

INSTITUTO TECNOLÓGICO DE COSTA RICA.

Carrera:
Ingeniería en Computación.

Nombre del curso:
Bases de datos I.
Grupo 1.

Proyecto 2:
Campeonato Mundial.

Integrantes del grupo:
Alejandro Schmidt Ramírez
201235853
Kenneth Fernández Fuentes.
2015017634

Fecha de entrega:
19 de noviembre de 2017.

II Semestre
2017.

Nombre del profesor:
William Mata Rodríguez.

TEC | Tecnológico
de Costa Rica

Contenido

2: Enunciado del Proyecto.

- Enunciado entregado por el profesor.

7: Temas investigados para el desarrollo del proyecto.

- Material no estudiado en el curso.
 1. Temas investigados.
 2. Software de manejo de versiones.

11: Diseño Conceptual.

12: Diseño Lógico.

13: Diseño Físico.

15: Programas Almacenados.

16: Transacciones, vistas, índices.

17: Conclusiones del trabajo.

- Problemas encontrados y soluciones a los mismos.
- Aprendizaje obtenido.

18: Rúbrica de evaluación y análisis de resultados.

- Rúbrica de evaluación para el proyecto.

18: Puntos desarrollados adicionalmente.

- Funcionalidades adicionales agregadas en el 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.

Temas investigados.

Como se puede observar, se tiene una variable para los nombres, apellidos, puestos, países y nacionalidades por aparte, de manera que, el algoritmo toma un valor random de cada variable para formar una persona nueva. En el caso del Número de Pasaporte, el cual debe ser único para cada persona, se comienza con un integer en cero, el cual aumenta en uno por cada persona creada. De esta manera nos aseguramos de que ningún número de pasaporte llegue a repetirse. A continuación, podemos observar parte del resultado final en la hoja de EXCEL, que contiene todas las personas que, posteriormente, llenarán la base de datos.

0	Tim	Diskrud	Bolanos	09-11-1997	AF	Portero	1
1	Matt	Gamboa	Brooks	21-11-1985	AF	Delantero	2
2	Alejandro	Gamboa	Cubero	17-05-1977	AF	Mediocam	3
3	Kyle	Tejeda	Howard	03-12-1970	AF	Mediocam	4
4	Keylor	Yeldin	Besler	20-08-1997	AF	Portero	5
5	Johnny	Diskrud	Beasly	15-07-1925	AF	Defensa	6
6	Matt	Bolanos	Beasly	22-07-1980	AF	Delantero	7
7	Omar	Bolanos	Duarte	20-03-1992	AF	Delantero	8
8	Julian	Howard	Duarte	21-04-1945	AF	Defensa	9
9	DeMarcus	Dempsey	Johansson	29-09-2017	AF	Defensa	10
10	Johnny	Yeldin	Brooks	03-06-2002	AF	Defensa	11
11	Keylor	Howard	Brenes	02-10-1953	AF	Portero	12
12	Yeltsin	Cambronei	Tejeda	01-11-1962	AF	Portero	13
13	Omar	Diaz	Diskrud	02-01-2000	AF	Mediocam	14
14	Giancarlo	Barrantes	Yeldin	21-06-1924	AF	Defensa	15
15	Julian	Cubero	Campbell	20-02-1967	AF	Mediocam	16
16	Christian	Duarte	Umana	12-08-1967	AF	Defensa	17
17	Oscar	Beasly	Brenes	28-11-1936	AF	Delantero	18
18	Giancarlo	Besler	Cambronei	13-03-1968	AF	Defensa	19
19	Marcos	Acosta	Miller	22-07-1992	AF	Portero	20
20	Christian	Guzman	Duarte	15-02-1952	AF	Mediocam	21
21	Yeltsin	Brenes	Barrantes	05-04-2002	AF	Mediocam	22
22	John	Miller	Myrie	01-05-1935	AF	Delantero	23
38	Junior	Umana	Besler	21-04-1937	AL	Delantero	1
39	Aron	Gonzales	Miller	13-05-1963	AL	Mediocam	2
40	Junior	Bedoya	Acosta	27-12-1938	AL	Delantero	3
41	Clint	Ruiz	Pemberton	12-06-1956	AL	Defensa	4
42	DeAndre	Pemberton	Urena	21-10-1975	AL	Mediocam	5
43	Yeltsin	Besler	Diskrud	19-03-2011	AL	Delantero	6

El archivo de EXCEL está dividido en tres hojas, para jugadores, asistentes y federativos por aparte. De esta manera es mucho más sencillo agarrar los datos que corresponden a cada tabla sin confundirse y enviar un dato erróneo. Finalmente utilizamos un algoritmo, diseñado por el compañero Alejandro, que obtiene esos datos y los almacena en Hash Maps, los cuales serán leídos posteriormente y enviados a Oracle a través de consultas ejecutadas por medio de Netbeans. A continuación, se mostrará parte del algoritmo, más específicamente un “for” el cual es uno de los encargados de sacar la información de los jugadores de la hoja de EXCEL.

```

try {
    HashMap<Jugador, String> jugadores = getJugadoresExcel("actores.xls");
    HashMap<Asistente, String> asistentes = getAsistentesExcel("actores.xls");
    HashMap<Federativo, String> federativos = getFederativosExcel("actores.xls");
    //carga inicial acá
    SimpleDateFormat sdf = new SimpleDateFormat(FORMATO_FECHA_BASE);
    for(Jugador player: jugadores.keySet()){
        setQuery("insert into personas values ('" + player.getNumeroPasaporte() + "','" +
            + sdf.format(player.getFechaNacimiento()) + "','" + player.getNombre() +
            "','" + player.getApellido1() + "','" + player.getApellido2() + "')");
    }
    for(Jugador player: jugadores.keySet()){
        if(!"NUL".equals(jugadores.get(player))){
            setQuery("insert into jugadores values ('" + player.getNumeroPasaporte() + "','" + player.getNumeroCamisa() +
                "','" + player.getPuesto() + "','" + jugadores.get(player) + "')");
        }else{
            setQuery("insert into jugadores values ('" + player.getNumeroPasaporte() + "','" + player.getNumeroCamisa() +
                "','" + player.getPuesto() + "','" + null + "')");
        }
    }
}

```

Así pues, de esta forma, conseguimos llenar algunas de las tablas necesarias para comenzar a trabajar de manera efectiva, logrando así realizar pruebas en otros métodos, y tener la certeza de que funcionaban.

SOFTWARE DE MANEJO DE VERSIONES

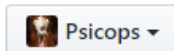
El simple hecho de tener acceso a un software de manejo de versiones, simplifica enormemente la carga de estar haciendo copias del proyecto, ya que, en caso de que algo falle, simplemente es necesario buscar la versión más actualizada, se descarga y se le aplican los cambios necesarios. Al principio es un poco difícil acostumbrarse, sin embargo, conforme pasa el tiempo, es mucho más sencillo hacerlo. Decidimos continuar haciendo uso de la plataforma GitHub, ya que, además de ser la más popular por el momento, es bastante simple de utilizar, y muy segura, por lo que representaba una opción perfecta para los principiantes. En su página, www.github.com, ofrece tutoriales básicos de uso, por lo cual no fue necesario buscar más información en otras páginas aparte.

Comenzar a trabajar en GitHub es tan sencillo como iniciar sesión o crear una nueva cuenta, en caso de no tener una, luego se necesita crear un repositorio, que es donde se comenzará a guardar el proyecto.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner



Repository name

Great repository names are short and memorable. Need inspiration? How about **jubilant-fiesta**.

Description (optional)

Para crear un repositorio simplemente se le debe dar un nombre, preferiblemente uno que sea acorde con lo que se va a realizar, además, se puede agregar una descripción, en donde se explicará, opcionalmente, los objetivos del proyecto.

Después de eso, se tienen varias formas para comenzar a subir los archivos del proyecto, entre ellas está la opción de usar la aplicación de escritorio, opción que nosotros usamos, en donde se utiliza las siguientes líneas de comando:

```
git remote add origin https://github.com/User_Name/Repository_Name.git
git push -u origin master
```

Al hacer uso de estos comandos, se sube al repositorio creado anteriormente, el cual se referencia en el primer comando.

Una vez se han subido los primeros archivos al repositorio, las cosas se simplifican aún más, ya que es posible hacer “Drag and Drop” directamente en el repositorio de la página de GitHub, por lo cual hacer actualizaciones de las versiones del proyecto es realmente simple, solo se necesita arrastrar el archivo que se desea actualizar y GitHub hará el resto. También es posible agregar una descripción para cada una de las versiones, de manera que se puedan explicar los cambios que se hicieron en el archivo actualizado, para llevar un mejor control sobre las versiones, por ejemplo, se puede tomar la descripción de la versión 4 del proyecto:



Psicops Versión 4 ...

- Se han corregido errores en algunas funcionalidades y operaciones del programa, que podían llegar a provocar incluso la caída del mismo.
- Se han añadido funciones faltantes.
- El Diccionario de Datos está en su etapa final. Pronto se podrá ejecutar y visualizar los atributos y dominio de cada tabla.

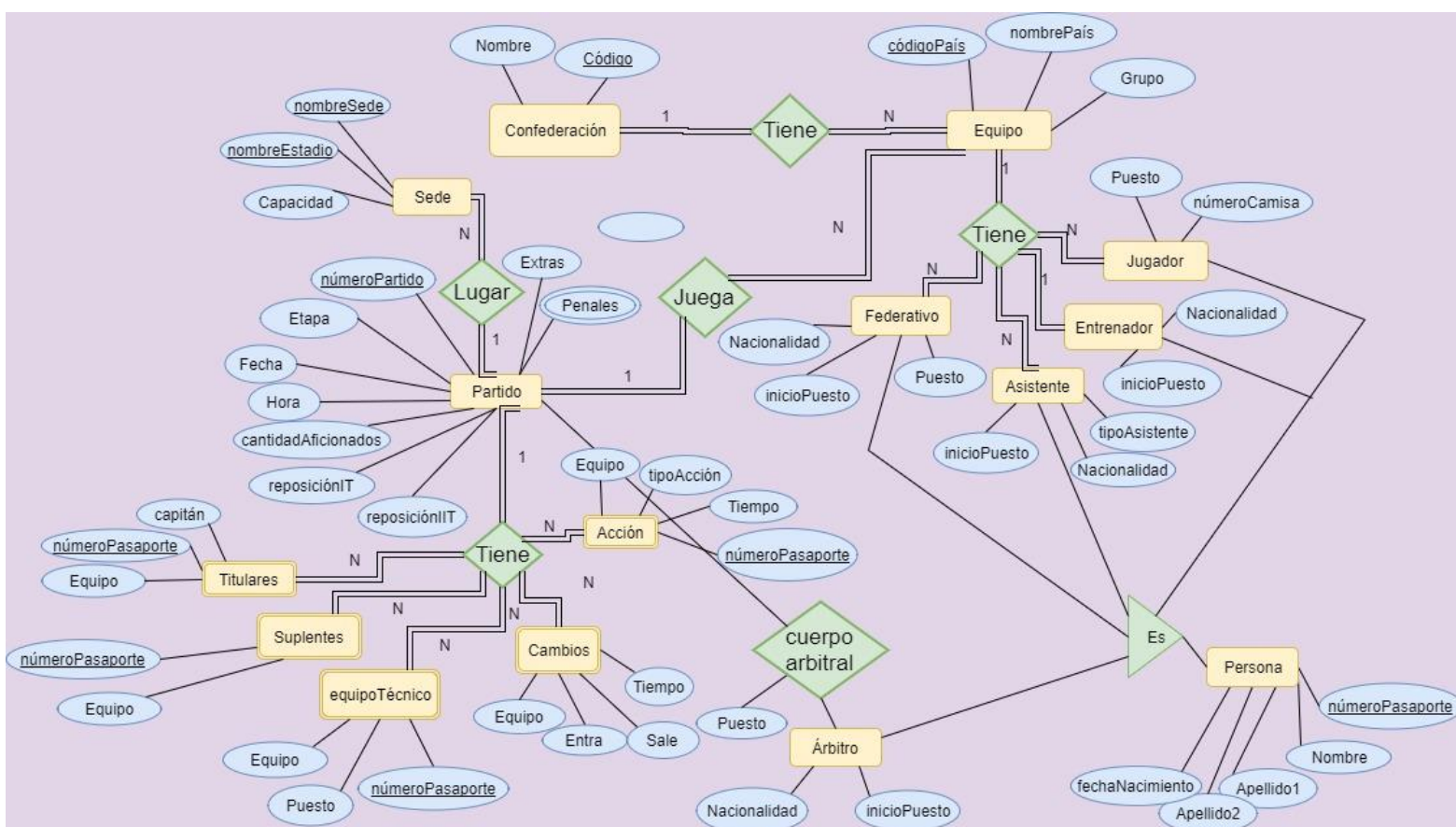
Actualmente se puede revisar el repositorio en el cual estuvimos subiendo todas las versiones del proyecto en la siguiente url:

<https://github.com/Psicops/BDProy2>

Desde esta dirección, se puede descargar la última versión del proyecto, con la cual se puede ejecutar el programa, además de que se puede leer una pequeña descripción del proyecto, y descargar los archivos necesarios para la ejecución correcta del programa.

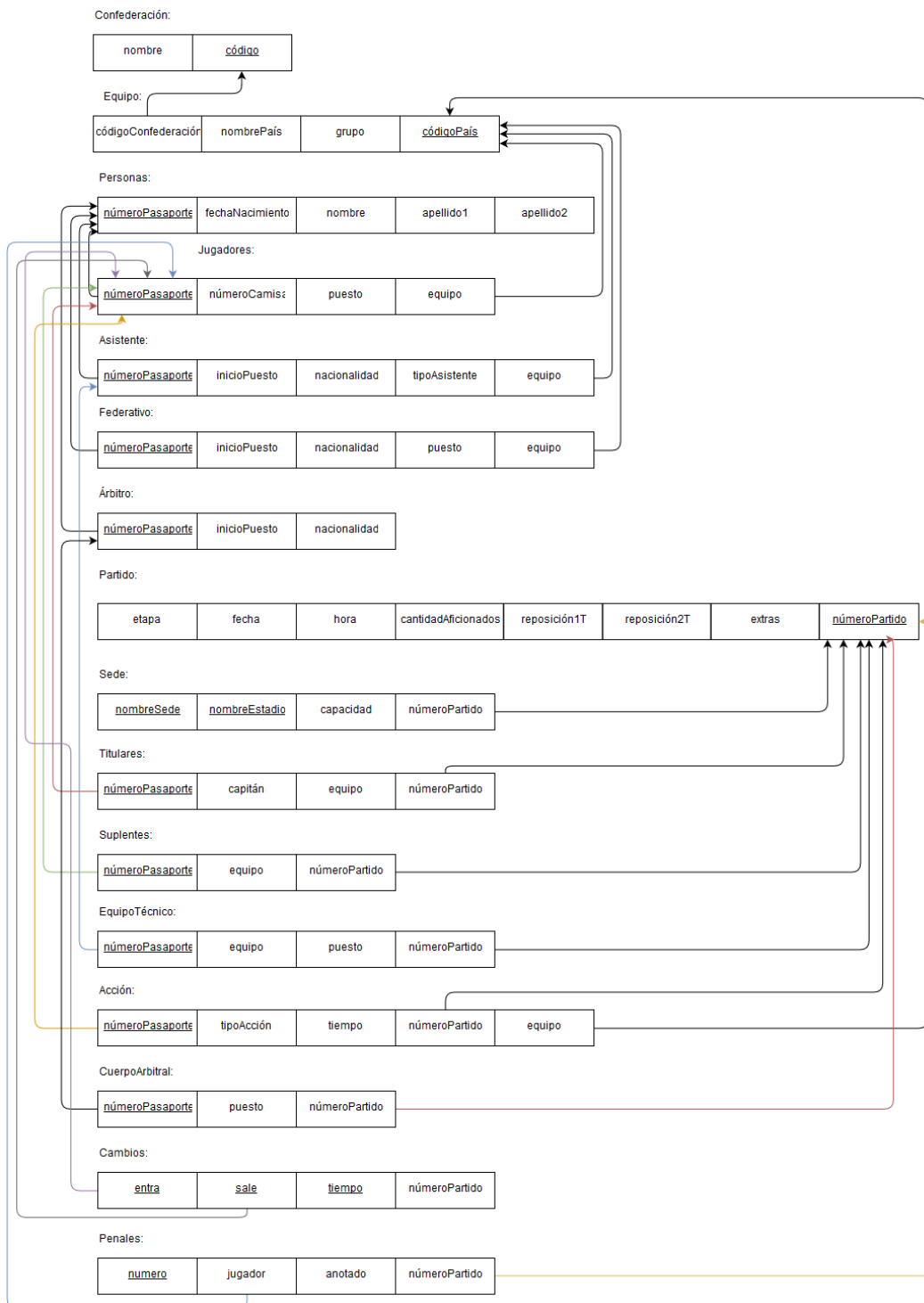
Diseño Conceptual

A continuación, se presentará el diseño conceptual utilizado para el desarrollo del proyecto.



Diseño Lógico

A continuación, se presentará el diseño lógico utilizado para el desarrollo del proyecto.



Diseño Físico

A continuación, se presentará el diseño físico utilizado para el desarrollo del proyecto.

```
create table CONFEDERACION(
    nombre varchar2(55) not null,
    codigo varchar2(10) check (codigo in ('Concacaf', 'UEFA', 'Conmebol', 'CAF', 'AFC','OFC'))primary key not null);

create table EQUIPO(
    codigoConfederacion varchar2(10) references CONFEDERACION(codigo) not null,
    nombrePais varchar2(40) not null,
    grupo varchar2(1) not null,
    codigoPais varchar2(3) primary key not null);

create table PERSONAS(
    numeroPasaporte varchar2(20) primary key not null,
    fechaNacimiento date not null,
    nombre varchar2(15) not null,
    apellido1 varchar2(15) not null,
    apellido2 varchar2(15) not null);

create table JUGADORES(
    numeroPasaporte varchar2(20) primary key references PERSONAS(numeroPasaporte) not null,
    numeroCamisa integer check(numeroCamisa between 1 and 99) not null,
    posicion varchar2(15) check (posicion in ('Portero', 'Defensa', 'Mediocampista', 'Delantero')) not null,
    equipo varchar2(3) references EQUIPO(codigoPais));

create table ASISTENTE(
    numeroPasaporte varchar2(20) primary key references PERSONAS(numeroPasaporte) not null,
    inicioPuesto date not null,
    nacionalidad varchar2(15) not null,
    tipoAsistente varchar2(25) check ( tipoAsistente in ('Entrenador', 'Técnico', 'Preparador Físico',
    'Médico', 'Psicólogo', 'Nutricionista', 'Administrativo', 'Delegado de Equipo')) not null,
    equipo varchar2(3) references EQUIPO(codigoPais));

create table FEDERATIVO(
    numeroPasaporte varchar2(20) primary key references PERSONAS(numeroPasaporte) not null,
    inicioPuesto date not null,
    nacionalidad varchar2(15) not null,
    puesto varchar2(15) check (puesto in ('Director', 'Vicepresidente', 'Secretario', 'Tesorero', 'Fiscal', 'Vocal')) not null,
    equipo varchar2(3) references EQUIPO(codigoPais));

create table ARBITRO(
    numeroPasaporte varchar2(20) primary key references PERSONAS(numeroPasaporte) not null,
    inicioPuesto date not null,
    nacionalidad varchar2(15) not null);

create table SEDE(
    nombreSede varchar2(40) not null,
    nombreEstadio varchar2(40) primary key not null,
    capacidad integer check (capacidad between 25000 and 150000) not null);
```



```

create table PARTIDOS(
    numeroPartido integer check(numeroPartido between 1 and 64) primary key not null,
    etapa varchar2(15) check(etapa in ('Fase de Grupos', 'Octavos de Final', 'Cuartos de Final',
        'Semifinales', 'Tercer Lugar', 'Final')) not null,
    fechaHora timestamp not null,
    cantidadAficionados integer not null,
    reposicion1T varchar2(10) not null,
    reposicion2T varchar2(10) not null,
    equipo1 varchar2(3) references EQUIPO(codigoPais) not null,
    equipo2 varchar2(3) references EQUIPO(codigoPais) not null,
    nombreEstadio varchar2(10) references SEDE(nombreEstadio) not null,
    extras varchar2(2) check (extras in('SI','NO')) not null);

create table TITULARES(
    numeroPasaporte varchar2(20) primary key references PERSONAS(numeroPasaporte) not null,
    capitan varchar(2)check (capitan in ('SI', 'NO')) not null,
    equipo varchar2(3) references EQUIPO(codigoPais) not null,
    numeroPartido integer references PARTIDOS(numeroPartido) not null);

create table SUPLENTES(
    numeroPasaporte varchar2(20) primary key references PERSONAS(numeroPasaporte) not null,
    equipo varchar2(3) references EQUIPO(codigoPais) not null,
    numeroPartido integer references PARTIDOS(numeroPartido) not null);

create table EQUIPOTECNICO(
    numeroPasaporte varchar2(20) primary key references PERSONAS(numeroPasaporte) not null,
    equipo varchar2(3) references EQUIPO(codigoPais) not null,
    puesto varchar2(20) check (puesto in ('Entrenador', 'Asistente Técnico', 'Médico', 'Delegado')) not null,
    numeroPartido integer references PARTIDOS(numeroPartido) not null);

create table ACCION(
    numeroPasaporte varchar2(20) references PERSONAS(numeroPasaporte) not null,
    tipoAccion varchar2(20) check (tipoAccion in ('Gol', 'Tarjeta Amarilla', 'Tarjeta Roja')) not null,
    tiempo varchar2(5) not null,
    numeroPartido integer references PARTIDOS(numeroPartido) not null,
    equipo varchar2(3) references EQUIPO(codigoPais) not null)

alter table ACCION add primary key (numeroPasaporte, tiempo, numeroPartido);

create table CUERPOARBITRAL(
    numeroPasaporte varchar2(20) primary key references PERSONAS(numeroPasaporte) not null,
    puesto varchar2(15) check(puesto in ('Principal', 'Guardalíneas', 'Cuarto Árbitro', 'Quinto Árbitro')) not null,
    numeroPartido integer references PARTIDOS(numeroPartido) not null);

create table CAMBIOS(
    entra varchar2(20) references PERSONAS(numeroPasaporte) not null,
    sale varchar2(20) primary key references PERSONAS(numeroPasaporte) not null,
    tiempo varchar2(5) not null,
    numeroPartido integer references PARTIDOS(numeroPartido) not null);

create table PENALES(
    numeroPenal varchar2(2) not null,
    anotado varchar2(2) check (anotado in ('SI', 'NO')) not null,
    jugador varchar2(20) references PERSONAS(numeroPasaporte) not null,
    numeroPartido integer references PARTIDOS(numeroPartido) not null)

alter table PENALES add primary key (jugador, numeroPartido);

```

Programas Almacenados

Básicamente son procedimientos almacenados físicamente en una base de datos, Su implementación varía de un gestor de bases de datos a otro. La ventaja de un procedimiento almacenado es que, al ser ejecutado, en respuesta a una petición de usuario, es ejecutado directamente en el motor de bases de datos, el cual usualmente corre en un servidor separado. Como tal, posee acceso directo a los datos que necesita manipular y sólo necesita enviar sus resultados de regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes. Entre éstos podemos encontrar los procedimientos, triggers y funciones, donde todos ellos se ejecutan dentro del mismo motor de base de datos. Por ejemplo, nosotros creamos un trigger que, al intentar cambiar el equipo de un asistente, federativo o jugador, muestre un mensaje en la pantalla, con el equipo viejo y el equipo nuevo para la persona en cuestión. Esta es la forma de implementación:

```
create or replace trigger trig_cambioEquipoA
before update on asistente for each row
DECLARE
BEGIN
    dbms_output.put_line('Equipo viejo: ' || :old.equipo);
    dbms_output.put_line('Equipo nuevo: ' || :new.equipo);
END;
/
```

De esta manera, se ejecutará antes de que se efectúe el cambio, mostrando los equipos dentro del mismo motor de Oracle.

Otro ejemplo que podemos usar, son las funciones, donde hemos creado una función que, al ser llamada, retorne por si misma el grupo al cual pertenece un equipo, como se muestra a continuación:

```
create or replace function getGrupoTeam (idP in varchar2)
return varchar2
is grupoP varchar2(2);
begin
    select grupo into grupoP
    from equipo
    where codigoPais = idP;
    return(grupoP);
end;
/
```


Transacciones, vistas e índices.

Cuando hablamos de transacciones, se hace referencia a un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica. Es decir, es capaz de mantener la integridad de los datos, haciendo que estas transacciones no puedan finalizar en un estado intermedio. Cuando por alguna causa el sistema debe cancelar la transacción, empieza a deshacer las órdenes ejecutadas hasta dejar la base de datos en su estado inicial. A esta sucesión de órdenes, se le llama “Rollback”. Cuando una transacción llega a ser completada de manera efectiva, se ejecuta un “Commit” el cual guarda las transacciones, validando los cambios.

En el caso de las vistas, es una consulta que se presenta como una tabla a partir de un conjunto de tablas en una base de datos relacional. ¿Que quiere decir esto? Bueno, esto significa que se pueden almacenar resultados de una consulta realizada a la base de datos, en una tabla no temporal, si no virtual, de manera que pueda ser utilizada para realizar guardar datos o incluso realizar consultas futuras. Nosotros, por ejemplo, hemos implementado una vista en la cual se guardan los datos de personas asistentes, que no pertenecen a ningún equipo, de manera tal, que todas las tuplas en esta vista, pueden ser integradas a un equipo en caso de ser necesario.

```
create view asistentes_libres as
select * from asistente
where equipo is null;
```

Los índices, por otra parte, son una estructura de datos que mejora la velocidad de las operaciones, por medio de identificador único de cada fila de una tabla, permitiendo un rápido acceso a los registros de una tabla en una base de datos. A continuación, veremos la implementación de un índice simple en la tabla “personas”, añadiendo este índice en la columna de “nombres”:

```
create index nombres on personas (nombre);
```

Conclusiones del Trabajo

Este proyecto fue especialmente útil para terminar de comprender el lenguaje de Oracle más a fondo, ya que tuvimos la necesidad de indagar más a fondo en las instrucciones SQL de este motor, con tal de obtener los resultados deseados en cuanto al programa se refiere. Sin embargo, se presentaron algunos problemas, especialmente a la hora de llenar las tablas con información de un EXCEL, tal y como se mencionó anteriormente. Afortunadamente, estuvimos investigando y logramos encontrar un método con el cual pudimos acceder a los datos de una forma simple y sin errores.

Otro problema que tuvimos fue a la hora de pasar información de una base de datos en SQL Server a nuestra base de datos en Oracle. Al investigar, nos dimos cuenta que debíamos hacer uso de una biblioteca que nos permitiría acceder a la base de datos de SQL Server, desde el SQL Developer de Oracle. Gracias a esta biblioteca, el paso de información entre ambos motores, se simplificó bastante, ya que, de no encontrar otra manera, habríamos leído y almacenado los datos de SQL Server en Netbeans, para finalmente pasarlos a la base de datos de Oracle por medio de consultas.

Aunque fue un proyecto difícil de desarrollar, podemos decir que realmente aprendimos mucho con él, ya que llegamos a comprender el lenguaje usado en SQL de forma más profunda, además de la necesidad de aplicar una conexión entre dos lenguajes prácticamente distintos, así que, con las investigaciones realizadas al respecto adquirimos conocimientos de los cuales carecíamos anteriormente, conocimientos que serán útiles en futuros proyectos, y que facilitarán su desarrollo.

Rúbrica

Concepto	Puntos	Puntos obtenidos	Avance %	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		80	Faltó la implementación de algunos métodos
Carga Inicial: lista de equipos	5		100	
Carga Inicial: lista jugadores	5		100	
Validaciones de datos y procesos	10		100	
Equipos participantes por Confederación	3		100	
Informe oficial del partido	12		60	Faltó parte de la implementación
Grupos y Clasificación	5		60	Faltó la implementación de algunos métodos
Tabla General Posiciones	5		40	Faltó la implementación de algunos métodos
Tabla Goleadores	5		100	
SQL: Consultas en vivo	10*			
Ayuda	5		100	
TOTAL	100*			
Partes desarrolladas adicionalmente				