Proyecto Inter modular

Liga de hockey

David Borja Mateo
Desarrollo Aplicaciones multiplataforma 1º
Curso 2024/2025

Proyecto Intermodular liga de hockey

Contenido

Proyecto Intermodular liga de hockey

1. Entrevista y Requisitos:

1.1. Entrevista:

En la entrevista con el cliente nos comunica que quiere que hagamos una aplicación para la gestión de su liga de hockey las principales preguntas de la entrevista son:

- ¿Qué se busca que haga la aplicación?
- ¿La aplicación debe implementar una interfaz gráfica?
- ¿Qué datos quiere que guarde la aplicación?
- ¿Qué funciones debe de tener?
- ¿Quién tendrá acceso a la aplicación?
- ¿Restricciones de los datos y valores por defecto?

1.2. Requisitos:

Después de la entrevista obtenemos los siguientes requisitos

1. La base de datos a usar guardara datos y sus respectivas relaciones de:
   - Jugadores (ISA persona)
   - Entrenadores (ISA persona)
   - Equipo
   - Partidos
   - Arbitro
   - Informe (partido)
   - Historial (arbitro)
2. El programa debe implementar una interfaz gráfica con un login que tenga un usuario por defecto llamado “ADMIN” y con contraseña por defecto “123456”. El administrador podrá modificar esta contraseña. Esta modificación deberá de cumplir los siguientes requisitos
   - Mínimo 5 y máximo 10 caracteres de duración
   - Tener una letra mayúscula
   - Tener dos números
3. El programa estará administrado por una única persona encargada de la modificación, eliminación e inserción de datos.
4. En el programa solo se podrá modificar el equipo al que pertenece un jugador si este es transferible.
5. El programa tendrá un log donde se almacenarán los inicios de sesión.
6. El programa tendrá un archivo donde se almacenará la URL, usuario y contraseña de la base de datos. Este será el archivo a usar para que el programa haga la conexión.
7. El equipo local y visitante de un partido no puede ser el mismo.
8. Un entrenador solo puede entrenar un equipo, así como un jugador solo puede jugar en un equipo.
9. Habrá un árbitro jefe que será el jefe del resto de árbitros.
10. Cada arbitro solo puede tener un informe que será actualizado por el administrador de la aplicación.
11. Por defecto un jugador al ser añadido no será transferible
12. Por defecto el palmarés del entrenador (Nº trofeos) sera 0.

13. Habrá otro usuario llamado Arbitro que solo podrá visualizar los datos, no podrá añadir borrar o modificar los datos que tiene la base este usuario tendrá una contraseña por defecto “123456” que no podrá ser modificada

2. Enunciado base de datos

Una liga de hockey nos encarga el diseño de una base de datos para que después esta pueda ser usada en una aplicación que también programaremos la información que se nos proporciona de la base de datos es

1. Una tabla persona que almacenara
   - DNI
   - Dirección
   - Teléfono
   - Apellidos
   - Nombre
   - Fecha de nacimiento
2. Además, se nos informa que una persona podrá ser
   2.1. Jugador que también guardara
      - Valor de mercado
      - Posición
      - Dorsal
      - Salario
      - Transferible
      - Equipo (id del equipo)

   2.2. Entrenador que también guardara
      - Años experiencia
      - Palmares
      - Numero equipos entrenados
3. Una tabla equipo que almacenara
   - Fundación
   - Nombre
   - Palmares
   - ID equipo
   - Entrenador (DNI Entrenador)
4. Una tabla partidos que almacenara
   - Id_partido
   - Fecha
   - Arbitro (Id de arbitro)
   - Ganador
   - Además, queremos almacenar los dos equipos que han jugado, su rol (local o visitante)
5. Una tabla arbitro Que guardara
   - Id
   - Nombre
   - Si el árbitro supervisa o no a otros árbitros(Es jefe o no)
6. Una tabla informe que almacena
   - Id_arbitro

Proyecto Intermodular liga de hockey

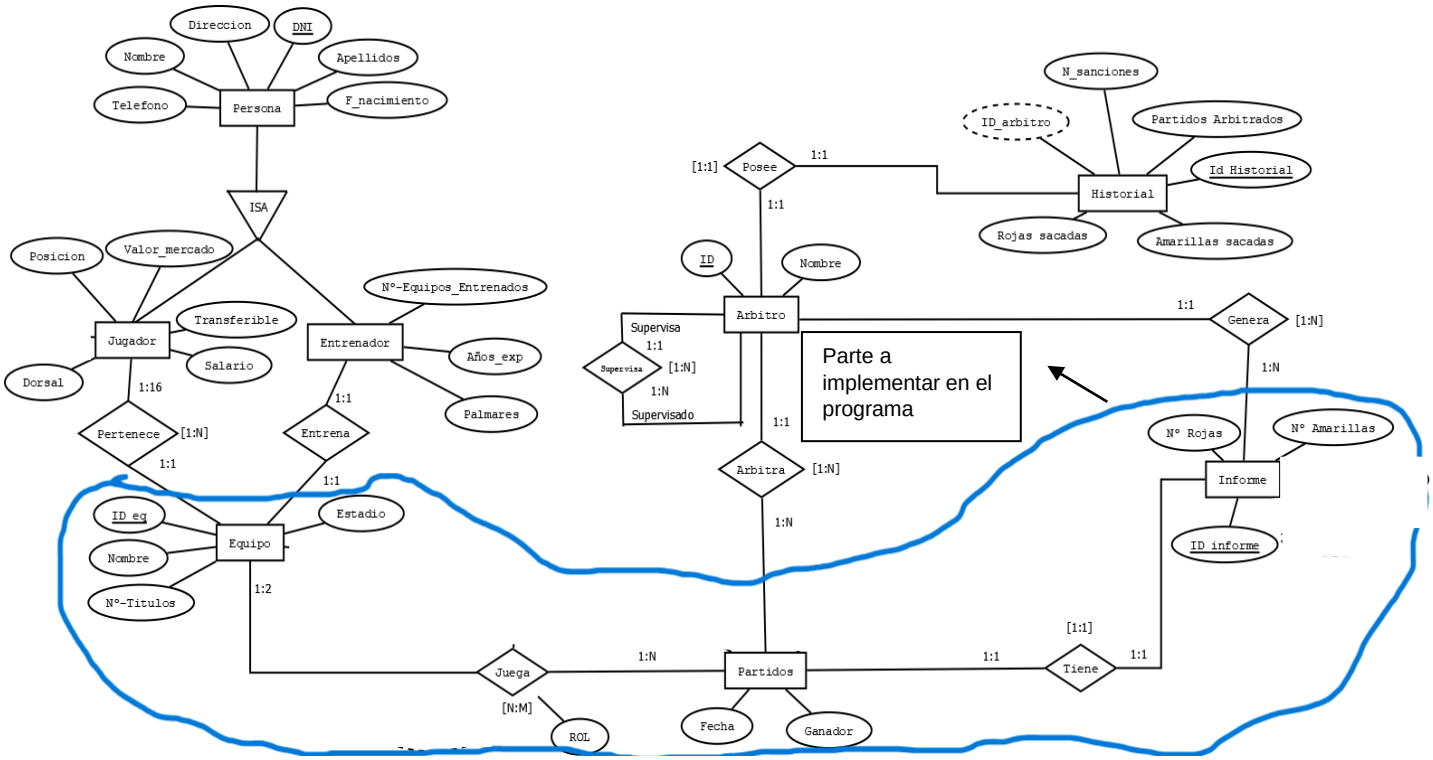
- Id_partido
- Id_informe
- Numero rojas
- Numero amarillas

7. Una tabla historial que guarda
    - Partidos arbitrados (cantidad)
    - Id_arbitro
    - Id_historial
    - Rojas Sacadas
    - Amarillas sacadas
    - Sanciones o denuncias (Cantidad)

Además, también se nos da la siguiente información

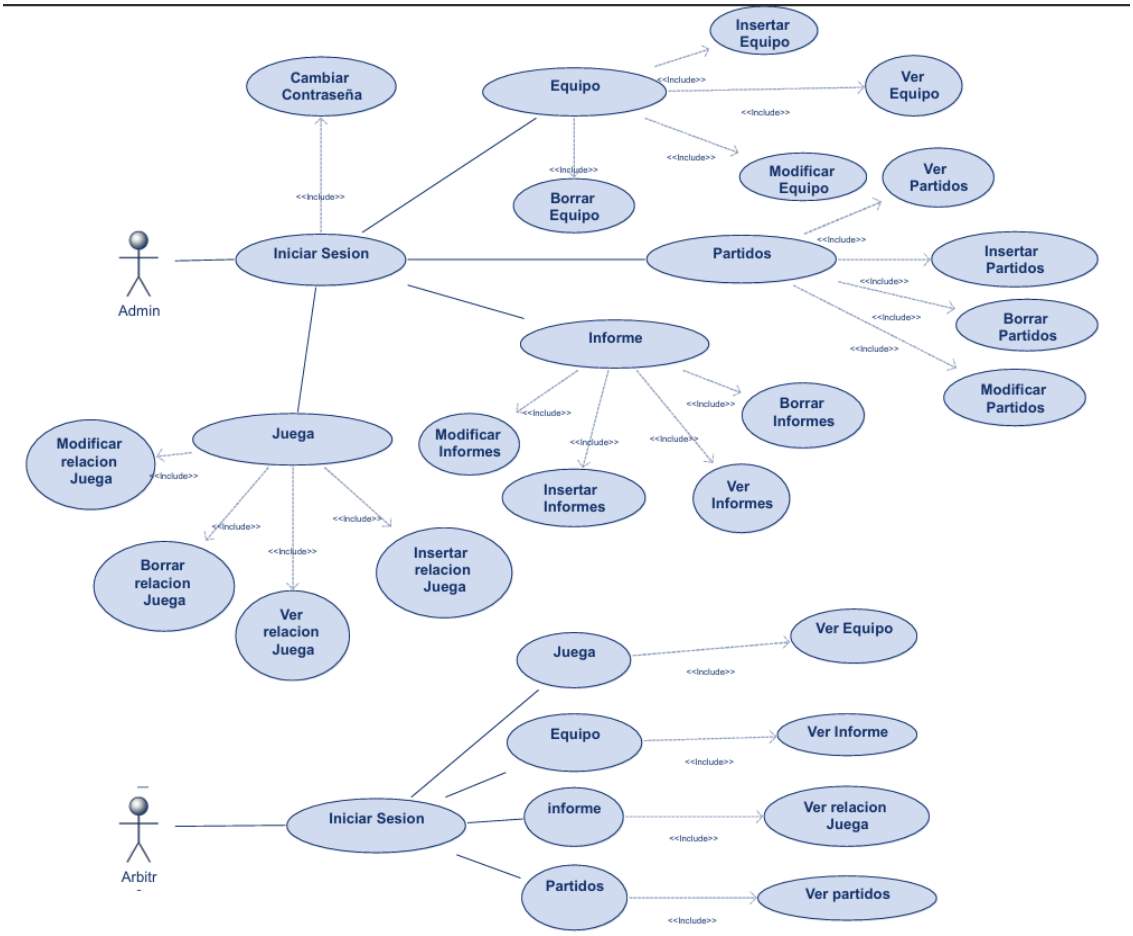
1. Un partido es jugado por dos equipos y los equipos juegan 38 partidos (Relación N:M)
2. Un jugador pertenece a un equipo y un entrenador entrena un equipo
3. El equipo Juega partidos, estos partidos son arbitrados por un árbitro que además también genera un informe de ese partido
4. Arbitro posee un historial donde almacenamos datos de su carrera

3. Entidad/Relación



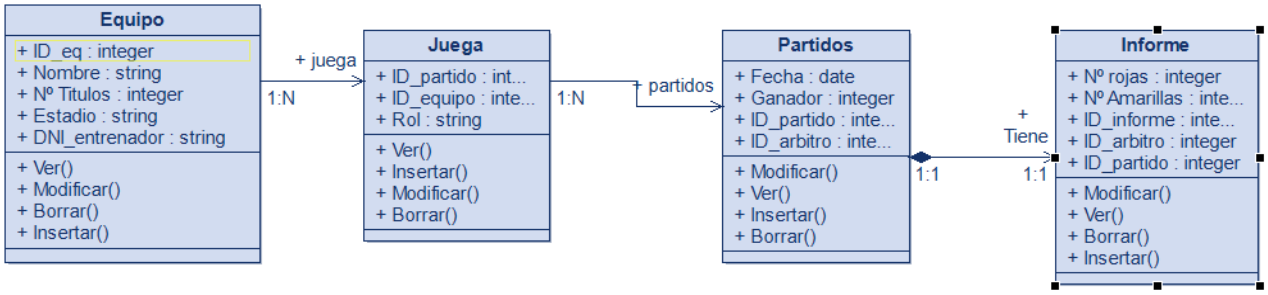
Proyecto Intermodular liga de hockey

4. Casos de uso



5. Diagrama de clases

Solo parte que implementare en el programa



Proyecto Intermodular liga de hockey

6. Paso a tablas de la base de datos:

1. Persona(DNI:VARCHAR(15), Nombre:VARCHAR(50), Apellidos:VARCHAR(50), Dirección:VARCHAR(100), Teléfono:VARCHAR(20), F_nacimiento:DATE)
2. Jugador(DNI:VARCHAR(15), ID_eq:INT, Posición:VARCHAR(30), Valor_mercado:DECIMAL(10,2), Dorsal:INT, Transferible:BOOLEAN, Salario:DECIMAL(10,2))
    a. Referencias: DNI → Persona(DNI), ID_eq → Equipo(ID_eq)
3. Entrenador(DNI:VARCHAR(15), Nº_Equipos_Entrenados:INT, Años_exp:INT, Palmarés:TEXT)
    a. Referencias: DNI → Persona(DNI)
4. Equipo(ID_eq:INT, Nombre:VARCHAR(50), Nº_Títulos:INT, Estadio:VARCHAR(100), DNI_entrenador:VARCHAR(15))
    a. Referencias: DNI_entrenador → Entrenador(DNI)
5. Árbitro(ID:INT, Nombre:VARCHAR(50), Supervisa:INT, Supervisado:INT)
6. Partidos(ID_partido:INT, Fecha:DATE, Ganador:INT)
    a. Referencias: Ganador → Equipo(ID_eq)
7. Historial(Id_Historial:INT, ID_arbitro:INT, Nº_sanciones:INT, Partidos_Arbitrados:INT, Rojas_sacadas:INT, Amarillas_sacadas:INT)
    a. Referencias: ID_arbitro → Árbitro(ID)
8. Informe(ID_informe:INT, Nº_Rojas:INT, Nº_Amarillas:INT, Id_arbitro:INT, Id_partido:INT)
    a. Referencias: Id_arbitro → Árbitro(ID), Id_partido → Partidos(ID_partido)
9. Arbitraje(ID_arbitro:INT, ID_partido:INT)
    a. Referencias: ID_arbitro → Árbitro(ID), ID_partido → Partidos(ID_partido)

Proyecto Intermodular liga de hockey

7. Normalizacion de la base de datos:

FN1: Eliminación de atributos compuestos y valores multivaluados. La cumple

FN2: Eliminación de dependencias parciales.
Entrenador(DNI:VARCHAR(15), Nombre:VARCHAR(50), Nº_Equipos_Entrenados:INT, Años_exp:INT, Palmarés:TEXT)
Referencias: DNI → Persona(DNI)
Equipo(ID_eq:INT, Nombre:VARCHAR(50), Nº_Títulos:INT, Estadio:VARCHAR(100), DNI_entrenador:VARCHAR(15))
Referencias: DNI_entrenador → Entrenador(DNI)

FN3: Eliminación de dependencias transitivas.
Supervision(ID_arbitro:INT, ID_partido:INT, Supervisado:BOOLEAN)
Referencias: ID_arbitro → Árbitro(ID), ID_partido → Partidos(ID_partido)

BASE FINAL

- Persona(DNI:VARCHAR(15), Nombre:VARCHAR(50), Apellidos:VARCHAR(50), Dirección:VARCHAR(100), Teléfono:VARCHAR(20), F_nacimiento:DATE)
- Jugador(DNI:VARCHAR(15), ID_eq:INT, Posición:VARCHAR(30), Valor_mercado:DECIMAL(10,2), Dorsal:INT, Transferible:BOOLEAN, Salario:DECIMAL(10,2))
  Referencias: DNI → Persona(DNI), ID_eq → Equipo(ID_eq)
- Entrenador(DNI:VARCHAR(15), Nombre:VARCHAR(50), Nº_Equipos_Entrenados:INT, Años_exp:INT, Palmarés:TEXT)
  Referencias: DNI → Persona(DNI)
- Equipo(ID_eq:INT, Nombre:VARCHAR(50), Nº_Títulos:INT, Estadio:VARCHAR(100), DNI_entrenador:VARCHAR(15))
  Referencias: DNI_entrenador → Entrenador(DNI)
- Árbitro(ID:INT, Nombre:VARCHAR(50))
  Referencias: Ninguna
- Partidos(ID_partido:INT, Fecha:DATE, Ganador:INT)
  Referencias: Ganador → Equipo(ID_eq)
- Historial(Id_Historial:INT, ID_arbitro:INT, Nº_sanciones:INT, Partidos_Arbitrados:INT, Rojas_sacadas:INT, Amarillas_sacadas:INT)
  Referencias: ID_arbitro → Árbitro(ID)
- Informe(ID_informe:INT, Nº_Rojas:INT, Nº_Amarillas:INT, Id_arbitro:INT, Id_partido:INT)
- Referencias: Id_arbitro → Árbitro(ID), Id_partido → Partidos(ID_partido)
- Arbitraje(ID_arbitro:INT, ID_partido:INT)
  Referencias: ID_arbitro → Árbitro(ID), ID_partido → Partidos(ID_partido)
- Supervision(ID_arbitro:INT, ID_partido:INT, Supervisado:BOOLEAN)
  Referencias: ID_arbitro → Árbitro(ID), ID_partido → Partidos(ID_partido)

Proyecto Intermodular liga de hockey

8. Consultas de la base de datos

1. Consulta en la que utilices el operador AND, LIKE y la cláusula ORDER BY.

```
SELECT Nombre, Apellidos, Direccion FROM Persona
WHERE Nombre LIKE 'J%' AND Direccion LIKE '%Plaza España%'
ORDER BY Apellidos ASC;
```

2. Consulta en la que uses el operador IN junto con una subconsulta.

```
SELECT Nombre, Apellidos FROM Persona
WHERE DNI IN (SELECT DNI FROM Jugador WHERE Valor_mercado > 5000);
```

3. Consulta en la que uses el operador ALL junto con una subconsulta.

```
SELECT DNI, Dorsal FROM Jugador
WHERE Salario > ALL (SELECT Salario FROM Jugador WHERE Posicion = 'Defensa');
```

4. Consulta en la que utilices tres subconsultas, una dentro de otra.

```
SELECT DNI, Nombre FROM Persona
WHERE DNI IN (SELECT DNI FROM Jugador WHERE ID_eq IN (
SELECT ID_eq FROM Equipo WHERE N_Titulos = (
SELECT MAX(N_Titulos) FROM Equipo)));
```

5. Consulta en la que sea necesario concatenar tres tablas usando JOIN y además haya una condición en un WHERE.

```
SELECT Persona.Nombre, Equipo.Nombre, Partidos.Fecha FROM Persona
JOIN Jugador ON Persona.DNI = Jugador.DNI
JOIN Equipo ON Jugador.ID_eq = Equipo.ID_eq
JOIN Partidos ON Equipo.ID_eq = Partidos.Ganador
WHERE Partidos.ID_arbitro =1;
```

6. Consulta en la que utilices la función COUNT, la cláusula WHERE y la operación JOIN.

```
SELECT Equipo.Nombre, COUNT(Jugador.DNI) AS Cantidad_Jugadores FROM
Equipo
JOIN Jugador ON Equipo.ID_eq = Jugador.ID_eq
WHERE Equipo.N_Titulos > 5
GROUP BY Equipo.Nombre;
```

7. Consulta en la que utilices la cláusula GROUP BY y la cláusula WHERE

```
SELECT ID_arbitro, COUNT(ID_partido) AS Partidos_Arbitrados FROM Partidos
WHERE ID_arbitro IN (SELECT ID FROM Arbitro WHERE Nombre LIKE 'M%')
GROUP BY ID_arbitro;
```

8. Consulta en la que uses las cláusulas GROUP BY y HAVING.

```
SELECT ID_eq, AVG(Salario) AS Salario_Promedio FROM Jugador
GROUP BY ID_eq
HAVING AVG(Salario) > 2000000;
```

9. Consulta en la que uses las cláusulas WHERE, HAVING y una subconsulta.

```
SELECT ID_eq, COUNT(DNI) AS Total_Jugadores FROM Jugador
WHERE ID_eq IN (SELECT ID_eq FROM Equipo WHERE N_Titulos > 1)
GROUP BY ID_eq
HAVING COUNT(DNI) > 1;
```

10. Consulta en la que utilices uno de los operadores de conjuntos (EXCEPT, INTERSECT o EXCEPT).

```
SELECT DNI FROM Jugador
INTERSECT
```

```
SELECT DNI FROM Entrenador;
```

9. Operaciones de inserción, modificación y eliminación

1. Inserción de datos a partir del resultado de una consulta.

   Inserto en la tabla 'Historial' una fila para cada árbitro cuyo nombre comienza con 'A' =

   ```
   INSERT INTO Historial (Id_Historial, ID_arbitro, N_sanciones, Partidos_Arbitrados, Rojas_sacadas, Amarillas_sacadas)
   SELECT ID, ID, 0, 0, 0, 0 FROM Arbitro
   WHERE Nombre LIKE 'A%';
   ```

2. Modificación de datos utilizando la cláusula WHERE junto con una subconsulta.

   Aumento un 10% el salario de los jugadores que pertenecen a equipos cuyo nombre empieza por 'CH'=

   ```
   UPDATE Jugador SET Salario = Salario * 1.1 WHERE ID_eq IN (SELECT ID_eq FROM Equipo WHERE Nombre LIKE 'CH%');
   ```

3. Modificación de datos, de forma que el nuevo valor se obtenga mediante una subconsulta.

   Hago que todos los equipos tengan el máximo valor de títulos de su misma tabla=

   ```
   UPDATE Equipo AS E SET E.N_Titulos = (SELECT MAX(T.Max_Titulos) FROM (SELECT N_Titulos AS Max_Titulos FROM Equipo) AS T);
   ```

4. Modificar o eliminar datos utilizando una consulta correlacionada.

   Elimino de la tabla Jugador jugadores cuyo salario es menor que el salario promedio de su equipo=

   ```
   DELETE J FROM Jugador AS J JOIN (SELECT ID_eq, AVG(Salario) AS SalarioPromedio FROM Jugador GROUP BY ID_eq) AS subconsulta ON J.ID_eq = Subconsulta.ID_eq WHERE J.Salario < Subconsulta.SalarioPromedio;
   ```

10. Aspectos a tener en cuenta a la hora de ejecutar el proyecto.

1. Configuración de la conexión a la base de datos

- Verificar que el archivo Informacion_conexion.txt contiene la URL, usuario y contraseña correctos.
- Confirmar que los valores se leen correctamente antes de ejecutar cualquier consulta.

2. Control de excepciones en acceso a la base de datos

- Manejar errores en la conexión, ejecución de consultas y preparación de sentencias para evitar fallos inesperados.
- Cerrar correctamente ResultSet, PreparedStatement y Statement para liberar recursos.

3. Funcionamiento de la interfaz gráfica

- Configurar correctamente los tamaños y disposición (LayoutManager) de las ventanas (JFrame).
- Definir el cierre adecuado (DISPOSE_ON_CLOSE) para evitar que la aplicación se cierre por completo.

- Asegurar que los JComboBox y JTable cargan correctamente los datos desde la base de datos.

4. Carga y actualización de datos en las tablas

- Limpiar las tablas antes de insertar nuevos datos (setRowCount(0)) para evitar duplicados.
- Actualizar los datos después de una inserción, eliminación o modificación.

5. Validación de entradas de usuario

- Verificar que los valores ingresados sean correctos (ejemplo: números enteros en campos numéricos).
- Evitar entradas vacías o incorrectas en los formularios (JTextField, JComboBox).

6. Integridad de las relaciones entre tablas

- Antes de eliminar un equipo, comprobar que no tenga dependencias en otras tablas (Juega, Informe, etc.).
- Prevenir la asignación de dos equipos con el mismo rol (Local o Visitante) en un partido.

7. Gestión de mensajes y retroalimentación al usuario

- Usar JOptionPane.showMessageDialog() para confirmar acciones correctas o advertir sobre errores.
- Informar al usuario cuando una acción no se puede completar por restricciones de datos.

8. Manejo de fechas y formatos

- Utilizar el formato adecuado (yyyy-MM-dd) al manejar fechas con JDateChooser.
- Validar que las fechas ingresadas sean correctas antes de realizar inserciones en la base de datos.

9. Liberación de recursos después de consultas SQL

- Cerrar adecuadamente ResultSet, PreparedStatement y Statement para evitar fugas de memoria.
- Garantizar que las conexiones a la base de datos no queden abiertas innecesariamente.

10. Ordenamiento y presentación de datos

- Ordenar listas (Collections.sort()) antes de mostrarlas en interfaces gráficas para mejorar la experiencia de usuario.
- Asegurar que los JTable presenten datos correctamente alineados y estructurados.

11. Problemas o dificultades encontradas durante el desarrollo.

- Errores de conexión a la base de datos: configuración incorrecta de la URL.
- Carga incorrecta de datos: problemas con JTable al cargarle datos
- Fallos en la interfaz gráfica: diseño desordenado, JComboBox sin datos.

Proyecto Intermodular liga de hockey

- Validación deficiente de datos: fechas incorrectas, campos vacíos no detectados.
- Conflictos al modificar/eliminar registros: restricciones que impiden cambios en equipos o partidos.
- Consultas SQL poco eficientes: consultas que tenían demasioadas subconsultas que acabe simplificando para tener mejor legibilidad y ahorrame mirar datos en tablas que no necesitaba
- Errores inesperados en ejecución: excepciones no controladas que hacían que el programa finalizara.