Descripción del problema DDSI

Antonio Checa Molina
Iñaki Madinabeitia
Bruno Santidrián
Darío Sierra

17 de octubre de 2017

Índice

1. Gestión de un registro de partidas en juegos generales

El problema a resolver es mantener un registro rápido y funcional de partidas para cualquier tipo de juegos, deportes o competiciones con el fin de facilitar análisis y entrenamiento de jugadores. Hay que desarrollar un sistema de información de organización de las partidas y de los juegos.

Al principio, un grupo de gestores necesita proporcionar información de un juego y de sus partidas: aquello que se guarda, como la puntuación, los equipos, los jugadores de los equipos o un vídeo del partido. Una vez guardados estos atributos, el usuario podrá acceder al registro de partidas de un juego concreto e incluir aquellas de las que tenga datos.

Por ejemplo, si se añade el juego baloncesto junto a un conjunto de atributos como {Puntuación, Equipos, Puntuación de cada parcial, vídeos}, cada partida contendrá esta información y los usuarios podrán acceder a esta. En esta práctica nos restringiremos a un pequeño número de juegos, realizando el diseño de base de datos para cada uno.

1.1. Consulta (Iñaki Madinabeitia)

La función principal es que el usuario pueda buscar con facilidad partidas en función de los atributos. Necesita permitir al usuario fijar uno de los atributos, buscar en un rango de parámetros o hacer búsquedas flexibles en cualquiera de los juegos soportados.

Imaginemos que estamos con Baloncesto con muchos partidos de varios equipos ya añadidos. La aplicación debe permitir al usuario buscar aspectos simples como 'Todos los partidos del Unicaja' o 'Partidos en los que ganó el Real Madrid' o 'Qué equipo jugó en tal partido como titular'. Todo esto son consultas a una base de datos, pero la aplicación debe permitir al usuario realizarlas sin tener conocimiento en absoluto sobre bases de datos, apoyadas con una intuitiva interfaz gráfica.

También debe permitir búsqueda donde el usuario otorgue uno o varios parámetros. Con la idea de usar la consulta en esta aplicación para estudiar al rival, también debe permitir buscar partidos donde, según el ejemplo anterior, la diferencia en el marcador fue menos de 10, o donde tal jugador jugó tantos minutos, o donde tal jugador metió tantos puntos. En estos ejemplos, el parámetro sería los números.

Como la aplicación debe permitir hacer consultas sin que el usuario conozca sobre cómo se consulta a una base de datos, debemos limitar qué aspectos se pueden consultar. En resumen, la aplicación permitirá mostrar cualquier atributo de cualquier partida, mostrar las partidas seleccionadas tras hacer una restricción del valor de los atributos (con paso de parámetros en algunos casos concretos); y todo esto siempre en el ámbito del mismo juego.

1.2. Consejos (Bruno Santidrián)

En todos los juegos en los que se guarde el estilo de juego, la aplicación aconseja cómo jugar en determinadas condiciones, recomendando al usuario aquellos estilos que más probabilidades tengan de ganar.

Dado un juego, el usuario podrá elegir uno o varios atributos de este (e.g. estilo de juego en un partido de baloncesto: ofensivo o defensivo) y ciertos datos extra, como el oponente al que se quiere enfrentar, o ciertos valores de los atributos que puede tener el oponente y el sistema analiza los datos guardados en la base de datos para aconsejar unos valores que puedan ser beneficiosos (e.g. jugar ofensivo).

El subsistema debería dar información extra sobre el proceso de como ha elgido los valores de los atributos, como el número de jugadas analizadas y la fiabilidad de los datos.

También es posible que el subsistema de consejos use funcionalidades del subsistema de estadísticas, siempre que dicha funcionalidad esté implementada, para que las recomendaciones sean más certeras.

1.3. Inclusión de partidas (Darío Sierra)

La inclusión de las partidas necesita que el usuario incluya aquellos atributos vitales en una partida, pudiendo o no incluir los opcionales. Necesitaría insertarlos por la interfaz gráfica.

El sistema de información que planteamos crear deberá incrementar el número de juegos disponibles con el avance del tiempo, para asegurar su uso continuado. Por tanto, distinguimos entre inclusión de juegos (trabajo de los administradores de la base de datos) y la inclusión de partidas (trabajo de los usuarios).

Una partida será un registro más de la tabla de datos asociada a cada juego. Como atributo vital se necesitará añadir el nombre del juego, así como la puntuación final de dicha partida; estos serán atributos comunes a cada a todas las partidas, pues todas pertenecerán a un juego y todas tendrán una puntuación asociada.

Los atributos opcionales serán aquellos inherentes a cada juego; no tiene sentido la inclusión de un atributo 'Jugadores del equipo' en un juego de cartas, ni un 'baraja del oponente' en uno que no lo sea. Aquí opcional no se entiende como 'poco relevante' pues estos atributos son la razón de ser del propio sistema de información. Los datos inherentes a cada juego, los que lo hacen distinguible del resto, serán objeto del análisis en el sistema.

Las partidas se incluirán desde la interfaz gráfica, donde una vez seleccionado el tipo de juego, se abrirán una serie de atributos (a definir por los administradores del sistema) que permitirán al jugador guardar su partida. Entre estos podremos encontrar: 'Jugadores del equipo', 'Mi Baraja', 'Baraja del oponente', 'Número de faltas cometido', 'Pichichi del encuentro', etc.

Aunque hemos clarificado que la inclusión de partidas debe ser trabajo de los usuarios, en juegos específicos (fútbol, baloncesto, etc), en los que, a diferencia de una partida entre dos jugadores de ajedrez, no queda claro quién debe almacenar el resultado. Es por esto, que se estudiará la posibilidad de responsabilizar a los administradores de mantener la base de datos actualizada para los juegos en los que se de esta situación.

1.4. Estadísticas (Antonio Checa)

Por último, el sistema debería agregar las estadísticas que pida el usuario. Hay diversos modos de generar una gráfica. El usuario tendrá que indicar al sistema qué modo de salida de los datos quiere (por ejemplo, una gráfica 2D con líneas). Una vez indicado, tendrá que añadir sobre qué atributos del juego quiere la gráfica de datos (por ejemplo, sobre la puntuación de los partidos de los Lakers en cada mes). Después, el sistema debe generar la representación elegida de los datos en caso de que pueda y presentarla ante el usuario. En caso de que no pueda deberá mostrar un mensaje de error mencionando la causa.

Los posibles modos de una gráfica son: gráfico 2D (al menos el eje Y debe ser numérico), gráfico 3D (al menos el eje Y debe ser numérico), columna agrupada (al menos dos elementos numéricos por cada uno no numérico), circular (atributos numéricos positivos).

Al tratar los atributos el usuario debe ser capaz de realizar funciones como la media, la suma o contar datos que cumplan una restricción concreta. Para esto, el sistema debe ser capaz de proporcionar ayuda con todas las funciones que se pueden insertar, y gestionarlas para que funcionen como es debido.

Las estadísticas, datos y gráficas generadas se tendrán que guardar en el sistema, para que el usuario pueda acceder a ellas sin tener que generarlas de nuevo. Habrá acceso a un registro con orden inverso a la fecha de creación para facilitar la búsqueda en el mismo.

Cada modo de salida tiene unas restricciones sobre los atributos, en el caso en el que el usuario incluya unos datos erróneos para una gráfica concreta (por ejemplo, seleccionar el modo de disco para representar datos que no son números). En este caso, el error tendrá que hacer referencia al error concreto, y mostrarlo por pantalla para que el usuario entienda qué ha pasado.

2. Análisis de requisitos

2.1. Primer subsistema, consulta

2.1.1. Requisito de datos

- **RD1.1 Juego**. Se compone de:
  - Un identificador de juego.
- **RD1.2 Partida** de un juego. Se compone de:
  - Un identificador de partida.
- **RD1.3 Partidas del juego**. Es en realidad todos los valores de todos los atributos de una partida, un valor por cada atributo. Cada valor puede ser de cualquier tipo, por ejemplo:
  - Valor númerico.
  - Una cadena de caracteres.
  - Una letra.
  - Una imagen.
  - Un vídeo.
- **RD1.4 Atributo** de una partida. Se encuentra añadido en la parte previa a la gestión de la base de datos.
- **RD1.5 Valor** de un atributo. Puede ser de cualquier tipo, por ejemplo:
  - Valor númerico.
  - Una cadena de caracteres.
  - Una letra.
  - Una imagen.
  - Un vídeo.
- **RD1.6 Atributos** de una partida. Es un conjunto de varios atributos añadidos en la parte previa a la gestión de la base de datos.
- **RD1.7 Valores de un mismo atributo**. Puede ser de cualquier tipo, aunque todos del mismo tipo, por ejemplo:
  - Valor númerico.
  - Una cadena de caracteres.
  - Una letra.
  - Una imagen.
  - Un vídeo.
- **RD1.8 Lista de condiciones**, de las cuales necesitan un valor de atributo, que son:
  - 'sea menor que'.
  - 'sea menor o igual que'.
  - 'sea igual que'.
  - 'sea mayor o igual que'.
  - 'sea mayor que'.
  - 'sea' (comparación entre cadena de caracteres).

- 'no sea' (comparación entre cadena de caracteres).

- **RD1.9 Valores de varios atributos**. Pueden ser de cualquier tipo, por ejemplo:
  - Valor númerico.
  - Una cadena de caracteres.
  - Una letra.
  - Una imagen.
  - Un vídeo.

2.1.2. Requisitos funcionales

- **RF1.1 Consultar partidas**: Consultar todas las partidas de un juego.

  Requisitos de entrada:

  - RD1.1

  Manejamiento de datos: Se lee el identificador del juego.

  - RD1.1

  Salida:

  - RD1.3

- **RF1.2 Restricción por condición**: Sobre un atributo, exigir que cumpla una condición dado un valor. Para facilitar esta función, sólo se puede exigir que se cumpla una condición sobre una lista de condiciones dada. Por ejemplo, Por ejemplo, si existiese un atributo en un juego que sólo tiene como valores Ganado, Empatado, Perdido, poder restringir a aquellas partidas donde dicho atributo sea 'Ganado'; o para el atributo 'Puntuación', poder elegir las partidas donde 'Se haya ganado por más de 10 puntos', donde la condición sobre el atributo 'Puntuación' sería 'es mayor que' y el valor pedido sería '10'.

  Requisitos de entrada:

  - RD1.1
  - RD1.4
  - RD1.8
  - RD1.5

  Manejamiento de datos: Se lee el identificador del juego, el atributo a restringir, la lista de condiciones y el valor para la condición.

  - RD1.1
  - RD1.4
  - RD1.8
  - RD1.5

  Salida:

  - RD1.7

- **RF1.3 Unión de restricciones**: Poder hacer la función 'OR' entre restricciones en una consulta. Por ejemplo, 'todos los partidos en los que ganó o empató el Unicaja' o 'partidos en los que ganó el Unicaja o la diferencia en la puntuación fue menos de 10'.

  Requisitos de entrada:

- RD1.1
- RD1.4
- RD1.8
- RD1.9

Manejamiento de datos: Se lee el identificador del juego, los atributos a realizar la unión de restricciones, la lista de condiciones donde se selecciona una condición por atributo y los valores para cada condición.

- RD1.1
- RD1.6
- RD1.8
- RD1.9

Salida:

- RD1.9

- **RF1.4 Intersección de restricciones**: Poder hacer la función 'AND' entre restricciones en una consulta. Por ejemplo, 'todos los partidos en los que ganó y empató el Unicaja' o 'partidos en los que ganó el Unicaja y la diferencia en la puntuación fue menos de 10'.

  Requisitos de entrada:

  - RD1.1
  - RD1.4
  - RD1.8
  - RD1.9

  Manejamiento de datos: Se lee el identificador del juego, los atributos a realizar la intersección de restricciones, la lista de condiciones donde se selecciona una condición por atributo y los valores para cada condición.

  - RD1.1
  - RD1.6
  - RD1.8
  - RD1.9

  Salida:

  - RD1.9

2.1.3. Restricciones semánticas

RS1.1. Para RF1.2, el valor de RD1.5 que acompaña a las concretas condiciones de 'es menor que', 'es menor o igual que', 'es igual que', 'es mayor o igual que', 'es mayor que' de la lista de condiciones RD1.8 ha de ser numérico.

2.2. Segundo subsistema, consejos

2.2.1. Requisitos de datos

- **RD2.1: Juego** sobre el que se va a aconsejar. Se require un identificador del juego.
- **RD2.2: Atributos** para los que se quiere consejo. Requiere un identificador de cada atributo.

- **RD2.3: Oponente** contra el que optimizar el atributo. Requiere un identificador del oponente.
- **RD2.4: Partidas del juego** almacenadas. Se requerirán los campos que se considere que tienen información relevante para el consejo.
- **RD2.5: Valor de los atributos** optimizado contra el oponente.
- **RD2.6: Juego** sobre el que se va a aconsejar. Se require un identificador del juego.
- **RD2.7: Atributos** para los que se quiere consejo. Requiere un identificador de cada atributo.
- **RD2.8: Valor de los atributos** optimizado contra todos los oponentes almacenados.
- **RD2.9: Juego** sobre el que se va a aconsejar. Se require un identificador del juego.
- **RD2.10: Atributos** para los que se quiere consejo. Requiere un identificador de cada atributo.
- **RD2.11: Valor de los atributos** de un supuesto oponente. Consta de una lista de valores junto con el identificador del atributo al que esta asocido cada uno.
- **RD2.12: Valor de los atributos** optimizado contra un oponente con los atributos especificados.
- **RD2.13: Juego** sobre el que se va a aconsejar. Se require un identificador del juego.
- **RD2.14: Atributos** para los que se quiere consejo. Requiere un identificador de cada atributo.
- **RD2.15: Jugador** del que se obtendrán atributos semejantes. Requiere un identificador del jugador.
- **RD2.16: Valor de los atributos** adaptado para que se parezca al del jugador especificado.

2.2.2. Requisitos funcionales

- **RF2.1: Recomendar atributos contra un oponente:** El usuario selecciona uno o varios atributos del juego y un oponente y el sistema le proporciona valores de los atributos que probablemente sean beneficiosos contra ese oponente.

  Requisitos de entrada: El usuario primero selecciona un juego y después se le presentan los atributos de este y otros jugadores registrados, para que escoja los atributos sobre los que quiere ser aconsejado y el oponenete.

  - RD2.1
  - RD2.2
  - RD2.3

  Manejo de datos: El sistema lee varias partidas registradas en la base de datos, en las que el oponente sea relevante, y hace inferencia sobre ellas para aconsejar buenos atributos.

  - RD2.4

  Salida: Se le presentan al usuario los valores optimizados para cada atributo escogido.

  - RD2.5

- **RF2.2: Recomendar atributos en general:** El usuario selecciona uno o varios atributos del juego y el sistema le proporciona valores de los atributos que probablemente sean beneficiosos en la mayoría de partidas que juegue.

  Requisitos de entrada: El usuario primero selecciona un juego y después se le presentan los atributos para que escoja sobre los que quiere ser aconsejado.

- RD2.6
- RD2.7

Manejo de datos: El sistema lee varias partidas registradas en la base de datos y hace inferencia sobre ellas para aconsejar buenos atributos.

- RD2.4

Salida: Se le presentan al usuario los valores optimizados para cada atributo escogido.

- RD2.8

- **RF2.3: Recomendar atributos con contexto:** El usuario selecciona uno o varios atributos del juego y el valor de uno o varios atributos del adversrio y el sistema proporciona un valor para los atributos seleccionados que probablemente sean beneficiosos contra un adversario con los atributos especificados.

  Requisitos de entrada: El usuario primero selecciona un juego y después se le presentan los atributos para que escoja sobre los que quiere ser aconsejado y los valores de los de un oponente hipotético.

  - RD2.9
  - RD2.10
  - RD2.11

  Manejo de datos: El sistema lee varias partidas registradas en la base de datos, en las que los valores de los atributos del oponente sean relevantes, y hace inferencia sobre ellas para aconsejar buenos atributos.

  - RD2.4

  Salida: Se le presentan al usuario los valores optimizados para cada atributo escogido.

  - RD2.12

- **RF2.4: Recomendar un estilo de juego parecido al de un jugador:** El usuario selecciona uno o varios atributos y el sistema proporciona valores que se asemejan a los que suele tener un jugador concreto.

  Requisitos de entrada:

  - RD2.13
  - RD2.14
  - RD2.15

  Manejo de datos:

  - RD2.4

  Salida:

  - RD2.16

2.2.3. Restricciones semánticas

2.3. Tercer subsistema, inclusión

2.3.1. Requisito de datos

- **RD3.1: Atributos Principales**, proporcionados por el usuario, se componen de:
  - Puntuación de la partida.
  - Nombre del juego
- **RD3.2: Atributos Secundarios**, se componen de:
  - Nombre
  - Valor del atributo

  La lista de atributos secundarios la definirań los administradores del sistema.
- **RD3.3 Identificador de partida** Asignado por el sistema.
- **RD3.4 Comentarios de otros jugadores** Se permitirá asociar un comentario a la partida de otro jugador.
- **RD3.5 Registro de partida** Se compone de:
  - Atributos principales (obligatorio)
  - Atributos secundarios (opcionales)
  - Identificador de partida (obligatorio)
  - Comentario de otros jugadores (opcional)

2.3.2. Requisitos funcionales

- **RF3.1 Inclusión de una partida**: El usuario inserta una pareja que consta de nombre de juego y puntuación, creando un nuevo registro de partida.

  Requisitos de entrada: Nombre del juego y puntuación
  - RD3.1

  Manejo de datos: Se leen los atributos, en caso de que el nombre del juego sea aceptado, se crea el registro.
  - RD3.5

  Salida: Identificador de la partida.
  - RD3.3
- **RF3.2 Modificación de un registro**: Una vez creado, el usuario podrá añadir atributos opcionales al registro.

  Requisitos de entrada: Lista de los atributos a insertar e identifificador de la partida.
  - RD3.3
  - RD3.5

  Manejo de datos: Se leen los atributos seleccionados y se insertan en el registro de la partida en cuestión.

- RD3.2

Salida: A pantalla se muestra una interfaz en la que se representa el registro completo.

- RD3.5

- **RF3.3 Eliminación de un registro** Se permitirá eliminar un resgitro en concreto si poseemos su identificador.

  Requisito de entrada: Identificador de la partida.

  - RD3.3

  Manejo de datos: Se busca el registro y se elimina de la base de datos.

  - RD5.5

  Salida: Mensaje de conformidad.

- **RF3.4 Comentar la partida de otro jugador**

  Requisito de entrada: String con el comentario.

  - RD3.4
  - RD3.3

  Manejo de datos: Se asocia un comentario a la partida de otro jugador

  - RD3.5

  Salida: Mensaje de conformidad

2.3.3. Restricciones semánticas

- **RS3.1 La puntuación de una partida es una pareja (a,b) donde a y b son naturales.**
  - RD3.1
  - RF3.1
- **RS3.2 El identificador de una partida será un entero.**
  - RD3.3
  - RF3.1
  - RF3.3
- **No se permitirá asociar un string vacío a la sección de comentarios de una partida.**
  - RD3.4
  - RF3.4

2.4. Cuarto subsistema, estadísticas

2.4.1. Requisito de datos

- **RD4.1: Atributos de entrada a una gráfica 2D**, proporcionados por el usuario, se componen de:
  - Atributo de la partida sobre la X
  - Atributo de la partida sobre la Y
- **RD4.2: Atributos de una partida**, se componen de:
  - Nombre
  - Valor del atributo
- **RD4.3: Gráfica 2D realizada**, se compone de:
  - Imagen con la gráfica
- **RD4.4: Atributos de entrada a una gráfica 3D**, se componen de:
  - Atributo de la partida sobre la X
  - Atributo de la partida sobre la Y
  - Atributo de la partida sobre la Z
- **RD4.5: Gráfica 3D**, se compone de:
  - Imagen con la gráfica
- **RD4.6: Atributos de entrada a una gráfica de columnas agrupadas**, se componen de:
  - Atributo de la partida sobre la X
  - Atributo de primer tipo de columna
  - Atributo del segundo tipo de columna
- **RD4.7: Gráfica de columnas agrupadas**, se compone de:
  - Imagen con la gráfica
- **RD4.8: Atributos de entrada a una gráfica circular**, se componen de:
  - Atributo sobre el que se hace el círculo
- **RD4.9: Gráfica circular**, se compone de:
  - Imagen con la gráfica
- **RD4.10: Atributo y función elemental**, se compone de:
  - Atributo sobre el que se hace la función
  - Función, a elegir, entre media, varianza, suma, contar, máximo o mínimo
- **RD4.11: Resultado de la función**, se compone de:
  - Resultado de la operación sobre el atributo
- **RD4.12: Petición de conseguir las últimas gráficas**, se compone de:
  - Activación de petición de la gráfica (pulsación de un botón)
- **RD4.13: Gráficas anteriores**, se compone de:

- Imágenes de gráficas

- **RD4.14: Gráficas anteriores como salida**, se compone de:
  - Imágenes de gráficas

2.4.2. Requisitos funcionales

- **RF4.1 Realizar gráfica 2D de unos atributos**: El usuario selecciona dos atributos, de los cuales se realiza una gráfica 2D

  Requisitos de entrada: Elegir qué dos atributos se escogen

  - RD4.1

  Manejo de datos: Se leen los dos atributos seleccionados y se inserta la gráfica en la base de datos

  - RD4.2

  Salida: Gráfica 2D

  - RD4.3

- **RF4.2 Realizar gráfica 3D de unos atributos**: El usuario selecciona tres atributos, de los cuales se realiza una gráfica 3D

  Requisitos de entrada: Elegir qué tres atributos se escogen

  - RD4.4

  Manejo de datos: Se leen los tres atributos seleccionados y se inserta la gráfica en la base de datos

  - RD4.2

  Salida: Gráfica 3D

  - RD4.5

- **RF4.3 Realizar gráfica de columnas agrupadas de unos atributos**: El usuario selecciona dos atributos para las columnas y un tercero para el eje de abscisas, de los cuales se realiza la gráfica

  Requisitos de entrada: Elegir qué atributos se escogen

  - RD4.6

  Manejo de datos: Se leen los atributos seleccionados y se inserta la gráfica en la base de datos.

  - RD4.2

  Salida: Gráfica

  - RD4.7

- **RF4.4 Realizar gráfica circular de unos atributos**: El usuario selecciona un atributo, del cual se hace una gráfica porcentual en forma de círculo

  Requisitos de entrada: Elegir qué atributos se escoge.

  - RD4.8

  Manejo de datos: Se lee el atributo seleccionado y se inserta la gráfica en la base de datos.

- RD4.2

Salida: Gráfica.

- RD4.9

- **RF4.5 Realizar una función elemental sobre un atributo concreto**: El usuario selecciona una función elemental (detalladas en la entrada de este requisito) y un atributo sobre el que operar, y se devuelve el resultado de la operación.

  Requisitos de entrada: El atributo y la función, a elegir entre media, varianza, suma, contar, máximo o mínimo.

  - RD4.10

  Manejo de datos: Se lee el atributo seleccionado y se opera.

  - RD4.2

  Salida: Resultado de la función.

  - RD4.11

- **RF4.6 Conseguir las últimas gráficas pedidas**: El usuario le da a un botón para ver las últimas gráficas que ha pedido.

  Requisitos de entrada: La activación de un botón.

  - RD4.12

  Manejo de datos: Se cogen las gráficas realizadas en función de su fecha.

  - RD4.13

  Salida: Gráficas, ordenadas de más a menos recientes.

  - RD4.14

2.4.3. Restricciones semánticas

- **RS4.1 El atributo sobre la Y de RD4.1 sobre el requisito funcional de hacer una gráfica 2D RF4.1 debe ser de tipo numérico.**
- **RS4.2 El atributo sobre la Z de RD4.4 sobre el requisito funcional de hacer una gráfica 3D RF4.2 debe ser de tipo numérico.**
- **RS4.3 El atributo de los tipos de columnas en RD4.6 sobre el requisito funcional de hacer una gráfica de columnas agrupadas en RF4.3 deben ser de tipo numérico.**
- **RS4.4 El atributo de entrada a una gráfica circular en RD4.8 sobre el requisito funcional RF4.4 debe ser de tipo numérico.**
- **RS4.5 Los atributos de entrada a una función elemental en RD4.10 sobre el requisito funcional RF4.5 deben ser de tipo numérico.**