



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA

 etsinf  
Escola Tècnica  
Superior d'Enginyeria  
Informàtica

UNIVERSIDAD POLITÉCNICA DE VALENCIA  
ESCUELA TÉCNICA SUPERIOR DE INFORMÁTICA APLICADA

*Análisis, diseño e implementación de un prototipo de  
manager de fútbol online*

PROYECTO FIN DE CARRERA

Autor-es: Carlos Fernández Aguilar

Director-es: Juan Vicente Oltra Gutiérrez

25-Junio-2010

# Índice

<b>1. Objeto y objetivos.....</b>	<b>2</b>
<b>2. Introducción.....</b>	<b>2</b>
<b>3. Especificación del Ámbito y Alcance.....</b>	<b>3</b>
<b>4. Descripción de las Alternativas de Solución.....</b>	<b>4</b>
<b>5. Identificación del Entorno Tecnológico.....</b>	<b>9</b>
<b>6. Identificación de estándares y normas.....</b>	<b>13</b>
<b>7. Determinación del Alcance del Sistema y descomposición en Subsistemas de Análisis.....</b>	<b>14</b>
<b>8. Elaboración del Modelo Conceptual de Datos.....</b>	<b>17</b>
<b>9. Obtención del Modelo de Procesos del Sistema.....</b>	<b>26</b>
<b>10. Diseño del Modelo Físico de Datos .....</b>	<b>105</b>
<b>11. Definición de Interfaces de Usuario.....</b>	<b>115</b>
<b>12. Definición de Niveles de Arquitectura .....</b>	<b>126</b>
<b>13. Identificación de subsistemas de diseño .....</b>	<b>127</b>
<b>14. Especificación de Excepciones .....</b>	<b>129</b>
<b>15. Especificación del Entorno de Construcción .....</b>	<b>133</b>
<b>16. Elaboración de Especificaciones de Construcción .....</b>	<b>133</b>
<b>17. Conclusiones .....</b>	<b>142</b>
<b>18. Anexos .....</b>	<b>143</b>
<b>Bibliografía .....</b>	<b>171</b>

# 1. Objeto y objetivos

El objeto del presente Proyecto Fin de Carrera es la obtención del título de Ingeniero en Informática expedido por la Universidad Politécnica de Valencia. Se trata de un proyecto realizado en el seno de la empresa valenciana The Netwizzy Company SL.

El objetivo principal es construir un prototipo de un manager de fútbol online, es decir, de un juego consistente en un simulador de una competición futbolística en el que varios usuarios (personas físicas) puedan manejar un equipo de fútbol y gestionar diversos aspectos del mismo en tiempo real.

Se desea documentar el proyecto siguiendo el método Métrica 3, una metodología de planificación, desarrollo y mantenimiento de sistemas de información promovida por el Ministerio de Administraciones Públicas del Gobierno de España.

El sistema a construir se pretende que sea la primera versión de un supuesto desarrollo final preparado para su uso en un portal online de juegos; una versión con gran parte de las funcionalidades que podría tener un desarrollo final con la particularidad de que la competición simulada no funcione en tiempo real (se implementará un paso del tiempo artificial, es decir, manejado por el administrador del sistema) y con algunas partes del mismo no desarrolladas hasta un punto totalmente profesional (por ejemplo el motor de simulación de los partidos, que podría ser un tema por si solo para otro PFC).

Dado que el objeto del proyecto es construir un prototipo de un juego pensando para ubicarse en portal de juegos online con forma de Red Social (<http://www.our.com>), uno de los objetivos del proyecto será dotar al sistema a construir de algún tipo de componente social, de interacción entre los usuarios del sistema.

Se pretende también que la aplicación web desarrollada tenga una interfaz gráfica atractiva; hay que tener en cuenta que el usuario para el que está destinado este tipo de desarrollos (juegos online) no son clientes de aplicaciones online al uso en los que el criterio principal de diseño es la sencillez y usabilidad, sino que son usuarios que usan la aplicación con fines recreativos.

Es conveniente construir ficheros que realicen una carga automática de datos inicial del sistema; el prototipo a construir necesita datos desde el principio de su uso por parte de los usuarios, por lo que se buscará implementar scripts que hagan una carga automática de datos y que puedan ser usados las veces que se quiera.

Por último, y desde una perspectiva más general, cabría decir que el objetivo del proyecto es construir una aplicación web cliente-servidor cubriendo todos los procesos que la tarea conlleva (diseño y construcción de la solución, elección entre sistemas de gestión de base de datos y servidor e implantación del sistema construido).

## 2. Introducción

Esta memoria se ha realizado siguiendo la metodología Métrica Versión 3 en su desarrollo orientado al diseño estructurado.

La mayor parte de los capítulos pueden identificarse con una tarea dentro de los procesos principales de la metodología Métrica 3; otros son una composición de varias de ellas. En cualquier caso, en este apartado se glosará cada capítulo del documento y se identificarán estas relaciones.

La estructura de cada capítulo (de los correspondientes a los procesos de Métrica 3) es la misma: se realiza una pequeña descripción de la tarea a realizar (a veces citando literalmente el manual Métrica 3, en cuyo caso se incluirá una reseña bibliográfica; otras, haciendo alguna pequeña modificación, aprovechando el carácter abierto de la metodología).

En el capítulo tres se puede encontrar la definición del ámbito y alcance del sistema a construir, una delimitación de los objetivos generales del sistema. Se corresponde con la tarea PSI 2.1 del método Métrica 3.

En el capítulo cuatro se hace una primera aproximación a la solución que se desea construir, obteniendo una primera versión del catálogo de requisitos del sistema y especificando las funcionalidades del sistema mediante la técnica de los casos de uso. Se correspondería con la tarea EVS 4.2 del método Métrica 3.

En el capítulo cinco se realiza una descripción detallada del entorno tecnológico necesario para dar respuesta a las necesidades del sistema. Se describe el tipo de solución que va a construirse y las necesidades tecnológicas de la misma, estudiando las diferentes alternativas que pueden emplearse. Se corresponde con el proceso PSI 2.1.

En el capítulo seis se hace una mención a los estándares y normas que deben tenerse en cuenta en el desarrollo a realizar. Corresponde al proceso ASI 1.2 de Métrica 3.

En el capítulo siete se realiza una definición del alcance del sistema obteniendo el Diagrama de Contexto del mismo, y se descompone dicho diagrama para obtener los Subsistema de Análisis. Se realiza desde un punto de vista lógico y se utilizan diagramas de flujos de datos (DFD's). Correspondería a los procesos ASI 1.1 y ASI 3.1.

En el capítulo ocho se obtiene el modelo conceptual de datos del sistema mediante diagramas entidad-relación: asimismo se añaden una serie de reglas de negocio o restricciones semánticas que definen de una manera más precisa al sistema. Se corresponde al proceso ASI 6.1.

En el capítulo nueve se realiza el modelado de procesos del sistema; a partir del diagrama de flujo de datos obtenido en el capítulo siete, se sigue descomponiendo cada diagrama obtenido hasta llegar al último nivel, en el que todos los procesos son primitivos. Asimismo, se realiza una especificación de cada proceso de este tipo y para cada diagrama obtenido se realiza una especificación de los flujos de datos y de los almacenes que intervienen. Se trata del proceso ASI 7.1 de Métrica 3.

En el capítulo diez se puede encontrar el Modelo Físico de Datos del sistema. A partir del Modelo Conceptual se diseñan todas las tablas de datos que necesitará el sistema, definiendo el tipo de cada atributo, añadiendo una descripción e indicando claves primarias y ajenas. Correspondería con la tarea DSI 6.1.

En el capítulo once se definen las interfaces del usuario del sistema. Se definen los principios generales de la interfaz que se desea construir, los formatos individuales de la misma (diálogos, formularios y menús), y el comportamiento dinámico de la interfaz. Corresponde a las tareas ASI 8.1 , 8.3 y 8.4.

En el capítulo doce se definen los niveles de arquitectura software del sistema; se incluye un diagrama de despliegue en los que se define las principales particiones físicas del sistema y sus comunicaciones. Corresponde al proceso DSI 1.1

En el capítulo trece se realiza una identificación de los subsistemas de diseño; se trata de una división en base a distintos criterios de los componentes que se deben construir para ensamblar el sistema. Se corresponde con el proceso DSI 1.5

En el capítulo catorce se implementa un catálogo de excepciones, de situaciones anómalas que pueden producirse en el sistema y la forma en que éste actúa ante ellas. Se trata del proceso DSI 1.3

En el capítulo quince se realiza la especificación del entorno de construcción; es una descripción de los recursos tecnológicos de los que se dispone para construir el sistema.

En el capítulo dieciseis se define mediante lenguaje natural cada componente (archivo) de los que consta el sistema. Se corresponde al proceso DSI 8.3 (Elaboración de Especificaciones de Construcción).

Por último, en el capítulo diecisiete se encuentran las conclusiones del trabajo, incluyendo alguna sugerencia sobre una hipotética continuación en el proyecto realizado, y el dieciocho corresponde al apartado de anexos, que cuenta con el Catálogo de Requisitos del proceso a construir (es un documento en constante cambio y evolución a lo largo de la metodología Métrica 3 y para evitar redundancia se ha optado por colocar su versión final como anexo), con la inclusión de algunos archivos no definidos en el capítulo dieciseis al no ser

necesarios en el funcionamiento del sistema (capítulos de carga de datos y de creación de las tablas de base de datos necesarias), y con un manual de usuario.

## 3. Especificación del Ámbito y Alcance

### Descripción de la tarea

*El objetivo de la tarea es definir el ámbito de los procesos de la organización a considerar y definir el alcance del sistema, es decir, los objetivos específicos del sistema a construir. Puede ser necesario determinar distintos objetivos para cada proceso del proyecto (Planificación de Sistemas de Información, Métrica 3:pág. 6).*

### Desarrollo de la tarea

El sistema de información descrito en este documento se refiere a un prototipo de un manager de fútbol online.

Por manager de fútbol online entendemos una aplicación web que simule una competición futbolística en el que cada equipo esté controlado por un ser humano o por el propio sistema.

Este documento está escrito siguiendo el método estándar del Ministerio de Administraciones Públicas español, el Métrica 3, siguiendo el enfoque dirigido al Diseño Estructurado del mismo.

De manera general podemos citar las principales funciones que deberá implementar el sistema:

- El sistema creará todos los datos necesarios para poder iniciar la simulación.
- El sistema otorgará un equipo a todo aquel usuario registrado que lo solicite mientras haya equipos libres(manejados por el propio sistema hasta ese momento).
- El sistema permitirá al usuario gestionar su equipo de una manera próxima a la realidad en la gestión de un equipo de fútbol.
- El sistema estará dotado de cierto contenido social (interacción entre usuarios).
- El sistema simulará de una manera lo más fidedigna a la realidad posible ciertos aspectos como la evolución de los jugadores con el entrenamiento o el desarrollo de un partido de fútbol.

Este último aspecto, el de la simulación fidedigna de un partido de fútbol, es el que menos prioridad tendrá en el desarrollo del prototipo dada su complejidad, pretendiéndose llegar a un desarrollo completo o casi completo en el resto de funciones.

El sistema tendrá dos vistas diferenciadas:

- La de del administrador, que podrá llevar a cabo las funciones destinadas a simular el comportamiento del sistema si fuera un sistema en tiempo real.
- La de los usuarios, que utilizarán el sistema para gestionar su equipo, interactuar con otros equipos, etc.

Lo que el sistema no hará será encargarse del proceso de registro de nuevos usuarios, pues la aplicación estará incluida en un portal ya en funcionamiento y se pretende que para poder participar en el manager online debas estar registrado en el portal.

## 4. Descripción de las Alternativas de Solución

### Descripción de la tarea

*El propósito de la tarea es una primera definición de la alternativa a construir. Para ello se elabora una primera versión del catálogo de requisitos y se especifica mediante la técnica de los casos de uso las funcionalidades que se desea contenga el sistema. (Estudio de la Viabilidad del Sistema, Métrica 3:pág. 11).*

## Desarrollo de la tarea

En este apartado se va a construir una primera versión del Catálogo de Requisitos de la solución , describiendo cada requisito y asignándoles una prioridad. En capítulos posteriores se ampliará el catálogo y se le adjudicará a cada uno un subsistema de diseño dónde ubicarlo.

Catálogo de Requisitos	
Id	Descripción
1	(A) El sistema simulará una competición futbolística, creando al inicio los equipos que la componen y asegurándose de que todos tengan un nivel semejante en un principio.
2	(A) El sistema asignará equipos a los usuarios a medida que le vayan siendo solicitados, y se encargará de gestionar los equipos que no estén asignados a ningún usuario asegurando un normal discurso de la competición.
3	(A) El sistema deberá hacer la clasificación de la competición ordenando a los equipos según los siguientes criterios (en orden de prioridad, si varios equipos empatan para un criterio se usa el siguiente y así sucesivamente): puntos totales, puntos ganados en los partidos que implican a los equipos empatados, diferencia de goles marcados y recibidos en los partidos que implican a los equipos empatados, diferencia general entre goles marcados y goles recibidos, sorteo.
4	(A) Cada equipo contará con su versión juvenil que también será gestionado por el usuario (se le asignará un equipo normal y uno juvenil)
5	(A) Además de los cambios producidos en el estado del sistema por los distintos eventos provocados por un usuario del mismo, en el sistema se implementarán dos eventos que producirán cambios en el estado del mismo: la simulación de una jornada de liga y la simulación del paso de una semana.
6	(A) El sistema creará un cierto número de jugadores libres, sin ningún equipo asignado, que podrán ser fichados en todo momento por los equipos.
7	(A) El sistema deberá dar la posibilidad al usuario de cambiar en cualquier momento la alineación(normal y juvenil) guardada por el sistema para su equipo.
8	(A) El usuario tendrá la posibilidad de cambiar en cualquier momento el entrenamiento guardado por el sistema para su equipo.
9	(A) El usuario podrá interactuar con todos los jugadores (componentes de los distintos equipo de sistema) del sistema, pudiendo realizar diversas acciones sobre cada uno de ellos.
9	(A)El usuario se ocupará también de la gestión de su equipo juvenil, cuya función esencial es nutrir de jugadores el equipo normal.
10	(A)Cada equipo del sistema cuenta con un balance económico en cada momento, en función del cual algunas operaciones (como el fichaje de jugadores) le pueden ser denegadas al usuario.
11	(A) El sistema implementará un sistema de créditos; el usuario podrá pedir un crédito al interés fijado por el sistema cuando quiera pero siempre que no tenga uno ya vigente, pudiéndolo devolver en cualquier momento siempre que el saldo del equipo lo permita.
12	(A) El sistema deberá simular un sistema de patrocinio; cada equipo podrá tener un patrocinador vigente y un acuerdo para próximas temporadas con un patrocinador. Cancelar un acuerdo de patrocinio vigente le supondrá al equipo una penalización económica, mientras que anular un acuerdo próximo no tendrá penalización.
13	(A) Cada equipo podrá tener empleados, pertenecientes a diversas categorías y con funciones concretas. Existirá en el sistema también la figura del empleado no asignado a ningún equipo.
14	(A) El sistema creará un cierto número de empleados sin equipo que serán mostrados al usuario y podrán ser fichados para su equipo.
15	(A) Los jugadores normales (no juveniles) y empleados estarán ligados a un equipo mediante contratos definidos por un sueldo y una duración.

16	(A) Cada equipo tendrá asignado un estadio que podrá ser ampliado por el usuario siempre que el saldo económico del equipo lo permita.
17	(A) El usuario podrá fijar el precio de las entradas para los partidos disputados en su estadio.
17	(A) La habilidad de los distintos jugadores de los equipos estará codificada en diversos parámetros cuyos valores variarán a lo largo del tiempo en función de la edad de cada jugador y del entrenamiento asignado a su equipo.
18	(A) Un usuario podrá en cualquier momento mandar un mensaje a otro usuario.
19	(A) Los usuarios con categoría de Administrador podrán en cualquier momento ejecutar una opción que simulará todos los partidos correspondientes a una jornada de la competición virtual.
20	(A) Para obtener los resultados de cada partido el sistema tendrá en cuenta la habilidad de los jugadores de cada equipo disputante del partido, de los empleados con categoría "entrenador" y de otros factores, además de añadir una componente de aleatoriedad.
21	(A) La simulación de un partido conllevará, además de la obtención del resultado del mismo, el registro de eventos como las posibles lesiones de jugadores y el cálculo de la recaudación por parte del equipo con carácter local.
22	(A) El sistema a desarrollar es un prototipo de un sistema que funcionará en tiempo real y que tendrá una serie de ejecución de procesos cada cierto tiempo (por ejemplo, cada semana). Por ello en nuestro prototipo un usuario con privilegios de administrador tendrá siempre la posibilidad de provocar la ejecución de estos procesos englobados en un subsistema llamado "simulación de paso de semana".
23	(A) Cuando se ejecute la opción de paso de semana el sistema deberá simular una sesión de entrenamiento. Este proceso tendrá efecto sobre diversos parámetros de los jugadores.
24	(A) El sistema deberá actualizar las ofertas de patrocinio vigentes para cada equipo, añadiendo una para todo equipo que en ese momento cuente con menos de cuatro.
25	(A) Con cada paso de semana el sistema recalculará el valor de un parámetro que represente la moral de los jugadores de cada equipo.
26	(A) Los gastos de cada equipo correspondientes a los sueldos de sus jugadores y empleados, obras en el estadio y devolución de créditos pedidos se llevarán cabo al lanzarse el proceso de simulación de paso de semana.

**Fig 1. Primera versión del Catálogo de Requisitos del sistema**

Una vez obtenido el catálogo se definen las funcionalidades del sistema mediante unos diagramas de casos de uso, cuyos objetivos pueden describirse de la siguiente manera:

- Capturar los requisitos funcionales del sistema y expresarlos desde el punto de vista del usuario.
- Guiar todo el proceso de desarrollo del sistema de información. (Técnicas y Prácticas, Métrica 3:pág. 7)

Se observa que el sistema a construir consta con dos actores posibles: el definido como "administrador", cuyo fin será gestionar el sistema, y el definido como "usuario", el que intervendrá en una mayoría de escenarios del sistema.

Por último se realiza una breve descripción de cada caso de uso empleando lenguaje natural:

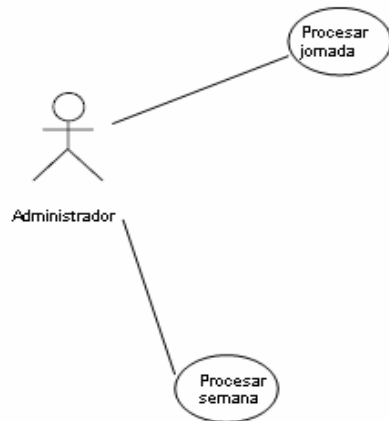


Fig 2. Diagrama 1 de Casos de Uso del prototipo

**Procesar Jornada:** En este caso de uso el actor Administrador (un usuario del sistema con categoría “administrador”) elige la opción de procesar jornada con el fin de que el sistema simule la disputa de una jornada del campeonato.

**Procesar Semana:** En este caso de uso el actor Administrador (un usuario del sistema con categoría “administrador”) elige la opción de procesar semana con el fin de que el sistema simule el paso de una semana.

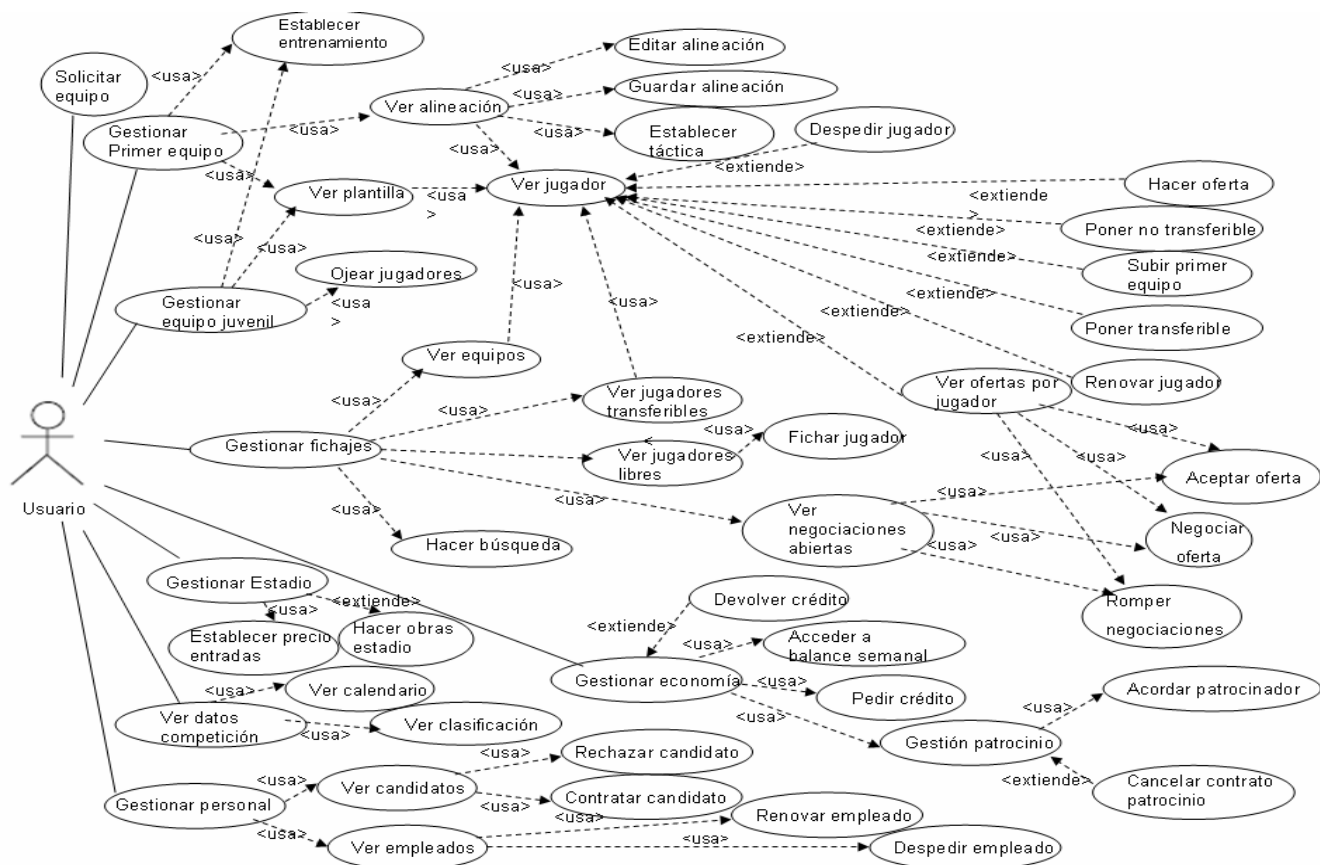


Fig 3. Diagrama 2 de Casos de Uso del prototipo

**Solicitar equipo:** En este caso de uso el actor Usuario (un usuario del sistema con categoría “usuario”) interactúa con el sistema solicitando que se le asigne un equipo.

**Gestionar primer equipo:** El usuario administra diversos aspectos de su equipo.

**Establecer entrenamiento:** El usuario elige la configuración del entrenamiento de su equipo.

**Ver alineación:** El usuario consulta al sistema cual es la alineación de su equipo.



**Editar alineación:** El usuario cambia el estado de la alineación de su equipo.

**Guardar alineación:** El usuario le indica al sistema que guarde la configuración actual de la alineación de su equipo.

**Establecer entrenamiento:** El usuario elige la configuración táctica de su equipo.

**Establecer táctica:** El usuario elige la configuración táctica de su equipo.

**Ver plantilla:** El sistema ofrece al usuario información sobre los jugadores pertenecientes a su equipo.

**Gestionar primer equipo:** El usuario administra diversos aspectos de su equipo juvenil.

**Ojear jugadores:** El usuario establece cuales son los jugadores de su equipo juvenil que desea ojear.

**Ver jugador:** El usuario desea que el sistema le muestre información detallada sobre algún jugador.

**Despedir jugador:** El usuario desea eliminar de su plantilla a algún jugador.

**Hacer oferta:** El usuario desea hacer una oferta de traspaso por algún jugador.

**Poner transferible:** El usuario desea poner a un jugador de su equipo el estado de transferible.

**Poner no transferible:** El usuario desea poner a un jugador de su equipo el estado de no transferible.

**Subir primer equipo:** El usuario desea que un jugador juvenil de su equipo pasa a formar parte de la plantilla del primer equipo.

**Renovar jugador:** El usuario desea hacer una oferta de renovación a un jugador de su equipo.

**Ver ofertas por jugador:** El sistema le muestra al usuario las ofertas de traspaso que posee por algún jugador de su equipo.

**Aceptar oferta:** El usuario desea aceptar una oferta de traspaso por alguno de sus jugadores.

**Negociar oferta:** El usuario desea realizar una contraoferta por alguna oferta de traspaso hecha a uno de los jugadores de su equipo.

**Romper negociaciones:** El usuario quiere rechazar una oferta recibida por uno de sus jugadores.

**Gestionar fichajes:** El usuario quiere administrar varios aspectos relacionados con los fichajes de jugadores.

**Ver equipos:** El sistema le muestra al usuario los demás equipos de la liga en la que participa el equipo del usuario.

**Ver jugadores transferibles:** El usuario desea que el sistema le muestre los jugadores de los diversos equipos de la competición implementada por el sistema que estén marcados como transferibles.

**Ver jugadores libres:** El usuario desea que el sistema le muestre los jugadores no asignados a ningún equipo.

**Ficha jugador:** El usuario desea fichar para su equipo a algún jugador libre.

**Hacer búsqueda:** El usuario desea realizar una búsqueda entre todos los jugadores del sistema estableciendo algunos criterios en la operación.

**Ver negociaciones abiertas:** El usuario le pide al sistema que le muestre el estado de las negociaciones de fichaje relacionadas con jugadores de su equipo o con ofertas hechas por el equipo del usuario a otros jugadores.

**Ver datos competición:** El usuario desea consultar diversos datos sobre la competición.

**Ver calendario:** El sistema le muestra al usuario datos sobre los partidos disputados y a disputar próximamente por su equipo.

**Ver clasificación:** El sistema le muestra al usuario información sobre la clasificación de la competición.

**Gestionar estadio:** El usuario administra varios aspectos relacionados con el estadio de su equipo.

**Hacer obras estadio:** El usuario desea aumentar la capacidad de su estadio y/o mejorar el nivel de los accesos y/o tienda del mismo.

**Establecer precio entradas:** El usuario fija el precio de las entradas de los partidos disputados en su estadio.

Nombre	Definición
SGBD	Sistema de Gestión de Base de Datos
IIS	Internet Information Server. Es el servidor web de Microsoft
Equipo	Son los entes que componen la competición simulada por el sistema
Jugadores	Elementos que forman los equipos
Jugadores juveniles	Son jugadores que forman parte de los equipos juveniles.
Empleados	Entes que forman parte de los equipos y se encargan de distintas tareas.
Candidatos	Son empleados ofrecidos a un equipo pero que no forman parte de él.

<b>Entrenador</b>	Es un empleado con la categoría de entrenador.
<b>Segundo entrenador</b>	Es un empleado con la categoría de segundo entrenador.
<b>Médico</b>	Es un empleado con la categoría de médico.
<b>Fisioterapeuta</b>	Es un empleado con la categoría de fisioterapeuta.
<b>Relaciones públicas</b>	Es un empleado con la categoría de relaciones públicas.
<b>Psicólogo</b>	Es un empleado con la categoría de psicólogo.
<b>Entrenador de porteros</b>	Es un empleado con la categoría de preparador físico.
<b>Preparador físico</b>	Es un empleado con la categoría de preparador físico.
<b>Ojeador</b>	Es un empleado con la categoría de ojeador.
<b>Patrocinador</b>	Es una empresa que ha suscrito un acuerdo de patrocinio con equipo.
<b>Patrocinador siguiente</b>	Es una empresa que ha suscrito un acuerdo de patrocinio que entrará en vigor cuando el actual acuerdo expire.
<b>Oferta traspaso</b>	Cantidad de dinero que el equipo de un usuario ofrece a otro con fin de fichar a un jugador.
<b>Indemnización</b>	Es la cantidad de dinero que obtiene un equipo cuando se disputa un partido en su estadio.
<b>Recaudación</b>	Cantidad de dinero que debe pagar un equipo cuando rescinde un contrato vigente con un empleado o con un patrocinador.

## 5. Identificación del entorno tecnológico

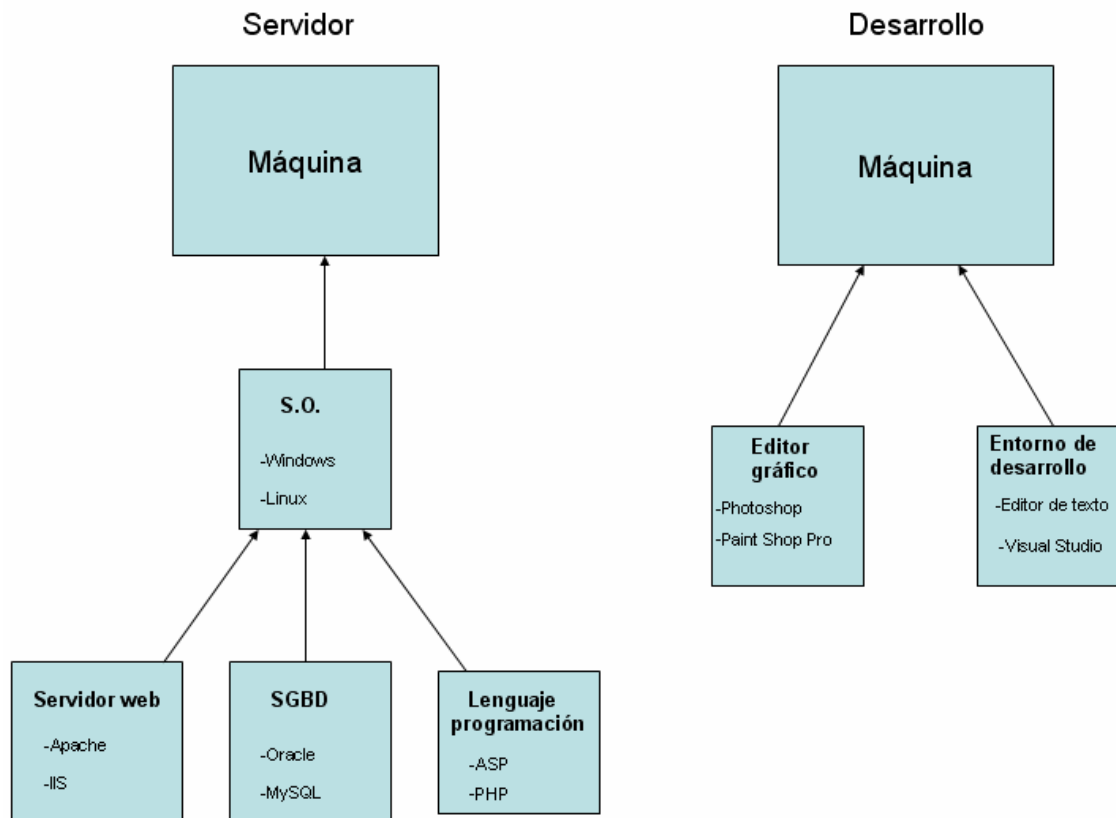
### Descripción de la tarea

*El objetivo de esta tarea es definir, a alto nivel, el entorno tecnológico requerido para dar respuesta a las necesidades de información del sistema a construir. (Análisis del Sistema de Información, Métrica 3: pág. 7)*

### Desarrollo de la tarea

El sistema a desarrollar se trata de una aplicación cliente-servidor. Se necesitarán máquina/s que haga/n la función de servidor, elegir una entre las alternativas de servidor web existentes en el mercado, escoger un Sistema de Gestión de Base Datos y emplear uno de entre los posibles lenguajes de programación web para implementar la aplicación (además de un entorno de desarrollo); asimismo podría ser necesario utilizar algún programa de tipo editor gráfico. Además habría que determinar que sistema operativo estaría soportado por la máquina/s que haga/n las funciones de servidor.

Este Diagrama de Representación modela el entorno tecnológico implicado en la construcción de nuestro desarrollo web:



**Fig 4. Entorno tecnológico del sistema**

Pasamos a describir las diferentes alternativas que podemos encontrar en la configuración del entorno tecnológico:

La decisión del S.O. con el que funcionará/n el/los servidor/es es fundamental porque de ella dependen la resolución de varias de las demás alternativas. Si optamos por usar una máquina con S.O. Windows y con un servidor web IIS, estaremos obligados a emplear el lenguaje ASP para el desarrollo de la aplicación y un entorno de desarrollo como Visual Studio (por ejemplo).

Si por el contrario optamos por Linux con un servidor Apache, podremos elegir entre lenguajes como PHP y PERL, sin vernos obligados a usar ningún entorno de desarrollo concreto (se podría usar un simple editor de texto).

Hay otras posibilidades, como por ejemplo, utilizar un servidor Apache sobre una máquina con Windows, pero las más extendidas en la práctica son las ya comentadas.

Dada la importancia de la decisión entre servidores Linux y servidores Windows, vamos a profundizar en la cuestión sobre las ventajas que tiene uno sobre el otro:

-Según el especialista en virtualización y mantenedor de servidores tanto Windows como Linux Andrew Kutz, uno de los problemas de los servidores Windows es la seguridad, lo que obliga a tener que instalar numerosos parches de manera frecuente, mientras que Linux es bastante más estable; otro problema es el de la inestabilidad del propio sistema operativo. Por el contrario, pese a que los servidores Linux requieren menos esfuerzo dedicado a comprobar su seguridad, pueden resultar menos amigable de cara a su configuración, y tienen que ser actualizados en cuanto hay una actualización del núcleo(kernel) del SO.

El especialista concluye aseverando que si bien cada una de las alternativas tiene sus ventajas e inconvenientes, ninguna de ellas es abrumadoramente superior como para recomendar incondicionalmente su uso, y que la decisión final debería depender del grado de comodidad del usuario en un sistema operativo u otro. (<http://searchservervirtualization.techtarget.com>, 2006).

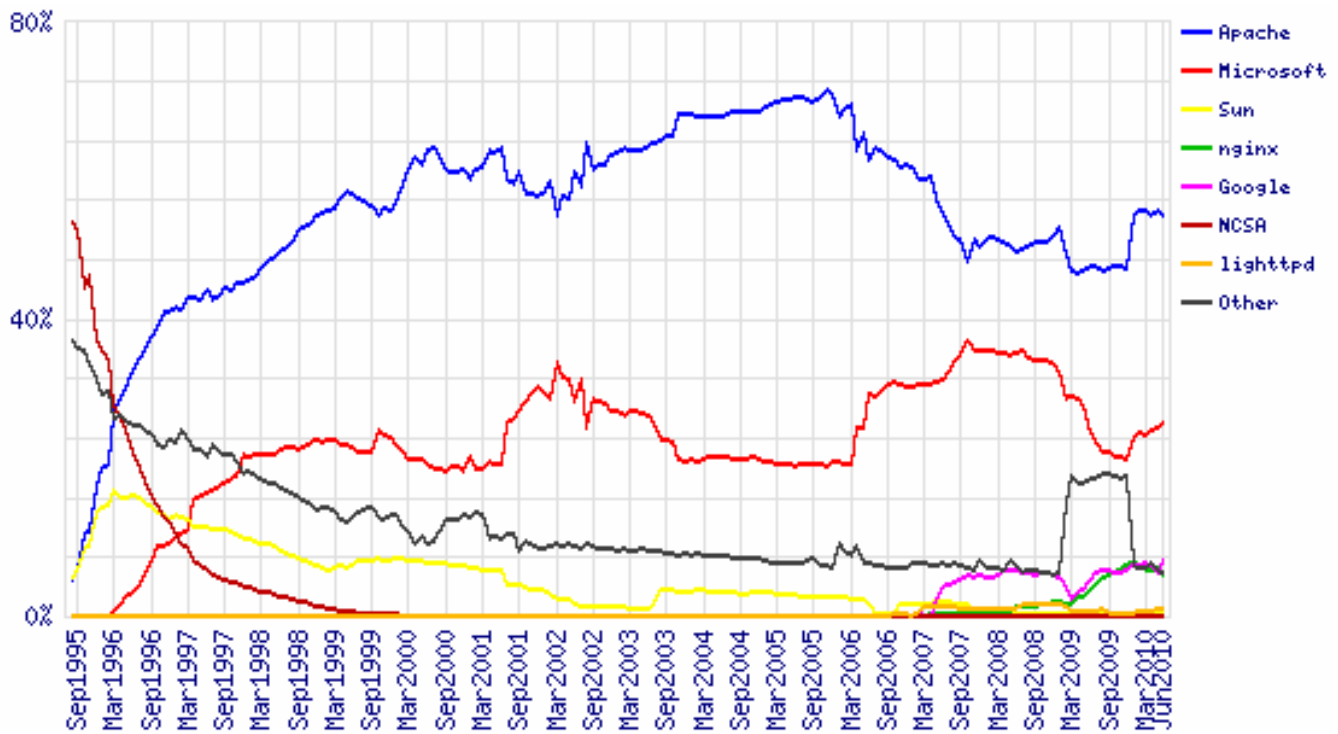
-El Web Server Survey de Netcraft (NetCraft, 2010) es un sondeo que recoge estadísticas de varios millones de sitios web acerca del uso de los diferentes servidores. Observamos los datos que recogió el estudio en Septiembre del 2008 y en su última actualización:

Third Quarter Results		
Server	No. of Sites	Market Share
Apache	91,068,713	50.24%
Microsoft-IIS	62,364,634	34.40%
GFE	10,072,687	5.56%
Unknown	3,835,616	2.12%
lighttpd	3,095,928	1.71%
nginx	2,562,554	1.41%
Oversee	1,938,953	1.07%
Other	1,204,143	0.66%
IdeaWebServer	573,812	0.32%
GNR	373,722	0.21%
Zeus	356,618	0.20%
Coyote	356,226	0.20%
Sun-ONE-Web-Server	354,133	0.20%
Resin	343,323	0.19%
Rapidsite	227,994	0.13%
BaseHTTP	206,270	0.11%
Jetty	201,075	0.11%
LiteSpeed	189,762	0.10%
tigershark	177,552	0.10%
httpd	174,224	0.10%

Fig 5. Utilización de distintos servidores en sitios web en Septiembre 2008

Developer	May 2010	Percent	June 2010	Percent	Change
Apache	112,663,533	54.68%	111,792,321	54.02%	-0.67
Microsoft	52,062,154	25.27%	53,865,345	26.03%	0.76
Google	12,357,212	6.00%	15,375,950	7.43%	1.43
nginx	13,490,726	6.55%	11,264,229	5.44%	-1.11
lighttpd	1,869,658	0.91%	1,704,797	0.82%	-0.08

Fig 6. Utilización de distintos servidores en sitios web en Mayo-Junio 2010



**Fig 6. Evolución en la utilización de distintos servidores en sitios web**

Se observa un dominio claro de los servidores Apache frente a los IIS(Microsft), con más de 50% de cuota de mercado frente a un 35%; en el gráfico de tendencias se puede apreciar como Apache pasó de tener un 70% del mercado a finales del año 2005 a un 50% en otoño del 2008, siendo más o menos estable desde entonces a la actualidad en la que lleva unos meses en torno al 55% de cuota. Por su parte en relación a los servidores Microsoft se observa una fuerte subida desde finales del 2006, un periodo estable desde otoño del 2007 a otoño del 2008 y una fuerte caída desde entonces al momento actual.

-Podríamos cerrar el apartado del tipo de servidor a emplear aludiendo a un artículo de la web de una empresa tecnológica entre cuyos focos de trabajo está todo lo relacionado con los servidores web(Candidinfo, 2006) que resume las ventajas y desventajas de ambas alternativas(las dos principales) de esta manera:

### APACHE

#### **-Ventajas**

Tiene constantes actualizaciones.

Es gratis.

Soporte multiplataforma (funciona tanto bajo Linux como Windows)

Es el más popular del mercado (poco probable que dejen de aparecer nuevas mejoras en el futuro).

#### **-Inconvenientes**

Los desarrolladores de Apache no proveen ninguna asistencia. Otras compañías si lo hacen, pero mediante pago.

Funciona mejor y más rápido en Linux que en Windows.

### IIS

#### **-Ventajas**

Es un producto Microsoft totalmente integrado en el S.O., con lo que ello conlleva de comodidad.

Está incluido gratuitamente en los sistemas operativos de Windows.

Tiene la posibilidad de limitar el ancho de banda de sus páginas web. Esto tiene utilidad si tu proveedor de servicios de Internet te carga una cantidad extra si superas cierta cantidad de ancho de banda en un mes, Protección frente a errores; si una aplicación que corre sobre el servidor falla, el servidor y el resto de aplicaciones continúan ejecutándose y la aplicación fallida se reinicia la siguiente vez que un usuario la requiere.

### **-Inconvenientes**

Limitado a sistemas Windows.

Es código cerrado, por lo que no hay muchas compañías que se dediquen a trabajar en mejoras del código como si ocurre en Apache.

Respecto a las alternativas entre los Sistemas de Gestión de Base de Datos, Microsoft SQL Server es un SGBD integrado en los S.O. Windows por lo que si montamos el servidor en una máquina Windows con un programa servidor web IIS puede ser la mejor opción al estar totalmente integrado ya en el sistema el Microsoft SQL Server.

Además de esta opción, hay otras que pueden aplicarse a cualquier S.O. y programa servidor web, entre las que se pueden nombrar Oracle, MySQL o PostGreSQL. Las dos últimas opciones son soluciones en código abierto y gratuitas mientras que Oracle es una herramienta de pago, pero vamos a comentar en más profundidad las particularidades de cada opción:

### **-Oracle**

Es un producto bastante potente pero también caro, por lo que su uso se da generalmente en empresas muy grandes y multinacionales. No tiene ninguna restricción en cuanto al tipo de sistema operativo con el que puede trabajar. En cuanto a los API's soportados (librerías para lenguajes a partir de los cuales conectarse a la base de datos), la última versión soporta todos los lenguajes de programación habituales (Java, PHP, PERL, C++,etc); en cuanto a los conectores (.net, ODBC, JDBC) no tiene ninguna restricción.

### **-MySQL**

MySQL es una alternativa muy extendida por ser una herramienta de código abierto. Aunque no alcanza el nivel de robustez y fiabilidad de Oracle es un sistema fiable y orientado a la velocidad de proceso y al consumo de pocos recursos. No tiene ninguna restricción respecto a API's y conectores. Según el portal MySQL Hispano algunas de las empresas de Internet más importantes como pueden ser Yahoo o Google usan MySQL (<http://www.mysql-hispano.org/articulos/num43/analisis-comparativo.pdf:93>).

### **-PostGreeSQL**

Es una herramienta de código abierto pero bastante más lenta y eficiente en el consumo de recursos que MySQL. Está considerada como una mejor opción para desarrollos en los que la consistencia de la BD sea fundamental frente a aspectos como la velocidad y el número de accesos concurrentes (Comunidad Fedora en Castellano, 2009).

Respecto al lenguaje de programación utilizado para construir aplicaciones web, las alternativas más extendidas son ASP y PHP.

PHP son las siglas de PHP Hypertext Processor, y es un lenguaje de programación muy potente, con una sintaxis parecida a la del lenguaje C, que nos permite ejecutar nuestro propio código antes de enviarle la página HTML al usuario que accede a la misma. Es decir, si un usuario intenta acceder a una determinada página web situada en una determinada URL, el código PHP de la misma es procesado y ejecutado en el servidor que aloja la página web, generando un código HTML resultante. Ese código HTML es el que se le enviará al navegador del usuario.

ASP es la versión Windows para generar páginas web dinámicas. Si se elige un entorno de desarrollo como Visual Studio (o .NET Framework) ASP (ASP.NET) es la alternativa más sensata, pues aunque han surgido ciertas extensiones que soportan el desarrollo con PHP en Visual Studio, ASP.NET es uno de los lenguajes originales para los que se desarrolló dicho entorno y resulta más integrado en la plataforma.

Si se elige PHP, los entornos de desarrollo en los que podría construirse la solución web son numerosos: PHPEdit (para Windows, con depurador, generador de tareas pendientes y varios pluggins), PHPCoder, etc. Incluso podría bastar con un simple editor de texto.

En cuanto a la idoneidad de la elección, lo más sensato parece seguir una cierta coherencia con el resto de las soluciones elegidas para el desarrollo del proyecto: IIS, Microsoft SQL Server, ASP.NET y Visual Studio, o Apache, MySQL y PHP. No obstante, podemos incidir un poco más en la comparación entre ASP.NET y PHP a partir de un artículo publicado por un profesional en la materia

-La más reciente versión de PHP, la 5, mantiene la eficiencia tradicional del lenguaje y añade funcionalidades orientadas a la orientación a objetos.

-Una gran ventaja de ASP.NET es el entorno Visual Studio; es una herramienta que facilita el proceso de generar código, ofrece un buen depurador y permite manipular bases de datos Oracle directamente.

-ASP.NET tiene unos mayores requerimientos de memoria en el servidor web.

-La conclusión del autor es que PHP, con la inclusión de funcionalidades orientadas a objetos y manteniendo su tradicional eficiencia es la mejor elección, particularmente con servidores Linux con el programa servidor Apache (Oracle.com, 2009)

En cuanto al editor gráfico a usar, tanto Photoshop, como Paint Shop Pro como GIMP cubrirían las necesidades del desarrollo a implementar; el inconveniente del Photoshop es un elevado coste, mientras que Paint Shop Pro tiene un precio bastante asequible y GIMP es una herramienta de código abierto y por tanto gratuita.

## 6. Especificación de estándares y normas

### Descripción de la tarea

*La realización de esta tarea permite considerar las referencias al sistema a implementar desde el punto de vista de estándares, normativas, leyes o recomendaciones que deben tenerse en cuenta a lo largo de todo el desarrollo. (Análisis del Sistema de Información, Métrica 3: pág. 8)*

### Desarrollo de la tarea

En el desarrollo de una aplicación web en España deben tenerse en cuenta varias leyes y/o normas. En primer lugar, para todas aquellas aplicaciones web dedicadas al comercio electrónico o a aquellas que sean parte de una actividad económica, la Ley de Servicios de la Sociedad de la Información (LSSI) especifica que debe mostrarse en la página web la siguiente información:

-Denominación social, NIF, domicilio y dirección de correo electrónico, teléfono o fax.

-Datos de inscripción registral.

-Códigos de conducta a que esté adherida.

-Precios de los productos o servicios que ofrecen.

En el caso del sistema a desarrollar en este proyecto no es necesario considerar los aspectos regulados por ley, por tratarse de un prototipo (y no de un desarrollo final) y porque el portal web en el que está previsto su uso, <http://www.our.com>, no tiene funciones de comercio electrónico ni ninguna actividad económica. No obstante, si este último aspecto cambiara (por ejemplo, si se empieza a cobrar a los usuarios por algún servicio, o se establecieran dos categorías, de usuarios normales y de pago con más servicios; también en el caso de que hubiera publicidad que reportara beneficio económico a los dueños del portal web), el portal entraría de pleno en los regulados por la LSSI.

Otra normativa a considerar, la Ley Orgánica de Protección de Datos (LOPD), que regula los procedimientos a seguir en el tratamiento de datos de carácter personal, si que tiene influencia sobre el portal en el que se alojaría un desarrollo completo del prototipo a construir, pero queda fuera de la influencia de los objetivos a alcanzar en este proyecto.







En posteriores apartados se seguirá descomponiendo el diagrama obtenido y se obtendrá un diccionario de datos que especifique los flujos de datos aparecidos en el diagrama.

Cabe reseñar que, con la aparición de los subsistemas, es posible catalogar cada requisito del Catálogo de Requisitos dentro de su subsistema correspondiente. No obstante, y como en paralelo a la construcción del desarrollo pueden ir surgiendo nuevos requisitos, y para evitar la redundancia, se incluirá una versión definitiva del Catálogo de Requisitos en la sección de anexos de este documento.

## 8. Elaboración del Modelo Conceptual de Datos

### Descripción de la tarea

*El objetivo de esta tarea es identificar y definir las entidades que quedan dentro del ámbito del sistema de información, los atributos de cada entidad (diferenciando aquellos que sean identificadores de la entidad), los dominios de los atributos y las relaciones existentes entre las entidades, indicando cardinalidades mínimas y máximas. (Análisis del Sistema de Información, Métrica 3: pág. 22)*

### Desarrollo de la tarea

Se va a realizar el Modelo Conceptual de Datos del sistema mediante modelos Entidad/Relación. Dada la complejidad del sistema, lo cual dificultaría la visibilidad e inteligibilidad si se realizara un solo diagrama, se va a modelar el sistema mediante varios diagramas, particionando el modelo global en subesquemas según la Entidad principal implicada; posteriormente se van a describir las reglas de negocio del modelo, también llamadas restricciones semánticas. Para cada entidad se definen sus atributos (en subrayado aquel que sea identificador) y para cada relación las cardinalidades de las entidades que la forman:

- **Subesquema de usuarios**

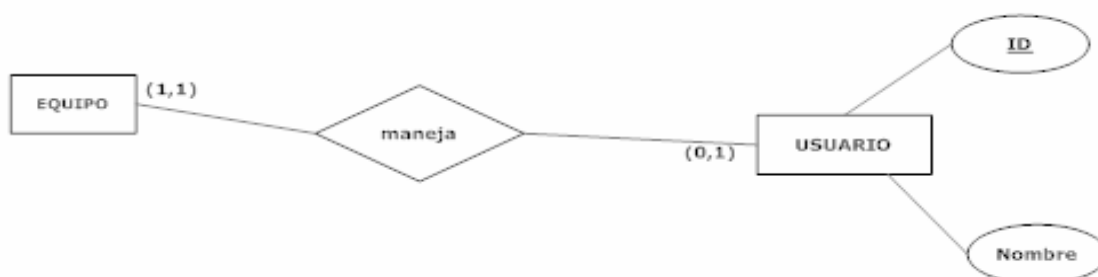
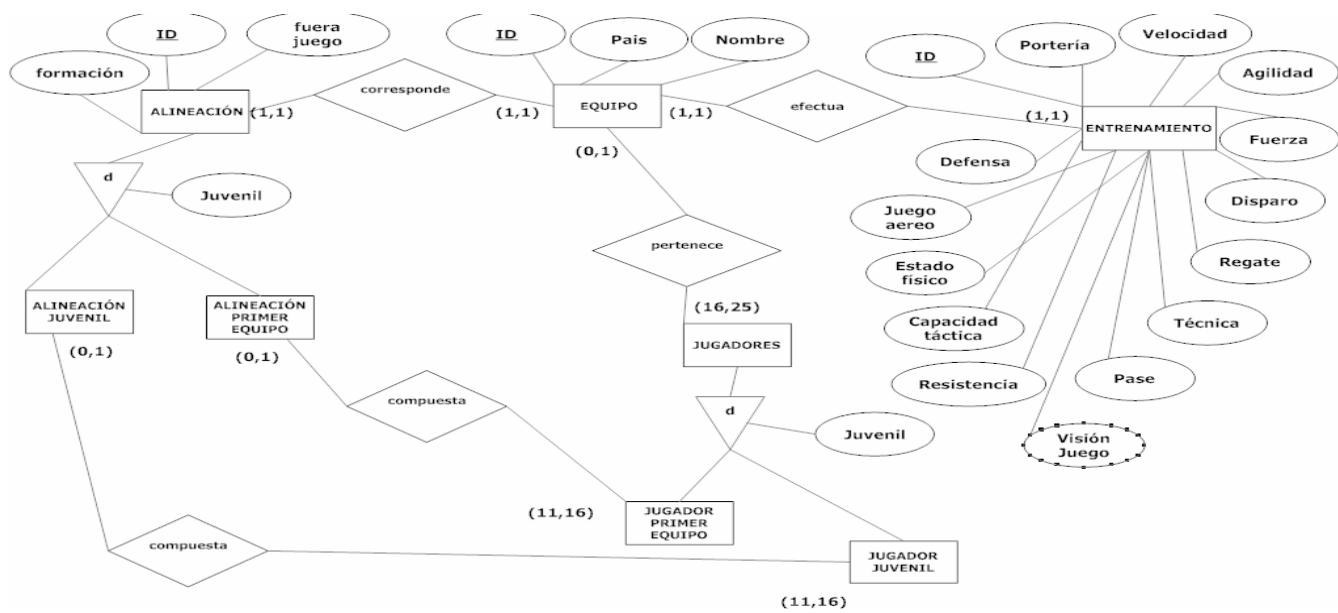


Fig 9. Diagrama Entidad-Relación 1

- **Subesquema de equipo, jugadores, alineación y entrenamiento**



**Fig 10. Diagrama Entidad-Relación 2**

Por razones de presentación se definen los atributos de la entidad JUGADORES en este apartado: id(clave primaria), nombre, apellidos, juvenil, pais, edad, temporadas contrato, sueldo, valor, posición, media, dorsal, partidos jugados, goles, tarjetas amarillas, tarjetas rojas, transferible, semanas lesionado, estado físico, moral, portería, velocidad, agilidad, fuerza, técnica, disparo, regate, pase, visión juego, resistencia, defensa, juego aéreo, capacidad táctica, talento especial, portería potencial, velocidad potencial, agilidad potencial, fuerza potencial, técnica potencial, disparo potencial, regate potencial, pase potencial, visión juego potencial, resistencia potencial, defensa potencial, juego aéreo potencial, capacidad táctica potencial, portería semana anterior, velocidad semana anterior, agilidad semana anterior, fuerza semana anterior, técnica semana anterior, disparo semana anterior, regate semana anterior, pase semana anterior, visión juego semana anterior, resistencia semana anterior, defensa semana anterior, juego aéreo semana anterior, capacidad táctica semana anterior, portería tapado, velocidad tapado, agilidad tapado, fuerza tapado, técnica tapado, disparo tapado, regate tapado, pase tapado, visión juego tapado, resistencia tapado, defensa tapado, juego aéreo tapado y capacidad táctica tapado.

- **Subesquema ofertas traspaso**

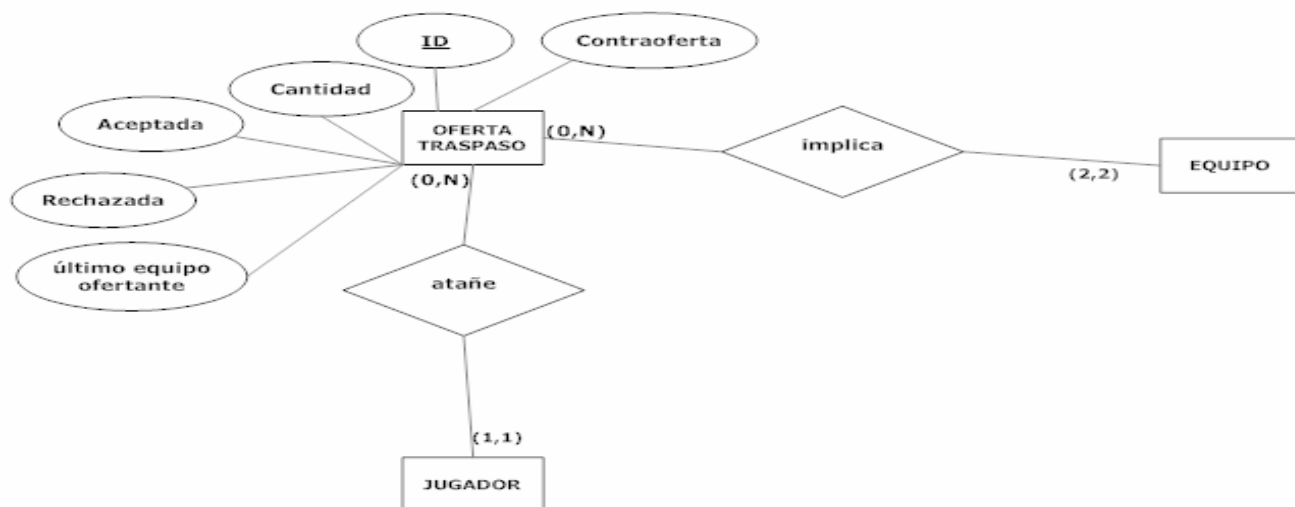


Fig 11. Diagrama Entidad-Relación 3

- Subesquema partidos, liga y clasificación

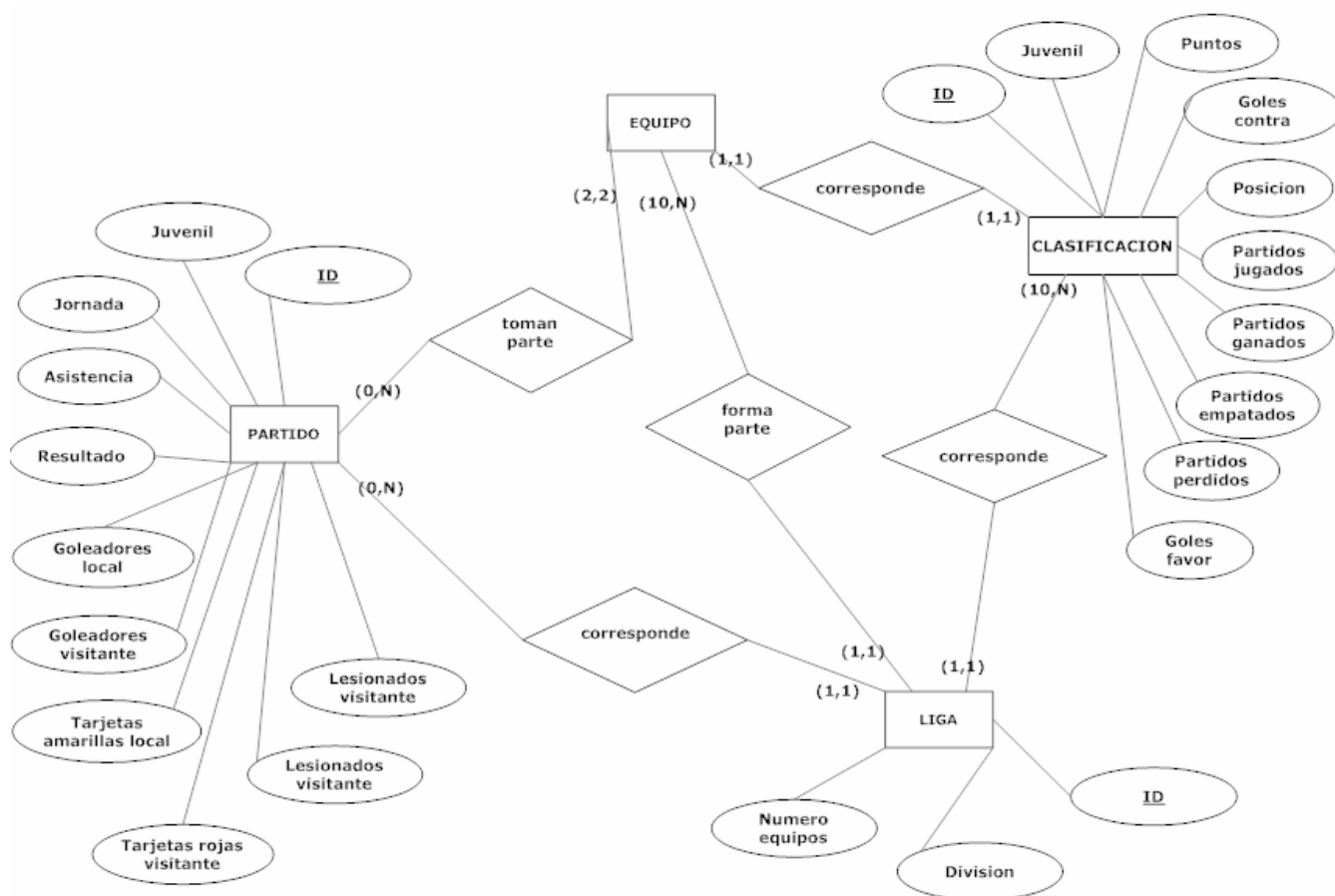


Fig 12. Diagrama Entidad-Relación 4

- Subesquema empleados

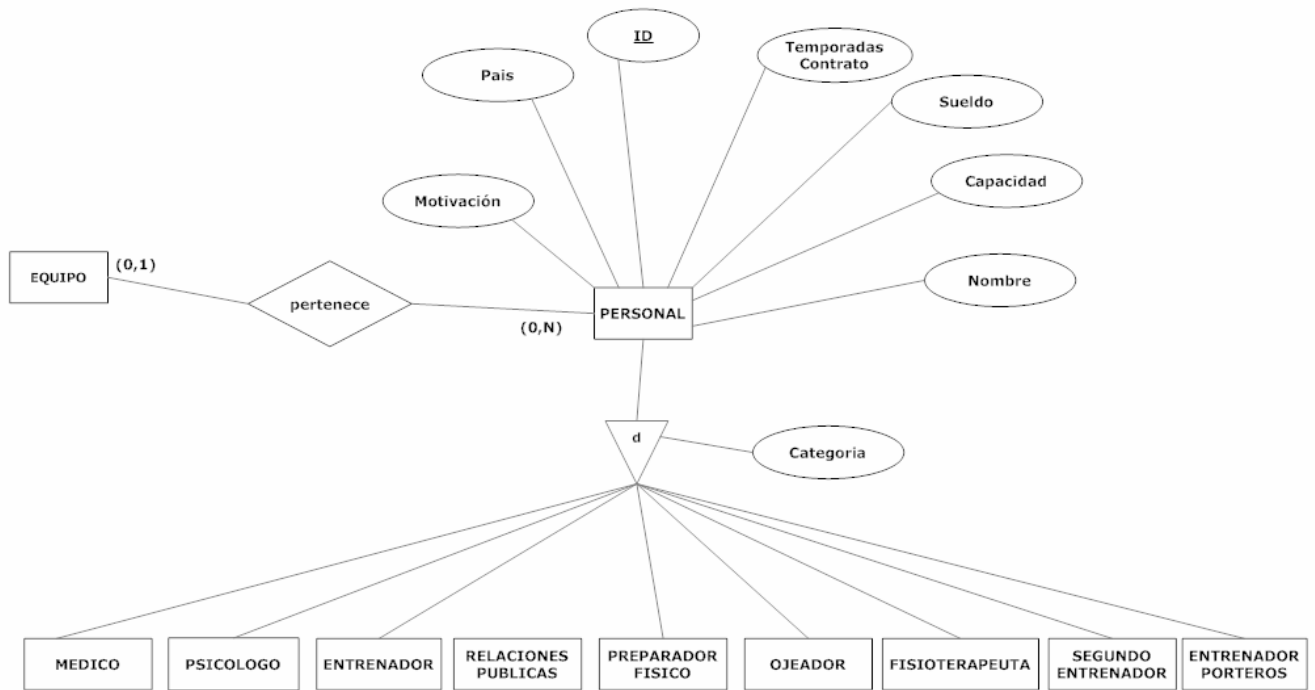


Fig 13. Diagrama Entidad-Relación 5

- Subesquema economía

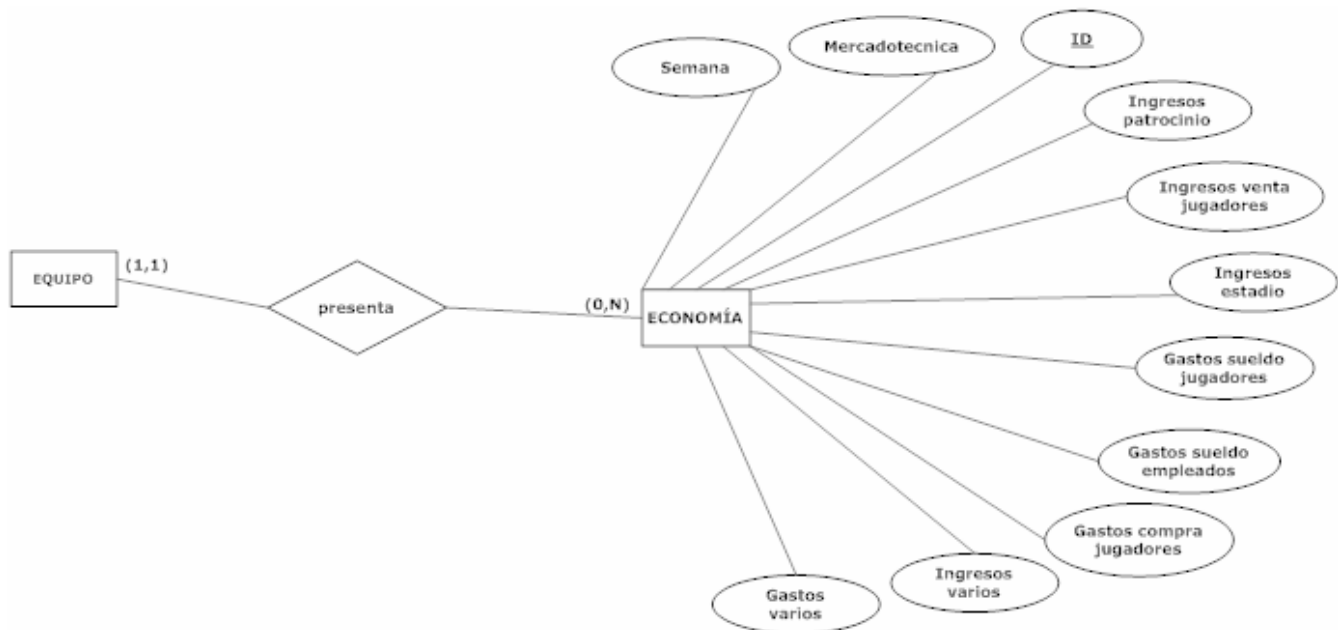


Fig 14. Diagrama Entidad-Relación 6

- Subesquema patrocinio

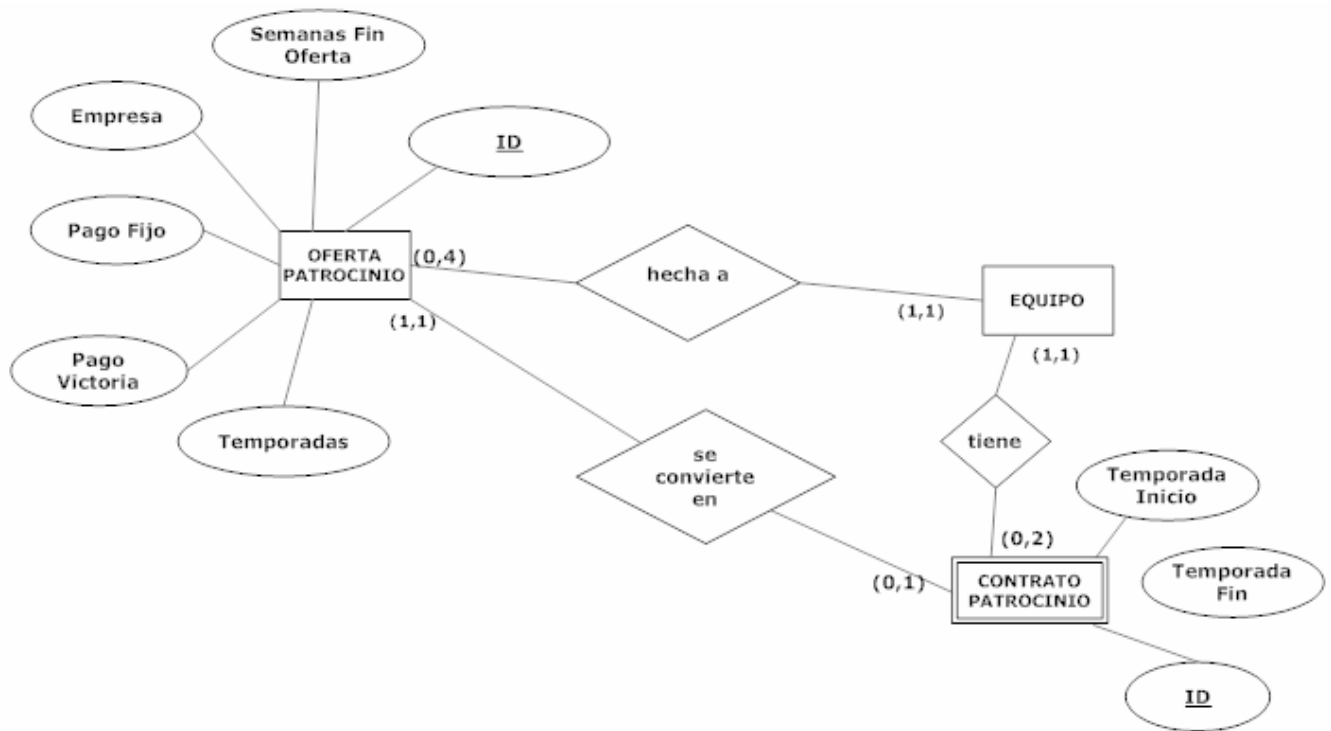


Fig 15. Diagrama Entidad-Relación 7

Observaciones:

La entidad CONTRATO PATROCINIO es débil porque depende de la existencia de una ocurrencia de la entidad OFERTA PATROCINIO para existir.

**Subesquema estadio**

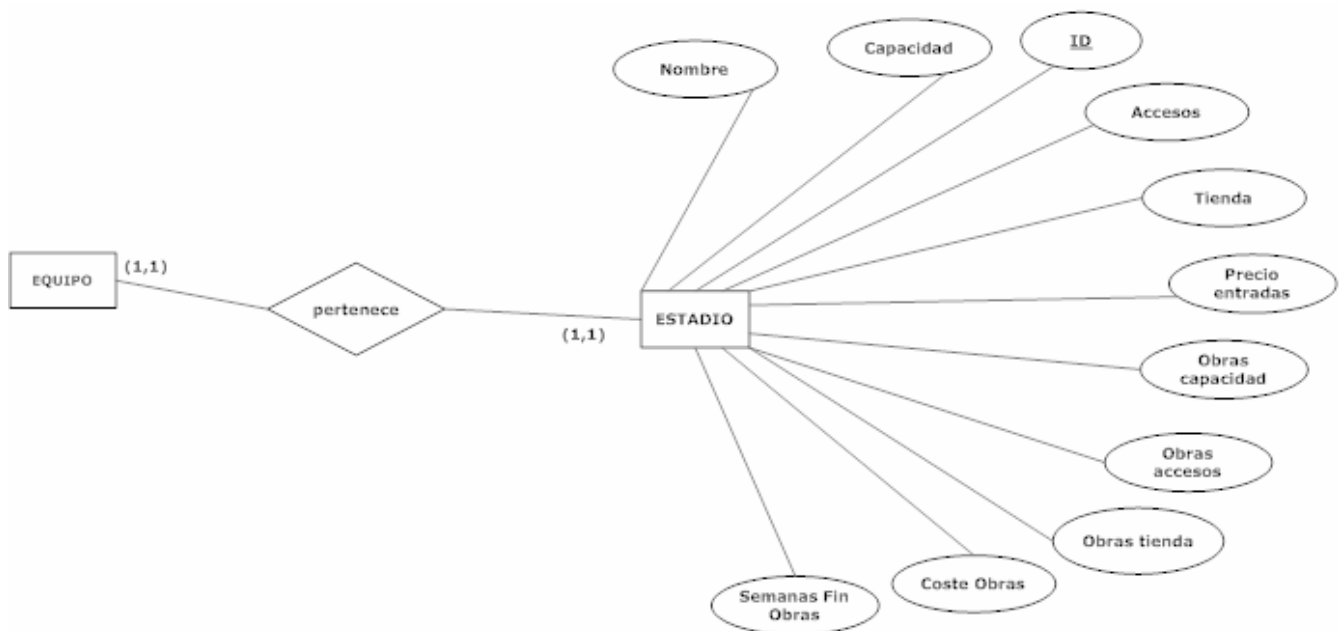


Fig 16. Diagrama Entidad-Relación 8

**Subesquema créditos**

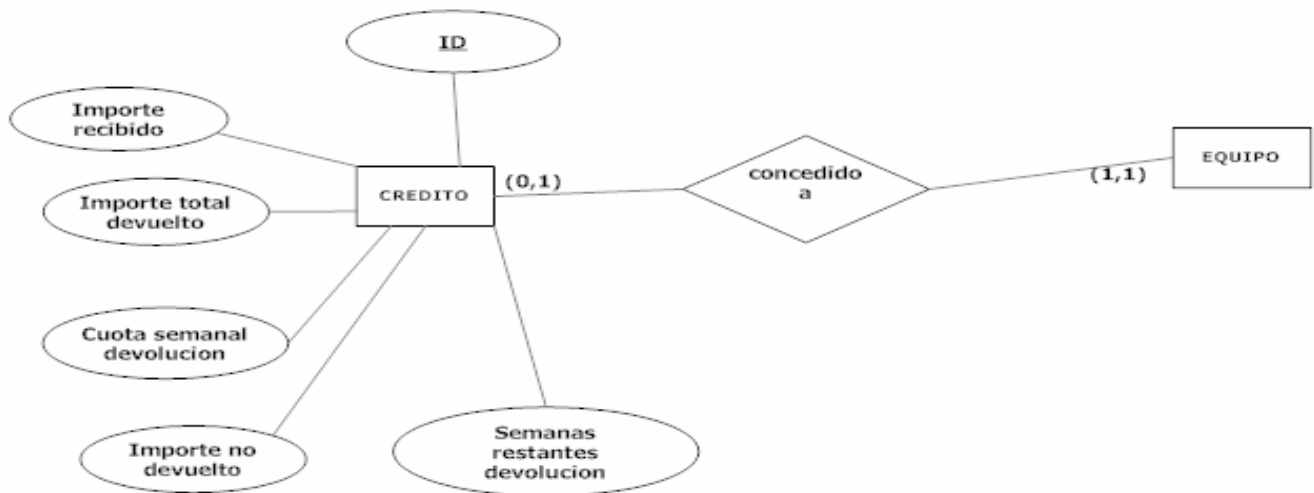


Fig 17. Diagrama Entidad-Relación 9

## Reglas de negocio

### Jugadores

RN1. Los jugadores vienen definidos por una serie de atributos:

RN1.1. Equipo al que pertenecen.

RN1.2. País de origen.

RN1.3 Un nombre.

RN1.4. Unos apellidos.

RN1.5 Una fecha de nacimiento.

RN1.6 Una edad de retirada.

RN1.7. Altura (en centímetros).

RN1.9. Pie natural (derecho, izquierdo, ambidiestro).

RN1.10. Un estado físico actual.

RN1.11. Moral actual.

RN1.12. Habilidad de portería.

RN1.13. Habilidad de portería potencial.

RN1.14. Velocidad.

RN1.15 Velocidad potencial.

RN1.16. Agilidad.

RN1.17. Agilidad potencial.

RN1.18. Fuerza.

RN1.19. Fuerza potencial.

RN1.20. Técnica.

RN1.21. Técnica potencial.

RN1.22. Disparo.

RN1.23. Disparo potencial.

RN1.24. Regate.

RN1.25. Regate potencial.

RN1.26. Pase.

RN1.27. Pase potencial.

RN1.28. Visión de juego.

RN1.29. Visión de juego potencial.

RN1.30. Resistencia.

RN1.31. Resistencia potencial.

RN1.32. Defensa.

RN1.33. Defensa potencial.

- RN1.34. Juego aéreo.
- RN1.35. Juego aéreo potencial.
- RN1.36. Capacidad táctica.
- RN1.37. Capacidad táctica potencial.
- RN1.38. Talento especial.
- RN1.39. Si se trata de un jugador juvenil o no.
- RN1.40. Si es no juvenil, sueldo percibido por semana.
- RN1.41. Si es no juvenil, temporadas de contrato restantes con su equipo actual.
- RN1.42. Si es no juvenil, valor económico del jugador.
- RN1.43. Posición ideal de juego.
- RN1.44. Media de sus atributos.
- RN1.45. Dorsal.
- RN1.46. Goles marcados en la temporada vigente.
- RN1.47. Tarjetas amarillas recibidas en la temporada vigente.
- RN1.48. Tarjetas rojas recibidas en la temporada vigente.
- RN1.49. Partidos jugados en la temporada vigente.
- RN1.50. Si está declarado como transferible o no.
- RN1.51. Si hay alguna oferta de otro equipo en ese momento por ese jugador.
- RN1.52. Si esta lesionado o no.
- RN1.53. Si esta lesionado, cuantas semanas dura su lesión.
- RN1.53. Por un tipo de cara.
- RN2. Un jugador con el valor para el atributo "Equipo" igual a 0 es un jugador libre, fichable en cualquier momento por su valor de mercado.
- RN3. Los valores de cada parámetro que tienen su equivalente "parámetro potencial" nunca podrán superar al valor potencial de ese parámetro.
- RN4. Cuando un administrador ejecute un proceso de "actualización semanal" , y si no están lesionados, se actualizan los siguientes atributos de los jugadores.
  - RN4.1. Estado físico actual.
  - RN4.2. Moral actual.
  - RN4.3. Habilidad de portería
  - RN4.4. Velocidad
  - RN4.5. Agilidad.
  - RN4.6. Fuerza.
  - RN4.7. Técnica.
  - RN4.8. Disparo.
  - RN4.9 Regate.
  - RN4.10. Pase.
  - RN4.11. Visión de juego
  - RN4.12. Resistencia
  - RN4.13. Defensa
  - RN4.14. Juego aéreo
  - RN4.15. Capacidad táctica
  - RN4.16. Media de sus atributos
  - RN4.17. Valor económico del jugador
- RN5. Al finalizar la disputa de un partido, todos los jugadores pertenecientes a cada equipo que haya jugado el partido tendrán la actualización de los siguientes atributos:
  - RN5.1. Partidos jugados
  - RN5.2. Goles marcados
  - RN5.3. Tarjetas amarillas
  - RN5.4. Tarjetas rojas
  - RN5.5. Si esta lesionado o no.
  - RN5.6. Si esta lesionado, cuantas semanas dura su lesión.
- RN6. Si un jugador pasa de un equipo juvenil al primer equipo (deja de ser juvenil) no puede jamás volver a adquirir la categoría de jugador juvenil.
- RN7. El usuario podrá ofrecer una renovación de contrato en cualquier momento a cualquiera de sus jugadores (no juveniles), que será aceptada en función de la oferta realizada y el nivel del jugador.
- RN8. El usuario sólo podrá vender un jugador(normal) o despedir un jugador juvenil cuando los equipos respectivos tengan más de 16 jugadores.



RN9.- Un JUGADOR cuyo atributo "juvenil" es NULL tendrá asimismo NULL los atributos "portería tapado", "velocidad tapado", "agilidad tapado", "fuerza tapado", "técnica tapado", "disparo tapado", "regate tapado", "pase tapado", "visión juego tapado", "resistencia tapado", "defensa tapado", "juego aéreo tapado" y "capacidad táctica tapado".

RN10.- Un JUGADOR cuyo atributo "juvenil" sea 1 tendrá NULL los atributos "temporadas contrato" y "sueldo", además de tener un valor entre 17 y 19 (inclusive) en el atributo "edad".

## **Equipos**

RN10.Los equipos están definidos por:

RN10.1. Un usuario que los gestiona.

RN10.2. Una liga de la que forman parte.

RN10.3. Los jugadores(normales y juveniles) que los componen.

RN10.4. Un estadio de su propiedad.

RN10.5. Unos empleados.

RN10.6. Un entrenamiento (individual y de equipo) programado.

RN10.7. Una alineación (normal y juvenil).

RN10.8. Un estado de su economía.

RN10.9. Los créditos bancarios que tengan solicitados en ese momento.

RN10.10. Si tienen o no contrato de patrocinio.

RN10.11. Una clasificación (normal y juvenil) en la liga que disputan.

## **Empleados**

RN11.Los empleados están definidos por:

RN11.1. Un nombre

RN11.2. Unos apellidos

RN11.3. Un país de origen

RN11.4. Fecha de nacimiento

RN11.5. La categoría profesional a la que pertenecen

RN11.6. Una capacidad laboral

RN11.7. Motivación actual

RN11.8. Equipo al que pertenecen

RN11.9. Sueldo semanal que perciben

RN11.10. Temporadas de contrato que le restan con su actual equipo.

RN12. Un empleado podrá ser despedido por el usuario en cualquier momento, y el despido será penalizado con una cantidad calculada en función del sueldo y de las temporadas de contrato del jugador.

RN13. El usuario podrá ofrecer una renovación de contrato en cualquier momento a cualquiera de sus empleados, que será aceptada o no en función de la oferta realizada y el nivel del empleado.

RN14. Los empleados disponibles para fichar por un equipo, denominados candidatos, tienen un valor 0 en sus atributos "sueldo" y "temporadas de contrato".

## **Alineación**

RN15.Una alineación está definida por:

RN15.1. Los jugadores que la componen

RN15.2. Un equipo al que hace referencia

RN15.3. Una disposición táctica

RN15.4. Una formación

RN15.5. Un tipo de defensa

RN16.Al finalizar cada partido, si un equipo ha sufrido alguna lesión o uno de sus jugadores ha visto una tarjeta que supone sanción, se modificará automáticamente la alineación del equipo introduciendo jugadores disponibles por aquellos lesionado/s y/o sancionado/s. El motivo es asegurar que los equipos manejados por el sistema siempre cuenten con una alineación coherente.

## **Estadio**

RN17.Un estadio está definido por:

RN17.1. Equipo al que pertenece.

RN17.2. Una categoría.

RN17.3. Una capacidad.

- RN17.4. Nivel de accesos.
- RN17.5. Tienda del club.
- RN17.6. Precio de las entradas.
- RN17.7. Si existen obras en marcha que afecten a la capacidad y en cuanta cantidad.
- RN17.8. Si existen obras en marcha que afecten a los accesos.
- RN17.9. Si existen obras en marcha que afecten a la tienda del estadio.
- RN17.10. El coste de las obras que están en marcha.
- RN17.11. Semanas para la finalización de las obras en marcha
- RN18. El precio de las entradas está limitado en un valor entre 1 y 50.
- RN19. Las semanas y el coste de las obras se calculan en función del tipo de obra.

### **Partido**

RN20. Un partido está definido por:

- RN20.1. La competición a la que pertenece.
- RN20.2. Si se trata de un partido de equipos juveniles.
- RN20.3. La jornada de la que forma parte.
- RN20.4. Los dos equipos que se enfrentan.
- RN20.5. Si ya ha sido disputado o no.
- RN20.6. La asistencia en espectadores que tuvo el partido.
- RN20.7. Un resultado en caso de haber sido ya disputado.
- RN20.8. Los goleadores de cada equipo.
- RN20.9. Los jugadores que han visto tarjeta amarilla y roja en cada equipo.
- RN20.10. Los minutos que han disputado los jugadores de cada equipo.

RN21: La asistencia se calcula en función de la posición en la clasificación del equipo local y visitante y de los precios de las entradas y accesos que tenga el estadio del equipo local, además de la habilidad del empleado de categoría "Relaciones Públicas" del equipo local.

### **Crédito bancario**

RN22. Un crédito se define mediante:

- RN22.1. El equipo que lo ha recibido.
- RN22.2. Una cantidad recibida.
- RN22.3. El interés de devolución.
- RN22.4. El número de semanas que restan hasta la devolución forzosa del crédito.
- RN22.5. La cantidad que ha de devolverse semanalmente.
- RN22.6. La cantidad que se ha devuelto hasta el momento.

RN23: Un equipo no puede pedir un crédito si tiene uno vigente en ese momento.

RN24: El valor del atributo "semanas" estará siempre en un rango entre 1 y 16 y "cantidad recibida" entre 1 y 1500000.

### **Ofertas patrocinio**

RN25. Una oferta de patrocinio se define por los siguientes atributos:

- RN25.1. El nombre de la empresa ofertante.
- RN25.2. El equipo que recibe la oferta.
- RN25.3. Una cantidad semanal fija a cobrar por el equipo.
- RN25.4. Cantidad a recibir a final de temporada si el equipo cumple el objetivo.
- RN25.5. Cantidad a recibir por cada partido ganado.
- RN25.6. Temporadas en las que estaría vigente el acuerdo de patrocinio.

RN26: En cada momento cada equipo no puede tener más de cuatro ofertas de patrocinio.

RN27: Si se acepta una oferta y ya hay un contrato de patrocinio vigente, se coloca como próximo patrocinador y pasa a ser el patrocinador cuando acaba el contrato del patrocinador vigente.

RN28: Si se acepta una oferta de patrocinio y el equipo ya tiene patrocinador vigente y un acuerdo para el próximo patrocinio, se cancela el acuerdo para el próximo patrocinio sin ninguna penalización.

### **Contratos patrocinio**

RN29. Está definido por:

- RN29.1. Un equipo.
- RN29.2. Una oferta de patrocinio.
- RN29.3. La temporada en la que comienza el patrocinio.
- RN29.4. La temporada en la que acaba el patrocinio.

RN30: Si un equipo decide romper un contrato de patrocinio vigente se penaliza con cantidad determinada por las temporadas y el montante del patrocinio.

### **Ofertas de traspaso**

RN31. Están definidas por:

- RN31.1. El jugador al que se refiere la oferta.
- RN31.2. El equipo al que pertenece el jugador.
- RN31.3. El equipo que pretende al jugador.
- RN31.4. La cantidad que se ofrece.
- RN31.5. El importe de la contraoferta.
- RN31.6. Si ha sido aceptada o rechazada.

RN32: Una oferta de traspaso solo puede referirse a un jugador normal (no juvenil) y perteneciente a un equipo.

### **Liga**

RN33. Una liga está definida por:

- RN33.1. División.
- RN33.2. Número de liga.
- RN33.3. Número de equipos que la forman.

### **Clasificación**

RN34. Esta definida por:

- RN34.1. Liga a la que se refiere.
- RN34.2. Posición de cada equipo.
- RN34.3. Puntos conseguidos por cada equipo.
- RN34.4. Partidos jugados por cada equipo.
- RN34.5. Partidos ganados por cada equipo.
- RN34.6. Partidos empatados por cada equipo.
- RN34.7. Partidos perdidos por cada equipo.
- RN34.8. Goles marcados por cada equipo.
- RN34.9. Goles encajados por cada equipo.

RN35: En una clasificación se ordenan a los equipos según los puntos conseguidos, y en caso de empate a punto se siguen los siguientes criterios (de mayor a menor preferencia).

- RN35.1. Puntos conseguidos en los partidos entre los equipos implicados en el empate.
- RN35.2. Diferencia de goles en los partidos entre los equipos implicados en el empate.
- RN35.3. Diferencia de goles general.
- RN35.4. Sorteo.

RN36.- La suma de los atributos "partidos ganados", "partidos perdidos" y "partidos empatados" debe ser exactamente igual al valor del atributo "partidos jugados" en la entidad CLASIFICACIÓN.

### **Economía**

RN37. Esta definida por:

- RN37.1. Un equipo
- RN37.2. Semana a la que se refiere.
- RN37.3. Ingresos del equipo durante esa semana en concepto de mercadotecnia
- RN37.4. Ingresos del equipo durante esa semana en concepto de patrocinio
- RN37.5. Ingresos del equipo durante esa semana por venta de jugadores
- RN37.6. Ingresos del equipo durante esa semana por venta de entradas
- RN37.7. Ingresos del equipo durante por asuntos varios
- RN37.8. Gastos del equipo durante esa semana en sueldos de jugadores
- RN37.9. Gastos del equipo durante esa semana en sueldos de empleados
- RN37.10. Gastos del equipo durante esa semana en compra de jugadores
- RN37.11. Gastos del equipo durante por asuntos varios

# 9. Obtención del Modelo de Procesos del Sistema

## Descripción de la tarea

*En esta tarea se lleva a cabo la descripción de los subsistemas definidos en la actividad Determinación de Subsistemas de análisis mediante la descomposición en sucesivos niveles de procesos. (Análisis del Sistema de Información, Métrica 3: pág. 26)*

## Desarrollo de la tarea

Se realiza la descomposición de los subsistemas obtenidos en el diagrama construido en la tarea ASI 3.1 (Figura 8), describiendo asimismo la estructura de los flujos y de los almacenes de datos. Para cada proceso primitivo (proceso que no se descompone en otros procesos; Técnicas y Prácticas, Métrica 3: pág. 34) se agrega una especificación del mismo:

En la definición de los flujos de datos se sigue el siguiente esquema:

**Dataflow:**  
**From-symbol:**  
**To-symbol:**  
**Data within flow:**

“Dataflow” presenta el nombre del flujo en el diagrama; “From-symbol” indica el nombre del componente (proceso o entidad externa) desde donde para la comunicación que implica al flujo de datos; “To-symbol” indica el nombre del componente (proceso o entidad externa) del destino del flujo de datos; por último, “Data within flow” indica los datos implicados en ese flujo determinado (sólo si es un dato compuesta por varios componentes; en caso contrario se deja el campo en blanco).

En la descripción de un almacén de datos se sigue el siguiente esquema:

**Datastore:**  
**Input flows**

**Output flows**

**Data within store:**

“Datastore” indica el nombre del almacén; en “Input flows” se listan todos los procesos desde los que parten flujos de datos que llegan al almacén el diagrama correspondiente; “Output flows” lista los procesos que reciben un flujo de datos que parte desde dicho almacén; en “Data within store” se incluyen todos los datos del almacén que están presentes en el diagrama.

En la especificación de un proceso se sigue el siguiente esquema:

**Process:**  
**Minispec or Description:**  
**PRE**  
**POST**

**Purpose:**

**Input flows**

**Output flows**

“Process” es el nombre del proceso especificado; en “Minispec or Description” se incluyen o bien una descripción del proceso en lenguaje natural o una serie de precondiciones (“PRE”) y postcondiciones (“POST”) que definen el estado en el que puede ocurrir el proceso especificado y la forma en que la ocurrencia del

proceso lo deja; "Purpose" indica el propósito general del proceso"; por último, en "Input flows" y "Output flows" se listan los flujos de datos que parten (Output) y llegan(Input) al proceso, así como el otro componente del diagrama con relación con ese flujo de datos.



**Fig 18. Diagrama de Flujos de Datos 1.**

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### **Process: Validar Login**

#### **Minispec or Description:**

PRE1. Un usuario se identifica ante el sistema.

POST1. El sistema compara los datos de identificación con lo almacenado en USUARIOS y devuelve una respuesta al usuario; si no hay ningún usuario registrado que coincida con los datos introducidos, se genera además el identificador del equipo a asignar al usuario por si este lo solicitara.

**Purpose:** Se verifican los datos de identificación introducidos por el usuario.

#### **Input flows**

datos login  
coming from USUARIO[External]

coming from USUARIOS[Data Store]

#### **Output flows**

respuesta login  
going to USUARIO[External]  
usuario sin equipo  
going to ASIGNAR EQUIPO[Process]

### **Process: Asignar Equipo**

#### **Minispec or Description:**

PRE1. Un usuario ha realizado un intento de login.

POST1. El sistema le ha informado de que no tiene un equipo asignado.

PRE2. El usuario pide que se le asigne un equipo.

POST2. El sistema recibe cual es el equipo sin usuario (controlado por el mismo sistema) que le debe asignar y actualiza USUARIOS.

**Purpose:** Se asigna un equipo al usuario..

#### **Input flows**

petición equipo  
coming from USUARIO[External]  
equipo sin usuario  
coming from VALIDAR LOGIN[Process]

#### **Output flows**

going to USUARIOS[Data Store]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** datos login

**From-symbol:** USUARIO[External]

**To-symbol:** Validar Login[Process]

**Data within flow:** {usuario+clave}

**Dataflow:** respuesta login

**From-symbol:** Validar Login[Process]

**To-symbol:** USUARIO[External]

**Data within flow:**

**Dataflow:** usuario sin equipo

**From-symbol:** USUARIO[External]

**To-symbol:** Cancelar Crédito[Process]

**Data within flow:**

**Dataflow:** petición equipo

**From-symbol:** USUARIO[External]

**To-symbol:** Asignar Equipo[Process]

**Data within flow:**

**Dataflow:** consulta balance semanal

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Economía[Process]

**Data within flow:**

**Dataflow:** balance semanal

**From-symbol:** Gestionar Economía[Process]

**To-symbol:** USUARIO[External]

**Data within flow:** {semana+ingresos\_traspos+ingresos\_estadio+ingresos\_patrocinadores+gastos\_sueldos+gastos\_fichajes+gastos\_obras+otros\_gastos}

**Dataflow:** consulta patrocinadores

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Economía[Process]

**Data within flow:**

**Dataflow:** datos patrocinadores

**From-symbol:** Gestionar Economía[Process]

**To-symbol:** USUARIO[External]

**Data within flow:** {id\_contrato\_patrocinio+pago\_semanal+pago\_victoria+pago\_objetivo+temp\_contrato+temp\_inicio}

**Dataflow:** Id contrato patrocinio eliminado

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Economía[Process]

**Data within flow:**

**Dataflow:** consulta ofertas patrocinio

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Economía[Process]

**Data within flow:**

**Dataflow:** datos ofertas patrocinio

**From-symbol:** Gestionar Economía[Process]

**To-symbol:** USUARIO[External]

**Data within flow:** {id\_oferta\_patrocinio+ equipo+ pago\_semanal+ pago\_victoria+ pago\_objetivo+temp\_contrato+contestada}

**Dataflow:** Id oferta patrocinio aceptada

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Economía[Process]

**Data within flow:**

**Dataflow:** Id oferta patrocinio rechazada

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Economía[Process]

**Data within flow:**

**Dataflow:** consulta crédito

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Economía[Process]

**Data within flow:**

**Dataflow:** datos crédito vigente

**From-symbol:** Gestionar Economía[Process]

**To-symbol:** USUARIO[External]



**Data within flow:** {id\_crédito+cantidad\_prestada+ cantidad\_devuelta  
semanas\_devolución+ cuota\_semanal}

+

**Dataflow:** datos crédito pedido  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Economía[Process]  
**Data within flow:** {cantidad+semanas\_devolución}

**Dataflow:** cancelación crédito  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Economía[Process]  
**Data within flow:**

**Dataflow:** consulta clasificación  
**From-symbol:** USUARIO[External],  
**To-symbol:** Consultar Datos Competición[Process]  
**Data within flow:**

**Dataflow:** clasificación  
**From-symbol:** Consultar Datos Competición[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {nombre\_equipo+puesto+puntos+goles\_favor+goles\_contra+partidos\_jugados+  
partidos\_ganados+partidos\_empatados+partidos\_perdidos}

**Dataflow:** consulta max goleadores  
**From-symbol:** USUARIO[External]  
**To-symbol:** Consultar Datos Competición[Process]  
**Data within flow:**

**Dataflow:** max goleadores  
**From-symbol:** Consultar Datos Competición[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {nombre\_jugador+goles\_marcados}

**Dataflow:** consulta calendario  
**From-symbol:** USUARIO[External]  
**To-symbol:** Consultar Datos Competición[Process]  
**Data within flow:**

**Dataflow:** calendario  
**From-symbol:** Consultar Datos Competición[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {jornada+nombre\_equipo\_local+nombre\_equipo\_visitante+goles\_local+goles\_visitante}

**Dataflow:** consulta último jugado  
**From-symbol:** USUARIO[External]  
**To-symbol:** Consultar Datos Competición[Process]  
**Data within flow:**

**Dataflow:** datos último partido jugado  
**From-symbol:** Consultar Datos Competición[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:**  
{jornada+nombre\_equipo\_local+nombre\_equipo\_visitante+goles\_local+goles\_visitante+goleadores+tarj  
etas+lesionados}

**Dataflow:** consulta datos estadio  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Estadio[Process]  
**Data within flow:**

**Dataflow:** datos estadio  
**From-symbol:** Gestionar Estadio[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {id\_estadio+equipo+ nombre+ capacidad+ valor\_entradas+ nivel\_accesos+ nivel\_tienda+obras\_capacidad+obras\_accesos+obras\_tienda+dias\_obras}

**Dataflow:** datos obras estadio  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Estadio[Process]  
**Data within flow:** : {id\_estadio+obras\_capacidad+obras\_accesos+obras\_tienda}

**Dataflow:** nuevo valor entradas  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Estadio[Process]  
**Data within flow:**

**Dataflow:** id jugador libre  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Transacciones[Process]  
**Data within flow:**

**Dataflow:** consulta jugadores libres  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Transacciones[Process]  
**Data within flow:**

**Dataflow:** datos jugadores libres  
**From-symbol:** Gestionar Transacciones[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {id\_jug+nombre+media+posicion+edad+pais+valor+sueldo+años\_contrato}

**Dataflow:** petición búsqueda  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Transacciones[Process]  
**Data within flow:**

**Dataflow:** resultado búsqueda  
**From-symbol:** Gestionar Transacciones[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {id\_jug+nombre+media+posicion+edad+pais+valor+sueldo+años\_contrato+transferible}

**Dataflow:** consulta equipos  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Transacciones[Process]  
**Data within flow:**

**Dataflow:** equipos  
**From-symbol:** Gestionar Transacciones[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {@id\_eq+nombre}

**Dataflow:** consulta jugadores equipo  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Transacciones[Process]  
**Data within flow:**

**Dataflow:** jugadores equipo  
**From-symbol:** Gestionar Transacciones[Process]  
**To-symbol:** USUARIO[External]

**Data within flow:** {id\_jug+nombre+media+posicion+edad+pais+valor+sueldo+años\_contrato+transferible}

**Dataflow:** consulta jug. trans.

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Transacciones[Process]

**Data within flow:**

**Dataflow:** datos jug. trans.

**From-symbol:** Gestionar Transacciones[Process]

**To-symbol:** USUARIO[External]

**Data within flow:**

{id\_jug+nombre+media+posicion+edad+pais+valor+sueldo+años\_contrato}

**Dataflow:** datos oferta

**From-symbol:** Gestionar Transacciones[Process]

**To-symbol:** USUARIO[External]

**Data within flow:** {id\_jug+cantidad}

**Dataflow:** info jugadores

**From-symbol:** Gestionar Parte Deportiva[Process]

**To-symbol:** USUARIO[External]

**Data within flow:** {@id\_jug+nombre+posicion+media+estado\_fisico+moral+pais+valor+sueldo+años\_contrato  
+juvenil}

**Dataflow:** consulta plantilla

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Parte Deportiva[Process]

**Data within flow:**

**Dataflow:** id jugador

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Parte Deportiva[Process]

**Data within flow:**

**Dataflow:** info jugador

**From-symbol:** Gestionar Parte Deportiva[Process]

**To-symbol:** USUARIO[External]

**Data within flow:** {@id\_jug+nombre+valor+sueldo+años\_contrato+cantidad\_oferta}

**Dataflow:** respuesta oferta renovación

**From-symbol:** Gestionar Parte Deportiva[Process]

**To-symbol:** USUARIO[External]

**Data within flow:**

**Dataflow:** id oferta traspaso rechazada

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Parte Deportiva[Process]

**Data within flow:**

**Dataflow:** orden subir jugador

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Parte Deportiva[Process]]

**Data within flow:**

**Dataflow:** id oferta traspaso aceptada

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Parte Deportiva[Process]

**Data within flow:**

**Dataflow:** contraoferta

**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Parte Deportiva[Process]  
**Data within flow:** {id\_oferta+cantidad}

**Dataflow:** orden poner transferible  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Parte Deportiva[Process]  
**Data within flow:**

**Dataflow:** orden poner no transferible  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Parte Deportiva[Process]  
**Data within flow:**

**Dataflow:** orden vender jugador  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Parte Deportiva[Process]  
**Data within flow:**

**Dataflow:** consulta entrenamiento  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Parte Deportiva[Process]  
**Data within flow:** {@id+usuario+equipo}

**Dataflow:** datos entrenamiento  
**From-symbol:** Gestionar Parte Deportiva[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {est\_fisico+port+vel+agil+fuer+tec+dis+reg+pase+vjue+res+def+jue\_ae+captact}

**Dataflow:** nuevo entrenamiento  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Parte Deportiva[Process]  
**Data within flow:** {equipo+est\_fisico+port+vel+agil+fuer+tec+dis+reg+pase+vjue+res+def+jue\_ae+captact}

**Dataflow:** consulta alineacion  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Parte Deportiva[Process]  
**Data within flow:**

**Dataflow:** datos alineacion  
**From-symbol:** Gestionar Parte Deportiva[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {formacion+fue\_jue+jug1+jug2+jug3+jug4+jug5+jug6+jug7+jug8+jug9+jug10+jug11}

**Dataflow:** nueva alineacion  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Parte Deportiva[Process]  
**Data within flow:** {equipo+formacion+fue\_jue+jug1+jug2+jug3+jug4+jug5+jug6+jug7+jug8+jug9+jug10+jug11}

**Dataflow:** info empleados  
**From-symbol:** Gestionar Empleados[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {@id+nombre+ edad+ pais+ categoria+ capacidad+ motivacion+ años\_contrato + sueldo}

**Dataflow:** consulta empleados  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Empleados[Process]  
**Data within flow:**

**Dataflow:** info candidatos  
**From-symbol:** Gestionar Empleados[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {@id+nombre+ edad+ pais+ categoria+ capacidad+ motivacion+ años\_contrato\_solicitados+ sueldo\_solicitado}

**Dataflow:** consulta candidatos  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Empleados[Process]  
**Data within flow:**

**Dataflow:** datos oferta renovación  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Empleados[Process]  
**Data within flow:** {años\_contrato\_ofrecidos+sueldo\_ofrecido}

**Dataflow:** datos oferta fichaje  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Empleados[Process]  
**Data within flow:** {años\_contrato+sueldo}

**Dataflow:** respuesta oferta renovacion  
**From-symbol:** Gestionar Empleados[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:**

**Dataflow:** respuesta oferta  
**From-symbol:** Gestionar Empleados[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:**

**Dataflow:** id empleado  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Empleados[Process]  
**Data within flow:**

**Dataflow:** id candidato  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Empleados[Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** USUARIOS

### **Input flows**

coming from Asignar Equipo[Process]

### **Output flows**

going to Validar Login[Process]

going to Gestionar Empleados[Process]

going to Gestionar Parte Deportiva[Process]

going to Gestionar Transacciones[Process]

going to Gestionar Estadio[Process]

going to Consultar Datos Competición[Process]

**Data within store:** {@id+usuario+equipo}

**Datastore:** EMPLEADOS

**Input flows**

coming from Gestionar Empleados[Process]

**Output flows**

going to Gestionar Empleados[Process]

**Data within store:** {@id+nombre+ edad+ pais+ categoria+ capacidad+ motivacion+ años\_contrato + sueldo }

**Datastore:** JUGADORES

**Input flows**

coming from Gestionar Parte Deportiva[Process]

coming from Gestionar Transacciones[Process]

**Output flows**

going to Gestionar Parte Deportiva[Process]

going to Gestionar Transacciones[Process]

going to Consultar Datos Competición[Process]

**Data within store:** {@id+equipo+temporadas\_contrato+sueldo+juvenil+transferible+valor+ goles\_marcados}

**Datastore:** OFERTAS TRASPASO

**Input flows**

coming from Gestionar Parte Deportiva[Process]

coming from Gestionar Transacciones[Process]

**Data within store:** {@id+jugador+equipo\_comprador+equipo\_vendedor+cantidad+contraoferta}

**Datastore:** EQUIPOS

**Input flows**

coming from Gestionar Parte Deportiva[Process]

coming from Gestionar Transacciones[Process]

coming from Gestionar Estadio[Process]

**Output flows**

going to Gestionar Parte Deportiva[Process]

going to Gestionar Transacciones[Process]

going to Gestionar Estadio[Process]

**Data within store:** {@id+nombre+saldo}

**Datastore:** ECONOMÍA

**Input flows**

coming from Gestionar Parte Deportiva[Process]

coming from Gestionar Transacciones[Process]

coming from Gestionar Estadio[Process]  
**Data within store:** {@id+ equipo+ semana+ ingresos\_ventajug+ gastos\_comprajug + gastos\_obras}

**Datastore:** ENTRENAMIENTO

**Input flows**

coming from Gestionar Parte Deportiva[Process]

**Output flows**

going to Gestionar Parte Deportiva[Process]

**Data within store:** {@id+ equipo+ estado\_fisico+ porteria+ velocidad+ agilidad+ fuerza+ tecnica+ disparo+ regate+ pase+ vision\_juego+ resistencia+ defensa+ juego\_aereo+ capacidad\_tactica}

**Datastore:** ALINEACIONES

**Input flows**

coming from Gestionar Parte Deportiva[Process]

**Output flows**

going to Gestionar Parte Deportiva[Process]

**Data within store:** {@id+ equipo+ formacion+ fuera\_juega+ juvenil+ jug1+ jug2+ jug3+ jug4+ jug5+ jug6+ jug7+ jug8+ jug9+ jug10+ jug11+ jug12+ jug13+ jug14+ jug15+ jug16}

**Datastore:** ESTADIOS

**Input flows**

coming from Gestionar Estadio[Process]

**Output flows**

going to Gestionar Estadio[Process]

**Data within store:** {@id+ equipo+ nombre+ capacidad+ valor\_entradas+ nivel\_aceesos+ nivel\_tienda+ obras\_capacidad+ obras\_aceesos+ obras\_tienda+ dias\_obras}

**Datastore:** CLASIFICACIÓN

**Output flows**

going to Consultar Datos Competicion[Process]

**Data within store:** {@id+ equipo+ puesto+ puntos+ goles\_favor+ goles\_contra+ partidos\_jugados+ partidos\_ganados+ partidos\_empatados+ partidos\_perdidos}

**Datastore:** PARTIDOS

**Output flows**

going to Consultar Datos Competición[Process]

**Data within store:** {@id+ jornada+ equipo\_local+ equipo\_visitante+ goles\_local+ goles\_visitante+ goleadores+ tarjetas+ lesionados }

**Datastore:** CRÉDITOS

**Input flows**

coming from Gestionar Economía[Process]

**Output flows**

going to Gestionar Economía[Process]

**Data within store:** {@id+ cantidad\_prestada+ cantidad\_devuelta + semanas\_devolución+ cuota\_semanal}

**Datastore:** CONTRATOS PATROCINIO

**Input flows**

coming from Gestionar Economía[Process]

**Output flows**

going to Gestionar Economía[Process]

**Data within store:** {@id + equipo+ pago\_semanal+ pago\_victoria+ pago\_objetivo+ temp\_contrato+ temp\_inicio}

**Datastore:** OFERTAS PATROCINIO

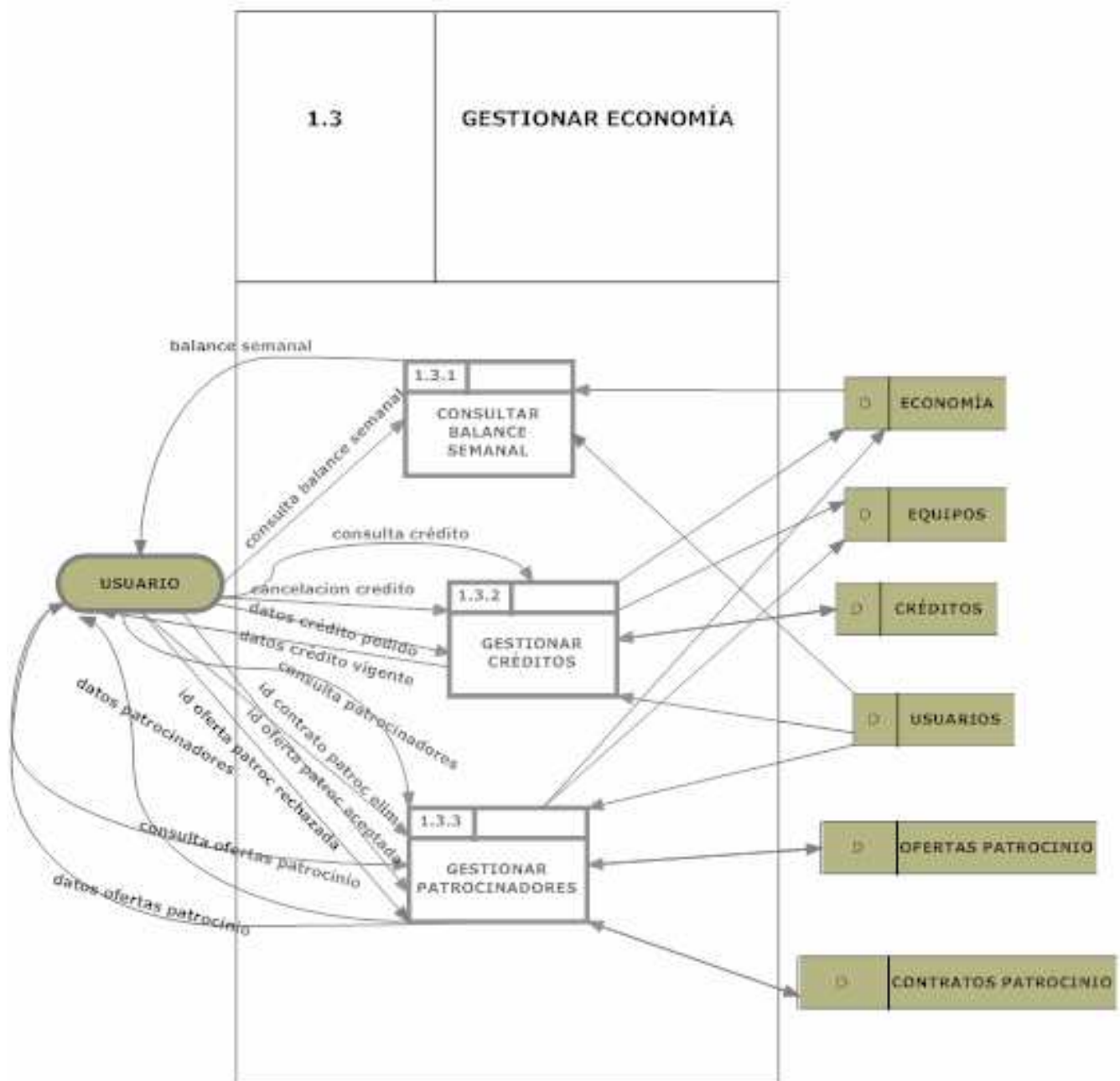
**Input flows**

coming from Gestionar Economía[Process]

**Output flows**

going to Gestionar Economía[Process]

**Data within store:** {@id + equipo+ pago\_semanal+ pago\_victoria+ pago\_objetivo+ temp\_contrato+ contesta}





**Fig 19. Diagrama de Flujos de Datos 2**

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### **Process: Consultar Balance Semanal**

#### **Minispec or Description:**

PRE1. Un usuario desea ver los datos económicos sobre su equipo relativos a la última semana(simulada).

POST1. El sistema obtiene los datos de ECONOMÍA y los muestra al usuario.

**Purpose:** Se lee el balance económico del equipo del usuario y se le muestra.

#### **Input flows**

consulta balance semanal

coming from USUARIO[External]

coming from ECONOMÍA[Data Store]

coming from USUARIOS[Data Store]

#### **Output flows**

balance semanal

going to USUARIO[External]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** consulta balance semanal

**From-symbol:** USUARIO[External]

**To-symbol:** Consultar Balance Semanal[Process],

**Data within flow:**

**Dataflow:** balance semanal

**From-symbol:** Consultar Balance Semanal[Process],

**To-symbol:** USUARIO[External]

**Data within flow:** {semana+ingresos\_traspos+ingresos\_estadio+ingresos\_patrocinadores+gastos\_sueldos+gastos\_fichajes+gastos\_obras+otros\_gastos}

**Dataflow:** consulta patrocinadores

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Patrocinadores[Process]

**Data within flow:**

**Dataflow:** datos patrocinadores

**From-symbol:** Gestionar Patrocinadores[Process]

**To-symbol:** USUARIO[External]

**Data within flow:** {id\_contrato\_patrocinio+pago\_semanal+pago\_victoria+pago\_objetivo+temp\_contrato+temp\_inicio}

**Dataflow:** Id contrato patrocinio eliminado

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Patrocinadores[Process]

**Data within flow:**

**Dataflow:** consulta ofertas patrocinio

**From-symbol:** USUARIO[External]

**To-symbol:** Gestionar Patrocinadores[Process]

**Data within flow:**

**Dataflow:** datos ofertas patrocinio

**From-symbol:** Gestionar Patrocinadores[Process]

**To-symbol:** USUARIO[External]

**Data within flow:** {id\_oferta\_patrocinio+ equipo+ pago\_semanal+ pago\_victoria+ pago\_objetivo+ temp\_contrato+contestada}

**Dataflow:** Id oferta patrocinio aceptada  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Patrocinadores[Process]  
**Data within flow:**

**Dataflow:** Id oferta patrocinio rechazada  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Patrocinadores[Process]  
**Data within flow:**

**Dataflow:** consulta crédito  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Créditos[Process]  
**Data within flow:**

**Dataflow:** datos crédito vigente  
**From-symbol:** Gestionar Créditos[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {id\_crédito+cantidad\_prestada+ cantidad\_devuelta  
semanas\_devolución+ cuota\_semanal}

+

**Dataflow:** datos crédito pedido  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Créditos[Process]  
**Data within flow:** {cantidad+semanas\_devolución}

**Dataflow:** cancelación crédito  
**From-symbol:** USUARIO[External]  
**To-symbol:** Gestionar Créditos[Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** ECONOMÍA

### Input flows

coming from Gestionar Créditos[Process]

coming from Gestionar Patrocinadores[Process]

### Output flows

going to Consultar Balance Semanal[Process]

**Data within store:** {@id+ equipo+ semana+ ingresos\_credito+ gastos\_credito+ gastos\_varios}

**Datastore:** USUARIOS

### Output flows

going to Consultar Balance Semanal[Process]

going to Gestionar Créditos[Process]

going to Gestionar Patrocinadores[Process]

**Data within store:** {@id+usuario+equipo}

**Datastore:** EQUIPOS

### Input flows

coming from Gestionar Créditos[Process]

coming from Gestionar Patrocinadores[Process]

**Data within store:** {@id+usuario+equipo}

**Datastore:** CRÉDITOS

**Input flows**

coming from Gestionar Créditos[Process]

**Output flows**

going to Gestionar Créditos[Process]

**Data within store:** {@id+ cantidad\_prestada+ cantidad\_devuelta + semanas\_devolución+ cuota\_semanal}

**Datastore:** OFERTAS PATROCINIO

**Input flows**

coming from Gestionar Patrocinadores[Process]

**Output flows**

going to Gestionar Patrocinadores[Process]

**Data within store:** {@id + equipo+ pago\_semanal+ pago\_victoria+ pago\_objetivo+ temp\_contrato+ contesta}

**Datastore:** CONTRATOS PATROCINIO

**Input flows**

coming from Gestionar Patrocinadores[Process]

**Output flows**

going to Gestionar Patrocinadores[Process]

**Data within store:** {@id + equipo+ pago\_semanal+ pago\_victoria+ pago\_objetivo+ temp\_contrato+ temp\_inicio}

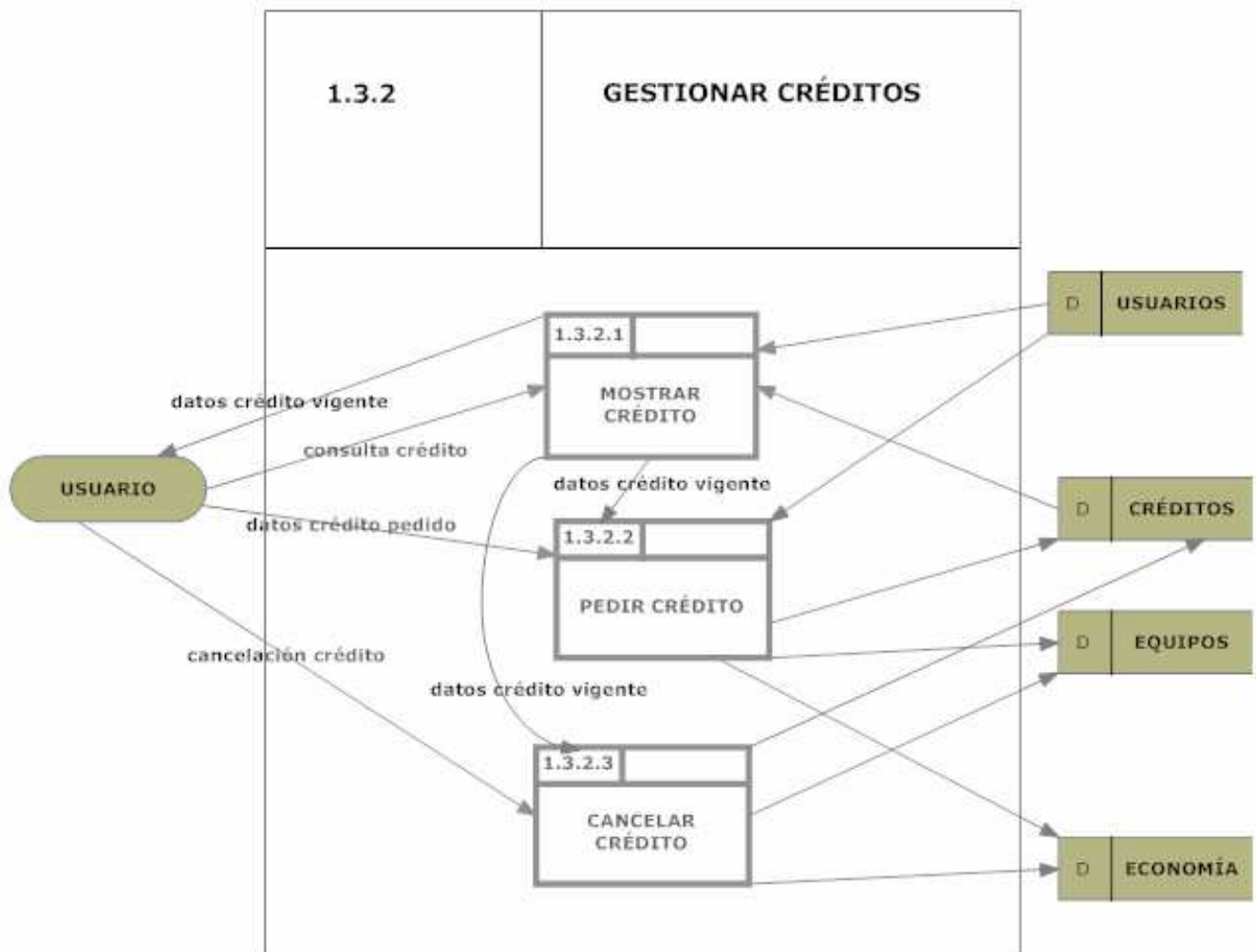


Fig 20. Diagrama de Flujos de Datos 3.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Mostrar Crédito

#### Minispec or Description:

PRE1. Un usuario desea conocer los datos acerca del crédito bancario concedido a su equipo (si existe)  
 POST1. El sistema comprueba si existe algún crédito vigente relacionado con el equipo del usuario y se lo muestra al mismo.

**Purpose:** Se muestran los datos sobre el crédito vigente de un equipo.

#### Input flows

consulta crédito  
 coming from USUARIO[External]  
  
 coming from CRÉDITOS[Data Store]  
  
 coming from USUARIOS[Data Store]

#### Output flows

datos crédito vigente  
 going to USUARIO[External], PEDIR CRÉDITO[Process]

### Process: Pedir Crédito

#### Minispec or Description:

PRE1. Un usuario desea conocer los datos acerca del crédito bancario vigente de su equipo(si existe).  
 POST1. El sistema le muestra los datos del crédito.

PRE2. El usuario pide un crédito e introduce los datos del mismo.

POST2. El sistema comprueba que no exista ningun crédito vigente, en cuyo caso actualiza CRÉDITOS introduciendo los datos del nuevo crédito y ECONOMÍA, registrando el ingreso, además de actualizar el saldo del equipo en EQUIPOS.

**Purpose:** Se crea un nuevo crédito bancario para un equipo.

**Input flows**

datos crédito pedido  
coming from USUARIO[External]  
datos crédito vigente  
coming from MOSTRAR CRÉDITO[Process]  
  
coming from USUARIOS[Data Store]

**Output flows**

going to CRÉDITOS[Data Store]  
going to EQUIPOS[Data Store]  
going to ECONOMÍA[Data Store]

**Process: Cancelar Crédito**

**Minispec or Description:**

PRE1. Un usuario desea conocer los datos acerca del crédito bancario vigente de su equipo(si existe).

POST1. El sistema le muestra los datos del crédito.

PRE2. El usuario cancela el crédito vigente.

POST2. El sistema, si existe un crédito vigente en ese momento, actualiza CRÉDITOS borrando el préstamo concedido al equipo del usuario, EQUIPOS, guardando el nuevo saldo y ECONOMÍA, registrando el gasto acarreado.

**Purpose:** Se le muestran al usuario las ofertas de patrocinio que hay para su equipo.

**Input flows**

cancelación crédito  
coming from USUARIO[External]  
datos crédito vigente  
coming from MOSTRAR CRÉDITO[Process]

**Output flows**

going to CRÉDITOS[Data Store]  
going to EQUIPOS[Data Store]  
going to ECONOMÍA[Data Store]

DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** consulta crédito

**From-symbol:** USUARIO[External]

**To-symbol:** Mostrar Crédito[Process]

**Data within flow:**

**Dataflow:** datos crédito vigente

**From-symbol:** Mostrar Crédito[Process]

**To-symbol:** USUARIO[External], Pedir Crédito[Process], Cancelar Crédito[Process]

**Data within flow:** {id\_crédito+cantidad\_prestada+ cantidad\_devuelta  
+semanas\_devolución+ cuota\_semanal}

**Dataflow:** datos crédito pedido

**From-symbol:** USUARIO[External]

**To-symbol:** Pedir Crédito[Process],

**Data within flow:** {cantidad+semanas\_devolución}

**Dataflow:** cancelación crédito  
**From-symbol:** USUARIO[External]  
**To-symbol:** Cancelar Crédito[Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

### **Datastore: USUARIOS**

#### **Output flows**

going to Mostrar Crédito[Process]

going to Pedir Crédito[Process]

**Data within store:** {@id+usuario+equipo}

### **Datastore: CRÉDITOS**

#### **Input flows**

coming from Pedir Crédito[Process]

coming from Cancelar Crédito[Process]

#### **Output flows**

going to Mostrar Crédito[Process]

**Data within store:** {@id+ cantidad\_prestada+ cantidad\_devuelta + semanas\_devolución+ cuota\_semanal}

### **Datastore: EQUIPOS**

#### **Input flows**

coming from Pedir Crédito[Process]

coming from Cancelar Crédito[Process]

**Data within store:** {@id+ saldo}

### **Datastore: ECONOMÍA**

#### **Input flows**

coming from Pedir Crédito[Process]

coming from Cancelar Crédito[Process]

**Data within store:** {@id+ equipo+ semana+ ingresos\_credito+ gastos\_credito}

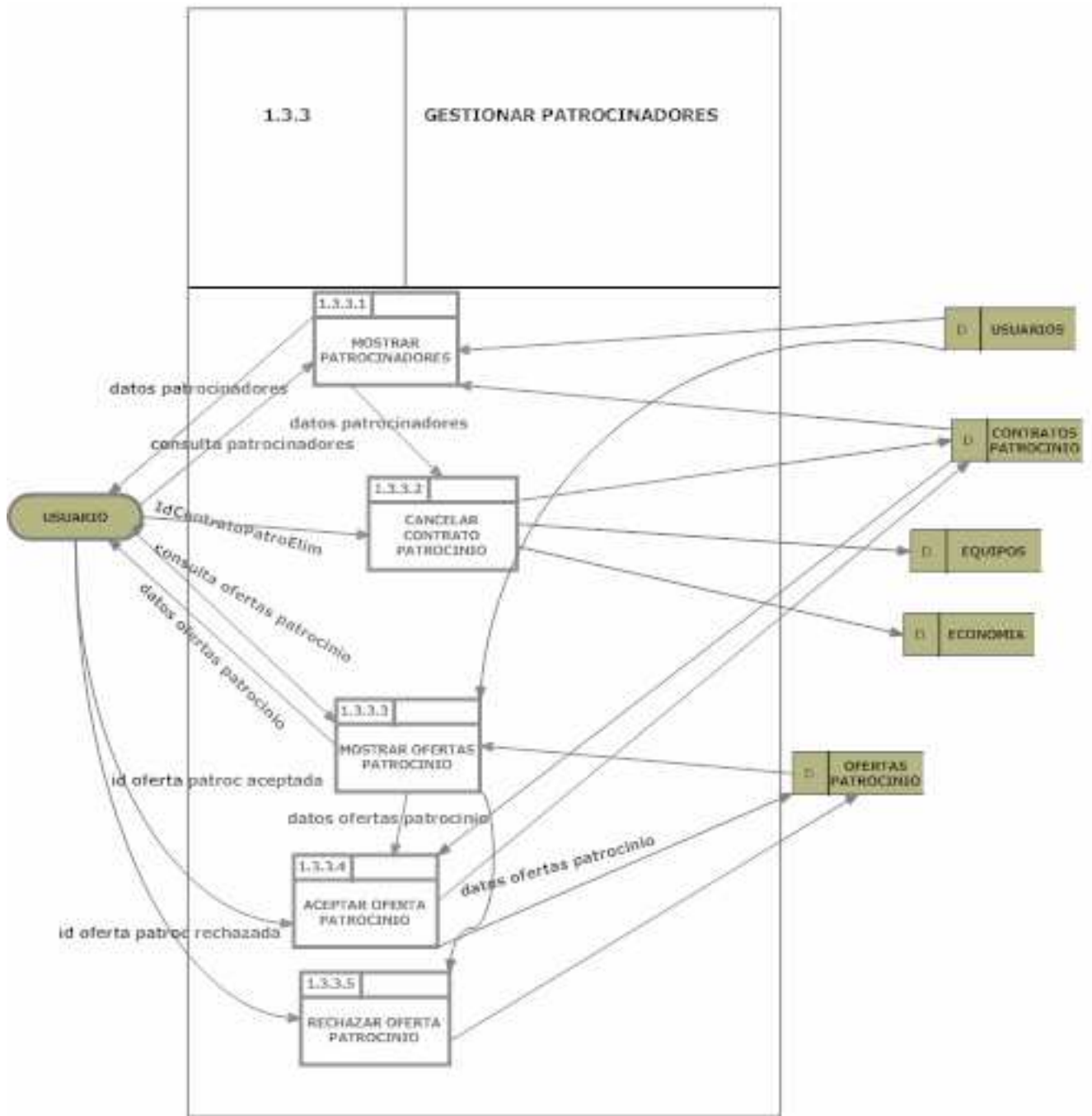


Fig 21. Diagrama de Flujos de Datos 4.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Mostrar Patrocinadores

#### Minispec or Description:

PRE1. Un usuario desea ver los acuerdo de patrocinio (vigente y futuro) que tiene su equipo.

POST1. El sistema obtiene los patrocinios relacionados con el equipo del usuario en CONTRATOS PATROCINIO y los muestra al usuario.

**Purpose:** Se leen los patrocinadores del equipo del usuario y se le muestran.

#### Input flows

consulta patrocinadores  
coming from USUARIO[External]

coming from CONTRATOS PATROCINIO[Data Store]

coming from USUARIOS[Data Store]

**Output flows**

datos patrocinadores

going to USUARIO[External], CANCELAR CONTRATO PATROCINIO[Process]

**Process: Cancelar Contrato Patrocinio**

**Minispec or Description:**

PRE1. Un usuario desea cancelar un acuerdo de patrocinio ya suscrito.

POST1. El sistema comprueba si se trata de un acuerdo de patrocinio vigente, en cuyo caso calcula la indemnización que debe pagar el equipo del usuario y actualiza el saldo; finalmente, sea un acuerdo vigente o futuro, elimina al contrato de patrocinio correspondiente.

**Purpose:** Se elimina un contrato de patrocinio suscrito por el equipo del usuario actualizando CONTRATOS PATROCINIO.

**Input flows**

IdContratoPatroElim

coming from USUARIO[External]

datos patrocinadores

coming from MOSTRAR PATROCINADORES[Process]

**Output flows**

going to EQUIPOS[Data Store]

going to ECONOMÍA[Data Store]

going to CONTRATOS PATROCINIO[Data Store]

**Process: Mostrar Ofertas Patrocinio**

**Minispec or Description:**

PRE1. El usuario desea consultar qué ofertas de patrocinio tiene un determinado momento.

POST1. El sistema obtiene el equipo del usuario y lee las ofertas de patrocinio existentes para ese equipo.

**Purpose:** Se le muestran al usuario las ofertas de patrocinio que hay para su equipo.

**Input flows**

consulta ofertas patrocinio

coming from USUARIO[External]

coming from USUARIOS[Data Store]

coming from OFERTAS PATROCINIO[Data Store]

**Output flows**

datos ofertas patrocinio

going to USUARIO[External], ACEPTAR OFERTA PATROCINIO[Process], RECHAZAR OFERTA PATROCINIO[Process]

**Process: Aceptar Oferta Patrocinio**

**Minispec or Description:**

PRE1. Un usuario desea ver las ofertas de patrocinio que tiene su equipo en un determinado momento.

POST1. El sistema le muestra al usuario las ofertas de patrocinio.

PRE2. El usuario decide aceptar una de las ofertas de patrocinio.

POST2. El sistema lee los datos de patrocinadores vigentes para comprobar si ya existe uno, en cuyo caso anula la operación; en caso contrario actualiza CONTRATOS PATROCINIO introduciendo el nuevo acuerdo y OFERTAS PATROCINIO, marcando la oferta como contestada.

**Purpose:** Se acepta una de las ofertas de patrocinio de un equipo.

**Input flows**

id oferta patroc aceptada

coming from USUARIO[External]



datos ofertas patrocinio  
coming from MOSTRAR OFERTAS PATROCINIO[Process]

coming from CONTRATOS PATROCINIO[Data Store]

**Output flows**

going to OFERTAS PATROCINIO[Data Store]

going to CONTRATOS PATROCINIO[Data Store]

**Process: Rechazar Oferta Patrocinio**

**Minispec or Description:**

PRE1. Un usuario desea ver las ofertas de patrocinio que tiene su equipo en un determinado momento.

POST1. El sistema le muestra al usuario las ofertas de patrocinio.

PRE2. El usuario decide rechazar una de las ofertas de patrocinio.

POST2. El sistema actualiza OFERTAS PATROCINIO marcando la oferta como rechazada.

**Purpose:** Se rechaza una de las ofertas de patrocinio de un equipo.

**Input flows**

id oferta patroc rechazada  
coming from USUARIO[External]

**Output flows**

going to OFERTAS PATROCINIO[Data Store]

DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** consulta patrocinadores  
**From-symbol:** USUARIO[External]  
**To-symbol:** Mostrar Patrocinadores[Process]  
**Data within flow:**

**Dataflow:** datos patrocinadores  
**From-symbol:** Mostrar Patrocinadores[Process]  
**To-symbol:** USUARIO[External], Cancelar Contrato Patrocinio[Process]  
**Data within flow:** {id\_contrato\_patrocinio+pago\_semanal+pago\_victoria+pago\_objetivo+temp\_contrato+temp\_inicio}

**Dataflow:** Id contrato patrocinio eliminado  
**From-symbol:** USUARIO[External]  
**To-symbol:** Cancelar Contrato Patrocinio[Process]  
**Data within flow:**

**Dataflow:** consulta ofertas patrocinio  
**From-symbol:** USUARIO[External]  
**To-symbol:** Mostrar Ofertas Patrocinio[Process]  
**Data within flow:**

**Dataflow:** datos ofertas patrocinio  
**From-symbol:** Mostrar Ofertas Patrocinio[Process]  
**To-symbol:** USUARIO[External], Aceptar Oferta Patrocinio[Process], Rechazar Oferta Patrocinio[Process]  
**Data within flow:** {id\_oferta\_patrocinio+ equipo+ pago\_semanal+ pago\_victoria+ pago\_objetivo+ temp\_contrato+contestada}

**Dataflow:** Id oferta patrocinio aceptada  
**From-symbol:** USUARIO[External]  
**To-symbol:** Aceptar Oferta Patrocinio[Process]  
**Data within flow:**

**Dataflow:** Id oferta patrocinio rechazada

**From-symbol:** USUARIO[External]  
**To-symbol:** Rechazar Oferta Patrocinio[Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** USUARIOS  
**Output flows**

going to Mostrar Patrocinadores[Process]

going to Mostrar Ofertas Patrocinio[Process]

**Data within store:** {@id+usuario+equipo}

**Datastore:** CONTRATOS PATROCINIO  
**Input flows**

coming from Cancelar Contrato Patrocinio[Process]

coming from Aceptar Oferta Patrocinio[Process]

**Output flows**

going to Mostrar Patrocinadores[Process]

going to Aceptar Oferta Patrocinio[Process]

**Data within store:** {@id + equipo+ pago\_semanal+ pago\_victoria+ pago\_objetivo+ temp\_contrato+temp\_inicio }

**Datastore:** EQUIPOS  
**Input flows**

coming from Cancelar Contrato Patrocinio[Process]

**Data within store:** {@id+saldo}

**Datastore:** ECONOMÍA  
**Input flows**

coming from Cancelar Contrato Patrocinio[Process]

**Data within store:** {@id+ equipo+ semana+ gastos\_varios}

**Datastore:** OFERTAS PATROCINIO  
**Input flows**

coming from Aceptar Oferta Patrocinio[Process]

coming from Rechazar Oferta Patrocinio[Process]

**Output flows**

going to Mostrar Ofertas Patrocinio[Process]

**Data within store:** {@id + equipo+ pago\_semanal+ pago\_victoria+ pago\_objetivo+ temp\_contrato+ contesta}

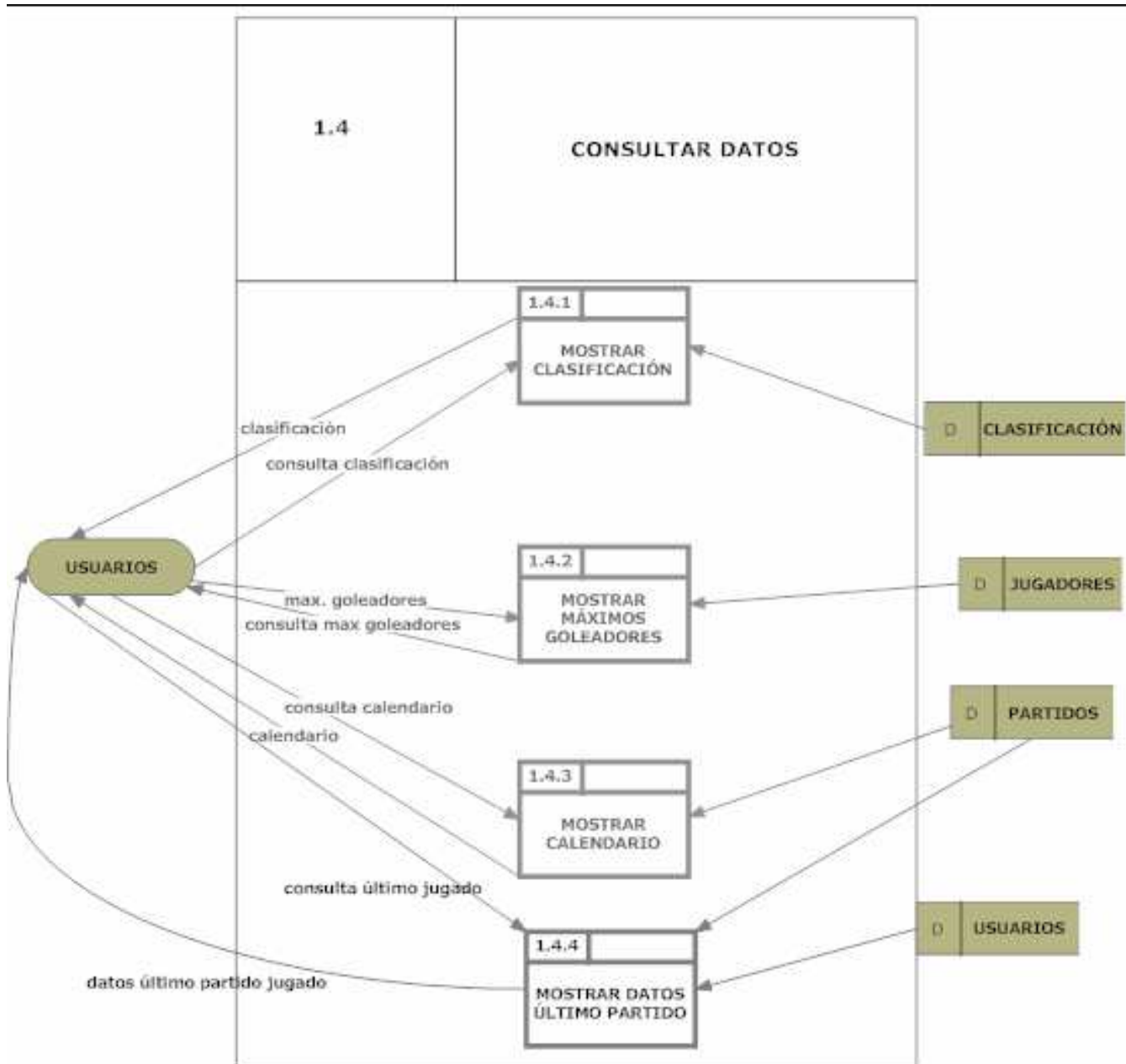


Fig 22. Diagrama de Flujos de Datos 5.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Mostrar Clasificación

#### Minispec or Description:

PRE1. Un usuario desea consultar la clasificación de la competición.

POST1. El sistema lee la clasificación y se la muestra al usuario.

**Purpose:** Se le muestra al usuario la clasificación de la competición.

#### Input flows

consulta clasificación  
coming from USUARIO[External]

coming from CLASIFICACIÓN[Data Store]

#### Output flows

clasificación  
going to USUARIO[External]

**Process: Mostrar Máximos Goleadores**

**Minispec or Description:**

PRE1. Un usuario desea consultar qué jugadores son los máximos goleadores de la competición.  
POST1. El sistema lee de JUGADORES y muestra al usuario los que más goles lleven en ese momento.

**Purpose:** Se le muestra al usuario la lista de máximos goleadores de la competición.

**Input flows**

consulta max goleadores  
coming from USUARIO[External]  
  
coming from JUGADORES[Data Store]

**Output flows**

max goleadores  
going to USUARIO[External]

**Process: Mostrar Calendario**

**Minispec or Description:**

PRE1. El usuario desea consultar cuál es el calendario (partidos ya jugados y próximos partidos) de la competición.

POST1. El sistema lee de PARTIDOS y muestra al usuario el calendario.

**Purpose:** Se le muestra al usuario el calendario de la competición.

**Input flows**

consulta calendario  
coming from USUARIO[External]  
  
coming from PARTIDOS[Data Store]

**Output flows**

calendario  
going to USUARIO[External]

**Process: Mostrar Datos Último Partido**

**Minispec or Description:**

PRE1. Un usuario desea consultar información acerca del último partido disputado por su equipo.

POST1. El sistema obtiene el equipo del usuario y lee de PARTIDOS los datos referentes al último partido disputado por el mismo, mostrándoselos al usuario.

**Purpose:** Se le muestra al usuario la información sobre el último partido disputado por su equipo.

**Input flows**

consulta último jugado  
coming from USUARIO[External]  
  
coming from PARTIDOS[Data Store]  
  
coming from USUARIOS[Data Store]

**Output flows**

datos último partido jugado  
going to USUARIO[External]

DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** consulta clasificación

**From-symbol:** USUARIO[External],

**To-symbol:** Mostrar Clasificación[Process]

**Data within flow:**

Dataflow: clasificación

**From-symbol:** Mostrar Clasificación[Process]

**To-symbol:** USUARIO[External]

**Data within flow:** {nombre\_equipo+puesto+puntos+goles\_favor+goles\_contra+partidos\_jugados+partidos\_ganados+partidos\_empatados+partidos\_perdidos}

**Dataflow:** consulta max goleadores  
**From-symbol:** USUARIO[External]  
**To-symbol:** Mostrar Máximos Goleadores[Process]  
**Data within flow:**

**Dataflow:** max goleadores  
**From-symbol:** Mostrar Máximos Goleadores[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {nombre\_jugador+goles\_marcados}

**Dataflow:** consulta calendario  
**From-symbol:** USUARIO[External]  
**To-symbol:** Mostrar Calendario[Process]  
**Data within flow:**

**Dataflow:** calendario  
**From-symbol:** Mostrar Calendario[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {jornada+nombre\_equipo\_local+nombre\_equipo\_visitante+goles\_local+goles\_visitante}

**Dataflow:** consulta último jugado  
**From-symbol:** USUARIO[External]  
**To-symbol:** Mostrar Datos Último Partido[Process]  
**Data within flow:**

**Dataflow:** datos último partido jugado  
**From-symbol:** Mostrar Datos Último Partido[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:**  
{jornada+nombre\_equipo\_local+nombre\_equipo\_visitante+goles\_local+goles\_visitante+goleadores+tarjetas+lesionados}

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** CLASIFICACIÓN  
**Output flows**

going to Mostrar Clasificación[Process]  
**Data within store:** {@id+ equipo+ puesto+ puntos+ goles\_favor+ goles\_contra+ partidos\_jugados+ partidos\_ganados+ partidos\_empatados+ partidos\_perdidos}

**Datastore:** JUGADORES  
**Output flows**

going to Mostrar Máximos Goleadores[Process]  
**Data within store:** {@id+ nombre + goles\_marcados}

**Datastore:** PARTIDOS  
**Output flows**

going to Mostrar Calendario[Process]  
**Data within store:** {@id jornada+ equipo\_local+ equipo\_visitante+ goles\_local+ goles\_visitante+ goleadores+ tarjetas+ lesionados }

**Datastore:** USUARIOS  
**Output flows**

going to Mostrar Calendario[Process]  
**Data within store:** {@id+usuario+equipo}

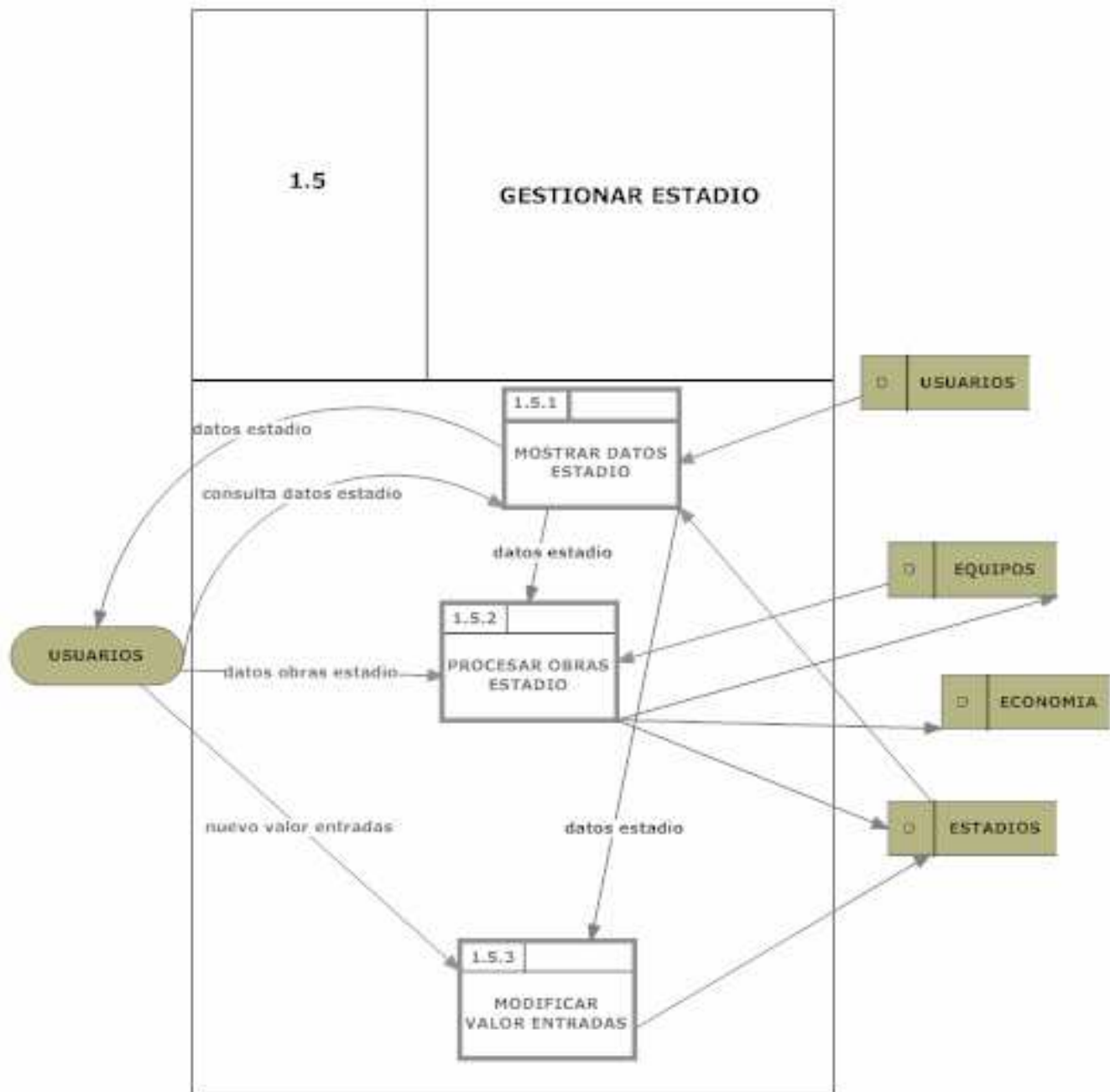


Fig 23. Diagrama de Flujos de Datos 6.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Mostrar Datos Estadio

#### Minispec or Description:

PRE1. Un usuario desea consultar la información relativa al estadio de su equipo.

POST1. El sistema obtiene el equipo del usuario y consulta los datos del estadio del mismo, mostrándoselos al usuario.

**Purpose:** Se le muestra al usuario la información sobre su estadio..

#### Input flows

consulta datos estadio  
coming from USUARIO[External]

coming from USUARIOS[Data Store]

coming from ESTADIOS[Data Store]

**Output flows**

datos estadio

going to USUARIO[External], PROCESAR OBRAS ESTADIO[Process],  
MODIFICAR VALOR ENTRADAS[Process]

**Process: Procesar Obras Estadio**

**Minispec or Description:**

PRE1. Un usuario desea consultar información acerca del estadio de su equipo.

POST1. El sistema muestra al usuario la información sobre su estadio.

PRE2. El usuario desea hacer algún tipo de obra sobre su estadio e introduce los datos sobre las mismas.

POST2. El sistema comprueba que no haya ninguna obra en curso, decide el coste de las obras, se asegura de que el equipo del usuario tenga el salgo necesario para acometerlas, y en caso afirmativo actualiza ESTADIOS indicando el tipo de obras acometidas, EQUIPOS guardando el nuevo saldo del equipo y ECONOMÍA, registrando la operación.

**Purpose:** Se decide si son posibles las obras en un estadio elegidas por un usuario y registra los cambios en el estadio y el coste.

**Input flows**

datos estadio

coming from MOSTRAR DATOS ESTADIO[Process]

datos obras estadio

coming from USUARIO[External]

coming from EQUIPOS[Data Store]

**Output flows**

going to EQUIPOS[Data Store], ECONOMÍA[Data Store]

going to ESTADIOS[Data Store]

**Process: Modificar Valor Entradas**

**Minispec or Description:**

PRE1. Un usuario desea consultar información acerca del estadio de su equipo.

POST1. El sistema muestra al usuario la información sobre su estadio.

PRE2. El usuario desea cambiar el valor de las entradas para los partidos jugados en su estadio e introduce el nuevo valor.

POST2. El sistema actualiza ESTADIOS.

**Purpose:** Se modifica el precio establecido de las entradas para los partidos jugados en el estadio del usuario.

**Input flows**

nuevo valor entradas

coming from USUARIO[External]

datos estadio

coming from MOSTRAR DATOS ESTADIO[Process]

**Output flows**

going to ESTADIO[Data Store]

**DESCRIPCIÓN FLUJOS DE DATOS**

**Dataflow:** consulta datos estadio

**From-symbol:** USUARIO[External]

**To-symbol:** Mostrar Datos Estadio[Process]

**Data within flow:**

**Dataflow:** datos estadio

**From-symbol:** Mostrar Datos Estadio[Process]  
**To-symbol:** USUARIO[External], Procesar Obras Estadio[Process], Modificar Valor Entradas[Process]  
**Data within flow:** {id\_estadio+equipo+ nombre+ capacidad+ valor\_entradas+ nivel\_aceesos+ nivel\_tienda+obras\_capacidad+obras\_aceesos+obras\_tienda+dias\_obras}

**Dataflow:** datos obras estadio  
**From-symbol:** USUARIO[External]  
**To-symbol:** Procesar Obras Estadio[Process]  
**Data within flow:** : {id\_estadio+obras\_capacidad+obras\_aceesos+obras\_tienda}

**Dataflow:** nuevo valor entradas  
**From-symbol:** USUARIO[External]  
**To-symbol:** Modificar Valor Entradas[Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** USUARIOS  
**Output flows**

going to Mostrar Datos Estadio[Process],  
**Data within store:** {@id+usuario+equipo}

**Datastore:** EQUIPOS  
**Input flows**

coming from Procesar Obras Estadio[Process]  
**Output flows**

going to Procesar Obras Estadio[Process]  
**Data within store:** {@id+saldo}

**Datastore:** ECONOMÍA  
**Input flows**

coming from Procesar Obras Estadio[Process]  
**Data within store:** {@id+equipo+semana+gastos\_obras}

**Datastore:** ESTADIOS  
**Input flows**

coming from Procesar Obras Estadio[Process]

coming from Modificar Valor Entradas[Process]

**Output flows**

going to Mostrar Datos Estadio[Process]  
**Data within store:** {@id + equipo+ nombre+ capacidad+ valor\_entradas+ nivel\_aceesos+ nivel\_tienda+ obras\_capacidad+ obras\_aceesos+ obras\_tienda+ dias\_obras}



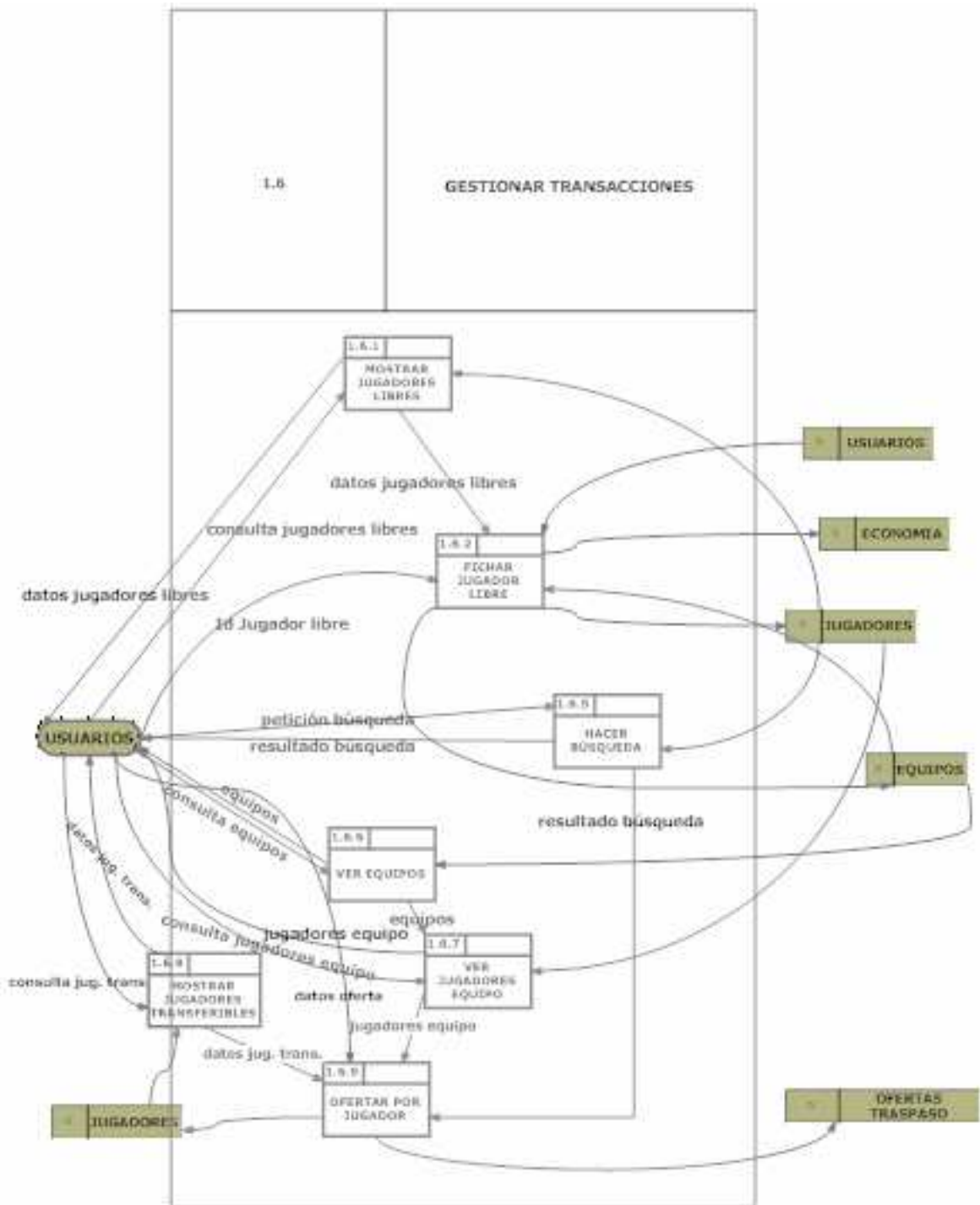


Fig 24. Diagrama de Flujos de Datos 7.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Mostrar Jugadores Libres

#### Minispec or Description:

PRE1. Un usuario desea ver todos los jugadores libres (sin equipo) presentes en la competición.  
 POST1. El sistema obtiene información de los jugadores sin equipo y la muestra al usuario.

**Purpose:** Se devuelve al usuario información sobre los jugadores libres que hay en el sistema.

#### Input flows

consulta jugadores libres  
 coming from USUARIO[External]

coming from JUGADORES[Data Store]

**Output flows**

datos jugadores libres  
going to USUARIO[External], FICHAR JUGADOR LIBRE[Process]

**Process: Fichar Jugador Libre**

**Minispec or Description:**

PRE1. Un usuario consulta los jugadores libres del sistema.

POST1. El sistema obtiene información de los jugadores sin equipo y la muestra al usuario.

PRE2. Un usuario decide fichar a un jugador libre.

POST2. El sistema comprueba si el equipo del usuario tiene saldo suficiente para hacer el fichaje y en caso afirmativo actualiza JUGADORES guardando el nuevo equipo del jugador, EQUIPOS indicando el nuevo saldo y ECONOMÍA registrando la operación.

**Purpose:** El usuario ficha a un jugador hasta entonces sin equipo.

**Input flows**

id jugador libre  
coming from USUARIO[External]  
  
coming from USUARIOS[Data Store]

coming from EQUIPOS[Data Store]  
datos jugadores libres  
coming from MOSTRAR JUGADORES LIBRES[Process]

**Output flows**

going to JUGADORES[Data Store]  
  
going to EQUIPOS[Data Store]  
  
going to ECONOMÍA[Data Store]

**Process: Hacer Búsqueda**

**Minispec or Description:**

PRE1. Un usuario desea hacer una búsqueda sobre todos los jugadores del sistema y establece los criterios de la misma.

POST1. El sistema obtiene información de los jugadores que cumplen con los criterios de búsqueda y la devuelve al usuario.

**Purpose:** El sistema devuelve al usuario todos los jugadores que cumplen los requisitos establecidos por él mismo.

**Input flows**

petición búsqueda  
coming from USUARIO[External]  
  
coming from JUGADORES[Data Store]

**Output flows**

resultado búsqueda  
going to USUARIO[External], OFERTAR POR JUGADOR[Process]

**Process: Ver Equipos**

**Minispec or Description:**

PRE1. Un usuario desea consultar todos los equipos presentes en la competición.

POST1. El sistema obtiene los equipos y los muestra al usuario.

**Purpose:** Se muestran todos los equipos al usuario.

**Input flows**

consulta equipos  
coming from USUARIO[External]  
  
coming from EQUIPOS[Data Store]

**Output flows**

equipos

going to USUARIO[External], VER JUGADORES EQUIPO[Process]

**Process: Ver Jugadores Equipo****Minispec or Description:**

PRE1. Un usuario consulta los equipos de la competición.

POST1. El sistema le muestra al usuario todos los equipos.

PRE2. El usuario consulta la plantilla de uno de los equipos.

POST2. El sistema obtiene información sobre todos los jugadores pertenecientes al equipo indicado y la muestra al usuario.

**Purpose:** Se muestra la plantilla de un equipo.**Input flows**

equipos

coming from VER EQUIPOS[Process]

consulta jugadores equipo

coming from USUARIO[External]

coming from JUGADORES[Data Store]

**Output flows**

jugadores equipo

going to USUARIO[External], OFERTAR POR JUGADOR[Process]

**Process: Mostrar Jugadores Transferibles****Minispec or Description:**

PRE1. Un usuario desea consultar los jugadores marcados como transferibles existentes en el sistema.

POST1. El sistema obtiene los jugadores transferibles y los muestra al usuario.

**Purpose:** Se muestran al usuario los jugadores transferibles.**Input flows**

consulta jug. trans.

coming from USUARIO[External]

coming from JUGADORES[Data Store]

**Output flows**

datos jug. trans.

going to USUARIO[External], OFERTAR POR JUGADOR[Process]

**Process: Ofertar Por Jugador****Minispec or Description:**

PRE1. Un usuario ha realizado una búsqueda de jugadores, ha consultado la plantilla de un equipo o los jugadores transferibles y hace una oferta por uno de ellos.

POST1. El sistema registra que el jugador tiene una oferta de traspaso actualizando JUGADORES e introduce los datos de la nueva oferta en OFERTAS TRASPASO.

**Purpose:** Se crea una nueva oferta de traspaso por algún jugador.**Input flows**

jugadores equipo

coming from VER JUGADORES EQUIPO[Process]

datos jug. trans.

coming from MOSTRAR JUGADORES TRANSFERIBLES[Process]

resultado búsqueda

coming from HACER BÚSQUEDA[Process]

datos oferta

coming from USUARIO[External]

coming from USUARIOS[Data Store]

**Output flows**

going to JUGADORES[Data Store]

going to OFERTAS TRASPASO[Data Store]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** id jugador libre

**From-symbol:** USUARIO[External]

**To-symbol:** Fichar Jugador Libre[Process]

**Data within flow:**

**Dataflow:** consulta jugadores libres

**From-symbol:** USUARIO[External]

**To-symbol:** Mostrar Jugadores Libres[Process]

**Data within flow:**

**Dataflow:** datos jugadores libres

**From-symbol:** Mostrar Jugadores Libres[Process]

**To-symbol:** USUARIO[External], Fichar Jugador Libre[Process]

**Data within flow:** {id\_jug+nombre+media+posicion+edad+pais+valor+sueldo+años\_contrato}

**Dataflow:** petición búsqueda

**From-symbol:** USUARIO[External]

**To-symbol:** Hacer Búsqueda[Process]

**Data within flow:**

**Dataflow:** resultado búsqueda

**From-symbol:** Mostrar Info Jugador[Process]

**To-symbol:** USUARIO[External], Ofertar Por Jugador[Process]

**Data within flow:** {id\_jug+nombre+media+posicion+edad+pais+valor+sueldo+años\_contrato+transferible}

**Dataflow:** consulta equipos

**From-symbol:** USUARIO[External]

**To-symbol:** Ver Equipos[Process]

**Data within flow:**

**Dataflow:** equipos

**From-symbol:** Mostrar Info Jugador[Process]

**To-symbol:** USUARIO[External], Ver Jugadores Equipo[Process]

**Data within flow:** {@id\_eq+nombre}

**Dataflow:** consulta jugadores equipo

**From-symbol:** USUARIO[External]

**To-symbol:** Ver Jugadores Equipo[Process]

**Data within flow:**

**Dataflow:** jugadores equipo

**From-symbol:** Ver Jugadores Equipo[Process]

**To-symbol:** USUARIO[External], Ofertar Por Jugador[Process]

**Data within flow:** {id\_jug+nombre+media+posicion+edad+pais+valor+sueldo+años\_contrato+transferible}

**Dataflow:** consulta jug. trans.

**From-symbol:** USUARIO[External]

**To-symbol:** Mostrar Jugadores Transferibles[Process]

**Data within flow:**

**Dataflow:** datos jug. trans.

**From-symbol:** Mostrar Jugadores Transferibles[Process]

**To-symbol:** USUARIO[External], Ofertar Por Jugador[Process]

**Data within flow:**

{id\_jug+nombre+media+posicion+edad+pais+valor+sueldo+años\_contrato}

**Dataflow:** datos oferta

**From-symbol:** Mostrar Jugadores Transferibles[Process]  
**To-symbol:** USUARIO[External], Ofertar Por Jugador[Process]  
**Data within flow:** {id\_jug+cantidad}

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** USUARIOS  
**Output flows**

going to Fichar Jugador Libre[Process], Ofertar Por Jugador[Process]  
**Data within store:** {@id+usuario+equipo}

**Datastore:** ECONOMÍA  
**Input flows**

coming from Fichar Jugador Libre[Process]  
**Data within store:** {@id+equipo+gastos\_comprajug}

**Datastore:** JUGADORES  
**Input flows**

coming from Fichar Jugador Libre[Process]

Ofertar Por Jugador[Process]

**Output flows**

going to Mostrar Jugadores Libres[Process]

going to Hacer Búsqueda[Process]

going to Ver Jugadores Equipo[Process]

going to Mostrar Jugadores Transferibles[Process]

**Data within store:**

{@id+equipo+nombre+media+posicion+edad+pais+valor+sueldo+años\_contrato+transferible}

**Datastore:** EQUIPOS  
**Input flows**

coming from Fichar Jugador Libre[Process]

**Output flows**

going to Fichar Jugador Libre[Process]

going to Ver Equipos[Process]

going to Ofertar Por Jugador[Process]

**Data within store:** {@id+ nombre+ saldo}

**Datastore:** OFERTAS TRASPASO  
**Input flows**

coming from to Ofertar Por Jugador[Process]

**Data within store:** {@id+jug+equipo\_comprador+equipo\_vendedor+cantidad}

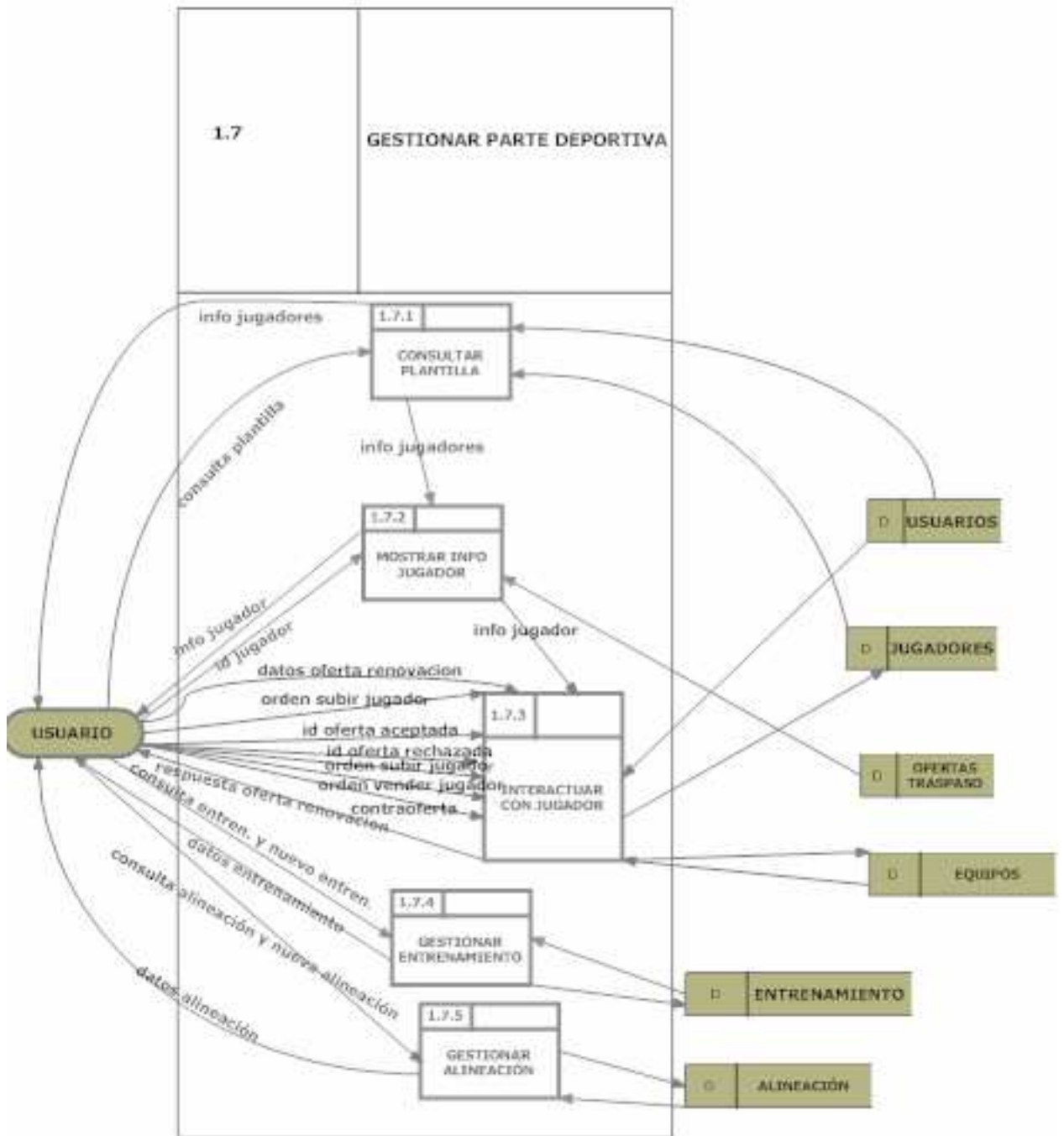


Fig 25. Diagrama de Flujos de Datos 8.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Consultar Plantilla

#### Minispec or Description:

PRE1. Un usuario realiza una consulta para obtener información sobre los jugadores de su plantilla.  
 POST1. Se lee del almacén USUARIOS para obtener el identificador del equipo del usuario y de JUGADORES para obtener los jugadores pertenecientes a dicho equipo, devolviendo entonces la información obtenida sobre los jugadores al usuario.

**Purpose:** Muestra al usuario información sobre los jugadores que forman parte de la plantilla de su equipo.

#### Input flows

consulta plantilla  
 coming from USUARIO[External]

coming from USUARIOS[Data Store]

coming from JUGADORES[Data Store]

**Output flows**

info jugadores

going to USUARIO[External], MOSTRAR INFO JUGADORES[Process]

**Process: Mostrar Info Jugador**

**Minispec or Description:**

PRE1. Un usuario desea consultar todos los datos relativos a un jugador de la plantilla de su equipo.

POST1. El sistema lee los datos del jugador y los muestra al usuario.

**Purpose:** El sistema devuelve al usuario todos los datos relativos a un jugador de su equipo.

**Input flows**

info jugadores

coming from CONSULTAR PLANTILLA [Process]

id jugador

coming from USUARIO[External]

coming from OFERTAS TRASPASO[Data Store]

**Output flows**

info jugador

going to USUARIO[External], INTERACTUAR CON JUGADOR[Process]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** info jugadores

**From-symbol:** Consultar Plantilla[Process]

**To-symbol:** USUARIO[External], Mostrar Info Jugador[Process]

**Data within flow:** {@id\_jug+nombre+posicion+media+estado\_fisico+moral+pais+valor+sueldo+años\_contrato+juvenil}

**Dataflow:** consulta plantilla

**From-symbol:** USUARIO[External]

**To-symbol:** Consultar Plantilla[Process]

**Data within flow:**

**Dataflow:** id jugador

**From-symbol:** USUARIO[External]

**To-symbol:** Mostrar Info Jugador[Process]

**Data within flow:**

**Dataflow:** info jugador

**From-symbol:** Mostrar Info Jugador[Process]

**To-symbol:** USUARIO[External], Interactuar Con Jugador[Process]

**Data within flow:** {@id\_jug+nombre+valor+sueldo+años\_contrato+cantidad\_oferta}

**Dataflow:** respuesta oferta renovación

**From-symbol:** Interactuar Con Jugador[Process]

**To-symbol:** USUARIO[External]

**Data within flow:**

**Dataflow:** id oferta traspaso rechazada

**From-symbol:** USUARIO[External]

**To-symbol:** Interactuar Con Jugador[Process]

**Data within flow:**

**Dataflow:** orden subir jugador

**From-symbol:** USUARIO[External]

**To-symbol:** Interactuar Con Jugador[Process]  
**Data within flow:**

**Dataflow:** id oferta traspaso aceptada  
**From-symbol:** USUARIO[External]  
**To-symbol:** Interactuar Con Jugador[Process]  
**Data within flow:**

**Dataflow:** contraoferta  
**From-symbol:** USUARIO[External]  
**To-symbol:** Interactuar Con Jugador[Process]  
**Data within flow:** {id\_oferta+cantidad}

**Dataflow:** orden poner transferible  
**From-symbol:** USUARIO[External]  
**To-symbol:** Interactuar Con Jugador[Process]  
**Data within flow:**

**Dataflow:** orden poner no transferible  
**From-symbol:** USUARIO[External]  
**To-symbol:** Interactuar Con Jugador[Process]  
**Data within flow:**

**Dataflow:** orden vender jugador  
**From-symbol:** USUARIO[External]  
**To-symbol:** Interactuar Con Jugador[Process]  
**Data within flow:**

**Dataflow:** consulta entrenamiento  
**From-symbol:** USUARIO[External]  
**To-symbol:** Consultar Entrenamiento[Process]  
**Data within flow:** {@id+usuario+equipo}

**Dataflow:** datos entrenamiento  
**From-symbol:** Consultar Entrenamiento [Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {est\_fisico+port+vel+agil+fuer+tec+dis+reg+pase+vjue+res+def+jue\_ae+captact}

**Dataflow:** nuevo entrenamiento  
**From-symbol:** USUARIO[External]  
**To-symbol:** Modificar Entrenamiento [Process]  
**Data within flow:** {equipo+est\_fisico+port+vel+agil+fuer+tec+dis+reg+pase+vjue+res+def+jue\_ae+captact}

**Dataflow:** consulta alineacion  
**From-symbol:** USUARIO[External]  
**To-symbol:** Consultar Alineacion[Process]  
**Data within flow:**

**Dataflow:** datos alineacion  
**From-symbol:** Consultar Alineacion[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {formacion+fue\_jue+jug1+jug2+jug3+jug4+jug5+jug6+jug7+jug8+jug9+jug10+jug11}

**Dataflow:** nueva alineacion  
**From-symbol:** USUARIO[External]  
**To-symbol:** Modificar Alineacion[Process]  
**Data within flow:** {equipo+formacion+fue\_jue+jug1+jug2+jug3+jug4+jug5+jug6+jug7+jug8+jug9+jug10+jug11}



## ESTRUCTURA ALMACENES DE DATOS

### **Datastore: JUGADORES**

#### **Output flows**

going to Consultar Plantilla[Process]

#### **Input flows**

coming from Interactuar Con Jugador[Process]

**Data within store:** {@id+equipo+temporadas\_contrato+sueldo+juvenil+transferible+valor}

### **Datastore: OFERTAS TRASPASO**

#### **Input flows**

coming from Interactuar Con Jugador[Process]

coming from Mostrar Info Jugador[Process]

**Data within store:** {@id+jugador+equipo\_comprador+equipo\_vendedor+cantidad+contraoferta}

### **Datastore: EQUIPOS**

#### **Input flows**

coming from Interactuar Con Jugador[Process]

#### **Output flows**

going to Interactuar Con Jugador[Process]

**Data within store:** {@id+saldo}

### **Datastore: ECONOMÍA**

#### **Input flows**

coming from Interactuar Con Jugador[Process]

**Data within store:** {@id+ equipo+ semana+ ingresos\_ventajug}

### **Datastore: USUARIOS**

#### **Output flows**

going to Gestionar Entrenamiento[Process]

going to Gestionar Alineacion[Process]

going to Consultar Plantilla[Process]

**Data within store:** {@id+usuario+equipo}

### **Datastore: ENTRENAMIENTO**

#### **Input flows**

coming from Gestionar Entrenamiento[Process]

#### **Output flows**

going to Consultar Entrenamiento[Process]

**Data within store:** {@id+ equipo+ estado\_fisico+ porteria+ velocidad+ agilidad+ fuerza+ tecnica+ disparo+ regate+ pase+ vision\_juego+ resistencia+ defensa+ juego\_aereo+ capacidad\_tactica}

### **Datastore: ALINEACIONES**

#### **Input flows**

coming from Gestionar Alineacion [Process]

#### **Output flows**



POST2. Se muestra al usuario la respuesta sobre su oferta, y en caso de ser positiva, se actualiza JUGADORES introduciendo los nuevos datos contractuales del jugador.

**Purpose:** Hace una oferta de renovación de contrato a un jugador.

**Input flows**

info jugador  
coming from MOSTRAR INFO JUGADOR[Process]  
datos oferta renovación  
coming from USUARIO[External]

**Output flows**

respuesta oferta renovación  
going to USUARIO [External]  
  
going to JUGADORES [Data Store]

**Process: Subir a Primer Equipo**

**Minispec or Description:**

PRE1. Un usuario desea que un jugador de su equipo juvenil pase al primer equipo.

POST1. Se actualiza JUGADORES indicando que el jugador marcado por el usuario es no juvenil.

**Purpose:** Pasa un jugador juvenil a jugador de primer equipo.

**Input flows**

info jugador  
coming from MOSTRAR INFO JUGADOR[Process]  
orden subir jugador  
coming from USUARIO[External]

**Output flows**

going to JUGADORES[Data Store]

**Process: Rechazar Oferta Traspaso**

**Minispec or Description:**

PRE1. Un usuario marca como rechazada alguna de las ofertas de traspaso recibidas por uno de sus jugadores.

POST1. Se actualiza OFERTAS TRASPASO indicando que la oferta ha sido rechazada.

**Purpose:** Rechaza una oferta de traspaso recibida por alguno jugador perteneciente al equipo del usuario.

**Input flows**

id oferta traspaso rechazada  
coming from USUARIO[External]  
info jugador  
coming from MOSTRAR INFO JUGADOR[Process]

**Output flows**

going to OFERTAS TRASPASO[Data Store]

**Process: Aceptar Oferta Traspaso**

**Minispec or Description:**

PRE1. Un usuario marca como aceptada alguna de las ofertas de traspaso recibidas por uno de sus jugadores.

POST1. El sistema lee el saldo disponible por el equipo que ha realizado la oferta.

PRE2. Se comprueba si el equipo comprador tiene saldo para afrontar el traspaso.

POST2. Si el equipo no tiene dinero suficiente, se anula la operación. En caso contrario se actualiza OFERTAS TRASPASO borrando todas las ofertas hechas por el jugador, además de modificar JUGADORES para guardar el nuevo equipo del jugador; por último, se actualiza ECONOMÍA (registrado la cantidad obtenida en concepto de traspaso de jugadores) y el saldo del equipo del usuario en EQUIPOS.

**Purpose:** Acepta una oferta de traspaso recibida por algún jugador perteneciente al equipo del usuario.

**Input flows**

info jugador  
coming from MOSTRAR INFO JUGADOR[Process]  
id oferta traspaso aceptada

coming from USUARIO[External]

coming from EQUIPOS[Data Store]

**Output flows**

going to JUGADORES[Data Store]

going to OFERTAS TRASPASO[Data Store]

going to ECONOMIA[Data Store]

going to EQUIPOS[Data Store],

**Process: Negociar Oferta Traspaso**

**Minispec or Description:**

PRE1. Un usuario lee las ofertas existentes por uno de sus jugadores y decide contraofertar solicitando una cantidad.

POST1. El sistema actualiza OFERTAS TRASPASO.

**Purpose:** El usuario hace una contraoferta a un equipo que previamente ha hecho una oferta por uno de sus jugadores.

**Input flows**

info jugador

coming from MOSTRAR INFO JUGADOR[Process]

contraoferta

coming from USUARIO[External]

**Output flows**

going to OFERTAS TRASPASO[Data Store]

**Process: Poner Transferible**

**Minispec or Description:**

PRE1. Un usuario accede a la pantalla con la información sobre un jugador no transferible y lo marca como transferible.

POST1. El sistema actualiza JUGADORES guardando la condición de transferible del jugador.

**Purpose:** El sistema coloca como transferible a un jugador..

**Input flows**

info jugador

coming from MOSTRAR INFO JUGADOR[Process]

orden poner transferible

coming from USUARIO[External]

**Output flows**

going to JUGADORES[Data Store]

**Process: Poner No Transferible**

**Minispec or Description:**

PRE1. Un usuario accede a la pantalla con la información sobre un jugador transferible y lo marca como no transferible.

POST1. El sistema actualiza JUGADORES guardando la condición de no transferible del jugador.

**Purpose:** El sistema coloca como no transferible a un jugador..

**Input flows**

info jugador

coming from MOSTRAR INFO JUGADOR[Process]

orden poner no transferible

coming from USUARIO[External]

**Output flows**

going to JUGADORES[Data Store]

### Process: Vender Jugador

#### Minispec or Description:

PRE1. Un usuario decide vender al sistema a un jugador perteneciente a su equipo.

POST1. Se actualiza JUGADORES indicando que el jugador está libre (no pertenece a ningún equipo), EQUIPOS para guardar el nuevo saldo del jugador, ECONOMÍA para registrar la operación y por el último se borran todas las ofertas de traspaso que existieran por ese jugador en OFERTAS TRASPASO.

**Purpose:** Modifica el entrenamiento del equipo del usuario actualizando ENTRENAMIENTO.

#### Input flows

orden vender jugador  
coming from USUARIO[External]

#### Output flows

going to JUGADORES[Data Store]  
going to OFERTAS TRASPASO[Data Store]  
going to EQUIPOS[Data Store]  
going to ECONOMIA[Data Store]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** info jugador

**From-symbol:** Mostrar Info Jugador[Process]

**To-symbol:** Hacer Oferta Renovación[Process], Subir a Primer Equipo[Process], Aceptar Oferta Traspaso[Process], Rechazar Oferta Traspaso[Process], Vender Jugador[Process], Negociar Oferta Traspaso[Process], Poner Transferible[Process], Poner No Transferible[Process]

**Data within flow:** {id\_jug+valor+sueldo+años\_contrato+juvenil+id\_oferta}

**Dataflow:** datos oferta renovación

**From-symbol:** USUARIO[External]

**To-symbol:** Hacer Oferta Renovación[Process]

**Data within flow:** {años\_contrato+sueldo}

**Dataflow:** respuesta oferta renovación

**From-symbol:** Hacer Oferta Renovación[Process]

**To-symbol:** USUARIO[External]

**Data within flow:**

**Dataflow:** id oferta traspaso rechazada

**From-symbol:** USUARIO[External]

**To-symbol:** Rechazar Oferta Traspaso[External]

**Data within flow:**

**Dataflow:** orden subir jugador

**From-symbol:** USUARIO[External]

**To-symbol:** Subir A Primer Equipo [Process]

**Data within flow:**

**Dataflow:** id oferta traspaso aceptada

**From-symbol:** USUARIO[External]

**To-symbol:** Aceptar Oferta Traspaso[External]

**Data within flow:**

**Dataflow:** contraoferta

**From-symbol:** USUARIO[External]

**To-symbol:** Negociar Oferta Traspaso[External]

**Data within flow:** {id\_oferta+cantidad}

**Dataflow:** orden poner transferible  
**From-symbol:** USUARIO[External]  
**To-symbol:** Poner Transferible[Process]  
**Data within flow:**

**Dataflow:** orden poner no transferible  
**From-symbol:** USUARIO[External]  
**To-symbol:** Poner No Transferible[Process]  
**Data within flow:**

**Dataflow:** orden vender jugador  
**From-symbol:** USUARIO[External]  
**To-symbol:** Vender Jugador[Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DATOS

**Datastore:** JUGADORES  
**Input flows**

coming from Hacer Oferta Renovación[Process]

coming from Subir A Primer Equipo[Process]

coming from Aceptar Oferta Traspaso [Process]

coming from Vender Jugador[Process]

**Data within store:** {@id+equipo+temporadas\_contrato+sueldo+juvenil+transferible+valor}

**Datastore:** OFERTAS TRASPASO  
**Input flows**

coming from Rechazar Oferta Traspaso[Process]

coming from Aceptar Oferta Traspaso[Process]

coming from Vender Jugador[Process]

coming from Negociar Oferta Traspaso[Process]

**Data within store:** {@id+jugador+equipo\_comprador+equipo\_vendedor+cantidad+contraoferta}

**Datastore:** EQUIPOS  
**Input flows**

coming from Aceptar Oferta Traspaso[Process]

coming from Vender Jugador[Process]

**Output flows**

going to Aceptar Oferta Traspaso [Process]

**Data within store:** {@id+saldo}

**Datastore:** ECONOMÍA  
**Input flows**

coming from Aceptar Oferta Traspaso[Process]

coming from Vender Jugador[Process]

**Data within store:** {@id+ equipo+ semana+ ingresos\_ventajug}

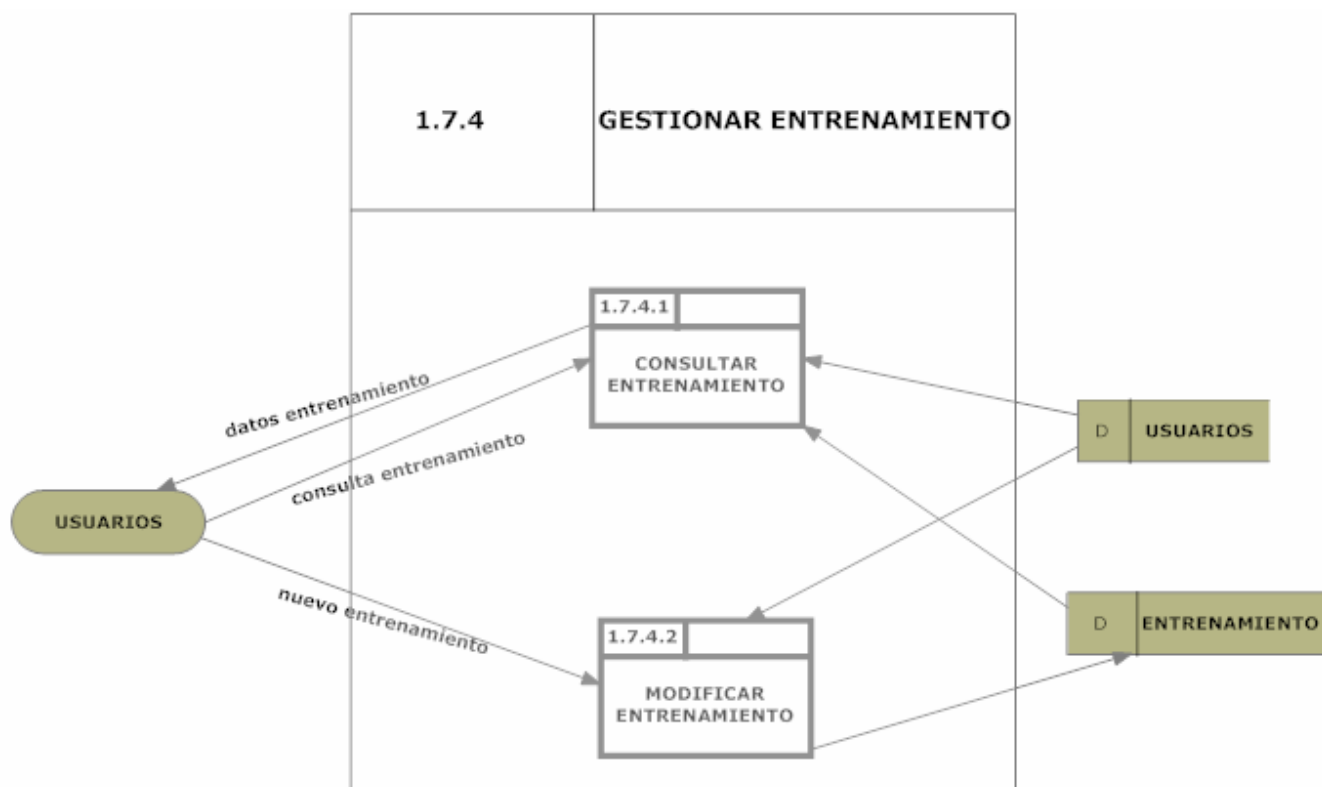


Fig 27. Diagrama de Flujos de Datos 10.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Consultar Entrenamiento

#### Minispec or Description:

PRE1. Un usuario hace una petición para ver el entrenamiento vigente de su equipo.

POST1. Se obtienen los datos del entrenamiento del equipo del usuario y se le muestra.

**Purpose:** Muestra al usuario el entrenamiento de su equipo.

#### Input flows

consulta entrenamiento  
coming from USUARIO[External]

coming from USUARIOS[Data Store]

coming from ALINEACIÓN [Data Store]

#### Output flows

datos entrenamiento  
going to USUARIO[External]

### Process: Modificar Entrenamiento

#### Minispec or Description:

PRE1. Un usuario modifica el entrenamiento de su equipo y manda la nueva disposición del mismo.

POST1. Se actualiza ENTRENAMIENTO con los nuevos datos introducidos por el usuario.

**Purpose:** Modifica el entrenamiento del equipo del usuario actualizando ENTRENAMIENTO.

#### Input flows

nuevo entrenamiento  
coming from USUARIO[External]

coming from USUARIOS[Data Store]

#### Output flows

going to ENTRENAMIENTO[Data Store]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** consulta entrenamiento  
**From-symbol:** USUARIO[External]  
**To-symbol:** Consultar Entrenamiento[Process]  
**Data within flow:** {@id+usuario+equipo}

**Dataflow:** datos entrenamiento  
**From-symbol:** Consultar Entrenamiento [Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {est\_fisico+port+vel+agil+fuer+tec+dis+reg+pase+vjue+res+def+jue\_ae+captact}

**Dataflow:** nuevo entrenamiento  
**From-symbol:** USUARIO[External]  
**To-symbol:** Modificar Entrenamiento [Process]  
**Data within flow:** {equipo+est\_fisico+port+vel+agil+fuer+tec+dis+reg+pase+vjue+res+def+jue\_ae+captact}

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** USUARIOS  
**Output flows**

going to Consultar Entrenamiento[Process]

going to Modificar Entrenamiento[Process]

**Data within store:** {@id+usuario+equipo}

**Datastore:** ENTRENAMIENTO  
**Input flows**

coming from Modificar Entrenamiento[Process]

**Output flows**

going to Consultar Entrenamiento[Process]

**Data within store:** {@id+ equipo+ estado\_fisico+ porteria+ velocidad+ agilidad+ fuerza+ tecnica+ disparo+ regate+ pase+ vision\_juego+ resistencia+ defensa+ juego\_aereo+ capacidad\_tactica}



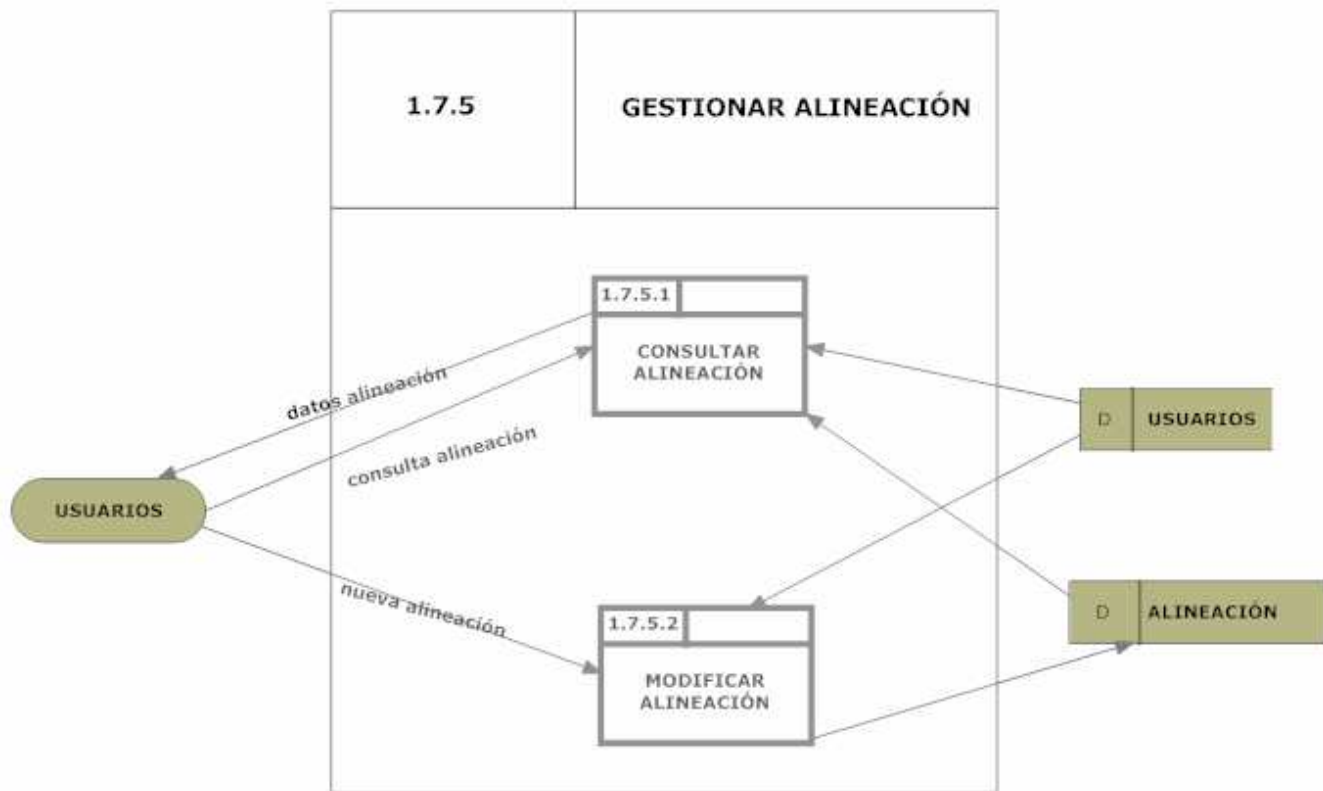


Fig 28. Diagrama de Flujos de Datos 11.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Consultar Alineación

#### Minispec or Description:

PRE1. Un usuario hace una petición para ver la alineación vigente de su equipo.

POST1. Se obtienen los datos de la alineación del equipo del usuario y se le muestra.

**Purpose:** Muestra al usuario la alineación de su equipo.

#### Input flows

- consulta alineación  
coming from USUARIO[External]
- coming from USUARIOS[Data Store]
- coming from ALINEACIÓN [Data Store]

#### Output flows

- datos alineación  
going to USUARIO[External]

### Process: Modificar Alineación

#### Minispec or Description:

PRE1. Un usuario modifica la alineación de su equipo y manda la nueva disposición de la misma.

POST1. Se actualiza ALINEACIÓN con los nuevos datos introducidos por el usuario.

**Purpose:** Modifica la alineación del equipo del usuario actualizando ALINEACIÓN.

#### Input flows

- nueva alineacion  
coming from USUARIO[External]
- coming from USUARIOS[Data Store]

#### Output flows

- going to ALINEACION[Data Store]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** consulta alineacion  
**From-symbol:** USUARIO[External]  
**To-symbol:** Consultar Alineacion[Process]  
**Data within flow:**

**Dataflow:** datos alineacion  
**From-symbol:** Consultar Alineacion[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:** {formacion+fue\_jue+jug1+jug2+jug3+jug4+jug5+jug6+jug7+jug8+jug9+jug10+jug11}

**Dataflow:** nueva alineacion  
**From-symbol:** USUARIO[External]  
**To-symbol:** Modificar Alineacion[Process]  
**Data within flow:** {equipo+formacion+fue\_jue+jug1+jug2+jug3+jug4+jug5+jug6+jug7+jug8+jug9+jug10+jug11}

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** USUARIOS  
**Output flows**

going to Consultar Alineación[Process]

going to Modificar Alineación [Process]

**Data within store:** {@id+usuario+equipo}

**Datastore:** ALINEACIONES  
**Input flows**

coming from Modificar Alineación [Process]

**Output flows**

going to Consultar Alineación [Process]

**Data within store:** {@id+ equipo+ formacion+ fuera\_juega+ juvenil+ jug1+ jug2+ jug3+ jug4+ jug5+ jug6+ jug7+ jug8+ jug9+ jug10+ jug11+ jug12+ jug13+ jug14+ jug15+ jug16}

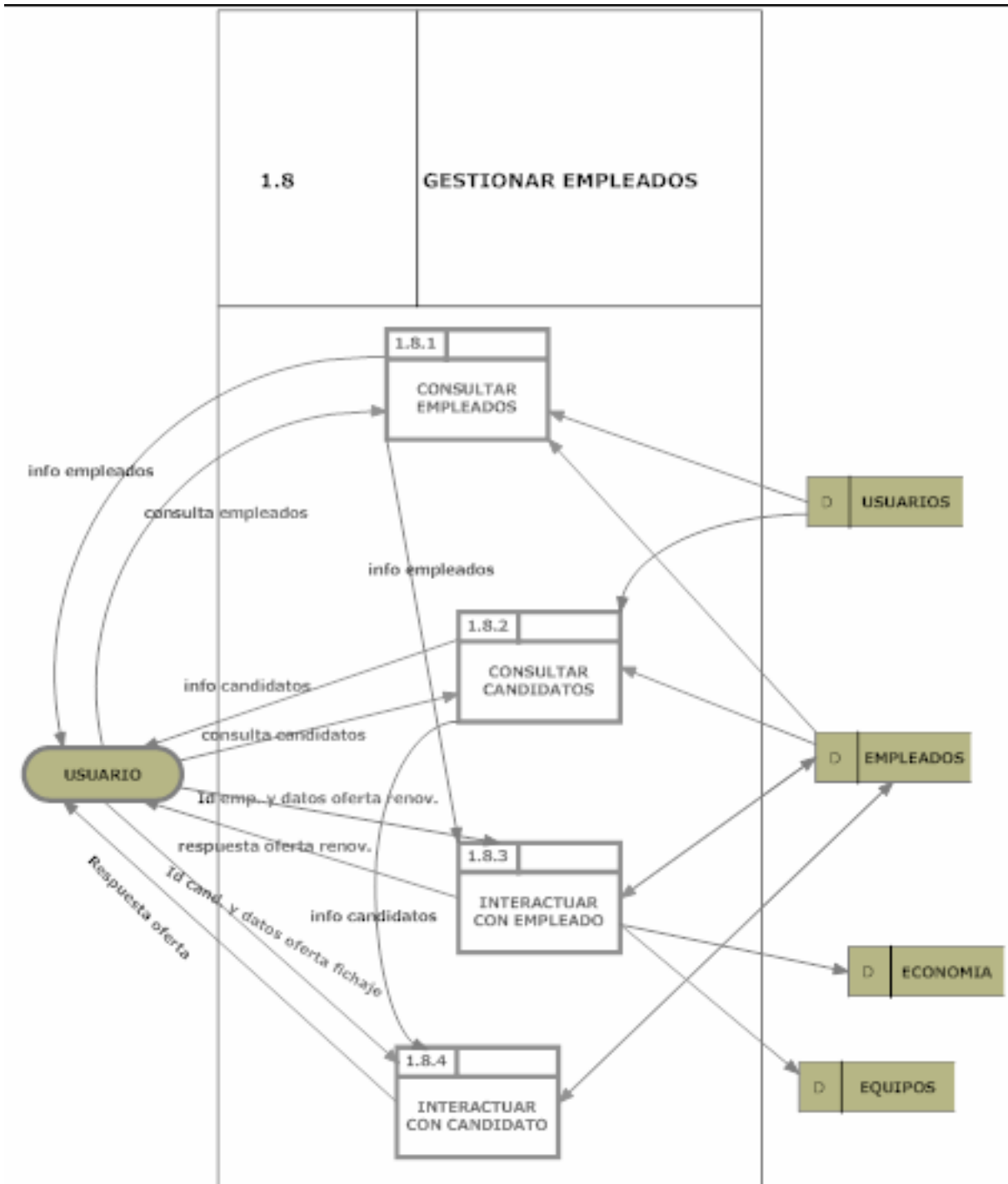


Fig 29. Diagrama de Flujos de Datos 12.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Consultar Empleados

#### Minispec or Description:

PRE1. Un usuario hace una petición para ver los empleados pertenecientes a su equipo.

POST1. Se obtienen los datos de los empleados y se le muestran al usuario.

**Purpose:** Muestra al usuario los empleados de su equipo.

#### Input flows

- consulta empleados
- coming from USUARIO[External]

coming from USUARIOS[Data Store]

coming from EMPLEADOS[Data Store]

**Output flows**

info empleados

going to USUARIO[External], INTERACTUAR CON EMPLEADO[Process]

**Process: Consultar Candidatos**

**Minispec or Description:**

PRE1. Un usuario hace una petición para ver los empleados libres que puede contratar para su equipo.

POST1. Se obtienen los datos de los empleados libres disponibles para el equipo del usuario y se le muestran al usuario.

**Purpose:** Muestra al usuario los empleados candidatos para su equipo.

**Input flows**

consulta candidatos

coming from USUARIO[External]

coming from USUARIOS[Data Store]

coming from EMPLEADOS[Data Store]

**Output flows**

info candidatos

going to USUARIO[External], INTERACTUAR CON CANDIDATO[Process]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** info empleados

**From-symbol:** Consultar Empleados[Process]

**To-symbol:** USUARIO[External], Interactuar Con Empleado[Process]

**Data within flow:** {@id+nombre+ edad+ pais+ categoria+ capacidad+ motivacion+ años\_contrato + sueldo}

**Dataflow:** consulta empleados

**From-symbol:** USUARIO[External]

**To-symbol:** Consultar Empleados[Process]

**Data within flow:**

**Dataflow:** info candidatos

**From-symbol:** Consultar Candidatos[Process]

**To-symbol:** USUARIO[External], Interactuar Con Candidato[Process]

**Data within flow:** {@id+nombre+ edad+ pais+ categoria+ capacidad+ motivacion+ años\_contrato\_solicitados+ sueldo\_solicitado}

**Dataflow:** consulta candidatos

**From-symbol:** USUARIO[External]

**To-symbol:** Consultar Candidatos[Process]

**Data within flow:**

**Dataflow:** datos oferta renovación

**From-symbol:** USUARIO[External]

**To-symbol:** Interactuar Con Empleado[Process]

**Data within flow:** {años\_contrato\_ofrecidos+sueldo\_ofrecido}

**Dataflow:** datos oferta fichaje

**From-symbol:** USUARIO[External]

**To-symbol:** Interactuar Con Candidato[Process]

**Data within flow:** {años\_contrato+sueldo}

**Dataflow:** respuesta oferta renovacion

**From-symbol:** Interactuar Con Empleado[Process]

**To-symbol:** USUARIO[External]

**Data within flow:**

**Dataflow:** respuesta oferta  
**From-symbol:** Interactuar Con Candidato[Process]  
**To-symbol:** USUARIO[External]  
**Data within flow:**

**Dataflow:** id empleado  
**From-symbol:** USUARIO[External]  
**To-symbol:** Interactuar Con Empleado[Process]  
**Data within flow:**

**Dataflow:** id candidato  
**From-symbol:** USUARIO[External]  
**To-symbol:** Interactuar Con Candidato[Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** USUARIOS

**Output flows**

going to Consultar Empleados[Process]

going to Consultar Candidatos[Process]

**Data within store:** {@id+usuario+equipo}

**Datastore:** EMPLEADOS

**Input flows**

coming from Interactuar Con Candidato[Process]

coming from Interactuar Con Empleado[Process]

**Output flows**

going to Consultar Empleados[Process]

going to Consultar Candidatos[Process]

going to Interactuar Con Candidato[Process]

going to Interactuar Con Empleado[Process]

**Data within store:** {@id+nombre+ edad+ pais+ categoria+ capacidad+ motivacion+ años\_contrato + sueldo }

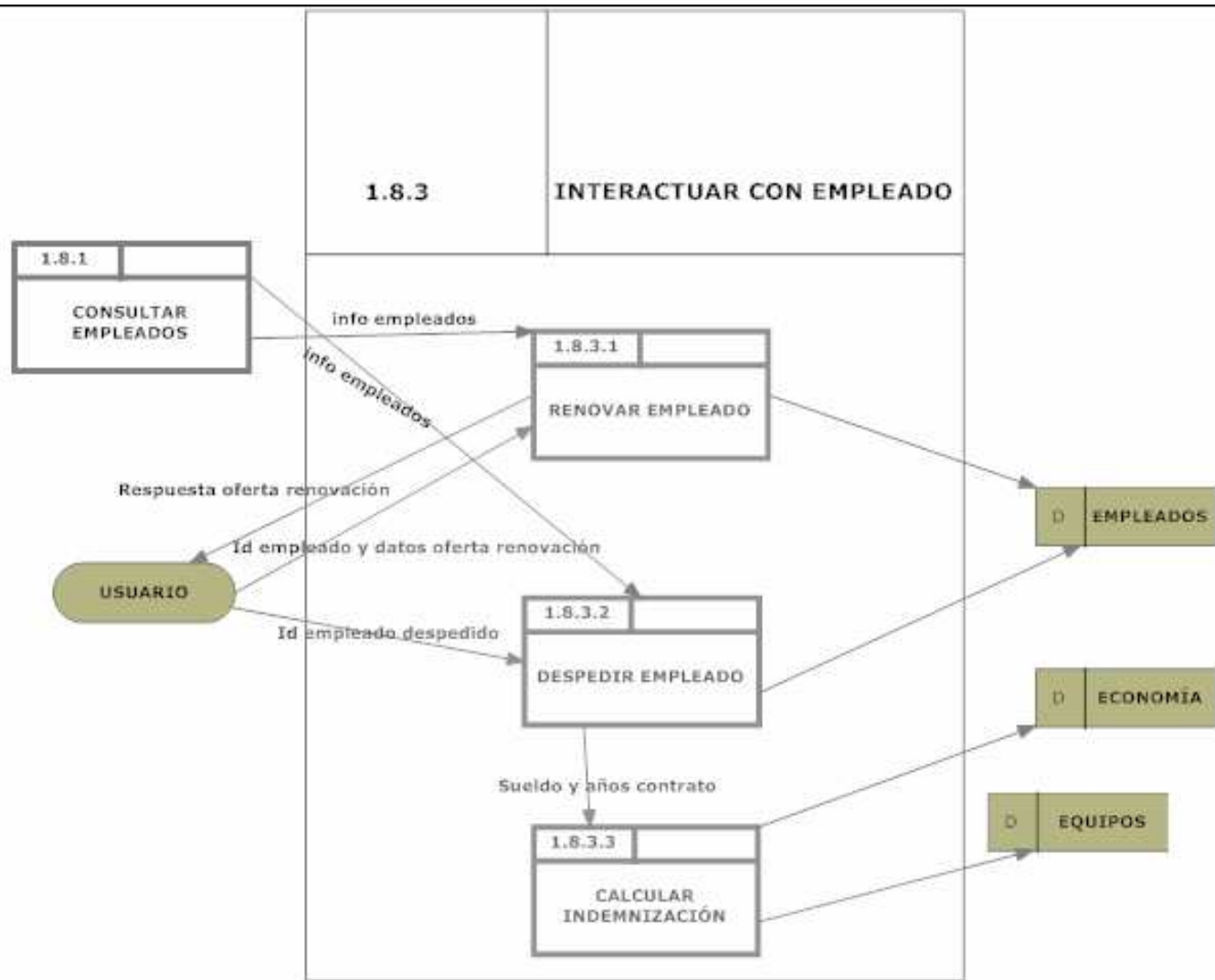


Fig 30. Diagrama de Flujos de Datos 13.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Renovar Empleado

#### Minispec or Description:

PRE1. Un usuario hace una oferta de renovación de contrato a un empleado que pertenece a su equipo.

POST1. Se leen los datos del contrato actual del empleado.

PRE2. Se compara la oferta con el contrato vigente y se decide si la oferta es aceptable.

POST2. Si la oferta no es aceptable se comunica al usuario, y si sí lo se actualizan los datos del contrato del empleado en el almacén de datos EMPLEADOS.

**Purpose:** Efectúa una oferta de renovación de contrato a un empleado del equipo del usuario.

#### Input flows

- id empleado  
coming from USUARIO[External]
- datos oferta renovación  
coming from USUARIO[External]
- info empleados  
coming from CONSULTAR EMPLEADOS[Process]

#### Output flows

- respuesta oferta renovación  
going to USUARIO[External]
- going to EMPLEADOS[Data Store]

**Process: Despedir Empleado****Minispec or Description:**

PRE1. Llega el identificador de un empleado al que usuario quiere despedir.

POST1. Se leen los datos del contrato del empleado y se le marca como empleado libre (sin equipo), actualizando EMPLEADOS.

**Purpose:** Rechaza el ofrecimiento de un empleado sin equipo a un determinado club.

**Input flows**

id empleado despedido  
coming from USUARIO[External]  
info empleados  
coming from CONSULTAR EMPLEADOS[Process]

**Output flows**

sueldo y años contrato  
going to CALCULAR INDEMNIZACIÓN[Process]  
  
going to EMPLEADOS[Data Store]

**Process: Calcular Indemnización****Minispec or Description:**

PRE1. Llegan el sueldo y los años de contrato restantes de un empleado que ha sido despedido por un equipo.

POST1. Se genera la cantidad en concepto de indemnización que debe pagar el equipo por despedir al usuario.

**Purpose:** Calcula la indemnización que debe pagar un equipo despedir a uno de sus empleados.

**Input flows**

sueldo y años contrato  
coming from DESPEDIR EMPLEADO[Process]

**Output flows**

going to EQUIPOS[Data Store]  
  
going to ECONOMIA[Data Store]

**DESCRIPCIÓN FLUJOS DE DATOS**

**Dataflow:** Id empleado

**From-symbol:** USUARIO[External]

**To-symbol:** Renovar Candidato[Process]

**Data within flow:**

**Dataflow:** datos oferta renovación

**From-symbol:** USUARIO[External]

**To-symbol:** Renovar Empleado[Process]

**Data within flow:** {años\_contrato\_ofrecidos+sueldo\_ofrecido}

**Dataflow:** info empleados

**From-symbol:** Consultar Empleados[Process]

**To-symbol:** Renovar Empleado[Process], Despedir Empleado[Process]

**Data within flow:** {@id+equipo+años\_contrato+sueldo }

**Dataflow:** Respuesta oferta renovación

**From-symbol:** Renovar Empleado[Process]

**To-symbol:** USUARIO[External]

**Data within flow:**

**Dataflow:** Id empleado despedido

**From-symbol:** USUARIO[External]

**To-symbol:** Despedir Empleado[Process]

**Data within flow:**

**Dataflow:** Sueldo y años contrato  
**From-symbol:** Despedir Empleado[Process]  
**To-symbol:** Calcular Indemnización[Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** EMPLEADOS  
**Input flows**

input from Renovar Empleado[Process]

input from Despedir Empleado[Process]

**Data within store:** {@id+equipo+años\_contrato +sueldo }

**Datastore:** EQUIPOS  
**Input flows**

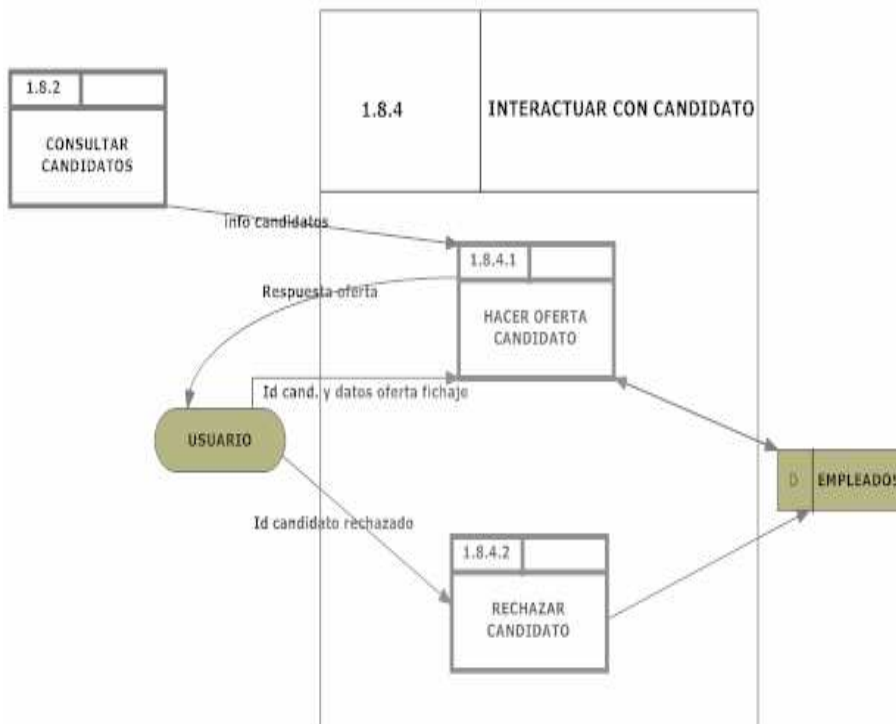
input from Calcular Indemnización [Process]

**Data within store:** {@id+saldo}

**Datastore:** ECONOMÍA  
**Input flows**

input from Calcular Indemnización [Process]

**Data within store:** {@id+ equipo+ gastos\_varios}



**Fig 30. Diagrama de Flujos de Datos 14.**



## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Hacer Oferta Candidato

#### Minispec or Description:

PRE1. Un usuario hace una oferta a un candidato para que pase a ser empleado del equipo que controla.

POST1. Se leen la petición contractual del candidato.

PRE2. Se compara la oferta con la petición y se comprueba si la oferta es aceptable.

POST2. Si la oferta no es aceptable se comunica al usuario, y si sí lo es el candidato pasa a ser empleado del equipo del usuario, actualizando EMPLEADOS.

**Purpose:** Efectúa una oferta por parte del usuario a un candidato a ser empleado de su equipo.

#### Input flows

id cand

coming from USUARIO[External]

datos oferta fichaje

coming from USUARIO[External]

info candidatos

coming from CONSULTAR CANDIDATOS[Process]

#### Output flows

respuesta oferta

going to USUARIO[External]

going to EMPLEADOS[Data Store]

### Process: Rechazar Candidato

#### Minispec or Description:

PRE1. Llega el identificador de un candidato indicado por el usuario a ser rechazado.

POST1. Se indica que ese empleado libre(candidato) no sea ofrecido al equipo del usuario actualizando EMPLEADOS.

**Purpose:** Rechaza el ofrecimiento de un empleado sin equipo a un determinado club.

#### Input flows

id candidato rechazado

coming from USUARIO[External]

#### Output flows

going to EMPLEADOS[Data Store]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** Id cand

**From-symbol:** USUARIO[External]

**To-symbol:** Hacer Oferta Candidato[Process]

**Data within flow:**

**Dataflow:** datos oferta fichaje

**From-symbol:** USUARIO[External]

**To-symbol:** Hacer Oferta Candidato[Process]

**Data within flow:** {años\_contrato+sueldo}

**Dataflow:** info candidatos

**From-symbol:** Consultar Candidatos[Process]

**To-symbol:** Hacer Oferta Candidato[Process]

**Data within flow:** {@id+años\_contrato\_solicitados+sueldo\_solicitado}

**Dataflow:** Respuesta oferta

**From-symbol:** Hacer Oferta Candidato[Process]

**To-symbol:** USUARIO[External]

**Data within flow:**

**Dataflow:** Id candidato rechazado

**From-symbol:** USUARIO[External]

**To-symbol:** Rechazar Candidato[Process]  
**Data within flow:**

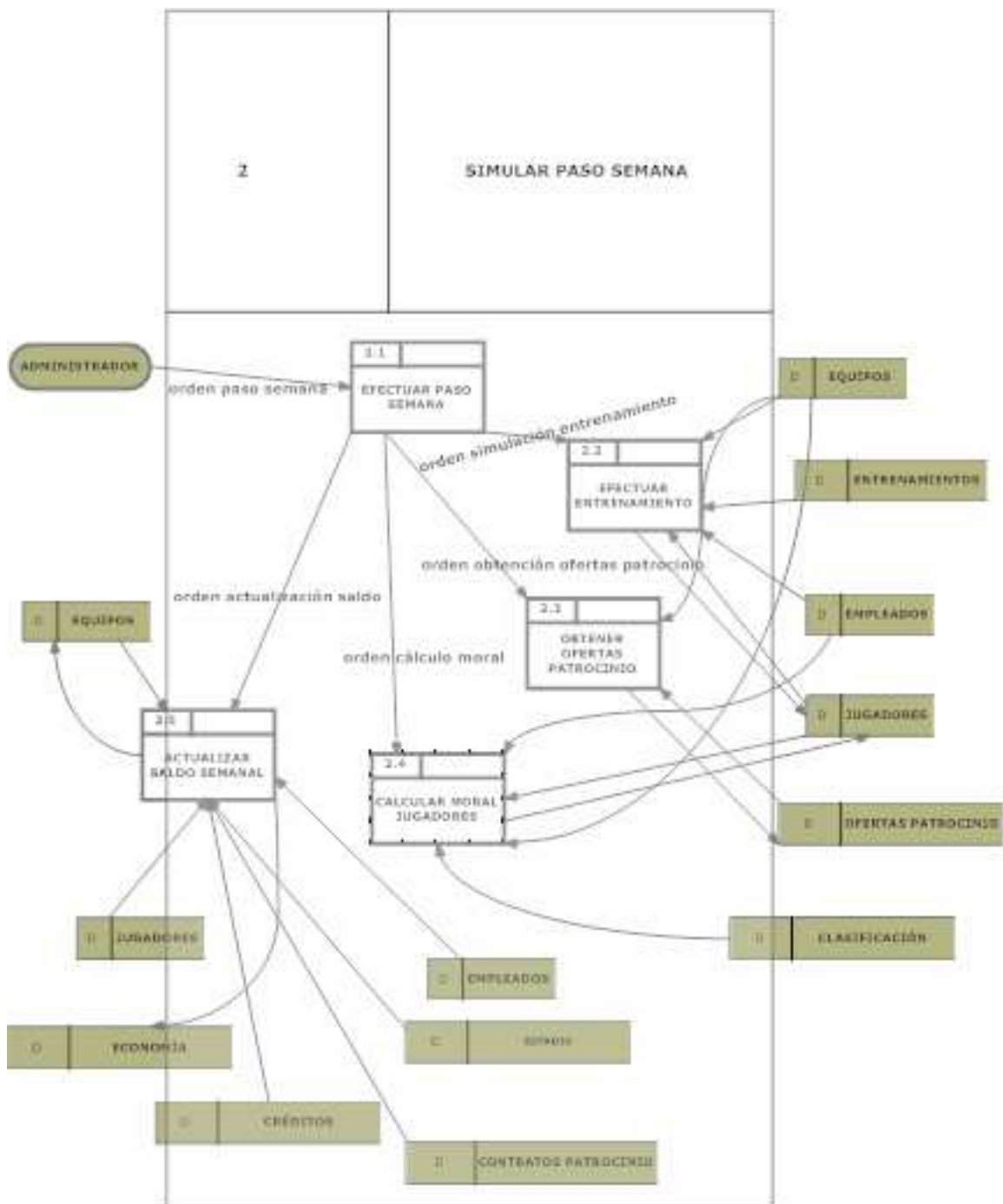
## ESTRUCTURA ALMACENES DE DATOS

**Datstore:** EMPLEADOS  
**Input flows**

input from Hacer Oferta Candidato[Process]

input from Rechazar Candidato [Process]

**Data within store:** { @id+equipo+años\_contrato +sueldo }



**Fig 31. Diagrama de Flujos de Datos 15.**

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### **Process: Efectuar Paso Semana**

#### **Minispec or Description:**

PRE1. Llega una orden para simular un paso de semana.

POST1. Se emiten todas las órdenes necesarias para comenzar los procesos implicados en la simulación de un paso de semana.

**Purpose:** Simula lo que ocurriría cuando se pasa de una semana a otra si la competición implementada por el sistema tuviera lugar en tiempo real.

#### **Input flows**

orden paso semana  
coming from ADMINISTRADOR [External]

#### **Output flows**

orden simulación entrenamiento  
going to EFECTUAR ENTRENAMIENTO[Process]  
orden obtención ofertas patrocinio  
going to OBTENER OFERTAS PATROCINIO[Process]  
orden cálculo moral  
going to CALCULAR MORAL JUGADORES[Process]  
orden actualización saldo  
going to ACTUALIZAR SALDO SEMANAL[Process]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** Orden paso semana

**From-symbol:** ADMINISTRADOR[External]

**To-symbol:** Efectuar Paso Semana[Process]

**Data within flow:**

**Dataflow:** orden actualización saldo

**From-symbol:** Efectuar Paso Semana[Process]

**To-symbol:** Actualizar Saldo Semanal[Process]

**Data within flow:**

**Dataflow:** orden cálculo moral

**From-symbol:** Efectuar Paso Semana[Process]

**To-symbol:** Calcular Moral Jugadores[Process]

**Data within flow:**

**Dataflow:** orden obtención ofertas patrocinio

**From-symbol:** Efectuar Paso Semana[Process]

**To-symbol:** Obtener Ofertas Patrocinio[Process]

**Data within flow:**

**Dataflow:** orden simulación entrenamiento

**From-symbol:** Efectuar Paso Semana[Process]

**To-symbol:** Efectuar Entrenamiento[Process]

**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** EQUIPOS

#### **Input flows**

coming from Actualizar Saldo Semanal[Process]

#### **Output flows**

going to Efectuar Entrenamiento[Process]

going to Obtener Ofertas Patrocinio[Process]

going to Calcular Moral Jugadores[Process]

**Data within store:** {@id+saldo}

**Datastore:** ENTRENAMIENTO

**Output flows**

going to Efectuar Entrenamiento[Process]

**Data within store: Data within store:** {@id+ equipo+ estado\_fisico+ porteria+ velocidad+ agilidad+ fuerza+ tecnica+ disparo+ regate+ pase+ vision\_juego+ resistencia+ defensa+ juego\_aereo+ capacidad\_tactica}

**Datastore:** JUGADORES

**Input flows**

coming from Calcular Moral Jugadores[Process]

coming from Efectuar Entrenamiento[Process]

**Output flows**

going to Efectuar Entrenamiento[Process]

going to Calcular Moral Jugadores[Process]

going to Actualizar Saldo Semanal[Process]

**Data within store:** {@id+ equipo+ partidos\_jugados+moral+sueldo+ posicion+ edad+ est\_fis+ port+ vel+ agil+ fuer+ tec+ disp+ rega+ pase+ vis\_jue+ resis+ def+ jue\_ae+ cap\_tac+ est\_fis\_pot+ port\_pot + vel\_pot + agil\_pot + fuer\_pot + tec\_pot + disp\_pot + rega\_pot + pase\_pot + vis\_jue\_pot + resis\_pot + def\_pot + jue\_ae\_pot + cap\_tac\_pot}

**Datastore:** OFERTAS PATROCINIO

**Input flows**

coming from Obtener Ofertas Patrocinio[Process]

**Output flows**

going to Obtener Ofertas Patrocinio[Process]

**Data within store:** {@id+equipo+pago\_fijo+pago\_victoria+temporadas+semanas\_vigencia}

**Datastore:** CLASIFICACIÓN

**Output flows**

going to Calcular Moral Jugadores[Process]

**Data within store:** {@id+equipo+partidos\_jugados+puesto}

**Datastore:** CONTRATOS PATROCINIO

**Output flows**

going to Actualizar Saldo Semanal[Process]

**Data within store:** {@id + equipo+ pago\_fijo+ pago\_victoria}

**Datastore:** ECONOMÍA

**Input flows**

coming from Actualizar Saldo Semanal[Process]

**Data within store:** {@id + equipo+ ingresos\_mercadotecnia+ gastos\_sueldosjug+ gastos\_sueldosemp+ gastos\_obras+ gastros\_credito+ ingresos\_patro}

**Datastore: CRÉDITOS**  
**Output flows**

going to Actualizar Saldo Semanal[Process]  
**Data within store:** {@id + equipo+devolucion\_semanal}

**Datastore: EMPLEADOS**  
**Output flows**

going to Efectuar Entrenamiento[Process]

going to Calcular Moral Jugadores[Process]

going to Actualizar Saldo Semanal[Process]

**Data within store:** {@id+equipo+categoria+capacidad+motivacion+temp\_contrato+sueldo}

**Datastore: ESTADIO**  
**Output flows**

going to Actualizar Saldo Semanal[Process]

**Data within store:** {@id + equipo+ tienda}

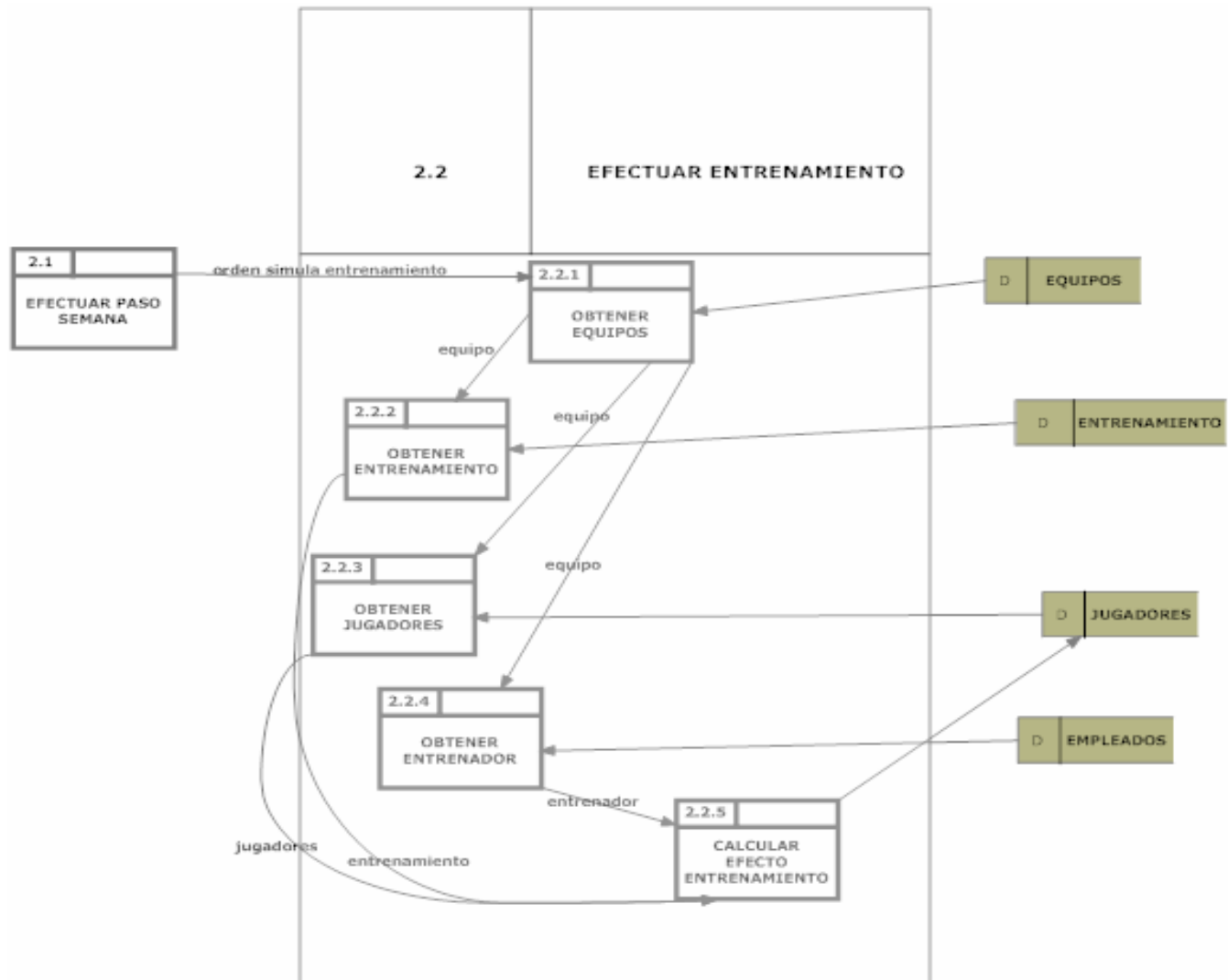


Fig 32. Diagrama de Flujos de Datos 16.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Obtener Equipos

#### Minispec or Description:

PRE1. Llega una orden simulación de entrenamiento a partir de la simulación del paso de semana.

POST1. Se obtienen todos los equipos existentes en el sistema.

**Purpose:** Lee todos los equipos existentes en el juego.

#### Input flows

coming from EQUIPOS [Data Store]  
orden simulación entrenamiento  
coming from EFECTUAR PASO SEMANA [Process]

#### Output flows

equipo  
going to OBTENER ENTRENAMIENTO[Process], OBTENER JUGADORES[Process], OBTENER ENTRENADOR[Process]

### Process: Obtener Entrenamiento

#### Minispec or Description:

PRE1. Llega el identificador de un equipo del que se quiere obtener sus opciones de entrenamiento.

POST1. Se obtienen los datos del entrenamiento de ese equipo.

**Purpose:** Obtiene el entrenamiento de un equipo para hacer el cálculo de los efectos del mismo sobre los jugadores.

#### Input flows

equipo  
coming from OBTENER EQUIPOS[Process]  
coming from ENTRENAMIENTO[Data Store]

#### Output flows

entrenamiento  
going to CALCULAR EFECTO ENTRENAMIENTO[Process]

### Process: Obtener Jugadores

#### Minispec or Description:

PRE1. Llega el identificador de un equipo del que se quieren obtener algunos datos de todos sus jugadores.

POST1. Se leen los datos de todos los jugadores pertenecientes al equipo.

**Purpose:** Lee algunos datos susceptibles de ser modificados por el efecto del entrenamiento simulado en el paso de semana.

#### Input flows

equipo  
coming from OBTENER EQUIPOS[Process]  
coming from JUGADORES[Data Store]

#### Output flows

jugadores  
going to CALCULAR EFECTO ENTRENAMIENTO[Process]

### Process: Obtener Entrenador

#### Minispec or Description:

PRE1. Llega el identificador de un equipo del que se quieren obtener algunos datos sobre su entrenador.

POST1. Se obtienen los datos del entrenador de ese equipo.

**Purpose:** Obtiene los datos del entrenador de un equipo para hacer el cálculo de los efectos del entrenamiento sobre los jugadores.

#### Input flows

equipo

coming from OBTENER EQUIPOS[Process]

coming from EMPLEADOS[Data Store]

**Output flows**

entrenador

going to CALCULAR EFECTO ENTRENAMIENTO[Process]

**Process: Calcular Efecto Entrenamiento**

**Minispec or Description:**

PRE1. Llegan los datos de unos jugadores, de un entrenador y de la configuración de entrenamiento del equipo al que pertenecen jugadores y entrenador.

POST1. Se calcula el nuevo valor de algunos atributos de los jugadores a partir de los datos de entrenamiento, nivel del entrenador y edad y posición del jugador, actualizando JUGADORES.

**Purpose:** Simula el efecto del entrenamiento en los jugadores.

**Input flows**

entrenador

coming from OBTENER ENTRENAMIENTO[Process]

entrenamiento

coming from OBTENER ENTRENAMIENTO[Process]

jugadores

coming from OBTENER JUGADORES[Process]

**Output flows**

going to JUGADORES[Data Store]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** Orden simulación entrenamiento

**From-symbol:** Efectuar Paso Semana[Process]

**To-symbol:** Obtener Equipos[Process]

**Data within flow:**

**Dataflow:** Equipo

**From-symbol:** Obtener Equipos[Process]

**To-symbol:** Obtener Entrenamiento[Process], Obtener Jugadores [Process], Obtener Entrenador[Process]

**Data within flow:**

**Dataflow:** jugadores

**From-symbol:** Obtener Jugadores [Process]

**To-symbol:** Calcular Efecto Entrenamiento[Process]

**Data within flow:** {id+ posicion+ edad+ est\_fis+ port+ vel+ agil+ fuer+ tec+ disp+ rega+ pase+ vis\_jue+ resis+ def+ jue\_ae+ cap\_tac+ est\_fis\_pot+ port\_pot + vel\_pot + agil\_pot + fuer\_pot + tec\_pot + disp\_pot + rega\_pot + pase\_pot + vis\_jue\_pot + resis\_pot + def\_pot + jue\_ae\_pot + cap\_tac\_pot}

**Dataflow:** entrenamiento

**From-symbol:** Obtener Entrenamiento[Process]

**To-symbol:** Calcular Efecto Entrenamiento[Process]

**Data within flow:** {estado\_fisico+ porteria+ velocidad+ agilidad+ fuerza+ tecnica+ disparo+ regate+ pase+ vision\_juego+ resistencia+ defensa+ juego\_aereo+ capacidad\_tactica}

**Dataflow:** entrenador

**From-symbol:** Obtener Entrenador[Process]

**To-symbol:** Calcular Efecto Entrenamiento[Process]

**Data within flow:** {capacidad+motivacion}

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** EQUIPOS

**Output flows**

going to Obtener Equipos[Process]  
**Data within store:** {@id}

**Datastore:** ENTRENAMIENTO  
**Output flows**

going to Obtener Entrenamiento [Process]  
**Data within store:** {@id+ equipo+ estado\_fisico+ porteria+ velocidad+ agilidad+ fuerza+ tecnica+ disparo+ regate+ pase+ vision\_juego+ resistencia+ defensa+ juego\_aereo+ capacidad\_tactica}

**Datastore:** JUGADORES  
**Input flows**

coming from Calcular Efecto Entrenamiento[Process]  
**Output flows**

going to Obtener Jugador [Process]  
**Data within store:** {@id+ equipo+ posicion+ edad+ est\_fis+ port+ vel+ agil+ fuer+ tec+ disp+ rega+ pase+ vis\_jue+ resis+ def+ jue\_ae+ cap\_tac+ est\_fis\_pot+ port\_pot + vel\_pot + agil\_pot + fuer\_pot + tec\_pot + disp\_pot + rega\_pot + pase\_pot + vis\_jue\_pot + resis\_pot + def\_pot + jue\_ae\_pot + cap\_tac\_pot}

**Datastore:** EMPLEADOS  
**Output flows**

going to Obtener Entrenador [Process]  
**Data within store:** {@id+ equipo+ capacidad+ motivacion}

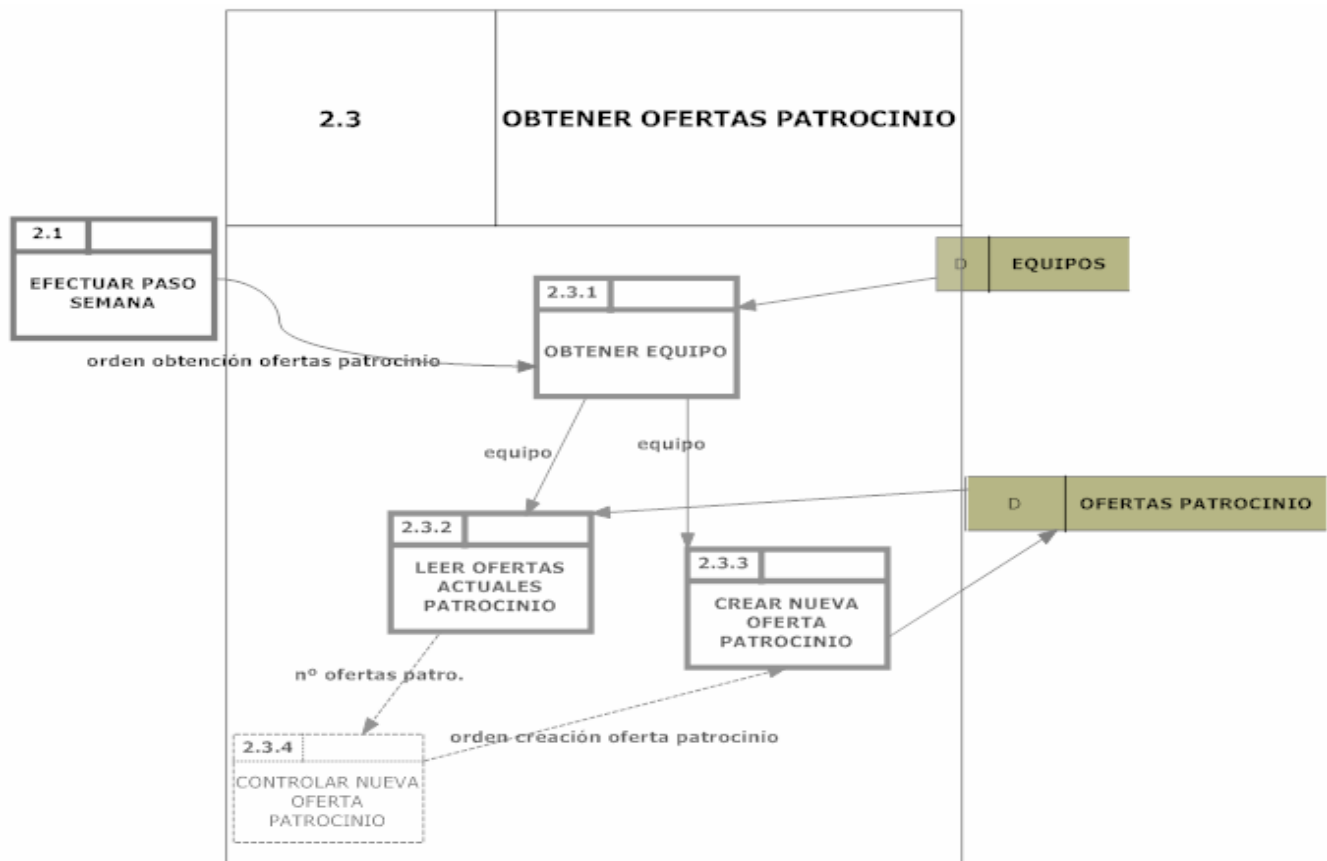


Fig 33. Diagrama de Flujos de Datos 17.



## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Obtener Equipos

#### Minispec or Description:

PRE1. Llega una orden de obtención de ofertas de patrocinio a partir de la simulación del paso de semana.

POST1. Se obtienen todos los equipos existentes en el sistema.

**Purpose:** Lee todos los equipos existentes en el juego.

#### Input flows

coming from EQUIPOS [Data Store]  
orden obtención ofertas patrocinio  
coming from EFECTUAR PASO SEMANA [Process]

#### Output flows

equipo  
going to LEER OFERTAS ACTUALES PATROCINIO[Process], CREAR NUEVA OFERTA PATROCINIO[Process]

### Process: Leer Ofertas Actuales Patrocinio

#### Minispec or Description:

PRE1. Llega el identificador de un equipo del que se quieren obtener el número de ofertas de patrocinio que tiene en ese momento.

POST1. Se obtuvo el número de ofertas de patrocinio vigentes de ese equipo.

**Purpose:** Obtiene el número de ofertas de patrocinio con las que cuenta un equipo en un determinado momento.

#### Input flows

equipo  
coming from OBTENER EQUIPOS[Process]

coming from OFERTAS PATROCINIO[Data Store]

#### Output flows

nº ofertas patro.  
going to CONTROLAR NUEVA OFERTA PATROCINIO[Process]

### Process: Controlar Nueva Oferta Patrocinio

#### Minispec or Description:

PRE1. Llega el número de ofertas de patrocinio vigentes de un determinado equipo.

POST1. Se emite una orden de creación de nueva oferta de patrocinio en caso de que el número que llegue sea menor que 4.

**Purpose:** Ordena la creación de una oferta de patrocinio para todos aquellos equipos que tengan menos de 4 ofertas.

#### Input flows

nº ofertas patro.  
coming from to LEER OFERTAS ACTUALES PATROCINIO [Process]

#### Output flows

orden creación oferta patrocinio  
going to CREAR NUEVA OFERTA PATROCINIO[Process]

### Process: Crear Nueva Oferta Patrocinio

#### Minispec or Description:

PRE1. Llega una orden de creación de oferta de patrocinio y el identificador del equipo al que se le tiene que añadir la oferta.

POST1. Se crea una oferta de patrocinio para aquellos equipos con menos de 4 ofertas vigentes y actualiza OFERTAS PATROCINIO.

**Purpose:** Crea una nueva oferta de patrocinio y la guarda en OFERTAS PATROCINIO.

#### Input flows

orden creación oferta patrocinio  
coming from CONTROLAR NUEVA OFERTA PATROCINIO[Process]  
equipo

coming from OBTENER EQUIPOS[Process]  
**Output flows**

going to OFERTAS PATROCINIO[Data Store]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** Orden obtención ofertas patrocinio  
**From-symbol:** Efectuar Paso Semana[Process]  
**To-symbol:** Obtener Equipos[Process]  
**Data within flow:**

**Dataflow:** Equipo  
**From-symbol:** Obtener Equipos[Process]  
**To-symbol:** Leer Ofertas Actuales Patrocinio[Process], Crear Nueva Oferta Patrocinio[Process]  
**Data within flow:**

**Dataflow:** nº ofertas patro  
**From-symbol:** Leer Ofertas Actuales Patrocinio[Process],  
**To-symbol:** Controlar Nueva Oferta Patrocinio[Process]  
**Data within flow:**

**Dataflow:** orden creación oferta patrocinio  
**From-symbol:** Controlar Nueva Oferta Patrocinio[Process]  
**To-symbol:** Crear Nueva Oferta Patrocinio [Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** EQUIPOS  
**Output flows**

going to Obtener Equipos[Process]  
**Data within store:** {@id}

**Datastore:** OFERTAS PATROCINIO  
**Input flows**

coming from Crear Nueva Oferta Patrocinio [Process]  
**Output flows**

going to Leer Ofertas Actuales Patrocinio [Process]  
**Data within store:** {@id+equipo+pago\_fijo+pago\_victoria+temporadas+semanas\_vigencia}

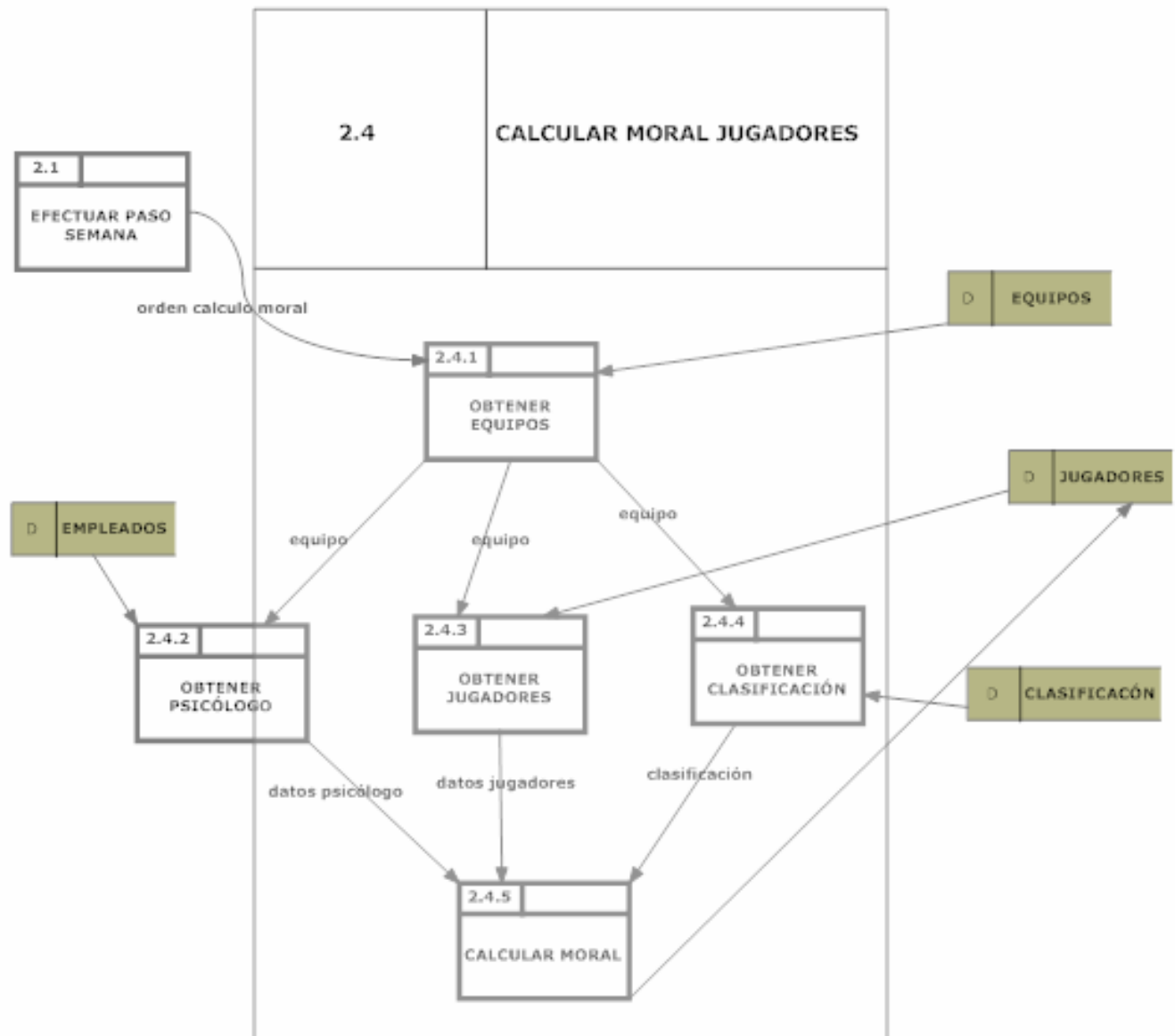


Fig 34. Diagrama de Flujos de Datos 18.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Obtener Equipos

#### Minispec or Description:

PRE1. Llega una orden de actualización semanal del atributo "moral" de cada jugador.

POST1. Se obtienen todos los equipos existentes en el sistema.

**Purpose:** Lee todos los equipos existentes en el juego.

#### Input flows

- coming from EQUIPOS [Data Store]
- orden calculo moral
- coming from EFECTUAR PASO SEMANA[Process]

#### Output flows

- equipo
- going to OBTENER PSICÓLOGO[Process], OBTENER JUGADORES[Process], OBTENER CLASIFICACIÓN[Process]

### Process: Obtener Jugadores

#### Minispec or Description:

PRE1. Llega el identificador de un equipo del que se quieren obtener algunos datos (moral y partidos jugados) de todos sus jugadores.

POST1. Se leen los datos de todos los jugadores pertenecientes al equipo.

**Purpose:** Lee algunos datos implicados en el cálculo del atributo moral de todos los jugadores pertenecientes a un equipo.

**Input flows**

equipo  
coming from OBTENER EQUIPOS[Process]  
  
coming from JUGADORES[Data Store]

**Output flows**

datos jugadores  
going to CALCULAR MORAL[Process]

**Process: Obtener Psicólogo**

**Minispec or Description:**

PRE1. Llega el identificador de un equipo del que se quiere obtener los datos de empleado de categoría "psicólogo".

POST1. Se leen los datos referentes al psicólogo del equipo.

**Purpose:** Proporciona los datos del psicólogo de un equipo.

**Input flows**

equipo  
coming from OBTENER EQUIPOS[Process]  
  
coming from EMPLEADOS[Data Store]

**Output flows**

datos psicólogo  
going to CALCULAR MORAL[Process]

**Process: Obtener Clasificación**

**Minispec or Description:**

PRE1. Llega el identificador de un equipo del que se quiere obtener su clasificación.

POST1. Se obtiene la clasificación del equipo.

**Purpose:** Lee la clasificación de un equipo.

**Input flows**

equipo  
coming from OBTENER EQUIPOS[Process]  
  
coming from CLASIFICACIÓN[Data Store]

**Output flows**

clasificación  
going to CALCULAR MORAL[Process]

**Process: Calcular Moral**

**Minispec or Description:**

PRE1. Llegan los datos de un jugador (moral y partidos jugados) así como la clasificación de su equipo y los datos del empleado "psicólogo" de su equipo.

POST1. Se calcula la nueva moral del jugador a partir de su moral vigente, de la relación entre partidos jugados por el jugador y por el equipo, el puesto en la clasificación de su equipo y el nivel del psicólogo, actualizando JUGADORES.

**Purpose:** Calcula el nuevo valor del atributo "moral" para cada jugador en la simulación de cambio de semana.

**Input flows**

datos jugadores  
coming from OBTENER JUGADORES[Process]  
datos psicólogo  
coming from OBTENER PSICÓLOGO[Process]  
clasificación  
coming from OBTENER CLASIFICACIÓN[Process]

**Output flows**

going to JUGADORES[Data Store]

## DESCRIPCIÓN FLUJOS DE DATOS

**Dataflow:** Orden cálculo moral

**From-symbol:** Efectuar Paso Semana[Process]

**To-symbol:** Obtener Equipos[Process]

**Data within flow:**

**Dataflow:** Equipo

**From-symbol:** Obtener Equipos[Process]

**To-symbol:** Obtener Psicólogo[Process], Obtener Jugadores[Process], Obtener Clasificación[Process]

**Data within flow:**

**Dataflow:** Datos psicólogo

**From-symbol:** Obtener Psicólogo [Process]

**To-symbol:** Calcular Moral [Process]

**Data within flow:** {moral+capacidad}

**Dataflow:** Datos jugadores

**From-symbol:** Obtener Jugadores[Process]

**To-symbol:** Calcular Moral [Process]

**Data within flow:** {id+moral+partidos\_jugados}

**Dataflow:** clasificación

**From-symbol:** Obtener Clasificación[Process]

**To-symbol:** Calcular Moral [Process]

**Data within flow:** { partidos\_jugados+puesto}

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** EQUIPOS

**Input flows**

coming from Calcular Moral [Process]

**Output flows**

going to Obtener Equipos[Process]

**Data within store:** {@id}

**Datastore:** JUGADORES

**Input flows**

coming from Calcular Moral [Process]

**Output flows**

going to Obtener Jugadores [Process]

**Data within store:** {@id+ equipo+ partidos\_jugados+moral}

**Datastore:** CLASIFICACIÓN

**Output flows**

going to Obtener Clasificación[Process]

**Data within store:** {@id+equipo+partidos\_jugados+puesto}

**Datastore:** EMPLEADOS

**Output flows**

going to Obtener Empleados[Process]  
**Data within store:** {@id+equipo+categoria+capacidad+motivacion+temp\_contrato}

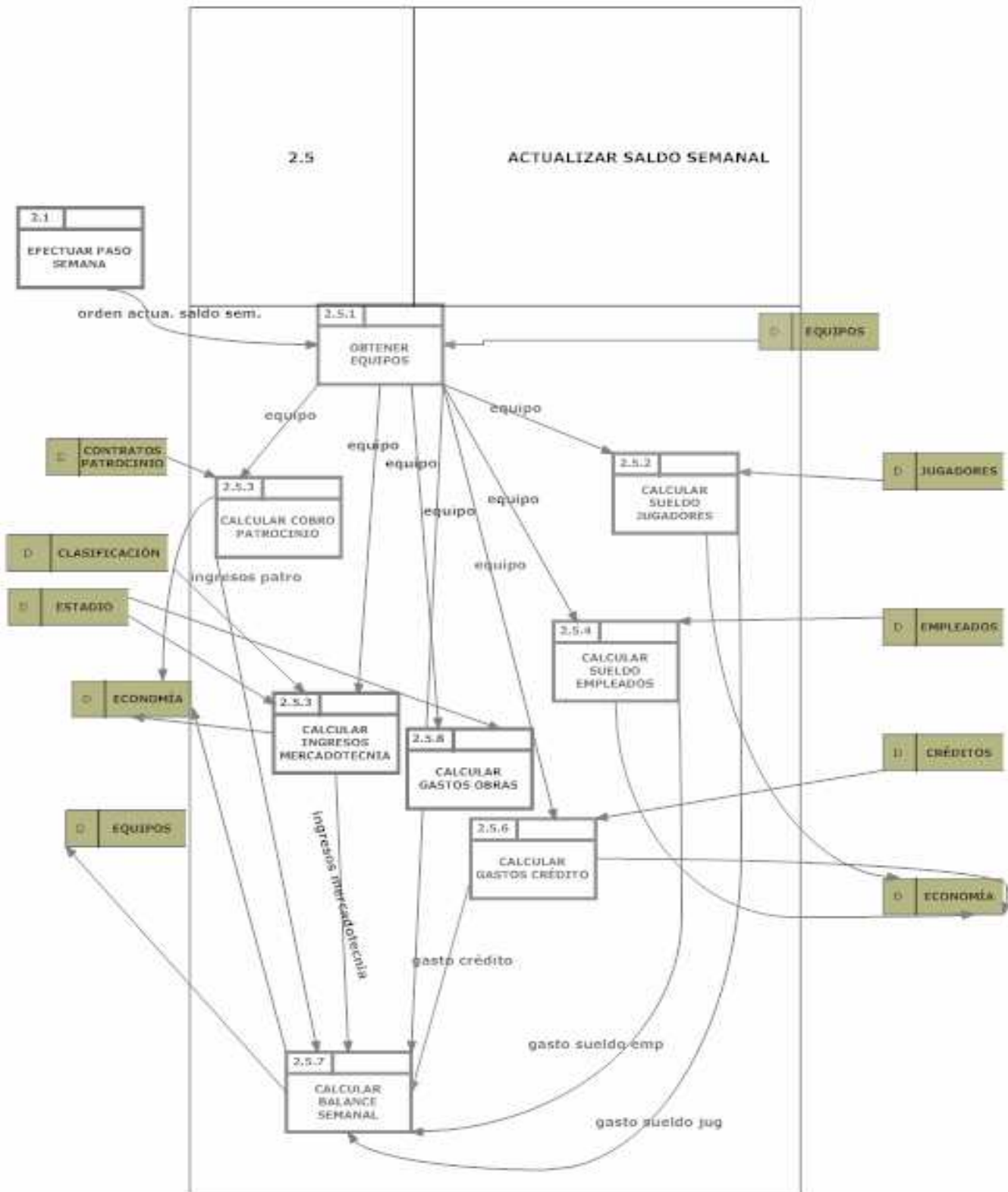


Fig 35. Diagrama de Flujos de Datos 19.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Obtener Equipos

#### Minispec or Description:

PRE1. Llega una orden de actualización semanal del saldo de cada equipo.

POST1. Se obtienen todos los equipos existentes en el sistema.

**Purpose:** Lee todos los equipos existentes en el juego.

#### Input flows

coming from EQUIPOS [Data Store]  
orden actualización saldo  
coming from EFECTUAR PASO SEMANA[Process]

#### Output flows

Equipo  
going to CALCULAR COBRO PATROCINIO[Process], CALCULAR SUELDO JUGADORES[Process], CALCULAR SUELDO EMPLEADOS[Process], CALCULAR GASTOS CREDITO[Process], CALCULAR BALANCE SEMANAL[Process], CALCULAR INGRESOS MERCADOTECNIA[Process]

### Process: Calcular Cobro Patrocinio

#### Minispec or Description:

PRE1. Llega el identificador de un equipo para el que se debe calcular la cantidad a cobrar en concepto de patrocinio.

POST1. Se obtienen los datos del último partido jugado por el equipo y de su contrato de patrocinio y se calcula qué cantidad tiene que cobrar ese equipo en concepto de patrocinio, actualizando ECONOMÍA.

**Purpose:** Calcula la cantidad que debe cobrar cada equipo en concepto de patrocinio cada semana(simulada).

#### Input flows

equipo  
coming from OBTENER EQUIPOS [Process]  
  
coming from PARTIDOS[Data Store]  
  
coming from CONTRATOS PATROCINIO[Data Store]

#### Output flows

ingresos patro  
going to CALCULAR BALANCE SEMANAL[Process]  
  
going to ECONOMÍA[Data Store]

### Process: Calcular Gastos Credito

#### Minispec or Description:

PRE1. Llega el identificador de un equipo para el que se debe calcular la cantidad a pagar en concepto de devolución de préstamo.

POST1. Se leen los datos del crédito vigente ese equipo y se calcula qué cantidad tiene que pagar ese equipo esa semana en concepto de devolución de préstamo, actualizando ECONOMÍA.

**Purpose:** Calcula la cantidad que debe pagar cada equipo en concepto de devolución de préstamo cada semana(simulada).

#### Input flows

equipo  
coming from OBTENER EQUIPOS [Process]  
  
coming from CRÉDITOS[Data Store]

#### Output flows

gastos crédito  
going to CALCULAR BALANCE SEMANAL[Process]

going to ECONOMÍA[Data Store]

**Process: Calcular Sueldo Jugadores**

**Minispec or Description:**

PRE1. Llega el identificador de un equipo de un equipo para el que se debe calcular la cantidad a pagar en concepto de sueldo de jugadores.

POST1. Se lee el sueldo de cada jugador perteneciente al equipo y se calcula qué cantidad en sueldos de jugadores se gasta cada semana(simulada), actualizando ECONOMÍA.

**Purpose:** Calcula la cantidad que debe pagar cada equipo en concepto de sueldo de jugadores cada semana(simulada).

**Input flows**

equipo  
coming from OBTENER EQUIPOS [Process]

coming from JUGADORES[Data Store]

**Output flows**

gastos sueldo jug  
going to CALCULAR BALANCE SEMANAL[Process]

going to ECONOMÍA[Data Store]

**Process: Calcular Sueldo Empleados**

**Minispec or Description:**

PRE1. Llega el identificador de un equipo de un equipo para el que se debe calcular la cantidad a pagar en concepto de sueldo de empleados.

POST1. Se lee el sueldo de cada empleado perteneciente al equipo y se calcula qué cantidad en sueldos de empleados se gasta cada semana(simulada), actualizando ECONOMÍA.

**Purpose:** Calcula la cantidad que debe pagar cada equipo en concepto de sueldo de empleados cada semana(simulada).

**Input flows**

equipo  
coming from OBTENER EQUIPOS [Process]

coming from EMPLEADOS[Data Store]

**Output flows**

gastos sueldo emp  
going to CALCULAR BALANCE SEMANAL[Process]

going to ECONOMÍA[Data Store]

**Process: Calcular Ingresos Mercadotecnia**

**Minispec or Description:**

PRE1. Llega el identificador de un equipo para el que se debe calcular la cantidad a ingresar en concepto de mercadotecnia durante esa semana.

POST1. Se obtiene el nivel de la tienda del estadio del equipo, el puesto en la clasificación y se calcula el ingreso, actualizando ECONOMÍA.

**Purpose:** Calcula la cantidad que ingresa un equipo durante una semana por la venta de diversos objetos (mercadotecnia).

**Input flows**

equipo  
coming from OBTENER EQUIPOS [Process]

coming from ESTADIO[Data Store]

coming from CLASIFICACIÓN[Data Store]

**Output flows**

ingresos mercadotecnia  
going to CALCULAR BALANCE SEMANAL[Process]

going to ECONOMÍA[Data Store]



**Process: Calcular Gastos Obras**

**Minispec or Description:**

PRE1. Llega el identificador de un equipo para el que se debe calcular la cantidad a gastar en esa semana en concepto de obras en el estadio.

POST1. Se obtiene el coste de las obras del estadio y se actualiza ECONOMÍA.

**Purpose:** Calcula la cantidad que gasta un equipo durante una semana por las obras en su estadio.

**Input flows**

equipo  
coming from OBTENER EQUIPOS [Process]

coming from ESTADIO[Data Store]

**Output flows**

Gastos obras  
going to CALCULAR BALANCE SEMANAL[Process]

going to ECONOMÍA[Data Store]

**Process: Calcular Balance Semanal**

**Minispec or Description:**

PRE1. Llega el identificador de un equipo, los ingresos en concepto de patrocinio y gastos en sueldos de jugadores, empleados y devolución de préstamo que ha tenido dicho equipo durante la semana (simulada).

POST1. Se calcula el balance semanal del equipo y se actualiza EQUIPOS (el saldo de cada equipo) y ECONOMÍA.

**Purpose:** Calcula el balance económico semanal de cada equipo durante la semana (simulada).

**Input flows**

equipo  
coming from OBTENER EQUIPOS [Process]

ingresos patro  
coming from CALCULAR COBRO PATROCINIO[Process]

gastos sueldo emp  
coming from CALCULAR SUELDO EMPLEADOS[Process]

gastos sueldo jug  
coming from CALCULAR SUELDO JUGADORES[Process]

gastos crédito  
coming from CALCULAR GASTROS CRÉDITO[Process]

**Output flows**

going to EQUIPOS[Data Store]

going to ECONOMÍA[Data Store]

**DESCRIPCIÓN FLUJOS DE DATOS**

**Dataflow:** Orden actualiza saldo

**From-symbol:** Efectuar Paso Semana[Process]

**To-symbol:** Obtener Equipos[Process]

**Data within flow:**

**Dataflow:** Equipo

**From-symbol:** Obtener Equipos[Process]

**To-symbol:** Calcular Cobro Patrocinio [Process], Calcular Gastos Credito [Process], Calcular Sueldo Empleados [Process], Calcular Sueldo Jugadores [Process], Calcular Ingresos Mercadotecnia [Process]

**Data within flow:**

**Dataflow:** Ingresos Patro

**From-symbol:** Calcular Cobro Patrocinio [Process],

**To-symbol:** Calcular Balance Semanal [Process]  
**Data within flow:**

**Dataflow:** Gasto Crédito  
**From-symbol:** Calcular Gastos Credito[Process]  
**To-symbol:** Calcular Balance Semanal [Process]  
**Data within flow:**

**Dataflow:** Gasto Sueldo Jug  
**From-symbol:** Calcular Sueldo Jugadores[Process]  
**To-symbol:** Calcular Balance Semanal [Process]  
**Data within flow:**

**Dataflow:** Gasto Sueldo Emp  
**From-symbol:** Calcular Sueldo Empleados[Process]  
**To-symbol:** Calcular Balance Semanal [Process]  
**Data within flow:**

**Dataflow:** ingresos mercadotecnia  
**From-symbol:** Calcular Ingresos Mercadotecnia [Process]  
**To-symbol:** Calcular Balance Semanal [Process]  
**Data within flow:**

**Dataflow:** gastos obras  
**From-symbol:** Calcular Gastos Obras [Process]  
**To-symbol:** Calcular Balance Semanal [Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** EQUIPOS  
**Input flows**

coming from Calcular Balance Semanal[Process]

**Output flows**

going to Obtener Equipos[Process]

**Data within store:** {@id + saldo}

**Datastore:** JUGADORES  
**Output flows**

going to Calcular Sueldo Jugadores[Process]

**Data within store:** {@id +equipo+ sueldo}

**Datastore:** EMPLEADOS  
**Output flows**

going to Calcular Sueldo Empleados[Process]

**Data within store:** {@id +equipo+ sueldo}

**Datastore:** CREDITOS  
**Output flows**

going to Calcular Gastos Credito[Process]

**Data within store:** {@id + equipo+devolucion\_semanal}

**Datastore:** ECONOMIA  
**Input flows**

coming from Calcular Sueldo Jugadores[Process]

coming from Calcular Sueldo Empleados[Process]

coming from Calcular Gastos Credito[Process]

coming from Calcular Cobro Patrocinio[Process]

**Data within store:** {@id + equipo+ ingresos\_mercadotecnia+ gastos\_sueldosjug+ gastos\_sueldosemp+ gastos\_obras+ gastros\_credito+ ingresos\_patro}

**Datastore:** CONTRATOS PATROCINIO

**Output flows**

going to Calcular Cobro Patrocinio[Process]

**Data within store:** {@id + equipo+ pago\_fijo+ pago\_victoria}

**Datastore:** PARTIDOS

**Output flows**

going to Calcular Cobro Patrocinio[Process]

**Data within store:** {@id + equipo+ jornada+ Eq1+ Eq2+ resultado}

**Datastore:** ESTADIO

**Output flows**

going to Calcular Ingresos Mercadotecnia [Process]

**Data within store:** {@id + equipo+ tienda+CosteObras+SemanasObras}

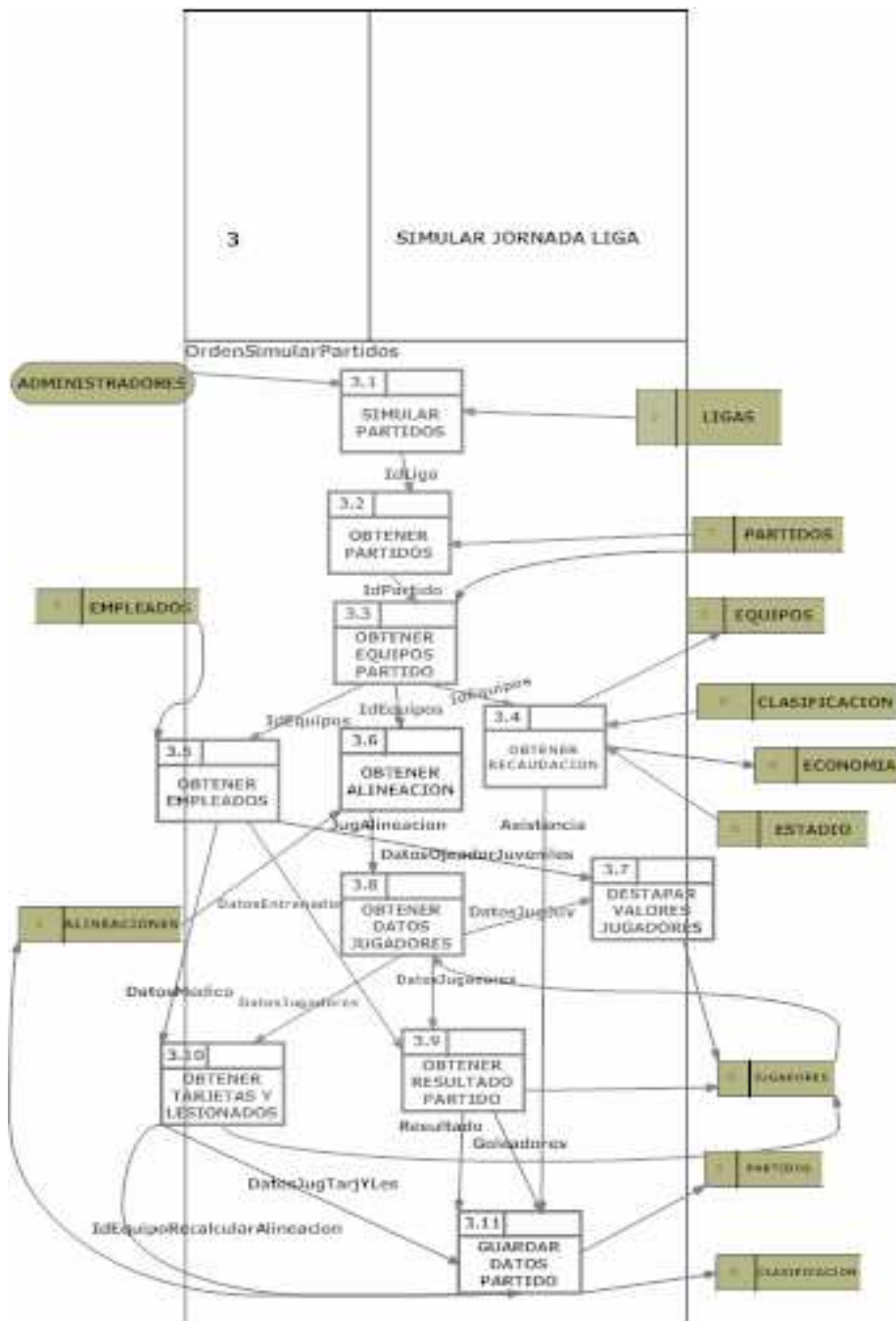


Fig 36. Diagrama de Flujos de Datos 20.

## ESPECIFICACIÓN PROCESOS PRIMITIVOS

### Process: Simular Partidos Liga

#### Minispec or Description:

PRE1. Llega una orden de simulación de partidos emitida por un usuario con privilegios de administrador.

POST1. Se simulan los partidos de una determinada jornada de liga.

**Purpose:** Simula los partidos de una jornada del campeonato.

#### Input flows

OrdenSimularPartidos

coming from ADMINISTRADOR [External]

coming from LIGAS [Data Store]  
**Output flows**  
IdLiga  
going to OBTENER PARTIDOS [Process]

**Process: Obtener Partidos**

**Minispec or Description:**

COMIENZA  
LEE IdLiga de SIMULAR PARTIDOS LIGA.  
NumJornada=número de jornada más bajo entre todos los partidos aún no disputados de la liga=Idliga.  
MIENTRAS exista un partido "no disputado" de la jornada=NumJornada en PARTIDOS  
LEE IdPartido de tipo "no disputado", liga=IdLiga y jornada=NumJornada  
DEVUELVE IdPartido  
FINMIENTRAS

TERMINA

**Purpose:** Obtiene los partidos que deben ser simulados

**Input flows**

IdLiga  
coming from SIMULAR PARTIDOS [Process]

coming from PARTIDOS [Data Store]

**Output flows**

IdPartido  
going to OBTENER EQUIPOS PARTIDO [Process]

**Process: Obtener Equipos Partido**

**Minispec or Description:**

PRE1. Llega un identificador del partido a simular.  
POST1. Se obtienen los identificadores de los equipos que forman parte del partido.

**Purpose:** Obtiene los equipos que disputan cada partido simulado.

**Input flows**

IdPartido  
coming from OBTENER PARTIDOS [Process]

coming from PARTIDOS [Data Store]

**Output flows**

IdEquipos  
going to OBTENER EMPLEADOS[Process], OBTENER JUGADORES[Process],  
OBTENER RECAUDACION[Process].

**Process: Obtener Empleados**

**Minispec or Description:**

PRE1. Llega un identificador de un equipo.  
POST1. Se obtienen los datos del médico y entrenador de ese equipo.

**Purpose:** Obtiene los empleados de un equipo implicado en la simulación de un partido.

**Input flows**

IdEquipos  
coming from OBTENER EQUIPOS PARTIDO [Process]

coming from EMPLEADOS [Data Store]

**Output flows**

DatosMedico  
going to OBTENER TARJETAS Y LESIONADOS[Process]  
DatosEntrenador  
going to OBTENER RESULTADO PARTIDO[Process]  
DatosOjeadorJuveniles

going to DESTAPAR VALORES JUGADORES[Process]

**Process: Obtener Alineacion**

**Minispec or Description:**

PRE1. Llega un identificador de un equipo.

POST1. Se obtienen los datos de la alineación en vigencia de ese equipo.

**Purpose:** Obtiene la alineación de un equipo implicado en la simulación de un partido.

**Input flows**

IdEquipos

coming from OBTENER EQUIPOS PARTIDO [Process]

coming from ALINEACIONES [Data Store]

**Output flows**

JugAlineación

going to OBTENER JUGADORES[Process]

**Process: Obtener Datos Jugadores**

**Minispec or Description:**

PRE1. Llegan los identificadores de una serie de jugadores que forman la alineación de un equipo.

POST1. Se obtienen los datos relativos a esos jugadores.

**Purpose:** Obtiene los datos de los jugadores de la alineación de un equipo implicado en la simulación de un partido.

**Input flows**

JugAlineación

coming from OBTENER ALINEACION [Process]

coming from JUGADORES[Data Store]

**Output flows**

DatosJugJuv

coming from DESTAPAR VALORES JUGADORES[Process]

DatosJugadores

going to OBTENER TARJETAS Y LESIONADOS [Process], OBTENER RESULTADO PARTIDO[Process]

**Process: Obtener Recaudacion**

**Minispec or Description:**

PRE1. Llegan los identificadores de los dos equipos que van a disputar un partido simulado.

POST1. Se lee la clasificación de cada equipo, el aforo del estadio del equipo local y el valor establecido de las entradas, datos a partir de los cuales se calcula la asistencia y recaudación.

**Purpose:** Obtiene la asistencia y recaudación de cada partido simulado.

**Input flows**

IdEquipos

coming from OBTENER EQUIPOS PARTIDO [Process]

coming from CLASIFICACION[Data Store]

coming from ESTADIOS[Data Store]

**Output flows**

going to EQUIPOS[Data Store]

going to ECONOMIA[Data Store]

Asistencia

going to GUARDAR DATOS PARTIDO[Process]

**Process: Obtener Tarjetas y Lesionados**

**Minispec or Description:**

PRE1. Llegan los datos de los jugadores y del médico de los equipos que van a disputar un partido simulado.

POST1. Se calculan las tarjetas amarillas, rojas y lesiones que sufre cada jugador en ese partido.

PRE2. Se obtiene un jugador sancionado (por acumulación de tarjetas amarillas o por haber visto una tarjeta roja) o lesionado..

POST2. Se genera una orden de cálculo de alineación.

**Purpose:** Obtiene la asistencia y recaudación de cada partido simulado y ordena rehacer la alineación de un equipo con jugador lesionado o sancionado.

**Input flows**

DatosJugadores  
coming from OBTENER JUGADORES[Process]

DatosMedico  
coming from OBTENER EMPLEADOS[Process]

**Output flows**

IdEquipoRecalcularAlineacion  
going to GUARDAR DATOS PARTIDO[Process]

DatosJugTarjYLes  
going to GUARDAR DATOS PARTIDO[Process]

going to JUGADORES[Data Store]

**Process: Obtener Resultado Partido**

**Minispec or Description:**

PRE1. Llegan los datos de los jugadores, alineación y entrenador de los equipos que van a disputar un partido simulado.

POST1. Se genera el resultado del partido y se asignan los goles marcados.

**Purpose:** Obtiene el resultado y los goleadores de cada partido.

**Input flows**

DatosEntrenador  
coming from OBTENER EMPLEADOS[Process]

DatosJugadores  
coming from OBTENER DATOS JUGADORES[Process]

**Output flows**

Resultado  
going to GUARDAR DATOS PARTIDO[Process]

Goleadores  
going to GUARDAR DATOS PARTIDO[Process]

going to JUGADORES[Data Store]

**Process: Destapar Valores Jugadores**

**Minispec or Description:**

PRE1. Llegan los datos de jugadores juveniles que van a disputar un partido y del ojeador de juveniles de los equipos que disputan el partido.

POST1. A partir de los datos del entrenador de juveniles se obtiene un número de atributos a destapar y se guardan los nuevos de jugadores juveniles en consecuencia.

**Purpose:** Destapa (hace visible al usuario) un cierto número de atributos de los jugadores juveniles.

**Input flows**

DatosJugJuv  
coming from OBTENER DATOS JUGADORES[Process]

DatosOjeadorJuveniles  
coming from OBTENER EMPLEADOS[Process]

**Output flows**

going to JUGADORES[Data Store]

**Process: Guardar Datos Partido**

**Minispec or Description:**

PRE1. Llegan los datos de jugadores juveniles que van a disputar un partido y del ojeador de juveniles de los equipos que disputan el partido.

POST1. A partir de los datos del entrenador de juveniles se obtiene un número de atributos a destapar y se guardan los nuevos de jugadores juveniles en consecuencia.

**Purpose:** Destapa (hace visible al usuario) un cierto número de atributos de los jugadores juveniles.

**Input flows**

Resultado  
coming from OBTENER RESULTADO PARTIDO[Process]  
Goleadores  
coming from OBTENER RESULTADO PARTIDO[Process]  
IdEquipoRecalcularAlineacion  
coming from OBTENER TARJETAS Y LESIONADOS[Process]  
DatosJugTarjYLes  
coming from OBTENER TARJETAS Y LESIONADOS[Process]  
Asistencia  
coming from OBTENER RECAUDACIÓN[Process]

**Output flows**

going to PARTIDOS[Data Store]  
going to CLASIFICACIÓN[Data Store]

**DESCRIPCIÓN FLUJOS DE DATOS**

**Dataflow:** OrdenSimularPartidos

**From-symbol:** ADMINISTRADORES [External]

**To-symbol:** Simular Partidos [Process]

**Data within flow:**

**Dataflow:** Id Liga

**From-symbol:** Simular Partidos [Process]

**To-symbol:** Obtener Partidos [Process]

**Data within flow:**

**Dataflow:** Id Partido

**From-symbol:** Obtener Partidos [Process]

**To-symbol:** Obtener Equipos Partido [Process]

**Data within flow:**

**Dataflow:** Id Equipos

**From-symbol:** Obtener Equipos Partido [Process]

**To-symbol:** Obtener Empleados [Process], Obtener Alineación [Process],  
Obtener Recaudación[Process]

**Data within flow:** [IdEquipo1+ IdEquipo2]

**Dataflow:** JugAlineación

**From-symbol:** Obtener Alineación [Process]

**To-symbol:** Obtener Datos Jugadores [Process]

**Data within flow:** [IdJug1+ IdJug2+ IdJug3+ IdJug4+ IdJug5+ IdJug6+ IdJug7+ IdJug8+ IdJug9+ IdJug10+  
IdJug11+ IdJug12+ IdJug13+ IdJug14+ IdJug15+ IdJug16]

**Dataflow:** Opciones Alineación

**From-symbol:** Obtener Alineación [Process]

**To-symbol:** Obtener Resultado Partido [Process]

**Data within flow:** [táctica+fuera\_juego]

**Dataflow:** Datos Jugadores



**From-symbol:** Obtener Datos Jugadores [Process]  
**To-symbol:** Obtener Resultado Partido [Process]  
**Data within flow:** [id+posición+estado\_fisico+moral+media]

**Dataflow:** Datos Jugadores Juveniles  
**From-symbol:** Obtener Datos Jugadores [Process]  
**To-symbol:** DestaparValoresJuveniles [Process]  
**Data within flow:** [id+Port\_tapado+Vel\_tapado+Agil\_tapado+Fuerza\_tapado+Tecnica\_tapado+Disp\_tapado+Reg\_tapado+Pase\_tapado+ViJue\_tapado+Res\_tapado+Def\_tapado+JueAe\_tapado+CapTac\_tapado+TalEsp\_tapado]

**Dataflow:** Datos Entrenador  
**From-symbol:** Obtener Empleados[Process]  
**To-symbol:** Obtener Resultado Partido [Process]  
**Data within flow:** [moral+capacidad]

**Dataflow:** Datos Medico  
**From-symbol:** Obtener Empleados[Process]  
**To-symbol:** Obtener Tarjetas Y Lesionados [Process]  
**Data within flow:** [id\_equipo+moral+capacidad]

**Dataflow:** Datos Ojeador Juveniles  
**From-symbol:** Obtener Empleados[Process]  
**To-symbol:** Destapar Valores Jugadores [Process]  
**Data within flow:** [moral+capacidad]

**Dataflow:** Asistencia  
**From-symbol:** Obtener Recaudación[Process]  
**To-symbol:** Guardar Datos Partido[Process]  
**Data within flow:**

**Dataflow:** Resultado  
**From-symbol:** Obtener Resultado Partido [Process]  
**To-symbol:** Guardar Datos Partido[Process]  
**Data within flow:**

**Dataflow:** Goleadores  
**From-symbol:** Obtener Resultado Partido [Process]  
**To-symbol:** Guardar Datos Partido[Process]  
**Data within flow:** {id+goles}

**Dataflow:** DatosJugTarjYLes  
**From-symbol:** Obtener Tarjetas Y Lesionados [Process]  
**To-symbol:** Guardar Datos Partido[Process]  
**Data within flow:** {id+tarj\_ama+tarj\_rojas+semanas\_lesion}

**Dataflow:** IdEquipoRecalcularAlineacion  
**From-symbol:** Obtener Tarjetas Y Lesionados [Process]  
**To-symbol:** Guardar Datos Partido[Process]  
**Data within flow:**

## ESTRUCTURA ALMACENES DE DATOS

**Datastore:** LIGAS  
**Output flows**

going to Simular Partidos [Process]  
**Data within store:** {@id+num\_equipos}

**Datastore:** PARTIDOS

### **Input flows**

coming from Guardar Datos Partido [Process]

### **Output flows**

going to Obtener Partidos [Process]

going to Obtener Equipos Partido [Process]

### **Data within store:**

{@id+liga+jornada+jugado+Eq1+Eq2+resultado+asistencia+goles\_eq1+goles\_eq2+tarj\_eq1+tarj\_eq2+les\_eq1+les\_eq2}

### **Datastore: EQUIPOS**

#### **Input flows**

coming from Obtener Recaudacion[Process]

**Data within store:** {@id + saldo}

### **Datastore: CLASIFICACIÓN**

#### **Input flows**

coming from GuardarDatosPartido[Process]

#### **Output flows**

going to ObtenerRecaudacion [Process]

**Data within store:** {@id+equipo+puntos+ posicion+partidos\_jugados+ partidos\_ganados+ partidos\_empatados+ partidos\_perdidos+goles\_fav+goles\_contra}

### **Datastore: ECONOMÍA**

#### **Input flows**

coming from Obtener Recaudación[Process]

**Data within store:** {@id+equipo+ semana+ingresos\_estadio}

### **Datastore: ESTADIO**

#### **Output flows**

going to Obtener Recaudacion [Process]

**Data within store:** {@id+equipo+capacidad+accesos+precio\_entradas}

### **Datastore: JUGADORES**

#### **Input flows**

coming from Destapar Valores Jugadores[Process]

coming from Obtener Resultado Partido[Process]

coming from Obtener Tarjetas Y Lesionados[Process]

#### **Output flows**

going to Obtener Jugadores [Process]

**Data within store:** {@id+equipo+juvenil+ posicion+ media+ estado\_fisico+ moral+ semanas\_lesionado+ port\_tap+ vel\_tap+ agil\_tap+ fuer\_tap+ tec\_tap+ dis\_tap+ reg\_tap+ pase\_tap+ vijue\_tap+ res\_tap+ def\_tap+ jueae\_tap+ captac\_tap+ talesp\_tap+ partidos\_jugados+ goles+ tarj\_ama+ tarj\_roja}

### **Datastore: EMPLEADOS**

#### **Output flows**

going to Obtener Empleados[Process]

**Data within store:** {@id+equipo+categoría+capacidad+motivacion+temp\_contrato}

**Datastore:** ALINEACIONES

**Output flows**

going to Obtener Alineación[Process]

**Input flows**

coming from Guardar Datos Partido[Process]

**Data within store:** {@id+ equipo+ formacion+ fuera\_juega+ juvenil+ jug1+ jug2+ jug3+ jug4+ jug5+ jug6+ jug7+ jug8+ jug9+ jug10+ jug11+ jug12+ jug13+ jug14+ jug15+ jug16}

## 10. Diseño del Modelo Físico de Datos

### Descripción de la tarea

*El objetivo de esta tarea es diseñar el Modelo Físico de Datos del sistema, determinando como convertir las entidades del Modelo Entidad-Relación ya construido en tablas, considerando las relaciones existentes entre las entidades y definiendo sus claves primarias y ajenas.*

### Desarrollo de la tarea

Se presentan las diferentes tablas de la base de datos del sistemas, incluyendo el nombre, tipo de dato, tamaño y breve descripción de cada campo. Se identificará la clave primaria de cada tabla anteponiendo el símbolo @ al nombre del campo que sea clave primaria y a cada posible clave ajena poniendo en cursiva el nombre del campo.

**Tabla: USUARIOS**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del usuario
<i>Equipo</i>	Int	11	Id del equipo manejado por el usuario
Usuario	Varchar	25	Nombre del usuario
Clave	Varchar	255	Clave del usuario

**Tabla: EQUIPOS**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del equipo
<i>Liga</i>	Int	11	Id de la liga a la que pertenece el equipo
Nombre	Varchar	50	Nombre del equipo
SaldoActual	Int	11	Saldo del equipo
Contrataque	Float		Nivel del equipo para la opción táctica Contrataque
Posesión	Float		Nivel del equipo para la opción táctica Posesión
MurallaDef	Float		Nivel del equipo para la opción táctica Muralla Defensivo
JuegBanEquipo	Float		Nivel del equipo para la opción táctica Juego Bandas
FueraJuego	Float		Nivel del equipo para la opción FueraJuego
BalonParado	Float		Nivel del equipo para la opción Balón Parado

**Tabla: LIGAS**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador de la liga
NumeroLiga	Int	11	Número de la liga
DivisionLiga	TinyInt	4	División de la Liga
NumEquipo	TinyInt	4	Número de equipos que componen la liga

**Tabla: ESTADIOS**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del estadio
Equipo	Int	11	Id del equipo al que pertenece el estadio
Nombre	Varchar	40	Nombre del estadio
Capacidad	Int	6	Capacidad del estadio
Accesos	TinyInt	4	Nivel de accesos del estadio
Tienda	TinyInt	4	Nivel de la tienda del estadio
EntradasLiga	TinyInt	4	Precio de las entradas para los partidos de liga disputados en ese estadio
EntradasCopa	TinyInt	4	Precio de las entradas para los partidos de copa disputados en ese estadio
EntradasAmistosos	TinyInt	4	Precio de las entradas para los partidos amistosos disputados en ese estadio
ObrasCapacidad	Int	6	Indica en cuantos asientos está siendo aumentada la capacidad del estadio por unas obras vigentes
ObrasAccesos	TinyInt	4	Indica si el nivel de accesos al estadio está siendo aumentado por unas obras vigentes
ObrasTienda	TinyInt	4	Indica si el nivel de la tienda del estadio está siendo aumentado por unas obras vigentes
CosteObras	Int	9	Indica el coste total de las obras vigentes de ampliación del estadio
SemanasObras	TinyInt	4	Indica el número de semanas que restan para finalizar las obras vigentes de ampliación del estadio

**Tabla: ENTRENAMIENTO**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del entrenamiento
Equipo	Int	11	Identificador del equipo al que se refieren los datos del entrenamiento
EstadoFisico	TinyInt	4	Puntos de entrenamiento asignados al parámetro "estado físico"
Porteria	TinyInt	4	Puntos de entrenamiento asignados al parámetro "portería"
Velocidad	TinyInt	4	Puntos de entrenamiento asignados al parámetro "velocidad"
Agilidad	TinyInt	4	Puntos de entrenamiento asignados al parámetro

Fuerza	TinyInt	4	"agilidad" Puntos de entrenamiento asignados al parámetro "fuerza"
Tecnica	TinyInt	4	Puntos de entrenamiento asignados al parámetro "técnica"
Disparo	TinyInt	4	Puntos de entrenamiento asignados al parámetro "disparo"
Regate	TinyInt	4	Puntos de entrenamiento asignados al parámetro "regate"
Pase	TinyInt	4	Puntos de entrenamiento asignados al parámetro "pase"
VisionJuego	TinyInt	4	Puntos de entrenamiento asignados al parámetro "visión juego"
Resistencia	TinyInt	4	Puntos de entrenamiento asignados al parámetro "resistencia"
Defensa	TinyInt	4	Puntos de entrenamiento asignados al parámetro "defensa"
JuegoAereo	TinyInt	4	Puntos de entrenamiento asignados al parámetro "juego aereo"
CapTactica	TinyInt	4	Puntos de entrenamiento asignados al parámetro "capacidad táctica"

**Tabla: ALINEACION**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador de la alineación
Equipo	Int	11	Id del equipo al que pertenece la alineacion
Juvenil	TinyInt	4	Indica si se trata de la alineación de un equipo normal o juvenil
Formacion	TinyInt	4	Formación de los jugadores de la alineación
Estrategia	TinyInt	4	Estrategia adoptada por el equipo
Defensa	TinyInt	4	Indica si el equipo usa defensa zonal o al hombre
FueraJuego	TinyInt	4	Indica si el equipo usa la opción del fuera de juego o no
Jug1	TinyInt	4	Id del primer jugador de la alineación
Jug2	TinyInt	4	Id del segundo jugador de la alineación
Jug3	TinyInt	4	Id del tercer jugador de la alineación
Jug4	TinyInt	4	Id del cuatro jugador de la alineación
Jug5	TinyInt	4	Id del quinto jugador de la alineación
Jug6	TinyInt	4	Id del sexto jugador de la alineación
Jug7	TinyInt	4	Id del séptimo jugador de la alineación
Jug8	TinyInt	4	Id del octavo jugador de la alineación

Jug9	TinyInt	4	Id del noveno jugador de la alineación
Jug10	TinyInt	4	Id del décimo jugador de la alineación
Jug11	TinyInt	4	Id del undécimo jugador de la alineación
Jug12	TinyInt	4	Id del duodécimo jugador de la alineación
Jug13	TinyInt	4	Id del decimotercer jugador de la alineación
Jug14	TinyInt	4	Id del decimocuarto jugador de la alineación
Jug15	TinyInt	4	Id del decimoquinto jugador de la alineación
Jug16	TinyInt	4	Id del decimosexto jugador de la alineación

**Tabla: JUGADORES**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del jugador
Equipo	Int	11	Id del equipo al que pertenece el jugador
Juvenil	TinyInt	4	Indica si se trata de un jugador normal o juvenil
Nombre	Varchar	25	Nombre del jugador
Apellidos	Varchar	75	Apellidos del jugador
Pais	Varchar	15	País de origen del jugador
FechaNac	Varchar	10	Fecha nacimiento del jugador
EdadRetirada	TinyInt	4	Edad en la que se retirará el jugador
Pie	Char	1	Pierna natural del jugador
Altura	TinyInt	4	Altura del jugador
TempContrato	TinyInt	4	Temporadas de contrato que le restan al jugador con su equipo
Valor	Int	9	Valor económico del jugador
Sueldo	Int	9	Sueldo del jugador
Posicion	Varchar	25	Posición del jugador
Dorsal	TinyInt	4	Número de dorsal del jugador
Media	Float		Refleja el nivel del jugador en su posición natural
PartJugados	TinyInt	4	Partidos jugados por el jugador
Goles	TinyInt	4	Goles marcados por el jugador
TarjAma	TinyInt	4	Tarjetas amarillas recibidas por el jugador
TarjRoja	TinyInt	4	Tarjetas rojas recibidas por el jugador
Transferible	Int	1	Indica si el jugador ha sido declarado transferible o no
TieneOfertas	Int	1	Indica si un jugador tiene ofertas de traspaso
EstadoFisico	TinyInt	4	Estado físico del jugador
Moral	TinyInt	4	Moral del jugador
Porteria	Float		Valor del jugador para el parámetro "porteria"
Velocidad	Float		Valor del jugador para el parámetro "velocidad"
Agilidad	Float		Valor del jugador para el parámetro "agilidad"
Fuerza	Float		Valor del jugador para el parámetro "fuerza"

Tecnica	Float		Valor del jugador para el parámetro "tecnica"
Disparo	Float		Valor del jugador para el parámetro "disparo"
Regate	Float		Valor del jugador para el parámetro "regate"
Pase	Float		Valor del jugador para el parámetro "pase"
ViJuego	Float		Valor del jugador para el parámetro "vision de juego"
Resistencia	Float		Valor del jugador para el parámetro "resistencia"
Defensa	Float		Valor del jugador para el parámetro "defensa"
JueAereo	Float		Valor del jugador para el parámetro "juego aereo"
CapTact	Float		Valor del jugador para el parámetro "capacidad tactica"
TalEsp	Varchar	25	Valor del jugador para el parámetro "talento especial"
PortPot	Float		Valor máximo(potencial) del jugador para el parámetro "porteria"
VelPot	Float		Valor máximo(potencial) del jugador para el parámetro "velocidad"
AgilPot	Float		Valor máximo(potencial) del jugador para el parámetro "agilidad"
FuerPot	Float		Valor máximo(potencial) del jugador para el parámetro "fuerza"
TecPot	Float		Valor máximo(potencial) del jugador para el parámetro "tecnica"
DispPot	Float		Valor máximo(potencial) del jugador para el parámetro "disparo"
RegPot	Float		Valor máximo(potencial) del jugador para el parámetro "regate"
PasePot	Float		Valor máximo(potencial) del jugador para el parámetro "pase"
ViJuePot	Float		Valor máximo(potencial) del jugador para el parámetro "vision de juego"
ResPot	Float		Valor máximo(potencial) del jugador para el parámetro "resistencia"
DefPot	Float		Valor máximo(potencial) del jugador para el parámetro "defensa"
JueAePot	Float		Valor máximo(potencial) del jugador para el parámetro "juego aereo"
CapTacPot	Float		Valor máximo(potencial) del jugador para el parámetro "capacidad tactica"
PortSemAnt	Float		Valor en la semana anterior del jugador del parámetro "porteria"
VelSemAnt	Float		Valor en la semana anterior del jugador del parámetro "velocidad"
AgilSemAnt	Float		Valor en la semana anterior

FuerSemAnt	Float		del jugador del parámetro "agilidad"
TecSemAnt	Float		Valor en la semana anterior del jugador del parámetro "fuerza"
DispSemAnt	Float		Valor en la semana anterior del jugador del parámetro "tecnica"
RegSemAnt	Float		Valor en la semana anterior del jugador del parámetro "disparo"
PaseSemAnt	Float		Valor en la semana anterior del jugador del parámetro "regate"
ViJueSemAnt	Float		Valor en la semana anterior del jugador del parámetro "pase"
ResSemAnt	Float		Valor en la semana anterior del jugador del parámetro "vision de juego"
DefSemAnt	Float		Valor en la semana anterior del jugador del parámetro "resistencia"
JueAeSemAnt	Float		Valor en la semana anterior del jugador del parámetro "defensa"
CapTacSemAnt	Float		Valor en la semana anterior del jugador del parámetro "juego aereo"
PorTapado	TinyInt	4	Valor en la semana anterior del jugador del parámetro "capacidad tactica"
VelTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "portería"
AgilTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "velocidad"
FuerTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "agilidad"
TecTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "fuerza"
DispTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "tecnica"
RegTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "disparo"
PaseTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "regate"
ViJueTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "pase"
ResTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "vision de juego"
DefTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "resistencia"
JueAeTapado	TinyInt	4	Indica si el jugador tiene visible(destapado) el valor del parámetro "defensa"



CapTacTapado	TinyInt	4	visible(destapado) el valor del parámetro "juego aereo" Indica si el jugador tiene visible(destapado) el valor del parámetro "capacidad tactica"
SemanasLesion	TinyInt	4	Indica si un jugador está lesionado y si lo está, en cuantas semanas se recuperará
Sancionado	TinyInt	4	Indica si un jugador está sancionado
Cara	TinyInt	4	Asigna al jugador un tipo de cara
Potencial	Float		Índice que indica el grado en que un jugador puede mejorar sus parámetros
Visibilidad	TinyInt	4	Índice usado para determinar si un jugador libre es visible a un equipo a o no

**Tabla: OFERTAS TRASPASO**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador de la oferta de traspaso
Jugador	Int	11	Id del jugador por el que se ha hecho la oferta
EquipoOfertante	Int	11	Id del equipo que ha realizado la oferta
EquipoReceptor	Int	11	Id del equipo que ha recibido la oferta
Cantidad	Int	11	Cantidad de dinero ofrecida
Contraoferta	Int	11	Cantidad de dinero solicitada por el equipo receptor de la oferta
Aceptada	TinyInt	4	Indica si la oferta ha sido aceptada
Rechazada	TinyInt	4	Indica si la oferta ha sido rechazada
IdEqUltOf	Int	11	Id del último equipo que ha hecho un movimiento (una oferta o contraoferta)

**Tabla: PARTIDOS**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del partido
Competicion	Int	11	Id de la liga a la que pertenece el partido
Equipo1	Int	11	Id del primer equipo que disputa el partido
Equipo2	Int	11	Id del segundo equipo que disputa el partido
Juvenil	TinyInt	4	Indica si se trata de un partido entre equipos normales o juveniles
Jornada	TinyInt	4	Indica si se trata de un partido entre equipos normales o juveniles
Asistencia	Int	11	Número de espectadores que asisten al partido
Goles1	TinyInt	4	Goles marcados por el equipo1

Goles2	TinyInt	4	Goles marcados por el equipo2
Jornada	TinyInt	4	Número de jornada a la que pertenece el partido
Goleadores1	Varchar	255	Id de los jugadores del primer equipo que han marcado gol durante el partido
Goleadores2	Varchar	255	Id de los jugadores del segundo equipo que han marcado gol durante el partido
TarjeteadosA1	Varchar	255	Id de los jugadores del primer equipo que han visto tarjeta amarilla durante el partido
TarjeteadosA2	Varchar	255	Id de los jugadores del segundo equipo que han visto tarjeta amarilla durante el partido
TarjeteadosR1	Varchar	255	Id de los jugadores del primer equipo que han visto tarjeta roja durante el partido
TarjeteadosR2	Varchar	255	Id de los jugadores del segundo equipo que han visto tarjeta roja durante el partido
Lesionados1	Varchar	255	Id de los jugadores del primer equipo que han sufrido una lesión durante el partido
Lesionados2	Varchar	255	Id de los jugadores del segundo equipo que han sufrido una lesión durante el partido
Jugadores1	Varchar	255	Id de los jugadores del equipo1 que han disputado el partido
Jugadores2	Varchar	255	Id de los jugadores del equipo2 que han disputado el partido
Minutos1	Varchar	255	Minutos disputados en el partido por cada jugador del equipo1
Minutos2	Varchar	255	Minutos disputados en el partido por cada jugador del equipo2
Jugado	TinyInt	4	Indica si el partido se ha disputado

**Tabla: CLASIFICACION**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador de la clasificación
Liga	Int	11	Id de la liga a la que se refiere la clasificación
Equipo	Int	11	Id del equipo al que se refieren los datos de la clasificación
Juvenil	TinyInt	4	Indica si se refiere a la versión normal o juvenil del equipo
Posicion	TinyInt	4	Posición en la tabla clasificatoria del equipo
Puntos	TinyInt	4	Puntos obtenidos por el

PartJug	TinyInt	4	equipo Partidos jugados por el equipo
PartGan	TinyInt	4	Partidos ganados por el equipo
PartEmp	TinyInt	4	Partidos empatados por el equipo
PartPerd	TinyInt	4	Partidos perdidos por el equipo
GolesFav	Int	11	Goles marcados por el equipo
GolesCon	Int	11	Goles encajados por el equipo

**Tabla: OFERTA PATROCINIO**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador de la oferta de patrocinio
Equipo	Int	11	Id del equipo que ha recibido la oferta de patrocinio
Nombre	Varchar	40	Nombre de la empresa que ha realizado la oferta
Contesta	TinyInt	4	Indica si la oferta ha sido contestada
Temporadas	TinyInt	4	Temporadas de patrocinio estipuladas en la oferta
PagoFijo	Int	9	Pago fijo semanal estipulado en la oferta
PagoVictoria	Int	9	Pago por partido ganado estipulado en la oferta

**Tabla: CONTRATO PATROCINIO**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del contrato de patrocinio
Patrocinador	Int	11	Id de la oferta de patrocinio que se ha convertido en contrato
Equipo	Int	11	Id del equipo que ha firmado el acuerdo de patrocinio
TempInicio	TinyInt	4	Temporada de inicio del acuerdo de patrocinio
TempFin	TinyInt	4	Temporada final del acuerdo de patrocinio

**Tabla: CREDITOS**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del crédito bancario
Equipo	Int	11	Id del equipo que ha recibido el crédito
ImporteRecib	Int	9	Cantidad de dinero prestada mediante el crédito
ImportDevolTot	Int	9	Cantidad que ha sido devuelta por el equipo
ImportDevolSem	Int	9	Cantidad semanal a devolver por el equipo
ImportPorDevol	Int	9	Cantidad que aún no ha sido devuelta por el equipo
SemanasFinCred	TinyInt	4	Semanas restantes hasta que se complete la

Tabla: PERSONAL

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del empleado
Equipo	Int	11	Id del equipo relacionado con el empleado
Capacidad	TinyInt	4	Capacidad del empleado
Motivacion	TinyInt	4	Motivación del empleado
Sueldo	Int	9	Sueldo del empleado
TempContrato	TinyInt	4	Temporadas de contrato que le restan al empleado
Nombre	Varchar	20	Nombre del empleado
Apellidos	Varchar	50	Apellidos del empleado
Pais	Varchar	20	País de origen del empleado
Categoria	Varchar	30	Categoría profesional del empleado

Tabla: ECONOMIA

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del balance económico
Equipo	Int	11	Id del equipo al que pertenecen los datos del balance económico
Semana	TinyInt	4	Semana a la que pertenece el balance
IngreMerca	Int	9	Cantidad ingresada por el equipo durante la semana en concepto de mercadotecnia
IngresosPatro	Int	9	Cantidad ingresada por el equipo durante la semana por el acuerdo de patrocinio
VentaJugadores	Int	11	Cantidad ingresada por el equipo durante la semana por venta de jugadores
VentaEntradas	Int	9	Cantidad ingresada por el equipo durante la semana por la venta de entradas
SueldoJug	Int	9	Cantidad gastada por el equipo durante la semana en sueldo de jugadores
SueldoEmp	Int	9	Cantidad gastada por el equipo durante la semana en sueldo de empleados
CompraJug	Int	11	Cantidad gastada por el equipo durante la semana en compra de jugadores
IngreVarios	Int	9	Cantidad ingresada por el equipo durante la semana por diversos conceptos
GastosVarios	Int	9	Cantidad gastada por el equipo durante la semana por diversos conceptos
Balance	Int	9	Balance del equipo

Tabla: DATOS\_TIEMPO

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador
Semana	Int	11	Número de semana vigente

TempActual	Int	11	Temporada vigente
------------	-----	----	-------------------

**Tabla: EVENTOS**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del evento
Equipo	Int	11	Id del equipo relacionado con el evento
Descrip	Varchar	255	Descripción del evento
Fecha	Varchar	10	Fecha en la que se produjo el evento

**Tabla: MENSAJES**

Nombre	Tipo de dato	Longitud	Descripción
@Id	Int	11	Identificador del mensaje
IdOrigen	Int	11	Id del equipo que manda el mensaje
IdDestino	Int	11	Id del equipo receptor del mensaje
Asunto	Varchar	150	Asunto del mensaje
Mens	Varchar	255	Cuerpo del mensaje
Fecha	Varchar	10	Fecha en la que se envió el mensaje

## 11. Definición de Interfaces de Usuario

### Descripción de la tarea

*El primer objetivo del capítulo es especificar los estándares, directrices y elementos generales a tener en cuenta en la definición de la interfaz de usuario. (Análisis de Sistemas de Información, Métrica 3: pág. 28).*

*Posteriormente se pretende especificar cada formato individual de la interfaz de pantalla desde un punto de vista estático. Teniendo en cuenta el Modelo de Procesos y la especificación de casos de uso, además de los principios determinados en la tarea anterior, se definirá cada interfaz de pantalla del sistema considerando aspectos interesantes para el posterior diseño y construcción tales como la posibilidad de cambio de tamaño o ubicación de elementos del interfaz; controles y elementos de diseño asociados, indicando cuáles aparecen inicialmente activos e inactivos al visualizar la interfaz de pantalla, etc. (Análisis de Sistemas de Información, Métrica 3: pág. 30).*

*Por último, se definirán los flujos entre los distintos formatos de interfaz de pantalla y dentro del propio forma o (Análisis de Sistemas de Información, Métrica 3: pág. 31).*

### Desarrollo de la tarea

#### Especificación de Principios Generales de la Interfaz

-El tipo de letra utilizado en las distintas pantallas de la aplicación será *arial*, con un tamaño acorde a cada situación (si se trata de un título, de un diálogo, etc.).

-Se señalarán los elementos “interactivos” de cada pantalla de modo que el usuario perciba claramente que puede realizar algún tipo de acción sobre ellos (aquellos elementos que no sean explícitamente interactivos como botones, barras de scroll, checkbox, etc.). Una manera de hacer visible al usuario esta interactividad es, por ejemplo, cambiando el color de fondo (o de la fuente de letra) del elemento cuando el puntero del ratón pase por dicho elemento.

-Todas aquellas acciones que vayan a suponer un cambio en el estado del sistema (es decir, un cambio en la base de datos del mismo) tendrán que ser ratificadas por el usuario para evitar descuidos y/o errores.

-No se emplearán botones predefinidos por el lenguaje HTML. Dado que el sistema es de carácter recreativo se buscará un diseño lo más atractivo posible al usuario (en lugar de botones se pueden usar imágenes con acciones asociadas cuando el usuario presione el botón del ratón sobre ellas).

## Especificación de Formatos Individuales de la Interfaz de Pantalla

En la parte de vista de nuestra aplicación están presentes los siguientes tipos de interfaces y objetos:

- Menús de opciones y navegación.
- Elementos de salto a otra pantalla/ventana.
- Listados de datos y valores.
- Formularios de entrada de datos.

### Menús de opciones y navegación.

Los menús de opciones se utilizan en todas las pantallas una vez que el usuario se ha conectado al sistema (siempre y cuando no se trate de un administrador) y su función es navegar entre las distintas pantallas. Se implementan mediante enlaces a las distintas páginas, cambiando el color del enlace a la página actual y especificando dicha página en la parte izquierda del menú; se definirán en un archivo independiente y se incluirán en todos aquellos que codifiquen pantallas que utilice el menú, para evitar la replicación de código.



**Fig. 37 Menú navegación entre pantallas**

Asímismo cuando el puntero del ratón pase por un enlace a una de las pantallas, además de cambiar la forma para indicar que se trata de un elemento sobre el que la pulsación del ratón desencadenaría alguna acción (esto ocurre en todos los elementos de todas las pantallas de la aplicación), se muestra la palabra completa, sin abreviar, para eliminar cualquier ambigüedad:



**Fig. 38 Menú navegación entre pantallas cuando el puntero del ratón está sobre enlace "Alin."**

Por otro lado hay que considerar la navegación entre distintas opciones de interfaz dentro de una misma pantalla. Para ello se incluyen pestañas (imágenes y texto con una acción asociada que hacen que cuando el usuario pulse sobre ellas aparezca el nuevo formato de la pantalla) lo más intuitivas posible:



**Fig.39 Pestañas de navegación entre opciones de pantalla.**

En el ejemplo de la pantalla Entrenamiento observamos que se trata de la opción de entrenamiento individual, y la pestaña correspondiente tiene el texto resaltado y un color verdoso, mientras que la pestaña de entrenamiento de equipo tiene un aspecto más difuminado, apagado.





Fig.40 Activación de otra pestaña de navegación en la pantalla Entrenamiento.

Estos elementos de navegación también están presentes en las pantallas Fichajes y Clasificación.



Fig.41 Navegación en la pantalla Fichajes.



Fig Navegación en la pantalla Clasificación.

En esta pantalla, en la pestaña activa en el ejemplo, cabe la posibilidad de anidar una navegación más: pulsando sobre alguno de los botones del panel derecho, se muestra una información u otra en el panel izquierdo:



Fig.41 Navegación en la pantalla Clasificación (tras pulsar sobre botón “Pichichi”).

### Elementos de salto a otra pantalla/ventana

Por estos nos elementos nos referimos a aquellos que provocan un salto a otra pantalla(o ventana con características especiales) cuando el usuario pulsa sobre ellos. Son elementos interactivos, por lo que al pasar el puntero del ratón sobre ellos se hará que éste tome forma de mano, y además se cambiará el color de todo el div para resaltar qué elemento está seleccionado.

Están presentes en todas las pantallas dónde puede aparecer un listado de jugadores(Plantilla, Juveniles, Alineación, Alineación juvenil, Fichajes, Negociaciones y Partido, Clasificación-Pestaña Pichichis), en los que se lanzará una ventana de ficha de jugador, y de partidos ya disputados(Clasificación-Pestaña Partidos anteriores), en los que se saltaría a la pantalla Periódico.

La ventana de ficha de jugador, aunque no es una pantalla propiamente dicha (siempre que se muestra una ficha de jugador se superpone en la pantalla en cuestión), se ha incluido en este apartado por su significación e importancia. El caso de la pantalla Plantilla(o Juveniles) es especial pues es la única pantalla que no carga un iframe para mostrar la ficha de jugador, ya que al tener que leerse los datos de los jugadores en esta pantalla se aprovechan y se muestran los que sean requeridos en la ficha del jugador sin tener que hacer una carga adicional de datos.

Nº	Est.	Mer.	Nombre	Pos.	Med	Nac	Edad	Po	Ve	Ag	Fu	Te	Di	Re	Pa	Vj	Rs	De	Ja	Ct	TalEs
2			E.Suriani		15		26	1	20	14	12	8	5	4	8	7	16	17	8	14	
3			F.Walker		14		26	2	20	14	11	8	8	7	7	3	16	16	8	12	

Fig.42 Jugador seleccionado en la pantalla Plantilla

#### Alineación

Nº	Nombre	Pos.	Med	Ef	Est.
1	P.Thompson		15	99	
2	H.Mendes		14	99	
3	F.Ribeiro		15	99	
4	A.Palma		14	99	
5	D.Kees		15	99	
6	J.Wood		14	99	
8	M.Tavares		15	99	
7	S.Luengo		15	99	

#### Tácticas y posicionam

Táctica: Ninguna

Formación: 4-4-2

Defensa: Zonal

Fuera de juego:



**Fig.43 Jugador seleccionado en la pantalla Alineación**

Se ha incluido un ejemplo de la pantalla Alineación porque cuenta con una peculiaridad: mientras que en todas las demás pantallas al poner el puntero del ratón sobre un jugador el usuario abre la ficha del jugador pulsado el botón primario del ratón, en Alineación lo hace apretando el secundario, pues el primario lo que hace es fijar el jugador para poder intercambiarlo con otros en la alineación.

Resultados anteriores					Clasificación	
Jornada	Compe.	Equipo Local	Equipo Visitante	Resul.		Equipo
1	Liga	Equipo9Div1Liga	Equipo2Div1Liga	1-0	1	fdwsd
2	Liga	Equipo2Div1Liga	Equipo7Div1Liga	0-0	2	Equipo7Div1L

**Fig.43 Partido seleccionado en la pantalla Clasificación-Partidos anteriores a partir del cual se puede saltar a la pantalla Periódico cargando los datos de dicho partido**

**Listado de datos y valores**

Las pantallas que muestran listados de datos los colocan en divs y según sean interactivos o no realizan acciones (cambio de color, de forma del puntero del ratón) cuando el puntero del ratón pasa sobre los datos.

Normalmente se presentan los datos al usuario en el mismo formato en el que están guardados en la base de datos, pero hay excepciones. Por ejemplo, cuando mostramos un listado de jugadores (en la pantalla Plantilla, Juveniles, Alineación, Alineación juvenil, Periódico, Clasificación, Fichajes o Negociaciones) los datos que muestran si el jugador está sancionado o no, lesionado o no, si es transferible, si tiene ofertas, su posición en el campo, su país de origen y su talento especial se representan mediante iconos(imágenes) predefinidos.

3		F.Ribeiro		15		27	2	18	16	13	10	8	3	8	3	20	16	9	14
4		A.Palma		14		25	2	12	8	17	5	6	4	10	3	10	20	16	17
5		D.Kees		15		25	3	13	10	15	7	6	4	9	5	11	22	18	17
6		J.Wood		14		27	3	10	10	14	12	9	3	22	14	17	11	11	18

**Fig.44 Conjunto de jugadores cargados en la pantalla plantilla.**

Si se trata de jugadores juveniles con algún parámetro oculto, el valor para dicho parámetro se representará con una "X", ya sea en la pantalla Juveniles o viendo la ficha de jugador de un jugador juvenil:

Nº	Est.	Mer.	Nombre	Pos.	Med	Nac	Edad	Po	Ve	Ag	Fu	Te	Di	Re	Pa	Vj	Rs	De	Ja	Ct	TalEs
1			T.Lula		X		18	19	5	13	X	1	1	1	5	2	4	2	1	6	

**Fig.45 Jugador con valor oculto para el parámetro fuerza.**

Hay interfaces en el sistema que además de tener como propósito listar un conjunto de datos(consulta por parte del usuario) sirven también para que el usuario guarde(modificación de la base de datos) una nueva configuración de esos datos:



**Fig.46 Interfaz de la pantalla Alineación**

Como observamos en la pantalla alineación se muestran por pantalla al usuario los jugadores que forman la alineación de su equipo, así como diversas opciones relativas a la táctica del mismo. Del mismo modo se da la opción de modificar cualquier aspecto de la alineación, pudiendo seleccionar un jugador con el botón izquierdo del ratón (el derecho se usa para abrir una ficha de jugador) e intercambiando su posición con otro jugador al pulsar de nuevo sobre ese otro jugador; por otro lado, y dado que para los valores de "Táctica", "Formación" y "Defensa" hay una serie de posibilidades definidas se usan *cajas de opciones* (combo box) para que el usuario pueda elegir la que desee; por último se usa una *caja de selección* (checkbox) para poner como activada o no la opción de fuera de juego. Una vez el usuario pulse sobre el botón de guardar cambios, y dado que va a producirse una modificación en la base de datos, el sistema mostrar una ventana de confirmación para completar la operación de guardado de configuración



**Fig.47 Ventana de confirmación de la pantalla Alineación**

La interfaz de la pantalla Entrenamiento posibilita al usuario modificar los puntos asignados a cada parámetro de diferentes formas. Pulsando sobre los botones de añadir y quitar puntos puede añadir/quitar un punto al parámetro (siempre que no se salga del rango válido 0-10 y no se sobrepase el total de puntos), y asimismo pulsando sobre la barra de la derecha puede elegir directamente cuantos puntos desea asignar (respetando asimismo los rangos).



**Fig.48 Interfaz pantalla Entrenamiento**

En la pantalla Economía se coloca un recuadro con los elementos necesarios para que el usuario pueda pedir un crédito para su equipo. Vemos que cuenta con un *input text* para que el usuario indique la cantidad que quiere recibir y las semanas para devolver el crédito; como siempre, al pulsar el botón de Pedir crédito el sistema abrirá una ventana de confirmación.

No obstante, si el equipo del usuario ya tiene un crédito vigente, el panel en esta pantalla tiene un aspecto distinto y no posibilita al usuario solicitar otro préstamo, sino que muestra los datos del crédito vigente y coloca un botón para cancelar (devolver) el préstamo



**Fig.48 Panel sin crédito vigente concedido**



**Fig.49 Panel si ya hay un crédito**

Hay pantallas que cargan datos pero no los muestran por completo, y muestran unos u otros según los botones que pulse el usuario en la interfaz:



**Fig.50 Pantalla de empleados**

Observando la interfaz de la pantalla empleados, vemos que en los dos paneles situados en la parte de arriba se muestran al usuario algunos datos de los empleados/candidatos de su equipo, y que abajo hay otros dos paneles en los que se colocan toda la información relativa al empleado/candidato seleccionado (mediante una pulsación de ratón). Además en esos paneles hay dos botones activos si y solo si hay algún elemento seleccionado para el panel correspondiente.

Por ultimo, en la pantalla de búsqueda de jugador (pantalla Fichajes, pestaña búsqueda) se cargan unos datos u otros en función de la búsqueda construida a partir de la configuración elegida por el usuario mediante diversos *combo box* e *input text*:



**Fig.51 Interfaz de la opción búsqueda de la pantalla Fichajes**

**-Formularios de entrada de datos.**

El sistema implementa en varias pantallas diálogos en los que el usuario mete datos y el sistema ofrece una respuesta (y ocurre una posible modificación en el estado de la base de datos). Estos formularios tienen el siguiente aspecto:

**Fig.52 Formulario para la renovación de un jugador**

Observamos que cuenta con dos *input text* para que el usuario coloque los datos que estime oportunos y botones para confirmar la oferta y para cerrar la ventana.

Existen también otro tipos de ventanas que provocan una entrada de datos en el sistema pero no exigen una entrada de datos a través de la interfaz de manera explícita; por ejemplo, si un usuario pulsa el botón vender situado en la ficha de un jugador se abrirá una ventana de confirmación de venta, que de confirmarse supone un cambio en el estado de la base de datos del sistema:

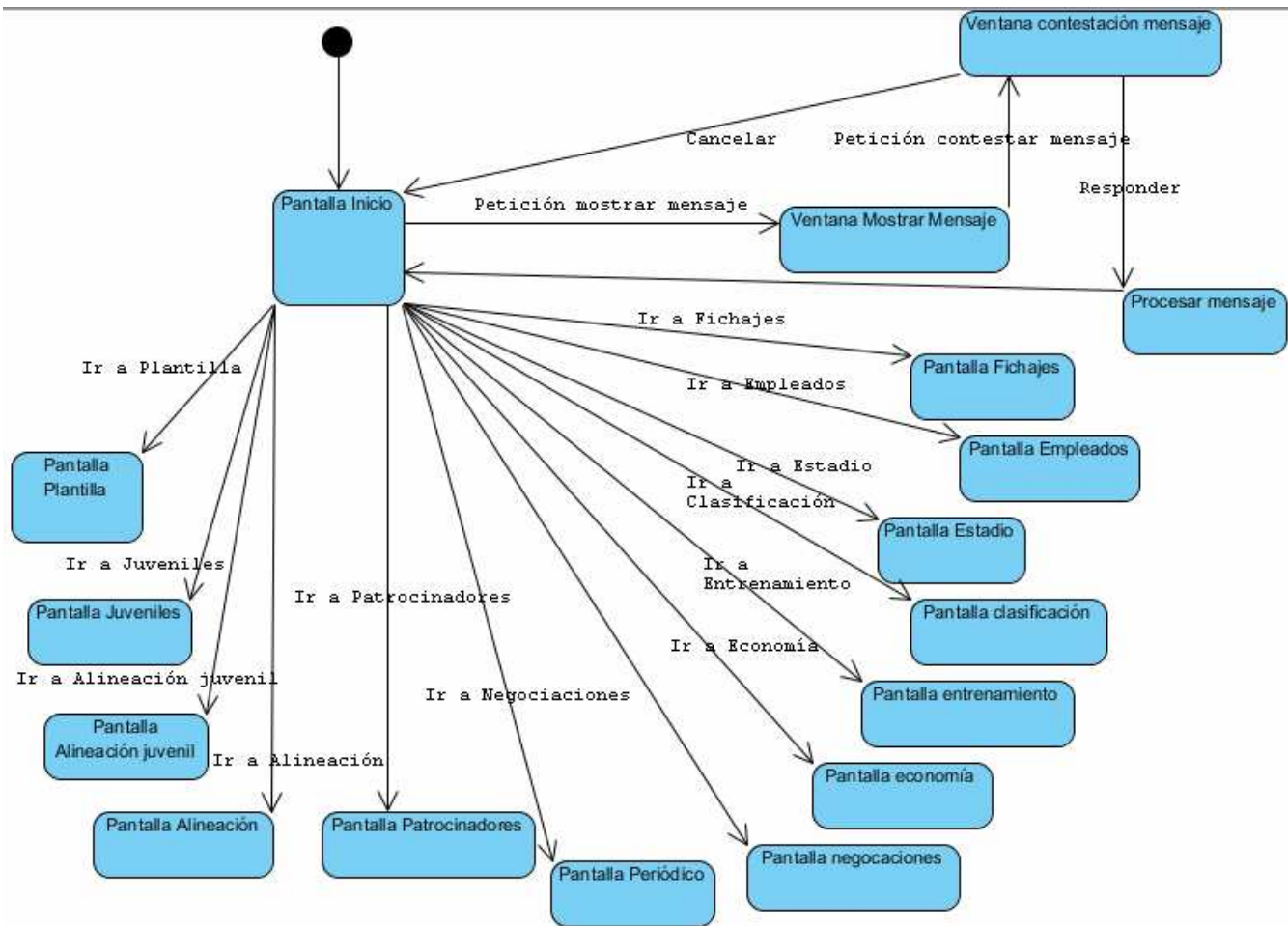
**Fig.53 Ventana de confirmación de venta de un jugador**

## **Especificación del comportamiento dinámico de la interfaz**

El objetivo de la tarea es definir los flujos entre los distintos formatos de interfaz de pantalla y dentro del propio formato. Se va a realizar un modelo de navegación de pantalla mediante un diagrama de transición de estados especificando mediante lenguaje textual el comportamiento de la interfaz para las pantallas de Inicio y Plantilla. El resto de pantallas siguen la misma lógica:

### **Pantalla Inicio**





**Fig.53 Diagrama Transición Estados: Comportamiento dinámico de interfaz**

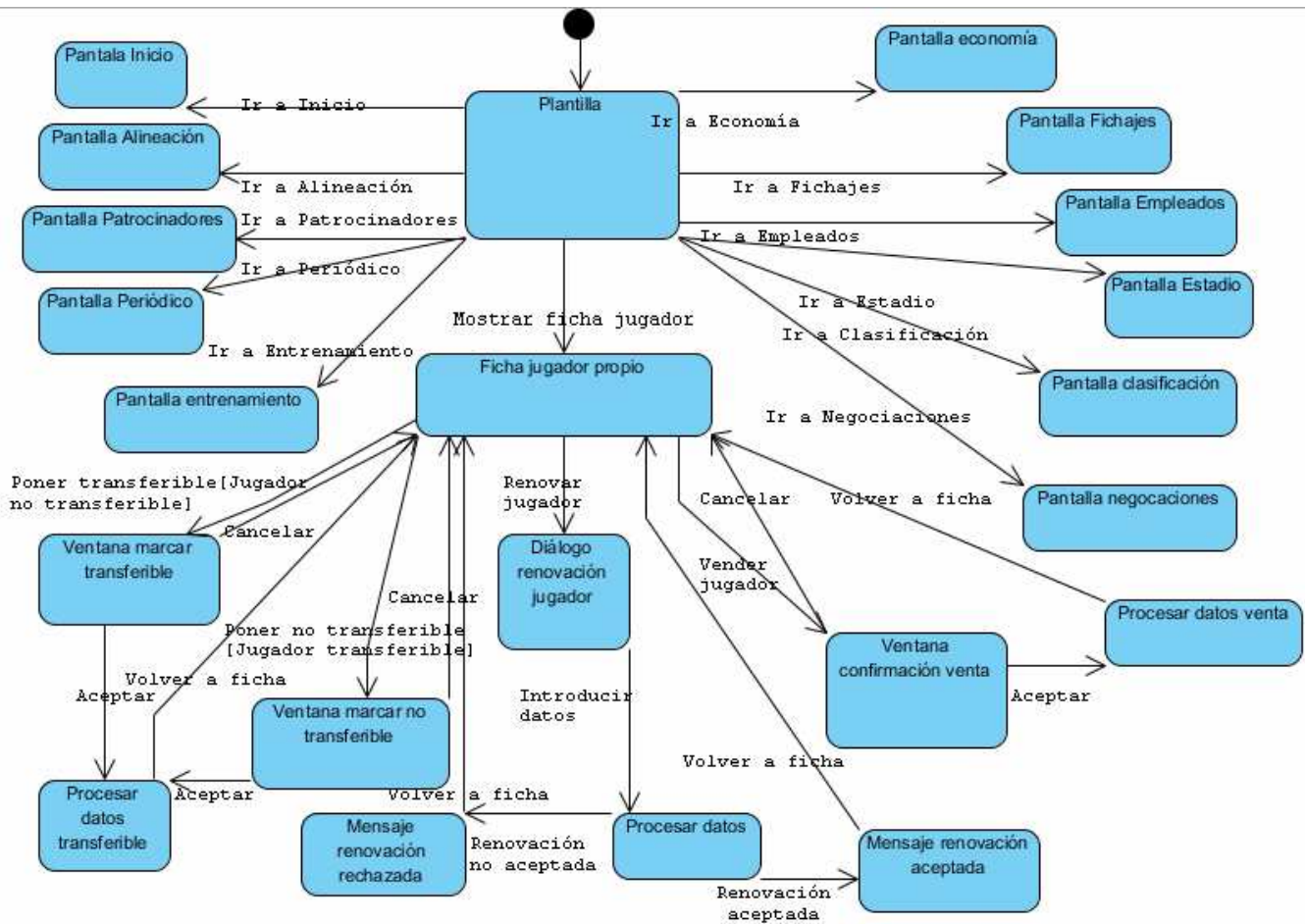
El estado inicial es la pantalla Inicio. Desde aquí se pueden cargar las pantallas de Plantilla, Juveniles, Alineación juvenil, Alineación, Patrocinadores, Periódico, Negociaciones, Economía, Entrenamiento, Clasificación, Estadio, Empleados y Fichajes. Esto ocurre cuando el usuario pulsa sobre el enlace a alguna de las pantallas (codificado con el evento "Ir a \*\*\*\*\*"), momento el cual se carga la pantalla correspondiente.

Por otro lado el usuario puede desear leer un mensaje y pulsar sobre el *div* que contiene algunos datos del mismo; este *div* tendría asociada a la pulsación del ratón sobre el mismo una acción consistente en mostrar una ventana con el texto del mensaje, para lo cual cargaría un *iframe* en el cual se mostrarían los datos del mensaje y dos botones con la posibilidad de contestar o de cerrar la ventana.

En este estado el usuario podría cerrar la ventana, y volver a la pantalla de Inicio, o pulsar sobre el botón de contestar, cuyo evento desencadenaría que se abriera una ventana con los elementos necesarios para hacerlo; una vez hecho (contestado), se mandarían los datos a un *iframe* oculto y ahí se procesarían (añadiendo el mensaje a la base de datos), volviendo entonces al estado inicial de la pantalla.

Hay que puntualizar que ninguna ventana ni en esta pantalla ni en ninguna otra del sistema son del tipo *modal*, es decir, del tipo que exigen al usuario que ejecute alguna acción sobre ellas, por lo que en cualquier momento el usuario puede pulsar sobre el enlace de alguna de las pantallas situadas en el menú principal y que el sistema las cargue.

### Pantalla Plantilla



**Fig.53 Diagrama Transición Estados: Comportamiento dinámico de interfaz 2**

En el Diagrama de Estados que codifica el comportamiento dinámico de la interfaz de la pantalla Plantilla se observa que, desde la pantalla Plantilla, solo hay un evento que puede influir en la interfaz de la misma (además de la pulsación en los enlaces del menú principal que llevan al sistema a cargar la correspondiente pantalla): la pulsación del ratón sobre el *div* que contiene información acerca de un jugador (acción codificada como “Mostrar ficha jugador”), que tendrá como consecuencia que se muestre una ventana con información sobre el jugador (“Ficha jugador propio”). En este caso el archivo con el código que se interpreta en el navegador de la máquina del usuario incluiría una acción asociada a la pulsación sobre un *div* de jugador consistente mostrar otro *div* con la ficha del jugador (en este caso no habría que llamar a un *iframe* que leyera datos de la base de datos pues la pantalla Plantilla carga inicialmente todos los datos, por lo que una llamada a un *iframe* supondría una carga superflua).

Sobre la ventana con la ficha del jugador hay varios eventos que pueden modificar el aspecto de la interfaz: la cancelación de la ventana (se volvería al estado inicial), el evento renovar jugador (asociado a un botón Renovar), el evento vender jugador (botón vender) y los eventos poner transferible y poner no transferible; como indica el diagrama son eventos que necesitan de una condición previa para poder ocurrir (que se trate de la ficha de un jugador no transferible para que ocurra el “Poner transferible” y al contrario para el suceso “Poner no transferible”).

Si ocurre el evento vender jugador se abre una ventana de confirmación de venta (un *div* definido en el mismo archivo); si se confirma la venta por parte del usuario, se carga un *iframe* invisible, se le pasan los datos de la venta, se procesan (se modifica el estado de la base de datos) y se vuelve el estado anterior (pantalla Plantilla con la ventana de la ficha del jugador abierta). Si por el contrario se cancela la venta, se vuelve al estado anterior igualmente.

La peculiaridad del estado “Diálogo renovación jugador” es que no se trata de una ventana de confirmación, sino de recogida de datos: el usuario introduce los datos (sueldo y temporadas) que desee y se envían a un *iframe* invisible que los procesa (decide si la oferta es aceptable), modifica la base de datos si procede y provoca la aparición de una ventana de respuesta para el usuario.

Por motivos de visibilidad no se han incluido algunas acciones de tipo “cancelar”, por lo que hay que reseñar que desde cualquier ventana (no desde cualquier estado, porque hay algunos que no suponen un cambio en la interfaz respecto al estado anterior, por ejemplo los iframes invisibles) puede ocurrir el evento cancelar (el usuario pulsa al botón que cancela la ventana) y volver al estado anterior; también cabe recordar que desde cualquier estado que suponga una ventana se puede ir cualquier pantalla del menú principal (pues no son ventanas modales y se pueden seguir pulsado los elementos de la pantalla Plantilla).

## 12. Definición de Niveles de Arquitectura

### Descripción de la tarea

*En esta tarea se describen los niveles de la arquitectura software, mediante la definición las principales particiones físicas del sistema de información, representadas como nodos y comunicaciones entre nodos.*

*Se entiende por nodo cada partición física o parte significativa del sistema de información, con características propias de ejecución o función, e incluso de diseño y construcción. (Diseño de Sistemas de Información, Métrica 3: pág. 9)*

### Desarrollo de la tarea

Se obtiene un diagrama de despliegue dónde se plasman las particiones físicas del sistema y las comunicaciones entre ellas. Posteriormente se explica brevemente la lógica del sistema desde el punto de vista del diagrama obtenido:

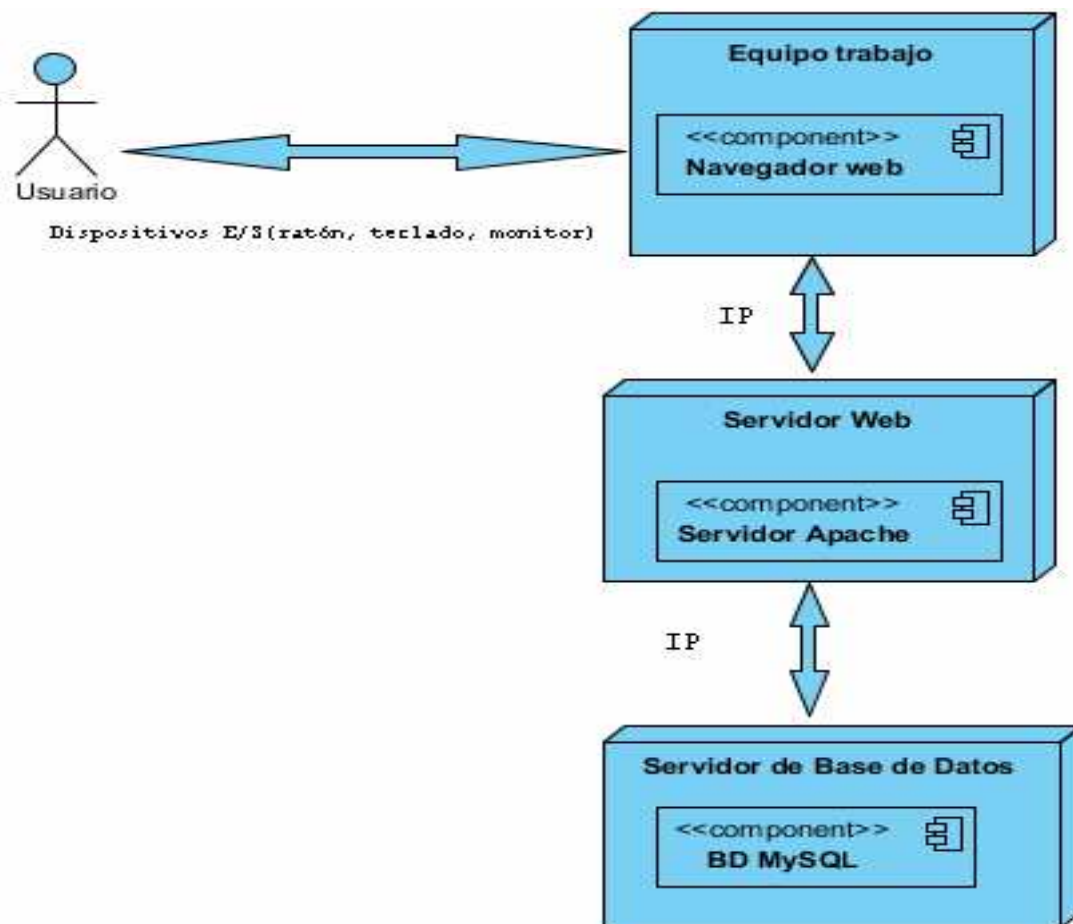


Fig.54 Diagrama Despliegue: Niveles de Arquitectura del Sistema



El sistema está construido mediante archivos codificados con lenguaje HTML, Javascript y PHP. Los archivos que conforman el sistema están alojados en el nodo **servidor web**, y cuando un usuario, a través su máquina(**equipo de trabajo**) quiere entrar al sistema lo que ocurre es que el navegador de su terminal pide al servidor web (a través de una comunicación **IP**) la página en concreto; en el servidor se ejecuta el archivo determinado (el lenguaje PHP del mismo se ejecuta en el servidor) y se manda el archivo con lenguaje HTML y Javascript al navegador del cliente para que este lo interprete y lo muestre al usuario por un dispositivo de entrada/salida (**monitor**).

Por último, hay que mencionar también que los archivos de la aplicación acceden a una base de datos MySQL, incluida en este diagrama en el nodo **Servidor Base de Datos**. Aunque en el diagrama se han separado en nodos diferentes el servidor Apache y el servidor MySQL, nada impide que ambos componentes estén ubicados en el mismo nodo.

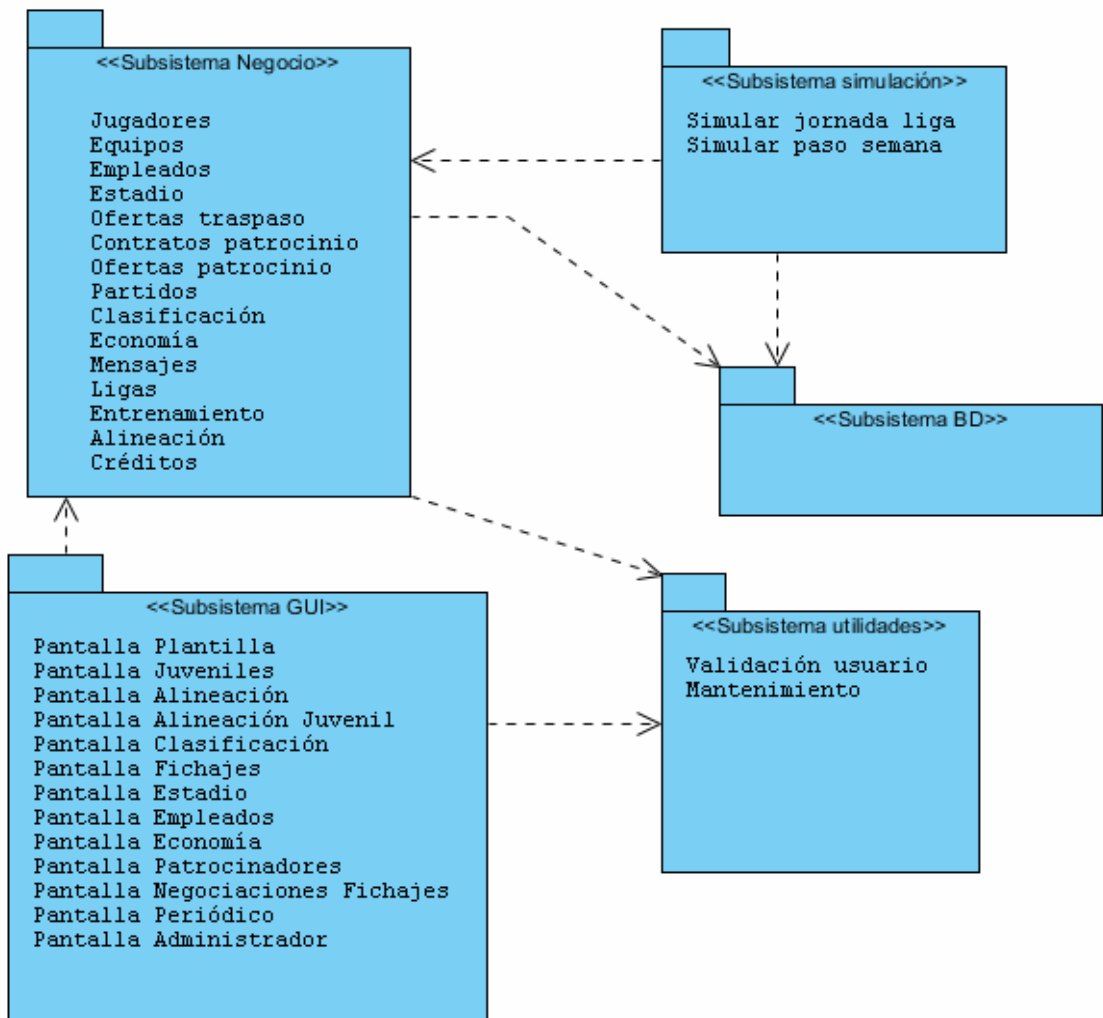
## 13. Identificación de subsistemas de diseño

### Descripción de la tarea

*En esta capítulo vamos a identificar los subsistemas de diseño atendiendo a criterios como facilidad de mantenimiento y reutilización de recursos, además de clasificarlos como subsistemas específicos (aquellos que implementan funcionalidades propias del sistema) o de soporte (servicios comunes o que proporcionan un acceso transparente a los recursos).*

### Desarrollo de la tarea

Se obtiene un diagrama de paquetes con los diferentes subsistemas y las relaciones entre ellos, añadiendo una breve descripción de cada uno de ellos:



**Fig.54 Diagrama de Paquetes: Subsistemas de diseño del Sistema**

Observamos que se han obtenido cinco subsistemas y que se han especificado los elementos que componen cada uno de ellos.

### **Subsistema Negocio**

Este subsistema es de tipo específico y está formado por todos los objetos del sistema y las reglas de negocio que rigen a los mismos.

### **Subsistema GUI**

Este subsistema es de tipo específico y corresponde a todos los elementos que conforman la interfaz gráfica del sistema con el usuario. Como elementos del subsistema se han nombrado a todas las pantallas de las que dispondrá el sistema, pero añadiendo más nivel de detalle se podría haber añadido todas las ventanas, cuadros de diálogo etc. relacionados con cada pantalla.

### **Subsistema Simulación**

Este subsistema es de tipo específico y su funcionalidad es implementar las funciones que simulan el comportamiento del sistema en un entorno real, es decir, si en lugar de un prototipo fuera un sistema final.

### **Subsistema BD**

Este subsistema es de tipo específico y contiene todos elementos necesarios relacionados con el acceso, modificación y borrado de datos del sistema.

## Subsistema Utilidades

Este subsistema es de soporte y esta formado por dos elementos, *validación de usuario*, que comprueba los datos de identificación del usuario y proporciona al sistema información necesaria en cada momento (tipo de usuario (normal o administrador) y equipo asignado en caso de ser un usuario normal) y *mantenimiento*, respuesta del sistema cuando un usuario intenta acceder al mismo mientras el sistema se encuentra en tareas de mantenimiento.

# 14. Especificación de Excepciones

## Descripción de la tarea

*El objetivo de esta tarea es la definición de los comportamientos no habituales en el sistema, que reflejan situaciones anómalas o secundarias en el funcionamiento y ejecución del sistema de información.*

*Las excepciones se describen incluyendo los siguientes conceptos:*

*-Tipo y descripción de la excepción.*

*-Condiciones previas del sistema.*

*-Elemento afectado*

*-Respuesta del sistema.*

*-Elemento asociado a la respuesta esperada del sistema.*(Diseño de Sistema de Información, Métrica

3:pág. 11)

## Desarrollo de la tarea

Se van a definir las posibles situaciones anómalas del sistema y la respuesta que éste ofrece en esos casos:

Tipo	Descripción	Condiciones previas	Elemento afectado	Respuesta del sistema	Elemento asociado a la respuesta
<b>Valor obligatorio</b>	El usuario, en la pantalla de login, deja en blanco el valor para algún campo del recuadro "Pedir Equipo"	El usuario está en la pantalla de login.	Proceso de asignar equipo.	Se le exige al usuario que rellene todos los campos.	Ventana de aviso.
<b>Rango no válido-Valor obligatorio-Formato incorrecto</b>	El usuario, en una ventana de diálogo para renovar a un jugador de su equipo, introduce un valor para el campo "sueldo" o "temporadas" fuera del rango permitido(1-100000 para "sueldo" y 1-5 para "temporadas") o con formato incorrecto.	El usuario ha abierto una ventana de renovación de un jugador de su equipo.	Proceso de renovar un jugador.	Se le da un formato válido al valor introducido: si es menor que el valor mínimo del rango se le da el valor mínimo del rango, si es mayor que el máximo se le asigna el máximo, y si contiene algún carácter no numérico se elimina.	Función javascript que modifica el valor introducido.
<b>Rango no válido-Valor obligatorio-Formato incorrecto</b>	El usuario, al pedir un crédito, introduce un valor para el campo "cantidad" o "semanas" fuera del rango	El sistema se encuentra en la pantalla Economía.	Proceso de pedir un crédito.	Se le da un formato válido al valor introducido: si es menor que el valor mínimo del rango se le da el valor mínimo del	Función javascript que modifica el valor introducido.

	permitido(1-150000 para "cantidad" y 1-16 para "semanas") o con formato incorrecto.			rango, si es mayor que el máximo se le asigna el máximo, y si contiene algún carácter no numérico se elimina.	
<b>Rango no válido-Valor obligatorio-Formato incorrecto</b>	El usuario, en una ventana de diálogo para renovar a un empleado de su equipo, introduce un valor para el campo "sueldo" o "temporadas" fuera del rango permitido(1-50000 para "sueldo" y 1-5 para "temporadas") o con formato incorrecto.	El sistema se encuentra en la pantalla Empleados.	Proceso de renovar un empleado.	Se le da un formato válido al valor introducido: si es menor que el valor mínimo del rango se le da el valor mínimo del rango, si es mayor que el máximo se le asigna el máximo, y si contiene algún carácter no numérico se elimina.	Función javascript que modifica el valor introducido.
<b>Rango no válido-Valor obligatorio-Formato incorrecto</b>	El usuario, en una ventana para hacer una oferta a un jugador de otro equipo, introduce un valor con formato incorrecto (con algún símbolo no numérico) o fuera del rango(que irá entre 1 y el saldo del equipo del usuario en ese momento).	El usuario ha abierto la ficha de un jugador de otro equipo,se encuentra en la pantalla Negociaciones o en la pantalla Fichajes.	Proceso de ofertar por un jugador.	Se le da un formato válido al valor introducido: si es menor que el valor mínimo del rango se le da el valor mínimo del rango, si es mayor que el máximo se le asigna el máximo, y si contiene algún carácter no numérico se elimina.	Función javascript que modifica el valor introducido.
<b>Rango no válido-Valor obligatorio-Formato incorrecto</b>	El usuario, en los campos para determinar el precio de las entradas de los partidos disputados en su estadio, introduce un valor fuera del rango válido(1-50) o con formato incorrecto.	El sistema se encuentra en la pantalla Estadio.	Proceso de	Se le da un formato válido al valor introducido: si es menor que el valor mínimo del rango se le da el valor mínimo del rango, si es mayor que el máximo se le asigna el máximo, y si contiene algún carácter no numérico se elimina.	Función javascript que modifica el valor introducido.
<b>Saldo insuficiente</b>	El usuario acepta una oferta de traspaso por uno de sus jugadores y el equipo ofertante no tiene suficiente saldo(si lo tenía en el momento de hacer la oferta, pero no cuando se acepta)	El usuario está viendo la ficha de un jugador de su equipo o esta en la pantalla Negociaciones.	Proceso de traspaso de un jugador.	Se le avisa al usuario de que no ha sido posible completar la operación porque el equipo ofertante no tiene suficiente saldo.	Ventana de aviso.
<b>Saldo</b>	El usuario quiere	El sistema se	Proceso de	Se le avisa al	Ventana de aviso.

<b>insuficiente</b>	hacer obras en el estadio de su equipo pero el coste sobrepasa el saldo del equipo.	encuentra en la pantalla Estadio y no existen obras en curso.	realizar obras en estadio.	usuario de que no ha sido posible completar la operación porque el equipo no tiene suficiente saldo.	
<b>Saldo insuficiente</b>	El usuario quiere rescindir un contrato de patrocinio y el coste de la indemnización sobrepasa el saldo del equipo.	El sistema se encuentra en la pantalla Patrocinadores y el equipo del usuario ya tiene suscrito un acuerdo de patrocinio.	Proceso de rescisión de contrato de patrocinio.	Se le avisa al usuario de que no ha sido posible completar la operación porque el equipo no tiene suficiente saldo.	Ventana de aviso.
<b>Saldo insuficiente</b>	El usuario quiere rescindir el contrato de un empleado de su equipo o contratar a un candidato cuya categoría comparte un empleado del equipo y el coste de la indemnización sobrepasa el saldo del equipo.	El sistema se encuentra en la pantalla Empleados.	Proceso de rescisión de contrato de un empleado.	Se le avisa al usuario de que no ha sido posible completar la operación porque el equipo no tiene suficiente saldo.	Ventana de aviso.
<b>Saldo insuficiente</b>	El usuario quiere devolver todo el importe de un crédito pedido por su equipo pero éste sobrepasa al saldo del equipo.	El sistema se encuentra en la pantalla Economía y el equipo del usuario tiene un crédito un vigencia.	Proceso de devolución de un crédito.	Se le avisa al usuario de que no ha sido posible completar la operación porque el equipo no tiene suficiente saldo.	Ventana de aviso.
<b>Jugadores insuficientes</b>	El usuario quiere vender un jugador de su equipo al sistema pero su plantilla tiene 16 jugadores.	El usuario está viendo la ficha de un jugador de su equipo.	Proceso de venta de un jugador.	Se le avisa al usuario de que no ha podido realizarse la operación porque su equipos se quedaría con menos jugadores de los necesarios.	Ventana de aviso.
<b>Jugadores insuficientes</b>	El usuario quiere aceptar una oferta de traspaso por un jugador de su equipo pero éste tiene 16 jugadores.	El usuario está viendo la ficha de un jugador de su equipo o esta en la pantalla Negociaciones.	Proceso de venta de un jugador.	Se le avisa al usuario de que no ha podido realizarse la operación porque su equipos se quedaría con menos jugadores de los necesarios.	Ventana de aviso.
<b>Jugadores insuficientes</b>	El usuario quiere dejar libre a un jugador juvenil pero la plantilla de su equipo juvenil tiene 25 jugadores.	El usuario está en la pantalla Juveniles.	Proceso de poner libre a un jugador juvenil.	Se le avisa al usuario de que no ha podido realizarse la operación porque su equipos se quedaría con menos jugadores juveniles de los necesarios.	Ventana de aviso.
<b>Demasiados jugadores</b>	El usuario quiere fichar un jugador del sistema pero	El usuario está en la pantalla Fichajes.	Proceso de compra de un jugador.	Se le avisa al usuario de que no ha podido	Ventana de aviso.

<b>Demasiados jugadores</b>	la plantilla de su equipo tiene 25 jugadores.			realizarse la operación porque su equipo tiene el máximo número de jugadores permitido.	
	El usuario acepta una oferta por un jugador de su equipo pero el equipo ofertante tiene 25 jugadores.	El usuario está viendo la ficha de un jugador de su equipo o esta en la pantalla Negociaciones.	Proceso de venta de un jugador.	Se le avisa al usuario de que no ha podido realizarse la operación porque el equipo ofertante tiene el máximo número de jugadores permitido.	Ventana de aviso.
<b>Demasiados jugadores</b>	El jugador quiere subir un jugador del equipo juvenil al normal pero la plantilla tiene 25 jugadores.	El usuario está viendo la ficha de un jugador juvenil de su equipo.	Proceso de subir un jugador juvenil.	Se le avisa al usuario de que no ha podido realizarse la operación porque su equipo tiene el máximo número de jugadores permitido.	Ventana de aviso.
<b>Cambio no permitido</b>	El jugador pone en el once inicial a un jugador lesionado o sancionado	El usuario está en la pantalla Alineación.	Proceso de guardar alineación.	Se le avisa al usuario que no puede colocar en el once titular a un jugador lesionado o sancionado y se anula el cambio.	Ventana de aviso.
<b>Mantenimiento</b>	El sistema está ejecutando los scripts de simulación de una jornada o del paso de una semana y no se quiere que se modifique el estado de la base de datos mientras estén en ejecución.	El sistema se encuentra en cualquier pantalla.	Todos.	Se muestra una pantalla avisando que el sistema está realizando tareas de mantenimiento y que próximamente volverá a estar operativo.	Variable global a todos los archivos que cuando está activada lanza una pantalla con el mensaje de mantenimiento.

# 15. Especificación del Entorno de Construcción

## Descripción de la tarea

*El objetivo de la tarea es la definición detallada y completa del entorno necesario para la construcción de los componentes del sistema. Se propone que la especificación se realiza en base a estos conceptos:*

- Entorno tecnológico: hardware, software y comunicaciones.
- Herramientas de construcción, generadores de código, etc.
- Restricciones técnicas del entorno.
- Requisitos de operación y seguridad del entorno de construcción. (Diseño de Sistema de Información, Métrica 3:pág. 45)

## Desarrollo de la tarea

- **Entorno tecnológico**

Se dispondrán de dos terminales para el desarrollo del sistema; en uno se desarrollará todo el código y diversos componentes que formarán el sistema, y en otro se mantendrán un servidor de base de datos(MySQL) y un servidor web(Apache) para ir probando y siguiendo el desarrollo del sistema.

Ambos equipos de trabajo se comunicarán mediante una red local. Se trata de dos máquinas con un procesador Intel Core 2 y con Sistema Operativo Windows XP, pero el sistema podría haberse desarrollado en máquinas diferentes sin ningún problema o restricción.

- **Herramientas de construcción**

*UltraEdit-32:* es el editor de texto empleado para la generación de los archivos con código PHP, HTML y JavaScript. Se trata de un programa sin funcionalidades para la ayuda en la generación y depuración del código pero elegido debido a la experiencia del desarrollador en su uso.

*Fireworks/PaintShopPro:* *Fireworks* es un programa de diseño gráfico empleado en el diseño de las pantallas de la aplicación web; *PaintShopPro* es un programa de retoque de imágenes. La metodología a seguir en la construcción de las distintas pantallas es la siguiente: con *Fireworks* se diseña el aspecto que va a tener la pantalla y con *PaintShopPro* se van desmenuzando y preparando los distintos componentes gráficos que tendrá cada pantalla.

*PHPMyAdmin:* es una herramienta empleada para crear y gestionar bases de datos MySQL a través de un interfaz web.

- **Seguridad**

Se realizan copias de seguridad periódicas de todos los archivos que se van generando y se guardan en más de un equipo (el usado para el desarrollo y el usado para mantener los servidores, por ejemplo). Incluso se aconseja guardarlos en equipos externos (fuera de la red local) o en dispositivos externos de almacenamiento para garantizar la seguridad.

# 16. Elaboración de Especificaciones de Construcción

## Descripción de la tarea

*El objetivo de la tarea es realizar una especificación detallada de cada componente, en pseudocódigo o lenguaje natural, completando la información que se considere necesaria según el entorno tecnológico. (Diseño de Sistema de Información, Métrica 3:pág. 47)*

## Desarrollo de la tarea

El sistema va a estar compuesto por diversos archivos .php. Cada pantalla estará implementada en un archivo propio (en el que se usará HTML y Javascript para la parte de interfaz gráfica con el usuario y PHP para el acceso a datos), y para algunas se necesitará implementar iframes en archivos distintos para el tratamiento de datos de la BD; además se crearán diversos archivos complementarios.

Se pasa a describir mediante lenguaje natural cada componente del sistema:

-*BD\_Funciones.php*: Se definen mediante código PHP varias funciones para la ejecución de sentencias SQL. El objetivo es poder usar estas funciones a lo largo de todo el desarrollo y ganar en comodidad y facilidad de uso.

-*BD\_OpManager.php*: Se define mediante código PHP una clase con varios métodos para gestionar (añadir, quitar o modificar) la BD. El objetivo es definir funciones de BD que se vayan a utilizar en diversos archivos del sistema, es decir, evitar la replicación de código. Por ello también se incluyen diversas funciones que van a ser utilizadas en distintos componentes del sistema.

-*NoCache.php*: Se incluye en todos los archivos del sistema que el navegador cargará directamente (archivos de pantallas e iframes llamados desde las pantallas). El objetivo del mismo es evitar que se cargue el archivo desde el caché del navegador.

-*Vacio.php*: Se trata de un archivo "artificial", sin código. Se crea porque cuando se define un iframe en HTML es necesario definir a que archivo se va a llamar cuando se cargue el iframe, y en ocasiones no se llama a ninguno; por ello se utiliza siempre este archivo.

-*Login.php*: Codifica todo lo necesario para representar la pantalla de entrada al sistema. Mediante código HTML se especifica la apariencia de la misma y mediante código Javascript su componente dinámico, las acciones que se desencadenan a partir de diversas interacciones del usuario con los componentes del interfaz.

También se puede volver a cargar este componente desde el archivo de validación de los datos introducidos por el usuario en esta pantalla, por lo que se definen tres variables PHP para recoger los datos enviados desde ese otro componente y regular algunos elementos de la interfaz según el valor de estas variables (se muestra o no una ventana que avisa de que no hay ningún equipo asignado a un usuario con el nombre introducido, por ejemplo).

-*Login2.php*: Se trata de un archivo únicamente compuesto por código PHP que recibe los datos introducidos por el usuario en el formulario de conexión al sistema del archivo *Login.php* y manda al navegador a la página de destino.

Se reciben los datos, se lee de la base de datos y se comprueba si el nombre y clave introducido corresponden al de un usuario con un equipo asignado, si hay algún error en el clave o nombre o si se trata de un usuario con privilegios de administrador, y se redirige al navegador a la pantalla de Login de nuevo (si no hay ningún equipo asignado para los datos introducidos), a la de Administrador o a la de Inicio.

Si se redirige a Inicio (es decir, si los datos son los de un usuario con un equipo asignado) se utiliza una función de PHP que inicia una sesión y que coloca en dos variables el nombre del usuario y el identificador del equipo mientras dure la sesión (es decir, mientras no se cierre el navegador).

-*Administrador.php*: En este archivo se incluye el código HTML necesario para mostrarle al administrador la pantalla desde la cual puede ejecutar sus operaciones; así, a dos imágenes se les asocia una función Javascript que se ejecuta cuando el usuario pulsa el botón del ratón con el puntero del mismo encima de las imágenes y carga en un iframe invisible (no hay modificación sobre lo que ve el usuario en la pantalla) un fichero .php que realiza la función correspondiente.

-*SimularJornadDivis.php*: Este archivo se carga desde *Administrador.php* y contiene el código necesario para simular una jornada de la competición. En él se lee de la base de datos que partidos se deben simular y ejecuta una función (definida en otro archivo incluido en este) que realiza todos los procesos implicados en la simulación de un partido.

-*ScriptResolPart.php*: Este archivo, incluido en *SimularJornadDivis.php*, incluye el código que realiza todos los procesos implicados en la simulación de un partido. Tiene una función principal, la de simulación del



partido, que calcula (tomando en cuenta diversos datos que lee de la BD) la asistencia al estadio del equipo donde se disputa el partido y actualiza el saldo y el registro economía (modificando la BD) del equipo local; posteriormente obtiene de la BD (mediante un método de la clase definida en el archivo `_BD_OpManager.php`) la alineación de los equipos y las características de sus jugadores; pasa a entonces a calcular mediante diversas funciones el resultado del partido así como qué jugadores meten un gol, ven un partido o se lesionan, además de destapar valores tapados si se trata de un partido entre equipos juveniles, y finaliza actualizando el estado de la BD con todo lo obtenido (resultado del partido, clasificación según el resultado, goles, tarjetas y lesiones de los jugadores, etc).

-*SimularPasoSemana.php*: Este archivo se carga desde `Administrador.php` y contiene el código necesario para simular el paso de una semana en el sistema. En él se obtienen de la BD los identificadores de todos los equipos del sistema y para cada uno se lee de la BD el patrocinador que tenga, y se actualiza en la BD los campos del saldo del equipo y el registro de economía para la semana correspondiente; también se obtienen de la BD los datos del crédito (si lo hubiera) del equipo, y se actualiza el saldo del equipo, el registro de la tabla ECONOMÍA y todos los campos necesarios de la tabla CRÉDITOS.

Este fichero incluye un archivo con las funciones necesarias para simular el efecto del entrenamiento sobre los jugadores, `FuncionesEntren.php` y `FuncionesCreacJugadores.php`; así, para cada equipo se obtienen los datos de sus jugadores y de su entrenador y mediante funciones definidas en esos archivos se simula el efecto del entrenamiento.

También se actualiza el número de ofertas de patrocinio creando una nueva si hubiera menos de 4 para ese equipo en ese momento, y se crea un nuevo jugador para el equipo juvenil si fuera necesario (por ello se incluye un archivo, `FuncionesCreacJugadores.php`, con las funciones pertinentes).

Por último, se crea un nuevo registro en la tabla de ECONOMÍA para la nueva semana, se actualizan los campos del jugadores influidos por el paso de una semana (`SemanasLesion` y `Sancionados`) y si el estadio del equipo tiene obras en curso se actualiza el número de semanas hasta su finalización y se descuenta del saldo del equipo el coste semanal por obras (además de actualiza el registro de economía).

- *FuncionesEntren.php*: Este archivo, llamado desde `SimularPasoSemana.php`, contiene las funciones necesarias para simular el entrenamiento (de jugadores y de equipo). Incluye funciones (definidas en `BDFunciones.php`) que modifican el estado de la BD (los campos de jugadores y equipo afectados por el entrenamiento).

-*FuncionesCreacJugadores*: Este archivo contiene funciones para crear jugadores, según la posición y el nivel del jugador. Son funciones en PHP que devuelven resultados pero que no modifican el estado de la BD, será en aquellos archivos que usen estas funciones donde se hagan las modificaciones de BD. Otro aspecto relevante es que mientras que en cada nuevo jugador algunos de sus datos serán creados aleatoriamente (como su nombre, apellidos o país, aunque los dos primeros estén condicionados por el tercero), aquellos que determinan el nivel del jugador tendrán un patrón. El objetivo es crear funciones que permitan crear equipos (en la carga inicial de datos del sistema) razonablemente equilibrados en lo que se refiere al nivel de sus jugadores.

-*NombresJugadores.php*: Es un mero conjunto de datos. Las funciones que crean jugadores (y empleados) necesitan generar aleatoriamente un nombre y apellidos, y las distintas posibilidades se almacenan en este componente. Se clasifican los nombres y apellidos según nacionalidad, cubriendo todas las nacionalidades posibles que pueden tener los jugadores y empleados del sistema.

-*PedirEquipo.php*: Se trata de un archivo únicamente compuesto por código PHP que recibe los datos introducidos por el usuario en el formulario de conexión al sistema del archivo `Login.php` y manda al navegador a la página de destino.

Se reciben los datos y se comprueban que todos los datos recibidos contengan cadenas distintas de la vacía; si no es así, se manda al navegar a la página de Login de nuevo, y si están todos los campos rellenos se lee de la BD para comprobar si hay algún equipo libre. Si hay alguno se le asigna (modificando la BD) al nuevo usuario y se manda al navegador a la página de Inicio, y si no lo hay se le manda a la página de Login de nuevo (mandando un parámetro que indica que se debe mostrar al usuario que no se pudo completar la operación de petición de equipo).

-*DameDatosLogin.php*: es un archivo incluido en aquellos archivos que necesitan saber el identificador del equipo del usuario que está conectado en el sistema en esa sesión. Es decir, cuando se carga una página, por ejemplo la de Plantilla, se necesita saber cual es el equipo del usuario para poder mostrarle por pantalla los jugadores de su equipo; ese valor, el del equipo del usuario, se guarda en una variable incluida en el archivo

`_DameDatosLogin.php`, que además redirige al navegador a la pantalla de Login si no hay ningún usuario conectado.

Se trabaja con el concepto de sesión modelado en PHP; se inicia la sesión con la instrucción `session_start`, y lo que guardamos en la variable `$_SESSION` es accesible en cualquier momento (y desde cualquier archivo) de la sesión actual, conservando la sesión actual hasta que se cierre el navegador.

*-Inicio.php*: es un archivo que contiene todo lo relacionado con la pantalla Inicio. Tiene una parte gráfica, codificada con lenguaje HTML, una parte en JavaScript que modela el comportamiento dinámico de la interfaz y una parte de acceso a datos mediante lenguaje PHP.

En este archivo se accede a la capa de datos para obtener los mensajes y eventos relacionados con el equipo del usuario (que, recordemos, es "conocido" en el archivo porque éste incluye `_DameDatosLogin.php`), además de diversos datos que desean mostrar por pantalla al usuario.

Además, si el usuario pulsa el botón del ratón sobre un mensaje se abre una ventana con dicho mensaje a través de la carga de un `iframe`, y si pulsa sobre un botón de responder en esa nueva ventana se carga un nuevo `iframe` con forma de ventana de respuesta de mensaje.

*-IFrame\_VerMensUsua.php*: es un `iframe` cargado desde la página de Inicio que lee y muestra por pantalla al usuario los datos sobre un mensaje (el mensaje cuyo identificador se recibe en el `iframe`).

Tiene una parte de acceso a datos, para leer de la BD los datos del mensaje, y una parte de interfaz (con código HTML, que aunque no se ha mencionado puede generarse a partir de código PHP. El navegador sólo recibe e interpreta código HTML, pero este puede estar escrito tal cual en los archivos o ser generado por código PHP).

*-IFrame\_RespMensUsua.php*: es un `iframe` cargado desde la página de Inicio (en realidad, desde el `iframe IFrame_VerMensUsua.php` cargado en la página Inicio) que recibe el identificador de un mensaje y lee sus datos de la BD, para posteriormente mostrar por pantalla al usuario una ventana para poder contestar dicho mensaje.

*-IFrame\_ProcEnviaMensAUsua.php*: es un `iframe` que recoge los datos enviados en un formulario desde `Iframe_EnviaMensAUsua.php` o `Iframe_RespMensUsua.php` y que llama a un método de la clase de gestión de la BD para insertar el nuevo mensaje. Posteriormente carga una función en el archivo padre que muestra una ventana informando que se ha enviado el mensaje.

*-Plantilla.php*: es un archivo que contiene todo el código relacionado con la pantalla Plantilla. Tiene una parte de acceso a datos en la que se leen de la BD todos los datos de los jugadores del equipo del usuario (cuyo identificador es conocido al incluir `_DameDatosLogin.php`), otra de representación gráfica (HTML) y otra de lógica de negocio, mediante código Javascript, que gobierna el comportamiento de los distintos elementos de la interfaz.

Para cada jugador de la plantilla se crea un `div` en el que se muestran algunos datos del jugador, y pinchando con el ratón sobre él se abre una ventana con toda la información del mismo (no es necesario llamar a un `iframe` porque todos los datos ya se han obtenido en la carga inicial de datos).

A partir de aquí, se usan distintos `iframes` para cubrir todas las posibles acciones que un jugador puede realizar sobre un jugador: ponerlo transferible, ponerlo no transferible, renovarlo, venderlo al sistema, aceptar una oferta de otro equipo, rechazar una oferta y negociar una oferta.

*-Juveniles.php*: es un archivo que contiene todo el código relacionado con la pantalla Juveniles. Es un archivo similar a `Plantilla.php` con la salvedad de que los datos se obtienen de los jugadores juveniles del equipo. Además, cuando se muestra la ficha de un jugador, el usuario tiene otras opciones: subir al jugador al primer equipo (botón activo sólo si el jugador tiene todos sus parámetros destapados) o darle la carta de libertad. Para ejecutar estos procesos se cargan sendos `iframes`.

*-IFrame\_PlantillaCartaLib.php*: es un `iframe` cargado desde `Juveniles.php` que recibe el identificador de un jugador y lo pone libre (sin equipo), para finalmente recargar la página desde la que se cargó el `iframe`. No hace la comprobación de que la plantilla juvenil se quede con más de 16 jugadores pues ésta se hace en `Juveniles.php`.

- *IFrame\_JuvSubirPrimEquipo.php*: es un iframe cargado desde Juveniles.php que recibe el identificador del jugador a subir al primer equipo. Para comprobar si se va a respetar el número máximo de jugadores se accede al identificador del equipo del usuario, se obtienen el número de jugadores normales que tiene el equipo, y en caso de ser menor de 25, se completa la operación y se recarga la página padre. Si se viola la restricción, se invoca una función de Juveniles.php que muestra una ventana al usuario avisándolo de la incidencia.

- *IFrame\_MultiusosPonerTrans.php*: Es un iframe cargado desde el archivo Plantilla.php y desde un iframe que carga la ficha de un jugador (de ahí viene el nombre *multiusos*, porque se llama desde distintos archivos). Su función es únicamente modificar el estado de la base de datos (poniendo un jugador como transferible o como no transferible, dependiendo del valor de un parámetro que recibe), no tiene por tanto elementos de interfaz.

- *IFrame\_MultiusosRenoJugad.php*: Como el iframe anterior, a éste se le llama de Plantilla.php y desde un iframe que muestra la ficha de un jugador. En este caso, aunque su función también es modificar el estado de la BD, necesita acceder primero a ella para leer datos sobre el jugador cuyo identificador recibe el iframe en forma de parámetro.

Con esos datos (edad, media y posición del jugador) y los datos sobre la oferta de renovación que se reciben como parámetros se decide si es una oferta aceptable; en caso afirmativo se modifica la BD y en caso contrario únicamente se avisa al usuario (mediante un función definida en el “archivo padre”, es decir, el archivo que carga el iframe).

- *IFrame\_MultiusosVendJugad.php*: Como el iframe anterior, a éste se le llama de Plantilla.php y desde un iframe que muestra la ficha de un jugador. Utiliza un método de la clase “BD\_OpManager” definida en BD\_Funciones.php para vender al sistema el jugador cuyo identificador recibe como parámetro. Posteriormente recarga la página desde la que se cargó el iframe (por ello se utiliza una función definida en el archivo “padre” o que carga el iframe, porque en este punto del código no sabríamos precisar qué página en concreto cargar).

- *IFrame\_MultiusosAcepOferta.php*: Como el iframe anterior, a éste se le llama de Plantilla.php y desde un iframe que muestra la ficha de un jugador. Se recibe como parámetro el identificador de la oferta que se debe aceptar, se leen diversos datos acerca de la misma y se comprueba si debe terminar la operación: si el equipo comprador tiene suficiente saldo y no se viola la regla que dice que un equipo no puede tener menos de 16 jugadores ni más de 25, se acepta la oferta (lo cual conlleva varias acciones, pero que están definidas en el método de aceptar una oferta de la clase BD\_OpManager, no en este archivo).

- *IFrame\_MultiusosRechOferta.php*: Como el iframe anterior, a éste se le llama de Plantilla.php y desde un iframe que muestra la ficha de un jugador. Se recibe el identificador de una oferta y se la marca como rechazada, sin necesidad de ninguna comprobación.

- *IFrame\_MultiusosHacerContraof.php*: Este iframe, llamado desde Plantilla.php o desde NegociacAbiertas.php, tiene como fin actualizar una oferta de traspaso con una contraoferta hecha por uno de los equipos.

Se reciben como parámetros el identificador de la oferta sobre la que se ha hecho una contraoferta, la cantidad contraoferta y un parámetro de control que indica si el equipo que ha realizado la contraoferta es el equipo poseedor del jugador implicado en la oferta o no; si lo es, se actualiza el campo “contraoferta” de la tabla de ofertas de traspaso, y si no lo es es el campo “Cantidad” el que se actualiza.

- *Alineacion.php*: es un archivo que contiene todo el código relacionado con la pantalla Alineación. En la parte de acceso a datos se incluye el archivo que contiene la variable con el identificador de equipo del usuario, se lee la alineación (su identificador y otros datos) del equipo y los datos de sus jugadores. Se incluye una función que crea una alineación a partir de los jugadores de su equipo con el objetivo de evitar alguna alineación incompleta (por ejemplo, si se ha vendido algún jugador que formaba parte de ella); en este punto se asegura tener una alineación coherente, pues se utilizan unos datos de jugadores obtenidos en ese momento y por tanto de validez vigente.

En la parte de vista se colocan dos divs principales, uno para colocar todos los jugadores del equipo (los de la alineación y los que no, para poder hacer cambios) y otro para elegir las distintas opciones de la alineación y mostrar visualmente la disposición de los jugadores.

Por último, como es habitual, se incluye código JavaScript para modelar el comportamiento dinámico de la interfaz. Por ejemplo, se impide el colocar en el once titular a algún jugador sancionado o lesionado; se muestra la ficha de un jugador pulsando el botón secundario del ratón sobre el div de cada jugador, pues al pulsar el botón primario lo que se hace es seleccionar el jugador para poder hacer un cambio.

Hay que reseñar, también, que desde este archivo se pueden cargar dos iframes: uno que muestre la ficha de un jugador(es, por tanto, un iframe visible) y otro que guarda la alineación dispuesta por el usuario, que es únicamente operativo, modifica la BD pero no tiene parte de interfaz.

- *IFrame\_GuardarAlineacion.php*: es un iframe cargado desde *Alineacion.php*. Su función es guardar en la BD los datos de la alineación dispuesta por el usuario. Recibe como parámetros los jugadores de la alineación y los valores para las distintas opciones y llama a una función que modifica la tabla *Alineación*, para posteriormente recargar la página que lo ha cargado (la página *Alineación*). No cuenta con elementos de interfaz.

- *IFrame\_MostrarJugador.php*: es un iframe que se usa para mostrar por pantalla la ficha de un jugador (en la pantalla de *Plantilla* y *Juveniles* se usa un div definido en el propio archivo, sin cargar el iframe; es así porque en la etapa de acceso a datos ya se han cargado todos los datos de los jugadores, por lo que llamar a un iframe supondría una carga superflua del sistema) desde los archivos *Alineacion.php*, *AlineacionJuvenil.php*, *Clasificacion.php*, *Fichajes.php*, *Iframe\_FichajesVerEquipos.php*, *Iframe\_FichMostrPantBusq.php*, *NegociacAbiertas.php* y *Periodico.php*.

El iframe recibe el identificador del jugador a mostrar y accede al identificador del usuario que está usando el sistema; de ese modo, carga la ficha con las opciones correspondientes (hay cuatro casos posibles: que se trate de un jugador normal del equipo del usuario, que sea un jugador normal de otro equipo, que sea un jugador normal libre, y que sea un jugador juvenil del equipo del usuario).

- *AlineacionJuvenil.php*: es un archivo idéntico a *Alineacion.php*, con la diferencia de que en la capa de acceso a datos se pide la alineación juvenil del equipo, y se leen los datos de los jugadores del equipo.

- *Empleados.php*: es un archivo que contiene todo el código relacionado con la pantalla *Empleados*. En la capa de acceso a datos se obtienen los datos de todos los empleados y candidatos del equipo (distingue entre empleados y candidatos porque los candidatos, a pesar de estar relacionados también con el equipo mediante una clave ajena en la tabla de la Base de Datos, tienen el valor "0" en el campo de temporadas de contrato) y los muestra por pantalla en dos divs principales.

Además del código habitual para modelar la interfaz de usuario y el comportamiento dinámico de la interfaz, el archivo necesita definir cuatro iframes para completar los cuatro procesos que se pueden lanzar desde la pantalla: renovar a un empleado, despedir a un empleado, contratar a un candidato y rechazar a un candidato.

- *IFrame\_EmpleRenovarEmplea.php*: este iframe, cargado desde *Empleados.php*, tiene como fin representar el proceso de renovación de un empleado. Recibe como parámetros el identificador del empleado y el sueldo y temporadas recibidas, y además accede a la BD para obtener los datos del empleado con el fin de determinar si la oferta de renovación es aceptable.

Si es aceptable se completa la operación y se introduce un nuevo evento en la BD (para lo cual se incluye *DameDatosLogin.php* con el fin de conocer el valor de la variable que contiene el identificador del equipo del usuario) y sino le es comunicado al usuario llamando a una función definida en *Empleados.php* que muestra una pantalla de aviso.

- *IFrame\_EmpleDespedEmplead.php*: este iframe, cargado desde *Empleados.php*, tiene como fin representar el proceso de despido de un empleado. Recibe como parámetros el identificador del empleado y del equipo y la indemnización que debe pagar el equipo; comprueba si el equipo dispone de saldo para pagar la indemnización, y de ser así invoca a una función que pone el empleado sin equipo, a otro que restar al saldo del equipo la cantidad de indemnización, actualiza el registro de economía de la BD (para ello debe acceder previamente a la BD y leer el valor de la semana actual) e introduce el nuevo evento.

Si, por el contrario, se determina que el equipo no puede afrontar la indemnización, se da visibilidad a un div definido en el archivo padre (*Empleados.php*) avisando que no se ha podido completar la operación.

- *IFrame\_EmpleFicharCandida.php*: este iframe, cargado desde *Empleados.php*, tiene como propósito completar un proceso de contratación de un candidato. Su lógica es similar al iframe de renovación de un empleado, con una singularidad: además de comprobar si la oferta es aceptable, debe verificar que el equipo tiene saldo suficiente para llevar a cabo la operación (ya que a veces el fichaje de un candidato supone el despido de un empleado).

- *Iframe\_EmpleadosRechazarCandidato.php*: es un iframe cargado desde la página de empleados que únicamente recibe como parámetro un identificador de un empleado y lo deja libre (pone a 0 el campo de la BD que relaciona a un empleado con un equipo).

- *Entrenamiento.php*: es un archivo con todo el código necesario implicado en la pantalla Entrenamiento (php para el acceso a datos, HTML para la interfaz y JavaScript para el comportamiento dinámico de la interfaz).

En la capa de acceso a datos se obtienen los parámetros del entrenamiento establecido en ese momento y el nivel del equipo del usuario para los distintos parámetros de equipo; se muestran por pantalla estos datos y se posibilita al usuario que los modifique a voluntad, respetando un máximo de 20 puntos repartidos entre los distintos parámetros entrenables (tanto individuales como de equipo).

Si el usuario desea guardar la configuración de entrenamiento establecida en un momento dado, se carga un iframe al que se le mandan como parámetros los datos correspondientes.

- *Iframe\_GuardarEntrenam.php*: este iframe, cargado desde la pantalla de Entrenamiento, recibe como parámetros los valores establecidos por el usuario para todos los elementos entrenables. Incluyendo *DameTodosLogin.php* se dispone también del identificador del equipo del usuario, quedando sólo usar el método de la clase de manejo de BD que guarda (modificando un registro de la tabla Entrenamiento de la BD) los valores de entrenamiento de un equipo.

- *Patrocinador.php*: codifica la pantalla Patrocinadores. En la capa de acceso a datos se obtiene el identificador del equipo del usuario, se inicializa la clase de gestión de BD, se obtienen los datos del patrocinador actual, del futuro, se obtienen las ofertas vigentes de patrocinio, se calcula la indemnización que habría que pagar en caso de rescisión del contrato de patrocinio (es un dato que quizás no haya que usarse posteriormente) y se halla la semana "virtual" en la que se encuentra el sistema.

Se presenta la interfaz gráfica, colocando las ofertas (4 como mucho) de patrocinio vigente, y los datos del patrocinador actual y futuro y se modela con JavaScript el comportamiento de la página ante diversos eventos provocados por el usuario; hay que reseñar que aunque el navegador del usuario solo recibe código HTML y Javascript, en los componentes del sistema (alojados en el servidor) se emplea código PHP no sólo para acceder a los datos, sino que en ocasiones para generar el propio código HTML o JavaScript.

A modo de ejemplo, en la capa de datos se obtiene la indemnización a pagar por rescindir el contrato de patrocinio actual y se guarda en una variable(PHP); posteriormente, en la parte de código JavaScript se necesita una variable JavaScript que tenga ese mismo valor, así que se define la variable (dándole un tipo y un nombre) y se emplea una instrucción PHP (un print) que imprime el valor de la variable PHP asignando el valor a la variable JavaScript.

Por último, cuando en la página ocurra algún evento que pueda suponer una modificación en la BD se llama a un iframe que según los parámetros que recibe tendrá una operativa determinada.

- *Iframe\_PatrocGuarOfert.php*: es un iframe cargado desde *Patrocinador.php* encargado de modelar cualquier proceso lanzado desde dicho archivo. Recibe cuatro parámetros, uno para indicar si se trata de una oferta de patrocinio aceptada o rechazada, otro que es el identificador de la oferta de patrocinio involucrada, otro que indica la indemnización a pagar (y que de ser distinto a 0 implica que se desea realizar una rescisión de contrato de patrocinio) y por último uno que indica el identificador del contrato de patrocinio vigente.. Asimismo, se accede a la BD para obtener en que semana y temporada "virtuales" se encuentra el sistema en ese momento.

Al ejecutar cualquier proceso relacionado con los patrocinadores, es un archivo con cierta complejidad; si el parámetro que recoge la indemnización tiene un valor mayor que cero, se procede a eliminar el contrato de patrocinio actual, junto a las operaciones implícitas(colocar al patrocinador siguiente como actual, actualizar el saldo del equipo y el registro de economía, e introducir el nuevo evento).

En otro caso (si el valor de la indemnización es igual a cero) se comprueba si se trata de una oferta rechazada o aceptada; si es rechazada sólo tiene que usar el método para rechazar una oferta de patrocinio, y en caso de ser aceptada, se comprueba si ya existe un patrocinador siguiente, pues en caso de haberlo habría que eliminar el contrato de patrocinador siguiente antes de introducir el nuevo.

- *Economia.php*: es un archivo que contiene todo el código relacionado con la pantalla Economía. En la capa de acceso a datos se obtiene en qué semana se encuentra el sistema, y en función de eso obtiene los datos del registro de economía de un determinado número de semanas, siempre de la semana vigente y de algunas anteriores (de hasta 5; a partir de la sexta semana sólo siempre se obtienen datos del mismo número de semanas, la vigente y las cinco anteriores).

Además, en este componente del archivo también se obtienen los datos del crédito pedido(si existiera) por el equipo del usuario conectado en la sesión.

En la capa de vista se muestran al usuario diversos datos económicos de la semana elegida, mostrando los datos ya cargados en función de la navegación entre las semanas (es decir, si el usuario pulsa sobre un botón para ver los datos de, por ejemplo, hace dos semanas, la capa vista carga los datos de hace dos semanas directamente; no hace falta acceder a la capa de acceso a datos pues todos los datos que se puedan necesitar ya han sido cargados anteriormente).

Por último, en la capa de negocio se modela el comportamiento de la capa de vista, cargando un iframe en el caso de que el usuario desee pedir o devolver un crédito (porque supone una modificación en el estado de la base de datos).

- *IFrame\_EconomiaPedirCred.php*: es un iframe definido y cargado desde *Economia.php* que recibe como parámetros de llamada la cantidad que se va a prestar al equipo mediante el crédito, las semanas que tendrá el equipo como plazo de devolución y los intereses que se le aplicarán; asimismo, se le incluye el archivo de sesión para conocer el valor del identificador del equipo del usuario.

Se comprueba si ya hay un crédito pedido por el equipo y en caso contrario se introduce en la BD los datos del crédito (las demás operaciones implicadas, como aumentar el saldo del equipo, actualizar el registro de economía, etc. se definen en la propia función de almacenamiento del crédito, no el iframe).

- *IFrame\_EconomiaDevCredito.php*: es un iframe cargado desde *Economia.php* cuyo fin es cancelar (devolver) un crédito solicitado por el equipo del usuario. No recibe parámetros, sino que accede a la BD para obtener los datos del crédito vigente y el saldo del equipo, para comprobar si es posible la devolución del crédito; si no lo es se le avisa al usuario llamando a una función definida en *Economia.php*, y si el equipo tiene suficiente saldo se usa un método de la clase de gestión de la BD que cancela un crédito.

- *Estadio.php*: es un archivo con todo el código relacionado con la pantalla Estadio. En la capa de acceso a datos se accede a la BD para leer todos los datos del estadio del equipo del usuario conectado.

En la capa de vista se le muestran estos datos al usuario; como en casi todos los archivos que tienen componentes de acceso a datos y de interfaz, ambas capas están relacionados. Por ejemplo, en este archivo si no hay obras(datos obtenidos en la capa de acceso a datos) en curso en el estadio se le muestran al usuario unos botones para poder hacer obras, mientras que si hay obras estos botones no aparecen.

Cuando el usuario pulse un botón para guardar los cambios introducidos a través de la interfaz, se carga un iframe que recibe como parámetros todos los posibles cambios para ejecutar los cambios pertinentes en la BD.

- *IFrame\_EstadioGuardarCamb.php*: es un iframe cargado desde *Estadio.php* que controla todos los cambios en la BD que pueden provocarse desde la pantalla Estadio. Recibe como parámetros el identificador del estadio, su nombre, el número de asientos en que se aumentará la capacidad al finalizar las obras, si mejorará el nivel de la tienda o los accesos al completarse las obras y el precio de las entradas. Además, se obtiene el identificador del equipo del usuario y la semana en curso del sistema.

A partir de los valores recogidos en la llamada al iframe se calcula el coste de las obras y las semanas que necesitarán para completarse, comprobando que el equipo disponga de saldo para poder acometer las obras; si no es así, se le avisa al usuario llamando a una función definida en *Estadio.php* que muestra una ventana al usuario notificando la incidencia, mientras que si se puede completar la operación se modifica el estado de la base de datos.

- *Clasificacion.php*: se trata de un archivo que contiene todo lo relacionado con la pantalla Clasificación. En la capa de datos se obtiene de la BD el identificador de la liga a la que pertenece el equipo del usuario, además de los jugadores máximos goleadores de esa liga, la clasificación normal y juvenil de dicha liga, además de los partidos disputados y por disputar. A partir de los datos leídos se tratan para construir variables más intuitivas a la hora de usarlas en la capa de vista.

En la capa de interfaz se presentan estos datos a los usuarios, usando etiquetas para navegar entre las distintas secciones (partidos jugados, clasificación y próximos partidos) y botones para navegar dentro de una sección (clasificación, clasificación juvenil y pichichis dentro de la sección clasificación).

Entre los elementos de la interfaz hay algunos interactivos, es decir, que provocan reacciones en el sistema en su interacción con el usuario. Así, si el usuario pulsa sobre un partido ya disputado, se manda al navegador al archivo *Periodico.php* pasándole el identificador del partido pulsado como parámetro de llamada; si por el contrario se pulsa sobre un jugador en la liga de máximos goleadores, se carga el ya comentado iframe *IFrame\_MostrarJugador.php*.

- *Periodico.php*: es un archivo con el código relacionado con la pantalla Periodico. Se muestra al usuario los datos de un partido jugado por el equipo del usuario, siguiendo la siguiente lógica: si el archivo es llamado

desde Clasificacion.php, se coge el valor del identificador mandado como parámetro y se leen los datos del partido con dicho identificador. Si por el contrario no se recibe ningún parámetro (es decir, se carga el archivo desde cualquier pantalla pulsando en el enlace situado en la barra de enlaces) se accede a la BD para obtener el identificador del último partido disputado por el equipo del usuario y se leen los datos de dicho partido.

En la capa de vista se muestran todos los datos del partido al usuario, pudiendo pulsar sobre cualquier jugador mostrado en la interfaz, momento en el que se carga el ya descrito iframe IFrame\_MostrarJugador.php.

*-Fichajes.php:* es un archivo con el código necesario para implementar la pantalla Fichajes. En la capa de acceso a datos sólo se obtienen los jugadores libres y transferibles (y que no pertenezcan al equipo del usuario) del sistema, pues al resto de datos se accede desde iframes independientes.

En la sección de jugadores libres se muestran al usuario algunos datos de los jugadores libres del sistema, y se coloca un botón para que puedan ficharlos. Si el usuario pulsa este botón, se carga un iframe que cuya función es ejecutar el proceso de fichaje de un jugador libres.

La lógica en la sección de jugadores transferibles es la misma, con la diferencia de que el proceso que puede desencadenar el usuario es el de ofertar por un jugador, cargando el iframe ya mencionado IFrame\_MultiusosOfertarJu.php. En todos los casos, al pulsar sobre un jugador, se carga el iframe IFrame\_MostrarJugador.php.

Para las otras dos secciones, Ver Equipos y Búsqueda, se usa un procedimiento diferente. En ambos casos se carga un iframe visible (es decir, con parte de interfaz) sobre la misma pantalla.

*-IFrame\_FichajesFicharJug.php:* es un iframe cargado desde Fichajes.php cuando el usuario desea fichar a un jugador libre. Se reciben como parámetros el identificador del jugador y del equipo, así como el valor del jugador; una vez comprobado que el equipo tiene saldo para realizar el fichaje (en caso contrario se carga una función de Fichajes.php que muestra una ventana de aviso al usuario), se actualiza el estado de la BD asignando el nuevo equipo al jugador, introduciendo el nuevo evento, actualizando el saldo del equipo y el registro económico relativo al equipo.

*-IFrame\_FichajesVerEquipos.php:* es un iframe cargado desde el archivo Fichajes.php cuyo fin es mostrar al usuario un menú de navegación por los distintos equipos de la competición, así como los jugadores de cada equipo y un botón para mandar un mensaje al usuario de otro equipo.

Puede recibir dos valores como parámetros de llamada, un identificador de liga (en cuyo caso sólo se coloca en la capa de vista un select para que el usuario pueda elegir qué equipo desea ver) y un identificador de equipo, en cuyo caso se le muestra la plantilla de dicho equipo al usuario junto con un botón para enviar un mensaje al equipo (siempre que el equipo elegido no sea el del usuario, en cuyo caso no aparece dicho botón), además del ya citado select.

Como en la capa de vista pueden mostrarse jugadores al usuario, se necesitará cargar el iframe que muestra un jugador; asimismo, cuando el usuario quiera mandar un mensaje a otro, se cargará un iframe que ejecutará e proceso.

*-IFrame\_EnviaMensAUsua.php:* es un iframe cargado desde IFrame\_FichajesVerEquipos.php cuyo fin es posibilitar al usuario el enviar un mensaje a otro usuario. Tiene una parte de vista, en la que se muestra una ventana al usuario junto con un formulario para rellenar los distintos campos del mensaje. Una vez el usuario pulsa sobre el botón para enviar el mensaje, se accede al identificador del destinatario usando una variable JavaScript declarada en el iframe padre y se mandan los datos al archivo definido en la declaración del formulario, IFrame\_ProcEnviaMensAUsua.php.

*-IFrame\_FichMostrPantBusq.php:* es un iframe cargado desde Fichajes.php cuyo propósito es mostrar una interfaz al usuario desde la que hacer búsquedas de jugadores.

Se carga en la misma interfaz que la compuesta en Fichajes.php, y se compone de dos partes principales: un recuadro a la izquierda con diversos criterios para realizar la búsqueda y otro a la derecha donde se presentan los resultados de la búsqueda. En el cuadro de la izquierda está insertado un formulario que manda los datos al mismo archivo IFrame\_FichMostrPantBusq.php, por lo que en cada carga del iframe primero se cogen los valores de los distintos parámetros de llamada, y se accede a la BD para recuperar los jugadores que concuerden con los requisitos de búsqueda (así, si realiza una carga sin haber realizado aún una búsqueda, todos los parámetros de llamada tienen valor nulo y la búsqueda devuelve un valor nulo, por lo que en la interfaz en la que se muestran los jugadores está vacía, dando la sensación al usuario de que no se ha hecho ninguna búsqueda, aunque en el archivo si haya ocurrido).

Si el usuario pulsa sobre algún jugador, se carga el iframe IFrame\_MostrarJugador.php.

*-Iframe\_MultiusosOfertarJu.php*: es un iframe cargado desde *Iframe\_MostrarJugador.php* o *Fichajes.php* cuando el usuario desea hacer una oferta por un jugador de otro equipo. En el caso de *Fichajes.php*, puede lanzarse cuando la interfaz muestra los jugadores transferibles del sistema y el usuario pulsa sobre el botón de realizar una oferta.

Este archivo no tiene capa de interfaz, simplemente recibe como parámetros el identificador del jugador por el que se oferta, los identificadores de los equipos que hacen y reciben la oferta, y la cantidad ofrecida; el sistema comprueba si el equipo ofertante ya ha hecho una oferta vigente (aún no rechazada), en cuyo caso se le avisa al usuario sin modificar la BD; si no es así, se inserta en la base de datos la oferta y se modifica el estado del jugador indicando que tiene ofertas de traspaso.

*-NegociacAbiertas.php*: es un archivo con todo el código relacionado con la pantalla Negociaciones Abiertas (como todos los archivos de este tipo, con capa de acceso a datos, comportamiento de objetos de negocio e interfaz).

En la capa de acceso a datos se obtienen las ofertas de traspaso que hayan recibido los jugadores del equipo del usuario y las que haya hecho el equipo del usuario a cualquier jugador (en ambos casos, ofertas que aún no hayan sido rechazadas).

Estos datos se le muestran al usuario en la interfaz de pantalla, junto con unos botones que les permiten tomar varias decisiones (rechazar, aceptar o negociar por ofertas recibidas y retirarse, aceptar contraoferta o contraofertar, según el estado de las negociaciones, por ofertas enviadas).

Como siempre, mediante código JavaScript se modela el comportamiento de los elementos de la interfaz, y para procesos que supongan una modificación en la base de datos se cargan iframes (en este archivo se usan iframes ya descritos, para rechazar, aceptar o negociar una oferta y para mostrar la ficha de un jugador).

*-\_Cod\_Mantenim\_Funciones.php*: Es un archivo que contiene unas funciones para implementar el estado de mantenimiento del sistema. Está incluido en los archivos de simulación de jornada y paso de semana y su funcionamiento es el siguiente: al principio de ambos archivos se crea un archivo auxiliar y al finalizar se borra. Este archivo sirve al resto de componentes del sistema para saber si hay un estado de mantenimiento o un estado de funcionamiento normal.

*\_Cod\_Mantenim\_BloqueaTodo.php*: Es un archivo incluido en todos los ficheros del sistema que codifican pantallas o cargan iframes desde ellas. Comprueba si existe un archivo auxiliar cuya existencia indica que el sistema está en mantenimiento, en cuyo caso se muestra al usuario una pantalla de aviso.

*-EnlacesPie.php*: es un archivo incluido en todos los archivos que se corresponden en una pantalla que contiene enlaces a cada una de las pantallas del juego. Como se necesita saber en que pantalla se encuentra el sistema para poner el enlace a esa pantalla inactivo (colocando un simple span), se utiliza una variable predefinida de PHP, `$_SERVER['PHP_SELF']`, que retorna el valor del nombre de archivo del código PHP.

## 17. Conclusiones

El sector del videojuego en red (también llamado videojuego de navegador, porque lo que único que el usuario necesita para jugar es un navegador web) está en plena explosión. Desde que nació Internet han existido los videojuegos en línea (desde dos ordenadores conectados, a una red local, hasta llegar a la World Wide Web), pero con el crecimiento del ancho de banda en las conexiones Internet nacieron una serie de juegos que han llegado a una gran difusión (World of Warcraft tiene más de 11.5 millones de suscriptores mensuales, Wikipedia, 2010).

El campo de los videojuegos de navegador que podemos categorizar como “manager de fútbol online” también ha experimentado una evolución llamativa en los últimos años. Aunque tienen presencia desde hace muchos años (Hattrick, posiblemente el juego de fútbol online más popular en España, nació en 1999), el desarrollo tecnológico ha permitido mejoras (principalmente gráficas) llamativas.

Así, se ha pasado de juegos en los que los usuarios gestionaban sus equipos y podían seguir sus partidos mediante frases que el motor del sistema iba generando, con una interfaz sencilla y austera, a juegos con una interfaz mucho más elaborada (Goalunited), hasta llegar a juegos con una simulación de partidos con gráficos en tres dimensiones (ManagerZone).



La previsión para el desarrollo final del prototipo obtenido en este proyecto es utilizarlo en el portal <http://www.our.com>; a partir del estado actual del sistema, hay varias vías sobre las que se podría/debería avanzar.

Un aspecto evidente sería trasladar al sistema al tiempo real. Sería un proceso relativamente sencillo, bastaría incluir todos los procesos relacionados con el módulo “Simulación Paso de Semana” en un script ejecutado en tiempo real (cada lunes a las 00:00:00 horas por ejemplo) y los relacionados con “Simulación de Jornada” en otro de la misma manera.

También habría que construir una estructura para poder tener varias ligas y construir un escenario de “paso entre temporadas” (qué procesos se lanzan cuando acaba una temporada y antes de empezar otra).

Otro apartado a retocar serían algunos elementos de la interfaz, por ejemplo diseñar un menú de navegación más atractivo (con imágenes y no con texto); además, evidentemente, se podrían incluir nuevas funcionalidades y pantallas.

No obstante, hay dos principales cuestiones donde sucesivos PFC o trabajos podrían ahondar: la introducción de diversas técnicas de Inteligencia Artificial en diversos apartados del sistema y el auge de las redes sociales en la difusión de los juegos multijugador online.

En el prototipo construido hay equipos no gestionados por un usuario, los llamados equipos del sistema, que no tienen ninguna iniciativa. Se intenta que mantengan una alineación coherente (haciendo cambios automáticamente en la alineación cuando ésta cuente con jugadores lesionados o sancionados) para asegurar una simulación lo más coherente posible, pero mediante técnicas de inteligencia artificial (estrategia de juegos, Redes Neuronales, etc.) podría dotarse a estos equipos de un comportamiento similar al que tendrían de estar controlados por un ser humano (podrían hacer ofertas, decidir sobre ofertas recibidas, cambiar la alineación en función del rival, programar entrenamientos, etc.)

Otro aspecto del sistema que podría desarrollarse ampliamente mediante técnicas de inteligencia artificial es la simulación de los partidos. En nuestro prototipo se calcula el resultado mediante un algoritmo que tiene en cuenta ciertos componentes (calidad de los jugadores, entrenador, equipo local, etc.) además de una cierta dosis de aleatoriedad; mediante técnicas como Redes Neuronales podría hacerse una simulación mucho más compleja y realista de un partido de fútbol.

El otro tema propuesto sobre el que se podría incidir, la importancia de las redes sociales en la difusión de los juegos online, es totalmente distinto al ya comentado pero igualmente importante. En los últimos meses se están incorporando los juegos de our.com a plataformas como Facebook y es que la aparición y auge de las redes sociales ha supuesto una verdadera revolución para los juegos online. Juegos creados específicamente para Facebook (aunque posteriormente pueden trasladarse a otras plataformas) como Farmville, RestaurantCity o MafiaWars han llegado a una enorme difusión en los últimos meses, con un crecimiento casi exponencial.

Son juegos más sencillos (están destinados a un público potencial mucho mayor que juegos pensados para un público más específico como pueden ser World of Warcraft o Hattrick) que los tradicionales juegos multijugador online y más orientados a la interacción social; así, si se decidiera continuar este trabajo en este sentido habría que realizar un estudio de distintas técnicas de viralidad (entendiendo viralidad como “término empleado para referirse a las técnicas de marketing que intentan explotar redes sociales y otros medios electrónicos para producir incrementos exponenciales en renombre de marca”, Wikipedia:2010) en redes sociales.

Otro aspecto relacionado con este PFC sobre el que se podría incidir, incluso, sería en la monetización de portales web. Hay diversas maneras de rentabilizar económicamente un portal de juegos online o un juego online: cobrando por uso mediante suscripciones temporadas (World of Warcraft, por ejemplo, usa esta modalidad; dando un uso general gratuito pero cobrando por ciertas funcionalidades adicionales (Goalunited, p.e.); mediante publicidad (Goalunited, p.e). Un trabajo complementario al realizado sería un estudio de estos procedimientos glosando ventajas e inconvenientes.

## 18. Anexos

## Catálogo de Requisitos

Catálogo de Requisitos		
Id	Subsistema	Descripción
1	General	(A) El sistema simulará una competición futbolística, creando al inicio los equipos que la componen y asegurándose de que todos tengan un nivel semejante en un principio.
2	General	(A) El sistema asignará equipos a los usuarios a medida que le vayan siendo solicitados, y se encargará de gestionar los equipos que no estén asignados a ningún usuario asegurando un normal discurso de la competición.
3	General	(A) El sistema deberá hacer la clasificación de la competición ordenando a los equipos según los siguientes criterios (en orden de prioridad, si varios equipos empatan para un criterio se usa el siguiente y así sucesivamente): puntos totales, puntos ganados en los partidos que implican a los equipos empatados, diferencia de goles marcados y recibidos en los partidos que implican a los equipos empatados, diferencia general entre goles marcados y goles recibidos, sorteo.
4	General	(A) Cada equipo contará con su versión juvenil que también será gestionado por el usuario (se le asignará un equipo normal y uno juvenil)
5	General	(A) Además de los cambios producidos en el estado del sistema por los distintos eventos provocados por un usuario del mismo, en el sistema se implementarán dos eventos que producirán cambios en el estado del mismo: la simulación de una jornada de liga y la simulación del paso de una semana.
6	General	(A) El sistema creará un cierto número de jugadores libres, sin ningún equipo asignado, que podrán ser fichados en todo momento por los equipos.
7	General	(A) Los jugadores estarán representados por una serie de parámetros, algunos estáticos (como su nombre posición en el campo) y otros dinámicos. Entre estos últimos están los que pueden variar por acciones del usuario (como equipo al que pertenece o sueldo que perciben) y otros que varían con el tiempo (los que representan las habilidades del jugador).
8	General	(A) Las habilidades de los jugadores pueden mejorar con el tiempo y el entrenamiento, pero sin pasar de un valor umbral que les asignará el sistema en el momento de su creación.

9	Subsistema de interacción con el usuario	(A) El sistema deberá dar la posibilidad al usuario de cambiar en cualquier momento la alineación(normal y juvenil) guardada por el sistema para su equipo.
10	Subsistema de interacción con el usuario	(A) El usuario tendrá la posibilidad de cambiar en cualquier momento el entrenamiento guardado por el sistema para su equipo.
11	Subsistema de interacción con el usuario	(A) El usuario podrá interactuar con todos los jugadores (componentes de los distintos equipo de sistema) del sistema, pudiendo realizar diversas acciones sobre cada uno de ellos.
12	Subsistema de interacción con el usuario	(A)El usuario se ocupará también de la gestión de su equipo juvenil, cuya función esencial es nutrir de jugadores el equipo normal.
13	Subsistema de interacción con el usuario	(A)Cada equipo del sistema cuenta con un balance económico en cada momento, en función del cual algunas operaciones (como el fichaje de jugadores) le pueden ser denegadas al usuario.
14	Subsistema de interacción con el usuario	(A) El sistema implementará un sistema de créditos; el usuario podrá pedir un crédito al interés fijado por el sistema cuando quiera pero siempre que no tenga uno ya vigente, pudiéndolo devolver en cualquier momento siempre que el saldo del equipo lo permita.
15	Subsistema de interacción con el usuario	(A) El sistema deberá simular un sistema de patrocinio; cada equipo podrá tener un patrocinador vigente y un acuerdo para próximas temporadas con un patrocinador. Cancelar un acuerdo de patrocinio vigente le supondrá al equipo una penalización económica, mientras que anular un acuerdo próximo no tendrá penalización.
16	Subsistema de interacción con el usuario	(A) Cada equipo podrá tener empleados, pertenecientes a diversas categorías y con funciones concretas. Existirá en el sistema también la figura del empleado no asignado a ningún equipo.
17	Subsistema de interacción con el usuario	(A) El sistema creará un cierto número de empleados sin equipo que serán mostrados al usuario y podrán ser fichados para su equipo.
18	Subsistema de interacción con el usuario	(A) Los jugadores normales (no juveniles) y empleados estarán ligados a un equipo mediante contratos definidos por un sueldo y una duración.
19	Subsistema de interacción con el usuario	(A) Cada equipo tendrá asignado un estadio que podrá ser ampliado por el usuario siempre que el saldo económico del equipo lo permita.
20	Subsistema de interacción con el usuario	(A) La habilidad de los distintos jugadores de los equipos estará codificada en diversos parámetros cuyos valores variarán a lo largo del

21	Subsistema de interacción con el usuario	tiempo en función de la edad de cada jugador y del entrenamiento asignado a su equipo. (A) Un usuario podrá en cualquier momento mandar un mensaje a otro usuario.
22	Subsistema de interacción con el usuario	(A) El usuario podrá pasar a un jugador juvenil al equipo normal si y sólo si dicho jugador tiene un conjunto determinado de parámetros con el estado visible.
23	Subsistema de interacción con el usuario	(A) Si su equipo tiene saldo suficiente y su equipo tiene menos de 25 jugadores, el usuario podrá en cualquier momento fichar un jugador libre pagando al sistema el valor estipulado por el jugador; asimismo, siempre que su venta no suponga que el equipo se quede con menos de 16 jugadores, el usuario podrá vender un jugador al sistema por su valor estipulado.
24	Subsistema de interacción con el usuario	(A) El usuario podrá consultar el calendario de partidos de su equipo, los partidos ya disputados y los que están por disputar, así como el estado de la clasificación normal y juvenil y la clasificación de pichichis de su liga.
25	Subsistema de interacción con el usuario	(A) El usuario, ante una oferta de traspaso recibida por un jugador de su equipo, podrá aceptarla, rechazarla o negociar haciendo una contraoferta.
26	Subsistema de interacción con el usuario	(A) El usuario podrá en cualquier momento declarar a un jugador normal de su equipo como transferible.
27		(A) El usuario podrá determinar el precio de las entradas para los partidos disputados en el estadio de su equipo.
28	Subsistema de simulación de jornada de liga	(A) Los usuarios con categoría de Administrador podrán en cualquier momento ejecutar una opción que simulará todos los partidos correspondientes a una jornada de la competición virtual.
29	Subsistema de simulación de jornada de liga	(A) Para obtener los resultados de cada partido el sistema tendrá en cuenta la habilidad de los jugadores de cada equipo disputante del partido, de los empleados con categoría “entrenador” y de otros factores, además de añadir una componente de aleatoriedad.
30	Subsistema de simulación de jornada de liga	(A) La simulación de un partido conllevará, además de la obtención del resultado del mismo, el registro de eventos como las posibles lesiones de jugadores y el cálculo de la recaudación por parte del equipo con carácter local.
31	Subsistema de simulación de paso	(A) El sistema a desarrollar es un

	de semana	prototipo de un sistema que funcionará en tiempo real y que tendrá una serie de ejecución de procesos cada cierto tiempo (por ejemplo, cada semana). Por ello en nuestro prototipo un usuario con privilegios de administrador tendrá siempre la posibilidad de provocar la ejecución de estos procesos englobados en un subsistema llamado “simulación de paso de semana”.
32	Subsistema de simulación de paso de semana	(A) Cuando se ejecute la opción de paso de semana el sistema deberá simular una sesión de entrenamiento. Este proceso tendrá efecto sobre diversos parámetros de los jugadores.
33	Subsistema de simulación de paso de semana	(A) El sistema deberá actualizar las ofertas de patrocinio vigentes para cada equipo, añadiendo una para todo equipo que en ese momento cuente con menos de cuatro.
34	Subsistema de simulación de paso de semana	(A) Con cada paso de semana el sistema recalculará el valor de un parámetro que represente la moral de los jugadores de cada equipo.
35	Subsistema de simulación de paso de semana	(A) Los gastos de cada equipo correspondientes a los sueldos de sus jugadores y empleados, obras en el estadio y devolución de créditos pedidos se llevarán cabo al lanzarse el proceso de simulación de paso de semana.

## Manual de Usuario

### Registro en el sistema

Lo primero debes hacer es registrarte en el sistema. Esto lo puedes hacer en la pantalla de Logia, rellenando los campos de “Pedir Equipo”. El sistema requiere que rellenes los tres campos, por lo que si dejas alguno en blanco se te avisará y no se completará la operación. Además, si utilizas un nombre de usuario ya registrado o el nombre “admin” (restringido a usuarios con privilegios de administrador) también se te avisará de que no es posible completar la asignación de equipo.

Una vez completado este proceso, el sistema te asignará un equipo y tu nombre y contraseña quedarán registrados. A partir de ahora podrás usarlos para rellenar el recuadro de la izquierda la pantalla y entrar al sistema.

### Gestión de equipo

- **Aspectos generales**

Tienes a tu cargo a un equipo que forma parte de una competición futbolística junto con otros 10 equipos, y la posibilidad de gestionar diversos aspectos deportivos y económicos del mismo, así como de interactuar con otros usuarios.

Tu equipo está formando por jugadores, que nunca podrán ser un número menos de 16 ni mayor que 25; asimismo, también tienes a tu cargo al llamado equipo juvenil, formado por jugadores juveniles, con la misma restricción numérica. Además, tu equipo cuenta con un conjunto de empleados y un estadio.

- **Jugadores**

Todos los jugadores del juego están representados mediante distintos parámetros. Además de algunos que indican cosas como el nombre, país, edad o posición del jugador, hay una serie de parámetros que determinan la habilidad o capacidad del jugador. Esta habilidad o destreza general está condensada en un parámetro, "media", que es el baremo que debes usar para orientarte sobre la valía de un jugador.

Cada jugador, dependiendo de su posición, puede tener o no un determinado talento especial. Este talento funciona como un bonus sobre la media del jugador.

No obstante, y dada su distinta naturaleza, se va a incidir en los dos tipos de jugadores que puedes encontrar en el juego: los jugadores normales y los juveniles.

**-Jugadores normales:** son los que forman tu equipo, los otros equipos de la competición (que también tendrán su versión juvenil, pero no es visible al resto de usuarios) y la llamada reserva del sistema. En cualquier pantalla que te encuentres con información sobre un jugador, puedes interactuar con él abriendo su ficha de jugador (clic de ratón), una ventana que te muestra todos los datos del jugador y te permite efectuar distintas acciones sobre él, descritas a continuación:

- **Renovar(Jugador de tu equipo):** Los jugadores están vinculados a tu equipo mediante un contrato, determinado por un sueldo y unos años de contrato. En cualquier momento puedes intentar renovar a un jugador de tu equipo; en función del nivel del jugador y de otros parámetros como su edad o posición, el jugador rechazará o aceptará tu oferta. Debes tener en cuenta que una oferta rechazada puede disminuir el valor del jugador para el parámetro "moral".
- 
- **Vender(Jugador de tu equipo):** Se trata de vender el jugador al sistema por su valor estipulado. Lo puedes hacer en cualquier momento (siempre que no implique violar la restricción del número de jugadores mínimo en plantilla).
- 
- **Aceptar oferta (Jugador de tu equipo):** Cuando se recibe una oferta de traspaso de un jugador por parte de otro equipo puedes aceptarla y vender al jugador por la cantidad determinada.
- 
- **Rechazar oferta (Jugador de tu equipo):** Cuando se recibe una oferta de traspaso de un jugador por parte de otro equipo puedes rechazarla y romper las negociaciones.
- 
- **Negociar oferta (Jugador de tu equipo):** Cuando se recibe una oferta de traspaso de un jugador por parte de otro equipo puedes optar por contraofertar pidiendo una determinada cantidad al equipo ofertante, considerando que si dicho equipo acepta tu contraoferta la operación se completaría sin necesidad de que des tu aprobación.
- 
- **Poner transferible/no transferible(Jugador de tu equipo):** La condición de transferible es una marca que indica al resto de usuarios que tienes cierta predisposición (si es transferible) a aceptar ofertas de traspaso por ese jugador.
- 
- **Fichar (Jugador libre):** Los jugadores libres pueden ser fichados en cualquier momento por su valor estipulado (siempre que no violes la restricción del número máximo de jugadores y tu equipo tenga saldo suficiente).
- **Ofertar (Jugador de otro equipo):** A los jugadores de otros equipos no puedes ficharlo directamente, debes conseguir la aceptación de tu oferta por parte del otro equipo. Oferta que el sistema exigirá sea como máximo el saldo de tu equipo.

**-Jugadores juveniles:** Son los jugadores que forman tu equipo juvenil. Las diferencias respecto a los jugadores normales es que no tienen un contrato y que al empezar el juego todos sus parámetros de nivel serán no visibles; a medida que vayan jugando partidos se les irán destapando esos valores.

Cuando un jugador tenga todos sus valores destapados podrá ser subido al primer equipo, y en cualquier momento un jugador juvenil podrá ser expulsado del equipo. Debes tener en cuenta que

cada semana se te añadir un nuevo juvenil a tu equipo (en todos los casos siempre que se respeten las restricciones en el número de jugadores de los equipos).

### ¿Cómo puedo interactuar con un jugador?

En cualquier pantalla del juego, si al pasar el puntero del ratón por el nombre de un jugador o alguno de sus datos se ilumina esa tira con información, puedes pulsar el botón primario del ratón (secundario en el caso de la pantalla de Alineación) y abrir una ventana con toda la información del jugador y botones para realizar las distintas operaciones con ellos.

#### • Empleados

Tu equipo tiene asignados empleados destinados a distintas tareas. Puedes despedirlos en cualquier momento, aunque con un costo económico. Además, tienes en cada momento un conjunto de empleados sin equipo a los que puedes fichar, pero si fichas a alguno cuya categoría ya cubre un empleado de tu equipo eso implica el tener que despedir al empleado de tu equipo.

Los empleados están ligados a tu equipo mediante un contrato (temporadas y sueldo), en función del cual el sistema calcula el costo del despido. Además, su habilidad para la tarea que lleven a cabo estará determinada por dos parámetros: capacidad y motivación. Por último, aquí está la relación de las distintas categorías de los empleados y de las funciones que cumplen cada uno:

**-Entrenador:** Su función es entrenar a los jugadores y al equipo, además de dirigir al equipo durante los partidos. A mayor nivel, más mejorarán los jugadores y el equipo con el entrenamiento, además de que el equipo tendrá una mayor fuerza general (bonus) en la disputa de los partidos.

**-Segundo entrenador:** Tiene exactamente las mismas funciones que el entrenador, con la diferencia de que su peso en cada uno de ellas es menor que el del entrenador.

**-Entrenador de porteros:** Su función es entrenar a los porteros y prepararlos para los partidos; a mayor nivel, más mejorarán los porteros con el entrenamiento y mejor preparados estarán en los partidos.

**-Preparador físico:** Su función es entrenar físicamente a los jugadores. En función de su nivel y del entrenamiento que se les asigne los jugadores tendrán una evolución determinada para su parámetro "estado físico".

**-Psicólogo:** Su cometido es preparar psicológicamente a los jugadores. Según su nivel, si participación en los partidos y la marcha deportiva del equipo los jugadores tendrán una evolución determinada para su parámetro "moral".

**-Ojeador:** Su función es descubrir jugadores y seguir la evolución de los jugadores juveniles de tu equipo. Según su nivel el usuario podrá ver un número determinado de jugadores libres (perteneciente al sistema) y el sistema descubrirá un determinado número de parámetros ocultos de jugadores juveniles tras la disputa de un partido.

**-Relaciones públicas:** su función es atraer espectadores a los partidos que el equipo del usuario dispute en su estadio.

**-Fisioterapeuta:** su cometido es disminuir el tiempo de recuperación de un jugador cuando estén lesionados.

**-Médico:** su función es disminuir las probabilidades de que un jugador se lesione durante la disputa de un partido.

### ¿Cómo puedo interactuar con un empleado o candidato?

En la pantalla Empleados verás dos recuadros con datos de tus empleados y de los candidatos que puedes contratar; si pulsas con el botón primario del ratón sobre alguno de ellos, todos los datos del empleado o candidato en cuestión aparecerán en los recuadros de abajo y los botones para realizar acciones serán operativos.

- **Entrenamiento**

Cada semana tu equipo realizará una sesión de entrenamiento. Mediante el entrenamiento tus jugadores pueden mejorar los valores para sus parámetros de habilidad, así como los valores de tu equipo para las distintas opciones tácticas.

Dispones de 20 puntos de entrenamiento individual y 20 de entrenamiento de equipo para distribuir como estimes conveniente. En la pantalla de entrenamiento puedes, además, qué nivel tiene tu equipo para los distintos parámetros tácticas.

La mejora de tus jugadores en los diferentes parámetros no depende únicamente de los puntos que asignes a cada parámetro. También tienen influencia la edad del jugador (cuanto más joven, más rápido mejorará), la posición (para cada posición son más importantes unos parámetros que otros), el nivel del entrenador, del entrenador de porteros y del preparador físico.

Otro aspecto que debes tener en cuenta es que los jugadores no mejoran eternamente los valores de sus parámetros; tienen un valor(oculto a los usuarios) para cada parámetro que indica su máximo valor posible.

- **Alineación**

Tu equipo tiene en todo momento una alineación que se usará cuando dispute un partido. Esta alineación la asigna el sistema al crear los equipos, pero tú puedes cambiarla en cualquier momento.

Además de los jugadores de la misma, puedes elegir qué disposición táctica adoptar y qué formación usar, así como practicar o no la opción del fuera de juego.

Hay dos aspectos relevantes que debes considerar a la hora de hacer tu alineación; primero, si ésta contiene a un jugador sancionado o lesionado tu equipo jugará con un jugador menos en el partido correspondiente; el segundo aspecto es que las medias de los jugadores indican su nivel para la posición natural que tengan en el campo, y colocarlos en posición no natural penaliza su media a la hora de jugar el partido.

Esta penalización no es la misma para todos: cuánto menos natural sea (un delantero que juegue de portero, p.e.), más penalización jugará.

También puedes elegir la alineación de tu equipo juvenil; recuerda, solo se destaparán valores al disputarse un partido de los jugadores que jueguen dicho partido.

#### **¿Cómo puedo hacer cambios en la alineación?**

Si pulsas sobre un jugador (del once titular, suplentes o no convocados) verás que la fila con su información cambia de color, eso significa que ese jugador queda seleccionado; si con un jugador seleccionado vuelves a pulsar (botón primario de ratón) ambos jugadores cambian sus posiciones.

#### **¿En qué posición queda cada jugador de la alineación?**

En la pantalla de Alineación verás un recuadro a la derecha que contiene un campo de fútbol con los jugadores de tu alineación repartidos por el mismo. Según la formación que elijas y como repartas a los jugadores en el once titular, podrás observar en ese cuadro la disposición de tus jugadores en el campo.

- **Estadio**

Puedes gestionar los aspectos relativos a tu estadio en la pantalla Estadio. Un estadio está determinado por tres características; capacidad, nivel de accesos y nivel de tienda.

La capacidad determina cuánta gente cabe en el estadio. En un principio todos los estadios tienen 5000 espectadores de capacidad.

El nivel de accesos es un aspecto que el sistema tiene en cuenta a la hora de calcular cuanta gente va a ir a tu estadio en cada partido. En un principio todos los estadios tienen nivel 1.

El nivel de la tienda determina cuánto dinero va a ganar tu equipo semanalmente en concepto de mercadotencia. En un principio todos los estadios tienen nivel de tienda 1.



También en esta pantalla puedes asignar un precio a las entradas para los partidos en tu estadio. El precio de las entradas es un factor a la hora de calcular la asistencia a los partidos en el estadio (a mayor precio, menor asistencia).

#### **Obras en el estadio**

Puedes realizar obras para aumentar la capacidad del estadio o mejorar los niveles de los accesos o de la tienda. Según el número de asientos en que aumentes la capacidad y/o si mejoras el nivel de la tienda o accesos, el sistema calculará un coste y un plazo para la realización de las obras.

El sistema exige que tu equipo tenga saldo suficiente para poder realizar las obras, y los cambios se verán reflejados al final del plazo determinado por el sistema (pero el pago se hace al confirmar la realización de las obras, no se reparte entre las semanas del plazo).

- **Patrocinio**

Puedes suscribir un acuerdo de patrocinio con alguna empresa interesada en patrocinar a tu equipo. En la pantalla Patrocinio se te mostrarán todas las ofertas de patrocinio que tengas en ese momento y los acuerdos que hayas suscrito.

Cada oferta(y cada acuerdo) estarán definidos por las temporadas de duración, el pago fijo y el pago por victoria. El pago semanal fijo lo recibirás cada vez que pase una semana, sin condiciones. El pago por victoria lo recibirás al acabar un partido (normal, no entre equipos juveniles) que gane tu equipo.

Puedes suscribir un acuerdo de patrocinio aunque ya tengas uno vigente, pero si eso supone la obligación de rescindir tendrá un coste económico (en función de las temporadas y del montante de los pagos fijos y por victoria).

- **Créditos**

En la pantalla Economía, además de consultar los ingresos y gastos de tu equipo en las últimas semanas además del saldo del que dispone, se te da la opción de solicitar un crédito.

Cuando solicitas (siempre que no tengas ya uno pedido, en cuyo caso no podrás solicitar otro) un crédito, el sistema te pide la cantidad que deseas y en cuantas semanas quieres devolverlo y calcula (en función del interés aplicado) cuanto debes devolver, cuantas semanas te quedan por hacerlo y cuanto vas a devolver por semana. Es decir, el dinero del crédito se te ingresa íntegramente al pedirlo y has de devolverlo en cuotas semanales.

También se te da la opción de devolverlo íntegramente en cualquier momento.

- **Periódico**

Desde la pantalla Periódico puedes observar todos los datos relativos a un determinado partido disputado por tu equipo. Se te muestra el resultado, las alineaciones de ambos equipos y qué jugadores marcaron un gol, vieron una tarjeta o sufrieron una lesión.

Si la cargas desde el menú de navegación te mostrará los datos sobre el último partido de tu equipo, mientras que si en la pantalla Clasificación pulsas sobre un partido ya disputado se te cargará la página Periódico mostrando los datos de dicho partido.

- **Clasificación de la competición**

Puedes consultar la clasificación de la competición en la pantalla Clasificación. El criterio principal para ordenar a los equipos son los puntos (cada victoria en un partido se premia con 3 puntos, un empate con 1 y una derrota con 0), y a igualdad de puntos entre equipos se utilizan, en orden de prioridad, los siguientes criterios: diferencia entre goles marcados y encajados y mayor número de goles marcados.

En la mencionada pantalla, además, puedes consultar la clasificación de la competición juvenil, así como los máximos goleadores de la competición, los partidos ya disputados por tu equipo y los que va a disputar próximamente.

- **Partidos**

En esta sección se va a explicar brevemente qué factores tiene en cuenta el sistema a la hora de calcular el resultado de un partido.

El factor con más peso es el nivel(media) de los jugadores que forman las alineaciones de los jugadores que juegan el partido. Recordemos que un jugador colocado fuera de su posición natural no tendrá en la práctica el rendimiento previsto, y cuanto más antinatural sea, más penalización (poner un delantero de segundo delantero penaliza menos que ponerlo de lateral izquierdo, por ejemplo).

Otro factor es el nivel del entrenador, segundo entrenador y el entrenador de porteros. Cuanto mayor sea (y por nivel entendemos una combinación entre capacidad y motivación), más fuerza global (o el portero para el caso del entrenador de porteros) tendrá el equipo a la hora de disputar el partido.

También se tiene en cuenta qué equipo juega como local o como suplente. En el fútbol real está estadísticamente comprobado que jugar de local supone una mayor probabilidad de ganar un partido, y en el sistema ocurre lo mismo.

Por último, interviene la aleatoriedad. El resultado de los partidos no está completamente determinado por lo anteriormente nombrado, siempre interviene una dosis de azar en la simulación de cada partido

### **Lesiones**

Disputar un partido puede suponer una lesión para cualquier jugador. Además del azar, en el cálculo de los lesionados en un partido interviene el valor del parámetro “estado físico” y el nivel del médico del equipo.

## **Archivos complementarios**

En el capítulo dieciseis de la memoria se ha descrito cada archivo necesario para el funcionamiento del sistema. No obstante, también se han elaborado diversos scripts utilizados para la carga inicial de datos (introducen en la BD todos los datos necesarios para el funcionamiento del sistema) y para la creación de las tablas de la BD. Se incluye el código de dichos archivos en este apartado. Se incluye el código necesario para la creación de la BD y una mínima parte del código del resto de scripts que ayude a entender la lógica de estos archivos:

### **Scripts Creación BD**

```
CREATE TABLE `alineacion` (  
  `Id` int(11) NOT NULL auto_increment,  
  `Equipo` int(11) NOT NULL default '0',  
  `Estrategia` tinyint(4) NOT NULL default '0',  
  `Formacion` tinyint(4) NOT NULL default '0',  
  `Defensa` tinyint(4) NOT NULL default '0',  
  `FueraJuego` tinyint(4) NOT NULL default '0',  
  `Jug1` int(11) NOT NULL default '0',  
  `Jug2` int(11) NOT NULL default '0',  
  `Jug3` int(11) NOT NULL default '0',  
  `Jug4` int(11) NOT NULL default '0',  
  `Jug5` int(11) NOT NULL default '0',  
  `Jug6` int(11) NOT NULL default '0',  
  `Jug7` int(11) NOT NULL default '0',  
  `Jug8` int(11) NOT NULL default '0',  
  `Jug9` int(11) NOT NULL default '0',  
  `Jug10` int(11) NOT NULL default '0',  
  `Jug11` int(11) NOT NULL default '0',  
  `Jug12` int(11) NOT NULL default '0',  
  `Jug13` int(11) NOT NULL default '0',  
  `Jug14` int(11) NOT NULL default '0',  
  `Jug15` int(11) NOT NULL default '0',  
  `Jug16` int(11) NOT NULL default '0',  
  `Juvenil` tinyint(4) NOT NULL default '0',  
  PRIMARY KEY (`Id`)  
) ENGINE=MyISAM AUTO_INCREMENT=21 DEFAULT CHARSET=latin1 AUTO_INCREMENT=21 ;
```

```

CREATE TABLE `clasificacion` (
  `Id` int(11) NOT NULL auto_increment,
  `Liga` int(11) NOT NULL default '0',
  `Juvenil` tinyint(4) NOT NULL default '0',
  `Equipo` int(11) NOT NULL default '0',
  `Posicion` tinyint(4) NOT NULL default '0',
  `PartJug` tinyint(4) NOT NULL default '0',
  `PartGan` tinyint(4) NOT NULL default '0',
  `PartEmp` tinyint(4) NOT NULL default '0',
  `PartPerd` tinyint(4) NOT NULL default '0',
  `GolesFav` int(11) NOT NULL default '0',
  `GolesCon` int(11) NOT NULL default '0',
  `Puntos` tinyint(4) NOT NULL default '0',
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=21 DEFAULT CHARSET=latin1 AUTO_INCREMENT=21 ;

```

```

CREATE TABLE `contratosequipospatrocinadores` (
  `Id` int(6) NOT NULL auto_increment,
  `Equipo` int(6) NOT NULL default '0',
  `Patrocinador` mediumint(6) NOT NULL default '0',
  `TempInicio` int(3) NOT NULL default '0',
  `TempFin` int(3) NOT NULL default '0',
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=19 DEFAULT CHARSET=latin1 AUTO_INCREMENT=19 ;

```

```

CREATE TABLE `creditos` (
  `Id` int(9) NOT NULL auto_increment,
  `Equipo` int(9) NOT NULL default '0',
  `ImporteRecib` int(9) NOT NULL default '0',
  `ImportDevolTot` int(9) NOT NULL default '0',
  `ImportDevolSem` int(9) NOT NULL default '0',
  `ImportPorDevol` int(9) NOT NULL default '0',
  `SemanasFinCred` tinyint(4) NOT NULL default '0',
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

```

```

CREATE TABLE `datos_tiempo` (
  `Id` int(11) NOT NULL default '0',
  `Semana` int(11) NOT NULL default '0',
  `TempActual` int(11) NOT NULL default '0',
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```

CREATE TABLE `economia` (
  `Id` int(9) NOT NULL auto_increment,
  `Equipo` int(9) NOT NULL default '0',
  `Semana` tinyint(4) NOT NULL default '0',
  `Merchandising` int(9) NOT NULL default '0',
  `IngresosPatro` int(9) NOT NULL default '0',
  `VentaJugadores` int(11) NOT NULL default '0',
  `VentaEntradas` int(9) NOT NULL default '0',
  `CompraJug` int(11) NOT NULL default '0',
  `SueldoJug` int(9) NOT NULL default '0',
  `SueldoEmp` int(9) NOT NULL default '0',
  `IngresosVarios` int(9) NOT NULL default '0',
  `GastosVarios` int(9) NOT NULL default '0',
  `Balance` int(9) NOT NULL default '0',
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=88 DEFAULT CHARSET=latin1 AUTO_INCREMENT=88 ;

```

```

CREATE TABLE `entrenamiento` (
  `Id` int(11) NOT NULL auto_increment,

```

```

`Equipo` int(11) NOT NULL default '0',
`EstadoFisico` tinyint(4) NOT NULL default '0',
`Porteria` tinyint(4) NOT NULL default '0',
`Velocidad` tinyint(4) NOT NULL default '0',
`Agilidad` tinyint(4) NOT NULL default '0',
`Fuerza` tinyint(4) NOT NULL default '0',
`Tecnica` tinyint(4) NOT NULL default '0',
`Disparo` tinyint(4) NOT NULL default '0',
`Regate` tinyint(4) NOT NULL default '0',
`Pase` tinyint(4) NOT NULL default '0',
`VisionJuego` tinyint(4) NOT NULL default '0',
`Resistencia` tinyint(4) NOT NULL default '0',
`Defensa` tinyint(4) NOT NULL default '0',
`JuegoAereo` tinyint(4) NOT NULL default '0',
`CapTactica` tinyint(4) NOT NULL default '0',
`Contrataque` tinyint(4) NOT NULL default '0',
`Posesion` tinyint(4) NOT NULL default '0',
`MurallaDef` tinyint(4) NOT NULL default '0',
`JuegBanEquipo` tinyint(4) NOT NULL default '0',
`FueraJuego` tinyint(4) NOT NULL default '0',
`BalonParado` tinyint(4) NOT NULL default '0',
PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

```

```

CREATE TABLE `equipos` (
`Id` int(11) NOT NULL auto_increment,
`Liga` int(11) NOT NULL default '0',
`Nombre` varchar(50) NOT NULL default "",
`SaldoActual` int(11) NOT NULL default '0',
`Contrataque` float NOT NULL default '0',
`Posesion` float NOT NULL default '0',
`MurallaDef` float NOT NULL default '0',
`JuegBanEquipo` float NOT NULL default '0',
`FueraJuego` float NOT NULL default '0',
`BalonParado` float NOT NULL default '0',
PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=11 DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;

```

```

CREATE TABLE `estadio` (
`Id` int(11) NOT NULL auto_increment,
`Nombre` varchar(40) NOT NULL default "",
`Equipo` int(9) NOT NULL default '0',
`Capacidad` int(6) NOT NULL default '0',
`Accesos` tinyint(4) NOT NULL default '0',
`Tienda` tinyint(4) NOT NULL default '0',
`EntradasLiga` tinyint(4) NOT NULL default '0',
`EntradasCopa` tinyint(4) NOT NULL default '0',
`EntAmis` tinyint(4) NOT NULL default '0',
`ObrasCapacidad` int(6) NOT NULL default '0',
`ObrasAccesos` tinyint(4) NOT NULL default '0',
`ObrasTienda` tinyint(4) NOT NULL default '0',
`CosteObras` int(9) NOT NULL default '0',
`SemanasObras` tinyint(4) NOT NULL default '0',
PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=11 DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;

```

```

CREATE TABLE `eventos` (
`Id` int(11) NOT NULL auto_increment,
`Equipo` int(11) NOT NULL default '0',
`Descrip` varchar(255) NOT NULL default "",
`Fecha` varchar(10) NOT NULL default "",
PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=15 DEFAULT CHARSET=latin1 AUTO_INCREMENT=15 ;

```

```

CREATE TABLE `jugadores` (
  `Id` int(11) NOT NULL auto_increment,
  `Nombre` varchar(25) NOT NULL default "",
  `Apellidos` varchar(75) NOT NULL default "",
  `FechaNac` varchar(10) NOT NULL default "",
  `EdadRetirada` tinyint(4) NOT NULL default '0',
  `Altura` int(3) NOT NULL default '0',
  `Pie` char(1) NOT NULL default "",
  `EstadoFisico` int(2) NOT NULL default '0',
  `Moral` int(2) NOT NULL default '0',
  `Porteria` float NOT NULL default '0',
  `Velocidad` float NOT NULL default '0',
  `Agilidad` float NOT NULL default '0',
  `Fuerza` float NOT NULL default '0',
  `Tecnica` float NOT NULL default '0',
  `Disparo` float NOT NULL default '0',
  `Regate` float NOT NULL default '0',
  `Pase` float NOT NULL default '0',
  `VisionJuego` float NOT NULL default '0',
  `Resistencia` float NOT NULL default '0',
  `Defensa` float NOT NULL default '0',
  `JuegoAereo` float NOT NULL default '0',
  `CapacidadTactica` float NOT NULL default '0',
  `TalentEspecial` varchar(25) NOT NULL default "",
  `PortPotencial` float NOT NULL default '0',
  `VelPotencial` float NOT NULL default '0',
  `AgilPotencial` float NOT NULL default '0',
  `FuerPotencial` float NOT NULL default '0',
  `TecnPotencial` float NOT NULL default '0',
  `DispPotencial` float NOT NULL default '0',
  `RegPotencial` float NOT NULL default '0',
  `PasePotencial` float NOT NULL default '0',
  `ViJuePotencial` float NOT NULL default '0',
  `ResisPotencial` float NOT NULL default '0',
  `DefenPotencial` float NOT NULL default '0',
  `JueAePotencial` float NOT NULL default '0',
  `CapTaPotencial` float NOT NULL default '0',
  `PortSemAnt` float NOT NULL default '0',
  `VelSemAnt` float NOT NULL default '0',
  `AgilSemAnt` float NOT NULL default '0',
  `FuerSemAnt` float NOT NULL default '0',
  `TecSemAnt` float NOT NULL default '0',
  `DisSemAnt` float NOT NULL default '0',
  `RegSemAnt` float NOT NULL default '0',
  `PaseSemAnt` float NOT NULL default '0',
  `VJueSemAnt` float NOT NULL default '0',
  `ResSemAnt` float NOT NULL default '0',
  `DefSemAnt` float NOT NULL default '0',
  `JAerSemAnt` float NOT NULL default '0',
  `CapTaSemAnt` float NOT NULL default '0',
  `PortTapado` tinyint(4) NOT NULL default '0',
  `VelTapado` tinyint(4) NOT NULL default '0',
  `AgilTapado` tinyint(4) NOT NULL default '0',
  `FuerTapado` tinyint(4) NOT NULL default '0',
  `TecTapado` tinyint(4) NOT NULL default '0',
  `DisTapado` tinyint(4) NOT NULL default '0',
  `RegTapado` tinyint(4) NOT NULL default '0',
  `PaseTapado` tinyint(4) NOT NULL default '0',
  `ViJueTapado` tinyint(4) NOT NULL default '0',
  `ResTapado` tinyint(4) NOT NULL default '0',
  `DefTapado` tinyint(4) NOT NULL default '0',
  `JueAeTapado` tinyint(4) NOT NULL default '0',
  `CapTacTapado` tinyint(4) NOT NULL default '0',
  `TalEspTapado` tinyint(4) NOT NULL default '0',
  `Equipo` int(9) NOT NULL default '0',
  `Pais` varchar(15) NOT NULL default "",

```

```

`Sueldo` int(9) NOT NULL default '0',
`TemporadasContrato` int(1) NOT NULL default '0',
`ValorJugador` int(9) NOT NULL default '0',
`Posicion` varchar(25) NOT NULL default "",
`Media` float NOT NULL default '0',
`MediaPotencial` float NOT NULL default '0',
`Dorsal` int(2) NOT NULL default '0',
`Forma` int(2) NOT NULL default '0',
`Goles` int(3) NOT NULL default '0',
`TarAma` int(3) NOT NULL default '0',
`TarRoj` int(3) NOT NULL default '0',
`ParJug` int(3) NOT NULL default '0',
`Transferible` int(1) NOT NULL default '0',
`TieneOfertas` int(1) NOT NULL default '0',
`Juvenil` tinyint(4) NOT NULL default '0',
`Potencial` float NOT NULL default '0',
`Visibilidad` tinyint(4) NOT NULL default '0',
`FechaUltActua` varchar(10) NOT NULL default "",
`SemanasLesion` tinyint(4) NOT NULL default '0',
`Sancionado` tinyint(4) NOT NULL default '0',
`Cara` tinyint(4) NOT NULL default '0',
PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=461 DEFAULT CHARSET=latin1 AUTO_INCREMENT=461 ;

```

```

CREATE TABLE `ligas` (
  `Id` int(11) NOT NULL auto_increment,
  `NumeroLiga` int(11) NOT NULL default '0',
  `DivisionLiga` tinyint(4) NOT NULL default '0',
  `NumEquipos` tinyint(4) NOT NULL default '0',
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=2 DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

```

```

CREATE TABLE `mensajes` (
  `Id` int(11) NOT NULL auto_increment,
  `Fecha` varchar(16) NOT NULL default '0000-00-00-00-00',
  `IdOrigen` int(11) NOT NULL default '0',
  `IdDest` int(11) NOT NULL default '0',
  `Asunto` varchar(150) NOT NULL default "",
  `Mens` varchar(255) NOT NULL default "",
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

```

```

CREATE TABLE `ofertas_traspaso` (
  `Id` int(11) NOT NULL auto_increment,
  `Jugador` int(9) NOT NULL default '0',
  `EquipoReceptor` int(9) NOT NULL default '0',
  `EquipoOfertant` int(9) NOT NULL default '0',
  `Cantidad` int(9) NOT NULL default '0',
  `Contraoferta` int(9) NOT NULL default '0',
  `IdEqUltOf` int(9) NOT NULL default '0',
  `Aceptada` tinyint(4) NOT NULL default '0',
  `Rechazada` tinyint(4) NOT NULL default '0',
  PRIMARY KEY (`Id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

```

```

CREATE TABLE `partidos` (
  `Id` int(11) NOT NULL auto_increment,
  `Competicion` int(11) NOT NULL default '0',
  `Juvenil` tinyint(4) NOT NULL default '0',
  `Jornada` tinyint(4) NOT NULL default '0',
  `Equipo1` int(11) NOT NULL default '0',
  `Equipo2` int(11) NOT NULL default '0',
  `Asistencia` int(11) NOT NULL default '0',

```

```

`Goles1` tinyint(4) NOT NULL default '0',
`Goles2` tinyint(4) NOT NULL default '0',
`Goleadores1` varchar(255) NOT NULL default "",
`Goleadores2` varchar(255) NOT NULL default "",
`TarjeteadosA1` varchar(255) NOT NULL default "",
`TarjeteadosA2` varchar(255) NOT NULL default "",
`TarjeteadosR1` varchar(255) NOT NULL default "",
`TarjeteadosR2` varchar(255) NOT NULL default "",
`Lesionados1` varchar(255) NOT NULL default "",
`Lesionados2` varchar(255) NOT NULL default "",
`Jugadores1` varchar(255) NOT NULL default "",
`Jugadores2` varchar(255) NOT NULL default "",
`Minutos1` varchar(255) NOT NULL default "",
`Minutos2` varchar(255) NOT NULL default "",
`Jugado` tinyint(4) NOT NULL default '0',
PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=181 DEFAULT CHARSET=latin1 AUTO_INCREMENT=181 ;

```

```

CREATE TABLE `patro_ofertas` (
`Id` int(11) NOT NULL auto_increment,
`Nombre` varchar(40) NOT NULL default "",
`Equipo` int(11) NOT NULL default '0',
`PagoSemanal` int(9) NOT NULL default '0',
`PagoVictoria` int(9) NOT NULL default '0',
`Temporadas` int(1) NOT NULL default '0',
`Contestada` tinyint(4) NOT NULL default '0',
PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=45 DEFAULT CHARSET=latin1 AUTO_INCREMENT=45 ;

```

```

CREATE TABLE `personal` (
`Id` int(9) NOT NULL auto_increment,
`Nombre` varchar(20) NOT NULL default "",
`Apellidos` varchar(50) NOT NULL default "",
`Pais` varchar(20) NOT NULL default "",
`FechaNac` varchar(10) NOT NULL default "",
`Categoria` varchar(30) NOT NULL default "",
`Capacidad` int(2) NOT NULL default '0',
`Motivacion` int(2) NOT NULL default '0',
`Equipo` int(9) NOT NULL default '0',
`Sueldo` int(9) NOT NULL default '0',
`TempContrato` int(1) NOT NULL default '0',
PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=291 DEFAULT CHARSET=latin1 AUTO_INCREMENT=291 ;

```

```

CREATE TABLE `usuarios` (
`Id` int(11) NOT NULL auto_increment,
`Usuario` varchar(25) NOT NULL default "",
`Clave` varchar(255) NOT NULL default "",
`Equipo` int(11) NOT NULL default '0',
PRIMARY KEY (`Id`)
) ENGINE=MyISAM AUTO_INCREMENT=12 DEFAULT CHARSET=latin1 AUTO_INCREMENT=12 ;

```

## Scripts Carga Inicial Datos

### Script creación jugadores

```

<?php
//Ficheros de datos: Nombres y apellidos
include("NombresJugadores.php");

//Posibles paises
$Pais[0]=13;
$Pais[1]="España";

```

```

$Pais[2]="Francia";
$Pais[3]="Italia";
$Pais[4]="Alemania";
$Pais[5]="Portugal";
$Pais[6]="Brasil";
$Pais[7]="Argentina";
$Pais[8]="Holanda";
$Pais[9]="Inglaterra";
$Pais[10]="Ecuador";
$Pais[11]="Colombia";
$Pais[12]="Chile";
$Pais[13]="Uruguay";

//Posibles talentos
$TalentoEspecial[0]=5;
$TalentoEspecial[1]="Hombre gol";
$TalentoEspecial[2]="Mago del balon";
$TalentoEspecial[3]="Muralla defensiva";
$TalentoEspecial[4]="Felino bajo palos";
$TalentoEspecial[5]="Gran organizador";

//Posibles posiciones
$Posicion[12]=0;
$Posicion[1]="Portero";
$Posicion[2]="Lateral izquierdo";
$Posicion[3]="Lateral derecho";
$Posicion[4]="Central";
$Posicion[5]="Mediocentro";
$Posicion[6]="Medio izquierdo";
$Posicion[7]="Medio derecho";
$Posicion[8]="Mediapunta";
$Posicion[9]="Extremo izquierdo";
$Posicion[10]="Extremo derecho";
$Posicion[11]="Segundo delantero";
$Posicion[12]="Delantero centro";

function DameNombre($Nacionalidad){
    global
    $NombresEsp,$NombresFra,$NombresIta,$NombresAle,$NombresPor,$NombresBra,$NombresArg,$NombresHol,$NombresIng,$Nombres
    Ecu,$NombresCol,$NombresChi,$NombresUru;
    if ($Nacionalidad=="España") {
        $i=rand(1,$NombresEsp[0]);
        return($NombresEsp[$i]);
    }
    else if ($Nacionalidad=="Francia") {
        $i=rand(1,$NombresFra[0]);
        return($NombresFra[$i]);
    }
    else if ($Nacionalidad=="Italia") {
        $i=rand(1,$NombresIta[0]);
        return($NombresIta[$i]);
    }
    else if ($Nacionalidad=="Alemania") {
        $i=rand(1,$NombresAle[0]);
        return($NombresAle[$i]);
    }
    else if ($Nacionalidad=="Portugal") {
        $i=rand(1,$NombresPor[0]);
        return($NombresPor[$i]);
    }
    else if ($Nacionalidad=="Brasil") {
        $i=rand(1,$NombresBra[0]);
        return($NombresBra[$i]);
    }
    else if ($Nacionalidad=="Argentina") {
        $i=rand(1,$NombresArg[0]);
        return($NombresArg[$i]);
    }
}

```



```

else if ($Nacionalidad=="Holanda") {
    $i=rand(1,$NombresHol[0]);
    return($NombresHol[$i]);
}

else if ($Nacionalidad=="Inglaterra") {
    $i=rand(1,$NombresIng[0]);
    return($NombresIng[$i]);
}

else if ($Nacionalidad=="Ecuador") {
    $i=rand(1,$NombresEcu[0]);
    return($NombresEcu[$i]);
}

else if ($Nacionalidad=="Colombia") {
    $i=rand(1,$NombresCol[0]);
    return($NombresCol[$i]);
}

else if ($Nacionalidad=="Chile") {
    $i=rand(1,$NombresChi[0]);
    return($NombresChi[$i]);
}

else if ($Nacionalidad=="Uruguay") {
    $i=rand(1,$NombresUru[0]);
    return($NombresUru[$i]);
}
}

function DameApellidos($Nacionalidad){
    global
    $ApellidosEsp,$ApellidosFra,$ApellidosIta,$ApellidosAle,$ApellidosPor,$ApellidosBra,$ApellidosArg,$ApellidosHol,$ApellidosIng,$Apellidos
    Ecu,$ApellidosCol,$ApellidosChi,$ApellidosUru;
    if ($Nacionalidad=="España") {
        $i=rand(1,$ApellidosEsp[0]);
        $j=rand(1,$ApellidosEsp[0]);
        $aux=$ApellidosEsp[$i].' '.$ApellidosEsp[$j];
        return($aux);
    }
    else if ($Nacionalidad=="Francia") {
        $i=rand(1,$ApellidosFra[0]);
        return($ApellidosFra[$i]);
    }
    else if ($Nacionalidad=="Italia") {
        $i=rand(1,$ApellidosIta[0]);
        return($ApellidosIta[$i]);
    }
    else if ($Nacionalidad=="Alemania") {
        $i=rand(1,$ApellidosAle[0]);
        return($ApellidosAle[$i]);
    }
    else if ($Nacionalidad=="Portugal") {
        $i=rand(1,$ApellidosPor[0]);
        return($ApellidosPor[$i]);
    }
    else if ($Nacionalidad=="Brasil") {
        $i=rand(1,$ApellidosBra[0]);
        return($ApellidosBra[$i]);
    }
    else if ($Nacionalidad=="Argentina") {
        $i=rand(1,$ApellidosEsp[0]);
        $j=rand(1,$ApellidosEsp[0]);
        $aux=$ApellidosEsp[$i].' '.$ApellidosEsp[$j];
        return($aux);
    }
    else if ($Nacionalidad=="Holanda") {
        $i=rand(1,$ApellidosHol[0]);

```

```

        return($ApellidosHol[$i]);
    }

    else if ($Nacionalidad=="Inglaterra") {
        $i=rand(1,$ApellidosIng[0]);
        return($ApellidosIng[$i]);
    }

    else if ($Nacionalidad=="Ecuador") {
        $i=rand(1,$ApellidosEsp[0]);
        $j=rand(1,$ApellidosEsp[0]);
        $aux=$ApellidosEsp[$i].'.'.$ApellidosEsp[$j];
        return($aux);
    }

    else if ($Nacionalidad=="Colombia") {
        $i=rand(1,$ApellidosEsp[0]);
        $j=rand(1,$ApellidosEsp[0]);
        $aux=$ApellidosEsp[$i].'.'.$ApellidosEsp[$j];
        return($aux);
    }

    else if ($Nacionalidad=="Chile") {
        $i=rand(1,$ApellidosEsp[0]);
        $j=rand(1,$ApellidosEsp[0]);
        $aux=$ApellidosEsp[$i].'.'.$ApellidosEsp[$j];
        return($aux);
    }

    else if ($Nacionalidad=="Uruguay") {
        $i=rand(1,$ApellidosEsp[0]);
        $j=rand(1,$ApellidosEsp[0]);
        $aux=$ApellidosEsp[$i].'.'.$ApellidosEsp[$j];
        return($aux);
    }
}

```

```

function DameNacionalidad(){
    global $Pais;
    $i=rand(1,100);
    if ($i< 15) return("España");
    else if ($i< 25) return("Inglaterra");
    else if ($i< 35) return($Pais[2]);
    else if ($i< 45) return($Pais[3]);
    else if ($i< 55) return($Pais[4]);
    else if ($i< 65) return($Pais[5]);
    else if ($i< 75) return($Pais[6]);
    else if ($i< 82) return($Pais[7]);
    else if ($i< 89) return($Pais[8]);
    else if ($i< 92) return($Pais[10]);
    else if ($i< 95) return($Pais[11]);
    else if ($i< 98) return($Pais[12]);
    else return($Pais[13]);
}

```

```

function DameFechaNac($Tipo){
    if ($Tipo=="Jug"){
        //Devuelve una fecha escogida aleatoriamente
        $Aux1=rand(1,100);
        if ($Aux1<15) $AuxAño=rand(1988,1990);
        else if ($Aux1<45) $AuxAño=rand(1985,1987);
        else if ($Aux1<75) $AuxAño=rand(1983,1984);
        else if ($Aux1<90) $AuxAño=rand(1981,1982);
        else $AuxAño=rand(1978,1980);
        $AuxMes=rand(1,12);
        if ($AuxMes<10) $AuxMes='0'.$AuxMes;
        if ($AuxMes==2) $AuxDia=rand(1,28);
        else if
        (($AuxMes=="01")||($AuxMes=="03")||($AuxMes=="05")||($AuxMes=="07")||($AuxMes=="08")||($AuxMes=="10")||($AuxMes=="12"))
        $AuxDia=rand(1,31);
        else $AuxDia=rand(1,30);
        if ($AuxDia<10) $AuxDia='0'.$AuxDia;
        $AuxFecha=$AuxAño.'-'. $AuxMes.'-'. $AuxDia;
    }
}

```

```

        return($AuxFecha);
    }
    else {
        //Devuelve una fecha escogida aleatoriamente
        $AuxAño=rand(1970,1979);
        $AuxMes=rand(1,12);
        if ($AuxMes<10) $AuxMes='0'.$AuxMes;
        if ($AuxMes==2) $AuxDia=rand(1,28);
        else if
(($AuxMes=="01")||($AuxMes=="03")||($AuxMes=="05")||($AuxMes=="07")||($AuxMes=="08")||($AuxMes=="10")||($AuxMes=="12"))
$AuxDia=rand(1,31);
        else $AuxDia=rand(1,30);
        if ($AuxDia<10) $AuxDia='0'.$AuxDia;
        $AuxFecha=$AuxAño.'-'. $AuxMes.'-'. $AuxDia;
        return($AuxFecha);
    }
}

function DameFechaNacJuv(){
    //Devuelve una fecha escogida aleatoriamente
    $Aux1=rand(1,100);
    if ($Aux1<33) $AuxAño=1990;
    else if ($Aux1<66) $AuxAño=1991;
    else $AuxAño=1992;
    $AuxMes=rand(1,12);
    if ($AuxMes<10) $AuxMes='0'.$AuxMes;
    if ($AuxMes==2) $AuxDia=rand(1,28);
    else if
(($AuxMes=="01")||($AuxMes=="03")||($AuxMes=="05")||($AuxMes=="07")||($AuxMes=="08")||($AuxMes=="10")||($AuxMes=="12"))
$AuxDia=rand(1,31);
    else $AuxDia=rand(1,30);
    if ($AuxDia<10) $AuxDia='0'.$AuxDia;
    $AuxFecha=$AuxAño.'-'. $AuxMes.'-'. $AuxDia;
    return($AuxFecha);
}

function DameTalentoEspecial($Posicion){
    //Devuelve un talento especial adecuado a la posicion del jugador en un determinado tanto por cien de probabilidades
    global $TalentoEspecial;
    $i=rand(1,10);
    if ($i==4) {
        if ($Posicion=="Portero") return $TalentoEspecial[4];
        if ($Posicion=="Central") return $TalentoEspecial[3];
        if ($Posicion=="Mediocentro") return $TalentoEspecial[5];
        if ($Posicion=="Mediapunta" || $Posicion=="Segundo delantero" ) return $TalentoEspecial[2];
        if ($Posicion=="Delantero centro") return $TalentoEspecial[1];
    }
}

function CalculaMediaPotencial(&$Jugador){
    if ($Jugador["Posicion"]=="Portero")
    $Jugador["MediaPotencial"]=round(((($Jugador["PortPotencial"]+0.1*$Jugador["VelPotencial"]+0.3*$Jugador["AgilPotencial"]+0.05*$Jugador["FuerPotencial"]+0.05*$Jugador["PasePotencialPotencial"])/(100+10+30+5+5))*100,4);
    if ($Jugador["Posicion"]=="Lateral izquierdo")
    $Jugador["MediaPotencial"]=round(((($Jugador["VelPotencial"]+0.4*$Jugador["AgilPotencial"]+0.2*$Jugador["FuerPotencial"]+0.1*$Jugador["TecPotencial"]+0.05*$Jugador["DisPotencial"]+0.05*$Jugador["RegPotencial"]+0.4*$Jugador["PasePotencial"]+0.05*$Jugador["ViJuePotencial"]+0.7*$Jugador["ResPotencial"]+0.7*$Jugador["DefPotencial"]+0.2*$Jugador["JueAePotencial"]+0.35*$Jugador["CapTacPotencial"])/(100+40+20+10+5+5+40+5+70+70+20+35))*100,4);
    if ($Jugador["Posicion"]=="Lateral derecho")
    $Jugador["MediaPotencial"]=round(((($Jugador["VelPotencial"]+0.4*$Jugador["AgilPotencial"]+0.2*$Jugador["FuerPotencial"]+0.1*$Jugador["TecPotencial"]+0.05*$Jugador["DisPotencial"]+0.05*$Jugador["RegPotencial"]+0.4*$Jugador["PasePotencial"]+0.05*$Jugador["ViJuePotencial"]+0.7*$Jugador["ResPotencial"]+0.7*$Jugador["DefPotencial"]+0.2*$Jugador["JueAePotencial"]+0.35*$Jugador["CapTacPotencial"])/(100+40+20+10+5+5+40+5+70+70+20+35))*100,4);
    if ($Jugador["Posicion"]=="Central")
    $Jugador["MediaPotencial"]=round(((0.6*$Jugador["VelPotencial"]+0.3*$Jugador["AgilPotencial"]+0.6*$Jugador["FuerPotencial"]+0.05*$Jugador["TecPotencial"]+0.15*$Jugador["PasePotencial"]+0.05*$Jugador["ViJuePotencial"]+0.4*$Jugador["ResPotencial"]+$Jugador["DefPotencial"]+0.7*$Jugador["JueAePotencial"]+0.65*$Jugador["CapTacPotencial"])/(60+30+60+5+15+5+40+100+70+65))*100,4);
    if ($Jugador["Posicion"]=="Mediocentro")
    $Jugador["MediaPotencial"]=round(((0.1*$Jugador["VelPotencial"]+0.15*$Jugador["AgilPotencial"]+0.3*$Jugador["FuerPotencial"]+0.65*$Jugador["TecPotencial"]+0.1*$Jugador["DisPotencial"]+0.1*$Jugador["RegPotencial"]+0.85*$Jugador["PasePotencial"]+0.60*$Jugador["ViJuePotencial"]+0.4*$Jugador["ResPotencial"]+0.4*$Jugador["DefPotencial"]+0.3*$Jugador["JueAePotencial"]+0.7*$Jugador["CapTacPotencial"])/(10+15+30+65+10+10+85+60+40+40+30+70))*100,4);
    if ($Jugador["Posicion"]=="Mediapunta")
    $Jugador["MediaPotencial"]=round(((0.4*$Jugador["VelPotencial"]+0.3*$Jugador["AgilPotencial"]+0.2*$Jugador["FuerPotencial"]+$Jugador["TecPotencial"]+0.15*$Jugador["PasePotencial"]+0.05*$Jugador["ViJuePotencial"]+0.4*$Jugador["ResPotencial"]+0.65*$Jugador["DefPotencial"]+0.7*$Jugador["JueAePotencial"]+0.65*$Jugador["CapTacPotencial"])/(60+30+60+5+15+5+40+100+70+65))*100,4);
}

```

```

cPotencial'+0.45*$Jugador['DisPotencial']+0.75*$Jugador['RegPotencial']+$Jugador['PasePotencial']+$Jugador['ViJuePotencial']+0.3*$Jugador['ResPotencial']+0.1*$Jugador['DefPotencial']+0.1*$Jugador['JueAePotencial']+0.3*$Jugador['CapTacPotencial'])/(40+30+20+100+45+75+100+100+30+10+10+30))*100,4);
if ($Jugador["Posicion"]=="Medio izquierdo")
$Jugador["MediaPotencial"]=round(((0.75*$Jugador['VelPotencial']+0.3*$Jugador['AgilPotencial']+0.2*$Jugador['FuerPotencial']+0.75*$Jugador['TecPotencial']+0.25*$Jugador['DisPotencial']+0.45*$Jugador['RegPotencial']+0.75*$Jugador['PasePotencial']+0.75*$Jugador['ViJuePotencial']+0.4*$Jugador['ResPotencial']+0.25*$Jugador['DefPotencial']+0.15*$Jugador['JueAePotencial']+0.5*$Jugador['CapTacPotencial'])/(75+30+20+75+25+45+75+75+40+25+15+50))*100,4);
if ($Jugador["Posicion"]=="Medio derecho")
$Jugador["MediaPotencial"]=round(((0.75*$Jugador['VelPotencial']+0.3*$Jugador['AgilPotencial']+0.2*$Jugador['FuerPotencial']+0.75*$Jugador['TecPotencial']+0.25*$Jugador['DisPotencial']+0.45*$Jugador['RegPotencial']+0.75*$Jugador['PasePotencial']+0.75*$Jugador['ViJuePotencial']+0.4*$Jugador['ResPotencial']+0.25*$Jugador['DefPotencial']+0.15*$Jugador['JueAePotencial']+0.5*$Jugador['CapTacPotencial'])/(75+30+20+75+25+45+75+75+40+25+15+50))*100,4);
if ($Jugador["Posicion"]=="Extremo izquierdo")
$Jugador["MediaPotencial"]=round(((0.75*$Jugador['VelPotencial']+0.4*$Jugador['AgilPotencial']+0.1*$Jugador['FuerPotencial']+0.85*$Jugador['TecPotencial']+0.45*$Jugador['DisPotencial']+$Jugador['RegPotencial']+0.75*$Jugador['PasePotencial']+0.60*$Jugador['ViJuePotencial']+0.4*$Jugador['ResPotencial']+0.05*$Jugador['DefPotencial']+0.05*$Jugador['JueAePotencial']+0.25*$Jugador['CapTacPotencial'])/(100+40+10+85+45+100+75+60+40+5+5+25))*100,4);
if ($Jugador["Posicion"]=="Extremo derecho")
$Jugador["MediaPotencial"]=round(((0.75*$Jugador['VelPotencial']+0.4*$Jugador['AgilPotencial']+0.1*$Jugador['FuerPotencial']+0.85*$Jugador['TecPotencial']+0.45*$Jugador['DisPotencial']+$Jugador['RegPotencial']+0.75*$Jugador['PasePotencial']+0.60*$Jugador['ViJuePotencial']+0.4*$Jugador['ResPotencial']+0.05*$Jugador['DefPotencial']+0.05*$Jugador['JueAePotencial']+0.25*$Jugador['CapTacPotencial'])/(100+40+10+85+45+100+75+60+40+5+5+25))*100,4);
if ($Jugador["Posicion"]=="Segundo delantero")
$Jugador["MediaPotencial"]=round(((0.75*$Jugador['VelPotencial']+0.4*$Jugador['AgilPotencial']+0.2*$Jugador['FuerPotencial']+0.85*$Jugador['TecPotencial']+0.75*$Jugador['DisPotencial']+0.75*$Jugador['RegPotencial']+0.8*$Jugador['PasePotencial']+0.75*$Jugador['ViJuePotencial']+0.4*$Jugador['ResPotencial']+0.05*$Jugador['DefPotencial']+0.3*$Jugador['JueAePotencial']+0.5*$Jugador['CapTacPotencial'])/(75+40+20+85+75+75+80+75+40+5+30+50))*100,4);
if ($Jugador["Posicion"]=="Delantero centro")
$Jugador["MediaPotencial"]=round(((0.6*$Jugador['VelPotencial']+0.3*$Jugador['AgilPotencial']+0.4*$Jugador['FuerPotencial']+0.6*$Jugador['TecPotencial']+$Jugador['DisPotencial']+0.35*$Jugador['RegPotencial']+0.6*$Jugador['PasePotencial']+0.35*$Jugador['ViJuePotencial']+0.4*$Jugador['ResPotencial']+0.6*$Jugador['JueAePotencial']+0.5*$Jugador['CapTacPotencial'])/(60+30+40+60+100+35+60+35+40+60+50))*100,4);
}

```

```

function CalculaMediaBis(&$Jugador){
$Jugador["MediaPort"]=round(((0.75*$Jugador['Porteria']+0.1*$Jugador['Velocidad']+0.3*$Jugador['Agilidad']+0.05*$Jugador['Fuerza']+0.05*$Jugador['Pase'])/(100+10+30+5+5))*100,4);
$Jugador["MediaLatIzq"]=round(((0.75*$Jugador['Velocidad']+0.4*$Jugador['Agilidad']+0.2*$Jugador['Fuerza']+0.1*$Jugador['Tecnica']+0.05*$Jugador['Disparo']+0.05*$Jugador['Regate']+0.4*$Jugador['Pase']+0.05*$Jugador['VisionJuego']+0.7*$Jugador['Resistencia']+0.7*$Jugador['Defensa']+0.2*$Jugador['JuegoAereo']+0.35*$Jugador['CapacidadTactica'])/(100+40+20+10+5+5+40+5+70+70+20+35))*100,4);
$Jugador["MediaLatDer"]=round(((0.75*$Jugador['Velocidad']+0.4*$Jugador['Agilidad']+0.2*$Jugador['Fuerza']+0.1*$Jugador['Tecnica']+0.05*$Jugador['Disparo']+0.05*$Jugador['Regate']+0.4*$Jugador['Pase']+0.05*$Jugador['VisionJuego']+0.7*$Jugador['Resistencia']+0.7*$Jugador['Defensa']+0.2*$Jugador['JuegoAereo']+0.35*$Jugador['CapacidadTactica'])/(100+40+20+10+5+5+40+5+70+70+20+35))*100,4);
$Jugador["MediaCentral"]=round(((0.6*$Jugador['Velocidad']+0.3*$Jugador['Agilidad']+0.6*$Jugador['Fuerza']+0.05*$Jugador['Tecnica']+0.15*$Jugador['Pase']+0.05*$Jugador['VisionJuego']+0.4*$Jugador['Resistencia']+$Jugador['Defensa']+0.7*$Jugador['JuegoAereo']+0.65*$Jugador['CapacidadTactica'])/(60+30+60+5+15+5+40+100+70+65))*100,4);
$Jugador["MediaMediocentro"]=round(((0.1*$Jugador['Velocidad']+0.15*$Jugador['Agilidad']+0.3*$Jugador['Fuerza']+0.65*$Jugador['Tecnica']+0.1*$Jugador['Disparo']+0.1*$Jugador['Regate']+0.85*$Jugador['Pase']+0.60*$Jugador['VisionJuego']+0.4*$Jugador['Resistencia']+0.4*$Jugador['Defensa']+0.3*$Jugador['JuegoAereo']+0.7*$Jugador['CapacidadTactica'])/(10+15+30+65+10+10+85+60+40+40+30+70))*100,4);
$Jugador["MediaMediapunta"]=round(((0.4*$Jugador['Velocidad']+0.3*$Jugador['Agilidad']+0.2*$Jugador['Fuerza']+$Jugador['Tecnica']+0.45*$Jugador['Disparo']+0.75*$Jugador['Regate']+$Jugador['Pase']+$Jugador['VisionJuego']+0.3*$Jugador['Resistencia']+0.1*$Jugador['Defensa']+0.1*$Jugador['JuegoAereo']+0.3*$Jugador['CapacidadTactica'])/(40+30+20+100+45+75+100+100+30+10+10+30))*100,4);
$Jugador["MediaMedDer"]=round(((0.75*$Jugador['Velocidad']+0.3*$Jugador['Agilidad']+0.2*$Jugador['Fuerza']+0.75*$Jugador['Tecnica']+0.25*$Jugador['Disparo']+0.45*$Jugador['Regate']+0.75*$Jugador['Pase']+0.75*$Jugador['VisionJuego']+0.4*$Jugador['Resistencia']+0.25*$Jugador['Defensa']+0.15*$Jugador['JuegoAereo']+0.5*$Jugador['CapacidadTactica'])/(75+30+20+75+25+45+75+75+40+25+15+50))*100,4);
$Jugador["MediaMedIzq"]=round(((0.75*$Jugador['Velocidad']+0.3*$Jugador['Agilidad']+0.2*$Jugador['Fuerza']+0.75*$Jugador['Tecnica']+0.25*$Jugador['Disparo']+0.45*$Jugador['Regate']+0.75*$Jugador['Pase']+0.75*$Jugador['VisionJuego']+0.4*$Jugador['Resistencia']+0.25*$Jugador['Defensa']+0.15*$Jugador['JuegoAereo']+0.5*$Jugador['CapacidadTactica'])/(75+30+20+75+25+45+75+75+40+25+15+50))*100,4);
$Jugador["MediaExtIzq"]=round(((0.75*$Jugador['Velocidad']+0.4*$Jugador['Agilidad']+0.1*$Jugador['Fuerza']+0.85*$Jugador['Tecnica']+0.45*$Jugador['Disparo']+$Jugador['Regate']+0.75*$Jugador['Pase']+0.60*$Jugador['VisionJuego']+0.4*$Jugador['Resistencia']+0.05*$Jugador['Defensa']+0.05*$Jugador['JuegoAereo']+0.25*$Jugador['CapacidadTactica'])/(100+40+10+85+45+100+75+60+40+5+5+25))*100,4);
$Jugador["MediaExtDer"]=round(((0.75*$Jugador['Velocidad']+0.4*$Jugador['Agilidad']+0.1*$Jugador['Fuerza']+0.85*$Jugador['Tecnica']+0.45*$Jugador['Disparo']+$Jugador['Regate']+0.75*$Jugador['Pase']+0.60*$Jugador['VisionJuego']+0.4*$Jugador['Resistencia']+0.05*$Jugador['Defensa']+0.05*$Jugador['JuegoAereo']+0.25*$Jugador['CapacidadTactica'])/(100+40+10+85+45+100+75+60+40+5+5+25))*100,4);
$Jugador["MediaSegDel"]=round(((0.75*$Jugador['Velocidad']+0.4*$Jugador['Agilidad']+0.2*$Jugador['Fuerza']+0.85*$Jugador['Tecnica']+0.75*$Jugador['Disparo']+0.75*$Jugador['Regate']+0.8*$Jugador['Pase']+0.75*$Jugador['VisionJuego']+0.4*$Jugador['Resistencia']+0.05*$Jugador['Defensa']+0.3*$Jugador['JuegoAereo']+0.5*$Jugador['CapacidadTactica'])/(75+40+20+85+75+75+80+75+40+5+30+50))*100,4);
$Jugador["MediaDelCen"]=round(((0.6*$Jugador['Velocidad']+0.3*$Jugador['Agilidad']+0.4*$Jugador['Fuerza']+0.6*$Jugador['Tecnica']+$Jugador['Disparo']+0.35*$Jugador['Regate']+0.6*$Jugador['Pase']+0.35*$Jugador['VisionJuego']+0.4*$Jugador['Resistencia']+0.6*$Jugador['JuegoAereo']+0.5*$Jugador['CapacidadTactica'])/(60+30+40+60+100+35+60+35+40+60+50))*100,4);

```





```

}

else if ($Jugador["Posicion"]=="Delantero centro") {
    $Jugador["MediaPort"]=$Jugador["MediaPort"]*0.2;
    $Jugador["MediaLatIzq"]=$Jugador["MediaLatIzq"]*0.2;
    $Jugador["MediaLatDer"]=$Jugador["MediaLatDer"]*0.2;
    $Jugador["MediaCentral"]=$Jugador["MediaCentral"]*0.2;
    $Jugador["MediaMediocentro"]=$Jugador["MediaMediocentro"]*0.2;
    $Jugador["MediaMedDer"]=$Jugador["MediaMedDer"]*0.4;
    $Jugador["MediaMediapunta"]=$Jugador["MediaMediapunta"]*0.7;
    $Jugador["MediaMedIzq"]=$Jugador["MediaMedIzq"]*0.4;
    $Jugador["MediaExtIzq"]=$Jugador["MediaExtIzq"]*0.6;
    $Jugador["MediaExtDer"]=$Jugador["MediaExtDer"]*0.6;
    $Jugador["MediaSegDel"]=$Jugador["MediaSegDel"]*0.8;
}

}

function PonerPotencial(&$Jugador){
    $Aux=rand(20,50);
    $Pot=($Aux/10);
    if ($Jugador["Posicion"]=="Portero"){
        $Jugador["PortPotencial"]=$Jugador["Porteria"]*$Pot;
        $Jugador["VelPotencial"]=$Jugador["Velocidad"]*$Pot;
        $Jugador["AgilPotencial"]=$Jugador["Agilidad"]*$Pot;
        $Jugador["FuerPotencial"]=$Jugador["Fuerza"]*$Pot;
        $Jugador["TecPotencial"]=$Jugador["Tecnica"];
        $Jugador["DisPotencial"]=$Jugador["Disparo"];
        $Jugador["RegPotencial"]=$Jugador["Regate"];
        $Jugador["PasePotencial"]=$Jugador["Pase"]*$Pot;
        $Jugador["ViJuePotencial"]=$Jugador["VisionJuego"]*$Pot;
        $Jugador["ResPotencial"]=$Jugador["Resistencia"]*$Pot;
        $Jugador["DefPotencial"]=$Jugador["Defensa"];
        $Jugador["JueAePotencial"]=$Jugador["JuegoAereo"];
        $Jugador["CapTacPotencial"]=$Jugador["CapacidadTactica"]*$Pot;
    }
    else {
        $Jugador["PortPotencial"]=$Jugador["Porteria"];
        $Jugador["VelPotencial"]=$Jugador["Velocidad"]*$Pot;
        $Jugador["AgilPotencial"]=$Jugador["Agilidad"]*$Pot;
        $Jugador["FuerPotencial"]=$Jugador["Fuerza"]*$Pot;
        $Jugador["TecPotencial"]=$Jugador["Tecnica"]*$Pot;
        $Jugador["DisPotencial"]=$Jugador["Disparo"]*$Pot;
        $Jugador["RegPotencial"]=$Jugador["Regate"]*$Pot;
        $Jugador["PasePotencial"]=$Jugador["Pase"]*$Pot;
        $Jugador["ViJuePotencial"]=$Jugador["VisionJuego"]*$Pot;
        $Jugador["ResPotencial"]=$Jugador["Resistencia"]*$Pot;
        $Jugador["DefPotencial"]=$Jugador["Defensa"]*$Pot;
        $Jugador["JueAePotencial"]=$Jugador["JuegoAereo"]*$Pot;
        $Jugador["CapTacPotencial"]=$Jugador["CapacidadTactica"]*$Pot;
    }
    if ($Jugador["PortPotencial"]>=99) $Jugador["PortPotencial"]=99;
    if ($Jugador["VelPotencial"]>=99) $Jugador["VelPotencial"]=99;
    if ($Jugador["AgilPotencial"]>=99) $Jugador["AgilPotencial"]=99;
    if ($Jugador["FuerPotencial"]>=99) $Jugador["FuerPotencial"]=99;
    if ($Jugador["TecPotencial"]>=99) $Jugador["TecPotencial"]=99;
    if ($Jugador["DisPotencial"]>=99) $Jugador["DisPotencial"]=99;
    if ($Jugador["RegPotencial"]>=99) $Jugador["RegPotencial"]=99;
    if ($Jugador["PasePotencial"]>=99) $Jugador["PasePotencial"]=99;
    if ($Jugador["ViJuePotencial"]>=99) $Jugador["ViJuePotencial"]=99;
    if ($Jugador["ResPotencial"]>=99) $Jugador["ResPotencial"]=99;
    if ($Jugador["DefPotencial"]>=99) $Jugador["DefPotencial"]=99;
    if ($Jugador["JueAePotencial"]>=99) $Jugador["JueAePotencial"]=99;
    if ($Jugador["CapTacPotencial"]>=99) $Jugador["CapTacPotencial"]=99;

    $Jugador["Potencial"]=$Pot;
}

}

function PonerMedia(&$Jugador,$Pos){
    if ($Pos=="Portero") $Jugador["Media"]=$Jugador["MediaPort"];
    else if ($Pos=="Lateral izquierdo") $Jugador["Media"]=$Jugador["MediaLatIzq"];
    else if ($Pos=="Lateral derecho") $Jugador["Media"]=$Jugador["MediaLatDer"];
    else if ($Pos=="Central") $Jugador["Media"]=$Jugador["MediaCentral"];
}

```

```

else if ($Pos=="Mediocentro") $Jugador["Media"]=$Jugador["MediaMediocentro"];
else if ($Pos=="Mediapunta") $Jugador["Media"]=$Jugador["MediaMediapunta"];
else if ($Pos=="Medio derecho") $Jugador["Media"]=$Jugador["MediaMedDer"];
else if ($Pos=="Medio izquierdo") $Jugador["Media"]=$Jugador["MediaMedIzq"];
else if ($Pos=="Extremo izquierdo") $Jugador["Media"]=$Jugador["MediaExtIzq"];
else if ($Pos=="Extremo derecho") $Jugador["Media"]=$Jugador["MediaExtDer"];
else if ($Pos=="Segundo delantero") $Jugador["Media"]=$Jugador["MediaSegDel"];
else if ($Pos=="Delantero centro") $Jugador["Media"]=$Jugador["MediaDelCen"];
}

function CalculaSueldo($Posicion,$Media,$Edad){
    if (($Posicion=="Portero") || ($Posicion=="Central") || ($Posicion=="Mediocentro") || ($Posicion=="Delantero centro"))
    $FactorPos=1.3;
    else $FactorPos=1;
    $FactorMedia=$Media/50;
    if ($Edad==18) $FactorEdad=1.7;
    else if ($Edad==19) $FactorEdad=1.65;
    else if ($Edad==20) $FactorEdad=1.6;
    else if ($Edad==21) $FactorEdad=1.55;
    else if ($Edad==22) $FactorEdad=1.5;
    else if ($Edad==23) $FactorEdad=1.45;
    else if ($Edad==24) $FactorEdad=1.4;
    else if ($Edad==25) $FactorEdad=1.35;
    else if ($Edad==26) $FactorEdad=1.3;
    else if ($Edad==27) $FactorEdad=1.25;
    else if ($Edad==28) $FactorEdad=1.2;
    else if ($Edad==29) $FactorEdad=1.15;
    else if ($Edad==30) $FactorEdad=1.1;
    else if ($Edad==31) $FactorEdad=1.05;
    else if ($Edad==32) $FactorEdad=1;
    else if ($Edad==33) $FactorEdad=0.95;
    else if ($Edad==34) $FactorEdad=0.9;
    else if ($Edad==35) $FactorEdad=0.85;
    else if ($Edad==36) $FactorEdad=0.8;
    else if ($Edad==37) $FactorEdad=0.75;
    else if ($Edad==38) $FactorEdad=0.7;
    else if ($Edad==39) $FactorEdad=0.65;
    else if ($Edad==40) $FactorEdad=0.6;
    return((int)$FactorEdad*$FactorMedia*$FactorPos*$FactorMedia*$FactorMedia*25000);
}

function CalculaValor($Posicion,$Media,$Edad){
    $Valor=(12000*($Media*$Media*$Media*$Media))-150000;
    $Valor=$Valor/6000;
    if ($Media>=0) {
        $Aux=(1.5-($Media/100));
        $Valor=$Valor*$Aux*$Aux*$Aux;
    }

    if (($Posicion=="Portero") || ($Posicion=="Delantero centro")) $Valor=$Valor*1.15;
    if ($Edad==18) $Valor=$Valor*1.8;
    if ($Edad==19) $Valor=$Valor*1.7;
    if ($Edad==20) $Valor=$Valor*1.6;
    if ($Edad==21) $Valor=$Valor*1.5;
    if ($Edad==22) $Valor=$Valor*1.4;
    if ($Edad==23) $Valor=$Valor*1.3;
    if ($Edad==24) $Valor=$Valor*1.2;
    if ($Edad==25) $Valor=$Valor*1.1;
    if ($Edad==26) $Valor=$Valor*1.05;
    if ($Edad==27) $Valor=$Valor*1;
    if ($Edad==28) $Valor=$Valor*0.95;
    if ($Edad==29) $Valor=$Valor*0.9;
    if ($Edad==30) $Valor=$Valor*0.8;
    if ($Edad==31) $Valor=$Valor*0.7;
    if ($Edad==32) $Valor=$Valor*0.6;
    if ($Edad==33) $Valor=$Valor*0.5;
    if ($Edad==34) $Valor=$Valor*0.4;
    if ($Edad==35) $Valor=$Valor*0.35;
    if ($Edad==36) $Valor=$Valor*0.3;
    if ($Edad==37) $Valor=$Valor*0.25;
    if ($Edad==38) $Valor=$Valor*0.2;
    if ($Edad==39) $Valor=$Valor*0.15;
    if ($Valor<0) $Valor=1;
    return((int)$Valor);
}

```



```
}
```

```
function CrearPorteroMalo(){  
  global $TalentoEspecial;  
  $Jugador["Pais"]=DameNacionalidad();  
  $Jugador["Posicion"]="Portero";  
  $Jugador["Nombre"]=DameNombre($Jugador["Pais"]);  
  $Jugador["Apellidos"]=DameApellidos($Jugador["Pais"]);  
  $Jugador["FechaNac"]=DameFechaNac("Jug");  
  $Jugador["TalentoEspecial"]=DameTalentoEspecial('Portero');  
  $Jugador["Porteria"]=rand(12,16);  
  $Jugador["Velocidad"]=rand(3,6);  
  $Jugador["Agilidad"]=rand(8,12);  
  $Jugador["Fuerza"]=rand(3,7);  
  $Jugador["Tecnica"]=rand(1,1);  
  $Jugador["Disparo"]=rand(1,1);  
  $Jugador["Regate"]=rand(1,1);  
  $Jugador["Pase"]=rand(2,5);  
  $Jugador["VisionJuego"]=rand(1,2);  
  $Jugador["Resistencia"]=rand(2,5);  
  $Jugador["Defensa"]=rand(1,2);  
  $Jugador["JuegoAereo"]=rand(1,2);  
  $Jugador["CapacidadTactica"]=rand(3,6);  
  $Jugador["Juvenil"]=0;  
  if (rand(1,100)<90) $Jugador["Pie"]="D";  
  else $Jugador["Pie"]="I";  
  PonerPotencial($Jugador);  
  CalculaMediaPotencial($Jugador);  
  CalculaMediaBis($Jugador);  
  PonerMedia($Jugador,$Jugador["Posicion"]);  
  $Jugador["Sueldo"]=CalculaSueldo($Jugador["Posicion"],$Jugador["Media"],DimeCuantosAnosHanPasadoDesde($Jugador["FechaNac"]));  
  $Jugador["Valor"]=CalculaValor($Jugador["Posicion"],$Jugador["Media"],DimeCuantosAnosHanPasadoDesde($Jugador["FechaNac"]));  
  return($Jugador);  
}
```

```
function CrearPorteroJuvMalo(){  
  global $TalentoEspecial;  
  $Jugador["Pais"]=DameNacionalidad();  
  $Jugador["Posicion"]="Portero";  
  $Jugador["Nombre"]=DameNombre($Jugador["Pais"]);  
  $Jugador["Apellidos"]=DameApellidos($Jugador["Pais"]);  
  $Jugador["FechaNac"]=DameFechaNacJuv();  
  $Jugador["TalentoEspecial"]=DameTalentoEspecial('Portero');  
  $Jugador["Porteria"]=rand(11,14);  
  $Jugador["Velocidad"]=rand(3,6);  
  $Jugador["Agilidad"]=rand(5,7);  
  $Jugador["Fuerza"]=rand(3,5);  
  $Jugador["Tecnica"]=rand(1,2);  
  $Jugador["Disparo"]=rand(1,2);  
  $Jugador["Regate"]=rand(1,1);  
  $Jugador["Pase"]=rand(2,3);  
  $Jugador["VisionJuego"]=rand(1,2);  
  $Jugador["Resistencia"]=rand(3,6);  
  $Jugador["Defensa"]=rand(1,2);  
  $Jugador["JuegoAereo"]=rand(1,2);  
  $Jugador["CapacidadTactica"]=rand(2,4);  
  $Jugador["Juvenil"]=1;  
  if (rand(1,100)<90) $Jugador["Pie"]="D";  
  else $Jugador["Pie"]="I";  
  PonerPotencial($Jugador);  
  CalculaMediaPotencial($Jugador);  
  CalculaMediaBis($Jugador);  
  PonerMedia($Jugador,$Jugador["Posicion"]);  
  $Jugador["Sueldo"]=CalculaSueldo($Jugador["Posicion"],$Jugador["Media"],DimeCuantosAnosHanPasadoDesde($Jugador["FechaNac"]));  
  $Jugador["Valor"]=CalculaValor($Jugador["Posicion"],$Jugador["Media"],DimeCuantosAnosHanPasadoDesde($Jugador["FechaNac"]));  
  return($Jugador);  
}
```

Se ha incluido sólo una porción del código que ayude a entender la lógica del script; al final presenta dos funciones (“CrearPorteroMalo” y “CrearPorteroJuvMalo”) de entre las 48 posibilidades que define (para cada una de las 96 posibilidades que define: 12 posiciones, 4 categorías según nivel del jugador, 2 categorías según carácter normal o juvenil del jugador;  $12 \cdot 8 \cdot 2 = 96$ ). Se observa que a cada jugador se le otorga una posición determinada por la función, y a partir de ahí casi todos los valores que se le colocan son aleatorios o aleatorios con cierta determinación (por ejemplo, el nombre y apellidos se eligen aleatoriamente entre un número grupo de posibilidades, pero el grupo sobre el que se eligen viene determinado por la nacionalidad previamente adjudicada).

### Script creación de equipo

La pretensión del script es crear equipos razonablemente equilibrados. Para ello utiliza siempre las mismas funciones cuando un crea jugador, que dan resultados distintos, pero dentro de un rango (por ejemplo, si se ejecuta dos veces una función que devuelve un portero de categoría “bueno” es probable que se obtengan jugadores con diferente nivel, pero ese nivel siempre estará dentro de un rango no muy amplio de posibles valores).

<?php

```
include('FuncionesCreacJugadores.php');
include('FuncionesCreacEmple.php');

function CrearEquipo($NombreEquipo,$IdLiga){
    global $BD_OpMan;
    $Jugadores[0]=46;
    $Jugadores[1]=CrearPorteroNormal();
    $Jugadores[2]=CrearLatDerNormal();
    $Jugadores[3]=CrearLatIzqNormal();
    $Jugadores[4]=CrearCentralNormal();
    $Jugadores[5]=CrearCentralNormal();
    $Jugadores[6]=CrearMediocentroNormal();
    $Jugadores[7]=CrearMedIzqNormal();
    $Jugadores[8]=CrearMedDerNormal();
    $Jugadores[9]=CrearMediapuntaNormal();
    $Jugadores[10]=CrearSegDelNormal();
    $Jugadores[11]=CrearDelCenNormal();
    $Jugadores[12]=CrearPorteroMalo();
    $Jugadores[13]=CrearLatDerMalo();
    $Jugadores[14]=CrearLatIzqMalo();
    $Jugadores[15]=CrearCentralMalo();
    $Jugadores[16]=CrearCentralMalo();
    $Jugadores[17]=CrearMediocentroMalo();
    $Jugadores[18]=CrearMedIzqMalo();
    $Jugadores[19]=CrearMedDerMalo();
    $Jugadores[20]=CrearMediapuntaMalo();
    $Jugadores[21]=CrearSegDelMalo();
    $Jugadores[22]=CrearDelCenMalo();
    $Jugadores[23]=CrearMediocentroMalo();
    $Jugadores[24]=CrearPorteroJuvBueno();
    $Jugadores[25]=CrearLatDerJuvNormal();
    $Jugadores[26]=CrearLatIzqJuvNormal();
    $Jugadores[27]=CrearCentralJuvExcelente();
    $Jugadores[28]=CrearCentralJuvMalo();
    $Jugadores[29]=CrearMediocentroJuvBueno();
    $Jugadores[30]=CrearMedIzqJuvNormal();
    $Jugadores[31]=CrearMedDerJuvBueno();
    $Jugadores[32]=CrearMediapuntaJuvNormal();
    $Jugadores[33]=CrearSegDelJuvNormal();
    $Jugadores[34]=CrearDelCenJuvExcelente();
    $Jugadores[35]=CrearPorteroJuvMalo();
    $Jugadores[36]=CrearLatDerJuvExcelente();
    $Jugadores[37]=CrearLatIzqJuvNormal();
    $Jugadores[38]=CrearCentralJuvNormal();
    $Jugadores[39]=CrearCentralJuvMalo();
    $Jugadores[40]=CrearMediocentroJuvMalo();
    $Jugadores[41]=CrearMedIzqJuvBueno();
    $Jugadores[42]=CrearMedDerJuvNormal();
    $Jugadores[43]=CrearMediapuntaJuvMalo();
    $Jugadores[44]=CrearSegDelJuvMalo();
    $Jugadores[45]=CrearDelCenJuvNormal();
}
```

```

$Jugadores[46]=CrearMediocentroJuvMalo();
$Empleados[0]=9;
$Empleados[1]=CrearEntrenadorMalo();
$Empleados[2]=CrearSegundoEntrenadorMalo();
$Empleados[3]=CrearMedicoMalo();
$Empleados[4]=CrearFisioterapeutaMalo();
$Empleados[5]=CrearRelacionesPublicasMalo();
$Empleados[6]=CrearPsicologoMalo();
$Empleados[7]=CrearEntrenadorPorterosMalo();
$Empleados[8]=CrearPreparadorFisicoMalo();
$Empleados[9]=CrearOjeadorMalo();
$BD_OpMan->CrearEquipo($IdLiga,$NombreEquipo,$Jugadores,$Empleados);
}
?>

```

## Script creación calendario

Cuando se crea una competición es preciso crear un calendario de partidos; no es una cuestión trivial conseguir un calendario de partidos para una competición de diez equipos siguiendo las premisas de que cada equipo juegue siempre dos veces contra cada equipo (una de local y otra de visitante) sin que ningún equipo juegue nunca tres partidos seguidos como local o visitante.

El código empleado es el siguiente:

```

<?php
//Archivo (usado por 'ScriptCreaDivision.php') con las funciones para crear un calendario de la liga.

function CreaCalendarioLiga($IdLiga){
    //Leemos los equipos pertenecientes a esa liga
    $SentenciaSQL="SELECT `equipos`.`Id`, `equipos`.`Nombre` FROM `equipos` WHERE (`equipos`.`Liga`='".$IdLiga."');";
    $EquiposLiga=SQL_LeeArray($SentenciaSQL);

    //Creamos el array con la lista de equipos:
    $Equipos[0]=$EquiposLiga[0][0];
    for ($i=1;$i<=$EquiposLiga[0][0];$i++) $Equipos[$i]=$EquiposLiga[$i][0];

    //Creamos un vector para guardar si en el ultimo partido el equipo fue local o visitante
    for ($i=1;$i<=$EquiposLiga[0][0];$i++) $UltimoPartido[$i]="";

    //Para todas las jornadas impares de la primera vuelta:
    for ($j=1;$j<=$Equipos[0]-1;$j++){
        //Creamos el vector 1:
        $Vector1[1]=$Equipos[1];
        $Vector1[1]=$Equipos[1];

        for ($i=2;$i<=$Equipos[0]/2;$i++) {
            if ($i>$j) {
                $Vector1[$i]=$Equipos[$i-$j+1];
                //UltimoPartido[$Equipos[$i-$j+1]]="l";
            }
            else {
                $Vector1[$i]=$Equipos[$Equipos[0]-$j+$i];
            }
        }

        //Creamos el vector 2:
        for ($i=1;$i<=$Equipos[0]/2;$i++) {
            if (($Equipos[0]-$j-$i+2)>1) {
                $Vector2[$i]=$Equipos[$Equipos[0]-$j-$i+2];
                //UltimoPartido[$Equipos[$Equipos[0]-$j-$i+2]]="v";
            }
            else {
                $Vector2[$i]=$Equipos[$Equipos[0]-$j-$i+2+$Equipos[0]-1];
                //UltimoPartido[$Equipos[$Equipos[0]-$j-$i+2+$Equipos[0]-1]]="v";
            }
        }
        //Vector2[$i]=$Equipos[$Equipos[0]-$j-$i+2];
    }

    //Recorremos los dos vectores para decidir quien juega en casa y quien fuera
    for ($i=1;$i<=$Equipos[0]/2;$i++) {

```

```

$Aux1=$Vector1[$i];
//Si la jornada es par y el equipo local es impar (o al revés), se cambia el equipo local por
el impar
        if (((($j%2==0)&& ($Aux1%2!=0)) || (($j%2!=0)&& ($Aux1%2==0))) {
            $Vector1[$i]=$Vector2[$i];
            $Vector2[$i]=$Aux1;
        }
        //Si el local jugó el anterior partido de local y el visitante de visitante, se intercambian
        if (($UltimoPartido[$Vector1[$i]]=="l") && ($UltimoPartido[$Vector2[$i]]=="v")) {
            $Aux1=$Vector1[$i];
            $Vector1[$i]=$Vector2[$i];
            $Vector2[$i]=$Aux1;
        }
        //Si el visitante lleva dos partidos seguidos como visitante, se le pasa a local
        if (($UltimoPartido[$Vector2[$i]]=="vv")) {
            $Aux1=$Vector1[$i];
            $Vector1[$i]=$Vector2[$i];
            $Vector2[$i]=$Aux1;
        }

        $UltimoPartido[$Vector1[$i]]="l";
        if ($UltimoPartido[$Vector2[$i]]=="l") $UltimoPartido[$Vector2[$i]]="v";
        else $UltimoPartido[$Vector2[$i]]="vv";
    }

    //Jornada de ida y vuelta:
    for ($p=1;$p<=$Equipos[0]/2;$p++){
        $Calendario[$j][$p]=$Vector1[$p].'-'. $Vector2[$p];
        $Calendario[$j+$Equipos[0]-1][$p]=$Vector2[$p].'-'. $Vector1[$p];
    }
}

$Calendario[0][0]=($Equipos[0]*2)-2;
GuardarCalendario($Calendario,$IdLiga);
}

function GuardarCalendario($Calendario,$IdLiga){
global $BD;
//Para todas las jornadas del calendario
for ($i=1;$i<=$Calendario[0][0];$i++){
//Introducimos los partidos de esa jornada
for ($j=1;$j<= ($Calendario[0][0]+2)/4;$j++){
//Partidos normales
        $SentenciaSQL="INSERT INTO `partidos`
('partidos`.`Competicion`,`partidos`.`Jornada`,`partidos`.`Equipo1`,`partidos`.`Equipo2`,`partidos`.`Juvenil`) VALUES
('".$IdLiga."','".$i."','".$Coge_TodoHasta_1er('-', $Calendario[$i][$j])."', '".$Coge_TodoDespues_1er('-', $Calendario[$i][$j])."', '0');";
        SQL_Inserta($SentenciaSQL);
//Partidos juveniles
        $SentenciaSQL="INSERT INTO `partidos`
('partidos`.`Competicion`,`partidos`.`Jornada`,`partidos`.`Equipo1`,`partidos`.`Equipo2`,`partidos`.`Juvenil`) VALUES
('".$IdLiga."','".$i."','".$Coge_TodoHasta_1er('-', $Calendario[$i][$j])."', '".$Coge_TodoDespues_1er('-', $Calendario[$i][$j])."', '1');";
        SQL_Inserta($SentenciaSQL);
    }
}
}

function MueveArrayADerecha($Equipos){
//Creamos array auxiliar
for ($i=1;$i<=$Equipos[0];$i++){
    $AuxEquipos[$i]=$Equipos[$i];
}
for ($i=1;$i<=$Equipos[0];$i++){
    if ($i< $Equipos[0]) $Equipos[$i+1]=$AuxEquipos[$i];
    else if ($i==$Equipos[0]) $Equipos[1]=$AuxEquipos[$i];
}
return ($Equipos);
}

function MueveArrayAlzquierda($Equipos){
//Creamos array auxiliar
for ($i=1;$i<=$Equipos[0];$i++){
    $AuxEquipos[$i]=$Equipos[$i];
}
for ($i=1;$i<=$Equipos[0];$i++){
    if ($i> 1) $Equipos[$i-1]=$AuxEquipos[$i];
    else if ($i==1) $Equipos[$Equipos[0]]=$AuxEquipos[1];
}
}

```

```

    }
return ($Equipos);
}

function Coge_TodoDespues_1er($Caracter,$Cadena){
    //Función que localiza la primera aparición de la cadena $Caracter y devuelve todo lo que hay detrás:
    $Aux_T=explode($Caracter,$Cadena); if (count($Aux_T)<1) return("");
    $Cad=$Aux_T[1]; for ($i=2;$i<count($Aux_T);$i++) $Cad.=$Caracter.$Aux_T[$i];
    return($Cad);
}

function Coge_TodoHasta_1er($Caracter,$Cadena){
    //Función que localiza la primera aparición de la cadena $Caracter y devuelve todo lo que hay antes:
    $Aux_T=explode($Caracter,$Cadena);
    return($Aux_T[0]);
}

?>

```

## Bibliografía

Candidinfo(2006), "Selecting a Server Operating system", <http://web-hosting.candidinfo.com/server-operating-system.asp>, India.

Comunidad Fedora en Castellano, <http://www.fedora-es.com/node/189>, España.

Kutz, Andrew(2006), "VMware Server on Windows vs. VMware Server on Linux", [http://searchservvirtualization.techtarget.com/news/column/0,294698,sid94\\_gci1232096,00.html](http://searchservvirtualization.techtarget.com/news/column/0,294698,sid94_gci1232096,00.html), EEUU,

LSSI (2002), "Ley de Servicios de la Sociedad de la Información", Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información del Ministerio de Industria, Turismo y Comercio del Gobierno de España.

Ministerio de Administraciones Públicas del Gobierno de España,"Análisis del Sistema de Información, Metodología Métrica Versión 3", <http://www.csi.map.es/csi/metrica3/asi.pdf>, pág.6-7-8-14-22-26-28-30-31

Ministerio de Administraciones Públicas del Gobierno de España,"Diseño del Sistema de Información, Metodología Métrica Versión 3", <http://www.csi.map.es/csi/metrica3/dsi.pdf>, pág. 11-45-47

Ministerio de Administraciones Públicas del Gobierno de España,"Estudio de Viabilidad del Sistema, Metodología Métrica Versión 3", <http://www.csi.map.es/csi/metrica3/evs.pdf>, pág. 11

Ministerio de Administraciones Públicas del Gobierno de España,"Planificación de Sistemas de Información, Metodología Métrica Versión 3", <http://www.csi.map.es/csi/metrica3/psi.pdf>, pág. 6

Ministerio de Administraciones Públicas del Gobierno de España,"Técnicas y prácticas, Metodología Métrica Versión 3", <http://www.csi.map.es/csi/metrica3/tecnicas.pdf>, pág.7-34

MySQL Hispano, "Análisis comparativo entre las diferentes alternativas de SGBD ", pág 93, España.

Netcraft, "June 2010 Web Server Survey", <http://news.netcraft.com/archives/2010/06/16/june-2010-web-server-survey.html>, EEUU.

Oracle.com,[http://www.oracle.com/technology/pub/columns/hull\\_php2.html](http://www.oracle.com/technology/pub/columns/hull_php2.html).

Wikipedia.com, [http://es.wikipedia.org/wiki/World\\_of\\_Warcraft](http://es.wikipedia.org/wiki/World_of_Warcraft), España.

Wikipedia.com, [http://es.wikipedia.org/wiki/Marketing\\_viral](http://es.wikipedia.org/wiki/Marketing_viral), España.

| Tipo de documento   | En el capítulo o anexo figura la siguiente referencia parcial  | En la bibliografía aparece la siguiente referencia bibliográfica   |
|---------------------|--|--|
| Recurso Internet    | -Podríamos cerrar el apartado del tipo de servidor a emplear aludiendo a un artículo de una web especializada en todos los aspectos relacionados con los servidores web(Candidinfo, 2006)  | Candidinfo(2006), " Selecting a Server Operating system", <a href="http://web-hosting.candidinfo.com/server-operating-system.asp">http://web-hosting.candidinfo.com/server-operating-system.asp</a> , India, 20/1/2009   |
| Recurso Internet    | Es una herramienta de código abierto pero bastante más lenta y eficiente en el consumo de recursos que MySQL. Está considerada como una mejor opción para desarrollos en los que la consistencia de la BD sea fundamental frente a aspectos como la velocidad y el número de accesos concurrentes (Comunidad Fedora en Castellano, 2009).  | Comunidad Fedora en Castellano, <a href="http://www.fedora-es.com/node/189">http://www.fedora-es.com/node/189</a> , España, 22/1/2009  |
| Recurso Internet    | <p>Según el especialista en virtualización y mantenedor de servidores tanto Windows como Linux Andrew Kutz, uno de los problemas de los servidores Windows es la seguridad, lo que obliga a tener que instalar numerosos parches de manera frecuente, mientras que Linux es bastante más estable; otro problema es el de la inestabilidad del propio sistema operativo. Por el contrario, pese a que los servidores Linux requieren menos esfuerzo dedicado a comprobar su seguridad, pueden resultar menos amigable de cara a su configuración, y tienen que ser actualizados en cuanto hay una actualización del núcleo(kernel) del SO.</p> <p>El especialista concluye aseverando que si bien cada una de las alternativas tiene sus ventajas e inconvenientes, ninguna de ellas es abrumadoramente superior como para recomendar incondicionalmente su uso, y que la decisión final debería depender del grado de comodidad del usuario en un sistema operativo u otro. (<a href="http://searchservervirtualization.techtarget.com/">http://searchservervirtualization.techtarget.com/</a>, 2006).</p> | Kutz, Andrew/(2006). <a href="http://searchservervirtualization.techtarget.com/news/column/0,294698,sid94_gci1232096,00.html">http://searchservervirtualization.techtarget.com/news/column/0,294698,sid94_gci1232096,00.html</a> , EEUU, 10/1/2009                 |
| Norma o legislación | <p>En primer lugar, para todas aquellas aplicaciones web dedicadas al comercio electrónico o a aquellas que sean parte de una actividad económica, la Ley de Servicios de la Sociedad de la Información (LSSI) especifica que debe mostrarse en la página web la siguiente información:</p> <ul style="list-style-type: none"> <li>-Denominación social, NIF, domicilio y dirección de correo electrónico, teléfono o fax.</li> <li>-Datos de inscripción registral.</li> <li>-Códigos de conducta a que esté adherida.</li> <li>-Precios de los productos o servicios que ofrecen.</li> </ul>   | LSSI (2002), "Ley de Servicios de la Sociedad de la Información", Secretaría de Estado de Telecomunicaciones y para la Sociedad de la Información del Ministerio de Industria, Turismo y Comercio del Gobierno de España, 11-7-2002                                |
| Recurso Internet    | <i>En esta tarea se delimita el sistema de información, se indica qué procesos pertenecen al ámbito del sistema, se identifican las entidades externas al mismo que aportan o reciben información y puede obtenerse un modelo conceptual de datos identificando las identidades y relaciones que forman parte del sistema. (Análisis del Sistema de Información, Métrica 3: pág. 6)</i>  | Ministerio de Administraciones Públicas del Gobierno de España,"Análisis del Sistema de Información, Tarea ASI 1.1, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/asi.pdf">http://www.csi.map.es/csi/metrica3/asi.pdf</a> , 3/3/2009 |
| Recurso Internet    | <i>El objetivo de esta tarea es definir, a alto nivel, el entorno tecnológico requerido para dar respuesta a las necesidades de información del sistema a construir. (Análisis del Sistema de Información, Métrica 3: pág. 7)</i>  | Ministerio de Administraciones Públicas del Gobierno de España,"Análisis del Sistema de Información, Tarea ASI 1.2, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3">http://www.csi.map.es/csi/metrica3</a>                            |

|                  |   |  |
|------------------|---|--|
|                  |   | /asi.pdf, 10/2/2009  |
| Recurso Internet | <i>La realización de esta tarea permite considerar las referencias al sistema a implementar desde el punto de vista de estándares, normativas, leyes o recomendaciones que deben tenerse en cuenta a lo largo de todo el desarrollo (Análisis del Sistema de Información, Métrica 3: pág. 8)</i>  | Ministerio de Administraciones Públicas del Gobierno de España, "Análisis del Sistema de Información, Tarea ASI 1.3, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/asi.pdf">http://www.csi.map.es/csi/metrica3/asi.pdf</a> , 10/2/2009 |
| Recurso Internet | <i>Además, la tarea de descomposición del sistema en subsistemas debe estar orientada, principalmente, a los procesos de negocio, aunque es posible adoptar otros criterios lógicos, como por ejemplo la homogeneidad de procesos, la afinidad de requisitos o la prioridad. (Análisis del Sistema de Información, Métrica 3: pág. 14)</i>  | Ministerio de Administraciones Públicas del Gobierno de España, "Análisis del Sistema de Información, Tarea ASI 3.1, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/asi.pdf">http://www.csi.map.es/csi/metrica3/asi.pdf</a> , 23/3/2009 |
| Recurso Internet | <i>El objetivo de esta tarea es identificar y definir las entidades que quedan dentro del ámbito del sistema de información, los atributos de cada entidad (diferenciando aquellos que sean identificadores de la entidad), los dominios de los atributos y las relaciones existentes entre las entidades, indicando cardinalidades mínimas y máximas. (Análisis del Sistema de Información, Métrica 3: pág. 22)</i>  | Ministerio de Administraciones Públicas del Gobierno de España, "Análisis del Sistema de Información, Tarea ASI 6.1, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/asi.pdf">http://www.csi.map.es/csi/metrica3/asi.pdf</a> , 12/4/2009 |
| Recurso Internet | <i>En esta tarea se lleva a cabo la descripción de los subsistemas definidos en la actividad Determinación de Subsistemas de análisis mediante la descomposición en sucesivos niveles de procesos. (Análisis del Sistema de Información, Métrica 3: pág. 26)</i>  | Ministerio de Administraciones Públicas del Gobierno de España, "Análisis del Sistema de Información, Tarea ASI 7.1, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/asi.pdf">http://www.csi.map.es/csi/metrica3/asi.pdf</a> , 4/5/2009  |
| Recurso Internet | <i>El primer objetivo del capítulo es especificar los estándares, directrices y elementos generales a tener en cuenta en la definición de la interfaz de usuario. (Análisis de Sistemas de Información, Métrica 3: pág. 28).</i>  | Ministerio de Administraciones Públicas del Gobierno de España, "Análisis del Sistema de Información, Tarea ASI 8.1, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/asi.pdf">http://www.csi.map.es/csi/metrica3/asi.pdf</a> , 12/6/2009 |
| Recurso Internet | <i>Posteriormente se pretende especificar cada formato individual de la interfaz de pantalla desde un punto de vista estático. Teniendo en cuenta el Modelo de Procesos y la especificación de casos de uso, además de los principios determinados en la tarea anterior, se definirá cada interfaz de pantalla del sistema considerando aspectos interesantes para el posterior diseño y construcción tales como la posibilidad de cambio de tamaño o ubicación de elementos del interfaz; controles y elementos de diseño asociados, indicando cuáles aparecen inicialmente activos e inactivos al visualizar la interfaz de pantalla, etc. (Análisis de Sistemas de Información, Métrica 3: pág. 30).</i> | Ministerio de Administraciones Públicas del Gobierno de España, "Análisis del Sistema de Información, Tarea ASI 8.3, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/asi.pdf">http://www.csi.map.es/csi/metrica3/asi.pdf</a> , 12/6/2009 |
| Recurso Internet | <i>Por último, se definirán los flujos entre los distintos formatos de interfaz de pantalla y dentro del propio forma o (Análisis de Sistemas de Información, Métrica 3: pág. 31).</i>  | Ministerio de Administraciones Públicas del Gobierno de España, "Análisis del Sistema de Información, Tarea ASI 8.4, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/asi.pdf">http://www.csi.map.es/csi/metrica3/asi.pdf</a> , 12/6/2009 |
| Recurso Internet | <i>El objetivo de esta tarea es la definición de los comportamientos no habituales en el sistema, que reflejan situaciones anómalas o secundarias en el funcionamiento y ejecución del sistema de información.<br/>Las excepciones se describen incluyendo los siguientes conceptos:<br/>-Tipo y descripción de la excepción.<br/>-Condiciones previas del sistema.<br/>-Elemento afectado<br/>-Respuesta del sistema.<br/>-Elemento asociado a la respuesta esperada del</i>   | Ministerio de Administraciones Públicas del Gobierno de España, "Diseño de Sistema de Información, Tarea DSI 1.3, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/dsi.pdf">http://www.csi.map.es/csi/metrica3/dsi.pdf</a> , 2/2/2010     |

|                  |   |   |
|------------------|---|---|
|                  | sistema.(Diseño de Sistema de Información, Métrica 3:pág. 11)   |   |
| Recurso Internet | <p><i>El objetivo de la tarea es la definición detallada y completa del entorno necesario para la construcción de los componentes del sistema. Se propone que la especificación se realiza en base a estos conceptos:</i></p> <ul style="list-style-type: none"> <li>-Entorno tecnológico: hardware, software y comunicaciones.</li> <li>-Herramientas de construcción, generadores de código, etc.</li> <li>-Restricciones técnicas del entorno.</li> <li>-Requisitos de operación y seguridad del entorno de construcción. (Diseño de Sistema de Información, Métrica 3:pág. 45)</li> </ul> | Ministerio de Administraciones Públicas del Gobierno de España,"Diseño de Sistema de Información, Tarea DSI 8.1, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/dsi.pdf">http://www.csi.map.es/csi/metrica3/dsi.pdf</a> , 21/3/2010                |
| Recurso Internet | <p><i>El objetivo de la tarea es realizar una especificación detallada de cada componente, en pseudocódigo o lenguaje natural, completando la información que se considere necesaria según el entorno tecnológico. (Diseño de Sistema de Información, Métrica 3:pág. 47)</i></p>  | Ministerio de Administraciones Públicas del Gobierno de España,"Diseño de Sistema de Información, Tarea DSI 8.3, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/dsi.pdf">http://www.csi.map.es/csi/metrica3/dsi.pdf</a> , 2/5/2010                 |
| Recurso Internet | <p><i>El propósito de la tarea es una primera definición de la alternativa a construir. Para ello se elabora una primera versión del catálogo de requisitos y se especifica mediante la técnica de los casos de uso las funcionalidades que se desea contenga el sistema. (Estudio de la Viabilidad del Sistema, Métrica 3:pág. 11).</i></p>  | Ministerio de Administraciones Públicas del Gobierno de España,"Estudio de Viabilidad del Sistema, Tarea EVS 4.2, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/evs.pdf">http://www.csi.map.es/csi/metrica3/evs.pdf</a> , 20/2/2009               |
| Recurso Internet | <p><i>El objetivo de la tarea es definir el ámbito de los procesos de la organización a considerar y definir el alcance del sistema, es decir, los objetivos específicos del sistema a construir. Puede ser necesario determinar distintos objetivos para cada proceso del proyecto (Planificación de Sistemas de Información, Métrica 3:pág. 6).</i></p>   | Ministerio de Administraciones Públicas del Gobierno de España,"Planificación de Sistemas de Información, Tarea PSI 2.1, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/psiproc.pdf">http://www.csi.map.es/csi/metrica3/psiproc.pdf</a> , 2/2/2009 |
| Recurso Internet | <p>casos de uso, cuyos objetivos pueden describirse de la siguiente manera:</p> <ul style="list-style-type: none"> <li>-Capturar los requisitos funcionales del sistema y expresarlos desde el punto de vista del usuario.</li> <li>-Guiar todo el proceso de desarrollo del sistema de información. (Técnicas y Prácticas, Métrica 3: pág. 7)</li> </ul>   | Ministerio de Administraciones Públicas del Gobierno de España,"Técnicas y prácticas, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/tecnicas.pdf">http://www.csi.map.es/csi/metrica3/tecnicas.pdf</a> , 30/3/2009                                 |
| Recurso Internet | <p>proceso primitivo(proceso que no se descompone en otros procesos;Técnicas y Prácticas, Métrica 3: pág. 34)</p>   | Ministerio de Administraciones Públicas del Gobierno de España,"Técnicas y prácticas, Metodología Métrica Versión 3", <a href="http://www.csi.map.es/csi/metrica3/tecnicas.pdf">http://www.csi.map.es/csi/metrica3/tecnicas.pdf</a> , 6/5/2009                                  |
| Estudio          | <p>Según el portal MySQL Hispano algunas de las empresas de Internet más importantes como pueden ser Yahoo o Google usan MySQL(<a href="http://www.mysql-hispano.org/articulos/num43/analisis-comparativo.pdf">http://www.mysql-hispano.org/articulos/num43/analisis-comparativo.pdf</a>:93).</p>   | MySQL Hipano, "Análisis comparativo entre las diferentes alternativas de SGBD ", pág 93, España.  |
| Recurso Internet | <p>Estudio Web Server Survey de Netcraft (NetCraft, 2010).</p>  | Netcraft, <a href="http://news.netcraft.com/">http://news.netcraft.com/</a> , EEUU, 16/6/2010   |
| Recurso Internet | <p>No obstante, podemos incidir un poco más en la comparación entre ASP.NET y PHP a partir de un artículo publicado por un profesional en la materia</p> <ul style="list-style-type: none"> <li>-La más reciente versión de PHP, la 5, mantiene la eficiencia tradicional del lenguaje y añade funcionalidades orientadas a la orientación a objetos.</li> <li>-Una gran ventaja de ASP.NET es el entorno Visual Studio; es una herramienta que facilita el proceso de generar código, ofrece un buen depurador y permite manipular bases de datos Oracle directamente.</li> </ul>            | Oracle.com, <a href="http://www.oracle.com/technology/pub/columns/hull_php2.html">http://www.oracle.com/technology/pub/columns/hull_php2.html</a> , EEUU, 20/2009   |



|                  |   |   |
|------------------|---|---|
|                  | <p>-ASP.NET tiene unos mayores requerimientos de memoria en el servidor web.</p> <p>-La conclusión del autor es que PHP, con la inclusión de funcionalidades orientadas a objetos y manteniendo su tradicional eficiencia es la mejor elección, particularmente con servidores Linux con el programa servidor Apache (Oracle.com, 2009)</p> |   |
| Recurso Internet | World of Warcraft tiene más de 11.5 millones de suscriptores mensuales( Wikipedia, 2010)  | Wikipedia.com,<br><a href="http://es.wikipedia.org/wiki/World_of_Warcraft">http://es.wikipedia.org/wiki/World_of_Warcraft</a> , España, 20/5/2010 |
| Recurso Internet | entendiendo viralidad como "término empleado para referirse a las técnicas de marketing que intentan explotar redes sociales y otros medios electrónicos para producir incrementos exponenciales en renombre de marca"( Wikipedia:2010)   | Wikipedia.com,<br><a href="http://es.wikipedia.org/wiki/Marketing_viral">http://es.wikipedia.org/wiki/Marketing_viral</a> , España, 2/6/2010      |