



**Instituto Tecnológico de Costa Rica**

**Sede Central**

**Escuela de Ingeniería en Computación**

**Base de Datos I**

**Grupo 01**

**Proyecto programado #2**

**Sistema de Información para Mundiales de Fútbol.**

**Profesor: William Mata Rodríguez**

**Estudiante:**

**Gerardo Monge Corella      2015145533**

**II Semestre 2017**

# **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

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									
Posicion	Equipo	Puntos	PJ	PG	PE	PP	GF	GC	GD
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

...

## 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 la 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.

Posicion	Equipo	Puntos	PJ	PG	PE	PP	GF	GC	Dif
1	Campeon								
2	Subcampeon								
3	Tercer lugar								
4	Cuarto lugar								
5	Cuartos de final								
6	...								
7	...								
8	...								
9	Octavos de final								
10	...								
...	...								

16	...								
17	Grupos								
18	...								
...	...								
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.

Posicion	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.

## **DOCUMENTACIÓN DEL PROYECTO**

### **REQUISITO PARA REVISAR EL PROYECTO**

El requisito consiste en presentar la documentación del proyecto indicada en esta sección. La nota de la documentación del proyecto sirve para aceptar o rechazar el proyecto: se revisan los proyectos que cumplan con este requisito en un 90% o más.

Enviar vía tecDigital, sección EVALUACIONES / PROYECTOS, una carpeta comprimida (.rar, .zip, etc.) de nombre proyecto2 que contenga:

#### Parte 1:

Documentación del proyecto (nombre:documentación\_campeonato\_mundial.PDF).

Portada. Contenido. (1 p)

Enunciado del proyecto. (1 p)

Temas investigados (material no estudiado en el curso). (12 p)

Por cada uno de estos temas debe poner el marco teórico: de qué trata, cómo se usa.

Software de control de versiones (12 p)

Diseño conceptual de la base de datos. (12 p, revisión final en semana 14)

Diseño lógico de la base de datos. (12 p, revisión final en semana 14)

Diseño físico de la base de datos. (4 p)

Programas almacenados: procedimientos, funciones, triggers. (6 p)

Transacciones, vistas, índices. (3 p)

Conclusiones del trabajo: (10 p)

Problemas y soluciones a los mismos.

Aprendizajes obtenidos.

Rúbrica de evaluación y análisis de resultados (PONGA LA HOJA DE LA RÚBRICA EN PÁGINA NUEVA). (12 p)

Tome la rúbrica de evaluación y por cada concepto calificado Usted debe indicar el % de avance y el análisis de resultados 100: Totalmente desarrollado. No hace falta análisis excepto que requiera hacer alguna observación. 80: Desarrollado parcialmente, un 80% (el % que corresponda). En el análisis indicar: ¿qué hace?, ¿qué falta?, ¿por qué no se completó? 0: No desarrollado. En el análisis indicar el motivo.

Partes que desarrolló adicionales a los requerimientos.

\*El rubro de SQL se califica de forma individual, tiene un valor de 10 puntos. Son consultas que se plantean y resuelven en el momento de la revisión conforme al modelo diseñado por el grupo. El puntaje obtenido por este rubro será sumado a los rubros que se obtienen de forma grupal (90 puntos) para obtener la calificación final del proyecto.

Manual de usuario (nombre: manual\_de\_usuario\_campeonato\_mundial.PDF). (15 p).

Es un documento de comunicación técnica utilizado para guiar a las personas que usan el software. Explica paso a paso cómo usar cada una de las funcionalidades del programa. Apóyese en imágenes, capturas de pantallas, menús, diagramas y los aspectos que considere van a servir como una guía útil para que el usuario pueda usar el programa. El manual se usará para la revisión funcional del proyecto.

Parte 2: Fuentes y otros objetos necesarios para ejecutar la aplicación.

**IMPORTANTE: CONOCIMIENTO DE LA SOLUCIÓN PRESENTADA.** En la revisión del trabajo, los estudiantes deben demostrar un completo dominio de la solución que implementaron, tanto desde el punto de vista técnico (uso de herramientas) como de la funcionalidad del proyecto. La revisión se puede hacer individualmente o en grupos, examinando la solución o temas específicos aplicados en el proyecto. Última línea

## Temas Investigados:

### Procesar sentencias de SQL en JDBC

Para esto generalmente se siguen los siguientes pasos:

1. Estableciendo una conexión.
2. Crear una instrucción.
3. Ejecute la consulta.
4. Procese el objeto de resultado.
5. Cierra la conexión.

#### 1. Estableciendo conexiones

En primer lugar, establezca una conexión con el origen de datos que desea utilizar. Un origen de datos puede ser un DBMS, un sistema de archivos heredado o alguna otra fuente de datos con un controlador JDBC correspondiente. Esta conexión está representada por un objeto *Connection*.

Primero se necesita establecer una conexión con la fuente de datos que se desea usar. Un origen de los datos puede ser un BDMS. Normalmente, una aplicación JDBC se conecta a un origen de datos de destino mediante una de dos clases:

- ✓ **DriverManager**: esta clase totalmente implementada conecta una aplicación a un origen de datos, que se especifica mediante una URL de base de datos.
- ✓ **DataSource**: esta interfaz permite que los detalles sobre el origen de datos subyacente sean transparentes a su aplicación. Las propiedades de un

objeto *DataSource* se establecen de manera que representa un origen de datos determinado.

## 2. Crear instrucciones

Una instrucción es una interfaz que representa una instrucción SQL. Se ejecutan objetos de instrucción y generan objetos de resultado, que es una tabla de datos que representa un conjunto de resultados de base de datos. Se necesita un objeto *Connection* para crear un objeto *Statement*.

Existen tres tipos diferentes de declaraciones:

- ✓ **Instrucción:** se utiliza para implementar instrucciones SQL simples sin parámetros.
- ✓ **PreparedStatement:** Se utiliza para precompilar sentencias SQL que pueden contener parámetros de entrada.
- ✓ **CallableStatement:** Se utiliza para ejecutar procedimientos almacenados que pueden contener parámetros de entrada y salida.

## 3. Procesamiento de objetos de resultado

Se accede a los datos en un objeto de resultado mediante un cursor. Note que este cursor no es un cursor de base de datos. Este cursor es un puntero que apunta a una fila de datos en el objeto de resultado. Inicialmente, el cursor se



coloca antes de la primera fila. Se llaman varios métodos definidos en el objeto de conjunto para mover el cursor.

#### 4. Cerrar conexiones

Cuando haya terminado de utilizar una instrucción, llame a la instrucción `Method.close` para liberar inmediatamente los recursos que está utilizando. Cuando se llama a este método, sus objetos de resultado se cierran.

### **Información sobre java.sql**

#### interfaz pública `SQLData`

El objeto `Class` de una clase que implementa la interfaz `SQLData` se introducirá en el mapa de tipo del objeto de conexión apropiado junto con el nombre SQL del definidor para el que se trata de una asignación personalizada.

Por lo general, una implementación de `SQLData` definirá un campo para cada atributo de un tipo estructurado SQL o un único campo para un tipo distinto de SQL. Cuando se recupera el definido de un origen de datos con el método `GetObject`, se asignará como instancia de esta clase. Un programador puede operar en esta instancia de clase igual que en cualquier otro objeto del lenguaje de programación Java y, a continuación, almacenar los cambios realizados en él llamando al método `PreparedStatement setObject`, que lo mapeará de nuevo al tipo SQL.

Se espera que la implementación de la clase para un mapeo personalizado será realizada por una herramienta. En una implementación típica, el programador simplemente proporcionaría el nombre del SQL definido, el nombre de la clase a la

que se está asignando, y los nombres de los campos a los que se va a mapear cada uno de los atributos del definidor. La herramienta usará esta información para implementar los métodos `SQLData readSQL` y `SQLData writeSQL`. El método `readSQL` llama a los métodos `SQLInput` apropiados para leer cada atributo de un objeto `SQLInput`, y el método `writeSQL` llama a métodos `SQLOutput` para escribir cada atributo de nuevo en el origen de datos a través de un objeto `SQLOutput`.

Un programador de aplicaciones normalmente no llamará a los métodos `SQLData` directamente, y los métodos `SQLInput` y `SQLOutput` se denominan internamente por métodos `SQLData`, no por código de aplicación.

### **Patrón vista controlador**

Se utilizó en el proyecto el patrón MVC o (Model-View-Controller) con la GUI implementada en JavaFX.

Este patrón busca la división de tareas entre la GUI y la lógica de negocios, se pretende que las vistas solo controlen la GUI y que se comuniquen con el Controlador y este se encargará de comunicarse con el Modelo de Datos.

En el caso de este proyecto, el modelo de datos es la Base de Datos en sí misma, el Controlador son las distintas clases que controlan cómo se comportan las vistas y las estas últimas son el código FXML generado para JavaFX.

El uso de una implementación en JavaFX permitió una división de tareas enorme, donde cada Controlador se encargaba de una tarea muy específica, donde por ejemplo; podían hacer las operaciones CRUD de una entidad en específico sin tener que necesitar una clase o dependencia con muchísimo código y difícil de debuggear y entender.

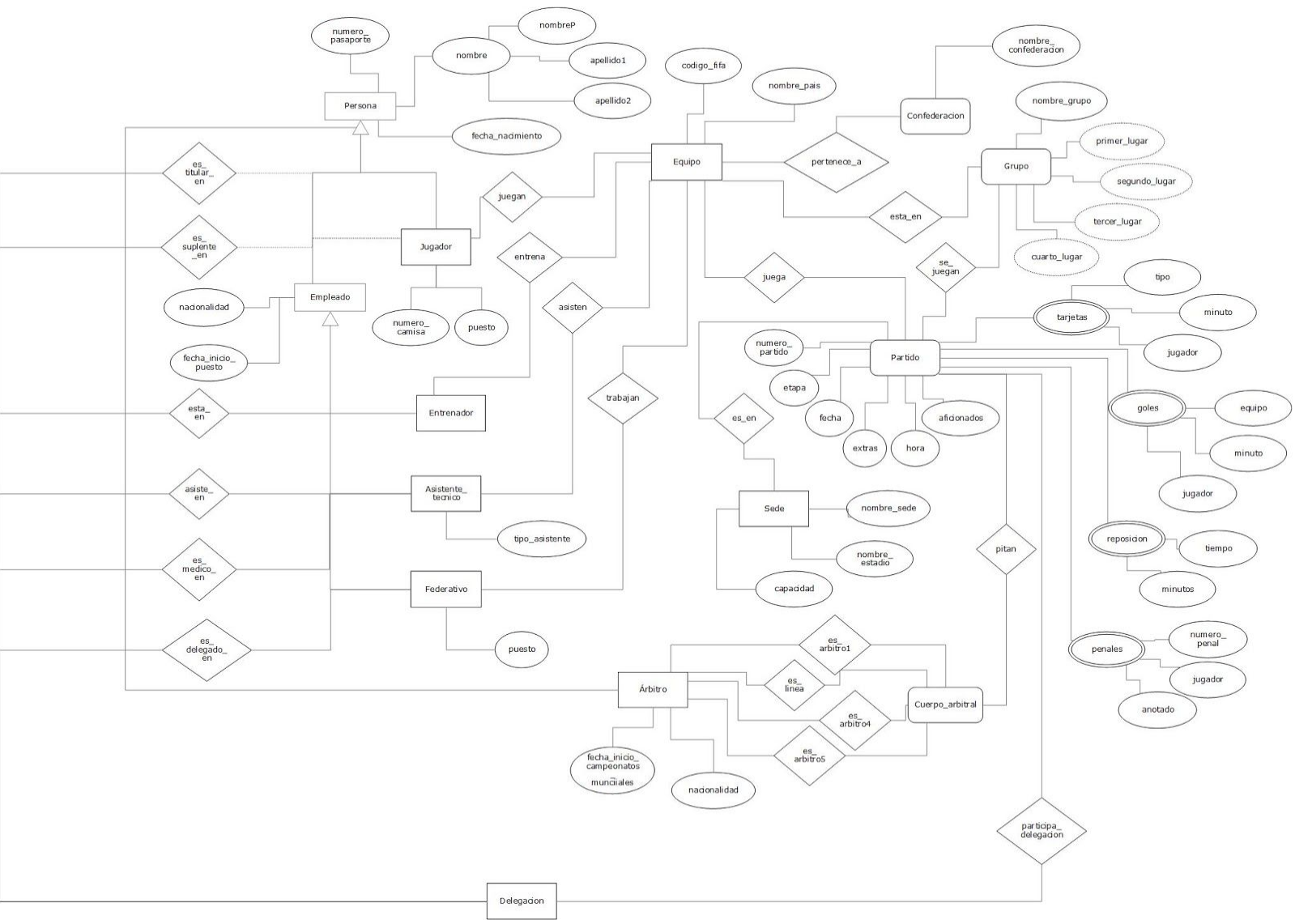
## **Software controlador de versiones**

Git es un sistema de control de revisiones rápido, escalable y distribuido con un conjunto de comandos inusualmente enriquecido que proporciona operaciones de alto nivel y acceso completo a internos.

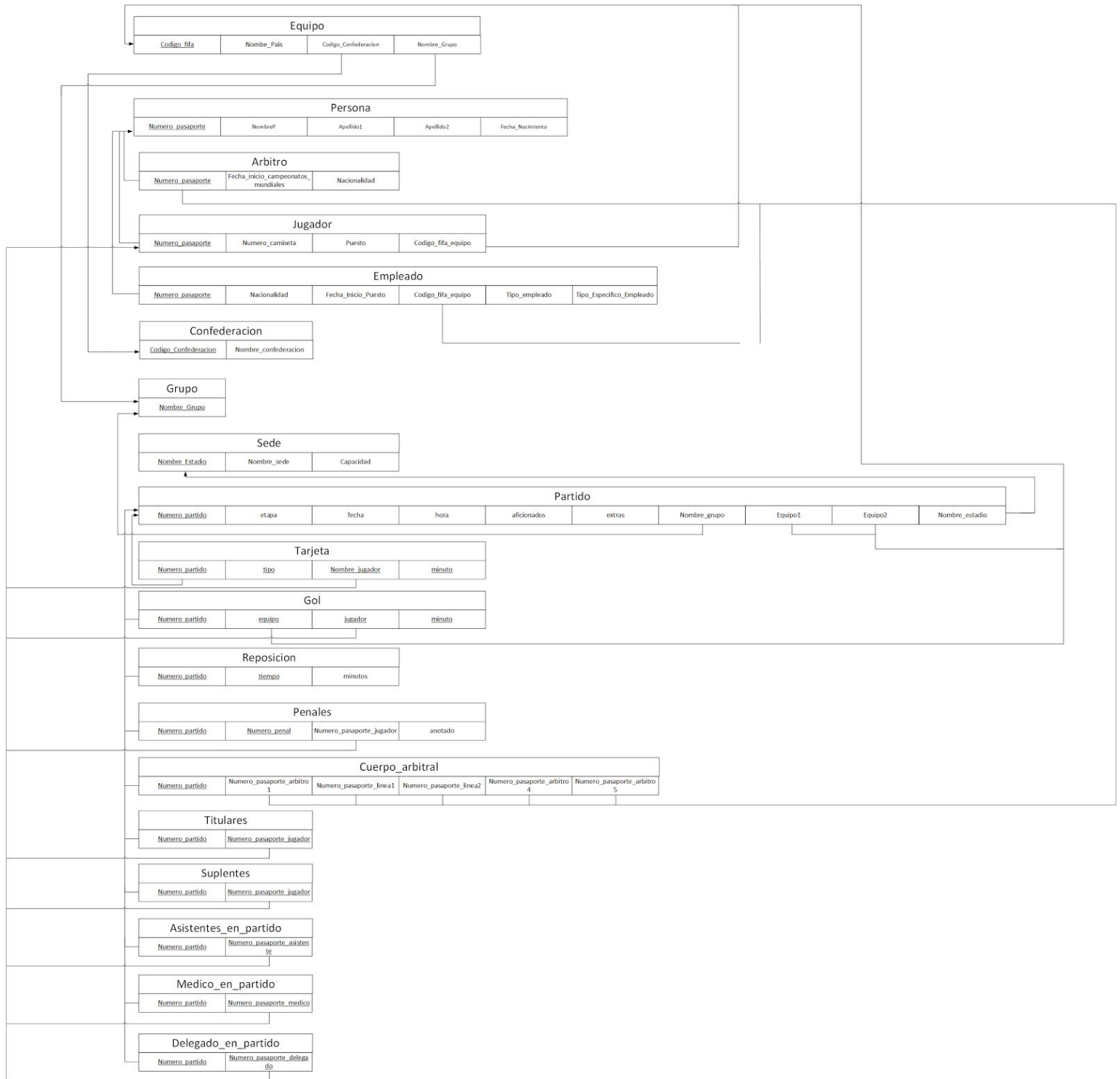
1. Primero se debe configurar el usuario y el correo de la persona que utiliza git.
2. Luego se debe crear el repositorio en la computadora con el comando *git init*
3. Después de agregar los archivos que estén en esa carpeta con el comando *git add* . para agregar a git todo lo que esté en la carpeta.
4. Después se debe hacer el commit, esto serían los comentarios de los cambios realizados a los archivos del repositorio, esto se hace con el comando *git commit -m "<mensaje de actualización>"*
5. Al usar github se puede crear un remote, que es un acceso remoto a un proyecto en github y editar los archivos desde git en la computadora, esto se hace con el comando *git remote add <nombre del remote> <url del proyecto en github u otra plataforma de git>*
6. Por último se hace el push para agregar a github los archivos de git, esto es con el comando *git push*
7. En caso de tener archivos que no estén en el repositorio de la computadora se debe hacer primero un pull con el comando *git pull*

Para poder ver el estado del proyecto, si hay archivos que no se han obtenido antes de hacer el push, se puede usar el comando *git status*, este muestra si hay archivos que necesitan ser agregados o ya se agregaron pero no se han hecho el commit.

**Diseño Conceptual (También incluido dentro del zip)**



## Diseño Lógico (También adjunto dentro del zip)



## **Normalización a 3FN**

Cuando se realizó el diseño conceptual y lógico de esta base de datos, siempre se tuvo muy presente el proceso de normalización y se realizó de una manera implícita, resguardando que los atributos fueran atómicos (1FN), que hubiera dependencia funcional de los atributos no llaves con todos los atributos llaves (2FN) y que no existiera una dependencia funcional entre valores no llaves y otros valores no llaves (3FN), lo que se puede notar con una simple vista al modelo lógico, por lo cual parece innecesario, revisar entidad por entidad para enunciar lo mismo en cada una.

## **Procedimientos**

### **Funciones**

### **Triggers**

### **Transacciones**

Las transacciones en este proyecto se encuentran en el código JDBC, en donde se apaga el bit de auto commit de la conexión para realizar transacciones en las que interviene la especialización de una persona, ya sea árbitro, empleado o jugador, entonces la transacción consiste de dos queries, donde se crea o actualiza la entidad persona primero y luego se actualiza la entidad que especializa a persona. Si algunas de las dos falla, se realiza un rollback. Además en la inicialización de datos también se utiliza una transacción donde básicamente se espera que se puede inicializar todos los datos y de no ser posible se hace un rollback.

### **Vistas**

### **Indices**

## **Conclusiones del trabajo:**

### **Problemas encontrados y soluciones**

- Aunque no fue necesario aprender a usar el sistema de versiones como en el primer proyecto, en este caso se presentó el problema de que, al trabajar solo; cuesta un poco hacerse al hábito de utilizar este software, realmente su utilidad se ve bastante reducida cuando no es grupo el que lo está utilizando.
  - Sin embargo, por requisitos del proyecto se utilizó, además resultó útil en algunos momentos usar branches experimentales sin tocar el código ya funcional.
- Normalmente cuando se trabaja es un poco más difícil acomodar el tiempo necesario para realizar un proyecto tan extenso como éste.
  - Realmente el problema no tiene solución, no hay más que buscar formar una disciplina para intentar completar el proyecto en su mayoría.
- Se presentó el problema del desconocimiento de la sintaxis de la base de datos de Oracle.
  - Este problema se solucionó buscando la referencia oficial de Oracle y tomando el tiempo para entender y asociar esta sintaxis a lo visto en clases.
- Se presentó el problema de que este proyecto tiene muchas entidades por modelar y como tal usar el MVC con JavaFX, puede mostrar ser una tarea pesada, tener un controlador por cada vista puede ser complejo.
  - Sin embargo la especialización que permite este modelo en conjunción con JavaFX mostró ser muy útil para realizar las tareas de una manera simple.

### **Lecciones aprendidas**

- Con este proyecto se aprendió a dominar un poco más la clase SQL de Java y el driver JDBC, tanto como sintaxis de PSQL además de la interacción entre los tipos de la interface SQL de Java y los tipos de la Base de Datos de Oracle.
  - Por ejemplo en el proyecto #1 se usó la clase Statement para hacer los queries necesarios, lo que llevó a la realización de validaciones manuales, que aunque funcionaban de manera perfecta para evitar la inyección SQL; eran fácilmente evitables con el uso de la clase PreparedStatement, que hace validación por sí misma.
- Se aprendió que aunque el sistema gestor de versiones puede ser una herramienta poderosa, hay momentos o casos específicos en los que podría presentar más problemas de los que un programador esperaría, aún así su uso es totalmente recomendado ya que permite experimentar con código de una forma que de otro modo no sería posible.
- Otra lección aprendida es la utilización de alguna herramienta que permita manejar las tareas por realizar, esto con el fin de completar un proyecto en su totalidad.



Concepto	Pts	Pts Obtenidos	Avance 100/%/0	Análisis de Resultados
Diseño conceptual	5		5	
Diseño lógico	5		5	
Diseño físico	5		5	
Consulta DD	5		0	No se pudo implementar por falta de tiempo
CRUD Equipos	5		5	
CRUD partidos	10		5	Se pueden inicializar los datos principales de los partidos, los datos de los jugadores titulares, suplentes, asistentes, delegados y medico, además del cuerpo arbitral, pero no dio tiempo de implementar lo restante
Carga inicial: lista de equipos	5		0	Se cargan los datos dentro del programa, ignorando los requisitos dados por razones de tiempo.
Carga inicial: lista de jugadores	5		0	Se cargan los datos dentro del programa, ignorando los requisitos dados por razones de tiempo.
Validaciones de datos y procesos	10		10	
Equipos participantes por confederacion	3		3	

Informe oficial del partido	12		0	No se pudo implementar por falta de tiempo
Grupos y clasificacion	5		0	No se pudo implementar por falta de tiempo
Tabla general de posiciones	5		0	No se pudo implementar por falta de tiempo
Tabla de goleadores	5		0	No se pudo implementar por falta de tiempo
SQL: consultas en vivo que van a desarrollar los estudiantes de forma individual según el modelo implementado por el grupo	10*			
Ayuda	5		5	
<b>TOTAL</b>	<b>100*</b>			
Partes desarrolladas adicionalmente				CRUDS en otras entidades no requeridas.

## **Bibliografía:**

1. Package java.sql, encontrado en:  
<https://docs.oracle.com/javase/7/docs/api/java/sql/package-summary.html>
2. Processing SQL Statements with JDBC, encontrado en:  
[https://docs.oracle.com/javase/tutorial/jdbc/basics/processingsqlstatements.html#establishing\\_connections](https://docs.oracle.com/javase/tutorial/jdbc/basics/processingsqlstatements.html#establishing_connections)
3. Estructura de las Aplicaciones Orientadas a Objetos El patrón  
Modelo-Vista-Controlador (MVC), Juan Pavón Mestras, España, 2008.
4. Git Reference, encontrado en: <https://git-scm.com/docs>