

1. DIAGRAMA DE CASOS DE USO

1.1. Identificador Caso de Uso.

En este proyecto trataremos de mantener un registro y un seguimiento de las puntuaciones obtenidas por cada equipo en los diferentes debates. Para ello, necesitaremos guardar información de los mismos, así como los participantes de los propios equipos.

1.2. Título del Caso de Uso.

En este caso, el título será “Gestión y Administración de Puntuaciones por Debate”.

1.3.Descripción.

El primer punto de nuestro caso de uso abarca el registro de las puntuaciones obtenidas por los diferentes equipos. En este punto necesitaremos guardar datos del debate en el que los equipos se enfrentan como las observaciones, penalizaciones, puntuación por equipo y equipo ganador. Igualmente, necesitaremos listar los participantes de cada equipo enfrentado, para ello, utilizaremos una lista de participantes que asignaremos o añadiremos a cada equipo.

Luego, seguiremos con la búsqueda de los resultados de los debates por fase. Para ello, se le pasarán por parámetro los criterios de búsqueda, es decir, lo que queremos buscar. Luego de esto, procederemos a buscarlos dentro de la lista de debates que tengamos registrados. Por último, al igual que en el anterior apartado, será necesario listar los participantes de los equipos que se enfrentan.

Por último, se tendrá en cuenta las posibles modificaciones que se quieran realizar en cuanto a un debate. Para poder modificarlo, se darán diferentes opciones como modificar la puntuación del equipo A, modificar la puntuación del equipo B, modificar las observaciones del debate o modificar el equipo ganador o el orador. Cada una de estas opciones pedirá los datos que requieren para hacer dichas modificaciones.

1.4.Precondición.

Antes de empezar mi parte será necesario que todos los equipos estén completos y con su respectiva información.

1.5.Requisitos relacionados.

- Será necesario tener el alta de los diferentes participantes de los equipos que se pueden enfrentar, de forma que queden correctamente añadidos a los equipos.
- Será necesario que los equipos estén compensados, es decir, que tengan el mismo número de participantes.
- Será necesario que un participante no esté repetido en varios equipos. De esta forma, ahorramos enfrentamientos del mismo participante en diferentes equipos.

- Será necesario que la información de los equipos esté completa para evitar posibles problemas en la ejecución del código.

1.6.Secuencia de pasos.

Si se desea registrar puntuaciones de un equipo, empezaremos preguntando la fase, luego identificando la fase en la que nos encontramos podemos visualizar los posibles cruces entre equipos de la misma fase. Seguimos modificando la información de cada equipo insertando las puntuaciones conseguidas en los debates así como las observaciones y las penalizaciones. Compararemos la puntuación de ambos equipos y de ahí seleccionamos el equipo ganador.

El mejor orador será elegido en mi caso definiendo una serie de puntos conseguidos por cada participante para el equipo. (SUMA DE PUNTOS PARTICIPANTES = PUNTOS EQUIPO). El participante con mayor puntos conseguidos para el equipo será el mejor orador del mismo para dicho debate.

Por último, listaremos por pantalla los participantes que componen los equipos enfrentados. Para ello, recorreremos la lista e imprimimos por pantalla los diferentes jugadores que la componen.

En el siguiente punto, recorreremos las diferentes listas donde guardamos los debates e imprimimos por pantalla el resultado de los mismos. En caso de querer buscar algún dato específico, lo pasaremos por parámetro y veremos si la lista de debates contiene dicho dato.

El último punto consiste en realizar modificaciones en la información de los debates. Para ello crearemos un menú en el que preguntemos qué información queremos consultar o modificar. Después, preguntaremos si se desea consultar o si se desea modificar. Más tarde, insertamos el dato que requiere ser modificado o simplemente imprimimos por pantalla la consulta del mismo.

1.7.Postcondición.

Al acabar mi parte, será necesario que las pruebas estén insertadas correctamente para después hacer los cruces de los diferentes equipos de la fase correspondiente. Luego, el ejercicio se basa en listar la diferente información que se crea al definir los diferentes métodos.

1.8.Excepciones.

Entre las posibles excepciones encontramos que intentemos enfrentar a un equipo contra otro al que ya se ha enfrentado anteriormente, por lo que al solo haber una posible opción de enfrentamiento, no sería posible. Podríamos controlarlo mediante una excepción o mediante una revisión continua de la lista que guarda los debates para asegurarnos de que ese debate no existe con anterioridad antes de añadirlo a la lista.

También podría saltarnos una excepción en el caso de que quieras buscar el resultado de un enfrentamiento que no se ha realizado. Para ello, tendríamos que volver a revisar continuamente la lista de nuevo.

Por último, podríamos querer consultar un dato de un equipo y que este esté vacío. Por ello, deberíamos asegurarnos de que introducimos todos los datos de los equipos que se enfrentan, es decir, que los datos de un mismo equipo no puedan ser nulos.

2. Modelo de datos. Definición de entidades y relaciones.

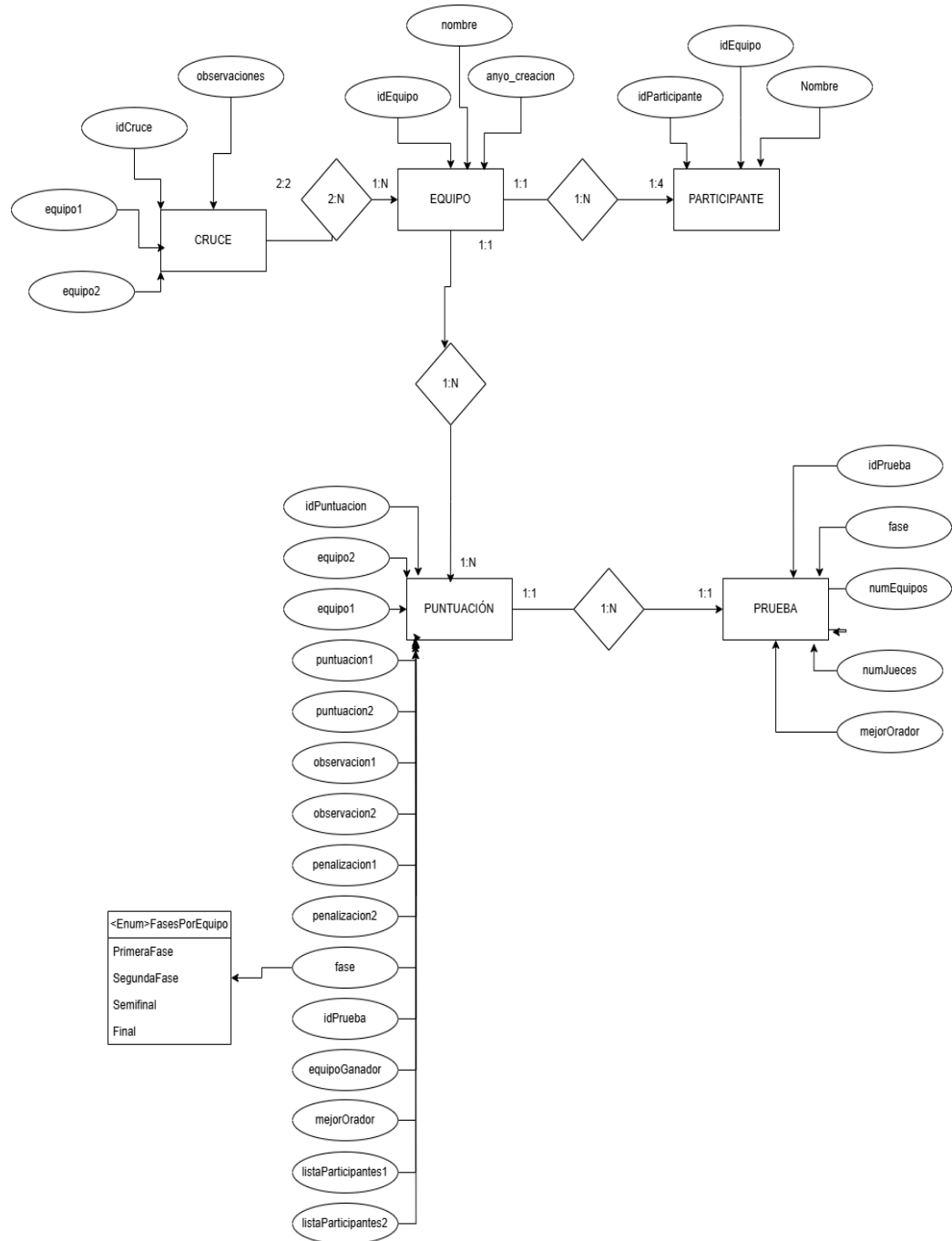
En este caso, para la parte que me corresponde, definiremos las siguientes entidades: Equipo, Participante, Puntuaciones, Prueba, Fase, Cruce y Debate. La clase Equipo y Participante no corresponde a la parte que he de realizar pero una vez conectada la base de datos con el código es necesario tener objetos contruidos de dichas clases.

Antes de empezar a definir las clases, es importante recalcar que todos los ids que tengamos como atributos serán autoincrementales. También es necesario tener en cuenta que todas las clases mencionadas tendrán una clave única que controlará la creación de objetos repetidos.

Pasamos a la definición de dichas clases/tablas:

- Equipo: tiene como atributos idequipo, nombre y año de creación.
- Participante: sus atributos son idparticipante, nombre e idequipo. Contiene una clave foránea que relaciona el idequipo con el atributo idequipo de la clase Equipo.
- Puntuaciones: contiene los atributos idpuntuacion, idequipo1, idequipo2, puntuacion1, puntuacion2, observacion1, observacion2, penalizacion1, penalizacion2, fase, idprueba, equipoGanador, mejorOrador, listaParticipantes1 y listaParticipantes2. Al igual que la clase anterior, cuenta con dos claves foráneas que relaciona los ids de los equipos con los mismos atributos en la clase Equipo.
- Prueba: tiene como atributos idprueba, idfase, numequipos, numjueces y mejorador.
- Fase: esta clase en realidad sería un enum que contiene las diferentes fases que entre todos hemos definido para este proyecto que son PrimeraFase, SegundaFase, Semifinal y Final.
- Cruce: los atributos de esta clase son idcruce, idequipo1, idequipo2 y observaciones. Igual que las clases anteriores, tiene dos claves foráneas en los identificadores de los equipos haciendo referencia a los identificadores de los equipos de la clase Equipo.
- Debate: sus atributos son iddebate, idequipo1 e idequipo2. Mantiene las mismas claves foráneas que las clases anteriores.

Por último, dejo por aquí el uml de dicho modelo de datos:



Link figma:

<https://www.figma.com/design/cgYlOuBWlvhEs44AuFlqb2/EvaPerez-ProyectoTransversal?node-id=0-1&t=4x6AenDjUpqlssAY-1>