROS	EST_JUG_ACTIVOS	Género del número de jugadores activos.
GENEROS	EST_GM_MJUG_ANY	Género del mejor jugador del año.
JUGADORES	EST_GM_MJUG_ANY	Identificador al mejor jugador del año.
JUGADORES	EST_TEMP_MAX_ANOTADOR	Identificador al máximo anotador.
JUGADORES	EST_TOTAL_GANANCIAS	Identificador al jugador que más ha ganado.
JUGADORES	EST_JUG_MEJOR_PAGADO	Identificador al jugador mejor pagado.
MODALIDADES	EST_JUG_ACTIVOS	Modalidad del número de jugadores activos.
MODALIDADES	EST_JUG_MEJOR_PAGADO	Modalidad del jugador mejor pagado.
MODALIDADES	EST_PTGM_SALARIO_MEDIO_ANUAL	Modalidad del salario medio anual.
MODALIDADES	EST_GM_MJUG_ANY	Modalidad de mejor jugador del año.
PAISES	EST_PTGM_SALARIO_MEDIO_ANUAL	El país del salario medio anual.
TEMPORADAS	EST_CT_PUNTOS_EQ	Temporada del mejor equipo defensivo y ofensivo.
TEMPORADAS	EST_TEMP_MAX_ANOTADOR	Temporada del máximo anotador.

### 2.2.1.3. Atributos de las entidades.

A continuación se muestra una tabla con todas las entidades y para cada entidad los atributos que tiene. También se define que atributos formarán la primary key estos están definidos entre paréntesis después de la palabra PK.

ENTIDAD	ATRIBUTOS
AMBITOS	PK (AMBITOS_ID), NOMBRE
ARBITROS	PK (PERSONES_ID), F_ALTA, F_BAJA, URL_WEB
BAJAS_DIAGNOSTICOS	PK (JUG_BAJA_ID, DIAGNOSTICOS_ID)
CLUBS	PK (CLUBS_ID), NOMBRE, MUNICIPIOS_ID, TIPO_CLUBS_ID, REPRESENTANTE_ID, DIR_OF_CENTRAL, CPOSTAL, NUM_SOCIOS, TELEFONO, URL_WEB
COMPETICIONES	PK (COMPETICIONES_ID), NOMBRE, CATEGORIA, CONTINENTES_ID, PAISES_ID, AMBITOS_ID, GENEROS_ID, MODALIDADES_ID, FACTOR_P, NUM_ARBITROS, TIPO_LIGA
CONTINENTES	PK (CONTINENTES_ID), NOMBRE
CONTRATOS_JUGADORES	PK (CONTRATOS_ID), JUGADORES_ID, REPRESENTANTES_ID, EQUIPOS_C_ID, EQUIPOS_V_ID, MONEDAS_S_ID, MONEDAS_T_ID, F_INICIO, F_FIN, F_FINPREVISTA, SALARIO_BRUTO_ANUAL, TRASPASO, SALARIO_BRUTO_EUROS, TRASPASO_EUROS
DIAGNOSTICOS	PK (DIAGNOSTICOS_ID), NOMBRE, DIAS_ESTIMADOS

EQUIPOS	PK (EQUIPOS_ID), CLUBS_ID, PABELLON_ID, NOMBRE, F_ALTA, F_BAJA
EQUIPOS_TEMPORADAS	PK (EQUIPOS_ID, COMPETICIONES_ID, F_INICIO)
EQUIPO_TECNICO	PK (TECNICOS_ID, EQUIPOS_ID, F_INICIO), TIPO_TECNICOS_ID, F_FIN
GENEROS	PK (GENEROS_ID), NOMBRE
JUGADORES	PK (PERSONES_ID), JUG_POSICIONES_ID, NACIONALIDAD_ID, FEDERACION_ID, ACTIVO, ALTURA, NUM_FEDE, PESO, URL_WEB, URL_VIDEO
JUG_BAJA	PK (JUG_BAJA_ID), JUGADOR_ID, TIPOS_BAJA_ID, F_INICIO, F_FIN, OBSERVACIONES
JUG_POSICIONES	PK (JUG_POSICIONES_ID), NOMBRE
JUG_REPRESENTANTES	PK (JUGADOR_ID, REPRESENTANTE_ID, F_ALTA), F_BAJA, F_PREVISTA
LOGS	PK (LOGS_ID), F_EJECUCION, PARAM_ENTRADA, PARAM_SALIDA, PROCEDIMIENTO
MODALIDADES	PK (MODALIDADES_ID), NOMBRE
MONEDAS	PK (MONEDAS_ID), NOMBRE, EURO, EQUIVALENCIA
MUNICIPIOS	PK (MUNICIPIOS_ID), PAISES_ID, NOMBRE, CPOSTAL
PABELLON	PK (PABELLON_ID), NOMBRE, MUNICIPIOS_ID, DIRECCION, CPOSTAL, CAPACIDAD
PAISES	PK (PAISES_ID), CONTINENTES_ID, NOMBRE
PARTIDOS	PK (PARTIDOS_ID), EQUIPOS_L_ID, EQUIPOS_V_ID, COMPETICIONES_ID, F_INICIO, DIA_HORA, JORNADA, PUNTOS_LOCAL, PUNTOS_VISITANTE
PARTIDOS_ARBITROS	PK (PARTIDOS_ID, ARBITROS_ID), TIPO_ARBITROS_ID
PARTIDOS_JUGADORES	PK (PARTIDOS_ID, JUGADORES_ID), EQUIPOS_ID, ASISTENCIAS, FALTAS_COMETIDAS, FALTAS_RECIBIDAS, MINUTOS_JUGADOS, PELOTAS_PERDIDAS, PELOTAS_RECUPERADAS, PUNTOS, REBOTES_DEFENSIVOS, REBOTES_OFENSIVOS, TAPONES_CONTRA, TAPONES_FAVOR, TIROS1, TIROS1E, TIROS2, TIROS2E, TIROS3, TIROS3E, VALORACION_JUGADOR
PERSONES	PK (PERSONES_ID), DNI, NOMBRE, APELLIDOS, SEXO, F_NACIMIENTO
REPRESENTANTES	PK (PERSONES_ID), TITULACIONES, URL_WEB, URL_VIDEO
TECNICOS	PK (PERSONES_ID), TITULACIONES, URL_WEB, F_TITULACION_CARGO
TEMPORADAS	PK (COMPETICIONES_ID, F_INICIO), F_FIN, NUM_JORNADAS
TIPO_ARBITROS	PK (TIPO_ARBITROS_ID), NOMBRE
TIPO_CLUBS	PK (TIPO_CLUBS_ID), NOMBRE
TIPOS_BAJA	PK (TIPOS_BAJA_ID), NOMBRE
TIPO_TECNICOS	PK (TIPO_TECNICOS_ID), NOMBRE
EST_CT_PUNTOS_EQ	PK (COMPETICIONES_ID, F_INICIO), EQUIPOS_D_ID, EQUIPOS_O_ID, PUNTOS_DEFENSIVOS, PUNTOS_OFENSIVOS
EST_GM_MJUG_ANY	PK (GENEROS_ID, MODALIDADES_ID, ANIO), JUGADORES_ID, MED_VP
EST_JUG_ACTIVOS	PK (GENEROS_ID, MODALIDADES_ID), NUM_JUGADORES
EST_JUG_MEJOR_PAGADO	PK (GENEROS_ID, MODALIDADES_ID), JUGADORES_ID, SALARIO_BRUTO_ANUAL_EUROS
EST_PTGM_SALARIO_MEDIO_ANUAL	PK (PAISES_ID, GENEROS_ID, MODALIDADES_ID), TEMPORADA_ANIO_INICI, SALARIO_MEDIO_ANUAL
EST_TEMP_MAX_ANOTADOR	PK (COMPETICIONES_ID, F_INICIO), JUGADORES_ID, PUNTOS_ANOTADOS
EST_TOTAL_GANANCIAS	PK (JUGADORES_ID), TOTAL_GANANCIAS





### 2.2.3. Diseño Físico

Lo siguiente es trasladar el diseño que hemos realizado al sistema de tablas específicas para Oracle. A continuación mostraremos detalladamente el diseño que definiremos.

TABLA	DESCRIPCIÓN TABLA		
<i>AMBITOS</i>	Están definidos si la competición tiene un ámbito nacional, continental, mundial y se podría dar el caso que también fuera provincial.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>AMBITOS_ID</b>	Identificador del Ámbito.		
VARCHAR2(2)	X		AMBITOS_ID IS NOT NULL
<b>NOMBRE</b>	Nombre del Ámbito en cuestión.		
VARCHAR2(20)			NOMBRE IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<i>ARBITROS</i>	Aquí se definen todos los árbitros que existen para realizar los partidos.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>PERSONES_ID</b>	El identificador de la persona.		
NUMBER	X	PERSONES	PERSONES_ID IS NOT NULL
<b>F_ALTA</b>	Fecha de alta del árbitro.		
DATE			F_ALTA IS NOT NULL
<b>F_BAJA</b>	Fecha de baja en el momento que se retire.		
DATE			
<b>URL_WEB</b>	La dirección web para promocionarse.		
VARCHAR2(200)			

TABLA	DESCRIPCIÓN TABLA		
<i>BAJAS_DIAGNOSTICOS</i>	Guarda todos los diagnósticos de una baja normalmente hay uno pero puede darse el caso que hayan más.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>JUG_BAJA_ID</b>	Identificador de la baja del jugador.		
NUMBER	X	JUG_BAJA	JUG_BAJA_ID IS NOT NULL
<b>DIAGNOSTICOS_ID</b>	Identificador del diagnóstico que tiene la baja.		
NUMBER	X	DIAGNOSTICOS	DIAGNOSTICOS_ID IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<i>CLUBS</i>	Registra todos los clubs que existen.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>CLUBS_ID</b>	Identificador del club. Este es una secuencia: SEQ_CLUBS.		
NUMBER	X		CLUBS_ID IS NOT NULL
<b>NOMBRE</b>	Nombre del club.		
VARCHAR2(50)			NOMBRE IS NOT NULL
<b>MUNICIPIOS_ID</b>	El municipio donde está registrado el club.		
NUMBER		MUNICIPIOS	MUNICIPIOS_ID IS NOT NULL
<b>TIPO_CLUBS_ID</b>	Identificador del tipo de club que es.		
VARCHAR2(2)		TIPO_CLUBS	TIPO_CLUBS_ID IS NOT NULL
<b>REPRESENTANTE_ID</b>	Identificador de la persona que es el representante o presidente.		
NUMBER		PERSONES	REPRESENTANTE_ID IS NOT NULL
<b>DIR_OF_CENTRAL</b>	Los datos de la dirección de las oficinas centrales.		
VARCHAR2(200)			DIR_OF_CENTRAL IS NOT NULL
<b>CPOSTAL</b>	Código de barras del municipio donde están las oficinas centrales.		
VARCHAR2(5)			CPOSTAL IS NOT NULL
<b>NUM_SOCIOS</b>	En caso que sea un club el número de socios.		
NUMBER(10)			

<b>TELEFONO</b>	El teléfono de las oficinas del club.		
VARCHAR2(10)			
<b>URL_WEB</b>	La web de referencia del club.		
VARCHAR2(200)			

TABLA	DESCRIPCIÓN TABLA		
<b>COMPETICIONES</b>	Se guardarán todas las competiciones.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>COMPETICIONES_ID</b>	El identificador de la competición. Este lo define la siguiente secuencia: SEQ_COMPETICIONES.		
NUMBER	X		COMPETICIONES_ID IS NOT NULL
<b>NOMBRE</b>	Nombre de la competición.		
VARCHAR2(100)			NOMBRE IS NOT NULL
<b>CATEGORIA</b>	Identificador de la categoría		
NUMBER			CATEGORIA IS NOT NULL
<b>CONTINENTES_ID</b>	Identificador del continente en el caso que esa competición se realice a nivel de continente.		
VARCHAR2(2)		CONTINENTES	
<b>PAISES_ID</b>	Identificador del país en el caso que esa competición se realice a nivel del país.		
NUMBER		PAISES	
<b>AMBITOS_ID</b>	Identificador del ámbito de la competición.		
VARCHAR2(2)		AMBITOS	AMBITOS_ID IS NOT NULL
<b>GENEROS_ID</b>	Identificador del género de la competición.		
VARCHAR2(1)		GENEROS	GENEROS_ID IS NOT NULL
<b>MODALIDADES_ID</b>	Identificador de la modalidad de la competición.		
VARCHAR2(1)		MODALIDADES	MODALIDADES_ID IS NOT NULL
<b>FACTOR_P</b>	Es el factor que se utilizara para obtener la valoración ponderada de un jugador.		
NUMBER(2)			FACTOR_P BETWEEN 0 AND 10
<b>NUM_ARBITROS</b>	Número de árbitros de los que dispondrá un partido.		
NUMBER(1)			
<b>TIPO_LIGA</b>	En el caso que sea una liga con unas jornadas pondremos una S y en caso que sean eliminatorias una N.		
VARCHAR2(1)			TIPO_LIGA IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<b>CONTINENTES</b>	Están definidos los continentes.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>CONTINENTES_ID</b>	Identificador del continente.		
VARCHAR2(2)	X		CONTINENTES_ID IS NOT NULL
<b>NOMBRE</b>	Es el nombre del continente.		
VARCHAR2(20)			NOMBRE IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<b>CONTRATOS JUGADORES</b>	Se guardarán todos los datos de los contratos de los jugadores con los equipos.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>CONTRATOS_ID</b>	El identificador del contrato.		
NUMBER	X		CONTRATOS_ID IS NOT NULL
<b>JUGADORES_ID</b>	El identificador del jugador que realiza el contrato.		
NUMBER		JUGADORES	JUGADORES_ID IS NOT NULL
<b>REPRESENTANTES_ID</b>	El representante que ha conseguido que el jugador firme el contrato. Se puede llegar a dar el caso que no haya representante.		
NUMBER		REPRESENTANTES	
<b>EQUIPOS_C_ID</b>	El identificador que compra al jugador.		
NUMBER		EQUIPOS	EQUIPOS_C_ID IS NOT NULL

<b>EQUIPOS_V_ID</b>	El identificador que vende al jugador. Se puede dar el caso que esto sea nulo, eso significa que el jugador no tenía ningún equipo.		
NUMBER	EQUIPOS		
<b>MONEDAS_S_ID</b>	El identificador de la moneda que se utiliza para el salario del jugador.		
VARCHAR2(3)	MONEDAS		MONEDAS_S_ID IS NOT NULL
<b>MONEDAS_T_ID</b>	El identificador de la moneda que se ha utilizado para el traspaso del jugador.		
VARCHAR2(3)	MONEDAS		
<b>F_INICIO</b>	Fecha de inicio del contrato.		
DATE			F_INICIO IS NOT NULL, F_FIN > F_INICIO
<b>F_FIN</b>	Fecha que se da de baja el contrato. Esta fecha puede variar y acortarse por alguna razón.		
DATE			F_FIN IS NOT NULL, F_FIN > F_INICIO, F_FIN <= F_FINPREVISTA
<b>F_FINPREVISTA</b>	Fecha que ponía en el contrato de baja del mismo.		
DATE			F_FINPREVISTA IS NOT NULL, F_FIN <= F_FINPREVISTA
<b>SALARIO_BRUTO_ANUAL</b>	El salario bruto anual que cobra dicho jugador.		
NUMBER(20)			
<b>TRASPASO</b>	El traspaso que ha pagado el equipo comprador al vendedor.		
NUMBER(20)			
<b>SALARIO_BRUTO_EUROS</b>	Es el salario bruto anual en euros, campo generado.		
NUMBER			
<b>TRASPASO_EUROS</b>	Es el traspaso en euros, es un campo generado.		
NUMBER			

TABLA	DESCRIPCIÓN TABLA		
<i>DIAGNOSTICOS</i>	Se definen todos los tipos de técnicos que puedan registrar.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>DIAGNOSTICOS_ID</b>	Identificador del tipo de técnico.		
NUMBER	X		DIAGNOSTICOS_ID IS NOT NULL
<b>NOMBRE</b>	Nombre del tipo de técnico.		
VARCHAR2(250)			NOMBRE IS NOT NULL
<b>DIAS_ESTIMADOS</b>	Numero de días estimados de baja del diagnóstico.		
NUMBER(5)			DIAS_ESTIMADOS IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<i>EQUIPOS</i>	Se guarda información sobre los equipos.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>EQUIPOS_ID</b>	Identificador del equipo.		
NUMBER	X		EQUIPOS_ID IS NOT NULL
<b>CLUBS_ID</b>	Club o Sociedad al que pertenece el equipo.		
NUMBER		CLUBS	CLUBS_ID IS NOT NULL
<b>PABELLON_ID</b>	Pabellón donde normalmente juegan los partidos.		
NUMBER		PABELLON	PABELLON_ID IS NOT NULL
<b>NOMBRE</b>	Nombre del equipo, no se puede repetir un mismo nombre para un mismo club.		
VARCHAR2(200)			NOMBRE IS NOT NULL
<b>F_ALTA</b>	Fecha de alta del equipo.		
DATE			F_ALTA IS NOT NULL
<b>F_BAJA</b>	Fecha en la que se dio de baja.		
DATE			

TABLA	DESCRIPCIÓN TABLA		
<i>EQUIPOS_TEMPORADAS</i>	Se definen todos los equipos de una temporada.		
COLUMNA	DESCRIPCIÓN COLUMNAS		

TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>EQUIPOS_ID</b>	El identificador del equipo.		
NUMBER	X	EQUIPOS	EQUIPOS_ID IS NOT NULL
<b>COMPETICIONES_ID</b>	El identificador de la competición.		
NUMBER	X	TEMPORADAS	COMPETICIONES_ID IS NOT NULL
<b>F_INICIO</b>	La fecha de inicio de la competición.		
DATE	X	TEMPORADAS	F_INICIO IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<b>EQUIPO_TECNICO</b>	Se guardan todos los técnicos y su cargo para un equipo.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>TECNICOS_ID</b>	El identificador del técnico.		
NUMBER	X	TECNICOS	TECNICOS_ID IS NOT NULL
<b>TIPO_TECNICOS_ID</b>	El identificador del cargo que realizara el técnico.		
VARCHAR2(3)		TIPO_TECNICOS	TIPO_TECNICOS_ID IS NOT NULL
<b>EQUIPOS_ID</b>	El identificador del equipo en el que desarrollara dicho cargo.		
NUMBER	X	EQUIPOS	EQUIPOS_ID IS NOT NULL
<b>F_INICIO</b>	La fecha de inicio del cargo.		
DATE	X		F_INICIO IS NOT NULL, F_FIN > F_INICIO
<b>F_FIN</b>	La fecha de baja del cargo.		
DATE			F_FIN IS NOT NULL, F_FIN > F_INICIO

TABLA	DESCRIPCIÓN TABLA		
<b>GENEROS</b>	Están definidos los géneros de competición que existe, normalmente son masculino o femenino pero se puede llegar a dar el caso que se cree uno nuevo que fuera mixto.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>GENEROS_ID</b>	Identificador del género.		
VARCHAR2(1)	X		GENEROS_ID IS NOT NULL
<b>NOMBRE</b>	Es el nombre del género.		
VARCHAR2(20)			NOMBRE IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<b>JUGADORES</b>	Se guarda información referente al jugador de baloncesto.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>PERSONES_ID</b>	Identificador de la persona que es jugador.		
NUMBER	X	PERSONES	PERSONES_ID IS NOT NULL
<b>JUG_POSICIONES_ID</b>	Identificador de la posición donde juega el jugador.		
NUMBER(1)		JUG_POSICIONES	JUG_POSICIONES_ID IS NOT NULL
<b>NACIONALIDAD_ID</b>	Se guarda el identificador de país del que es nacionalizado.		
NUMBER		PAISES	NACIONALIDAD_ID IS NOT NULL
<b>FEDERACION_ID</b>	Se guarda el identificador al país en el que se ha federado.		
NUMBER		PAISES	FEDERACION_ID IS NOT NULL
<b>ACTIVO</b>	Se guarda una S si esta en activo y una N si no lo está en el caso que se N significa que está retirado.		
VARCHAR2(1)			ACTIVO IS NOT NULL, ACTIVO IN ('S','N')
<b>ALTURA</b>	Se guarda la altura en metros		
NUMBER(3,2)			ALTURA IS NOT NULL
<b>NUM_FEDE</b>	Se guarda el numero federado que tiene		
VARCHAR2(10)			NUM_FEDE IS NOT NULL
<b>PESO</b>	Se guarda el peso que tiene en Quilogramos		
NUMBER(3)			PESO IS NOT NULL
<b>URL_WEB</b>	Se guarda la url de la web promocional del jugador, no es obligatorio.		
VARCHAR2(200)			
<b>URL_VIDEO</b>	Se guarda la url del video promocional del jugador, no es obligatorio.		
VARCHAR2(200)			

TABLA	DESCRIPCIÓN TABLA		
JUG_BAJA	Guarda las bajas de un jugador.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
JUG_BAJA_ID	Es una secuencia que se genera para cada baja de un jugador: SEQ_JUG_BAJA.		
NUMBER	X		JUG_BAJA_ID IS NOT NULL
JUGADOR_ID	El identificador del jugador que está de baja.		
NUMBER		JUGADORES	JUGADOR_ID IS NOT NULL
TIPOS_BAJA_ID	El tipo de baja que tiene el jugador.		
VARCHAR2(1)		TIPOS_BAJA	TIPOS_BAJA_ID IS NOT NULL
F_INICIO	Fecha de inicio de la baja.		
DATE			F_INICIO IS NOT NULL
F_FIN	Fecha de fin y a partir del día siguiente ya está de alta.		
DATE			
OBSERVACIONES	Campo descriptivo para escribir cualquier observación de la baja.		
VARCHAR2(250)			OBSERVACIONES IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
JUG_POSICIONES	Se definen las posiciones de juego de los jugadores.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
JUG_POSICIONES_ID	Identificador de la posición.		
NUMBER(1)	X		JUG_POSICIONES_ID IS NOT NULL
NOMBRE	Nombre de la posición de juego.		
VARCHAR2(20)			NOMBRE IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
JUG_REPRESENTANTES	Se guarda los contratos de los jugadores con los responsables.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
JUGADOR_ID	El identificador del jugador que realiza el contrato.		
NUMBER	X	JUGADORES	JUGADOR_ID IS NOT NULL
REPRESENTANTE_ID	El identificado del representante que tendrá que representar al jugador.		
NUMBER	X	REPRESENTANTES	REPRESENTANTE_ID IS NOT NULL
F_ALTA	La fecha de alta del contrato.		
DATE	X		F_ALTA IS NOT NULL, F_BAJA > F_ALTA
F_BAJA	La fecha de baja del contrato. Esta fecha se puede variar siempre que se anule antes el contrato, por cualquier circunstancia.		
DATE			F_BAJA IS NOT NULL, F_BAJA > F_ALTA
F_PREVISTA	Esta es la fecha de baja del contrato que pone en el contrato que se firmó.		
DATE			F_PREVISTA IS NOT NULL, F_BAJA <= F_PREVISTA

TABLA	DESCRIPCIÓN TABLA		
LOGS	Es un registro de todas las funciones y procedimientos que se han ejecutado en el sistema.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
LOGS_ID	Es el identificador de logs, es la secuencia SEQ_LOGS.		
NUMBER	X		LOGS_ID IS NOT NULL
F_EJECUCION	La fecha y hora de ejecución del procedimiento o función.		
DATE			F_EJECUCION IS NOT NULL
PARAM_ENTRADA	Los parámetros de entrada.		
VARCHAR2(4000)			
PARAM_SALIDA	Los parámetros de salida que retorna.		
VARCHAR2(4000)			

<b>PROCEDIMIENTO</b>	El nombre del procedimiento que ha ejecutado.		
VARCHAR2(200)			PROCEDIMIENTO IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<i>MODALIDADES</i>	Están definidos las modalidades existentes normal o silla de ruedas.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>MODALIDADES_ID</b>	Identificador de la modalidad.		
VARCHAR2(1)	X		MODALIDADES_ID IS NOT NULL
<b>NOMBRE</b>	Es el nombre de la modalidad.		
VARCHAR2(20)			NOMBRE IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<i>MONEDAS</i>	Se insertan los tipos de monedas con la que realizamos los contratos de los jugadores.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>MONEDAS_ID</b>	Identificador de la moneda.		
VARCHAR2(3)	X		MONEDAS_ID IS NOT NULL
<b>NOMBRE</b>	Nombre de la moneda.		
VARCHAR2(50)			NOMBRE IS NOT NULL
<b>EURO</b>	Valor en euros.		
NUMBER(10)			EURO IS NOT NULL
<b>EQUIVALENCIA</b>	Equivalencia de la moneda en base a los euros.		
NUMBER(20,10)			EQUIVALENCIA IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<i>MUNICIPIOS</i>	Se insertan todos los municipios que se necesitan.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>MUNICIPIOS_ID</b>	Identificador del municipio. Este es una secuencia: SEQ_MUNICIPIOS.		
NUMBER	X		MUNICIPIOS_ID IS NOT NULL
<b>PAISES_ID</b>	Identificador del país al que hace referencia el municipio.		
NUMBER		PAISES	PAISES_ID IS NOT NULL
<b>NOMBRE</b>	Es el nombre del municipio.		
VARCHAR2(50)			NOMBRE IS NOT NULL
<b>CPOSTAL</b>	Es el Código Postal del municipio o la parte fija del código postal.		
VARCHAR2(5)			CPOSTAL IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<i>PABELLON</i>	Se guarda la información referente a los pabellones donde se juegan los partidos.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>PABELLON_ID</b>	Identificador del pabellón. Este es una secuencia: SEQ_PABELLON.		
NUMBER	X		PABELLON_ID IS NOT NULL
<b>NOMBRE</b>	Nombre del pabellón, este será único.		
VARCHAR2(50)			NOMBRE IS NOT NULL
<b>MUNICIPIOS_ID</b>	El identificador del municipio donde está el pabellón.		
NUMBER		MUNICIPIOS	MUNICIPIOS_ID IS NOT NULL
<b>DIRECCION</b>	La dirección del pabellón.		
VARCHAR2(200)			DIRECCION IS NOT NULL
<b>CPOSTAL</b>	El código postal exacto de la dirección.		
VARCHAR2(5)			CPOSTAL IS NOT NULL
<b>CAPACIDAD</b>	El número de capacidad que tiene el pabellón.		
NUMBER(10)			

TABLA	DESCRIPCIÓN TABLA		
<i>PAISES</i>	Se insertan todos los países que tienen Competiciones o algún jugador nacionalizado o federado.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>PAISES_ID</b>	Identificador del país. Utiliza la secuencia: SEQ_PAISES.		

NUMBER	X		PAISES_ID IS NOT NULL
<b>CONTINENTES_ID</b>	Identificador del continente al que hace referencia el país.		
VARCHAR2(2)		CONTINENTES	CONTINENTES_ID IS NOT NULL
<b>NOMBRE</b>	Es el nombre del país.		
VARCHAR2(50)			NOMBRE IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<i>PARTIDOS</i>	Aquí se definen todos los partidos de todas las temporadas.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>PARTIDOS_ID</b>	El identificador del partido. Este utilizara la secuencia: SEQ_PARTIDOS.		
NUMBER	X		PARTIDOS_ID IS NOT NULL
<b>EQUIPOS_L_ID</b>	El identificador del equipo Local.		
NUMBER		EQUIPOS	EQUIPOS_L_ID IS NOT NULL
<b>EQUIPOS_V_ID</b>	El identificador del equipo Visitante.		
NUMBER			EQUIPOS_V_ID IS NOT NULL
<b>COMPETICIONES_ID</b>	El identificador de la competición.		
NUMBER		TEMPORADAS	COMPETICIONES_ID IS NOT NULL
<b>PABELLON_ID</b>	El identificador del pabellón donde se jugara el partido.		
NUMBER		PABELLON	PABELLON_ID IS NOT NULL
<b>F_INICIO</b>	La fecha de inicio de la temporada para esa competición.		
DATE		TEMPORADAS	F_INICIO IS NOT NULL
<b>DIA_HORA</b>	La fecha y hora del partido.		
DATE			DIA_HORA IS NOT NULL
<b>JORNADA</b>	El número de jornada en caso sea necesario.		
NUMBER(3)			
<b>PUNTOS_LOCA_L</b>	Los puntos final del partido del equipo local.		
NUMBER			
<b>PUNTOS_VISITANTE</b>	Los puntos final del partido del equipo visitante.		
NUMBER			

TABLA	DESCRIPCIÓN TABLA		
<i>PARTIDOS_ARBITROS</i>	Aquí se definen todos los árbitros designados para un partido y la función de cada uno de ellos.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>PARTIDOS_ID</b>	El identificador del partido.		
NUMBER	X	PARTIDOS	PARTIDOS_ID IS NOT NULL
<b>TIPO_ARBITROS_ID</b>	El tipo de árbitro que definirá la función a realizar.		
VARCHAR2(2)		TIPO_ARBITROS	TIPO_ARBITROS_ID IS NOT NULL
<b>ARBITROS_ID</b>	El identificador del árbitro designado.		
NUMBER	X	ARBITROS	ARBITROS_ID IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
<i>PARTIDOS_JUGADORES</i>	Aquí se insertan los jugadores convocados para el partido y sus datos para después realizar las estadísticas.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>PARTIDOS_ID</b>	El identificador del partido.		
NUMBER	X	PARTIDOS	PARTIDOS_ID IS NOT NULL
<b>JUGADORES_ID</b>	El identificador del jugador.		
NUMBER	X	JUGADORES	JUGADORES_ID IS NOT NULL
<b>EQUIPOS_ID</b>	Identificador del equipo. Si esta vacío un trigger recupera el equipo para ese partido del jugador (RET_EQUIPO_JUGADOR(PARTIDOS_ID,JUGADORES_ID)).		
NUMBER		EQUIPOS	
<b>ASISTENCIAS</b>	Número de asistencias del jugador en el partido.		
NUMBER(3)			
<b>FALTAS_COMETIDAS</b>	Faltas realizadas.		
NUMBER(1)			

<b>FALTAS_RECIBIDAS</b>	Faltas recibidas.		
NUMBER(3)			
<b>MINUTOS_JUGADOS</b>	Minutos del jugador que ha jugado.		
NUMBER(2)			
<b>PELOTAS_PERDIDAS</b>	Pelotas perdidas que ha realizado.		
NUMBER(3)			
<b>PELOTAS_RECUPERADAS</b>	Pelotas recuperadas que ha realizado.		
NUMBER(3)			
<b>PUNTOS</b>	Los puntos calculados que ha realizado el jugador sumando las canastas que ha hecho de 1,2 y 3 puntos.		
NUMBER			
<b>REBOTES_DEFENSIVOS</b>	Los rebotes defensivos.		
NUMBER(3)			
<b>REBOTES_OFENSIVOS</b>	Los rebotes ofensivos.		
NUMBER(3)			
<b>TAPONES_CONTRA</b>	Los taponos que le han realizado.		
NUMBER(3)			
<b>TAPONES_FAVOR</b>	Taponos que ha efectuado a otros jugadores.		
NUMBER(3)			
<b>TIROS1</b>	Tiros de 1 que ha efectuado.		
NUMBER(3)			
<b>TIROS1E</b>	Tiros de 1 que ha encestado.		
NUMBER(3)			TIROS1E<=TIROS1
<b>TIROS2</b>	Tiros de 2 que ha efectuado.		
NUMBER(3)			
<b>TIROS2E</b>	Tiros de 2 que ha encestado.		
NUMBER(3)			TIROS2E<=TIROS2
<b>TIROS3</b>	Tiros de 3 que ha efectuado.		
NUMBER(3)			
<b>TIROS3E</b>	Tiros de 3 que ha encestado.		
NUMBER(3)			TIROS3E<=TIROS3
<b>VALORACION_JUGADOR</b>	Es la valoración total del jugador lo que ha hecho bien menos lo que ha hecho mal.		
NUMBER			
<b>VALORACION_JUGADOR_PONDERADA</b>	Es la valoración total del jugador ponderada, lo realizara una función (RET_VALORACION_JUG_POND(PARTIDO NUMBER, VALORACION NUMBER)).		
NUMBER			

TABLA	DESCRIPCIÓN TABLA		
<i>PERSONES</i>	Se guarda información referente a una persona.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
<b>PERSONES_ID</b>	Identificador de la persona.		
NUMBER	X		PERSONES_ID IS NOT NULL
<b>DNI</b>	El DNI/Pasaporte de la persona.		
VARCHAR2(10)			DNI IS NOT NULL
<b>NOMBRE</b>	Nombre de la persona.		
VARCHAR2(50)			NOMBRE IS NOT NULL
<b>APELLIDOS</b>	Apellidos de la persona.		
VARCHAR2(100)			APELLIDOS IS NOT NULL
<b>SEXO</b>	Sexo de la persona. Tendrá dos valores H/M (Hombre/Mujer)		
VARCHAR2(1)			SEXO IS NOT NULL, SEXO IN ('H','M')
<b>F_NACIMIENTO</b>	Fecha de nacimiento de la persona		
DATE			F_NACIMIENTO IS NOT NULL

TABLA	DESCRIPCIÓN TABLA
-------	-------------------

<b>REPRESENTANTES</b>	Se guarda información referente a un representante.		
<b>COLUMNA</b>	<b>DESCRIPCIÓN COLUMNAS</b>		
<b>TIPO</b>	<b>PK</b>	<b>TABLA FORANEA</b>	<b>RESTRICCIONES</b>
<b>PERSONES_ID</b>	Identificador de la persona que es representante.		
<b>NUMBER</b>	X	PERSONES	PERSONES_ID IS NOT NULL
<b>TITULACIONES</b>	Campo descriptivo donde pone las titulaciones que tiene		
<b>VARCHAR2(250)</b>			TITULACIONES IS NOT NULL
<b>URL_WEB</b>	La web donde se puede promocionar el representante no es obligatoria.		
<b>VARCHAR2(200)</b>			
<b>URL_VIDEO</b>	El video donde se puede promocionar el representante no es obligatorio.		
<b>VARCHAR2(200)</b>			

<b>TABLA</b>	<b>DESCRIPCIÓN TABLA</b>		
<b>TECNICOS</b>	Se guarda la información referente a los pabellones donde se juegan los partidos.		
<b>COLUMNA</b>	<b>DESCRIPCIÓN COLUMNAS</b>		
<b>TIPO</b>	<b>PK</b>	<b>TABLA FORANEA</b>	<b>RESTRICCIONES</b>
<b>PERSONES_ID</b>	Identificador de la persona que es tiene una titulación para ejercer de técnico.		
<b>NUMBER</b>	X	PERSONES	PERSONES_ID IS NOT NULL
<b>TITULACIONES</b>	Nombre de la titulación.		
<b>VARCHAR2(250)</b>			TITULACIONES IS NOT NULL
<b>URL_WEB</b>	Web del técnico para publicitarse.		
<b>VARCHAR2(200)</b>			
<b>F_TITULACION_CARGO</b>	Fecha de la última titulación del cargo que esta ejerciendo o que quiere ejercer.		
<b>DATE</b>			F_TITULACION_CARGO IS NOT NULL

<b>TABLA</b>	<b>DESCRIPCIÓN TABLA</b>		
<b>TEMPORADAS</b>	Se definen las temporadas para cada competición.		
<b>COLUMNA</b>	<b>DESCRIPCIÓN COLUMNAS</b>		
<b>TIPO</b>	<b>PK</b>	<b>TABLA FORANEA</b>	<b>RESTRICCIONES</b>
<b>COMPETICIONES_ID</b>	El identificador de la competición.		
<b>NUMBER</b>	X	COMPETICIONES	COMPETICIONES_ID IS NOT NULL
<b>F_INICIO</b>	La fecha de inicio de la competición.		
<b>DATE</b>	X		F_INICIO IS NOT NULL, F_FIN > F_INICIO
<b>F_FIN</b>	La fecha de baja de la competición.		
<b>DATE</b>			F_FIN IS NOT NULL, F_FIN > F_INICIO
<b>NUM_JORNADAS</b>	En el caso que la competición sea de tipo liga se definirá el número de jornadas que tiene ida + vuelta.		
<b>NUMBER(3)</b>			

<b>TABLA</b>	<b>DESCRIPCIÓN TABLA</b>		
<b>TIPO_ARBITROS</b>	Se definen los tipos de árbitros que hay.		
<b>COLUMNA</b>	<b>DESCRIPCIÓN COLUMNAS</b>		
<b>TIPO</b>	<b>PK</b>	<b>TABLA FORANEA</b>	<b>RESTRICCIONES</b>
<b>TIPO_ARBITROS_ID</b>	Identificador de la moneda.		
<b>VARCHAR2(2)</b>	X		TIPO_ARBITROS_ID IS NOT NULL
<b>NOMBRE</b>	Nombre de la moneda.		
<b>VARCHAR2(50)</b>			NOMBRE IS NOT NULL

<b>TABLA</b>	<b>DESCRIPCIÓN TABLA</b>		
<b>TIPO_CLUBS</b>	Se insertan los tipos de clubs que existen, en estos momentos hay club, sociedad anónima, limitada, ...		
<b>COLUMNA</b>	<b>DESCRIPCIÓN COLUMNAS</b>		
<b>TIPO</b>	<b>PK</b>	<b>TABLA FORANEA</b>	<b>RESTRICCIONES</b>
<b>TIPO_CLUBS_ID</b>	Identificador del tipo de club.		
<b>VARCHAR2(2)</b>	X		TIPO_CLUBS_ID IS NOT NULL

<b>NOMBRE</b>	Nombre del tipo de club.		
VARCHAR2(20)			NOMBRE IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
TIPOS_BAJA	Se insertan los tipos de baja de un jugador.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
TIPOS_BAJA_ID	Identificador del tipo de baja de un jugador.		
VARCHAR2(1)	X		TIPOS_BAJA_ID IS NOT NULL
<b>NOMBRE</b>	Nombre del tipo de baja.		
VARCHAR2(50)			NOMBRE IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
TIPO_TECNICOS	Se definen todos los tipos de técnicos que puedan registrar.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
TIPO_TECNICOS_ID	Identificador del tipo de técnico.		
VARCHAR2(3)	X		TIPO_TECNICOS_ID IS NOT NULL
<b>NOMBRE</b>	Nombre del tipo de técnico.		
VARCHAR2(50)			NOMBRE IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
EST_CT_PUNTOS_EQ	Tabla estadística que guarda según la competición y la temporada los mejores equipos defensivos y ofensivos.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
COMPETICIONES_ID	El identificador de la competición.		
NUMBER	X	TEMPORADAS	COMPETICIONES_ID IS NOT NULL
F_INICIO	La fecha de inicio de la temporada.		
DATE	X	TEMPORADAS	F_INICIO IS NOT NULL
EQUIPOS_D_ID	El identificador al mejor equipo defensivo de la temporada y competición.		
NUMBER		EQUIPOS	EQUIPOS_D_ID IS NOT NULL
EQUIPOS_O_ID	El identificador al mejor equipo ofensivo de la temporada y competición.		
NUMBER		EQUIPOS	EQUIPOS_O_ID IS NOT NULL
PUNTOS_DEFENSIVOS	Total de puntos que le han marcado al mejor equipo defensor.		
NUMBER(10)			
PUNTOS_OFENSIVOS	Total de puntos realizados del mejor equipo ofensivo.		
NUMBER(10)			

TABLA	DESCRIPCIÓN TABLA		
EST_GM_MJUG_ANY	Para cada género, modalidad y año se guarda el mejor jugador del mundo. Para sacar el del último año se recupera el primer registro ordenado descendente por el campo anio, siempre que este no sea el año actual.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
GENEROS_ID	El identificador del género.		
VARCHAR2(1)	X	GENEROS	GENEROS_ID IS NOT NULL
MODALIDADES_ID	El identificador de la modalidad.		
VARCHAR2(1)	X	MODALIDADES	MODALIDADES_ID IS NOT NULL
JUGADORES_ID	El identificador del jugador mejor del mundo para ese año, modalidad y género.		
NUMBER		JUGADORES	JUGADORES_ID IS NOT NULL
ANIO	El año donde se identificara al mejor jugador.		
NUMBER(4)	X		ANIO IS NOT NULL
MED_VP	La media ponderada del jugador.		
NUMBER(10,3)			MED_VP IS NOT NULL

TABLA	DESCRIPCIÓN TABLA		
-------	-------------------	--	--

<i>EST_JUG_ACTIVOS</i>	Para cada género y modalidad se guarda en número total de jugadores activos.		
<b>COLUMNA</b>	<b>DESCRIPCIÓN COLUMNAS</b>		
<b>TIPO</b>	<b>PK</b>	<b>TABLA FORANEA</b>	<b>RESTRICCIONES</b>
<b>GENEROS_ID</b>	El identificador del género.		
VARCHAR2(1)	X	GENEROS	GENEROS_ID IS NOT NULL
<b>MODALIDADES_ID</b>	El identificador de la modalidad.		
VARCHAR2(1)	X	MODALIDADES	MODALIDADES_ID IS NOT NULL
<b>NUM_JUGADORES</b>	El número de jugadores activos.		
NUMBER			NUM_JUGADORES IS NOT NULL

<b>TABLA</b>	<b>DESCRIPCIÓN TABLA</b>		
<i>EST_JUG_MEJOR_PAGADO</i>	En cada género y modalidad tendremos al jugador que mejor esta pagado.		
<b>COLUMNA</b>	<b>DESCRIPCIÓN COLUMNAS</b>		
<b>TIPO</b>	<b>PK</b>	<b>TABLA FORANEA</b>	<b>RESTRICCIONES</b>
<b>GENEROS_ID</b>	El identificador del género.		
VARCHAR2(1)	X	GENEROS	GENEROS_ID IS NOT NULL
<b>MODALIDADES_ID</b>	El identificador de la modalidad.		
VARCHAR2(1)	X	MODALIDADES	MODALIDADES_ID IS NOT NULL
<b>JUGADORES_ID</b>	El identificador al jugador mejor pagado.		
NUMBER		JUGADORES	JUGADORES_ID IS NOT NULL
<b>SALARIO_BRUTO_ANUAL_EUROS</b>	Lo que cobra bruto anualmente el jugador.		
NUMBER			SALARIO_BRUTO_ANUAL_EUROS IS NOT NULL

<b>TABLA</b>	<b>DESCRIPCIÓN TABLA</b>		
<i>EST_PTGM_SALARIO_MEDIO_ANUAL</i>	Dado un país, un género, una modalidad y un año de inicio de las temporadas se guardará el salario medio anual de los jugadores.		
<b>COLUMNA</b>	<b>DESCRIPCIÓN COLUMNAS</b>		
<b>TIPO</b>	<b>PK</b>	<b>TABLA FORANEA</b>	<b>RESTRICCIONES</b>
<b>PAISES_ID</b>	El identificador a la tabla de países.		
NUMBER	X	PAISES	PAISES_ID IS NOT NULL
<b>GENEROS_ID</b>	El identificador a la tabla de géneros.		
VARCHAR2(1)	X	GENEROS	GENEROS_ID IS NOT NULL
<b>MODALIDADES_ID</b>	Identificador de la modalidad.		
VARCHAR2(1)	X	MODALIDADES	MODALIDADES_ID IS NOT NULL
<b>TEMPORADA_ANIO_INICI</b>	El año de inicio de la temporada.		
DATE			TEMPORADA_ANIO_INICI IS NOT NULL
<b>SALARIO_MEDIO_ANUAL</b>	El salario media anual para una temporada, modalidad, género y país.		
NUMBER			SALARIO_MEDIO_ANUAL IS NOT NULL

<b>TABLA</b>	<b>DESCRIPCIÓN TABLA</b>		
<i>EST_TEMP_MAX_ANOTADOR</i>	Dada una competición retorna al máximo anotador de la última temporada o la actual.		
<b>COLUMNA</b>	<b>DESCRIPCIÓN COLUMNAS</b>		
<b>TIPO</b>	<b>PK</b>	<b>TABLA FORANEA</b>	<b>RESTRICCIONES</b>
<b>COMPETICIONES_ID</b>	El identificador de la competición.		
NUMBER	X	TEMPORADAS	COMPETICIONES_ID IS NOT NULL
<b>F_INICIO</b>	La fecha de inicio de la temporada.		
DATE	X	TEMPORADAS	F_INICIO IS NOT NULL
<b>JUGADORES_ID</b>	Identificador al jugador que más puntos ha anotado.		
NUMBER		JUGADORES	JUGADORES_ID IS NOT NULL
<b>PUNTOS_ANOTADOS</b>	La cantidad de puntos que ha anotado dicho jugador.		

NUMBER			PUNTOS_ ANOTADOS IS NOT NULL
--------	--	--	------------------------------

TABLA	DESCRIPCIÓN TABLA		
EST_TOTAL_GANANCIAS	Se guardarán todas las ganancias que ha ganado cada jugador, si las ordenamos descendente por el total y recuperamos solo el primer registro, tenemos el que ha ganado más.		
COLUMNA	DESCRIPCIÓN COLUMNAS		
TIPO	PK	TABLA FORANEA	RESTRICCIONES
JUGADORES_ID	El identificador del jugador.		
NUMBER	X	JUGADORES	JUGADORES_ID IS NOT NULL
TOTAL_GANANCIAS	El total de ganancias que ese jugador ha ganado.		
NUMBER(20,3)			TOTAL_GANANCIAS IS NOT NULL

A continuación mostraré las tablas que utilizarán una secuencia para su clave primaria y como se llamará dicha secuencia y el trigger que se generará en la base de datos para su ejecución.

SECUENCIA	TABLA	COLUMNA REFERENCIA	TRIGGER
SEQ_CLUBS	CLUBS	CLUBSID	CLUBS_AUTONUMBER
SEQ_COMPETICIONES	COMPETICIONES	COMPETICIONESID	COMPETICIONES_AUTONUMBER
SEQ_CONTRATOS_JUGADORES	CONTRATOS_JUGADORES	CONTRATOS_JUGADORESID	CONTRATOS_JUGADORES_AUTONUMBER
SEQ_DIAGNOSTICOS	DIAGNOSTICOS	DIAGNOSTICOSID	DIAGNOSTICOS_AUTONUMBER
SEQ_EQUIPOS	EQUIPOS	EQUIPOSID	EQUIPOS_AUTONUMBER
SEQ_JUG_BAJA	JUG_BAJA	JUG_BAJAID	JUG_BAJA_AUTONUMBER
SEQ_LOGS	LOGS	LOGSID	LOGS_AUTONUMBER
SEQ_MUNICIPIOS	MUNICIPIOS	MUNICIPIOSID	MUNICIPIOS_AUTONUMBER
SEQ_PABELLON	PABELLON	PABELLONID	PABELLON_AUTONUMBER
SEQ_PAISES	PAISES	PAISESID	PAISES_AUTONUMBER
SEQ_PARTIDOS	PARTIDOS	PARTIDOSID	PARTIDOS_AUTONUMBER
SEQ_PERSONES	PERSONES	PERSONESID	PERSONES_AUTONUMBER

## 3. Implementación.

### 3.1. Instalación SGBD Oracle.

Este sistema lo instalare en una máquina virtual. Utilizare el programa VMware Workstation para instalar el sistema operativo Ubuntu server 13.04 64 bits. Una vez instalado realizare los siguientes pasos.

- Bajamos los siguientes paquetes que se utilizaran.
  - ~\$ sudo apt-get install openssh-server
  - ~\$ sudo apt-get install alien libaio1 unixodbc unzip
- Aumentamos el tamaño de swap.
  - ~\$ cat /proc/meminfo | grep -i swap
  - ~\$ sudo su -
  - ~\$ dd if=/dev/zero of=/home/swapfile bs=1024 count=1048576
  - ~\$ mkswap /home/swapfile
  - ~\$ swapon /home/swapfile
  - ~\$ swapon -a
  - ~\$ cp /etc/fstab /etc/fstab.backup\_`date +%N`
  - ~\$ echo '/home/swapfile swap swap defaults 0 0' >> /etc/fstab
  - ~\$ exit
  - ~\$ swapon -s
- Modificamos los parámetros del kernel
  - ~\$ sudo su -
  - ~\$ cat > /etc/sysctl.d/60-oracle.conf <<-EOF
  - # Oracle 11g XE kernel parameters
  - fs.file-max=6815744
  - net.ipv4.ip\_local\_port\_range=9000 65500
  - kernel.sem=250 32000 100 128
  - # kernel.shmmax=429496729
  - kernel.shmmax=107374183
  - EOF
  - ~\$ service procps start
  - ~\$ sudo sysctl -q fs.file-max
  - ~\$ sudo sysctl -q kernel.shmmax
  - ~\$ sudo sysctl -q net.ipv4.ip\_local\_port\_range
  - ~\$ sudo sysctl -q kernel.sem
- ORA-00845: MEMORY\_TARGET
  - ~\$ sudo su -
  - ~\$ cat > /etc/init.d/oracle-shm <<-EOF
  - #!/bin/sh
  - # /etc/init.d/oracle-shm
  - #
  - #
  - case "\$1" in
  - start)
  - echo "Starting script /etc/init.d/oracle-shm"
  - # Run only once at system startup
  - if [ -e /dev/shm/.oracle-shm ]; then
  - echo "/dev/shm is already mounted, nothing to do"
  - else
  - rm -f /dev/shm
  - mkdir /dev/shm
  - mount -t tmpfs shmfs -o size=2048m /dev/shm
  - touch /dev/shm/.oracle-shm
  - fi
  - ;;
  - stop)
  - echo "Stopping script /etc/init.d/oracle-shm"
  - echo "Nothing to do"

```

;;
*)
echo "Usage: /etc/init.d/oracle-shm {start|stop}"
exit 1
;;
esac
#
### BEGIN INIT INFO
# Provides:      oracle-shm
# Required-Start:  $remote_fs $syslog
# Required-Stop:  $remote_fs $syslog
# Default-Start:  2 3 4 5
# Default-Stop:   0 1 6
# Short-Description: Bind /run/shm to /dev/shm at system startup.
# Description:    Fix to allow Oracle 11g use AMM.
### END INIT INFO
EOF
~$ chmod 755 /etc/init.d/oracle-shm
~$ update-rc.d oracle-shm defaults 01 99
~$ reboot
~$ sudo cat /etc/mstab | grep shm
~$ sudo df -h /run/shm

```

- **Instalamos el Oracle 11gR2 Express Edition**

```

~$ unzip oracle-xe-11.2.0-1.0.x86_64.rpm.zip
~$ cd Disk1
~$ sudo alien --to-deb --scripts oracle-xe-11.2.0-1.0.x86_64.rpm
~$ sudo ln -s /usr/bin/awk /bin/awk
~$ sudo mkdir /var/lock/subsys
~$ sudo su -
~$ cat > /sbin/chkconfig <<-EOF
#!/bin/bash
# Oracle 11gR2 XE installer chkconfig hack for Debian based Linux (by dude)
# Only run once.
echo "Simulating /sbin/chkconfig..."
if [[ ! `tail -n1 /etc/init.d/oracle-xe | grep INIT` ]]; then
cat >> /etc/init.d/oracle-xe <<-EOM
#
### BEGIN INIT INFO
# Provides:      OracleXE
# Required-Start:  \\$remote_fs \\$syslog
# Required-Stop:  \\$remote_fs \\$syslog
# Default-Start:  2 3 4 5
# Default-Stop:   0 1 6
# Short-Description: Oracle 11g Express Edition
### END INIT INFO
EOM
fi
update-rc.d oracle-xe defaults 80 01
EOF
~$ exit
~$ sudo chmod 755 /sbin/chkconfig
~$ sudo dpkg --install ./oracle-xe_11.2.0-2_amd64.deb
~$ sudo /etc/init.d/oracle-xe configure
~$ sudo passwd oracle

```

- **Después de la instalación**

```

~$ su - oracle
~$ cp /etc/skel/.bash_logout ./
~$ cp /etc/skel/.bashrc ./
~$ cp /etc/skel/.profile ./
~$ echo "" >> ./profile

```

```

~$ echo '. /u01/app/oracle/product/11.2.0/xe/bin/oracle_env.sh' >>./profile
~$ su - oracle
~$ sqlplus / as sysdba
SQL> EXEC DBMS_XDB.SETLISTENERLOCALACCESS(FALSE);
exit

```

### 3.2. Configuración Oracle y herramientas.

Instalamos el cliente de Oracle para Windows. Para ello nos bajaremos el “instant client” de la web de Oracle (<http://www.oracle.com/technetwork/topics/winsoft-085727.html>). Sobre todo tenemos que tener en cuenta en bajarnos el de 32 bits, ya que el de 64 no funciona.

Necesitamos dos paquetes en concreto: “Instant Client Package – Basic” y el “Instant Client Package - SQL\*Plus”.

Una vez bajados, creamos manualmente las siguientes carpetas:

```

C:\oracle\client_32
C:\oracle\client_32\bin
C:\oracle\client_32\network
C:\oracle\client_32\network\admin

```

Descomprimos los ficheros que los paquetes que nos hemos descargado de la web de oracle en la carpeta C:\oracle\cliente\_32\bin. En la carpeta C:\oracle\cliente\_32\network\admin creamos los ficheros tnsnames.ora y sqlnet.ora con el siguiente contenido:

- Fichero tnsnames.ora:
 

```

XE =
(DESCRIPTION =
(AADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.14)(PORT = 1521))
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = XE)
)
)

```
- Fichero sqlnet.ora
 

```

SQLNET.AUTHENTICATION_SERVICES = (NTS)

```

Creamos las siguientes variables del sistema:

*pc -> propiedades -> configuración avanzada del sistema -> Variables de entorno*

```

ORACLE_HOME          c:\oracle\client_32
ORACLE_HOME_NAME     c:\oracle\client_32
TNS_ADMIN            c:\oracle\client_32\network\admin
SQL_PATH             c:\oracle\client_32
LD_LIBRARY_PATH      c:\oracle\client_32\bin

```

Además en la de PATH como ya estará creado hay que añadir este valor:  
C:\oracle\client\_32\bin o %LD\_LIBRARY\_PATH%

A partir de ahí podemos instalar tanto el golden32, toad for Oracle o el sqldeveloper. Para realizar las sentencias pl/sql.

### 3.3. Implementación Diseño E/R.

Para realizar la implementación en el SGBD tendremos que crear los tablespaces donde se guardaran todos los datos y el usuario donde se generara toda la estructura. Para ello generaremos 2 tablespaces, uno para datos y otro para índices. Estos se realizaran en los siguientes ficheros: TABLESPACES (tablespace.sql) y USUARIO (user.sql).

#### 3.3.1. Diseño Tablas.

Para la creación de las tablas antes tengo que crear unas funciones deterministic que serán campos virtuales en algunas tablas, como aun no tengo las tablas generadas solo las creo y devuelvo un valor fijo. Una vez creadas las tablas generare el script con el código bien. Estos scripts están en estos dos ficheros: functions\_init.sql (inicialización funciones), tablas.sql (generación de las tablas) y functions.sql (implementación funciones).

#### 3.3.2. Diseño de índices y disparadores.

Descripción de los índices que son únicos.

TABLA	NOMBRE INDICE	DESCRIPCIÓN
PERSONES	UK_PERSONES_DNI	El DNI es único en la tabla de personas. (DNI)
PARTIDOS	UK_PARTIDOS_L	Un equipo local puede jugar un partido un día a un mismo tiempo (EQUIPOS_L_ID, DIA_HORA).
PARTIDOS	UK_PARTIDOS_V	Un equipo visitante puede jugar un partido un día a un mismo tiempo (EQUIPOS_L_ID, DIA_HORA).
PABELLON	UK_PABELLON_NOMBRE	El nombre del pabellón es único, no puede existir otro pabellón que se llame igual. (NOMBRE)
JUG_BAJA	UK_JUG_BAJA_JUG_TIP_INICIO	Una baja de un jugador lo define el id del jugador y la fecha de inicio de la baja (JUGADOR_ID, F_INICIO). No puede existir otra baja de ese jugador en la misma fecha de inicio.
JUGADORES	UK_JUGADORES_NUM_FEDE	Un jugador solo puede tener un número único de federación del país donde esta federado. (FEDERACION_ID, NUM_FEDE)
EQUIPOS	UK_EQUIPOS_NOMBRE	El nombre de un equipo para un club debe de ser único. (NOMBRE, CLUBS_ID)
CONTRATOS_JUGADORES	UK_CONTRATOS_JUGADORES	Un jugador solo puede tener un contrato el mismo día de inicio. (JUGADORES_ID, F_INICIO)
COMPETICIONES	UK_COMPETICIONES_NOMBRE	El nombre de una competición para un país debe de ser único. (NOMBRE, PAISES_ID)
CLUBS	UK_CLUBS_NOMBRE	El nombre de un club para un municipio debe de ser único. (NOMBRE, MUNICIPIOS_ID)

Se generará todos los índices y en el apartado de disparadores se crearán aquellos que necesitan de una secuencia para insertar un nuevo registro. Esto se realizará en los siguientes ficheros: index.sql (índices), triggers.sql (disparadores) y seq.sql (secuencias).

### 3.4. Implementación de funcionalidades.

#### 3.4.1. Implementación procedimientos ABM.

Todos los procedimientos estarán dentro de la carpeta asociada a la que pertenecen. Tendremos dos carpetas: la de los procedimientos de ABM que se llamara "procedimientos" y la de procedimientos de consulta que se llamara "procedimientos\_consulta". Cada procedimiento estará asociado a un fichero sql.

#### Procedimientos ABM

<i>Alta Jugador: p_alta_jugador.</i>
Da de alta un jugador. En caso que el jugador exista como persona se le modificaran los datos no nulos de la tabla personas. En caso que no exista la persona dará un error en caso que intente insertar un valor en un campo que no permita nulos.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"><li>• pdni: varchar2, dni de la persona.</li><li>• pnombre: varchar2, nombre de la persona.</li><li>• papellidos: varchar2, apellidos de la persona.</li><li>• psexo: varchar2, sexo de la persona (H/M).</li><li>• pf_nacimiento: date, fecha de nacimiento de la persona.</li><li>• pid_pos_jugador: number, id de la posición en la que juega el jugador.</li><li>• pid_nacionalidad: number, id del país donde tiene la nacionalidad el jugador.</li><li>• pid_federacion: number, id del país donde esta federado el jugador.</li><li>• paltura: number, altura del jugador en cm.</li><li>• pnum_fede: varchar2, es un código que define el numero federado del jugador.</li><li>• ppeso: number, peso del jugador.</li><li>• purl_web: varchar2, url de la web del jugador.</li><li>• purl_video: varchar2, url del video promocional del jugador.</li></ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"><li>• rsp: varchar2.</li></ul>
<b>Precondición</b>
<ul style="list-style-type: none"><li>• El pid_pos_jugador tiene que existir en la tabla de JUG_POSICIONES.</li><li>• El pid_nacionalidad tiene que existir en la tabla de PAISES.</li><li>• El pid_federacion tiene que existir en la tabla de PAISES.</li><li>• El dni debe de ser correcto y tener algo no se puede pasar uno vacío.</li><li>• Todos los campos varchar2 y number deben de tener menos del tamaño máximo definido en la base de datos.</li><li>• La fecha debe de ser una fecha correcta y no nula.</li></ul>
<b>Post-condición</b>
<ul style="list-style-type: none"><li>• Creado nuevo jugador con el estado en activo.</li></ul>
<b>Retorna</b>
<ul style="list-style-type: none"><li>• "OK": el procedimiento ha sido un éxito.</li><li>• "ERROR: JUGADOR YA EXISTE": Dicho jugador ya está dado de alta, ya existe una persona con ese DNI.</li><li>• "ERROR: NUM FEDERACIO YA EXISTE": El núm. federación ya está dado de alta.</li><li>• "ERROR: SIN ALTURA": El campo de altura no puede ser nulo.</li><li>• "ERROR: SIN PESO": El campo del peso no puede ser nulo.</li><li>• "ERROR: SIN NOMBRE": El campo del nombre no puede ser nulo.</li><li>• "ERROR: SIN APELLIDOS": El campo del apellidos no puede ser nulo.</li><li>• "ERROR: SIN SEXO": El campo del sexo no puede ser nulo.</li><li>• "ERROR: SIN NUM FEDERACIO": El campo núm. federación no puede ser nulo.</li><li>• "ERROR: [SQLERRM]": El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li></ul>
<i>Modificar Jugador: p_modificar_jugador</i>
Modifica los datos de un jugador que exista. Solo modificara aquellos datos que el parámetro no sea nulo.
<b>Parámetros de entrada</b>

<ul style="list-style-type: none"> <li>• pdni: varchar2, dni de la persona.</li> <li>• pnombre: varchar2, nombre de la persona.</li> <li>• papellidos: varchar2, apellidos de la persona.</li> <li>• psexo: varchar2, sexo de la persona (H/M).</li> <li>• pf_nacimiento: date, fecha de nacimiento de la persona.</li> <li>• pid_pos_jugador: number, id de la posición en la que juega el jugador.</li> <li>• pid_nacionalidad: number, id del país donde tiene la nacionalidad el jugador.</li> <li>• pid_federacion: number, id del país donde esta federado el jugador.</li> <li>• paltura: number, altura del jugador en cm.</li> <li>• pnum_fede: varchar2, es un código que define el numero federado del jugador.</li> <li>• ppeso: number, peso del jugador.</li> <li>• purl_web: varchar2, url de la web del jugador.</li> <li>• purl_video: varchar2, url del video promocional del jugador.</li> <li>• pactivo: varchar2, define si el usuario está activo o no lo esta (S/N).</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>• rsp: varchar2.</li> </ul>
<b>Precondición</b>
<ul style="list-style-type: none"> <li>• El pid_pos_jugador tiene que existir en la tabla de JUG_POSICIONES.</li> <li>• El pid_nacionalidad tiene que existir en la tabla de PAISES.</li> <li>• El pid_federacion tiene que existir en la tabla de PAISES.</li> <li>• El dni debe de ser correcto y tener algo no se puede pasar uno vacío.</li> <li>• Todos los campos varchar2 y number deben de tener menos del tamaño máximo definido en la base de datos.</li> <li>• La fecha debe de ser una fecha correcta y no nula.</li> <li>• El pactivo debe tener uno de estos dos valores 'S' o 'N'.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>• Modifica los datos del jugador con los datos recibidos.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>• "OK": el procedimiento ha sido un existo.</li> <li>• "ERROR: JUGADOR NO EXISTE": Dicho jugador no está dado de alta.</li> <li>• "ERROR: NUM FEDERACIO YA EXISTE": El núm. federación ya está dado de alta con otro jugador.</li> <li>• "ERROR: [SQLERRM]": El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Modificar Jugador Asignación Representante: p_modjug_repre</i>
Da de alta o modifica los datos de un representante durante una fecha en la que representara a un jugador. Durante ese intervalo de fecha no podrá tener otro representante.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>• pdni_jugador: varchar2, dni de la persona a asignar un responsable</li> <li>• pdni_repre: varchar2, dni del responsable.</li> <li>• pnombre: varchar2, nombre del responsable.</li> <li>• papellidos: varchar2, apellidos del responsable.</li> <li>• psexo: varchar2, sexo del responsable (H/M).</li> <li>• pf_nacimiento: date, fecha de nacimiento del responsable.</li> <li>• ptitulaciones: varchar2, las titulaciones del representante.</li> <li>• purl_web: varchar2, url de la web del representante.</li> <li>• purl_video: varchar2, url del video promocional del representante.</li> <li>• pf_alta: date, fecha de inicio del contrato con el responsable.</li> <li>• pf_baja: date, fecha de fin del contrato con el responsable.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>• rsp: varchar2.</li> </ul>
<b>Precondición</b>
<ul style="list-style-type: none"> <li>• El parámetro pdni_jugador tiene que existir en la tabla de JUGADORES.</li> <li>• Las fechas de alta y baja tienen el formato DD/MM/YYYY sin horas. Estas no deben de estar vacias.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>• El pdni_repre si no existe en la tabla de responsables lo dará de alta y lo asignara a dicho jugador. Si no existe como persona también lo creara.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>• "OK": el procedimiento ha sido un existo.</li> </ul>

- “ERROR: JUGADOR YA TIENE REPRESENTANTE ASIGNADO”: El jugador en esas fechas ya tiene asignado un representante.
- “ERROR: SIN NOMBRE”: El campo del nombre no puede ser nulo si tiene que crear la persona.
- “ERROR: SIN APELLIDOS”: El campo del apellidos no puede ser nulo si tiene que crear la persona.
- “ERROR: SIN SEXO”: El campo del sexo no puede ser nulo si tiene que crear la persona.
- “ERROR: FECHA BAJA CONTRATO DEBE SER POSTERIOR AL ALTA”: La fecha de baja de contrato debe de ser posterior a la fecha de inicio del contrato
- “ERROR: REPRESENTANTE SIN TITULACION”: No se puede insertar un representante sin ninguna titulación.
- “ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.

### *Modificar Jugador Baja Representante: p\_modjug\_bajarepre*

Este procedimiento cambia la fecha de baja de un contrato que une al jugador con el representante. Se utiliza para ampliar o reducir el contrato.

#### **Parámetros de entrada**

- pdni\_jugador: varchar2, dni de la persona a asignar un responsable
- pdni\_repre: varchar2, dni del responsable.
- pf\_alta: date, fecha de inicio del contrato con el responsable.
- pf\_baja: date, fecha de fin del contrato con el responsable.

#### **Parámetros de salida**

- rsp: varchar2.

#### **Precondición**

- El contrato debe de existir, debe de haber un registro en la tabla JUG\_REPRESENTANTES con el jugador, representante y fecha de inicio del contrato.
- Las fechas de inicio y fin tienen el formato DD/MM/YYYY sin horas y no pueden ser nulas.

#### **Post-condición**

- Modificación de la fecha de finalización del contrato.

#### **Retorna**

- “OK”: el procedimiento ha sido un éxito.
- “ERROR: FECHA BAJA CONTRATO DEBE SER POSTERIOR AL ALTA”: La fecha de baja de contrato debe de ser posterior a la fecha de inicio del contrato.
- “ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.

### *Modificar Jugador nueva baja: p\_modjug\_nuevabaja*

Asigna una baja a un jugador. Dicha no debe de existir.

#### **Parámetros de entrada**

- pdni: varchar2, dni del jugador que está de baja.
- ptipos\_baja\_id: varchar2, identificador del tipo de baja que tiene el jugador.
- pdiagnosticos\_id: number, identificador del diagnóstico de baja que tiene el jugador. Este campo es nulo en el caso que la baja sea personal.
- pobservaciones: varchar2, es un campo descriptivo para apuntar alguna observación de la baja.
- pf\_inicio: date, fecha de inicio de la baja.
- pf\_fin: date, fecha de fin de la baja. En caso que sea nula y tenga un diagnostico se le sumara a la fecha de inicio los días estimados de la baja.

#### **Parámetros de salida**

- rsp: varchar2.

#### **Precondición**

- El pdni es el dni de un jugador. Este debe estar activo.
- El ptipos\_baja\_id tiene que existir en la tabla TIPOS\_BAJA.
- En caso de no ser null pdiagnosticos\_id debe de existir en la tabla DIAGNOSTICOS.
- Las fechas de inicio y fin tienen el formato DD/MM/YYYY sin horas. La fecha de inicio es obligatoria.

#### **Post-condición**

- Se crea una baja para un jugador.

#### **Retorna**

- “OK”: el procedimiento ha sido un éxito.
- “ERROR: BAJA MEDICA SIN DIAGNOSTICO”: No puede haber una baja médica sin diagnóstico.
- “ERROR: BAJA PERSONAL NO DEBE TENER DIAGNOSTICO”: No puede haber una baja personal con diagnóstico.

- “ERROR: FECHA INICIO BAJA DEBE DE SER ANTERIOR FIN BAJA”: La fecha de inicio de la baja debe de ser inferior a la fecha final de la baja.
- “ERROR: YA EXISTE UNA BAJA EN ESE PERIODO”: No puede haber más de una baja al mismo tiempo.
- “ERROR: OBSERVACIONES NO NULO”: El campo de observaciones no puede ser nulo.
- “ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.

*Modificar Jugador añade diagnóstico a baja: p\_modjug\_adddiagnostico*

Añade un diagnóstico a una baja ya creada.

**Parámetros de entrada**

- pdni: varchar2, dni del jugador que está de baja.
- pdiagnosticos\_id: number, identificador del diagnóstico de baja que tiene el jugador. Este campo es nulo en el caso que la baja sea personal.
- pf\_inicio: date, fecha de inicio de la baja.
- pf\_fin: date, fecha de fin de la baja. En caso que sea nula y se le sumara a la fecha de inicio los días estimados del diagnóstico/os.

**Parámetros de salida**

rsp: varchar2.

**Precondición**

- El pdni es el dni de un jugador. Este debe estar activo.
- En caso de no ser null pdiagnosticos\_id debe de existir en la tabla DIAGNOSTICOS.
- Las fechas de inicio y fin tienen el formato DD/MM/YYYY sin horas. La fecha de inicio es obligatoria.

**Post-condición**

- Se añade un diagnóstico a una baja médica.

**Retorna**

- “OK”: el procedimiento ha sido un éxito.
- “ERROR: NO EXISTE BAJA”: No existe una baja para añadir el diagnóstico.
- “ERROR: FECHA INICIO BAJA DEBE DE SER ANTERIOR FIN BAJA”: La fecha de inicio de la baja debe de ser inferior a la fecha final de la baja.
- “ERROR: YA EXISTE UNA BAJA EN ESE PERIODO”: No puede haber más de una baja al mismo tiempo.
- “ERROR: DIAGNOSTICO YA EXISTE”: ya existe ese diagnóstico para esa baja.
- “ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.

*Modificar Jugador elimina diagnóstico a baja: p\_modjug\_deldiagnostico*

Elimina un diagnóstico de una baja. Una baja médica no podrá quedarse sin un diagnóstico, en ese caso no lo podrá eliminar.

**Parámetros de entrada**

- pdni: varchar2, dni del jugador que está de baja. pdiagnosticos\_id: number, identificador del diagnóstico de baja que tiene el jugador. Este campo es nulo en el caso que la baja sea personal.
- pf\_inicio: date, fecha de inicio de la baja.
- pf\_fin: date, fecha de fin de la baja. En caso que sea nula y se le sumara a la fecha de inicio los días estimados del diagnóstico/os que hayan.

**Parámetros de salida**

- rsp: varchar2.

**Precondición**

- El pdni es el dni de un jugador. Este debe estar activo.
- En caso de no ser null pdiagnosticos\_id debe de existir en la tabla DIAGNOSTICOS.
- Las fechas de inicio y fin tienen el formato DD/MM/YYYY sin horas. La fecha de inicio es obligatoria.
- Debe de existir un diagnóstico para esa baja para poderlo eliminar.

**Post-condición**

- Se elimina el diagnóstico, si existe como mínimo uno más para esa baja. En el caso que se elimine si no se pasa fecha fin se recalculara con las estimadas entre todos los diagnósticos de esa baja.

**Retorna**

- “OK”: el procedimiento ha sido un éxito.
- “ERROR: FECHA INICIO BAJA DEBE DE SER ANTERIOR FIN BAJA”: La fecha de inicio de la baja debe de ser inferior a la fecha final de la baja.

- “ERROR: SE QUEDA BAJA SIN DIAGNOSTICO”: Esto ocurre en el caso que para esa baja solo exista ese diagnóstico.
- “ERROR: DIAGNOSTICO YA EXISTE”: ya existe ese diagnóstico para esa baja.
- “ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.

#### *Modificar Jugador modificar baja: p\_modjug\_modificarbaja*

Se modificará la fecha de fin de la baja y en caso necesario las observaciones.

##### **Parámetros de entrada**

- pdni: varchar2, dni del jugador que está de baja.
- ptipos\_baja\_id: varchar2, identificador del tipo de baja que tiene el jugador.
- pobservaciones: varchar2, es un campo descriptivo para apuntar alguna observación de la baja. Si es nulo no se modificar dicho campo.
- pf\_inicio: date, fecha de inicio de la baja.
- pf\_fin: date, fecha de fin de la baja. Si es nulo no se modificara dicho campo.

##### **Parámetros de salida**

- rsp: varchar2.

##### **Precondición**

- El pdni es el dni de un jugador. Este debe estar activo.
- Las fechas de inicio y fin tienen el formato DD/MM/YYYY sin horas. La fecha de inicio y bajas son obligatorias.
- Debe de existir esa baja para poderla modificar.

##### **Post-condición**

- Se realiza la modificación de la baja del jugador.

##### **Retorna**

- “OK”: el procedimiento ha sido un éxito.
- “ERROR: FECHA INICIO BAJA DEBE DE SER ANTERIOR FIN BAJA”: La fecha de inicio de la baja debe de ser inferior a la fecha final de la baja.
- “ERROR: DIAGNOSTICO YA EXISTE”: ya existe ese diagnóstico para esa baja.
- “ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.

#### *Baja jugador: p\_baja\_jugador*

Pone a un jugador con el estado de retirado.

##### **Parámetros de entrada**

- pdni: varchar2, dni del jugador que se quiere retirar.

##### **Parámetros de salida**

- rsp: varchar2.

##### **Precondición**

- El jugador tiene que estar activo y no tener ningún contrato vigente.

##### **Post-condición**

- El jugador se le pone en un estado de retirado, o mejor dicho activo='N'.

##### **Retorna**

- “OK”: el procedimiento ha sido un éxito.
- “ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.

#### *Baja jugador: p\_activo\_jugador*

Pone a un jugador con el estado de activo.

##### **Parámetros de entrada**

- pdni: varchar2, dni del jugador que se quiere poner en activo.

##### **Parámetros de salida**

- rsp: varchar2.

##### **Precondición**

- El jugador tiene que estar retirado.

<b>Post-condición</b>
<ul style="list-style-type: none"> <li>El jugador se le pone en un estado de activo, o mejor dicho activo='S'.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>"OK": el procedimiento ha sido un éxito.</li> <li>"ERROR: [SQLERRM]": El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Alta Contrato: p_alta_contrato</i>
Crea un contrato para un jugador que lo liga con un equipo durante un cierto tiempo. Si el contrato es por un traspaso de un equipo a otro, se cancelará a un día de antes del inicio del nuevo contrato, el contrato antiguo con él equipo que vende.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>pdni_jugador: varchar2, dni del jugador que hace el contrato.</li> <li>pdni_repre: varchar2, dni del representante del jugador que ha realizado el contrato.</li> <li>pequipos_c_id: number, identificador del equipo comprador.</li> <li>pequipos_v_id: number, identificador del equipo vendedor.</li> <li>pmonedas_s_id: varchar2, identificador de la moneda en la que está el salario del jugador.</li> <li>pmonedas_t_id: varchar2, identificador de la moneda en la que se hizo el traspaso.</li> <li>psalario_bruto_anual: number, salario bruto anual que cobra dicho jugador en la moneda que se ha seleccionado.</li> <li>ptraspaso: number, importe pagado por el traspaso del jugador cuando lo vende otro equipo.</li> <li>pf_inicio: date, fecha de inicio del contrato.</li> <li>pf_fin: date, fecha de fin del contrato.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>rsp: varchar2.</li> </ul>
<b>Precondición</b>
<ul style="list-style-type: none"> <li>El dni del jugador debe de existir en la tabla de JUGADORES y debe de estar en activo.</li> <li>El dni del representante debe de existir en la tabla de REPRESENTANTES.</li> <li>Los identificadores de las monedas deben de existir en la tabla de MONEDAS.</li> <li>Los identificadores de los equipos deben de existir en la tabla de EQUIPOS. Los equipos debén de estar asignados a una temporada mínimo.</li> <li>No puede haber un contrato posterior a la fecha de inicio del contrato que se esta dando de alta.</li> <li>No se puede dar de alta un contrato en la misma fecha de inicio de otro contrato anterior.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>El jugador tiene contrato con un equipo durante el tiempo que establece el contrato. En el caso que se firme un contrato antes que se haya finalizado el contrato con el equipo vendedor, se modificará la fecha de baja del contrato que realizo con dicho equipo.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>"OK": el procedimiento ha sido un éxito.</li> <li>"ERROR: ES OBLIGATORIO PONER EQUIPO COMPRADOR": Es obligatorio que en el contrato exista un equipo comprador.</li> <li>"ERROR: REPRESENTANTE NO TIENE CONTRATO CON JUGADOR": En esas fechas que se firma el contrato con el equipo dicho representante no está asignado al jugador.</li> <li>"ERROR: FECHA INICIO CONTRATO DEBE SER INFERIOR A FECHA FIN": La fecha de alta del contrato debe de ser menor que la fecha de fin del contrato.</li> <li>"ERROR: JUGADOR CON CONTRATO OTRO EQUIPO": Este caso se da cuando intente realizar un contrato a un jugador y el club que vende no tiene contrato con dicho jugador, o dicho jugador en esas fechas ya tiene contrato.</li> <li>"ERROR: NO EXISTE TRASPASO SIN EQUIPO VENDEDOR": No puede existir una cantidad de traspaso si no hay ningún equipo que venda.</li> <li>"ERROR: [SQLERRM]": El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Modificar Contrato: p_modificar_contrato</i>
Se modifica los datos de un contrato para que se pueda ampliarlo o reducirlo.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>pcontratos_id: number, identificador del contrato que se quiere modificar.</li> <li>pf_fin: date, fecha de fin del contrato.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>rsp: varchar2.</li> </ul>

<b>Precondición</b>
<ul style="list-style-type: none"> <li>El contrato debe de existir.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>Se modifica la fecha de finalización del contrato.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>“OK”: el procedimiento ha sido un éxito.</li> <li>“ERROR: EXISTE OTRO CONTRATO ENTRE ESAS FECHAS”: Entre esas fechas ya existe otro contrato.</li> <li>“ERROR: FECHA INICIO CONTRATO DEBE SER INFERIOR A FECHA FIN”: La fecha de alta del contrato debe de ser menor que la fecha de fin del contrato.</li> <li>“ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Alta Partido: p_alta_partido</i>
Da de alta un partido para una competición y temporada.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>pequipos_l_id: number, identificador del equipo local.</li> <li>pequipos_v_id: number, identificador del equipo visitante.</li> <li>pcompeticiones_id: number, identificador de la competición.</li> <li>pf_inicio: date, fecha de inicio de la temporada en que se juega el partido.</li> <li>pdia_hora: date, fecha y hora del partido.</li> <li>pjornada: number, en caso que la competición sea por jornadas el número de jornada que es.</li> <li>ppabellon_id: number, identificador del pabellón donde se juega el partido.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>rsp: varchar2.</li> </ul>
<b>Precondición</b>
<ul style="list-style-type: none"> <li>Los identificadores de los equipos deben de existir en la tabla de equipos. No se permite que sean nulos.</li> <li>La competición y la pf_inicio de la temporada deben de existir en la tabla de TEMPORADAS.</li> <li>El identificador del pabellón debe de existir en la tabla de PABELLON.</li> <li>El pdia_hora tendrá el siguiente formato DD/MM/YYYY HH24:MI.</li> <li>Los equipos deben estar inscritos en la temporada, tabla EQUIPOS_TEMPORADAS.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>Se genera un partido entre dos equipos para una temporada y competición.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>“OK”: el procedimiento ha sido un éxito.</li> <li>“ERROR: NO SE PUEDE JUGAR UN EQUIPO CONTRA SI MISMO”: No se puede poner el mismo equipo en el local y visitante.</li> <li>“ERROR: FALTA NUMERO JORNADA”: En caso que la competición lo requiera poner el número de jornada que es el partido.</li> <li>“ERROR: EQUIPO LOCAL YA TIENE PARTIDO ESE DIA”: Esto ocurre cuando el equipo local ya tiene partido para ese día y hora.</li> <li>“ERROR: EQUIPO VISITANTE YA TIENE PARTIDO ESE DIA”: Esto ocurre cuando el equipo visitante ya tiene partido para ese día y hora.</li> <li>“ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Modificación Partido: p_modif_partido</i>
Cambia la fecha o el pabellón donde se disputara el encuentro.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>ppartido_id: number, identificador del partido.</li> <li>pdia_hora: date, fecha y hora del partido.</li> <li>ppabellon_id: number, identificador del pabellón donde se juega el partido.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>rsp: varchar2.</li> </ul>
<b>Precondición</b>
<ul style="list-style-type: none"> <li>El identificador del partido debe de existir en la tabla PARTIDOS.</li> <li>El pdia_hora sino es nulo, tendrá el siguiente formato DD/MM/YYYY HH24:MI.</li> </ul>

<ul style="list-style-type: none"> <li>El pabellón si no es nulo, deber de existir en la tabla de PABELLON.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>Modifica la fecha o el pabellón de un partido siempre que no sea nulo.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>“OK”: el procedimiento ha sido un éxito.</li> <li>“ERROR: ALGUN EQUIPO YA TIENE PARTIDO ESE DIA”: Esto ocurre cuando uno de los dos equipos ya tiene partido para ese día y hora.</li> <li>“ERROR: [SQLERRM]”: El sqlerm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Modificación Partido asignación arbitro: p_asignarb_partido</i>
Asigna a un árbitro para realizar un partido.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>ppartido_id: number, identificador del partido.</li> <li>parbitro_id: number, identificador del árbitro.</li> <li>ptipo_arbitro_id: varchar2, identificador de las funciones que realizará en el partido.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>rsp: varchar2.</li> </ul>
<b>Precondición</b>
<ul style="list-style-type: none"> <li>Los tres campos deben de existir en las tablas relacionadas.</li> <li>El árbitro estará en activo, es decir que no estará dado de baja.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>Asignación de un árbitro para un partido.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>“OK”: el procedimiento ha sido un éxito.</li> <li>“ERROR: ARBITRO YA ASIGNADO”: ya está asignado a dicho partido.</li> <li>“ERROR: PARTIDO TIENE ASIGNADOS TODOS LOS ARBITROS”: A través de la competición del partido nos dice el número de árbitros que tiene que tener un partido y no puede exceder de ese número.</li> <li>“ERROR: ARBITRO YA HA SIDO DESIGNADO PARA ESE DIA”: Se controlara que un árbitro no pueda ser designado para arbitrar varios partidos el mismo día.</li> <li>“ERROR: [SQLERRM]”: El sqlerm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Modificación Partido desasignar arbitro: p_desasignarb_partido</i>
Quita la designación de un árbitro para un partido.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>ppartido_id: number, identificador del partido.</li> <li>parbitro_id: number, identificador del árbitro.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>rsp: varchar2.</li> </ul>
<b>Precondición</b>
<ul style="list-style-type: none"> <li>Debe de existir un árbitro asignado a ese partido.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>Se desasigna el árbitro al partido.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>“OK”: el procedimiento ha sido un éxito.</li> <li>“ERROR: [SQLERRM]”: El sqlerm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Modificación Partido insertar convocados equipo: p_convequi_partido</i>
Inserta los 12 jugadores como máximo convocados de un equipo para un partido. Si anteriormente se ha insertado otra convocatoria se eliminara y se modificar por esta. Si el equipo el club al que pertenece es de tipo ‘SE’ (Selección) no controlara si tiene contrato con ese club ya que las selecciones no tienen contratos con los jugadores. Dichos jugadores no pueden estar lesionados para la disputa del partido.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>ppartido_id: number, identificador del partido.</li> </ul>

<ul style="list-style-type: none"> <li>• p equipo_id: number, identificador del equipo que ha realizado la convocatoria de los jugadores.</li> <li>• p jugador_id1: number, identificador del jugador.</li> <li>• p jugador_id2: number, identificador del jugador.</li> <li>• p jugador_id3: number, identificador del jugador.</li> <li>• p jugador_id4: number, identificador del jugador.</li> <li>• p jugador_id5: number, identificador del jugador.</li> <li>• p jugador_id6: number, identificador del jugador.</li> <li>• p jugador_id7: number, identificador del jugador.</li> <li>• p jugador_id8: number, identificador del jugador.</li> <li>• p jugador_id9: number, identificador del jugador.</li> <li>• p jugador_id10: number, identificador del jugador.</li> <li>• p jugador_id11: number, identificador del jugador.</li> <li>• p jugador_id12: number, identificador del jugador.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>• rsp: varchar2.</li> </ul>
<b>Precondición</b>
<ul style="list-style-type: none"> <li>• El identificador del partido tiene que existir y el identificador del equipo debe de ser del equipo local o visitante.</li> <li>• Los 12 identificadores de jugadores pueden ser nulos pero en caso contrario deben de existir en la tabla de jugadores. No se puede repetir el identificador de un jugador.</li> <li>• El jugador no puede estar de baja cuando se celebra el partido ya sea media o personal.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>• Inserta la convocatoria para un partido.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>• "OK": el procedimiento ha sido un éxito.</li> <li>• "ERROR: PARTIDO CON ESTADISTICAS": Una convocatoria no se puede insertar si ya hay datos estadísticos para ese partido.</li> <li>• "ERROR: FALTAN JUGADORES, MINIMO 5": En la convocatoria no hay el mínimo de jugadores para disputar el partido que es de 5.</li> <li>• "ERROR: ALGUN JUGADOR NO PERTENECE AL EQUIPO": Esto ocurre cuando un jugador no tiene contrato el día que se celebra el partido.</li> <li>• "ERROR: [SQLERRM]": El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Modificación Partido insertar estadísticas jugador: p_estjug_partido</i>
Inserta o modifica los datos estadísticos de un jugador para un partido. Si un dato estadístico es nulo no lo cambiará.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>• p partido_id: number, identificador del partido.</li> <li>• p jugador_id1: number, identificador del jugador.</li> <li>• asistencias: number, número de asistencias.</li> <li>• pfaltas_cometidas: number, número de faltas cometidas.</li> <li>• pfaltas_recibidas: number, número de faltas recibidas.</li> <li>• p minutos_jugados: number, número de minutos jugados.</li> <li>• p pelotas_perdidas: number, número de pelotas perdidas.</li> <li>• p pelotas_recuperadas: number, número de pelotas recuperadas.</li> <li>• prebotes_defensivos: number, número de rebotes defensivos.</li> <li>• prebotes_ofensivos: number, número de rebotes ofensivos.</li> <li>• p tapones_contra: number, número de tapones que le han hecho.</li> <li>• p tapones_favor: number, número de tapones que ha hecho.</li> <li>• p tiros1: number, número de tiros de 1 que ha realizado.</li> <li>• p tiros1e: number, número de tiros de 1 que ha encestado.</li> <li>• p tiros2: number, número de tiros de 2 que ha realizado.</li> <li>• p tiros2e: number, número de tiros de 2 que ha encestado.</li> <li>• p tiros3: number, número de tiros de 3 que ha realizado.</li> <li>• p tiros3e: number, número de tiros de 3 que ha encestado.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>• rsp: varchar2.</li> </ul>
<b>Precondición</b>
<ul style="list-style-type: none"> <li>• Debe de existir un registro en la tabla PARTIDOS_JUGADORES con el identificador del partido y el jugador.</li> </ul>

<ul style="list-style-type: none"> <li>• Todos los datos estadísticos no permiten números negativos.</li> <li>• Todos los datos estadísticos serán acorde a las reglas del baloncesto, como por ejemplo que no puede poner más de 5 faltas realizadas, o el número de minutos de un partido no puede exceder de minutos jugados.</li> </ul>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>• Se ha insertado o modificado la estadística para ese partido y ese jugador.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>• “OK”: el procedimiento ha sido un éxito.</li> <li>• “ERROR: TIROS 1 ENCESTADO &gt; TIROS 1’”: Esto se produce cuando se meten más tiros de 1 que los que se han lanzado.</li> <li>• “ERROR: TIROS 2 ENCESTADO &gt; TIROS 2’”: Esto se produce cuando se meten más tiros de 2 que los que se han lanzado.</li> <li>• “ERROR: TIROS 3 ENCESTADO &gt; TIROS 3’”: Esto se produce cuando se meten más tiros de 3 que los que se han lanzado.</li> <li>• “ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

## Procedimientos de consulta.

<i>Procedimiento 1: p_jug_competicion</i>
Listado de todos los jugadores de una competición.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>• pcompeticion: number, identificador de la competición.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>• rsp: varchar2.</li> <li>• tabla_rsp: TB_DADES_JUGADORS.</li> </ul>
<b>Tipo Asociado</b>
<pre>drop type T_DADES_JUGADORS FORCE;  create or replace type T_DADES_JUGADORS as object(   PERSONES_ID      NUMBER,   DNI               VARCHAR2(10 BYTE),   NOMBRE            VARCHAR2(50 BYTE) ,   APELLIDOS         VARCHAR2(100 BYTE),   SEXO              VARCHAR2(1 BYTE),   F_NACIMIENTO     DATE,   JÜG_POSICIONES_ID NUMBER(1),   NACIONALIDAD_ID  NUMBER,   FEDERACION_ID    NUMBER,   ACTIVO            VARCHAR2(1 BYTE),   ALTURA           NUMBER(3,2),   NUM_FEDE          VARCHAR2(10 BYTE),   PESO              NUMBER(3),   URL_WEB           VARCHAR2(200 BYTE),   URL_VIDEO         VARCHAR2(200 BYTE),   DFIN_CONTRATO_ACT DATE );  DROP TYPE TB_DADES_JUGADORS;  CREATE OR REPLACE TYPE TB_DADES_JUGADORS as table of T_DADES_JUGADORS ;</pre>
<b>Precondición</b>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>• Se devuelve una tabla y el resultado del procedimiento.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>• “OK”: el procedimiento ha sido un éxito.</li> <li>• “ERROR: SIN DATOS”: El procedimiento no ha devuelto ningún dato.</li> <li>• “ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

### Procedimiento 2: p equip\_tpunts\_comp

Listado de todos los equipos de una competición ordenados por número de puntos totales a favor en la temporada actual.

#### Parámetros de entrada

- pcompeticion: number, identificador de la competición.

#### Parámetros de salida

- rsp: varchar2.
- tabla\_rsp: TB\_EQUIPOS\_PUNTOS

#### Tipo Asociado

```
drop type T_EQUIPOS_PUNTOS FORCE;
```

```
create or replace type T_EQUIPOS_PUNTOS as object(  
  EQUIPOS_ID  NUMBER,  
  PUNTOS      NUMBER  
);
```

```
DROP TYPE TB_EQUIPOS_PUNTOS;
```

```
CREATE OR REPLACE TYPE TB_EQUIPOS_PUNTOS as table of T_EQUIPOS_PUNTOS ;
```

#### Post-condición

- Se devuelve una tabla y el resultado del procedimiento.

#### Retorna

- "OK": el procedimiento ha sido un éxito.
- "ERROR: SIN DATOS": El procedimiento no ha devuelto ningún dato.
- "ERROR: [SQLERRM]": El sqlerrm tendrá el mensaje del error que no ha sido tratado.

### Procedimiento 3: p\_5mej\_jug\_posicion

Dado un ámbito, género y modalidad, muestra los 5 mejores jugadores por posición en función de su valoración ponderada.

#### Parámetros de entrada

- pambito varchar2, ámbito de la competición del jugador.
- pgenero varchar2, genero de la competición del jugador.
- pmodalidad varchar2, modalidad de la competición del jugador.

#### Parámetros de salida

- rsp: varchar2.
- tabla\_rsp: TB\_JUG\_POSICION.

#### Tipo Asociado

```
drop type T_JUG_POSICION FORCE;
```

```
create or replace type T_JUG_POSICION as object(  
  JUGADOR_ID  NUMBER,  
  JUG_POSICIONES_ID NUMBER(1)  
);
```

```
DROP TYPE TB_JUG_POSICION;
```

```
CREATE OR REPLACE TYPE TB_JUG_POSICION as table of T_JUG_POSICION ;
```

#### Post-condición

- Se devuelve una tabla y el resultado del procedimiento.

#### Retorna

- "OK": el procedimiento ha sido un éxito.
- "ERROR: SIN DATOS": El procedimiento no ha devuelto ningún dato.
- "ERROR: [SQLERRM]": El sqlerrm tendrá el mensaje del error que no ha sido tratado.

### Procedimiento 4: p\_contratos\_any\_repre

Dado un año y un representante, el número de contratos firmados de un jugador y el valor económico que supone para ese jugador.

#### Parámetros de entrada

<ul style="list-style-type: none"> <li>pany: varchar2, un año a consultar.</li> <li>prepresentante_id: number, identificador del representante.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>rsp: varchar2.</li> <li>tabla_rsp: TB_CONT_JUGA.</li> </ul>
<b>Tipo Asociado</b>
<pre>drop type T_CONT_JUGA FORCE;</pre> <pre>create or replace type T_CONT_JUGA as object(   JUGADOR_ID NUMBER,   NUM_CONTRATOS NUMBER,   VALOR_ECONOMICO NUMBER );</pre> <pre>DROP TYPE TB_CONT_JUGA;</pre> <pre>CREATE OR REPLACE TYPE TB_CONT_JUGA as table of T_CONT_JUGA ;</pre>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>Se devuelve una tabla y el resultado del procedimiento.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>“OK”: el procedimiento ha sido un éxito.</li> <li>“ERROR: SIN DATOS”: El procedimiento no ha devuelto ningún dato.</li> <li>“ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Procedimiento 5: p_10equip_gastadores</i>
Recopila los 10 equipos en un año determinado que más han gastado en la adquisición de jugadores.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>pany: varchar2, un año a consultar.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>rsp: varchar2.</li> <li>tabla_rsp: TB_EQUIP_TRASPASO.</li> </ul>
<b>Tipo Asociado</b>
<pre>drop type T_EQUIP_TRASPASO FORCE;</pre> <pre>create or replace type T_EQUIP_TRASPASO as object(   EQUIPOS_ID NUMBER,   TRASPASO NUMBER );</pre> <pre>DROP TYPE TB_EQUIP_TRASPASO;</pre> <pre>CREATE OR REPLACE TYPE TB_EQUIP_TRASPASO as table of T_EQUIP_TRASPASO ;</pre>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>Se devuelve una tabla y el resultado del procedimiento.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>“OK”: el procedimiento ha sido un éxito.</li> <li>“ERROR: SIN DATOS”: El procedimiento no ha devuelto ningún dato.</li> <li>“ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

<i>Procedimiento 6: p_jug_final_contrato</i>
Dado un país, un género y una modalidad, sacará un listado de todos los jugadores que acaban contrato o jugadores activos que en la temporada actual no tengan equipo.
<b>Parámetros de entrada</b>
<ul style="list-style-type: none"> <li>ppaises-id: number, identificador del país.</li> <li>pgenero varchar2, genero de la competición del jugador.</li> <li>pmodalidad varchar2, modalidad de la competición del jugador.</li> </ul>
<b>Parámetros de salida</b>
<ul style="list-style-type: none"> <li>rsp: varchar2.</li> </ul>

<ul style="list-style-type: none"> <li>tabla_rsp: TB_JUG_LIBRE.</li> </ul>
<b>Tipo Asociado</b>
<pre>drop type T_JUG_LIBRE FORCE;  create or replace type T_JUG_LIBRE as object(   JUGADORES_ID    NUMBER,   DFIN_CONTRATO DATE );  DROP TYPE TB_JUG_LIBRE;  CREATE OR REPLACE TYPE TB_JUG_LIBRE as table of T_JUG_LIBRE ;</pre>
<b>Post-condición</b>
<ul style="list-style-type: none"> <li>Se devuelve una tabla y el resultado del procedimiento.</li> </ul>
<b>Retorna</b>
<ul style="list-style-type: none"> <li>“OK”: el procedimiento ha sido un éxito.</li> <li>“ERROR: SIN DATOS”: El procedimiento no ha devuelto ningún dato.</li> <li>“ERROR: [SQLERRM]”: El sqlerrm tendrá el mensaje del error que no ha sido tratado.</li> </ul>

A continuación se incluirá el código que se ha generado. Este estará en ficheros sql diferentes llamados con el nombre del procedimiento.

### 3.4.2. Implementación módulo estadístico.

Para el desarrollo del módulo estadístico se han implementado una serie de disparadores de manera que una vez se active y actualice las tablas que necesite para tener realizado el cálculo. Para la implementación de este módulo necesitaremos de unas tablas con sus respectivos índices. Esto estará en los ficheros: tablas\_estadistiques.sql y index\_estadistiques.sql.

A continuación se detallarán los triggers creados para el cálculo de las estadísticas. Estos estarán implementados en la carpeta de “modulo\_estadistico” y cada uno estará en un fichero sql.

- **EST\_JUG\_ACTIVOS**: Número total de jugadores en activo en todos los géneros y modalidades. Para ello tendremos que crear dos triggers:
  - o TR\_UPDATE\_ACTIVO: Cuando se modifica el campo activo de la tabla jugadores. En el caso que dicho jugador no tenga ningún contrato no se insertara nada en la tabla estadística, ya que no conocemos ni la modalidad ni el género al no tenerlo asociado a ningún equipo.
  - o TR\_I\_JC\_EST\_ACT: Antes de insertar en la tabla contratos\_jugadores se comprueba si el género y modalidad del equipo que lo contrata ya ha sido insertado o no, en caso negativo se realiza la inserción en la tabla estadística. Se puede dar el caso que el equipo nuevo tenga otra modalidad u otro género, en ese caso se resta uno en el caso anterior y se le incrementa en 1 en el nuevo caso.
- **EST\_TEMP\_MAX\_ANOTADOR**: dada una competición, el máximo anotador en la temporada actual o en la última temporada, si nos encontramos entre medio de una temporada. En esta tabla se guardaran todos los puntos de todos los jugadores por temporadas, y para sacar la última temporada tendremos que realizar la siguiente instrucción:

```
select a.* from est_temp_max_anotador a where
competiciones_id=2 and rownum<2
order by f_inicio desc, puntos_anotados desc
```

Para ello crearemos el siguiente trigger:

- o TR\_MAX\_ANOTADOR: se crea en la tabla partidos\_jugadores donde cada vez después de una modificación se actualice la tabla estadística siempre y cuando los puntos que ha anotado el jugador varíen.

- *EST\_JUG\_MEJOR\_PAGADO*: El jugador mejor pagado de cada modalidad y género. Para ello necesitaremos crear el siguiente trigger:
  - TR\_MODIF\_SALARIO: se ejecutará después de insertar o modificar el campo salario\_bruto\_anual. Recuperaré el género y modalidad del equipo que ha hecho el contrato, si el salario bruto en euros es menor que el que está insertado en la tabla estadística se actualizará y se definirá el jugador que tiene el mejor contrato. En el caso que no exista ninguno registro con el mismo género y modalidad en la tabla estadística se insertará.
  
- *EST\_TOTAL\_GANANCIAS*: El jugador que más ha ganado a lo largo de toda su carrera deportiva. Para ello necesitaremos el siguiente trigger:
  - TR\_JUG\_MAS\_GANADO: se ejecutará después de insertar o modificar los campos f\_fin o salario\_bruto anual. Si estamos insertando o modificando el contrato se sumará lo que cobra dicho jugador a la estadística y en el caso que este modificando se le restará el valor que cobraba anteriormente. Para saber lo que cobra un jugador se creará esta función:
    - F\_GANANCIAS\_CONTRATO: pasaremos la fecha de inicio, fecha de finalización del contrato y el salario anual bruto en euros. Calculamos los meses completos que han pasado entre las dos fechas y lo multiplicamos por lo que cobra bruto un mes (*salario bruto anual / 12*). A continuación nos falta sumar lo que cobrará por los días restantes. Tendremos dos posibilidades:
      - Caso que los días restantes estén en el mismo mes, realizaré el siguiente cálculo:
 
$$((\text{días restantes} * \text{salario bruto mensual}) / (\text{días que tiene ese mes})).$$
      - Caso que los días restantes estén repartidos entre dos meses, realizaré el siguiente cálculo:
 
$$((\text{días restantes del mes que finaliza} * \text{salario bruto mensual}) / (\text{días que tiene ese mes})) + ((\text{días restantes del mes que empieza} * \text{salario bruto mensual}) / (\text{días que tiene ese mes})).$$
  
- *EST\_PTGM\_SALARIO\_MEDIO\_ANUAL*: dado un país, una temporada, un género y una modalidad el salario medio anual de los jugadores. En la tabla se guardará solo el año de inicio de la temporada, con lo que en un año pueden haber varias competiciones de esta manera se tienen en cuenta.

En esta tabla se guardará el número de contratos de los jugadores de los que se ha generado la media. Una temporada lo definirá el año de inicio y la competición. Para ello crearemos los siguiente trigger:

- TR\_PTGM\_SM\_EQTEMP\_INS: se ejecutará después de realizar un insert en la tabla equipos\_temporadas. Recuperaré la media de los salarios de todos los contratos del equipo que se está insertando y que estén vigentes en esa temporada. A continuación recalcularé la tabla estadística (est\_ptgm\_salario\_medio\_anual) con el nuevo cálculo en donde corresponda dependiendo de la temporada, modalidad, género y país.
- TR\_PTGM\_SM\_EQTEMP\_DEL: se ejecutará después de eliminar a un equipo de una temporada. Esta acción prácticamente no se utiliza, pero se implementará por si alguna vez ocurre. Este realiza lo contrario al trigger TR\_PTGM\_SM\_EQTEMP\_INS. Recuperaré la media de los salarios de todos los contratos del equipo que está eliminando y que estén vigentes en esa temporada. A continuación recalcularé la tabla estadística restándole la media calculada en donde corresponda dependiendo de la temporada, modalidad, género y país.

- TR\_PTGM\_SM\_CONT\_JUG: se ejecutará en cualquier acción de la tabla contratos\_jugadores. Hay dos casos:
  - Caso que sea una eliminación o modificación de las fechas del contrato o del salario, se tendrá que eliminar ese contrato o los datos antiguos, de las temporadas y filtros de la competición en las que afecte y hacer el recalcular.
  - Caso que sea una inserción o modificación de las fechas del contrato o del salario, se tendrá que añadir ese contrato o los datos nuevos, de las temporadas y filtros de la competición en las que afecte y hacer el recalcular.
- *EST\_CT\_PUNTOS\_EQ*: Dado una competición y una temporada, los mejores equipos ofensivos y defensivos. Equipo ofensivo y defensivo son los que mas puntos han realizado o los que menos, no se tendrá en cuenta si los han hecho en más o menos partidos. Para ello crearemos el siguiente trigger:
  - TR\_PJ\_MEJORES\_EQUIPOS: se ejecutara en la modificación de la tabla de partidos\_jugadores. Este será de tipo **compound trigger** para que no mute, ya que tenemos que consultar la misma tabla que hacemos la modificación. Primero recuperara los datos estadísticos:
    - Caso que no exista se creará poniendo al equipo del jugador que se esta modificando como mejor equipo defensivo y ofensivo.
    - Caso que ya exista.
      - El equipo máximo anotador es el que estamos modificando y la nueva puntuación es mayor a la que tenia antes. Se actualizará con la nueva puntuación.
      - El equipo máximo anotador no es el que estamos modificando o la nueva puntuación es menor a la que tenia antes. Se recalculara todas las puntuaciones para los equipos de esa temporada, y se pondrá al equipo con mayor puntuación.
      - El equipo menos anotador es el que estamos modificando y la nueva puntuación es menor a la que tenia antes. Se actualizará con la nueva puntuación.
      - El equipo menos anotador no es el que estamos modificando o la nueva puntuación es mayor a la que tenia antes. Se recalculara todas las puntuaciones para los equipos de esa temporada, y se pondrá al equipo con menor puntuación.
- *EST\_GM\_MJUG\_ANY*: para cada género y modalidad el mejor jugador del mundo en el último año (media más alta de las valoraciones ponderadas de un jugador). Para ello crearemos el siguiente trigger:
  - TR\_PJ\_GM\_MJUG\_ANY: se ejecutara en la modificación de la tabla de partidos\_jugadores. Este será de tipo **compound trigger** para que no mute, ya que tenemos que consultar la misma tabla que hacemos la modificación. Primero recuperara los datos estadísticos según género, modalidad y año del partido:
    - Caso que no exista se creará poniendo al jugador que estamos modificando como el mejor para ese género, modalidad y año.
    - Caso que ya exista.
      - Si el jugador que estamos modificando es diferente al que tenemos como el mejor o la valoración se ha decrementado por la modificación, se recalculará con la modificación realizada.

### 3.5. Juego de pruebas.

#### 3.5.1. Diseño del juego de pruebas.

Para realizar el juego de pruebas necesitamos la carga inicial de las tablas que no hemos implementado los procedimientos de ABM, estará implementado en el fichero: inserts.sql.

#### 3.5.2. Script del juego de pruebas.

Una vez creado el tablespace y el usuario he creado un script que realiza la implementación de todo el sistema, este estará implementado en el fichero: todo.sql.

A continuación se mostrara el fichero donde están todas las pruebas realizadas tanto a los procedimientos de ABM como a los de consulta.

- *Juego de pruebas de ABM de jugadores.* El fichero se llama "jp\_abm\_jugadores.sql". Este realizará 72 inserciones de jugadores cada uno con su representante. Se modificará algún dato del jugador y del representante. Se realizará un mantenimiento de bajas tanto de personales como de médicas para un jugador. Se realizará control de algunas excepciones.
- *Juego de pruebas de ABM de contratos.* El fichero se llama "jp\_abm\_contratos.sql". Se dará de alta contratos a 12 jugadores para cada equipo. Estos equipos serán: 'DKV JUVENTUD', 'MADRID', 'BARCELONA', 'VALENCIA', 'MILAN F1', 'MILAN F2'. Se realizará alguna modificación sobre un contrato y se generará la compra y venta de varios jugadores de un equipo a otro. Se realizará control de algunas excepciones.
- *Juego de pruebas de ABM de partidos.* El fichero se llama "jp\_abm\_partidos.sql". Se insertará los partidos de las temporadas 2012 ,2013 y 2014 para las competiciones de la ACB y ACB F, en la que cada equipo juegue con todos sus contrincantes. Se realizará la modificación del día en que se juega un partido por cambio. Se le designara el trio arbitral. Una vez definidos los partidos realizaremos las convocatorias de todos los jugadores de dicho equipo para todos los partidos. Después generaremos las estadísticas aleatorias para cada jugador convocado. Se realizará control de algunas excepciones.
- *Juego de pruebas de los procedimientos de consultas.* El fichero se llama "jp\_pro\_consulta.sql". Para cada procedimiento se realizará una consulta pintando lo que retorna.

#### 3.5.3. Ejecución y depuración del juego de pruebas.

Para meter las estadísticas de los jugadores se ha generado de forma aleatoria unos datos, es decir que cada vez que se inserte la estadística los resultados serán distintos. Se ha hecho así por el volumen de información a insertar. Se ha generado un script para cada tipo de procedimiento, estos ficheros serán los siguientes: jp\_abm\_jugadores.sql, jp\_abm\_contratos.sql, jp\_abm\_partidos.sql, jp\_pro\_consulta.sql.

## Valoración Económica y Recursos

A continuación se mostrará una tabla con todas las tareas del proyecto y que recursos humanos se utilizarán y el número de horas que realizarán para cada tarea según la planificación antes descrita.

TAREA	RECURSOS	Nº HORAS
<b>01 - PAC1 - Planificación del TFC</b>		<b>33</b>
01.01 - Lectura del enunciado	Jefe del Proyecto y Analista	3
01.02 - Descripción del sistema	Jefe del Proyecto	6
01.03 - Descripción del sistema de Gantt	Jefe del Proyecto	6
01.04 - Explicación diagrama de Gantt	Jefe del Proyecto	6
01.05 - Recursos Humanos y Materiales	Jefe del Proyecto	3
01.06 - Análisis de riesgos	Jefe del Proyecto	3
01.07 - Coste Inicial del Proyecto	Jefe del Proyecto	3

01.08 - Elaboración y Entrega Documentación PAC1	Jefe del Proyecto	3
<b>02 - PAC2 - Análisis y diseño</b>		<b>75</b>
02.01.01 - Requisitos Funcionales	Analista	36
02.01.02 - Requisitos No funcionales	Analista	9
02.02.01 - Diseño Conceptual	Analista	9
02.02.02 - Diseño Lógico	Analista	9
02.02.03 - Diseño Físico	Analista	9
02.03 - Elaboración y Entrega documentación PAC2	Analista	3
<b>03 - PAC3 - Implementación</b>		<b>60</b>
03.01 - Instalación SGBD Oracle	Administrador BD	3
03.02 - Configuración Oracle y Herramientas	Administrador BD	3
03.03.01 - Diseño Tablas	Programador	6
03.03.02 - Diseño Índices y disparadores	Programador	6
03.04.01 - Implementación procedimientos ABM	Programador	12
03.04.02 - Implementación modulo estadístico	Programador	9
03.05.01 - Diseño del juego de pruebas	Analista	6
03.05.02 - Script del juego de pruebas	Programador	6
03.05.03 - Ejecución del juego de pruebas	Analista	3
03.05.04 - Depuración del juego de pruebas	Programador	3
03.06 - Elaboración y Entrega documentación PAC3	Analista	3
<b>04 - Producto, Memoria y Presentación</b>		<b>54</b>
04.01 - Pruebas y depuración de Errores	Analista y Programador	15
04.02 - Redacción de la memoria	Jefe del Proyecto y Analista	27
04.03 - Redacción de la presentación	Jefe del Proyecto	9
04.04 - Entrega Final del TFC	Jefe del Proyecto	3
<b>TOTAL HORAS PROYECTO</b>		<b>222</b>

Para realizar el cálculo de los recursos humanos dispongo de la siguiente tabla de tarificación (sin IVA Incluido).

Tarifa de precios	
Recurso	Precio por Hora
Jefe de Proyecto	50 €
Analista	45 €
Administrador BD	45 €
Programador	30 €

A continuación se mostrará una tabla de tareas agrupadas por recursos humanos y el coste total que supone dicho proyecto.

JEFE DE PROYECTO		
TAREA	Nº HORAS	COSTE
01.01 - Lectura del enunciado	1,5	75 €
01.02 - Descripción del sistema	6	300 €
01.03 - Descripción del sistema de Gantt	6	300 €
01.04 - Explicación diagrama de Gantt	6	300 €
01.05 - Recursos Humanos y Materiales	3	150 €
01.06 - Análisis de riesgos	3	150 €
01.07 - Coste Inicial del Proyecto	3	150 €
01.08 - Elaboración y Entrega Documentación PAC1	3	150 €
04.02 - Redacción de la memoria	17	850 €
04.03 - Redacción de la presentación	9	450 €
04.04 - Entrega Final del TFC	3	150 €
<b>TOTAL JEFE DE PROYECTO</b>	<b>60,5</b>	<b>3.025 €</b>
ANALISTA		
01.01 - Lectura enunciado	1,5	68 €
02.01.01 - Requisitos Funcionales	36	1.620 €
02.01.02 - Requisitos No funcionales	9	405 €
02.02.01 - Diseño Conceptual	9	405 €
02.02.02 - Diseño Lógico	9	405 €
02.02.03 - Diseño Físico	9	405 €
02.03 - Elaboración y Entrega documentación PAC2	3	135 €
03.05.01 - Diseño del juego de pruebas	6	270 €
03.05.03 - Ejecución del juego de pruebas	3	135 €

03.06 - Elaboración y Entrega documentación PAC3	3	135 €
04.01 - Pruebas y depuración de Errores	7,5	338 €
04.02 - Redacción de la memoria	10	450 €
<b>TOTAL ANALISTA</b>	<b>106</b>	<b>4.770 €</b>

<b>ADMINISTRADOR BD</b>		
03.01 - Instalación SGBD Oracle	3	135 €
03.02 - Configuración Oracle y Herramientas	3	135 €
<b>TOTAL ADMINISTRADOR BD</b>	<b>6</b>	<b>270 €</b>
<b>PROGRAMADOR</b>		
03.03.01 - Diseño Tablas	6	180 €
03.03.02 - Diseño Índices y disparadores	6	180 €
03.04.01 - Implementación procedimientos ABM	12	360 €
03.04.02 - Implementación modulo estadístico	9	270 €
03.05.02 - Script del juego de pruebas	6	180 €
03.05.04 - Depuración del juego de pruebas	3	90 €
04.01 - Pruebas y depuración de Errores	7,5	225 €
<b>TOTAL PROGRAMADOR</b>	<b>49,5</b>	<b>1.485 €</b>
<b>TOTAL PROYECTO sin IVA</b>	<b>222</b>	<b>9.550 €</b>
IVA 21		<b>2.006 €</b>
<b>TOTAL PROYECTO</b>		<b>11.556 €</b>

## Conclusiones

El trabajo realizado en este proyecto ha sido muy enriquecedor, al tener que ver los distintos roles de un proyecto y diferenciar el ámbito de cada uno de ellos. Se ha definido cuatro roles Jefe del proyecto, analista, programador y gestor de base de datos. Al realizar todo el proyecto de principio a fin he visto y analizado todas las fases por las que ha ido pasando.

Lo que he visto que más importancia tiene es realizar una buena planificación teniendo en cuenta todos los recursos de los que disponemos, especificando en cada momento la tarea a realizar, teniendo en cuenta el plazo establecido.

He ampliado mis conocimientos a la hora de realizar la instalación de un SGBD sobre Ubuntu, ya que nunca lo había realizado, y ampliado con las nuevas ventajas del Oracle 11, ya que siempre había trabajado con el 10.

He intentado realizar la implementación de manera fácil, sencilla y sin complejos para una mejor comprensión del código e intentando que el sistema sea lo más abierto posible para una posible mejora de cambios o actualizaciones.

Se ha intentado que el juego de pruebas abarque la mayor parte de posibilidades y controles impuestas en el sistema, para reflejar el buen funcionamiento del sistema, y comprobar que los requerimientos están implementados en nuestro diseño.

## Glosario

- **ABM:** Abreviatura de Alta, Baja y Modificación.
- **Base de datos relacional:** Conjunto de datos organizados y relacionados entre sí.
- **BD:** Abreviatura de Base de Datos.
- **SGBD:** Abreviatura de Sistema gestor de Base de Datos.
- **Clave foranea:** El valor o los valores de un registro que hacen referencia a la clave primaria de otra tabla o entidad.
- **Clave primaria:** El valor o los valores de un registro que lo identifican de forma unívoca en una tabla de la BD.
- **Entidad:** Representación de un objeto o concepto del mundo real dentro de un base de datos.
- **Entidad débil:** Se dice de la entidad que necesita, para su identificación, hacer uso en su clave primaria de la clave primaria de otra entidad con la que estará interrelacionada .

- **Esquema E/R:** Es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información, así como sus interrelaciones y propiedades.
- **EST\_:** Prefijo que identifica a las tablas que componen el módulo estadístico.
- **Modelo relacional:** Modelo de datos basado en la lógica de los predicados y en la teoría de conjuntos.
- **Módulo estadístico:** Conjunto de tablas que contendrán información que podrá ser consultada en tiempo constante.
- **Oracle:** Sistema de gestión de Bases de Datos utilizado en el desarrollo de este proyecto.
- **P\_:** Prefijo que precede al nombre dado a los procedimientos almacenados.
- **TR\_:** Prefijo que precede al nombre dado a los trigger.
- **Trigger:** Procedimiento que se ejecuta cuando se cumplen una serie de condiciones en la tabla con la que está relacionado.

## Bibliografía

- Apuntes correspondientes a la asignatura. Gestió i desenvolupament de projectes.
- Apuntes facilitados por la UOC correspondientes a la asignatura Ingeniería del Software.
- Apuntes facilitados por la UOC correspondientes a la asignatura Bases de datos I.
- Apuntes facilitados por la UOC correspondientes a la asignatura Bases de datos II.
- Manual SQL para Oracle
- <http://www.google.es>
- <http://www.orafaq.com>
- <http://www.techonthenet.com/>

## Anexos

En el fichero zip se encontrara los siguientes ficheros:

- functions.sql
- functions\_init.sql
- index.sql
- index\_estadistiques.sql
- inserts.sql
- jp\_abm\_contratos.sql
- jp\_abm\_jugadores.sql
- jp\_abm\_partidos.sql
- jp\_pro\_consulta.sql
- seq.sql
- tablas.sql
- tablas\_estadistiques.sql
- tablespace.sql
- todo.sql
- triggers.sql
- user.sql
- modulo\_estadistico\est\_ct\_puntos\_eq.sql
- modulo\_estadistico\est\_gm\_mjug\_any.sql
- modulo\_estadistico\est\_jug\_activos.sql
- modulo\_estadistico\est\_jug\_mejor\_pagado.sql
- modulo\_estadistico\est\_ptgm\_salario\_medio\_anual.sql
- modulo\_estadistico\est\_temp\_max\_annotador.sql
- modulo\_estadistico\est\_total\_ganancias.sql
- procedimientos/p\_activo\_jugador.sql
- procedimientos/p\_alta\_contrato.sql
- procedimientos/p\_alta\_jugador.sql

- procedimientos/p\_alta\_partido.sql
- procedimientos/p\_asignarb\_partido.sql
- procedimientos/p\_baja\_jugador.sql
- procedimientos/p\_convequi\_partido.sql
- procedimientos/p\_desasignarb\_partido.sql
- procedimientos/p\_estjug\_partido.sql
- procedimientos/p\_modificar\_contrato.sql
- procedimientos/p\_modificar\_jugador.sql
- procedimientos/p\_modif\_partido.sql
- procedimientos/p\_modjug\_adddiagnostico.sql
- procedimientos/p\_modjug\_bajarepre.sql
- procedimientos/p\_modjug\_deldiagnostico.sql
- procedimientos/p\_modjug\_modificarbaja.sql
- procedimientos/p\_modjug\_nuevabaja.sql
- procedimientos/p\_modjug\_repre.sql
- procedimientos\_consulta/p\_10equip\_gastadores.sql
- procedimientos\_consulta/p\_5mej\_jug\_posicion.sql
- procedimientos\_consulta/p\_contratos\_any\_repre.sql
- procedimientos\_consulta/p\_equip\_tpunts\_comp.sql
- procedimientos\_consulta/p\_jug\_competicion.sql
- procedimientos\_consulta/p\_jug\_final\_contrato.sql