

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias
Introducción a la Programación y Computación 2

ENTREGABLE 1

Nombre: Daniel Estuardo Chicas Carías

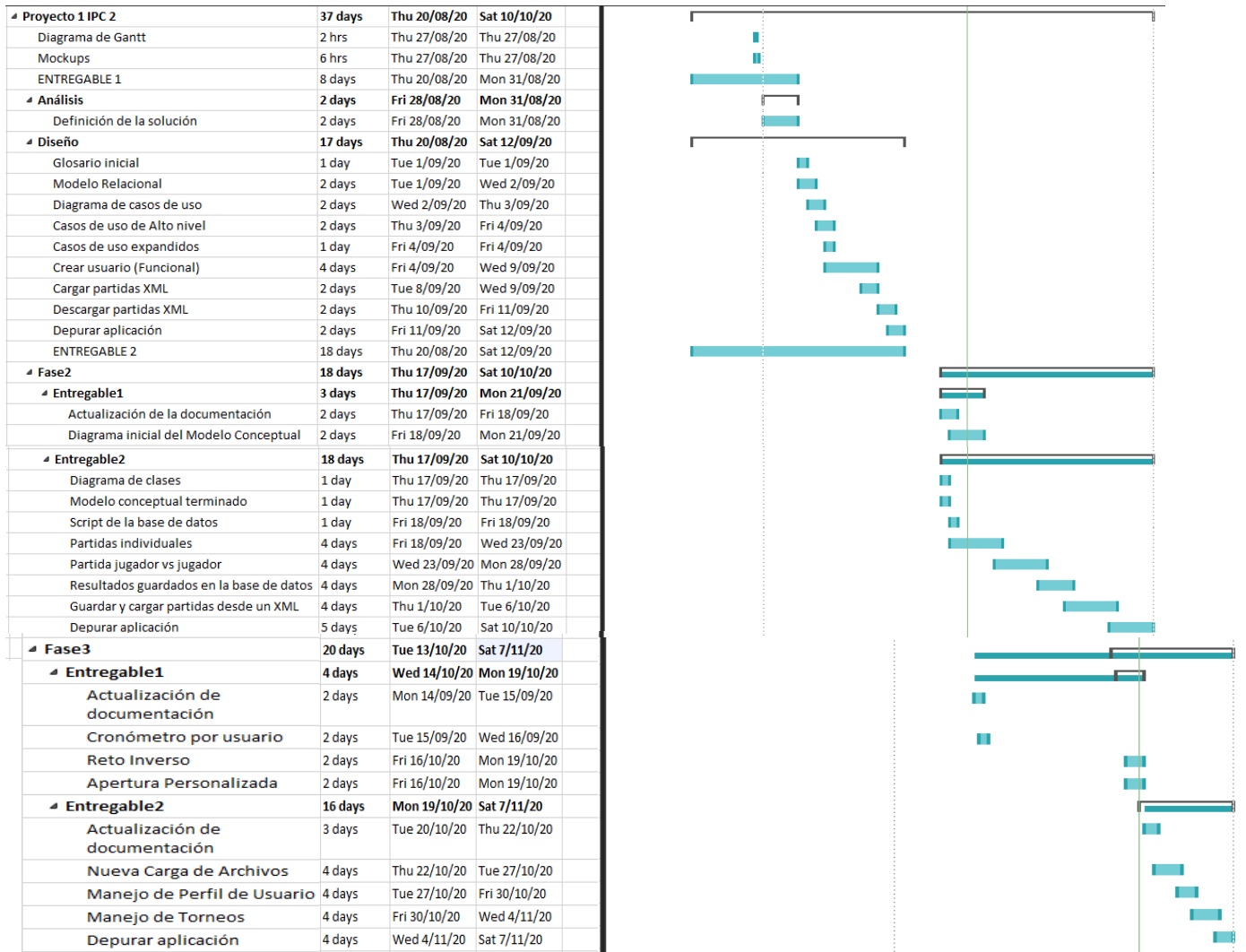
Registro Académico: 201807079

Auxiliar: Tanya Muhun

Sección: A

Fecha: 19/10/2020

1. PLANIFICACIÓN DEL PROYECTO UTILIZANDO DIAGRAMA DE GANTT



2. LOOK & FEEL DE LA SOLUCIÓN

**¡Juega un poco para distraerte
y olvidar todos los problemas! :D**

Iniciar sesión:

Usuario:

Contraseña:

*Datos erróneos.
Inténtelo nuevamente.*

Crear cuenta

Empezar a Jugar

*Ingrese los datos
solicitados*

CREAR CUENTA

Nuevo Usuario:

Nombres:

Apellidos:

Nombre Usuario:

Contraseña:

Fecha de Nacimiento:

País:

Correo Electrónico:

*Nombre de usuario
existente*

Crear cuenta

MODOS DE JUEGO

Cargar partida

Partida Individual

Contra Otro Jugador

Torneo

Editar Usuario

Salir

CARGA DE ARCHIVO .xml

Ruta del archivo:

Cargar

Salir

EDITAR ESTADO DE USUARIO

Nombre de estado:

Cambiar estado a:

Estado	▼
Activo	
Inactivo	

Cargar

Salir

Torneo

Crear un torneo

Nombre del Torneo:

Número de Integrantes:

4	▼
8	
16	

*Este torneo ya existe.
Ingrese uno nuevo.*

Crear

Unirse a un torneo

Nombre del Torneo:

*Nombre incorrecto
Inténtelo nuevamente*

Unirme

Volver

Jugador1

Movimientos 2:

100%

Mostrar Ficha

An 8x8 grid with columns labeled A through H and rows labeled 1 through 8. The grid is shown with a light blue background and a green border. The columns are labeled A, B, C, D, E, F, G, H from left to right. The rows are labeled 1, 2, 3, 4, 5, 6, 7, 8 from top to bottom. In the center of the grid, four circles are placed in a 2x2 arrangement. The circles are colored as follows: the circle at D4 is white, the circle at E4 is black, the circle at D5 is black, and the circle at E5 is white.

Guardar

Volver a jugar

Nombre de jugador:

Usuario:

© 2006 The Authors

Apellido:

© 2006 The Authors

No.	Tipo de Partida	Estado	Movimientos	Fecha

Juegos ganados:

Volver

LINK DEL REPOSITORIO DE GITHUB: https://github.com/Daniel-Chicas/IPC2_FASE1_201807079.git

3. DEFINICIÓN DE LA SOLUCIÓN

3.1. OBJETIVOS

3.1.1. GENERAL

Desarrollar un sistema de software a través de los conceptos de análisis y diseño de sistemas, utilizando el lenguaje c# en el entorno de aplicaciones web "ASP.NET".

3.1.2. Específicos

- *Comprender la importancia de las fases de análisis y diseño en el desarrollo de un sistema de información.*
- *Utilizar el modelo relacional para gestionar los datos que soportan un sistema de información.*
- *Desarrollar casos de uso de alto nivel para dimensionar el sistema a desarrollar.*
- *Definir la planificación que permitirá la implementación del software.*
- *Utilizar el modelo conceptual para describir el contexto del problema.*
- *Utilizar el diagrama de clases para diseñar soluciones de software.*
- *Definir la planificación que permitirá terminar la implementación del software.*

3.2. ALCANCES

Realizar la tercera fase del proyecto del curso: Introducción a la programación y computación 2. El cual consiste en:

- *Manejar usuarios a través de SQL Server, mediante un login al cuál se podrá ingresar si y solo si el usuario ha sido creado en el apartado "crear usuario".*
- *Cargar y guardar partidas por medio de documentos .xml, el cuál tendrá el color de la ficha y la posición de esta en el tablero.*
- *Jugar partidas individuales, contra otro jugador, crear campeonatos o unirse a los que ya estén existentes. Para jugar contra otro jugador será posible enviar solicitudes o dejar las partidas abiertas para quien quiera unirse. Los campeonatos se realizarán en un formato de muerte súbita, tomando en cuenta el punteo que obtuvo cada jugador para poder determinar el ganador. Cada torneo tiene un nombre único.*
- *Poder ver los datos de cada usuario por medio de la vista "perfil", en el cuál permitirá al usuario ver las partidas ganadas, perdidas y en los torneos que ha participado y el resultado final de estos.*
- *Jugar con distintos colores para cada jugador, así como poder definir el tamaño del tablero en el cuál desea jugar.*

- *Nuevos modos de juego. En este se tendrán opciones como “reto inverso”, en el cuál el usuario podrá jugar con distintos colores y las reglas de esta modalidad son totalmente distintas al juego clásico.*

3.3. REQUERIMIENTOS INICIALES DEL SISTEMA

3.3.1. REQUERIMIENTOS FUNCIONALES

REQUERIMIENTOS FUNCIONALES			
Código	Sección	Clasificación	Descripción
R.1	Login	Evidente/oculta	Validar en la base de datos que el usuario y contraseña ya hayan sido creados.
R.2	Crear usuario	Evidente	Agregar nuevos usuarios a la base de datos.
R.3	Cargar Partida	Evidente	Cargar datos de un documento .xml.
R.4	Partida Individual	Evidente	Jugar contra la máquina.
R.5	Contra Otro jugador	Evidente	Jugar partida contra otro jugador, al cuál se le envió una solicitud.
R.6	Campeonatos	Evidente/Oculta	Crear o unirse a campeonatos de 4, 8 o 16 equipos. Cada equipo tendrá 3 integrantes. Para unirse a una partida debe validarse que exista en la base de datos. Y que el usuario esté invitado si el campeonato es privado.
R.7	Editar Usuarios	Evidente/Oculta	Habilitar o deshabilitar otros usuarios.
R.8	Salir	Evidente	Sale del juego y regresa al login.
R.9	Perfil de Usuario	Evidente	La plataforma mostrará al usuario las estadísticas que este ha tenido durante su estadía en el juego. Se le mostrará al usuario las partidas que ha jugado, el resultado que ha obtenido en estas, las fechas en las que ha jugado y el color de las fichas que tenía.
R.10	Guardar	Evidente/Oculta	Guarda la partida por medio de un documento xml.
R.11	Volver a Jugar	Evidente	Vuelve a poner el tablero en blanco para jugar nuevamente.
R.12	Salir	Evidente	Regresa a la pantalla “Modos de juego”.
R.13	Reto Inverso	Evidente	Jugar con reglas inversas que en el juego Othello Clásico.
R.14	Apertura Personalizada	Evidente	El usuario empezará la partida colocando sus colores en el centro del tablero para iniciar.
R.15	Cronómetro	Evidente	Cada jugador tendrá un cronómetro para tener un estimado del tiempo en que tarda en responder la jugada.

3.3.2. ATRIBUTOS DEL SISTEMA

ATRIBUTOS DEL SISTEMA		
No.	Tipo	¿Por qué?
1	Intuitivo	Los botones están bien posicionados y cada uno está etiquetado de tal manera que el usuario al ver el nombre del botón sepa la acción que este pueda realizar.
2	Facilidad de uso	No se hace nada externo a la plataforma. Todo se maneja dentro de la ella, por lo que solo hay que ingresar y disfrutar de ella.
3	Tolerable a Fallos	Todos los fallos que se pueden tener al navegar en la plataforma están controlados, por lo tanto, no se detendrá la ejecución de esta. Se mostrará un mensaje de tipo alerta si se realiza una acción incorrecta, como por ejemplo cargar un archivo distinto al formato .xml.
4	Plataforma	Microsoft SQL Server management Studio 18, Windos 10 , VisualStudio 2019 con el proyecto, Chrome.

4. GLOSARIO INICIAL

4.1. ASIDE:

Representa una sección de una página que consiste en contenido que está indirectamente relacionado con el contenido principal del documento.

4.2. ASP.NET:

Es un entorno para aplicaciones web desarrollado y comercializado por Microsoft.

4.3. C#:

Es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft como parte de su plataforma .NET

4.4. Casos de Uso:

Es la descripción de una acción o actividad. Un diagrama de caso de uso es una descripción de las actividades que deberá realizar alguien o algo para llevar a cabo algún proceso.

4.5. Commit:

En el contexto de la ciencia de la computación y la gestión de datos, a la idea de confirmar un conjunto de cambios provisionales de forma permanente.

4.6. Controlador:

Actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

4.7. Database:

Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

4.8. DDL:

Es un lenguaje de programación para definir una estructura de datos.

4.9. Diagrama de Gantt:

Es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

4.10. Entity Framework:

Es un asignador relacional de objetos (ORM) que permite a los desarrolladores de .NET trabajar con datos relacionales utilizando objetos específicos del dominio.

4.11. Github:

es una forja para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador.

4.12. JQuery:

Nos permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web

4.13. Look & Feel:

Es una metáfora utilizada dentro del entorno de marketing para poder dar una imagen única a los productos, incluyendo áreas como el diseño exterior, trade dress, la caja en que se entrega al cliente, etc.

4.14. Modelo Relacional:

Es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos.

4.15. MVC:

Es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones.

4.16. Othello:

Juego de mesa entre dos personas.

4.17. Repositorio:

Es un espacio centralizado donde se almacena, organiza, mantiene y difunde información digital, habitualmente archivos informáticos, que pueden contener trabajos científicos, conjuntos de datos o software.

4.18. SQL:

Es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.

4.19. Usabilidad:

Cualidad de la página web o del programa informático que son sencillos de usar porque facilitan la lectura de los textos, descargan rápidamente la información y presentan funciones y menús sencillos, por lo que el usuario encuentra satisfechas sus consultas y cómodo su uso.

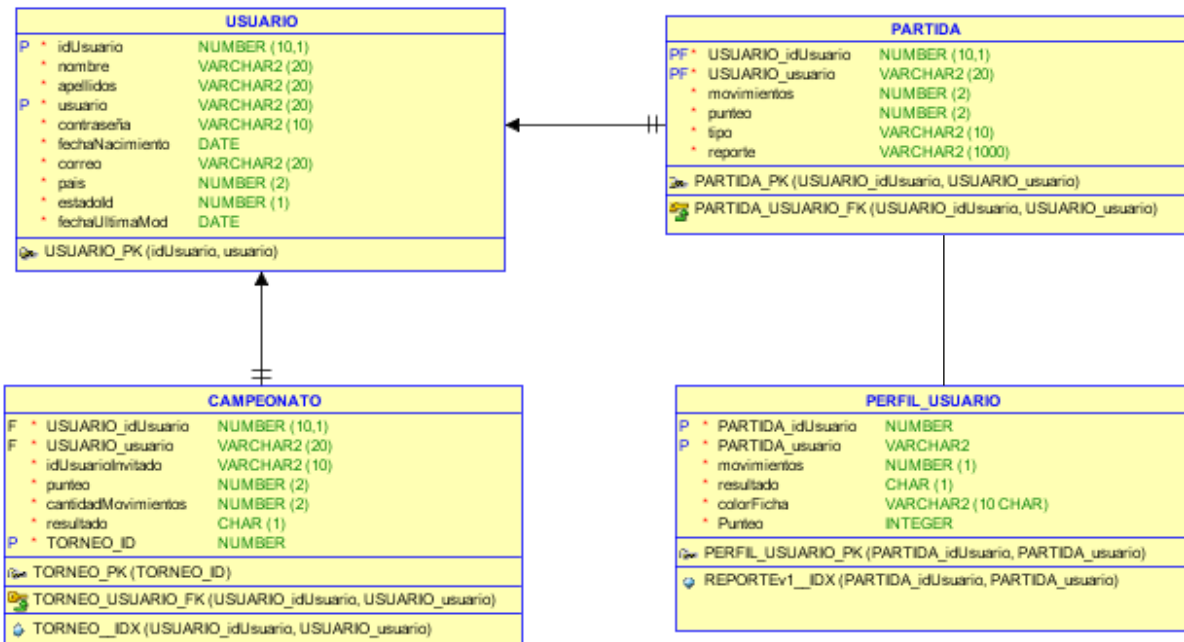
4.20. Visual Studio:

Es un entorno de desarrollo integrado para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic . Es un IDE.

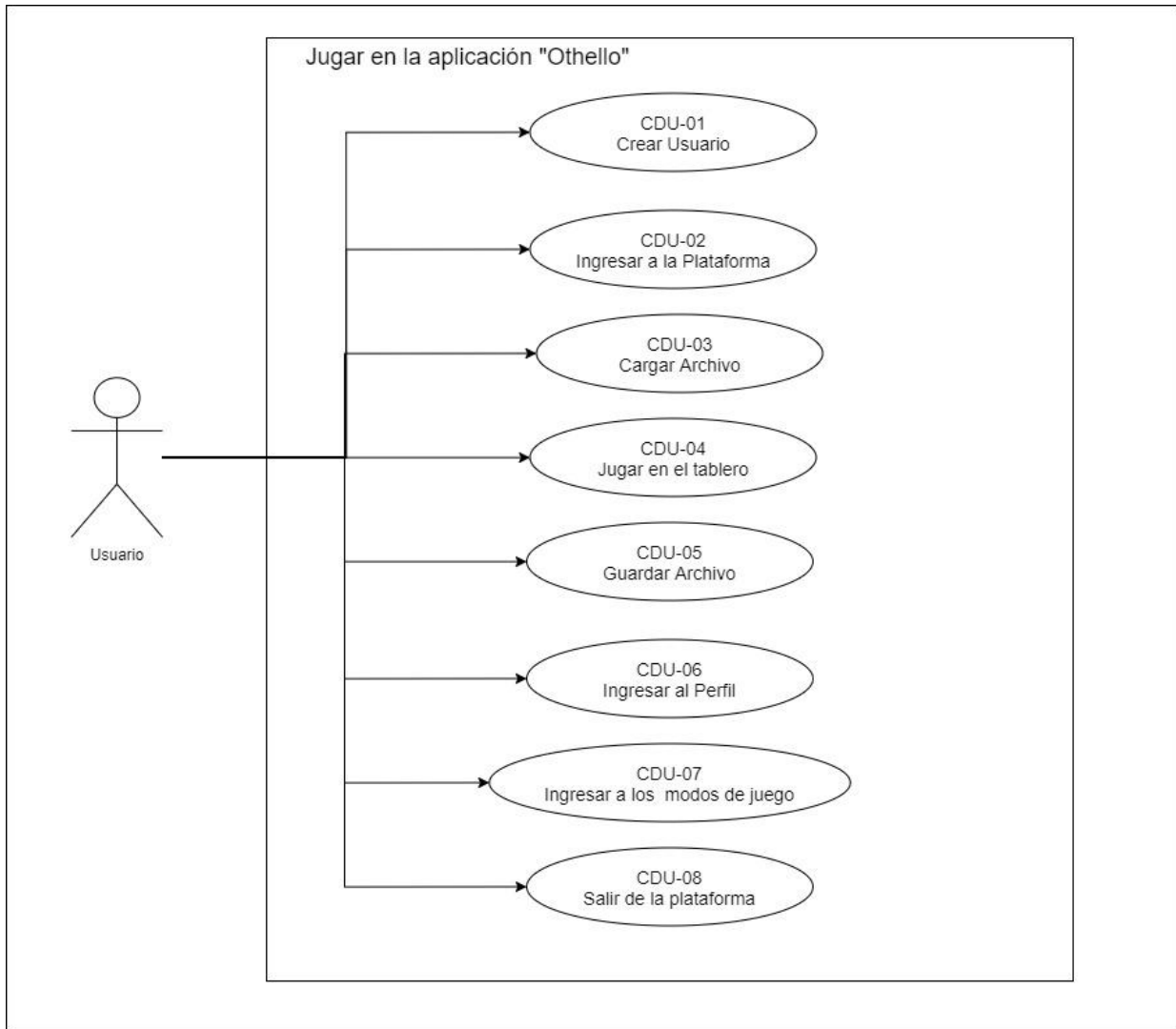
4.21. JavaScript:

lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

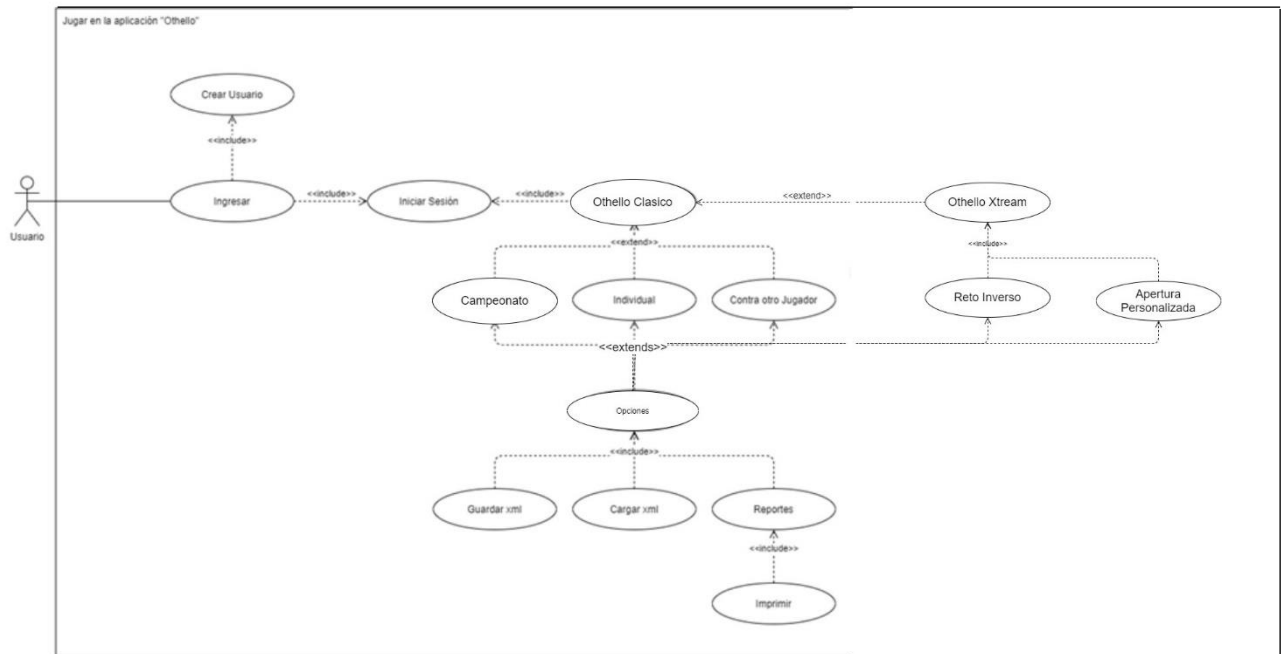
5. MODELO RELACIONAL



6. DIAGRAMA DE CASOS DE USO



7. RELACIONES DEL DIAGRAMA DE CASOS DE USO



8. CASOS DE USO DE ALTO NIVEL

CASOS DE USO DE ALTO NIVEL				
No	Caso de uso	Actores	Tipo	Descripción
1.	CDU-01 Crear usuario	Usuario	Primario	La persona crea su cuenta para poder ingresar a la plataforma. Al crearla se le redireccionará al login.
2.	CDU-02 Ingresar a la plataforma	Usuario	Primario	Se le solicita al usuario su nombre de usuario y contraseña para poder ingresar, si los datos son correctos, se acepta el ingreso y se muestran el menú de opciones.
3.	CDU-03 Cargar archivo	Usuario	Opcional	Se pide la ruta del archivo que desea cargar a la plataforma, si la ruta es correcta, se carga el archivo y se muestra el tablero.
4.	CDU-04 Jugar en el tablero	Usuario	Primario	Se muestra el tablero y el usuario selecciona la posición del tablero que desea marcar de su color.
5.	CDU-05 Guardar archivo	Usuario	Secundario	El usuario presiona el botón "Guardar" para poder almacenar en un archivo xml todas las posiciones seleccionadas con su color.
6.	CDU-06 Ingresar al Perfil	Usuario	Secundario	El usuario podrá ingresar a su perfil para ver sus resultados durante la plataforma. Podrá consultar sus partidas ganadas, partidas perdidas, torneos en los que ha participado y el resultado que ha obtenido en estos.
7.	CDU-07 Ingresar a los modos del juego	Usuario	Primario	El usuario podrá elegir el modo en el que desea jugar. Para esto existen 2 modos, los cuales son: clásico y Xtream. En el juego clásico se tienen las reglas clásicas del juego, los colores básicos (blanco y negro), mientras que en el Xtream el usuario puede elegir entre 10 opciones de colores los colores de su preferencia y elegir el modo "reto inverso", en el cual cambian las reglas del juego.
8.	CDU-08 Salir de la plataforma	Usuario	Opcional	Si el usuario está en el tablero se muestra el botón "Salir" el cuál lo redirecciona al menú de los modos de juego y luego dará clic en el botón salir para poder salir de la plataforma y regresar al login.

9. CASOS DE USO EXPANDIDOS

- **Caso de Uso:** CDU1- Crear Usuario
- **Actores:** Usuario (iniciador)
- **Propósito:** Crear un nuevo usuario en la base de datos de la plataforma.
- **Resumen:** La persona crea su cuenta para poder ingresar a la plataforma. Al crearla se le redireccionará al login.
- **Tipo:** Primario, Real.
- **Referencia Cruzada:** R.1
- **[Sección Principal]**
 - **Curso Normal de Eventos**
 1. El usuario ingresa a la plataforma.
 2. La plataforma le solicita al usuario el nombre de usuario y contraseña para poder ingresar.
 3. El usuario no tiene cuenta, por lo que va a la sección “Crear nuevo usuario”.
 4. Ingresa sus datos a la plataforma y crea su nuevo usuario.
 5. La plataforma lo regresa nuevamente al login.
 - **Cursos Alternos:**
 1. Línea 4: Se ingresa un usuario que ya existe en la plataforma. Indica error.

-
- **Caso de Uso:** CDU-02 Ingresar a la Plataforma
 - **Actores:** Usuario (iniciador)
 - **Propósito:** Poder ingresar a la plataforma con los datos del usuario.
 - **Resumen:** Se le solicita al usuario su nombre de usuario y contraseña para poder ingresar, si los datos son correctos, se acepta el ingreso y se muestran el menú de opciones.
 - **Tipo:** Principal, Esencial.
 - **Referencia Cruzada:** R.1, CDU-01.
 - **[Sección Principal]**
 - **Curso Normal de Eventos**
 1. El usuario ingresa sus datos en el login.
 2. La plataforma valida que la contraseña y el usuario existan en la base de datos.
 3. Entra a la plataforma y se muestran los modos de juego, siendo estos: “cargar partida”, “Partida Individual”, “Contra otro jugador”, “Torneos”, “Cambiar estados de otros usuarios”, “salir”.
 - **Cursos Alternos:**
 1. Línea 1: Se ingresa el usuario o la contraseña incorrecta. Indica error.
-

-
- **Caso de Uso:** CDU-03 Cargar Archivo
 - **Actores:** Usuario (iniciador)
 - **Propósito:** Cargar partidas guardadas en un archivo xml a la plataforma para poder continuar con el juego.
 - **Resumen:** Se pide la ruta del archivo que desea cargar a la plataforma, si la ruta es correcta, se carga el archivo y se muestra el tablero.
 - **Tipo:** Opcional, Real.
 - **Referencia Cruzada:** R.1, R.3, CDU-02
 - **[Sección Principal]**
 - **Curso Normal de Eventos**
 1. El usuario selecciona la opción “Cargar partida”.
 2. Se redirecciona a la pagina en donde colocará la ruta en la cual se encuentra el archivo.
 3. Selecciona la opción “Cargar”.
 4. Se muestra el tablero con las fichas en las posiciones que indica el archivo xml.
 - **Cursos Alternos:**
 1. **Línea 2:** Se coloca una ruta de un archivo distinto a un xml. Indica error.
 2. **Línea 3:** No da clic en el botón. No hace nada.
-

- **Caso de Uso:** CDU-04 Jugar en el tablero.
 - **Actores:** Usuario (iniciador), jugadores invitados.
 - **Propósito:** Jugar solo, competir con otros usuarios, continuar partidas guardadas.
 - **Resumen:** Se muestra el tablero y el usuario selecciona la posición del tablero que desea marcar de su color.
 - **Tipo:** Primario, Real.
 - **Referencia Cruzada:** R.3, R.4, R.5, R.6, CDU-02
 - **[Sección Principal]**
 - **Curso Normal de Eventos**
 1. El usuario selecciona cualquiera de las opciones de la sección “Modos de juego” (excepto cargar datos).
 2. Se muestra el tablero y puede jugar.
 - **Cursos Alternos:**
 1. El usuario selecciona “cargar datos”. Redirecciona a otra pantalla, no al tablero.
-

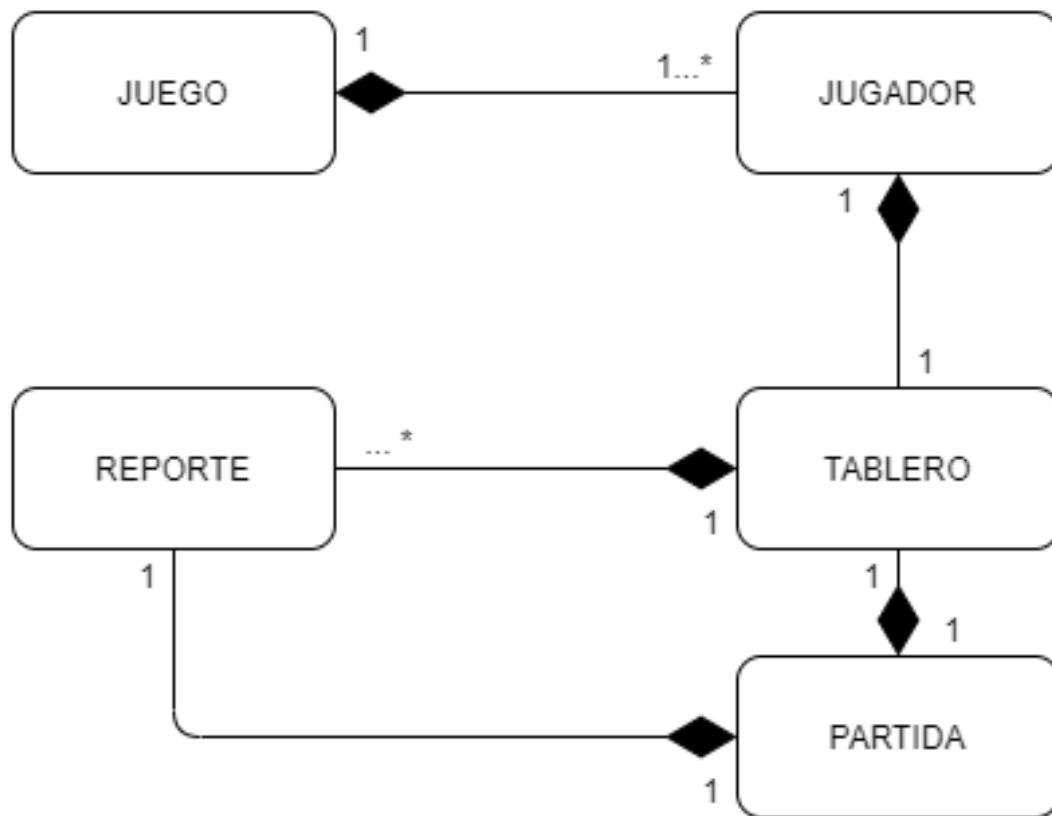
-
- **Caso de Uso:** CDU-05 Guardar archivo
 - **Actores:** Usuario (iniciador)
 - **Propósito:** Guardar en un archivo .xml los últimos movimientos que hizo el usuario para que después pueda continuar con el juego.
 - **Resumen:** El usuario presiona el botón "Guardar" para poder almacenar en un archivo xml todas las posiciones seleccionadas con su color.
 - **Tipo:** Opcional, Real.
 - **Referencia Cruzada:** R.1, R.4, R.5, R.6, R.10, CDU-02, CDU-04
 - **[Sección Principal]**
 - **Curso Normal de Eventos**
 1. El usuario ingresa al tablero y juega en cualquiera de los modos disponibles.
 2. Presiona el botón guardar.
 3. El usuario sale de la partida.
 - **Cursos Alternos:**
 1. **Línea 2:** El usuario no presiona el botón de la partida y sale sin guardar. La plataforma no guarda el programa.
-

- **Caso de Uso:** CDU-06 Ingresar al Perfil
 - **Actores:** Usuario (iniciador)
 - **Propósito:** Mostrar al usuario las partidas que ganó, perdió o empate a través de un formato dentro de la aplicación.
 - **Resumen:** El usuario podrá ver sus estadísticas que ha obtenido durante su estadía dentro de la plataforma. Podrá consultar la cantidad de partidas ganadas, perdidas y torneos en lo que ha participado y resultados obtenidos en estos.
 - **Tipo:** Secundario, Esencial.
 - **Referencia Cruzada:** R.1, R.4, R.5, R.6, R.9, R.10, CDU-02, CDU-04
 - **[Sección Principal]**
 - **Curso Normal de Eventos**
 1. El usuario ingresa a la plataforma.
 2. Selecciona la opción "Perfil".
 3. Se muestra el perfil del usuario con los datos mencionados.
 - **Cursos Alternos:**
 1. Si el usuario es nuevo y no ha jugado ni una vez o no sube ningún archivo, no habrá reporte que mostrar. Indicará error.
-

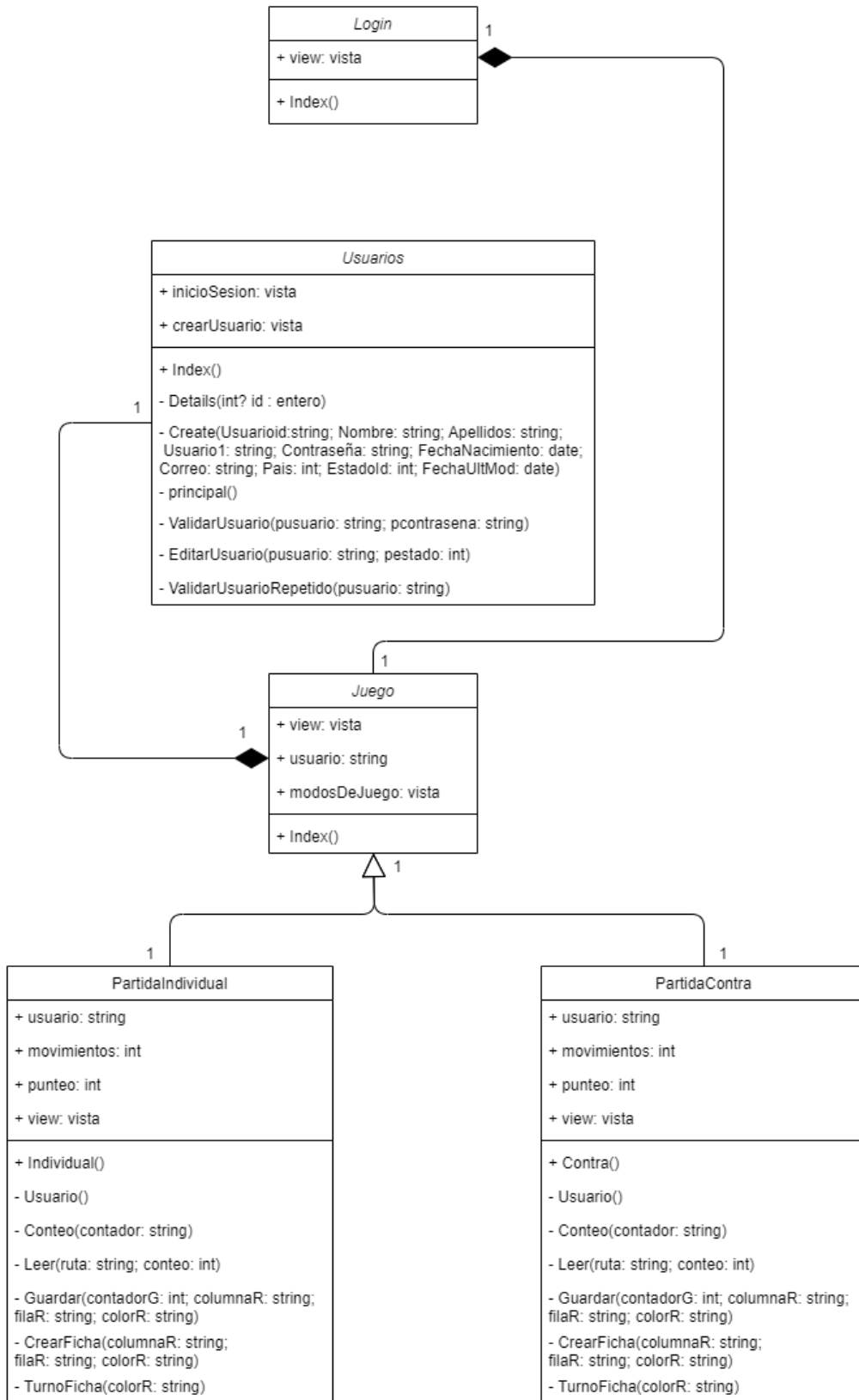
-
- **Caso de Uso:** CDU-07 Ingresar a los modos del juego
 - **Actores:** Usuario (iniciador)
 - **Propósito:** Jugar con distintas reglas y colores de fichas dentro de la aplicación y salir de lo común para que el usuario tenga algo más divertido.
 - **Resumen:** El usuario puede elegir entre el modo clásico o el modo Xtream, en el cual cambian las reglas del juego y el usuario puede escoger los colores de sus fichas.
 - **Tipo:** Secundario, Esencial.
 - **Referencia Cruzada:** R.1, R.13, R.14, CDU-02, CDU-04
 - **[Sección Principal]**
 - **Curso Normal de Eventos**
 1. El usuario ingresa a la plataforma.
 2. Selecciona la opción del modo en el que desea jugar (Othello Clásico u Xtream).
 3. Se muestran los tipos de juego que puede acceder dentro de estos modos.
 4. Escoge uno de los tipos de juego.
 5. Comienza a jugar.
 - **Cursos Alternos:**
 1. El usuario se confunde y selecciona el modo que no quería.
 2. Selecciona la opción “Salir” para regresar al menú anterior.
-

- **Caso de Uso:** CDU-08 Salir de la plataforma
 - **Actores:** Usuario (iniciador)
 - **Propósito:** Cerrar sesión y regresar al login.
 - **Resumen:** Si el usuario está en el tablero se muestra el botón “Salir” el cuál lo redirecciona al menú de los modos de juego y luego dará clic en el botón salir del menú para poder salir de la plataforma y regresar al login.
 - **Tipo:** Opcional, esencial.
 - **Referencia Cruzada:** R.1, R.2, R.12, CDU-02
 - **[Sección Principal]**
 - **Curso Normal de Eventos**
 1. El usuario selecciona la opción salir hasta que llega al login.
 - **Cursos Alternos:**
 1. El usuario no desea salir, por lo que la plataforma seguirá ejecutándose.
-

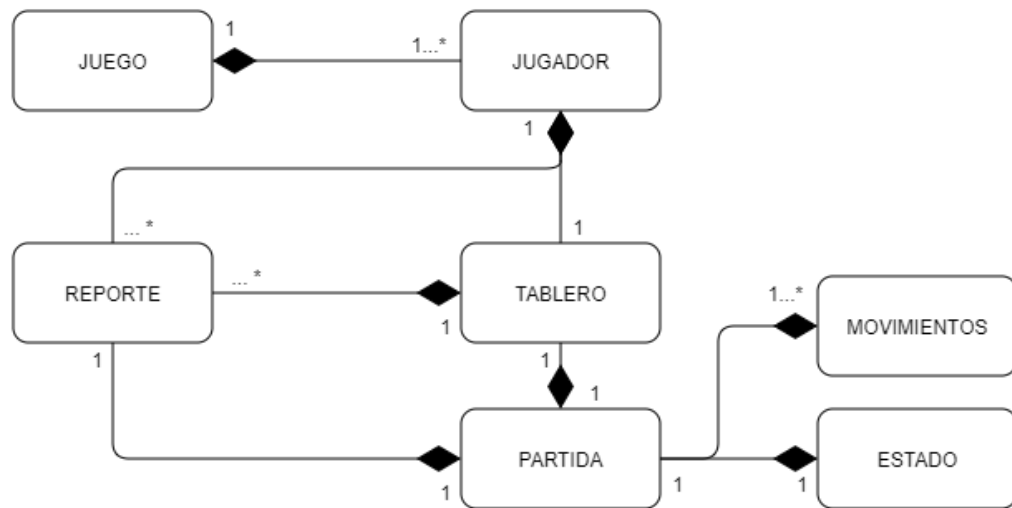
10. DIAGRAMA INICIAL DEL MODELO CONCEPTUAL



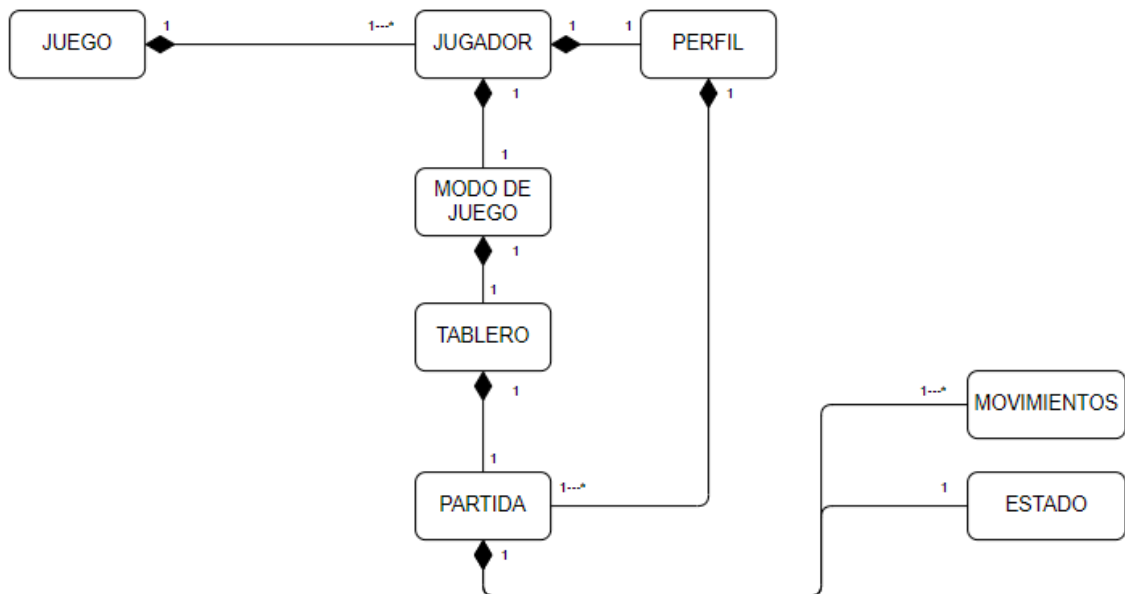
11. DIAGRAMA DE CLASES



12. MODELO CONCEPTUAL TERMINADO



13. MODELO CONCEPTUAL ACTUALIZADO



14. SCRIPT DE LA BASE DE DATOS

CREATE TABLE MOVIMIENTO

```
(
  idMovimiento INTEGER NOT NULL ,
  posicionInicial INTEGER NOT NULL ,
  posicionFinal INTEGER NOT NULL ,
  ficha VARCHAR (4) NOT NULL ,
  color BIT NOT NULL ,
  PARTIDA_idUsuario NUMERIC (10,1) NOT NULL ,
  PARTIDA_usuario VARCHAR (20) NOT NULL ,
  TORNEO_TORNEO_ID1 NUMERIC (28) NOT NULL
)
GO
```

ALTER TABLE MOVIMIENTO ADD CONSTRAINT MOVIMIENTO_PK PRIMARY KEY CLUSTERED (idMovimiento)

```
WITH (
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON )
GO
```

CREATE TABLE PARTIDA

```
(
  USUARIO_idUsuario NUMERIC (10,1) NOT NULL ,
  USUARIO_usuario VARCHAR (20) NOT NULL ,
  movimientos NUMERIC (2) NOT NULL ,
  punteo NUMERIC (2) NOT NULL ,
  tipo VARCHAR (10) NOT NULL ,
  reporte VARCHAR (1000) NOT NULL
)
GO
```

ALTER TABLE PARTIDA ADD CONSTRAINT PARTIDA_PK PRIMARY KEY CLUSTERED (USUARIO_idUsuario, USUARIO_usuario)

```
WITH (
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON )
GO
```

CREATE TABLE REPORTE

```
(
  PARTIDA_idUsuario NUMERIC (10,1) NOT NULL ,
  PARTIDA_usuario VARCHAR (20) NOT NULL ,
  movimientos NUMERIC (1) NOT NULL ,
  resultado BIT NOT NULL
)
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'true = victoria
false = derrota
', 'USER' , 'dbo' , 'TABLE' , 'REPORTE' , 'COLUMN' , 'resultado'
GO
```

```
CREATE UNIQUE NONCLUSTERED INDEX
    REPORTEv1__IDX ON REPORTE
(
    PARTIDA_idUsuario ,
    PARTIDA_usuario
)
GO
```

```
CREATE TABLE TORNEO
(
    USUARIO_idUsuario NUMERIC (10,1) NOT NULL ,
    USUARIO_usuario VARCHAR (20) NOT NULL ,
    idUsuarioInvitado VARCHAR (10) NOT NULL ,
    punteo NUMERIC (2) NOT NULL ,
    cantidadMovimientos NUMERIC (2) NOT NULL ,
    resultado BIT NOT NULL ,
    TORNEO_ID NUMERIC (28) NOT NULL IDENTITY NOT FOR REPLICATION
)
GO
```

```
EXEC sp_addextendedproperty 'MS_Description' , 'true = victoria
false = derrota' , 'USER' , 'dbo' , 'TABLE' , 'TORNEO' , 'COLUMN' , 'resultado'
GO
```

```
CREATE UNIQUE NONCLUSTERED INDEX
    TORNEO__IDX ON TORNEO
(
    USUARIO_idUsuario ,
    USUARIO_usuario
)
GO
```

```
ALTER TABLE TORNEO ADD CONSTRAINT TORNEO_PK PRIMARY KEY CLUSTERED (TORNEO_ID)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE USUARIO
(
    idUsuario NUMERIC (10,1) NOT NULL ,
    nombre VARCHAR (20) NOT NULL ,
    apellidos VARCHAR (20) NOT NULL ,
    usuario VARCHAR (20) NOT NULL ,
    contraseña VARCHAR (10) NOT NULL ,
    fechaNacimiento DATE NOT NULL ,
    correo VARCHAR (20) NOT NULL ,
    pais NUMERIC (2) NOT NULL ,
    estadold NUMERIC (1) NOT NULL ,
    fechaUltimaMod DATETIME NOT NULL
)
GO
```

```
ALTER TABLE USUARIO ADD CONSTRAINT USUARIO_PK PRIMARY KEY CLUSTERED (idUserio, usuario)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO
```

```
ALTER TABLE MOVIMIENTO
ADD CONSTRAINT MOVIMIENTO_PARTIDA_FK FOREIGN KEY
(
    PARTIDA_idUsuario,
    PARTIDA_usuario
)
REFERENCES PARTIDA
(
    USUARIO_idUsuario ,
    USUARIO_usuario
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```

```
ALTER TABLE MOVIMIENTO
ADD CONSTRAINT MOVIMIENTO_TORNEO_FKv2 FOREIGN KEY
(
    TORNEO_TORNEO_ID1
)
REFERENCES TORNEO
(
    TORNEO_ID
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```

```
ALTER TABLE PARTIDA
ADD CONSTRAINT PARTIDA_USUARIO_FK FOREIGN KEY
(
    USUARIO_idUsuario,
    USUARIO_usuario
)
REFERENCES USUARIO
(
    idUsuario ,
    usuario
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```



```
ALTER TABLE REPORTE
ADD CONSTRAINT REPORTEv1_PARTIDA_FK FOREIGN KEY
(
    PARTIDA_idUsuario,
    PARTIDA_usuario
)
REFERENCES PARTIDA
(
    USUARIO_idUsuario ,
    USUARIO_usuario
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```

```
ALTER TABLE TORNEO
ADD CONSTRAINT TORNEO_USUARIO_FK FOREIGN KEY
(
    USUARIO_idUsuario,
    USUARIO_usuario
)
REFERENCES USUARIO
(
    idUsuario ,
    usuario
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```