



INSTITUTO TECNOLÓGICO DE COSTA RICA

INGENIERÍA EN COMPUTACIÓN

BASES DE DATOS I, GRUPO 1

Proyecto #2

CAMPEONATO MUNDIAL DE FÚTBOL

Documentación del proyecto

Realizado por:

JEREMY LIVE GONZÁLEZ, 2015068507

JURGENN MORALES JIMÉNEZ, 2016145484

Profesor:

WILLIAM MATA RODRIGUEZ

II semestre, 2017

Tabla de Contenido

Enunciado del proyecto	2
Temas investigados (De qué trata, cómo se usa)	15
Software de control de versiones	17
Diseño conceptual de la base de datos	18
Diseño lógico de la base de datos	19
Diseño físico de la base de datos	21
Programas almacenados: procedimientos, funciones, triggers	22
Transacciones, vistas, índices	23
Conclusiones del trabajo (Problemas y soluciones a los mismos):	26
Rúbrica de evaluación y análisis de resultados	27

Enunciado del proyecto

PROYECTO 2

CAMPEONATO MUNDIAL DE FÚTBOL

El proyecto consiste en desarrollar un software que contenga información de un campeonato mundial de fútbol.

El desarrollo del proyecto es en equipos de 3 estudiantes máximo, uno de ellos lo deben nombrar como coordinador. Importante: las experiencias han demostrado que los proyectos en equipos que no han sido administrados adecuadamente van a fallar, así que en cuanto noten que se presentan problemas al respecto de inmediato trátenlo primeramente con los miembros del equipo, y de no resolver lo comunican al profesor. Cualquier comunicación al profesor que vaya copiada a todos los miembros del equipo.

El DBMS (SABD) a usar es Oracle XE (Express Edition). Las otras herramientas de desarrollo son definidas por Ustedes garantizando que puedan cumplir con los requerimientos definidos en el proyecto.

Buenas prácticas de la ingeniería de software: deben usar un software para administrar el desarrollo de proyectos en equipo el cual incluye un control de versiones (Git, otros).

REQUERIMIENTOS DEL PROYECTO

En el campeonato mundial participan 32 equipos que se clasificaron de etapas previas de competición. El campeonato se divide en varias fases.

Fase de grupos

Los 32 equipos se dividen en 8 grupos de 4 equipos cada uno. Dentro de cada grupo cada equipo juega tres partidos, uno contra cada uno de los demás miembros del grupo dando un total de 48 partidos en toda la fase de grupos. Según el resultado de cada partido, se

otorgan tres puntos al ganador y ninguno al perdedor, en caso de empate, se otorga un punto a cada equipo.

Pasan a la siguiente ronda los dos equipos de cada grupo mejor clasificados. El orden de clasificación se determina teniendo en cuenta los siguientes criterios en este orden de preferencia:

- a. El mayor número de puntos obtenidos teniendo en cuenta todos los partidos del grupo
- b. La mayor diferencia de goles sumados teniendo en cuenta todos los partidos del grupo
- c. El mayor número de goles a favor anotados teniendo en cuenta todos los partidos del grupo

Si dos o más equipos quedan igualados según los criterios anteriores sus posiciones, se determinarán mediante los siguientes criterios, en orden de preferencia:

- a. El mayor número de puntos obtenidos entre los equipos en cuestión
- b. La diferencia de goles teniendo en cuenta los partidos entre los equipos en cuestión
- c. El mayor número de goles a favor anotados por cada equipo en los partidos disputados entre los equipos en cuestión
- d. Sorteo del comité organizador de la Copa Mundial

Octavos de final

Participan los 16 equipos (8 partidos) que pasaron la primera ronda. Los partidos se hacen según distribución predeterminada:

Partido 49	1ero Grupo A	-	2do Grupo B
Partido 50	1ero Grupo C	-	2do Grupo D
Partido 51	1ero Grupo E	-	2do Grupo F
Partido 52	1ero Grupo G	-	2do Grupo H

Partido 53	1ero Grupo B	-	2do Grupo A
Partido 54	1ero Grupo D	-	2do Grupo C
Partido 55	1ero Grupo F	-	2do Grupo E
Partido 56	1ero Grupo H	-	2do Grupo G

Los ganadores de estos partidos pasan a la siguiente fase y los perdedores quedan eliminados.

Si luego de los 90 minutos de juego el partido se encuentra empatado se juegan dos tiempos extras de 15 minutos. Si luego de ello el partido sigue empatado, el ganador se define por una tanda de cinco tiros de penal. El equipo que menos falle será el ganador. Si después de esta tanda de penales siguen ambos equipos empatados se recurre a la ejecución de un nuevo tiro por cada equipo, repitiéndose hasta que un equipo aventaje al otro habiendo ejecutado ambos el mismo número de tiros. Estas condiciones siguen valiendo para todas las demás etapas.

Cuartos de final

Participan los 8 equipos (4 partidos) ganadores de la etapa anterior según distribución predeterminada:

Partido 57	Ganador Partido 49	-	Ganador Partido 50
Partido 58	Ganador Partido 53	-	Ganador Partido 54
Partido 59	Ganador Partido 51	-	Ganador Partido 52
Partido 60	Ganador Partido 55	-	Ganador Partido 56

Semifinales

Participan los 4 equipos (2 partidos) ganadores de la etapa anterior según distribución predeterminada:

Partido 61	Ganador Partido 57	-	Ganador Partido 58
Partido 62	Ganador Partido 59	-	Ganador Partido 60

Tercer puesto

Los equipos perdedores de las semifinales juegan un partido por el tercer y cuarto puesto:

Partido 63 Perdedor Partido 61 - Perdedor Partido 62

Final

Los equipos ganadores de las semifinales disputan el partido final, el ganador obtiene la Copa del Mundo:

Partido 64 Ganador Partido 61 - Ganador Partido 62

La etapa de requerimientos de software determinó que se debe registrar la siguiente información:

CONFEDERACIONES DE FÚTBOL

A nivel de fútbol hay una clasificación en la cual cada país pertenece a una confederación.

Los códigos y nombres de las confederaciones son:

- Concacaf: Confederación Centroamericana y del Caribe de Fútbol
- UEFA. Unión Europea de Fútbol Asociado
- Conmebol: Confederación Sudamericana de Fútbol
- CAF: Confederación de fútbol de Africa
- AFC: Confederación de fútbol de Asia
- OFC: Confederación de fútbol de Oceanía

EQUIPOS

Código del país según códigos de la FIFA. Todos los datos que se refieran a país usan estos códigos. -char(3)

Nombre del país. -char(40)

Cada país está afiliado a una confederación.

Cada equipo tiene jugadores, entrenador, asistentes y federativos.

Grupo en que inicia el campeonato (8 grupos de 4 equipos): A, B, C, D, E, F, G, H.

JUGADORES

Número de pasaporte. Es único a nivel de todas las entidades que lo vayan a usar. -x(20).

Nombre. -char(15)

Apellido 1. -char(15)

Apellido 2. -char(15)

Puesto (Portero, Defensa, Mediocampista, Delantero).

Fecha de nacimiento.

Número de camiseta. El equipo le asigna un número único entre 1 y 99.

ENTRENADORES

Número de pasaporte.

Nombre.

Apellido 1.

Apellido 2.

País de nacionalidad.

Fecha de nacimiento.

Fecha en que inició el puesto en el equipo.

ASISTENTES

Número de pasaporte.

Nombre.

Apellido 1.

Apellido 2.

País de nacionalidad

Tipo de asistente (Técnico, Preparador Físico, Médico, Psicólogo, Nutricionista, Administrativo, Delegado de equipo) Fecha de nacimiento.

Fecha en que inició el puesto en el equipo.

FEDERATIVOS

Número de pasaporte.

Nombre.

Apellido 1.

Apellido 2.

País de nacionalidad.

Puesto (Presidente, Vicepresidente, Secretario, Tesorero, Fiscal, Vocal) Fecha de nacimiento.

Fecha en que inició el puesto en el equipo.

SEDES

Son los lugares en donde se realizan los partidos.

Nombre de la sede. Es único. –char(40)

Nombre del estadio. Es único. –char(40)

Capacidad del estadio (cantidad máxima de aficionados). De 25,000 a 150,000.

PARTIDOS

Número de partido. Es único. De 1 a 64.

Etapas a que corresponde el partido: Fase de grupos, Octavos de final, Cuartos de final, Semifinales, Tercer Puesto, Final.

Nombre del equipo (país) 1.

Nombre del equipo (país) 2.

Nombre del estadio.

Fecha.

Hora.

Cantidad de aficionados.

Jugadores titulares del equipo 1. Consiste de una lista con los 11 jugadores que inician el partido. De estos titulares uno de ellos es el capitán.

Jugadores suplentes del equipo 1. Consiste de una lista con los 12 posibles jugadores que pueden sustituir a los que están jugando este partido.

El entrenador del equipo 1, 2 asistentes técnicos, 1 médico y 1 delegado de equipo.

Cambios realizados del equipo 1. Consiste de una lista con los jugadores que estaban jugando pero salieron porque fueron sustituidos. Contiene minutos y segundos del cambio, jugador que sale, jugador que entra. Se permite un máximo de 3 cambios. Si el capitán sale del partido, el equipo nombra a otro capitán de los que quedan jugando.

Jugadores titulares del equipo 2.

Jugadores suplentes del equipo 2.

El entrenador del equipo 2, 2 asistentes técnicos, 1 médico y 1 delegado de equipo.

Cambios realizados del equipo 2.

Cuerpo arbitral: 1 árbitro principal, 2 árbitros asistentes (guardalíneas), 1 4to árbitro, 1 5to árbitro. El rol de un árbitro puede variar en cada partido, por ejemplo, puede ser el principal en un partido y asistente en otro, etc.

Todos estos actores pueden cambiar de partido a partido, pero siempre deben estar en las listas oficiales respectivas, por ejemplo, un jugador debe estar en la lista de jugadores de su equipo, etc.

Goles del equipo 1: minuto y segundos del gol (o autogol) y jugador que lo realizó. Si aquí aparece el nombre de un jugador del otro equipo significa que fue un autogol. Goles del equipo 2.

Tarjetas amarillas: minuto y segundos, jugador tarjetado.

Tarjetas rojas: minuto y segundos, jugador tarjetado.

Minutos de reposición del primer tiempo.

Minutos de reposición del segundo tiempo.

Para los partidos que correspondan:

- ¿ Se jugaron tiempos extras ?
- ¿ Se tiraron penales ? Detalle de penales tirados:
 - + número de penal (1, 2, 3, 4, 5, 6, etc.)
 - + jugador que lo hizo
 - + penal anotado: Si, No.

ÁRBITROS

Número de pasaporte.

Nombre.

Apellido 1.

Apellido 2.

País de nacionalidad.

Fecha de nacimiento.

Fecha en que inició el arbitraje en campeonatos mundiales.

DESARROLLAR LOS SIGUIENTES REPORTES (A IMPRESORA Y PANTALLA)

A) EQUIPOS PARTICIPANTES POR CONFEDERACIÓN

Concacaf

México

Costa Rica

Panamá

...

El usuario puede solicitar el reporte de una confederación o de todas.

B) INFORME OFICIAL DEL PARTIDO

Este informe presenta todos los datos indicados en la sección PARTIDOS.

El usuario puede solicitar el reporte de un partido o de todos.

C) GRUPOS Y CLASIFICACIONES

Estas tablas por grupo se van actualizando en línea conforme se registran los resultados de los partidos. Los equipos se colocan en la tabla según reglas mencionadas en la fase de grupos. Inicialmente se colocan en orden alfabético.

Grupo A

Posición	Equipo	Puntos	PJ	PG	PE	PP	GF	GC	Dif
1	A1								
2	A2								
3	A3								
4	A4								

Equipo: Nombre del país

Puntos: 3 por cada partido ganado (PG), 1 por cada partido empatado (PE)

PJ: Partidos Jugados

PG: Partidos Ganados

PE: Partidos Empatados

PP: Partidos Perdidos

GF: Goles a Favor

GC: Goles en Contra

Dif: GF – GC

...

Grupo H

Posición	Equipo	Puntos	PJ	PG	PE	PP	GF	GC	Dif
1	H1								
2	H2								
3	H3								
4	H4								

El usuario puede solicitar el reporte de un grupo o de todos.

D) TABLA GENERAL DE POSICIONES

Refleja la posición final de cada equipo en el campeonato. Se actualiza cuando termina el campeonato. La clasificación es en estos bloques: de la posición 32 a la 17 quedan los 16 equipos eliminados en la primera fase de grupos, de la posición 16 a la 9 quedan los 8 equipos eliminados en octavos de final, de las posiciones 8 a la 5 quedan los 4 equipos eliminados en cuartos de final, luego estarán los primeros 4 lugares. Dentro de cada

bloque se clasifican de acuerdo a los puntos obtenidos, luego por criterio de mayor diferencia de goles, más goles a favor, menos goles en contra, por último en caso de empate en todos estos criterios se usa el orden alfabético.

Posición	Equipo	Puntos	PJ	PG	PE	PP	GF	GC	Dif
1	Campeón								
2	Subcampeón								
3	Tercer lugar								
4	Cuarto lugar								
5	Cuartos de final								
6	..								
7	..								
8	..								
9	Octavos de final								
10	..								
...									
16	..								
17	Fase de grupos								
...	..								
32	..								

E) TABLA DE GOLEADORES

Refleja la lista de goleadores del campeonato. Se actualiza con cada partido. Tiene un orden descendente, de mayor a menor cantidad de goles, en caso de empate se usa el orden alfabético. No cuentan los goles de las series de penales lanzados después de los tiempos extras.

Posición	Nombre del jugador	Equipo	Goles
1			
2			
...			

POR HACER:

Diseño de la base de datos:

- Diseño conceptual.
- Diseño lógico. Este modelo debe garantizar que todas las entidades están en 3FN.
Para cada entidad explicar porque está en 3FN.
- Diseño físico. Incluir entidades, atributos, llaves primarias, toda la integridad referencial, vistas, índices y todas las demás restricciones.

Definir al menos dos de cada uno de los siguientes componentes:

- Procedimientos almacenados (2)
- Funciones (2)
- Triggers (2)
- Transacciones (usando commit y rollback) (2)
- Vistas (2)
- Índices (2)

Desarrollar una consulta al diccionario de datos (DD): para cada archivo (tabla), se debe mostrar los nombres de los campos (atributos), descripción (indicar descripción de cada campo), tipos de datos, tamaños, llaves foráneas, llave primaria. La consulta puede ser para todas las tablas o alguna específica.

Desarrollar los programas necesarios para hacer el CRUD (mantenimiento de datos) de la información de:

- Equipos. No debe hacer los CRUD de jugadores, entrenadores, asistentes y federativos, pero sí debe tener la posibilidad de consultar esos actores para un equipo.
- Partidos: incluye toda la información determinada en la etapa de requerimientos: seleccionar equipos del partido, seleccionar los jugadores titulares, etc.

Carga inicial de datos:

- Lista de equipos: código y nombre del país, confederación. La información está en la tabla Equipos en la base de datos Campeonato en MS SQL Server, de aquí tiene que cargar la información de equipos para su base de datos. Los atributos tienen la misma estructura indicada en los requerimientos.
- Lista de jugadores: código del país, nombre y apellidos, puesto, fecha de nacimiento (AAAA-MM-DD), número. La información va a estar en un archivo de Excel y de ahí la carga a su base de datos. Los atributos tienen la misma estructura indicada en los requerimientos.
- Otros datos iniciales necesarios los pueden cargar como decidan.

Desarrollar las siguientes aplicaciones

- Equipos participantes por confederación
- Informe oficial del partido
- Grupos y clasificación
- Tabla general de posiciones
- Tabla de goleadores

CONSIDERACIONES ADICIONALES

- 1- En caso de que falte información acerca de requerimientos para completar los modelos de datos, debe consultarla con el profesor.
- 2- El sistema usará un menú para acceder a las diferentes funcionalidades del software.
- 3- El sistema brindará una opción de ayuda que desplegará de inmediato en el monitor el manual de usuario.
- 4- Se deben hacer todas las validaciones de datos y procesos.
- 5- Para interactuar con el sistema su equipo diseña una GUI, tanto para los procesos interactivos como de diseño de reportes y consultas, buscando consistencia, funcionalidad y facilidad de uso.
- 6- Pueden agregar otras funcionalidades que consideren van a mejorar el producto.

Temas investigados (De qué trata, cómo se usa)

1-) Sintaxis de ORACLE: Comparado a la sintaxis trabajada en el proyecto anterior MS, la de ORACLE se usa varchar2 como default siempre para un texto, y con la fecha se introduce con TO_TIMESTAMP('datosAInsertar', 'formato')

2-) Conexión que se hizo entre JAVA y ORACLE: Se utiliza esta biblioteca java.sql.DriverManager y java.sql.Connection para establecer la conexión entre NetBeans y ORACLE. Aquí se muestra el código que se usa para acceder a la base de datos que está almacenada en un servidor de la red del TEC.

```
//Establezco conexión de ORACLE
public Conexion Conectar()
{
    try{

        Class.forName("oracle.jdbc.OracleDriver");
        String BaseDeDatos = "jdbc:oracle:thin:@//172.19.32.101:1521/grupo07.basedatos";
        setConexion(DriverManager.getConnection(BaseDeDatos,getName(),getPass()));

        if(getConexion() != null)
        {
            this.status = true;
            System.out.println("Conexion exitosa a esquema JLIVE");
        } else {
            System.out.println("Conexion fallida a JLIVE");
        }
    } catch(Exception e) {
        e.printStackTrace();
    }
    return this;
}
```

3-) Se investigó cómo mostrar un PDF: Trata de mostrar un .PDF en una interfaz JAVA, se usó estas bibliotecas java.nio.ByteBuffer y java.nio.channels.FileChannel para la conversión debida con el fin de mostrarlo en la interfaz de JAVA:

```
public void actionPerformed(ActionEvent e)
{
    indice=0;
    selector=new JFileChooser();
    int op=selector.showOpenDialog(Principal.this);
    if(op==JFileChooser.APPROVE_OPTION)
    {
        try{
            File file = selector.getSelectedFile();
            RandomAccessFile raf = new RandomAccessFile(file, "r");
            FileChannel channel = raf.getChannel();
            ByteBuffer buf = channel.map(FileChannel.MapMode.READ_ONLY,0, channel.size());
            pdfFile = new PDFFile(buf);
            PDFPage page = pdfFile.getPage(indice);
            panelpdf.showPage(page);
            repaint();
        }catch(IOException ioe){
            JOptionPane.showMessageDialog(null, "Error al abrir el archivo");
        }
    }
}
```



```
}  
}
```

4-) Bibliotecas que se usaron para hacer consultas en ORACLE:

->java.sql.ResultSet, java.sql.ResultSetMetaData, java.sql.Statement: Se usan para obtener los resultados de las consultas que hacemos a la Base de Datos.

->java.sql.SQLException, java.util.logging.Level, java.util.logging.Logger: Se usa para los errores que pueden pasar en la ejecución de una consulta en sql a la base de datos.

->java.util.ArrayList, java.util.Vector, import javax.swing.JOptionPane, javax.swing.table.DefaultTableModel: Se tratan de utilizar las herramientas que nos brinda para poder trabajar con datos en java, por ejemplo un vector se utiliza para que almacene los resultados de una tabla con el fin de meterla en la TABLEMODEL.

5-) Cómo se hizo la migración de la tabla EQUIPO de Microsoft Server a ORACLE:

Se hizo la conexión al programa SQL SERVER para después sacar la info de la tabla e insertarla a la tabla de oracle.

6-) Cómo se hizo la migración de la tabla jugadores de EXCEL a ORACLE:

Se trabajo con las bibliotecas específicas para poder trabajar con el excel en java y para una vez insertarlo en ORACLE.

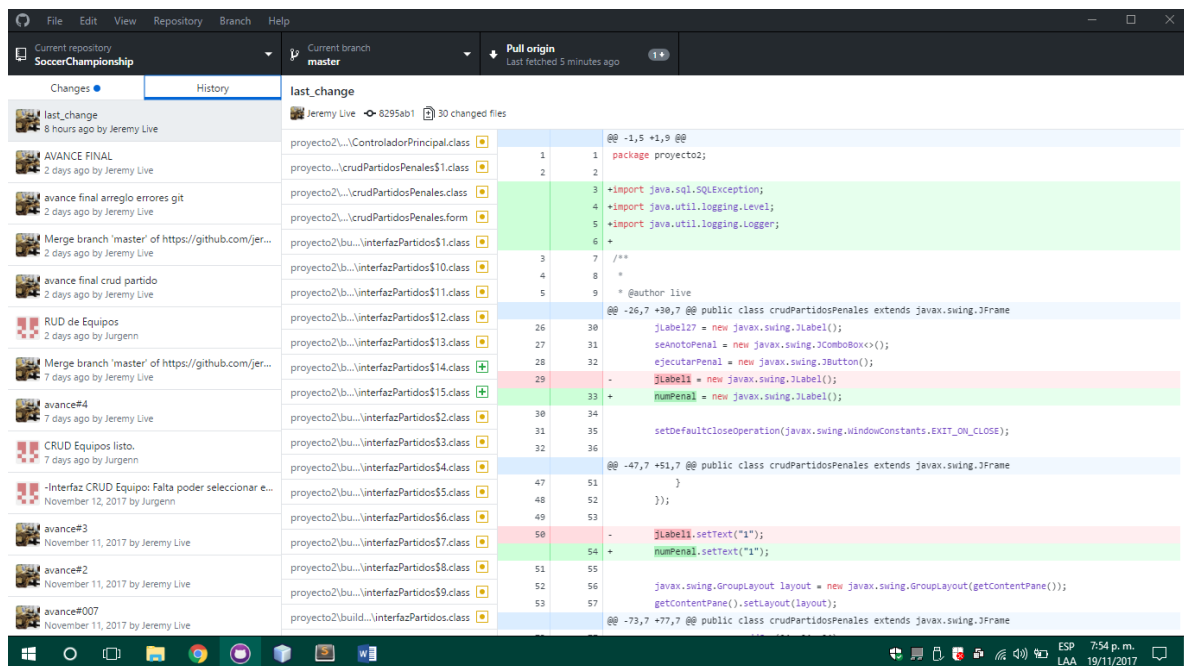
Software de control de versiones

GITHUB.COM - GitHub desktop

->Se investigó la herramienta que se usa en la nueva industria tecnológica, el poder desarrollar software y tenerlo siempre en la NUBE, y gracias a la aplicación se vuelve muy fácil y sencillo manejar GitHub, el poder subir commits y llevar un control de los avances ellos hasta el momento.

-> Se usa para dar mantenimiento al software de manera profesional.

-> Con la aplicación para desktop se puede ver el historial de una manera muy util para el manejo de codigo entre grupos, por ejemplo en una reunión de grupo de trabajo se quiere saber que lineas especificas cambio el compañero, gracias a esta herramienta se logra visualizar muy bien.

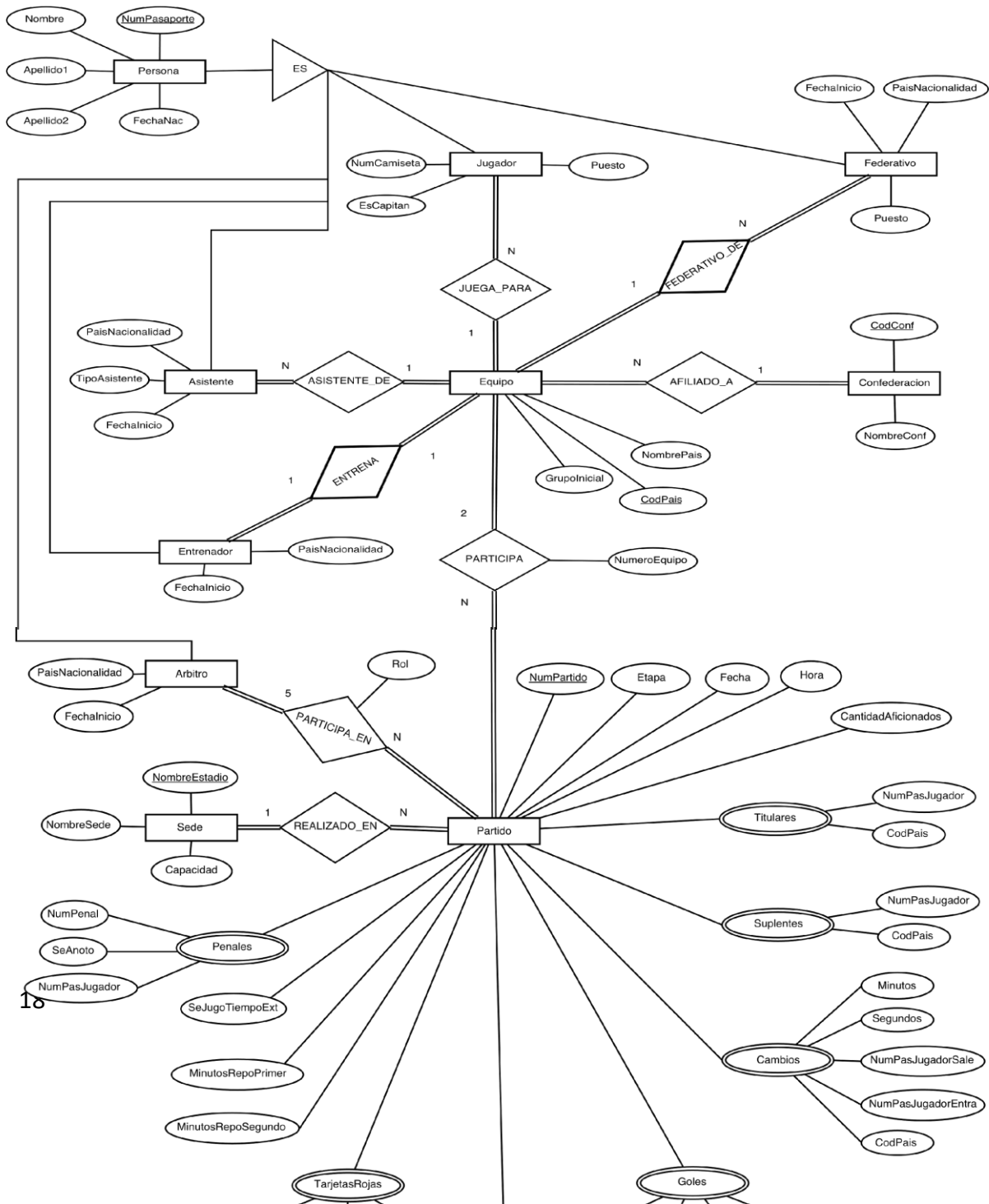


-> Y en la otra pestaña de la aplicación “CHANGES” se seleccionan los cambios que quieran hacerse para luego hacer el COMMIT y después el PUSH. Todo esto con simples clicks.

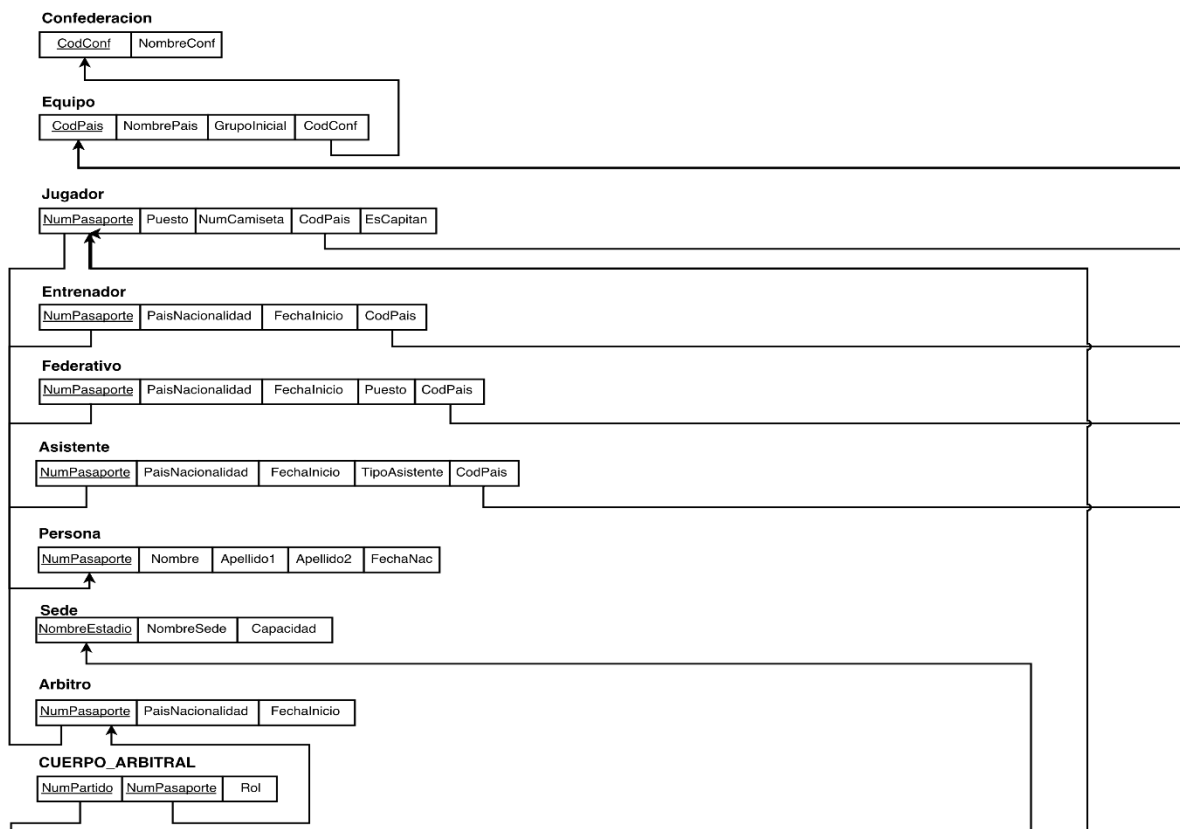
-> También tienes otras opciones como “current branch” se puede crear o escoger una branch.

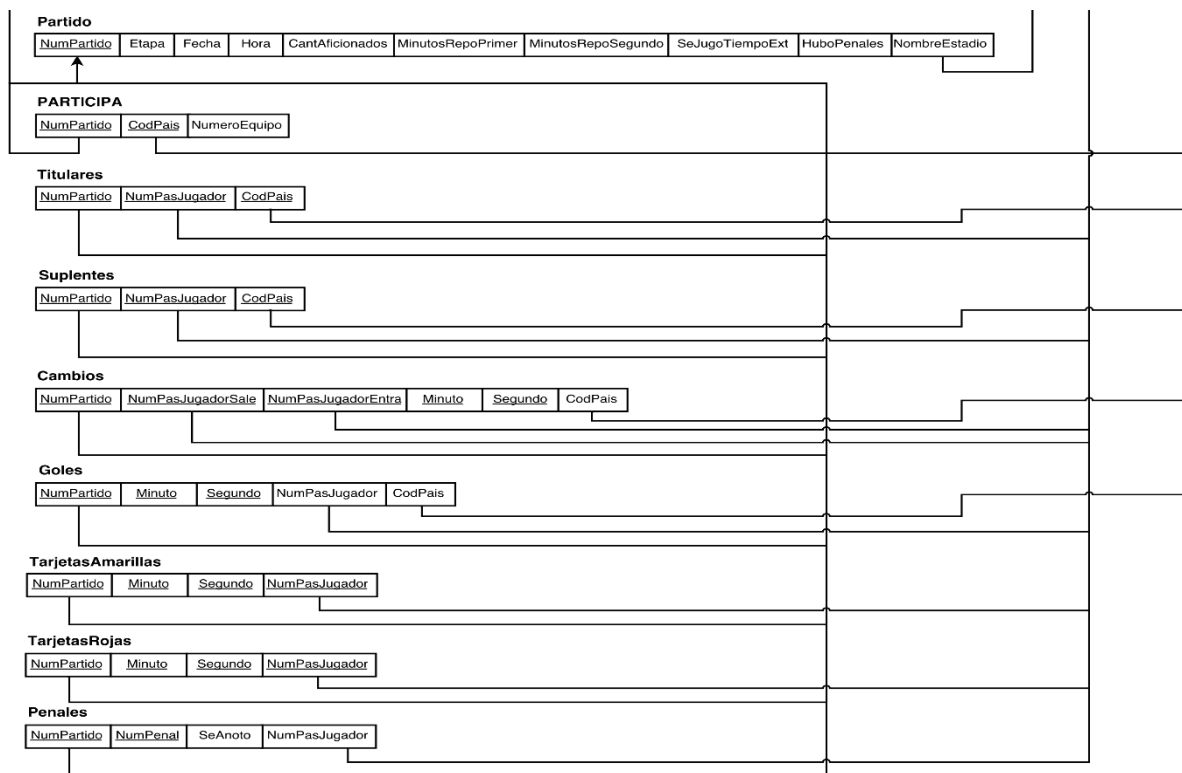
-> También está la opción “current repository” se puede elegir las clouds que se han subido hasta ahora a la APP.

Diseño conceptual de la base de datos

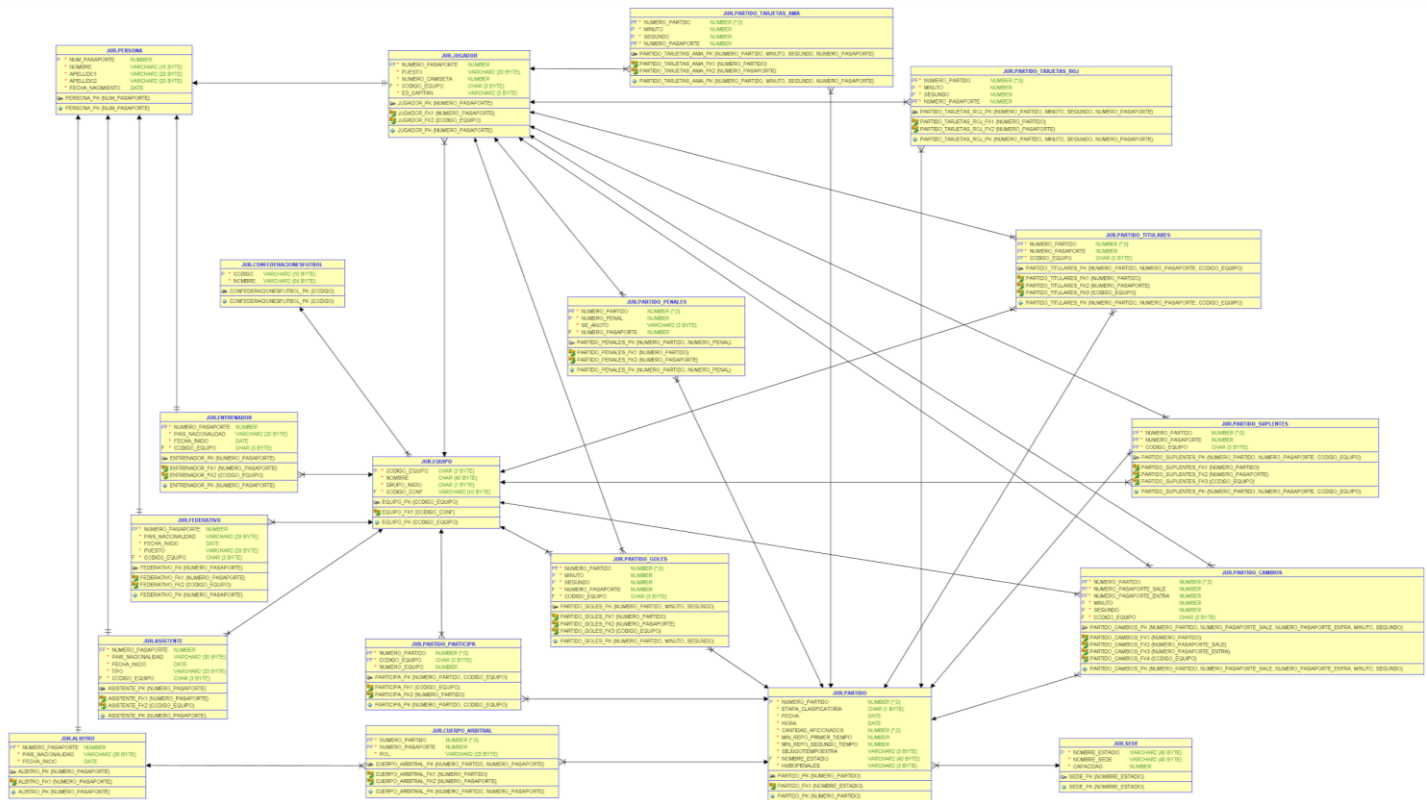


Diseño lógico de la base de datos





21



Programas almacenados: procedimientos, funciones, triggers

Procedimientos:(1)(2)

Funciones:(1)(2)

Triggers:(1)(2)

Transacciones, vistas, índices

Transacciones:

(1)

```
//INSERTO EN LA TABLA CUERPO_ARBITRAL USANDO UNA TRANSACCION
public void ejecutarArbitro() throws SQLException
{
    //Hago conexion
    connection = Conexion.getConnection();
    statement = connection.createStatement();

    //se deshabilita el modo de confirmación automática
    connection.setAutoCommit(false);

    //Creo query's
    String insertCUERPO_Principal = "INSERT INTO CUERPO_ARBITRAL VALUES
    (" + getNumero_partido() + "," + getArbitro_principal() + "," + 'Principal' + ")";
    String insertCUERPO_Asist1 = "INSERT INTO CUERPO_ARBITRAL VALUES
    (" + getNumero_partido() + "," + getArbitro_asistente_1() + "," + 'Asistente' + ")";
    String insertCUERPO_Asist2 = "INSERT INTO CUERPO_ARBITRAL VALUES
    (" + getNumero_partido() + "," + getArbitro_asistente_2() + "," + 'Asistente' + ")";
    String insertCUERPO_4to = "INSERT INTO CUERPO_ARBITRAL VALUES
    (" + getNumero_partido() + "," + getArbitro_4to() + "," + 'Cuarto' + ")";
    String insertCUERPO_5to = "INSERT INTO CUERPO_ARBITRAL VALUES
    (" + getNumero_partido() + "," + getArbitro_5to() + "," + 'Quinto' + ")";

    //INSERTS A PARTIDO_CAMBIOS
    statement.executeUpdate(insertCUERPO_Principal);
    statement.executeUpdate(insertCUERPO_Asist1);
    statement.executeUpdate(insertCUERPO_Asist2);
    statement.executeUpdate(insertCUERPO_4to);
    statement.executeUpdate(insertCUERPO_5to);
    getConnection().commit();

    //se habilita el modo de confirmación automática
    connection.setAutoCommit(true);
}
```

->Al llamar a esta función se utiliza el try catch para la exención de rollback

```
try {
    control.cargarEquipos(equipo1.getSelectedItemId().toString(), equipo2.getSelectedItemId().toString(),
    nombreEstadio.getSelectedItemId().toString(), cantAficionados.getText(),
    jugaronTiemposExtras.getSelectedItemId().toString(), tiraronPenales.getSelectedItemId().toString(),
    grupo_clasificatoria.getSelectedItemId().toString(), primerTiempoR.getSelectedItemId().toString(),
    segundoTiempoR.getSelectedItemId().toString(), hora.getSelectedItemId().toString(),
    minutos.getSelectedItemId().toString(), año.getSelectedItemId().toString(), mes.getSelectedItemId().toString(),
    dia.getSelectedItemId().toString());
} catch (SQLException ex) {
    System.err.println("ERROR: " + ex.getMessage());
    if(control.getConnection()!=null)
    {
        System.out.println("Rollback");
        try {
            //deshace todos los cambios realizados en los datos
            control.getConnection().rollback();
        } catch (SQLException ex1) {
            System.err.println( "No se pudo deshacer" + ex1.getMessage() );
        }
    }
}
```



```

    }
}

```

(2)

```

//INSERTO EN LA TABLA PARTIDO_PARTICIPA Y EN PARTIDO - Se usa una transaccion
public void queryPartido() throws SQLException
{
    //Hago conexion
    connection = Conexion.getConnection();
    statement = connection.createStatement();

    //se deshabilita el modo de confirmación automática
    connection.setAutoCommit(false);

    //Variables a usar en la funcion de partidos, total partidos 64
    String insercionHacer = "";
    insercionHacer += getNumero_partido()+", "+getGrupo_clasificatoria()+", "+TO_TIMESTAMP(" "+getAño_partido()+"-
"+getMes_partido()+"- "+getDia_partido()+", 'YYYY-MM-DD HH24:MI:SS.FF9'), "+
    "TO_DATE(" "+getHora_partido()+", 'hh24:mi')"+", "+getCantidad_aficionados()+", "+getPrimerTiempoRepMin()+", "+
    getSegundoTiempoRepMin()+", "+getTiempo_extra()+", "+getNombre_estadio()+", "+getTieraron_penales()+"";

    //Inserts a PARTICIPA
    String insertPARTICIPA_Equipo1 = "INSERT INTO PARTIDO_PARTICIPA VALUES (" +getNumero_partido()+", "+getEquipo_1()+", "+1)";
    String insertPARTICIPA_Equipo2 = "INSERT INTO PARTIDO_PARTICIPA VALUES (" +getNumero_partido()+", "+getEquipo_2()+", "+2)";

    //Inserts a PARTIDO
    String insertPartido = "INSERT INTO PARTIDO (NUMERO_PARTIDO, ETAPA_CLASIFICATORIA, FECHA, HORA,
CANTIDAD_AFICIONADOS, "+
    "MIN_REPO_PRIMER_TIEMPO, MIN_REPO_SEGUNDO_TIEMPO, SEJUGOTIEMPOEXTRA, NOMBRE_ESTADIO, HUBOPENALES)
VALUES (" +insercionHacer+")";

    System.out.println(insertPartido);

    statement.executeUpdate(insertPartido);
    statement.executeUpdate(insertPARTICIPA_Equipo1);
    statement.executeUpdate(insertPARTICIPA_Equipo2);

    getConnection().commit();
    connection.setAutoCommit(true);
}

```

->Al llamar a esta funcion se utiliza el try catch para la exención de rollback

```

try {
    //llamo a la función de queryPartido
} catch (SQLException ex) {
    if(control.getConnection()!=null)
    {
        System.out.println("Rollback");
        try {
            //deshace todos los cambios realizados en los datos
            control.getConnection().rollback();
        } catch (SQLException ex1) {
            System.err.println( "No se pudo deshacer" + ex1.getMessage() );
        }
    }
}
}

```

Vistas: (1)(2)

Índices:

(1)

```
-- Indice INDICE_PARTIDOSDELEQUIPO
-- Nos permite encontrar los partidos que jugó un equipo de forma más rápida.
CREATE INDEX "JLIVE"."INDICE_PARTIDOSDELEQUIPO" ON "JLIVE"."PARTIDO_PARTICIPA" ("CODIGO_EQUIPO")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE
DEFAULT)
TABLESPACE "SYSTEM" ;
```

(2)

```
-- Indice INDICE_GOLESPOREQUIPO
-- Sirve para hallar la cantidad de goles que hizo un equipo en cada partido de manera más rápida..
CREATE INDEX "JLIVE"."INDICE_GOLESPOREQUIPO" ON "JLIVE"."PARTIDO_GOLES" ("CODIGO_EQUIPO")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE
DEFAULT)
TABLESPACE "SYSTEM" ;
```

Conclusiones del trabajo (Problemas y soluciones a los mismos):

1-) El problema más grande que se tuvo que investigar para llegar a una solución, fue con el GITHUB... al haber hecho dos commit de la misma clase... se creaba código basura cuando chocaban líneas de código y creaba muchos errores.

Solución: La solución que se aplicó fue no trabajar en las mismas funciones y si se trabaja en la misma clase, es recomendable hacer distintas tareas para que no choquen líneas de código.

2-) Con el código que se maneja para poner los pts a la tabla GRUPO Y CLASIFICACIONES, se daba una corrupción al momento de actualizar los datos en la tabla cada que se insertaba una tupla.

Sol: La solución que se aplicó fue hacer lo más cronológico posible las actualizaciones.

3-) Se dificultó cómo imprimir tablas de diferentes tamaños en la ventana gráfica; Solución: Se utilizó una lógica para obtener ese resultado.

4-) Me daba el error "NullPointerException" después de imprimir las tablas temporales.

Sol: El fallo fue que al final apuntaba a una casilla en blanco, lo que hice para solucionarlo fue restarle UNO.

5-) Los nombres de los atributos de dichas tablas. El estar poniendo esquema.tabla.atributo, se demoró mucho en la elaboración de concatenar strings.

Solución: con paciencia se logró.

Aprendizajes obtenidos:

1. Establecer conexión entre SQL y JAVA, es decir, entre ORACLE y NETBEANS.
2. El manejo de la herramienta de desarrollo de software GIT
3. El uso de las bibliotecas necesarias para la elaboración de la lógica del código en java y extracción de datos a la BD de ORACLE.

4. El uso de nuevas prácticas de programación.
5. El uso de la herramienta SQL DEVELOPER de Oracle.

Rúbrica de evaluación y análisis de resultados

Concepto	Puntos	Puntos obtenidos	Avance 100%/0	Análisis de resultados
Diseño conceptual	5		100	
Diseño lógico	5		100	
Diseño físico	5		100	
Consulta DD	5		100	
CRUD Equipos	5		100	
CRUD Partidos	10		100	
Carga inicial: lista de equipos	5		50	Se insertó manualmente en ORACLE
Carga inicial: lista de jugadores	5		100	
Validaciones de datos y procesos	10		100	
Equipos participantes por confederación	3		100	
Informe oficial del partido	12		100	
Grupos y clasificación	5		40	Nos dio problemas
Tabla general de posiciones	5		40	Nos dio problemas
Tabla de goleadores	5		100	
SQL: consultas en vivo que van a desarrollar los estudiantes de forma individual según el modelo implementado por el grupo	10*			
Ayuda	5		100	
TOTAL	100*			
Partes desarrolladas adicionalmente				