

Nombre: **Luis Antonio Cutzal Chali**

A

Carné: **201700841**

Entregable 2 Fase 3

Ingeniero Marlon Antonio PérezTürk

Auxiliar: Tanya Muhun

Fecha de Entrega: 07/11/2020

Sistema de la empresa IGamesGT

1. Toma de Requerimientos

I. Objetivo General

Elaborar una plataforma basada en el juego de mesa “Othello” en la cual diversos usuarios previamente registrados podrán disfrutar, creando partidas, realizando campeonatos o simplemente jugar individualmente.

II. Objetivos Específicos

- Hacer que más personas alrededor del mundo puedan conocer y jugar diversos tipos de juegos de mesa online como “Othello”.
- Atraer jugadores dispuestos a poner su mente a pensar en el juego, además de bríndales una experiencia agradable con la interfaz que se tendrá en la aplicación.
- Innovar y actualizar la experiencia del juego “Othello” a través del uso de una plataforma ya antes mencionada.

III. Alcance del Proyecto

La plataforma contara con diversos modos de juego, desde campeonatos internacionales, donde los usuarios previamente registrados competirán unos contra otros, el cual a su vez contara con un modo de JCJ (jugador contra jugador) esto para las personas que quieran unas partidas amistosas con amigos ya que este modo de juego contara con un apartado de enviar solicitudes de juego a otros usuarios.

Para los usuarios que quieran ponerse a prueba contra una computadora, este sistema contara con ese modo de juego, todos los usuarios registrados en la aplicación podrán acceder a este modo de juego el cual consistirá en: vencer a la computadora, ya que está basado para personas que les guste reatar a una maquina o simplemente quieran jugar solos.

Se agregarán nuevas funciones al juego Othello, con el fin de llamar la atención de todo público el cual a su vez popularizar el

sistema y los modos de juego. Dichas funcionalidades crearan un mayor entretenimiento y reto para los usuarios que prueben el sistema.

IV. Requerimientos Iniciales del Sistema

• Requerimientos funcionales

Ref#	Función	Categoría
R1.1	Registrar a los diversos usuarios (jugadores) que usaran el sistema.	Evidente
R1.2	Incrementa el registro de todos los usuarios.	Oculto
R1.3	Los usuarios ya registrados deben de introducir las credenciales: nombre de usuario y contraseña para poder usar el sistema.	Evidente
R1.4	Crear partidas individuales (jugador contra la máquina)	Evidente
R1.5	Crear partidas públicas (jugador contra jugador)	Evidente
R1.6	Guardar partidas en los modos de juego individual o contra otro jugador.	Evidente
R1.7	Incrementar el registro de partidas guardadas por un usuario.	Oculto
R1.8	Registrar las solicitudes de juego a otros usuarios.	Oculto
R1.9	Incrementar el registro de estadísticas de los usuarios.	Oculto
R1.10	Cargar juegos guardados por un usuario.	Evidente
R1.11	Mostrar estadísticas del usuario en cada modo de juego en el que este haya participado.	Evidente
R1.12	Crear torneos, dichos torneos tendrán un nombre único y con la capacidad de aceptar hasta 16 participantes.	Evidente
R1.13	Incrementar el registro de torneos creados.	Oculto
R1.14	Mostrar al ganador de cada ronda de los torneos	Evidente
R1.15	Mostrar al ganador del torneo.	Evidente
R1.16	Registrar al equipo ganador de cada ronda del torneo, así como al equipo ganador del mismo	Oculto
R1.17	Registrar los movimientos de cada jugador sin importar el modo de juego.	Oculto
R1.18	Mostrar diversos reportes en base a la cuenta del usuario.	Evidente
R1.19	El usuario podrá cerrar sesión de la aplicación cuando este lo desee, excepto en partidas de	Evidente

	torneos hasta que este haya sido declarado como perdedor en cualquier ronda del juego.	
R1.20	Unirse a partidas en modo versus, sin importar si el usuario esta registrado o no en el sistema.	Evidente
R1.21	Cualquier usuario podrá iniciar una partida en el modo Othello Extream	Evidente
R1.22	Introducir dimensiones para el Tablero	Evidente
R1.23	El sistema generara un Tablero con las dimensiones introducidas por el usuario.	Evidente
R1.24	El sistema validara las dimensiones para el Tablero	Oculto
R1.25	Los usuarios podrán elegir diversos colores	Evidente
R1.26	El sistema guardara la selección de colores por jugador y las usara para los turnos	Oculto
R1.27	Validar las reglas de colores para caga jugador	Oculto
R1.28	Validar reglas del Reto Inverso	Oculto
R1.29	Cualquier usuario podrá iniciar una partida en el modo Reto Inverso	Evidente
R1.30	Validar colores de fichas para el Reto Inverso	Oculto
R1.31	Validar Apertura Personalizada	Oculto
R1.32	Validar control de tiempo (cronometro) para cada modo de juego	Oculto
R1.33	En cada modo (excepto en el modo campeonato) se verá un cronometro.	Evidente
R1.34	Validar equipos para el campeonato	Oculto
R1.35	Registrar la puntuación de cada equipo por partida en los campeonatos	Oculto
R1.36	Cada ronda terminada del campeonato se podrá ver la cantidad de puntos que se tienen.	Evidente
R1.37	Validar empates por torneos	Oculto
R1.38	Validar esquema del torneo	Oculto
R1.39	Mostrar esquema del torneo a los jugadores	Evidente

• **Atributos del Sistema (No funcionales)**

Atributo	Detalles y restricciones de frontera
Tiempo de respuesta	<ol style="list-style-type: none"> 1. Cada vez que se registre un usuario, este ingresara a su cuenta en un tiempo aproximado de 1 a 3 segundos. 2. La opción de generar las estadísticas del jugador en un PDF aparecerá en un tiempo aproximado de 2 a 5 segundos. 3. La visualización de la gráfica de las estadísticas del jugador aparecerá en un tiempo aproximado de 1 a 2 segundos. 4. Las invitaciones que reciben los jugadores aparecerán de 1 a 2 segundos después de estas ser enviadas. 5. La invitación será enviada al momento de la confirmación de la invitación. 6. Los movimientos de las piezas del juego tardaran un tiempo aproximado de 1 a 3 segundos. 7. Cuando el juego haya terminado, el ganador será mostrado aproximadamente 1 a 3 segundos después. 8. Entrar a la cuenta demorara de 1 a 3 segundos.
Metáfora de interfaz	<ol style="list-style-type: none"> 1. Crear una sistema atractivo y muy fácil de usar para los usuarios. 2. Realizar diversas actividades (torneos) atractivas para los jugadores de hoy en día. 3. Innovar, actualizar la experiencia del juego Othello. 4. Crear diversos modos de juego con el fin de retar a la mente de los jugadores.

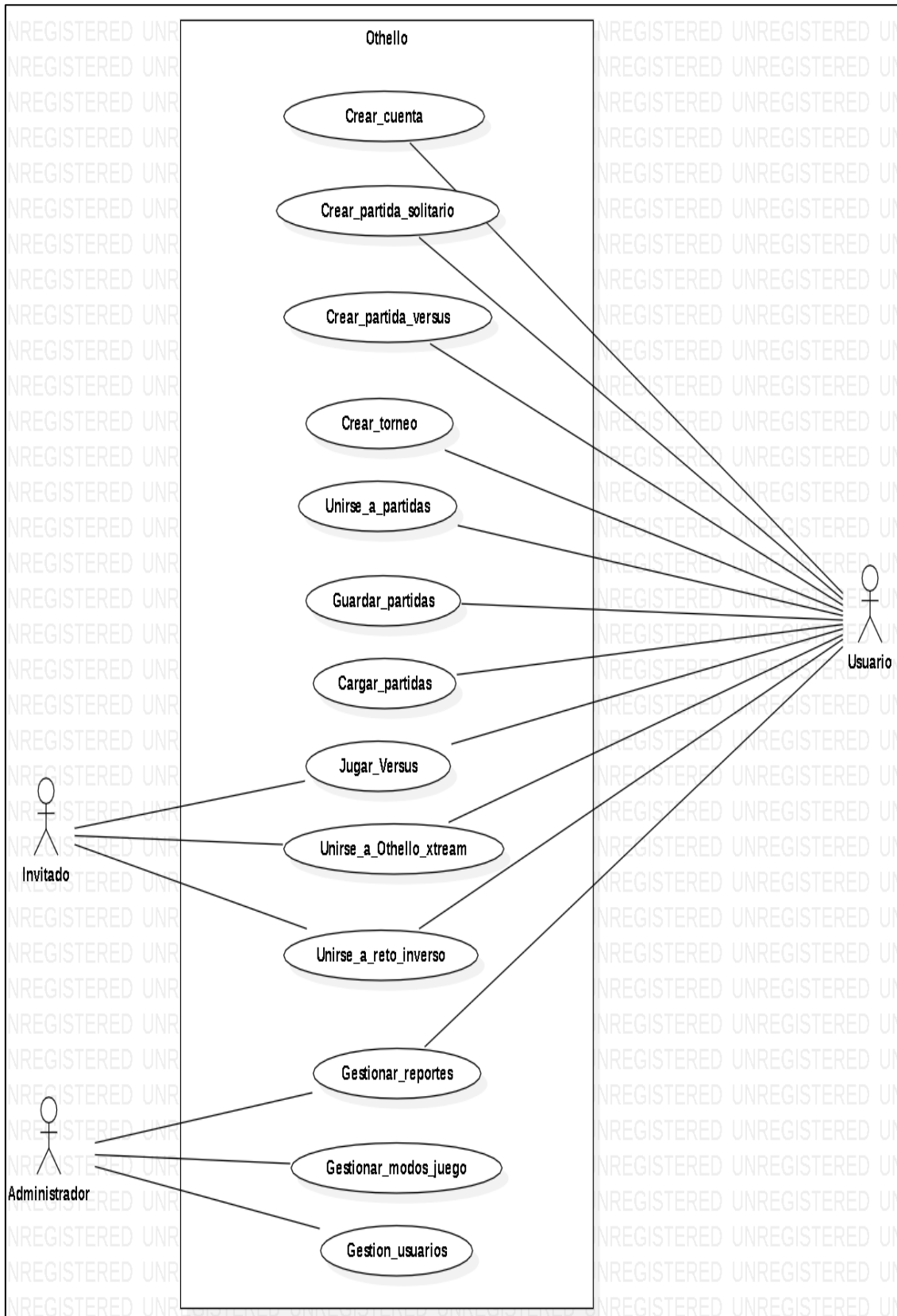
Tolerancia a fallas	<ol style="list-style-type: none"> 1. Los usuarios podrán registrarse no importando las fallas que produzca la energía eléctrica o el equipo que se esté utilizando. 2. Cualquier usuario registrado podrá ingresar a su cuenta no importando que tan alto o bajo sea el ancho de banda que este use. 3. El sistema no se vendrá abajo, sin importa cuantos usuarios estén utilizándolo al mismo tiempo. 4. Los mensajes que realicen los jugadores serán enviados no importando el ancho de banda con que este cuente. 5. Las partidas guardadas siempre podrán cargarse, no importando el tiempo que haya pasado luego de jugarla.
Plataformas del sistema operativo	Google Chrome, Mozilla Firefox, Opera, Safari.

V. Glosario Inicial

- **Registro:** Es el reconocimiento de una determinada acción que se considera relévate.
- **Plataforma:** Es un sistema que sirve para hacer funcionar determinados módulos de hardware o software con los cuales es compatible.
- **Reportes:** Informe acerca de un echo o suceso reciente
- **Credenciales:** Conjunto de datos que incluye la identificación y prueba de identificación que se utiliza para obtener acceso a recursos locales y de red.
- **Interfaz:** Es una conexión entre dos máquinas de cualquier tipo, a las cuales les brinda un soporte para la comunicación a diferentes estratos.

- **Sistema Operativo:** Conjunto de órdenes y programas que controlan los procesos básicos de una computadora y permiten el funcionamiento de otros programas.
- **Torneo:** Conjunto de pruebas en que una serie de contrincantes compiten por conseguir el triunfo
- **Jugador contra jugador (JCJ):** Designar una modalidad en la que dos o más jugadores se enfrentan entre ellos.
- **Ancho de banda:** Es la medida de datos y recursos de comunicación disponible
- **Usuario:** Que usa habitualmente un servicio.
- **Partidas de torneo:** Diseñadas para aquellos jugadores que deseen poner a prueba sus habilidades, ofrecen una experiencia más seria que los modos: JCJ y solitario (contra la máquina).

II. Diagrama de Casos de Uso



III. Casos de Uso de Alto Nivel

Caso de Uso: Crear_cuenta

Actores: Usuario.

Tipo: Primario.

Descripción: Cualquier persona podrá crear una cuenta en el sistema para así poderlo utilizar, siempre y cuando esta no tenga una ya creada.

Caso de Uso: Crear_partida_solitario

Actores: Usuario.

Tipo: Primario.

Descripción: Cualquier jugador dentro del sistema tendrá la posibilidad de crear una partida en solitario (jugar contra la computadora).

Caso de Uso: Gestionar_usuarios

Actores: Administrador.

Tipo: Primario.

Descripción: El sistema se encargará de la gestión de todos los usuarios registrados, manipulará el estado de la cuenta de los usuarios dentro de la aplicación.

Caso de Uso: Crear_partida_versus

Actores: Usuario

Tipo: Primario

Descripción: Todos los usuarios registrados en el sistema podrán entrar o crear una partida: "jugador versus jugador".

Caso de Uso: Gestionar_reportes

Actores: Administrador, Usuario.

Tipo: Principal.

Descripción: El sistema tendrá la capacidad de crear reportes en base a los distintos usuarios que están registrados en la aplicación.

Caso de Uso: Crear_torneo

Actores: Usuario.

Tipo: Principal.

Descripción: Cualquier usuario podrá crear un torneo de hasta 16 participantes, en el cual solo podrán participar usuarios registrados en el sistema.

Caso de Uso: Cargar_partidas

Actores: Usuario.

Tipo: Secundario

Descripción: Se cargarán diversas partidas ya sea en solitario o contra un jugador y que el usuario haya guardado previamente.

Caso de Uso: Guardar_partidas

Actores: Usuario.

Tipo: Secundario.

Descripción: Se podrán guardar las partidas en los modos solitario o contra un jugador.

Caso de Uso: Gestionar_modos_juego

Actores: Administrador.

Tipo: Primario

Descripción: El sistema contara con 3 modos de juego, el cual el usuario podrá ingresar y jugar, ya sea en solitario, contra otro jugador o estar en un torneo.

Caso de Uso: Unirse_a_partidas

Actores: Usuario.

Tipo: Primario

Descripción: Cualquier usuario podrá unirse a las partidas ya sea contra otro jugador o en un torneo.

Caso de Uso: Jugar_Versus.

Actores: Invitado, Usuario.

Tipo: Primario

Descripción: Cualquier persona, ya sea registrado o no en el sistema, podrá jugar contra otro usuario, siempre y cuando uno de ellos tenga una cuenta y estén en la misma sesión.

Caso de Uso: Unirse_a_Othello_xtream.

Actores: Invitado, Usuario.

Tipo: Primario

Descripción: Cualquier persona, ya sea registrado o no en el sistema, podrá jugar contra otro usuario en este modo de juego nuevo, siempre y cuando uno de ellos tenga una cuenta y estén en la misma sesión.

Caso de Uso: Unirse_a_reto_inverso.

Actores: Invitado, Usuario.

Tipo: Primario

Descripción: Cualquier persona, ya sea registrado o no en el sistema, podrá jugar contra otro usuario en este nuevo modo de juego, siempre y cuando uno de ellos tenga una cuenta y estén en la misma sesión.

IV. Casos de Uso Expandidos

Caso de Uso: Unirse_a_reto_inverso.

Actores: Invitado, Usuario.

Propósito: Hacer que cualquier persona (usuario registrado o no), pueda jugar y divertirse en un modo de juego innovador y más competitivo.

Resumen: Este modo de juego es totalmente diferente al Othello tradicional, en este modo ganara el usuario que tenga una cantidad de fichas menor al oponente.

Tipo: Primario y esencial.

Referencias cruzadas: *Funciones:* R1.28, R1.29, R1.30, R1.31, R1.32, R1.33

Curso normal de eventos:

Acción de los actores	Respuestas del sistema
1. Una de las dos personas debe de ingresar a su cuenta.	2. El sistema validara las credenciales que el usuario escriba.
a) Si las dos personas no poseen una cuenta véase el caso <i>Crear_cuenta</i> .	
	3. Lo redireccionara al inicio de su cuenta.
4. Dar clic al botón de Versus.	5. Mostrará la ventana para cargar una partida.
	6. Mostrará un botón para ir al tablero y poder jugar.
	7. Cargara la pantalla de la partida.
8. Cuando esté listo comienza a jugar contra la otra persona que este junto al usuario.	9. Cualquier usuario podrá elegir le turno que le corresponde, eligiendo el color de la ficha.
a) Si se tiene una partida guardada véase el caso de uso <i>Cargar_partida</i> .	

b) Si se quiere guardar la partida véase el caso de uso *Guardar_partida*.

- | | |
|---|--|
| 10. La partida terminara hasta que ninguno pueda jugar una ficha o el tablero este sin ningún espacio para tirar. | 11. Se guardarán los datos de la partida, pero solo para la persona que está registrada. |
| 14. Cuando se termine la partida aparecerá en pantalla al ganador de la misma. | 12. En las estadísticas del jugador, se incrementará en uno, si gano, perdió, o empate la partida.
13. Se llevará un contador de fichas colocadas por jugador y se declarará ganador al usuario que tenga menos fichas que el otro colocadas en el tablero. |

Cursos alternos:

- Línea 1: La persona no cuenta con ningún amigo para jugar en el sistema, podrá crear una cuenta y jugar contra otros usuarios registrados en la ampliación, sin este tener un amigo al lado y jugar en la misma sesión.
- Línea 8: Cada jugador tendrá un tiempo límite por turno, si el tiempo se termina pasará al turno del otro jugador automáticamente.
- Línea 10: Si en la partida hay un empate se incrementara en 1 en el apartado de estadísticas del usuario registrado dentro del sistema y aparecerá una opción para hacer el desempate de la partida anterior.

Caso de Uso: Unirse_a_Othello_xtream.

Actores: Invitado, Usuario.

Propósito: Hacer que cualquier persona (usuario registrado o no), pueda jugar un modo innovador y extremo del juego clásico.

Resumen: En este modo de juego, el usuario podrá crear un tablero MxN donde cada uno es un número entero par, además de que podrá elegir un máximo de 5 colores de

fichas para poder jugar, este modo de juego es para poder a prueba las diversas destrezas de los usuarios y poder entretenerlos innovando el juego clásico.

Tipo: Primario y esencial.

Referencias cruzadas: *Funciones:* R1.20

Curso normal de eventos:

Acción de los actores

1. Una de las dos personas debe de ingresar a su cuenta.

b) Si las dos personas no poseen una cuenta véase el caso *Crear_cuenta*.

4. Dar clic al botón de Versus.

8. El usuario podrá darle dimensiones MxN al tablero, donde M y N son números enteros pares positivos.

10. Los 2 jugadores podrán elegir hasta un máximo de 5 colores para cada ficha y un mínimo de 1 color por ficha.

12. Cuando esté listo comienza a jugar contra la otra persona que este junto al usuario.

a) Si se tiene una partida guardada véase el caso de uso *Cargar_partida*.

b) Si se quiere guardar la partida véase el caso de uso *Guardar_partida*.

14. La partida terminara hasta que ninguno pueda jugar una ficha o el tablero

Respuestas del sistema

2. El sistema validara las credenciales que el usuario escriba.

3. Lo redireccionara al inicio de su cuenta.

5. Mostrará la ventana para cargar una partida.

6. Mostrará un botón para ir al tablero y poder jugar.

7. Cargara la pantalla de la partida.

9. El sistema validara las dimensiones del tablero escritas por el usuario, creara y mostrara el tablero.

11. Se guardará la selección de colores elegidas por los 2 usuarios para el juego.

13. El turno de los 2 usuarios se elegirá manualmente.

15. Se guardarán los datos de la partida, pero

este sin ningún espacio para tirar. solo para la persona que está registrada.

16. En las estadísticas del jugador, se incrementará en uno, si gano, perdió, o empate la partida.

Cursos alternos:

- Línea 1: La persona no cuenta con ningún amigo para jugar en el sistema, podrá crear una cuenta y jugar contra otros usuarios registrados en la ampliación, sin este tener un amigo al lado y jugar en la misma sesión.
- Línea 10: Si en la partida hay un empate se incrementara en 1 en el apartado de estadísticas del usuario registrado dentro del sistema y aparecerá una opción para hacer el desempate de la partida anterior.

Caso de Uso: Jugar_Versus.

Actores: Invitado, Usuario.

Propósito: Hacer que cualquier persona (usuario registrado o no), pueda jugar y divertirse en el modo versus del sistema.

Resumen: Cada vez que un usuario quiera jugar con un amigo que no tenga cuenta creada, podrá hacerlo, entrando al modo versus y jugando en la misma sesión del usuario que si tiene una cuenta.

Tipo: Primario y esencial.

Referencias cruzadas: *Funciones:* R1.20, R1.32, R1.33

Curso normal de eventos:

Acción de los actores

1. Una de las dos personas debe de ingresar a su cuenta.

c) Si las dos personas no poseen una cuenta véase el caso *Crear_cuenta*.

4. Dar clic al botón de Versus.

Respuestas del sistema

2. El sistema validara las credenciales que el usuario escriba.

3. Lo redireccionara al inicio de su cuenta.

5. Mostrará la ventana para cargar una partida.

6. Mostrará un botón para ir al tablero y poder jugar.
7. Cargará la pantalla de la partida.
8. Cuando esté listo comienza a jugar contra la otra persona que este junto al usuario.
 - a) Si se tiene una partida guardada véase el caso de uso *Cargar_partida*.
 - b) Si se quiere guardar la partida véase el caso de uso *Guardar_partida*.
9. Cualquier usuario podrá elegir le turno que le corresponde, eligiendo el color de la ficha.
10. La partida terminará hasta que ninguno pueda jugar una ficha o el tablero este sin ningún espacio para tirar.
11. Se guardarán los datos de la partida, pero solo para la persona que está registrada.
12. En las estadísticas del jugador, se incrementará en uno, si gano, perdió, o empate la partida.

Cursos alternos:

- Línea 1: La persona no cuenta con ningún amigo para jugar en el sistema, podrá crear una cuenta y jugar contra otros usuarios registrados en la ampliación, sin este tener un amigo al lado y jugar en la misma sesión.

Caso de Uso: Unirse_a_partidas

Actores: Usuario

Propósito: Hacer que los usuarios puedan unirse a partidas a las cuales fueron invitados.

Resumen: Cada vez que un usuario quiera unirse a una partida deberá de aceptar previamente una invitación hecha por otro usuario (creador de la partida a la cual quiere unirse).

Tipo: Primario y esencial.

Referencias cruzadas: *Funciones:* R1.8, R1.5, R1.3

Curso normal de eventos:

Acción de los actores

Respuestas del sistema

1. El usuario debe de ingresar a su cuenta.
- d) Si el usuario no tiene cuenta véase el caso *Crear_cuenta*
2. El sistema validara las credenciales que el usuario escriba.
3. Lo redireccionara al inicio de su cuenta.
4. Dar clic al apartado de notificaciones.
5. Mostrará una lista con las invitaciones que este tiene para jugar.
6. Mostrará la opción unirse al juego.
7. Cargara la pantalla de la partida.
8. Cuando esté listo comienza a jugar contra otro usuario.
9. El sistema elegirá el turno de los dos usuarios.
- a) Si se tiene una partida guardada véase el caso de uso *Cargar_partida*.
- b) Si se quiere guardar la partida véase el caso de uso *Guardar_partida*.

Cursos alternos:

- Línea 3: El usuario no tiene ninguna invitación, podrá esperar a que le llegue alguna o crear el mismo una partida enviando una invitación a un jugador dentro del sistema.

Caso de Uso: Gestionar_modos_juego

Actores: Administrador.

Propósito: Velar por los modos de juego de los usuarios, con el fin de bríndales un sistema competente y en óptimo funcionamiento.

Resumen: El administrador podrá ver los diversos modos de juego, y poder manipular la información con que se cuenta.

Tipo: Primario y esencial.

Referencias cruzadas: *Funciones:* R1.3, R1.4, R1.5, R1.12.

Curso normal de eventos:

Acción de los actores

Respuestas del sistema

1. El administrador desea consultar los modos de juegos que se tiene.
2. Ingresa a la gestión de modos
3. El sistema le muestra los modos de juego.
4. Muestra las opciones con cada modo de juego.

Cursos alternos:

- Línea 2: ningún usuario ha creado un torneo, el sistema mostrara un mensaje al respecto.

Caso de Uso: Guardar_partidas

Actores: Usuario.

Propósito: Se guardarán todas las partidas que el usuario desee hacerlo, con el fin de brindar un mejor servicio dentro del modo de juego.

Resumen: En los modos de juego solitario y versus, se podrán guardar las partidas que el usuario considere, para así poder volver a jugarlas cuando el desee.

Tipo: Secundario y opcional.

Referencias cruzadas: *Funciones* R1.3, R1.5, R1.6, R1.7

Casos de uso: El usuario debe de haber terminado el caso de uso: *Crear_cuenta*.

Casos de uso: El usuario debe de haber elegido el caso de uso: *Crear_partida_solitario*.

Casos de uso: El usuario debe de haber elegido el caso de uso: *Crear_partida_versus*.

Curso normal de eventos:

Acción de los actores	Respuestas del sistema
1. El usuario debe de haber elegido un modo de juego.	
a) Si entra al modo de juego "solitario" véase el caso de uso <i>Crear_partida_solitario</i>	
b) Si entra al modo de juego "versus" véase el caso de uso <i>Crear_partida_versus</i>	

- | | |
|--|--|
| 2. Se desea guardar la partida en los modos de juegos mencionados. | 3. El sistema le muestra la opción de guardar. |
| 4. El usuario presiona el botón guardar. | 5. El sistema guarda la partida. |

Cursos alternos:

- Línea 1: El usuario debió entrar al modo de juego “solitario” o “versus” para guardar una partida.
- Línea 2: El usuario no desea guardar nada, el sistema no guardara partidas automáticamente.

Caso de Uso: Cargar_partidas

Actores: Usuario.

Propósito: Se tendrá la posibilidad de cargar juegos guardados en el sistema.

Resumen: Cada partida que haya sido guardada en el sistema podrá volver a cargarse, con todos los movimientos realizados por el usuario.

Tipo: Secundario y opcional.

Referencias cruzadas: *Funciones:* R1.3, R1.6, R1.7, R1.10

Casos de uso: El usuario debe de haber terminado el caso de uso: *Crear_cuenta*.

Casos de uso: El usuario debe de haber terminado el caso de uso: *Guardar_partidas*.

Curso normal de eventos:

Acción de los actores	Respuestas del sistema
1. El usuario desea retomar el juego guardado.	
2. Entra al modo de juego que quiere cargar	
a) Si entra al modo de juego “solitario” véase el caso de uso <i>Crear_partida_solitario</i>	
b) Si entra al modo de juego “versus” véase el caso de uso <i>Crear_partida_versus</i>	
3. El usuario presiona la opción Cargar	4. Se verifica él usuario.

5. Se verifica el modo de juego.
6. El sistema busca y muestra un listado de partidas guardadas por el usuario en dicho modo de juego.
7. Elije la partida que quiere retomar.
8. Carga y muestra la partida seleccionada.

Cursos alternos:

- Línea 3: El usuario debió guardar una partida previamente antes de usar esta opción, dicha partida no importa el modo de juego excepto el modo Torneo.

Caso de Uso: Crear_torneo

Actores: Usuario.

Propósito: Crear diversos eventos (torneos) para atraer a más jugadores a la plataforma.

Resumen: Cada usuario podrá crear los torneos que desee, cada uno con nombres diferentes y podrán ser de hasta 16 participantes.

Tipo: Principal y esencial.

Referencias cruzadas: *Funciones:* R1.3, R1.12, R1.13, R1.14, R1.15, R1.16, R1.17, R1.34, R1.35, R1.36, R1.37, R1.38, R1.39.

Curso normal de eventos:

Acción de los actores

1. Se llena un formulario para crear un campeonato
4. Se registran a los equipos para el campeonato
7. Se inscriben a los participantes de cada equipo.
9. En cada ronda, los 3 jugadores de cada equipo participaran.

Respuestas del sistema

2. El sistema compara el nombre del torneo con los otros torneos creados.
3. Se crea el torneo.
5. Se guarda el registro de los usuarios inscritos para el campeonato.
6. Se validará que el número de equipos sea 4, 8 o 16 equipos.
8. Se validará que cada equipo cuente con 3 integrantes.
10. En cada ronda ganada se obtendrán 3 puntos y 1

- punto por partida empatada.
11. Se registrarán los puntos por cada partida.
12. Si los puntos de un equipo son mayores al equipo contrincante pasan a la siguiente ronda.
13. El equipo ganó las rondas del toreo.
14. Pasa a las siguientes etapas hasta llegar a la final.
15. Se registran las partidas de todos los jugadores (ganadores o perdedores).
16. Se declara al equipo ganador del torneo.
17. Se registra al ganador de todo el torneo.

Cursos alternos:

- Línea 9: Empate por parte de los equipos en cualquier ronda del torneo, se tendrá que realizar otra partida para desempatar, hasta concluir con él ganador y pasar a la siguiente ronda o ganar el torneo.
- Línea 2: El torneo no se puede crear porque el nombre ya ha sido usado, se mostrará un mensaje con esta información y se pedirá que escriba otro nombre.
- Línea 3: Si no se llega a un número de 4, 8 o 16 equipos no se creará el torneo.
- Línea 5: Si un equipo no cuenta con 3 participantes como máximo no podrá inscribirse al campeonato en cuestión.

Caso de Uso: Gestionar_reportes

Actores: Administrador, Usuario.

Propósito: Usuarios registrados puedan ver sus estadísticas de juego en los diversos modos con que cuenta el sistema.

Resumen: Cada usuario podrá crear los diversos reportes con que cuenta el sistema, con el fin de ver sus propias estadísticas de juego y poder mejorar si se desea hacerlo.

Tipo: Primario y esencial

Referencias cruzadas: *Funciones:* R1.2, R1.3, R1.9, R1.11, R1.16, R1.17, R1.18.

Curso normal de eventos:

Acción de los actores	Respuestas del sistema
1. Se desea ver las estadísticas de juego propias.	
2. Ingresar a la opción reportes.	3. El sistema capturara los datos de la cuenta de usuario. 4. Se mostrará las diversas opciones de reporte que tiene el sistema.
5. Se elige el tipo de reporte que se desea consultar.	6. El sistema generará el reporte y lo abrirá automáticamente.

Cursos alternos:

- Línea 4: El sistema generara reportes en blanco, porque el usuario no ha jugado ningún modo de juego dentro del sistema.

Caso de Uso: Crear_partida_versus

Actores: Usuario.

Propósito: Usuarios puedan tener un modo de juego amistoso contra otros usuarios que estén interesados en demostrar sus habilidades.

Resumen: Cualquier usuario podrá enviar una invitación a otro usuario y poder jugar contra él, con el fin de demostrar las habilidades de cada uno.

Tipo: Primario y esencial

Referencias cruzadas: *Funciones:* R1.3, R1.5, R1.6, R1.8, R1.10, R1.11, R1.17, R1.32, R1.33

Curso normal de eventos:

Acción de los actores	Respuestas del sistema
1. El usuario presiona el apartado de crear partida pública.	2. El sistema mostrara un listado de usuarios registrados en el sistema.
3. Se enviará una solicitud de juego a otro jugador.	4. Se registrará la solicitud de juego y se enviará al buzón del jugador seleccionado. 5. Se mostrará la solicitud al otro jugador.

6. El jugador quien recibe la invitación confirmara la invitación de juego.
7. Comenzará el juego cuando los dos usuarios estén listos.
8. El sistema elegirá el orden de los jugadores.
9. Se registrarán los movimientos de los jugadores.
10. Uno de los usuarios será el ganador
11. Se incrementa el registro de victorias o derrotas del jugador en cuestión.

Cursos alternos:

- Línea 3: El jugador tiene varias solicitudes de juego pendientes y no podrá aceptar a todos al mismo tiempo.
- Línea 4: El jugador invitado no acepta la solicitud de juego, se tendrá que enviar otra solicitud de juego a otro jugador para poder usar este modo de juego.
- Línea 7: El juego queda en empate, se mostrará un mensaje por parte del sistema, el cual será para el juego de desempate, se incrementará el registro de juegos empatados por parte de los dos jugadores.

Caso de Uso: Gestionar_usuarios

Actores: Administrador

Propósito: Manipular la información de cada usuario registrado.

Resumen: El administrador será el único en poder manipular los datos de los distintos usuarios que están registrados en la aplicación.

Tipo: Primario y esencial.

Referencias cruzadas: *Funciones:* R1.2, R1.3

Casos de uso: El usuario debe de haber terminado el caso de uso: *Crear_cuenta.*

Curso normal de eventos:

Acción de los actores	Respuestas del sistema
1. Comienza cuando el administrador quiere consultar los datos de los usuarios.	

- | | |
|--|---|
| <p>2. El administrador ingresa al sistema.</p> <p>4. Elige una de las diversas opciones que le aparecen en la página de inicio.</p> <p>5. Selecciona al usuario que desea consultar.</p> <p>7. El administrador desea manipular los datos del usuario.</p> <p style="padding-left: 40px;">a) El administrador desea eliminar al usuario, véase la sección <i>Eliminar_usuario</i></p> <p style="padding-left: 40px;">b) El administrador desea modificar los datos del usuario, véase la sección: <i>Modificar_usuario</i></p> | <p>3. El sistema lo direcciona a las opciones del administrador</p> <p>6. El sistema mostrara los datos del usuario que se elige, con el fin de poder manipular los mismos.</p> <p>8. El sistema guardara las modificaciones en la base de datos hechas por el administrador.</p> |
|--|---|

Cursos alternos:

- Línea 2: El administrador ingresa mal su contraseña o el nombre de usuario, el sistema le mostrará un mensaje de error y le volverá a pedir sus datos.
- Línea 4: No existe ningún usuario registrado en el sistema, no mostrara más que una página en blanco el botón regresar.
- Línea 5: No se modifica ningún dato de los usuarios, el sistema no guardara ninguna modificación.

Sección: Eliminar_usuario

Acción de los actores	Respuestas del sistema
1. Se selecciona al usuario que se desea eliminar.	2. El sistema mostrara la opción eliminar en la parte inferior del usuario.
3. Se presiona la opción eliminar.	4. El sistema mostrara un mensaje de alerta y pedirá una confirmación, dicha

confirmación mostrara un botón para aceptar o cancelar la acción realizada por el administrador.

4. Se confirma la eliminación.

5. El sistema borrará todos los datos del usuario seleccionado.

Cursos alternos:

- Línea 3: El administrador eligió a un usuario diferente, podrá darle al botón cancelar eliminación para que el sistema no elimine al usuario equivocado en la selección.

Sección: Modificar_usuario

Acción de los actores

1. Se selecciona al usuario que se desea modificar.

3. Se presiona la opción modificar.

5. El administrador modificara los datos del usuario, excepto el nombre de usuario.

Respuestas del sistema

2. El sistema mostrara la opción modificar en la parte inferior del usuario.

4. El sistema le mostrará un formulario el cual contendrá los datos del usuario.

6. Se guardarán las modificaciones realizadas.

7. Se actualizará la base de datos con los datos actualizados del usuario en cuestión.

Cursos alternos:

- Línea 3: El administrador eligió a un usuario diferente, podrá regresar a la página anterior para que el sistema modifique al usuario equivocado en la selección.
- Línea 4: No se modificaron datos del usuario, el sistema no registrara ninguna actualización de los datos.

Caso de Uso: Crear_partida_solitario

Actores: Usuario.

Propósito: Usuarios puedan tener un modo de juego contra la computadora, para poder retarse a sí mismos.

Resumen: Cualquier usuario que desee jugar contra la computadora, podrá hacerlo dando clic en el botón “Solitario” ubicado en la ventana “modos de juego”.

Tipo: Primario y esencial.

Referencias cruzadas: *Funciones:* R1.4, R1.7, R1.9, R1.10, R1.11, R1.17, R1.18, R1.19, R1.32, R1.33.

Curso normal de eventos:

Acción de los actores	Respuestas del sistema
1. Comienza cuando un usuario ingresa a la página del “login” de la aplicación.	
2. El usuario comienza a colocar sus datos correspondientes.	3. El sistema valida las credenciales que el usuario escribió en la parte del login.
a) Si no tiene cuenta, véase el Caso de Uso <i>Crear_cuenta</i>	
	4. El sistema lo redirecciona a la página principal de su cuenta.
5. Da clic a la opción “solitario” el cual estará ubicado en la ventana “modos de juego”	5. El sistema lo redirige a la opción solicitada.
	6. Comienza el juego, siendo aleatorio el quien inicie la partida (usuario o máquina).

Cursos alternos

- Línea 2: El usuario coloco mal su nombre de usuario o contraseña, el sistema le dará un error y le volverá a pedir sus credenciales.
- Línea 3: El internet del usuario es inestable, el sistema le mostrara la página principal de su cuenta, aunque se tardara un poco más de lo habitual.

Caso de Uso: Crear_cuenta

Actores: Usuario.

Propósito: Registrar los datos del usuario.

Resumen: Un usuario desea utilizar el sistema, tendrá que registrar los datos que le pedirá el formulario llenar, una vez la cuenta esta creada podrá acceder a todas las funciones del sistema.

Tipo: Primario y esencial.

Referencias cruzadas: *Funciones:* R1.1, R1.2, R1.19.

Curso normal de eventos:

Acción de los actores

Respuestas del sistema

- | | |
|---|---|
| 1. Este caso de uso comienza cuando usuario ingresa al sistema, en el apartado de Registro. | 2. El sistema lo direcciona al apartado que el desea, presionando el botón con la opción que el desea ir. |
| 3. Comienza a llenar los campos del formulario con la información que le pide. | |
| 4. Al terminar de llenar el formulario el usuario presiona el botón aceptar para poder finalizar la creación de una cuenta. | 5. El sistema almacena los datos del usuario en la base de datos. |
| | 6. Se muestra la pantalla inicial de la cuenta usuario |
| 7. El usuario ya puede ingresar al sistema y poder jugar los diversos modos con que cuenta dicho sistema. | |

Cursos alternos

- Línea 2: el usuario introduce un nombre de usuario ya existente, indica el error: Nombre de usuario ya existente.
- Línea 3: el usuario no registra un dato del formulario, indica el error: Faltan campos por llenar.
- Línea 3: el usuario no presiona el botón regresar y no el de aceptar, no se guardarán los datos de dicho usuario en el sistema.

```

classDiagram
    class Admin {
        +id_admin: int
        -Nombre_admin: string
        -Nombre: string
        -Apellido: string
        -Contraseña: string
        -FechaNac: date
        -Pas: string
        -Correo: string
        -Estado: boolean
        +Modificar_estado_usuario()
        +Login_usuario()
        +Cerrar_sesion()
        +Crear_torneo()
    }

    class Equipo {
        +id_equipo: int
        -Nombre_equipo: string
        -Nombre_participante1: string
        -Nombre_participante2: string
        -Nombre_participante3: string
        +CrearEquipo()
        +Participar_Torneo()
    }

    class Usuario {
        +id_usuario: int
        -Nombre_usuario: string
        -Apellido: string
        -Contraseña: string
        -FechaNac: date
        -Pas: string
        -Correo: string
        -Estado: boolean
        +Iniciar_sesion()
        +Cerrar_sesion()
        +Crear_partida_Solitario()
        +Index_Versus()
        +Partida_Altersolitario()
        +Registrar_equipo()
        +Consultar_partida()
        +Cargar_partida()
    }

    class Invitación {
        +id_invitación: int
        -Contenido: string
        +Idea_usuario()
        +Aceptar()
    }

    class Versus {
        +id_versus: int
        -ModoJuegoVersus()
        +IndexVersus(Archivo life)
        +Crear(Archivo)
    }

    class ModuloJuegoVersus {
        +Tablero_Solitario()
        +Tablero_Livresol()
        +Tablero_Xivresol()
    }

    class Solitario {
        +id_solitario: int
        +ModoJuegoSolitario()
        +Index(Archivo life)
        +Crear(Archivo)
    }

    class ModuloJuegoSolitario {
        +Tablero_Solitario()
        +Tablero_Livresol()
        +Tablero_Xivresol()
    }

    class Tablero {
        +id_tablero: int
        -fila: int
        -columna: string
        -fecha: string
        +Movimiento(string fila, string columna)
        +CambiarTurno(string turnoJugador)
        +PrimerCondicion()
        +SegundaCondicion()
        +CambiarColor()
    }

    class Archivo {
        +id_archivo: int
        +Obtener_modo_solitario()
        +Obtener_modo_versus()
        +Obtener_usuario()
    }

    class EstadoPartido {
        +Posicion, fecha()
        +Turno_usuario()
    }

    class JuegoPartido {
        +Obtener_usuario()
        +Obtener_partida_solitario()
        +Obtener_partida_versus()
        +Obtener_partidas_partidas()
    }

    class JuegoGanado {
        +Obtener_usuario()
        +Obtener_partida_solitario()
        +Obtener_partida_versus()
        +Obtener_partidas_ganadas()
        +Obtener_movimientos_partidas()
    }

    class JuegoEmpate {
        +Obtener_usuario()
        +Obtener_partida_solitario()
        +Obtener_partida_versus()
        +Obtener_partidas()
        +Obtener_movimientos_partidas()
    }

    class Reporte {
        +id_reporte: int
    }

    class JuegoTorneo {
        +Obtener_usuario()
        +Obtener_torneo_jugados()
        +Obtener_torneo_ganados()
        +Obtener_torneo_eliminados()
    }

    class Torneo {
        +id_torneo: int
        -NombreTorneo: string
    }

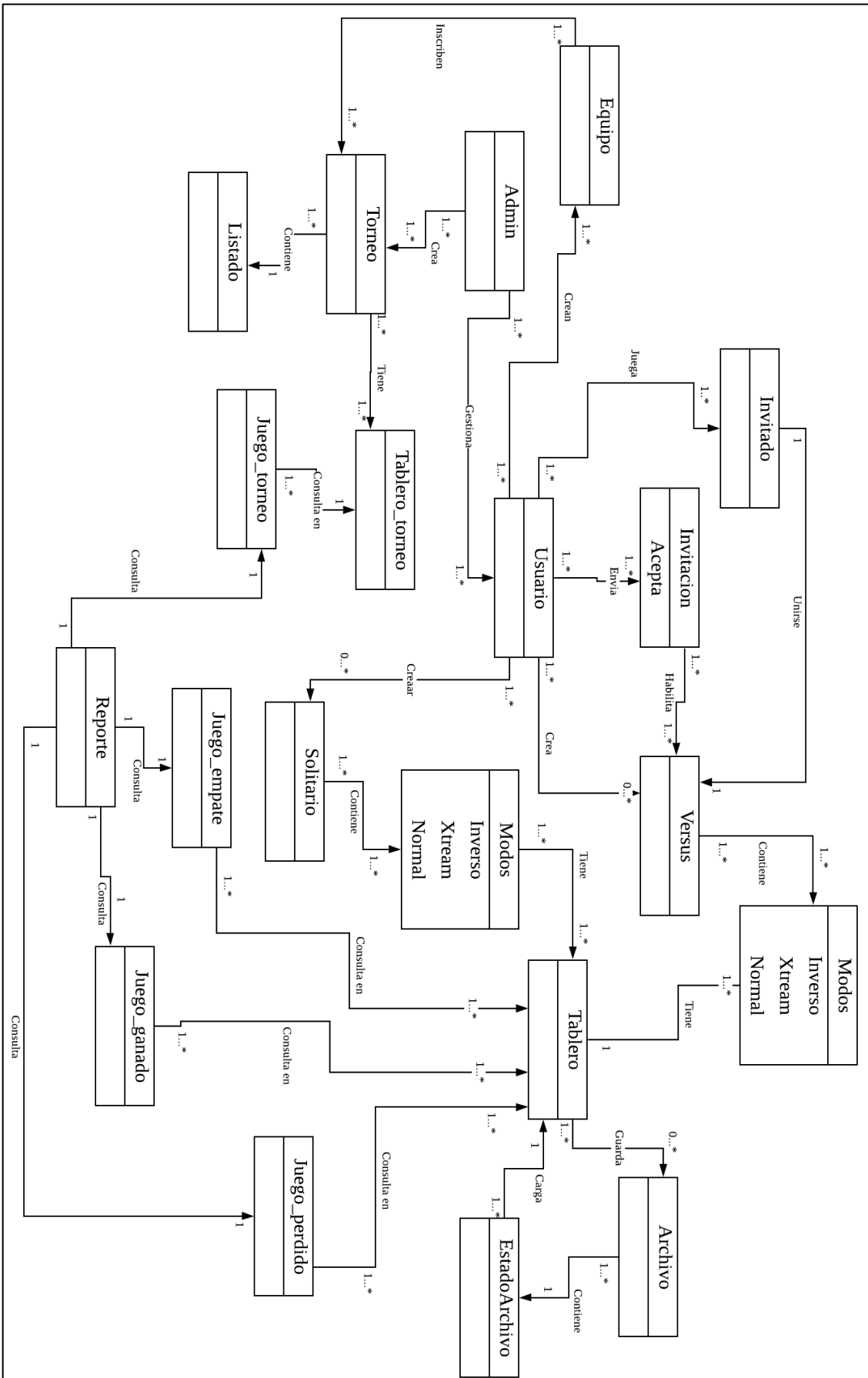
    class Lado {
        +id_lado: int
    }

    class TorneoParticipantes {
        +Registro_usuario_participantes()
    }

    Admin "1..*" -- "1..*" Equipo : Gestiona
    Admin "1..*" -- "1..*" Usuario : Inscriben
    Equipo "1..*" -- "0..*" Usuario : Crea
    Usuario "1..*" -- "0..*" Invitación : Envía
    Invitación "1..*" -- "1..*" Versus : Habilitan
    Versus "1..*" -- "1..*" ModuloJuegoVersus : Crea
    ModuloJuegoVersus "1..*" -- "1..*" Solitario : Tiene
    ModuloJuegoVersus "1..*" -- "1..*" ModuloJuegoSolitario : Tiene
    Solitario "1..*" -- "0..*" Usuario : Crea
    ModuloJuegoSolitario "1..*" -- "0..*" Usuario : Crea
    ModuloJuegoSolitario "1..*" -- "1..*" Tablero : Tiene
    Tablero "1..*" -- "1..*" Archivo : Guarda
    Archivo "1..*" -- "1..*" EstadoPartido : Actualiza
    EstadoPartido "1..*" -- "1..*" JuegoPartido : Consulta en
    JuegoPartido "1..*" -- "1..*" JuegoGanado : Consulta en
    JuegoGanado "1..*" -- "1..*" JuegoEmpate : Consulta en
    Reporte "1..*" -- "1..*" JuegoTorneo : Consulta en
    JuegoTorneo "1..*" -- "1..*" Torneo : Consulta en
    Torneo "1..*" -- "1..*" Lado : Tiene
    Torneo "1..*" -- "1..*" TorneoParticipantes : Tiene

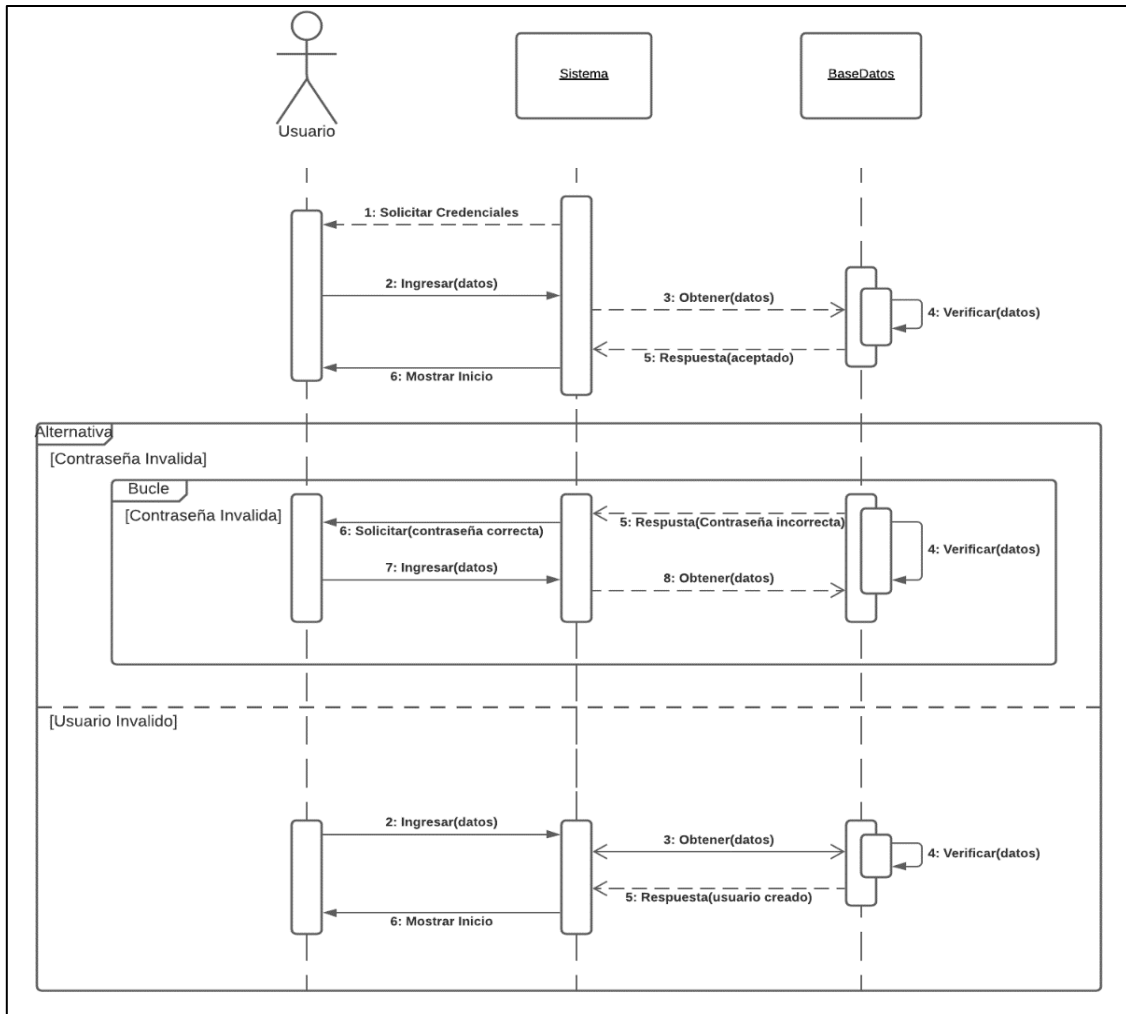
```

VI. Modelo Conceptual

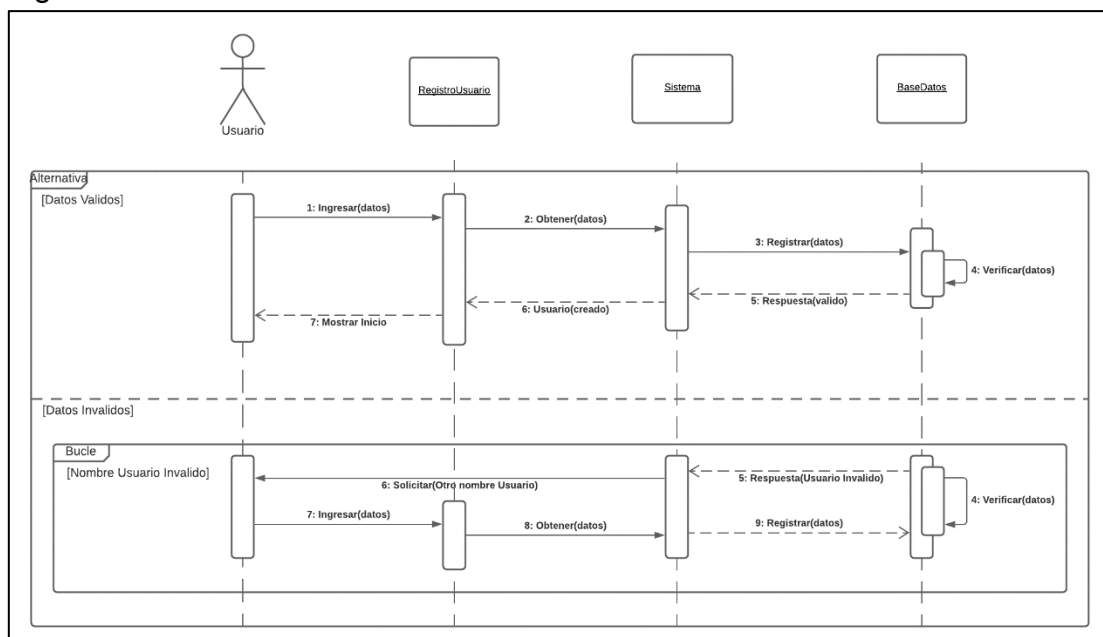


VII. Diagrama de Secuencias

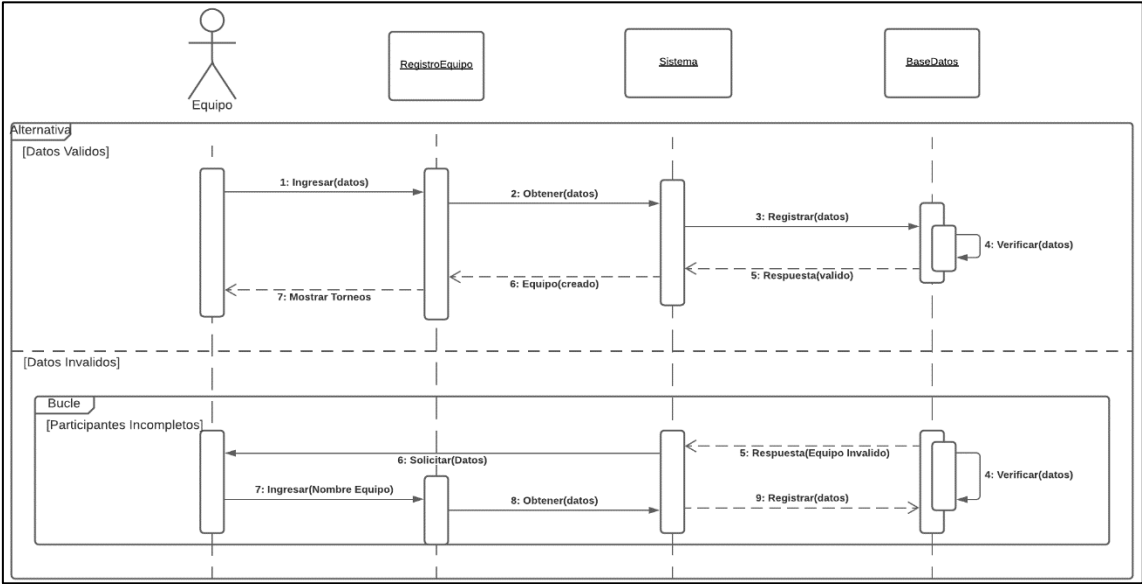
Ingresar al Sistema



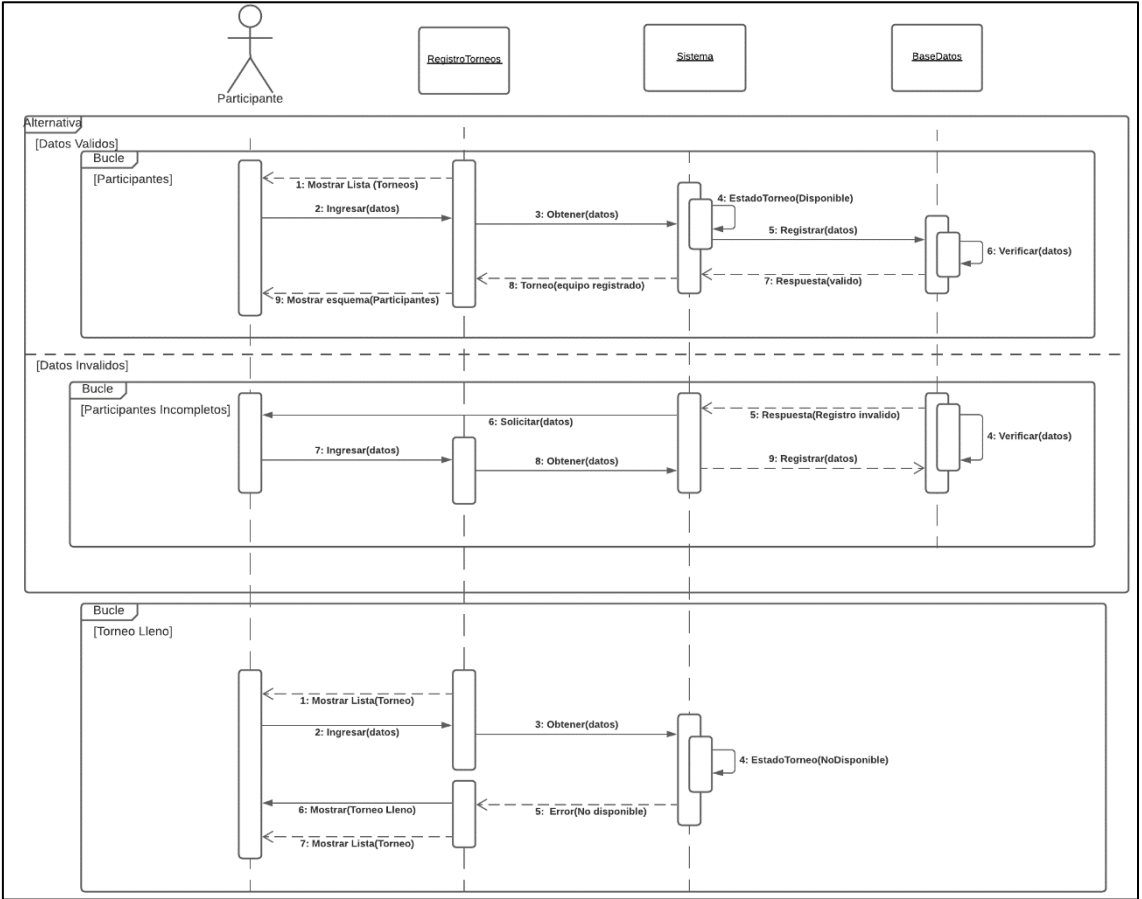
Registrar Usuario



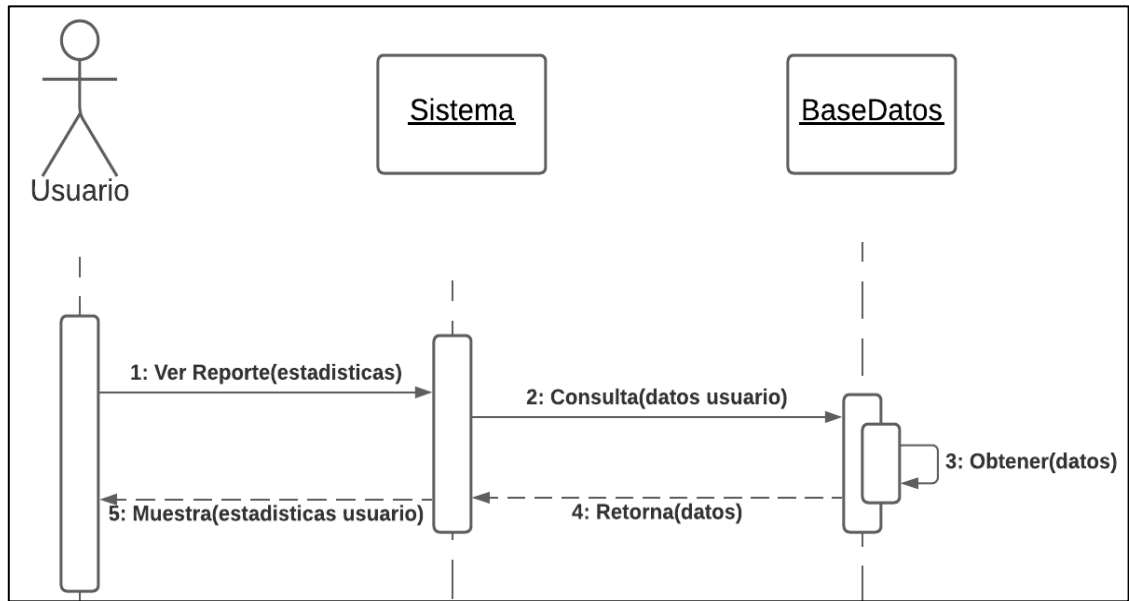
Registrar Equipo



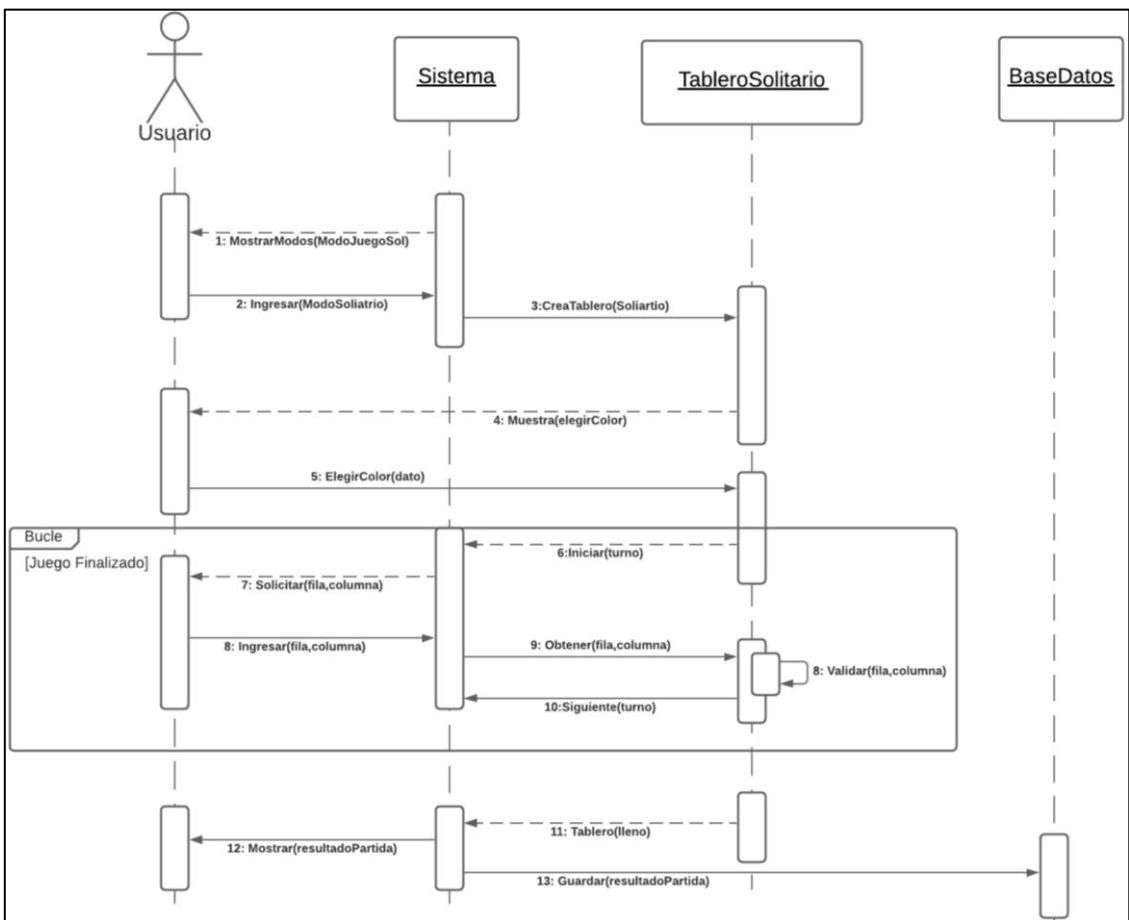
Registrar en Torneos



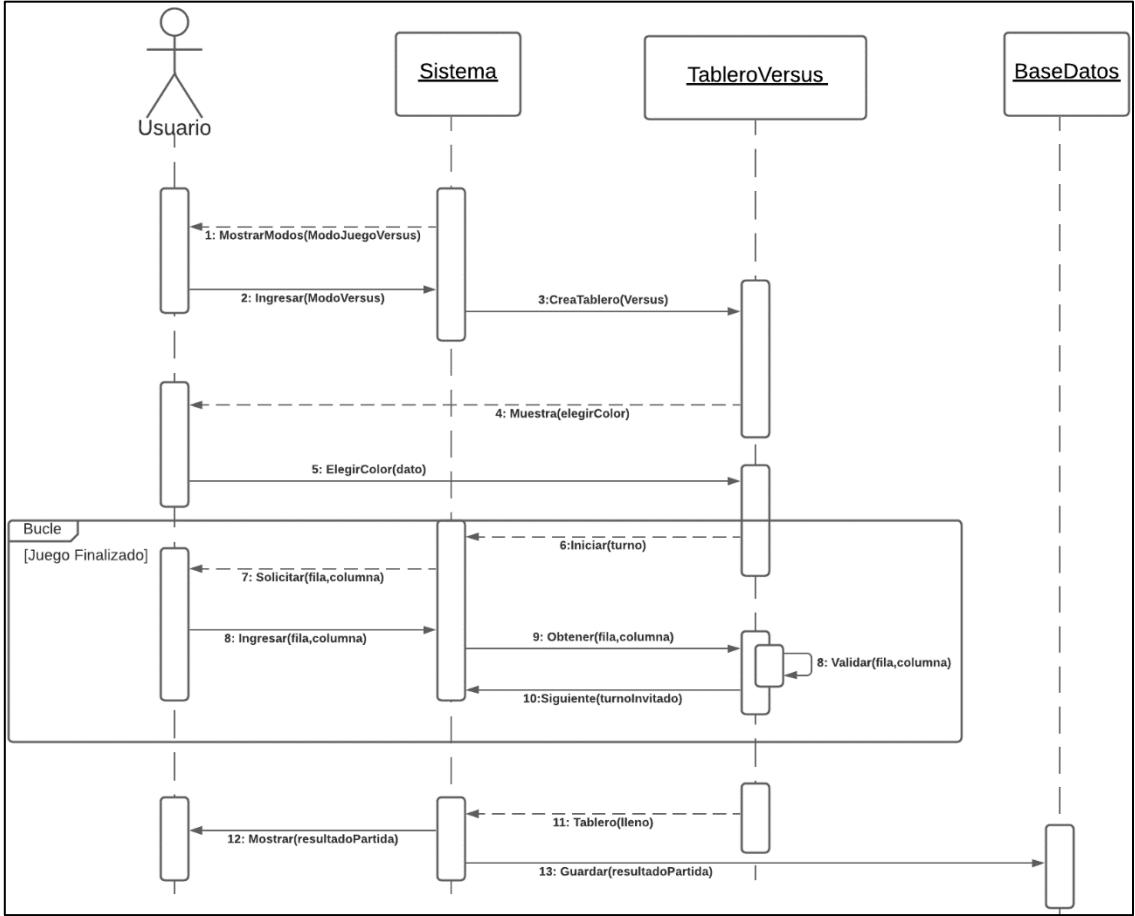
Gestionar Reportes



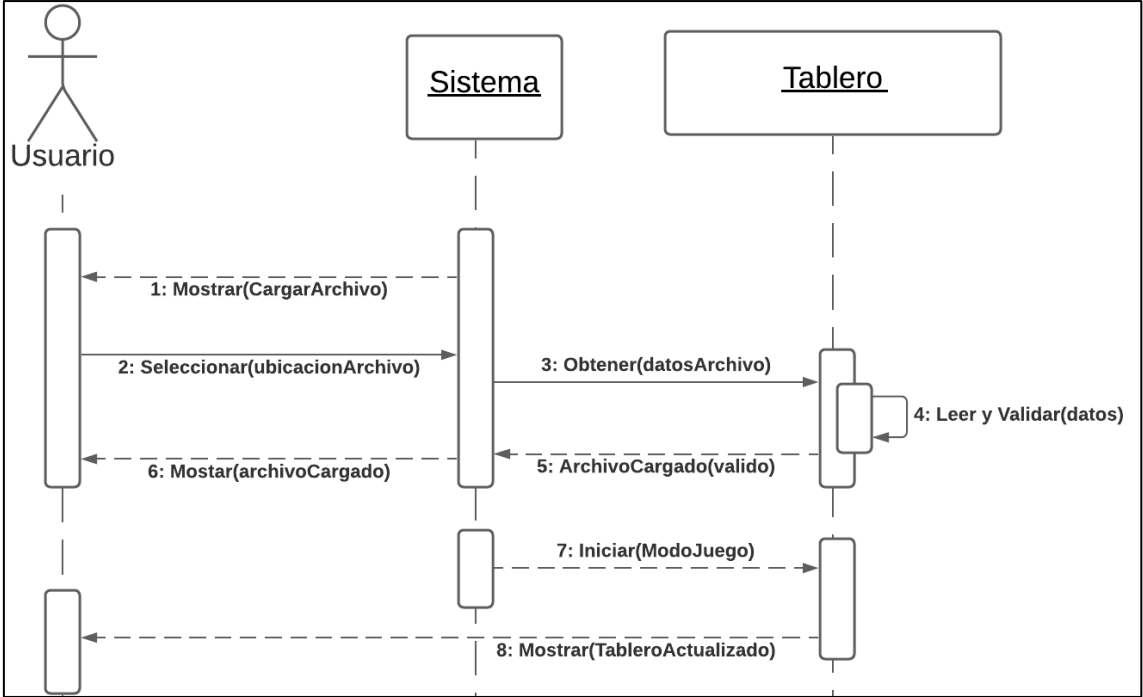
Crear Partida Solitario



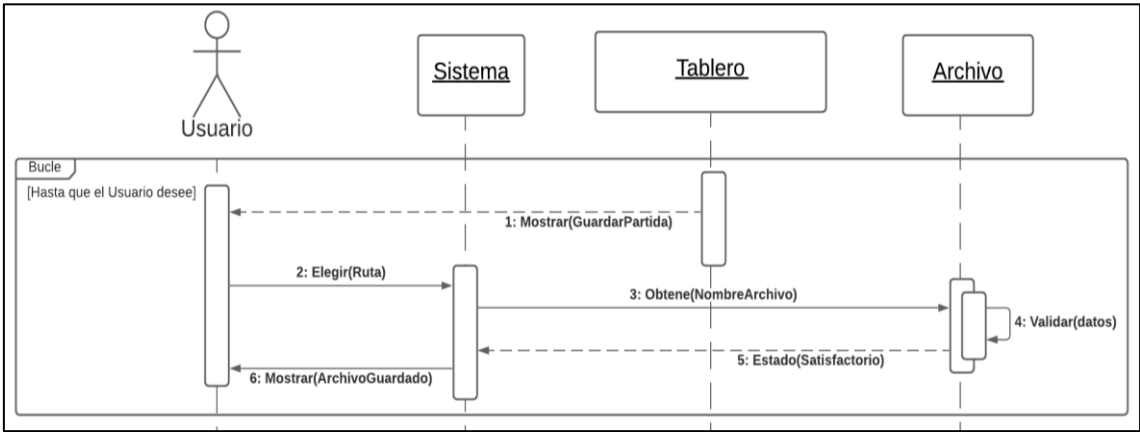
Crear Partida Versus



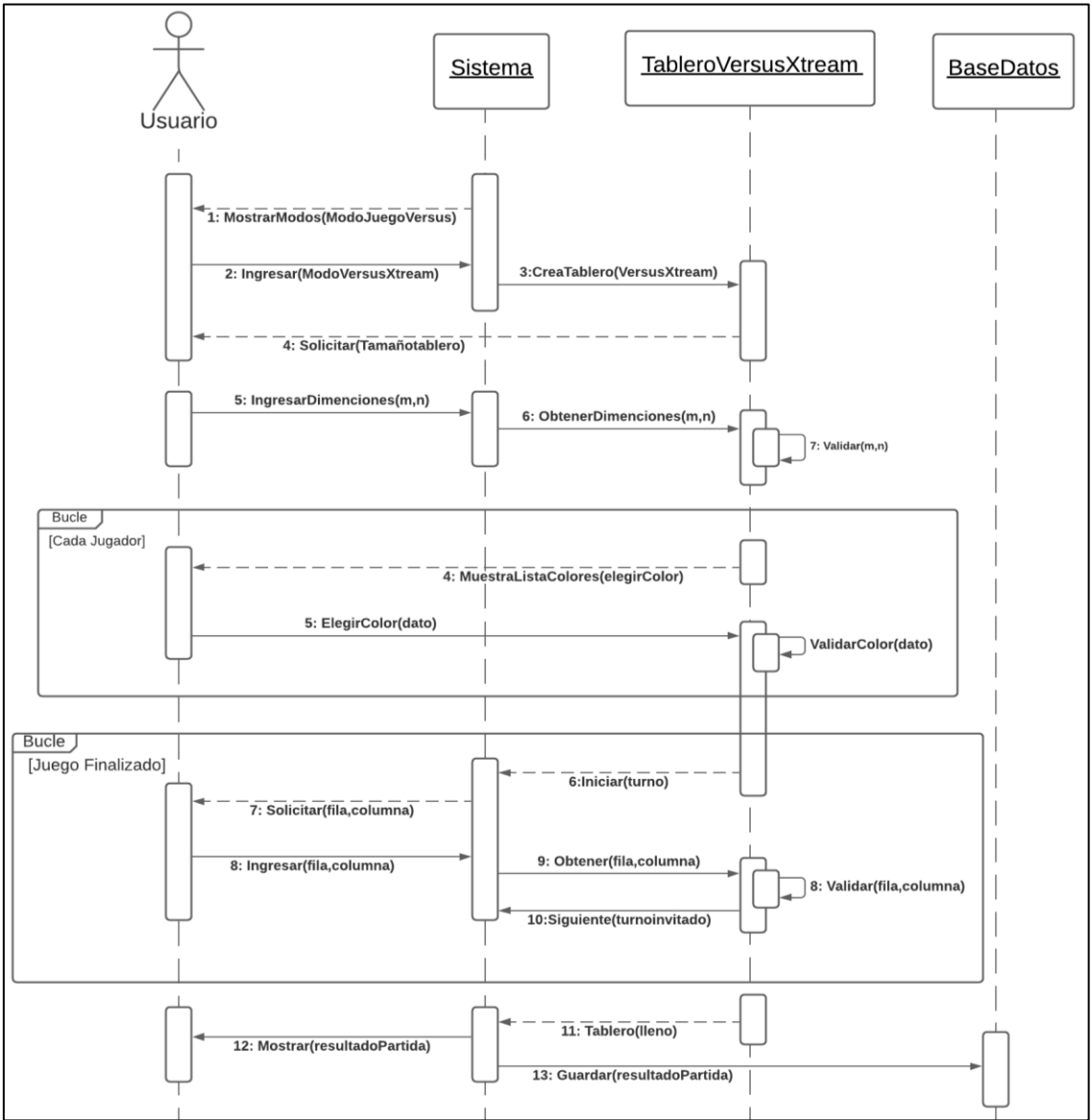
Cargar Archivos



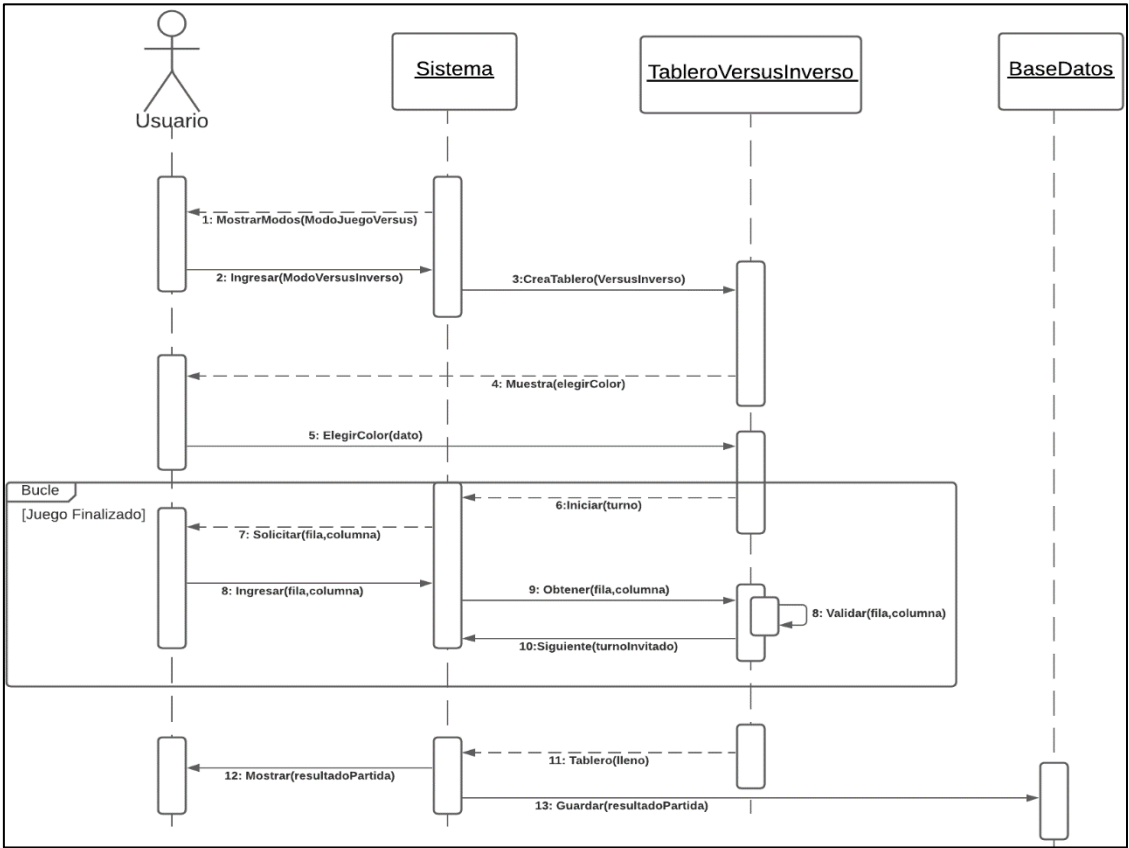
Guardar Archivos



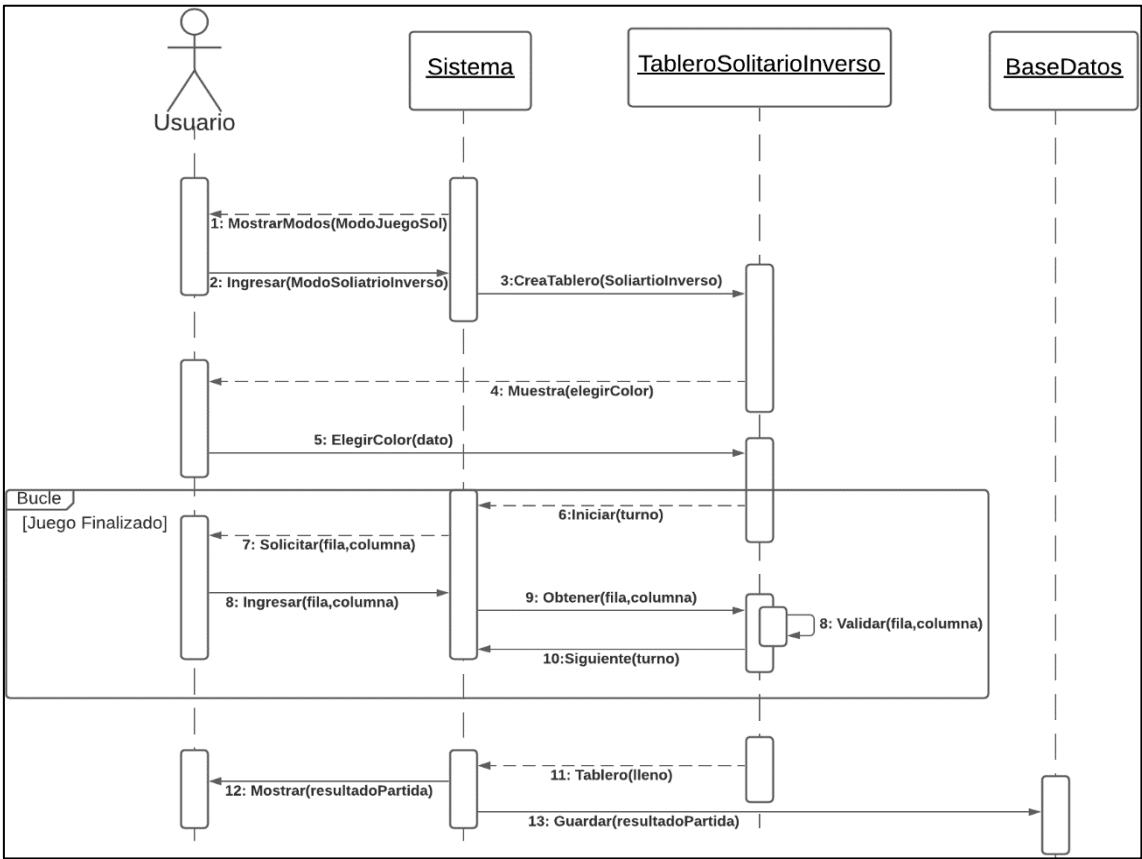
Crear Othello Xstream



Unirse Reto Inverso Versus

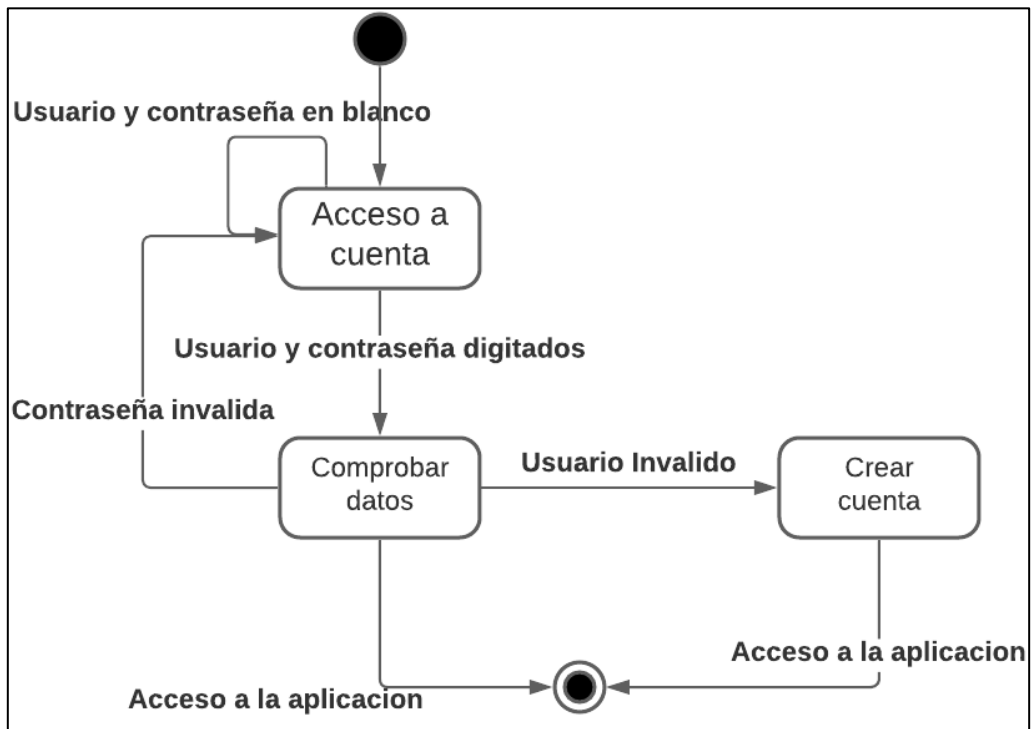


Crear Partida Solitario Inverso

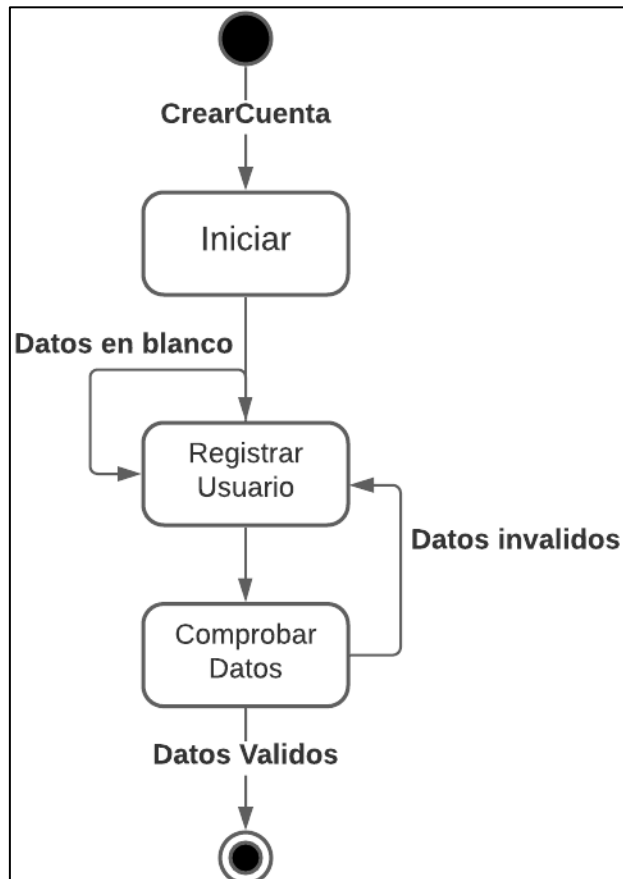


VIII. Diagrama de Estados

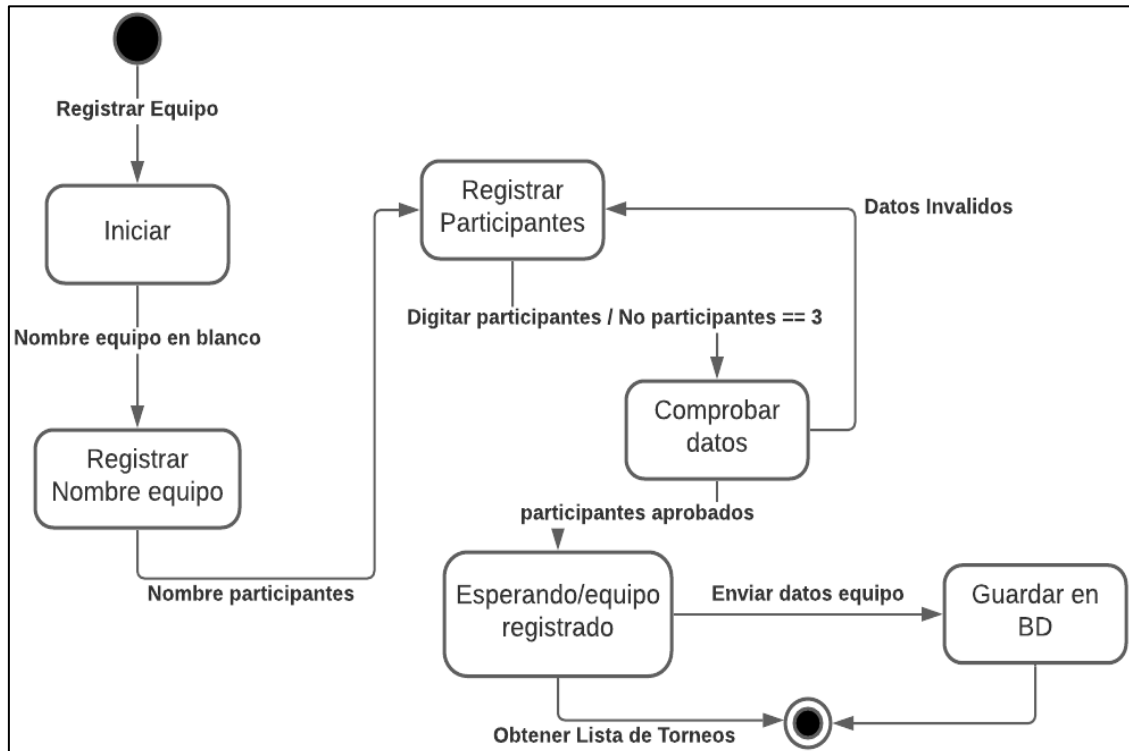
Ingresar al Sistema



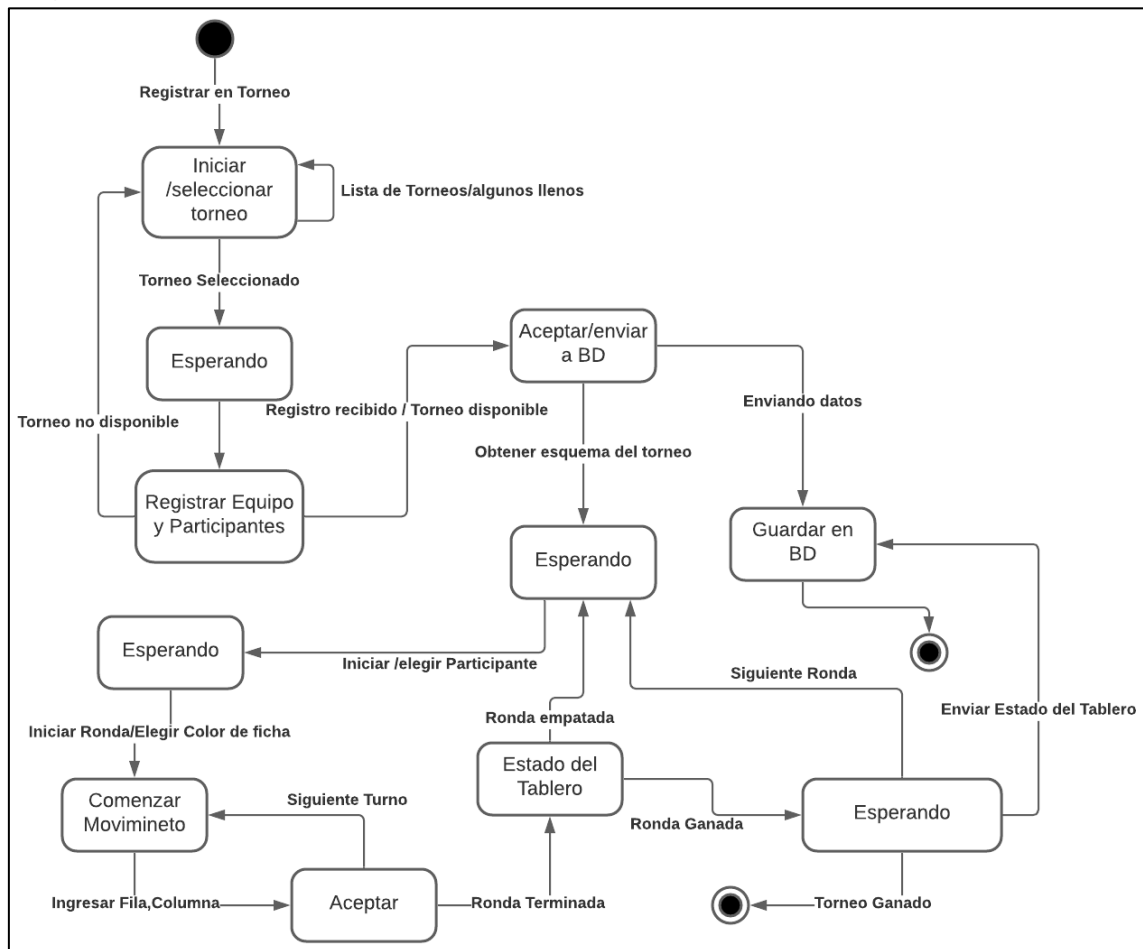
Registrar Usuario



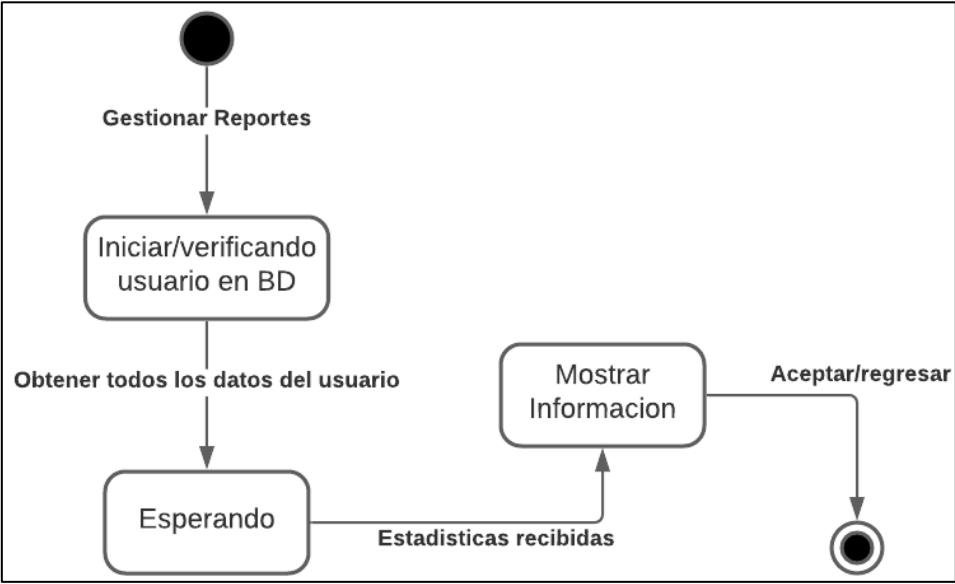
Registrar Equipo



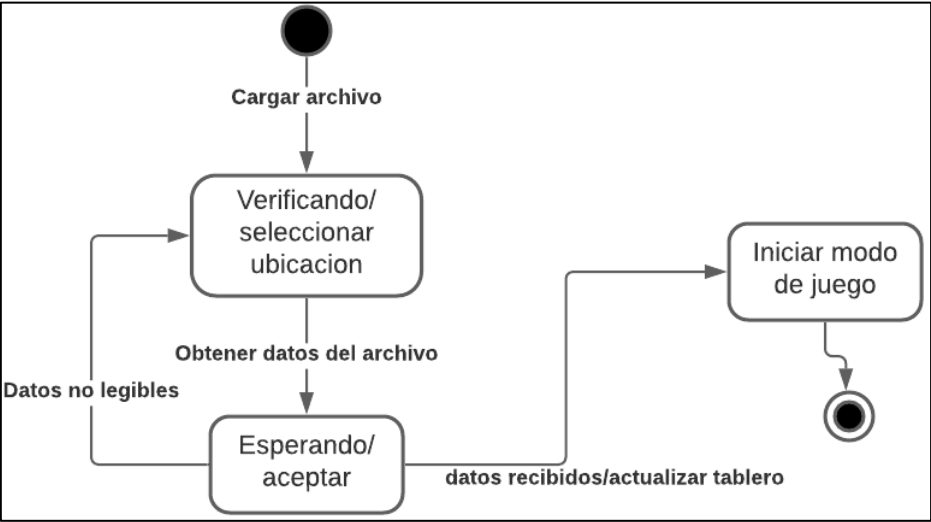
Registrar en Torneo



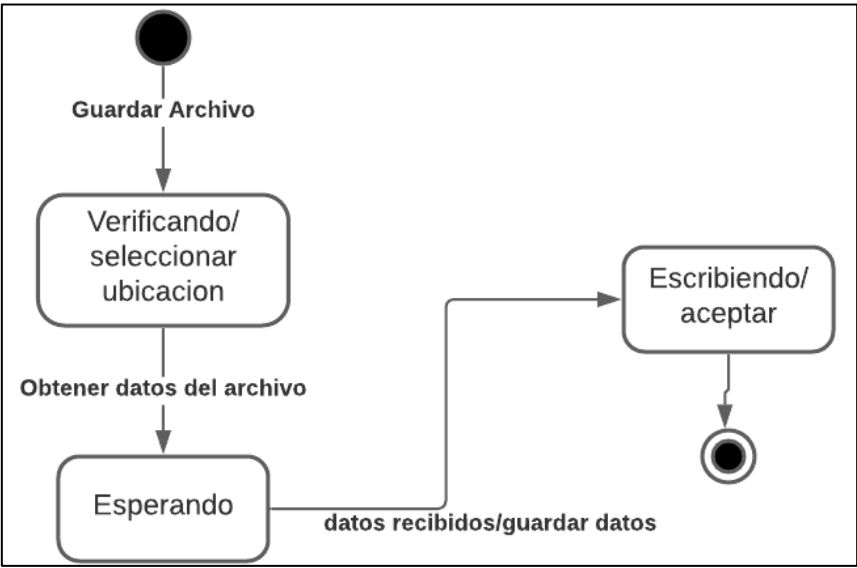
Gestionar Reportes



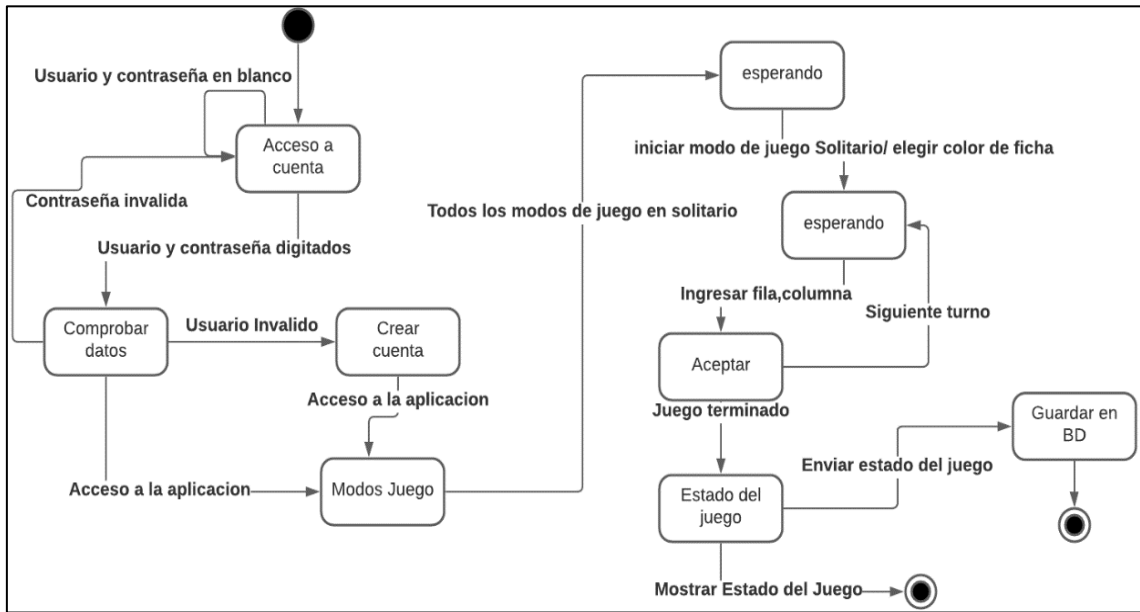
Cargar Archivos



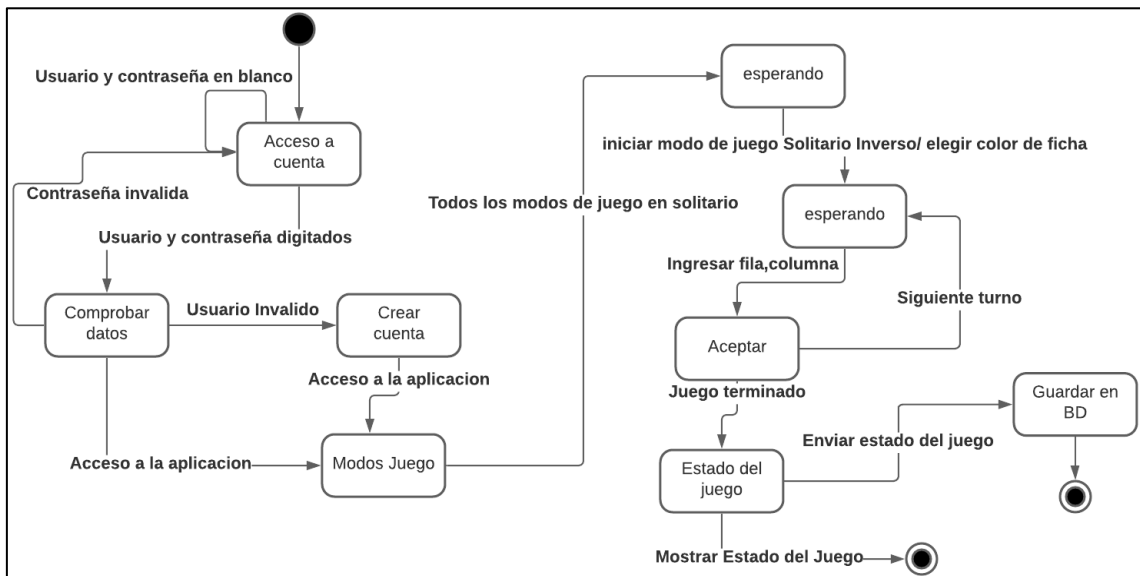
Guardar Archivos



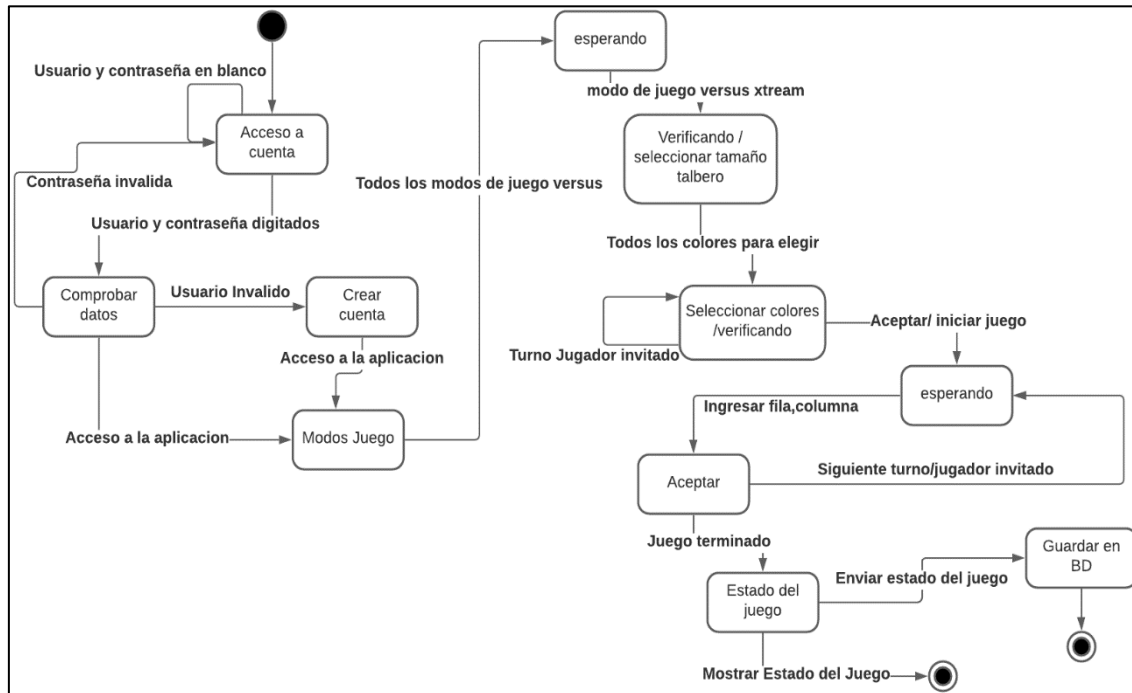
Crear una Partida Solitario



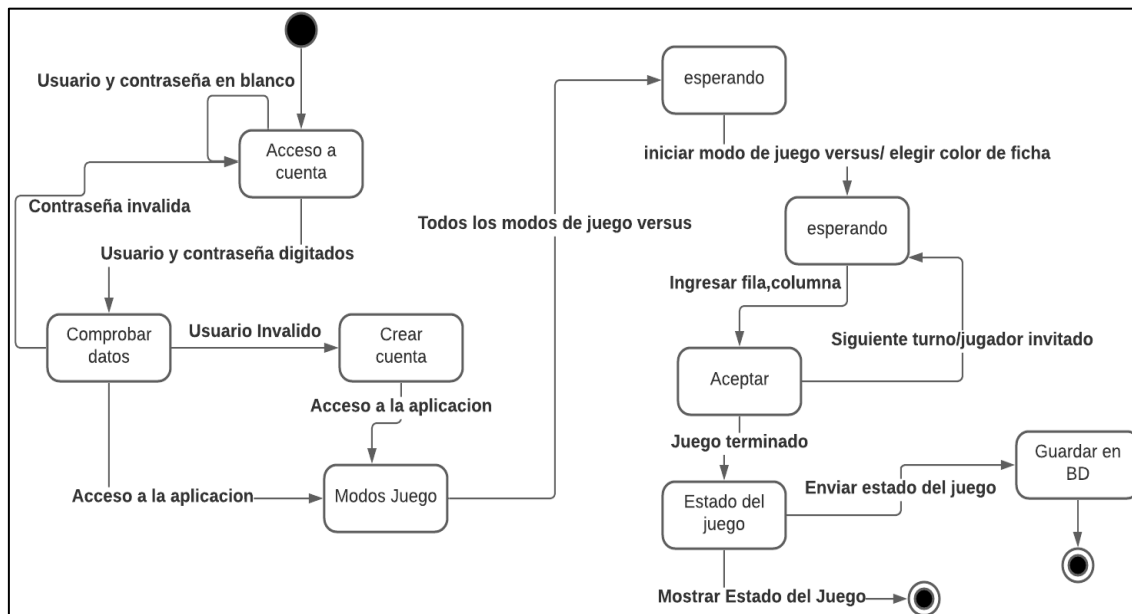
Crear Partida Solitario Inverso



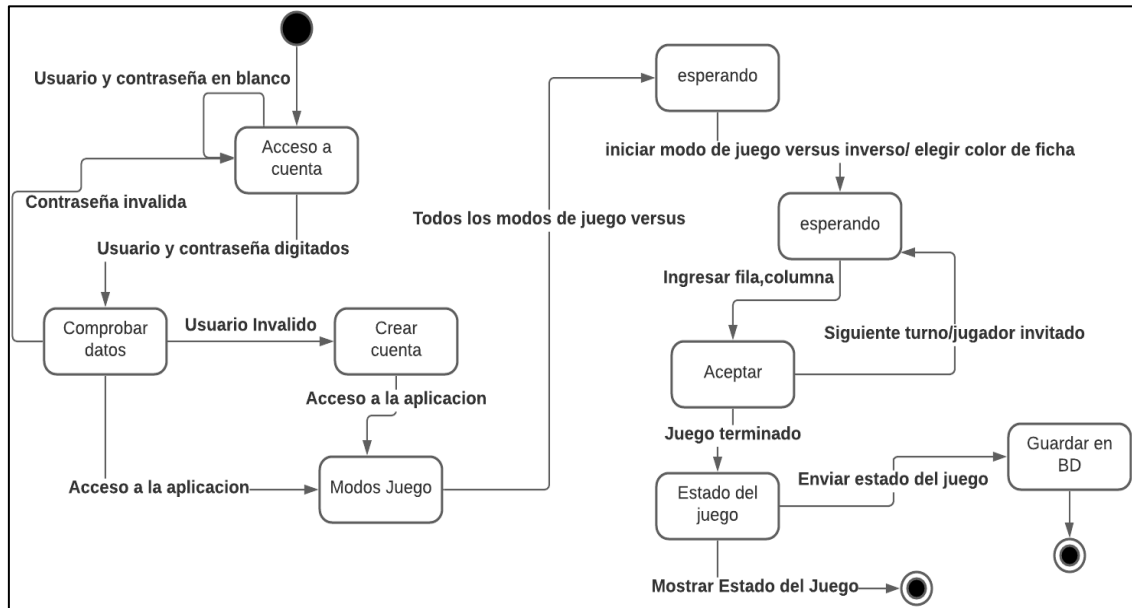
Crear Othello Xstream



Crear una Partida Versus

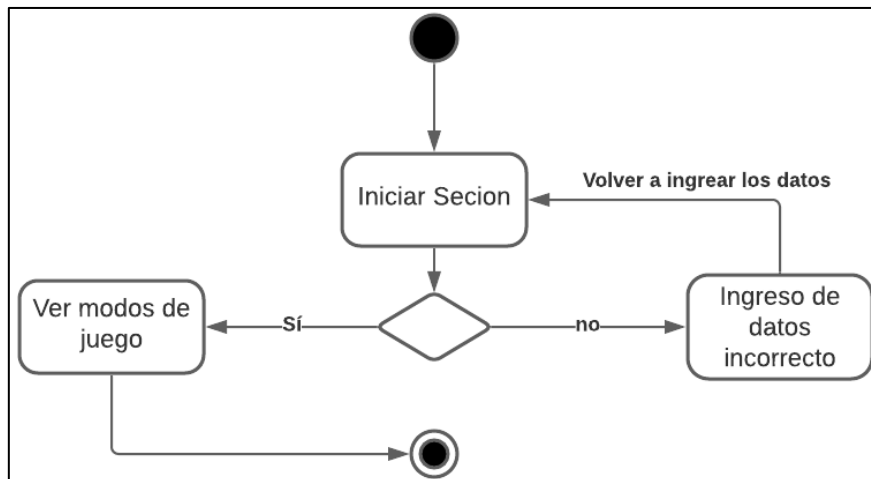


Unirse Reto Inverso Versus

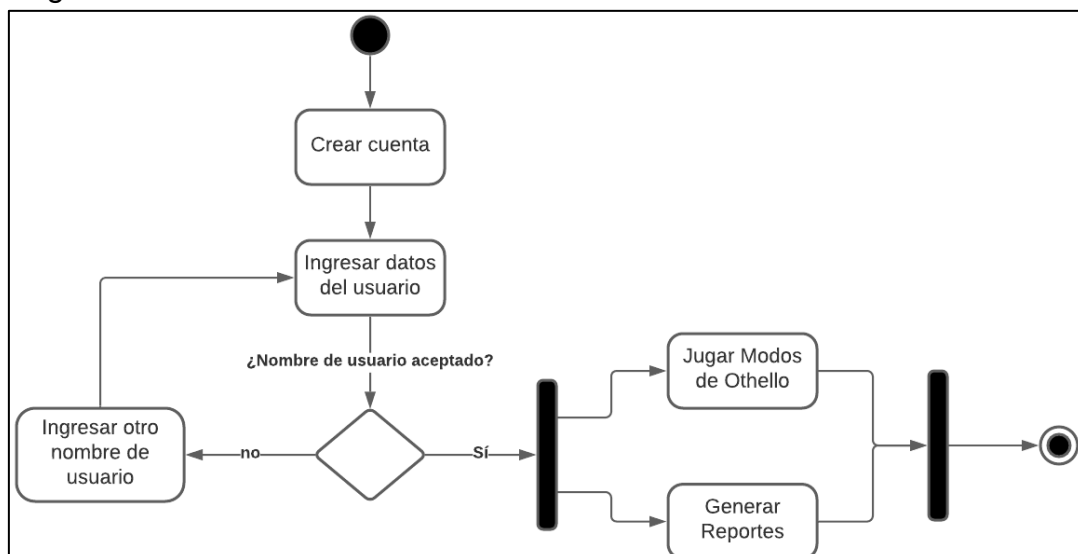


IX. Diagrama de Actividades

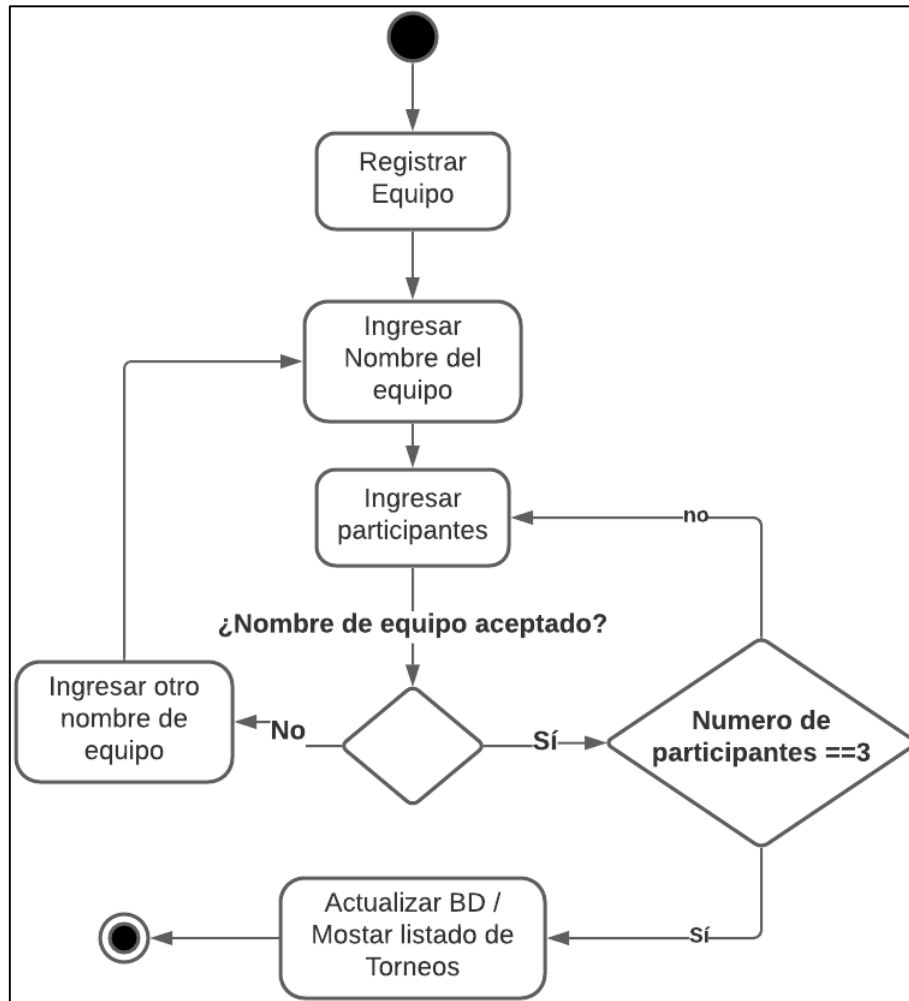
Ingresar al Sistema



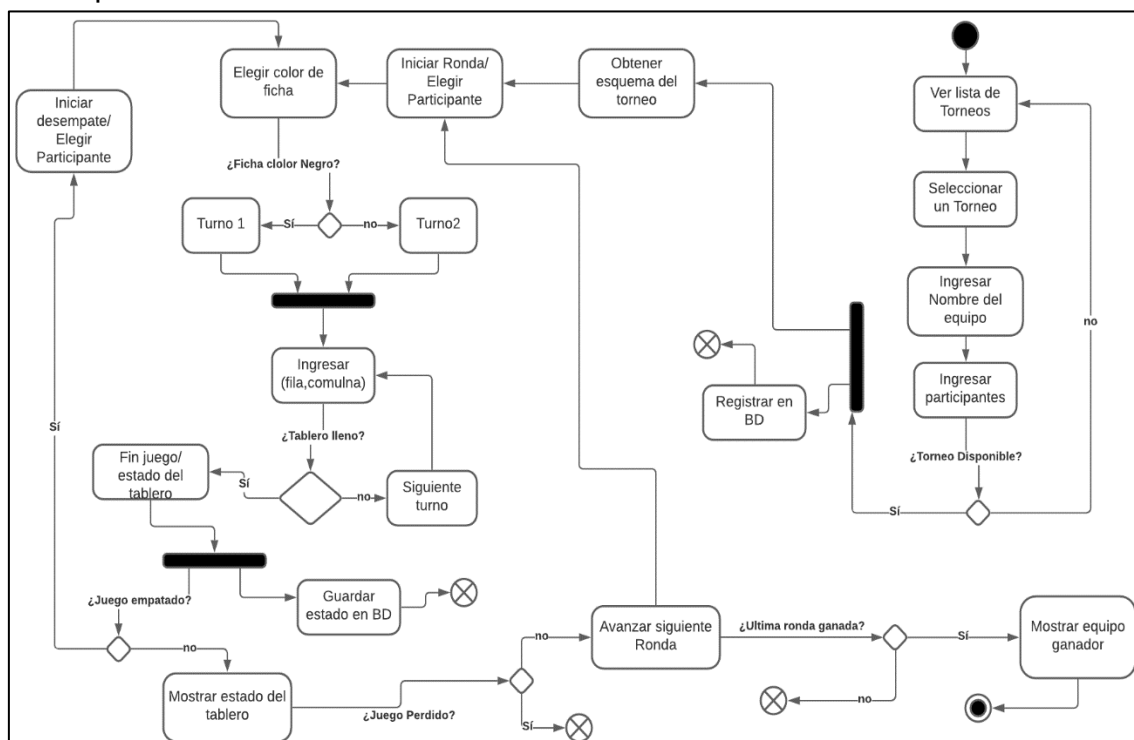
Registrar Usuario



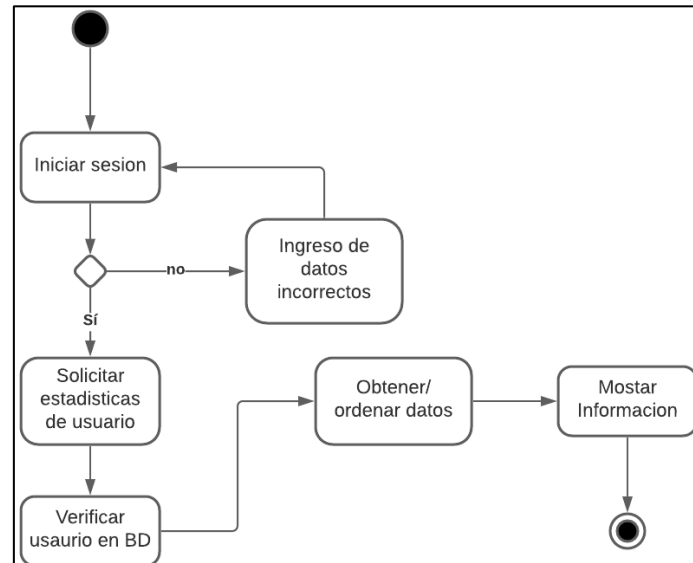
Registrar Equipo



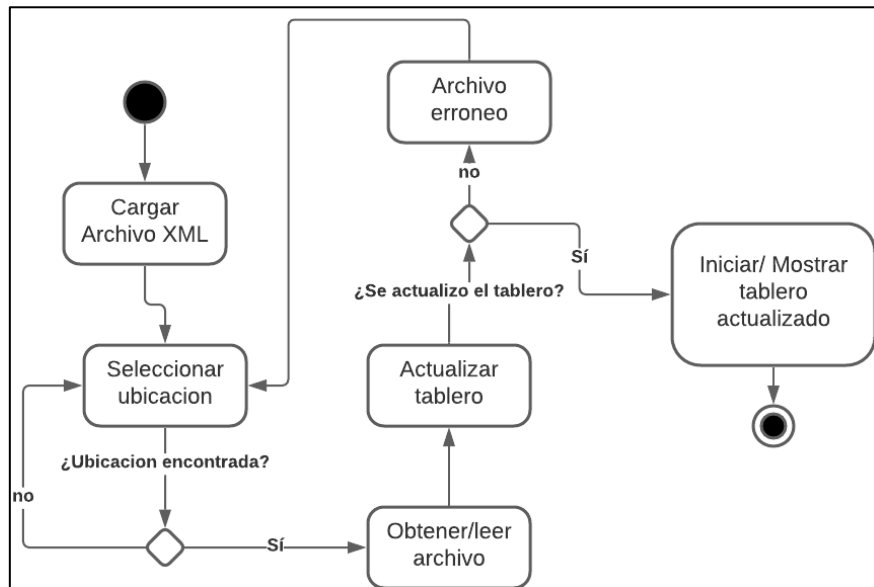
Participar en Torneo



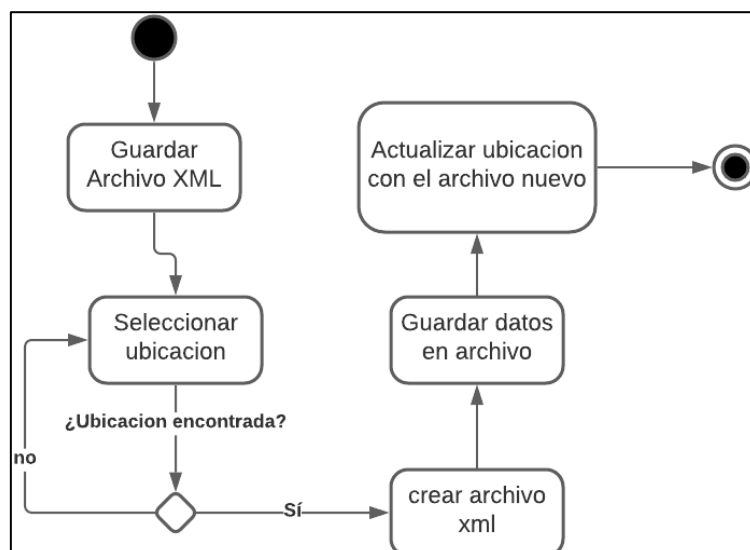
Gestionar Reportes



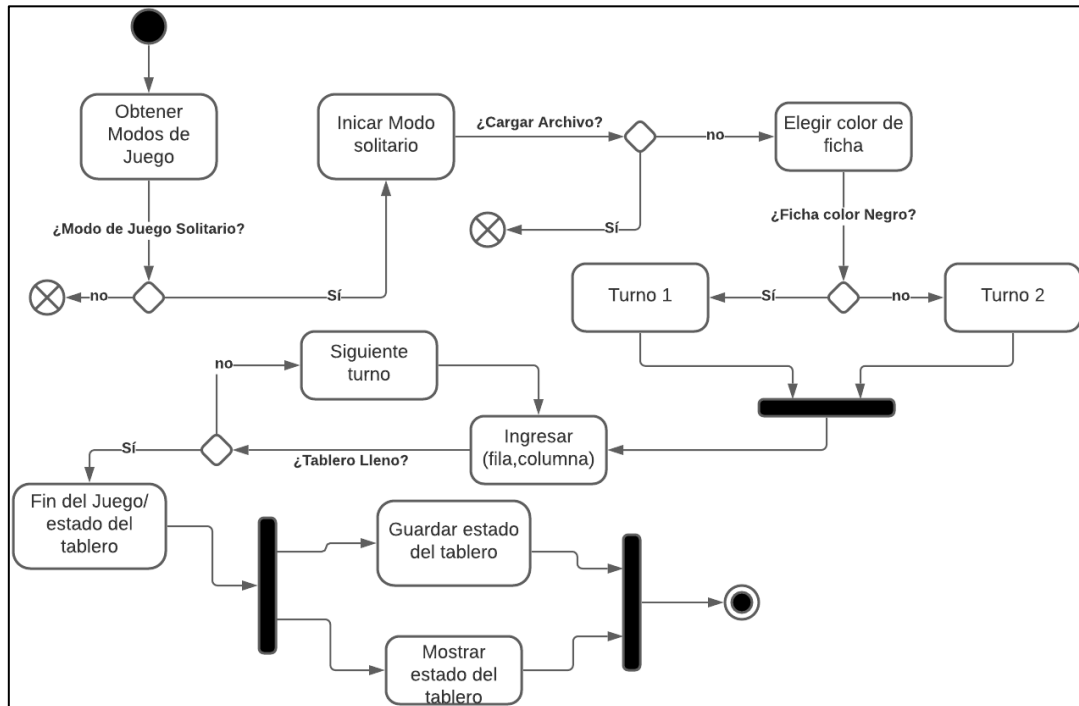
Cargar Archivos



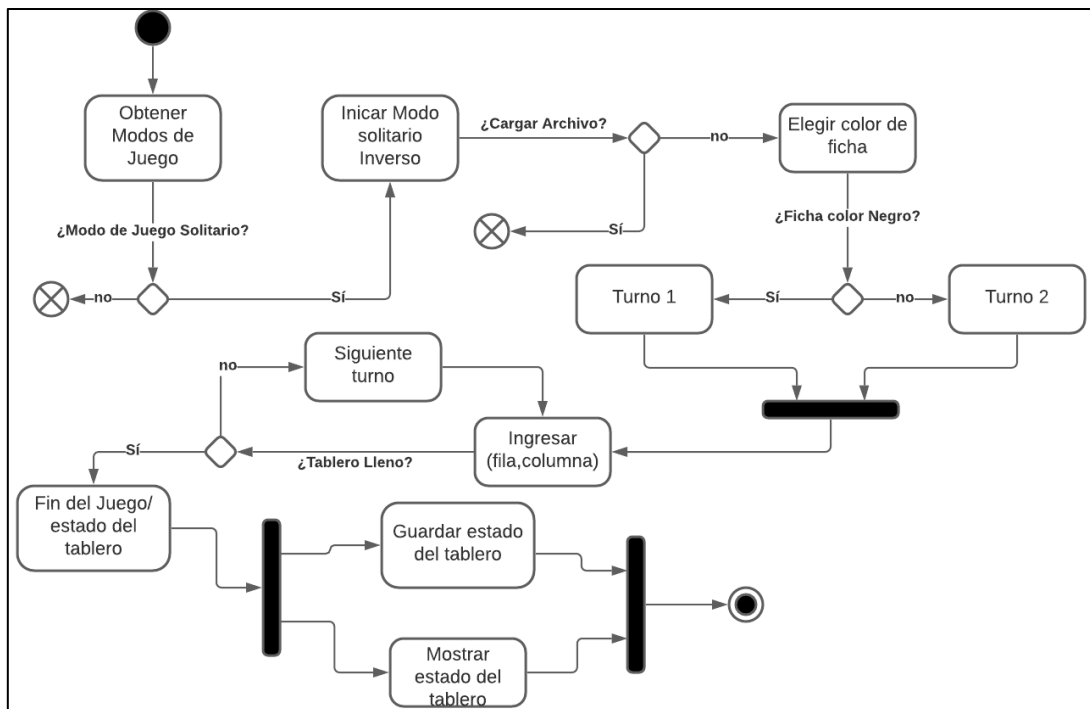
Guardar Archivos



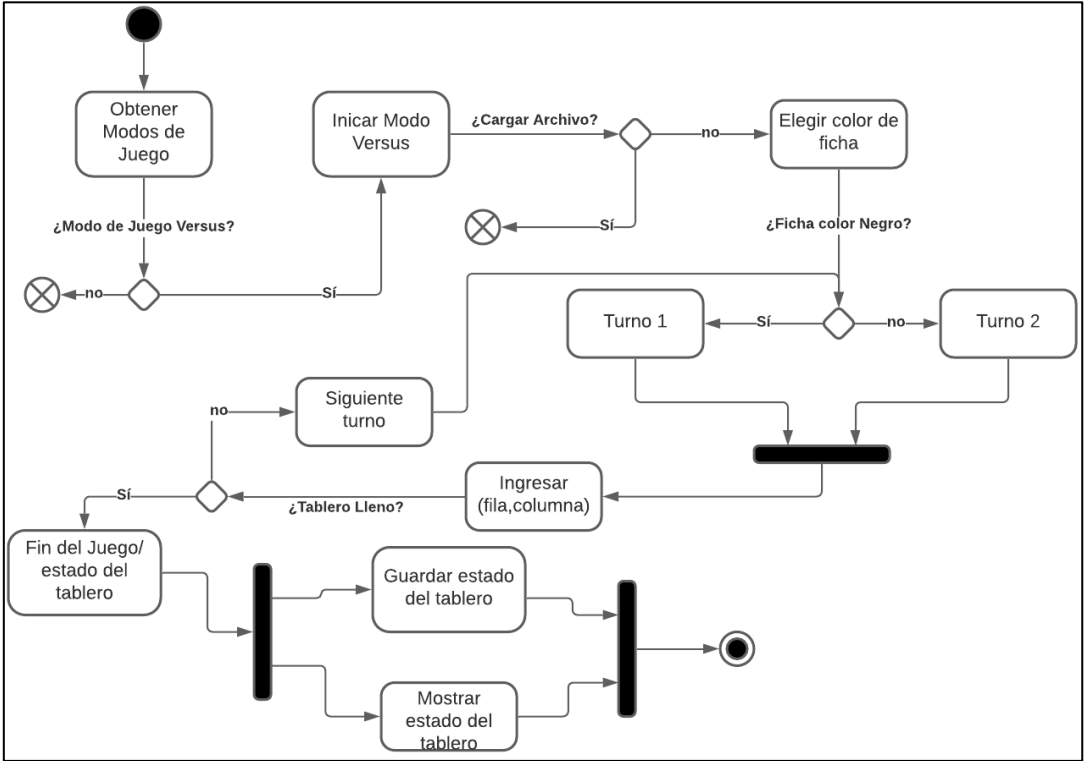
Crear una Partida Solitario



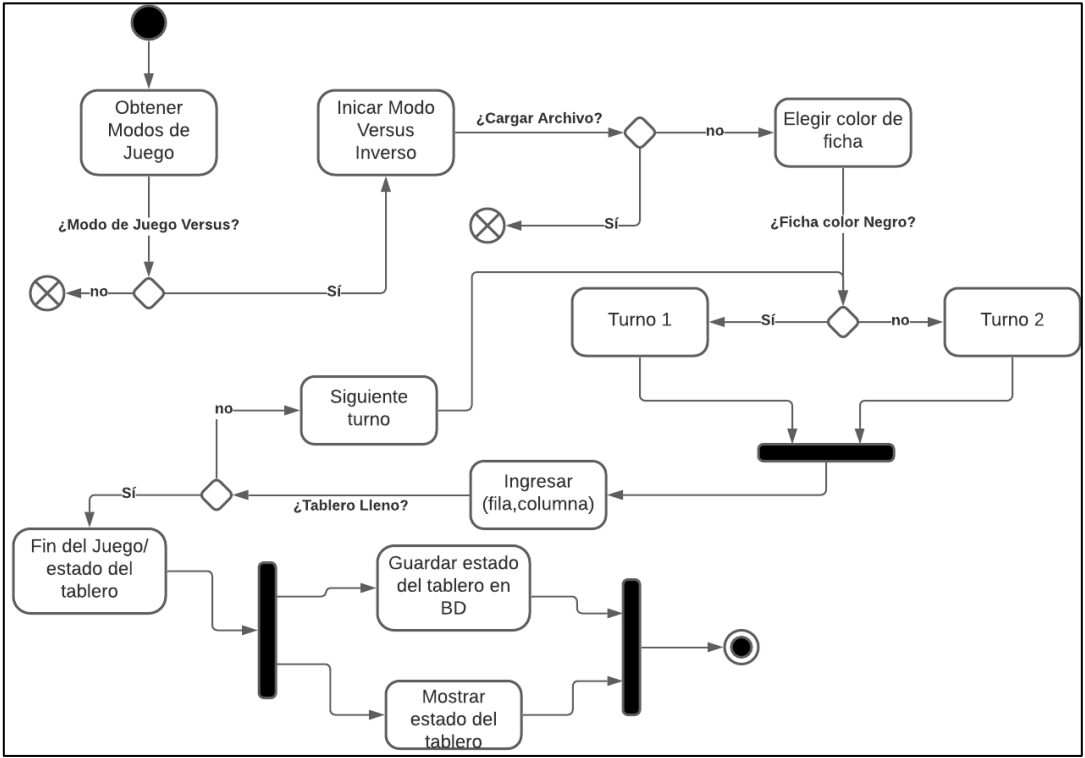
Crear Partida Solitario Inverso



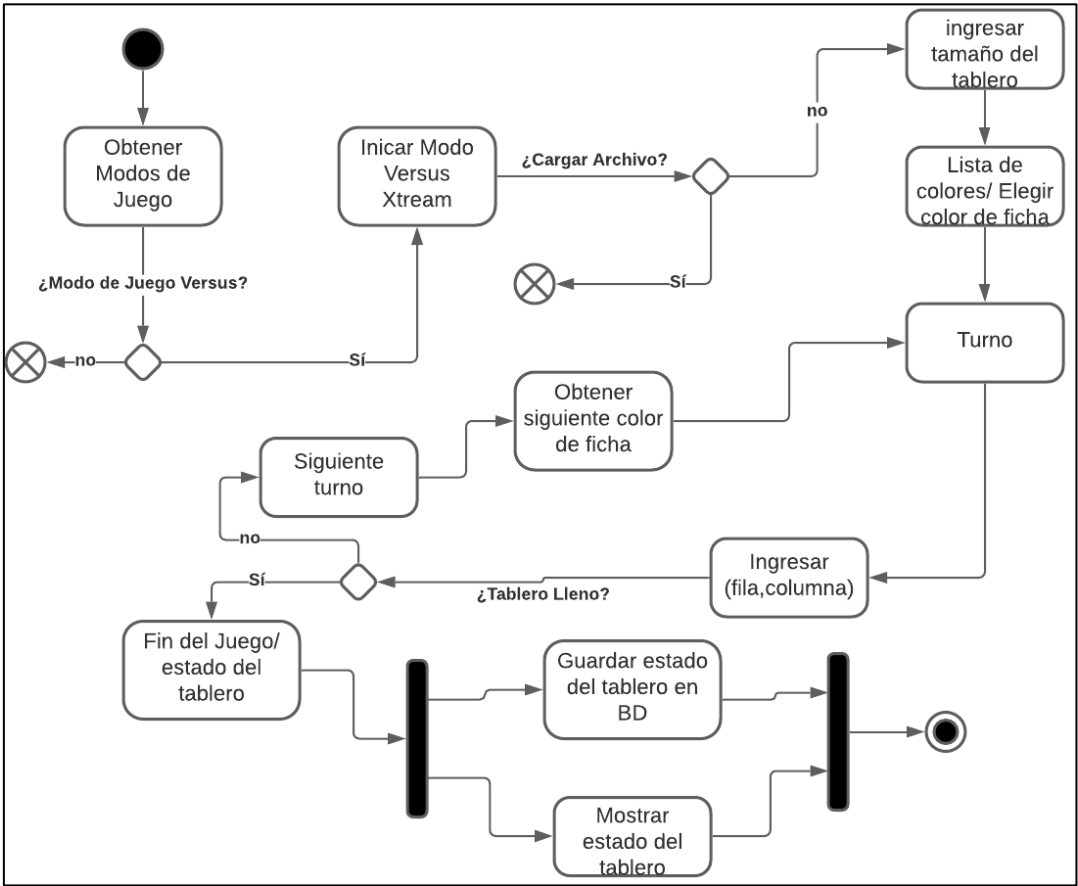
Crear Partida Versus



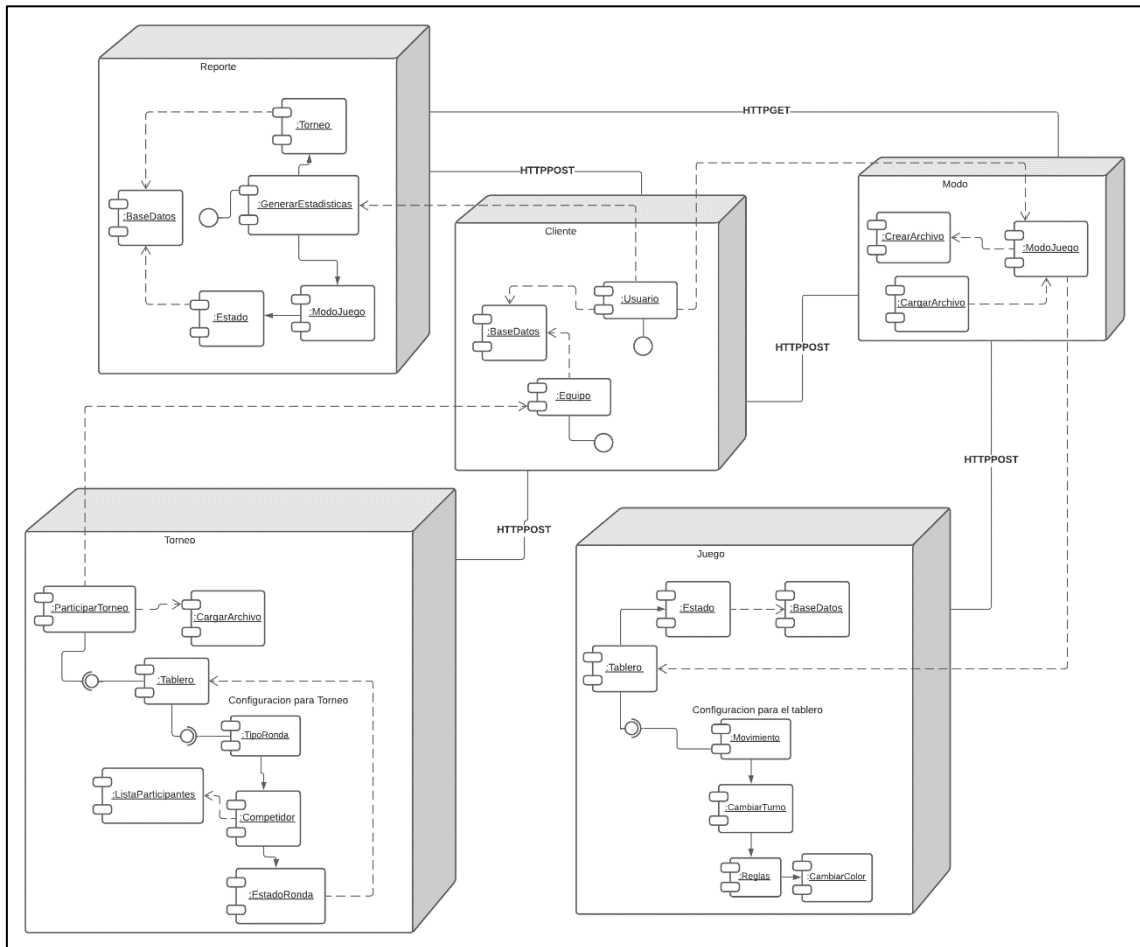
Unirse Reto Inverso Versus



Crear Othello Xtreem

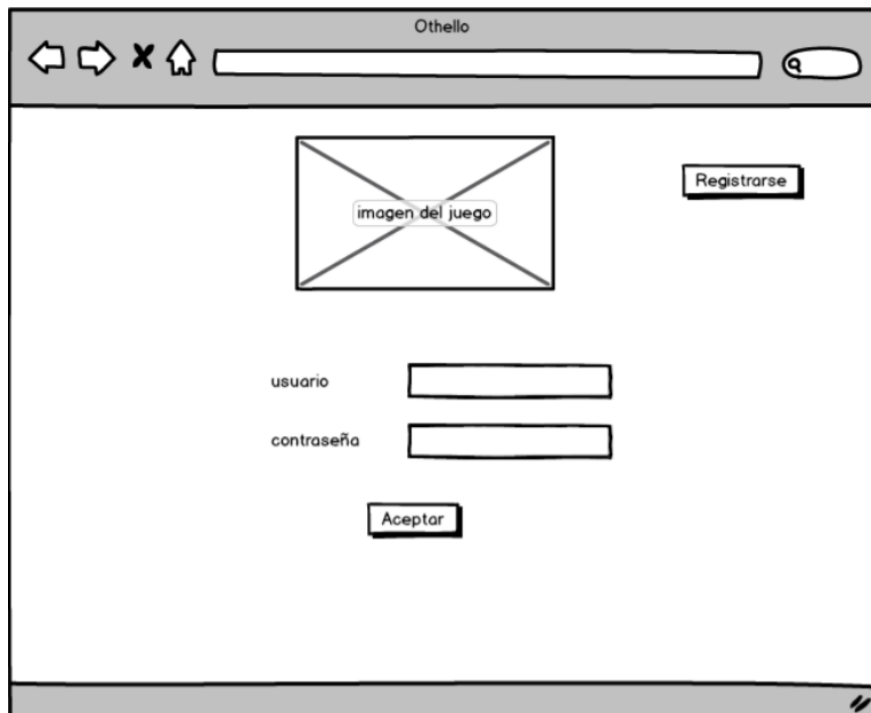


X. Diagrama de Despliegue incluyendo componentes



XI. Look & Feel de la solución

Página index



Página de inicio

The screenshot shows a web browser window with the title "Othello". The address bar is empty. The page content includes a header with "Usuario" and "individual". Below this is a large rectangular area labeled "tablero" (board) in the center. To the right of the board are three buttons: "Cargar partida", "Nueva partida", and "Salir". At the bottom left, the text "Movimientos realizados" is visible.

Página de Registro

Othello

Nombres

Apellidos

Usuario

Contraseña

Correo

Fecha de Nac.

DD/MM/AAAA

Pais

Guatemala

Mexico

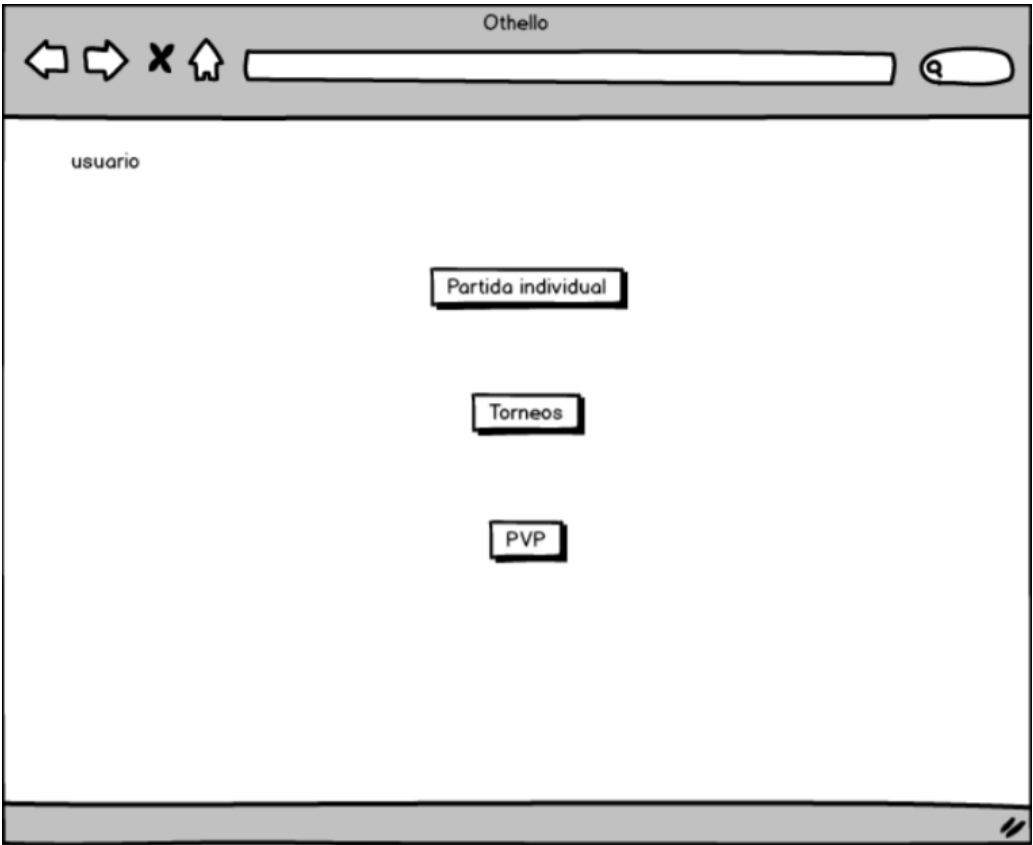
Costa Rica

Otro...

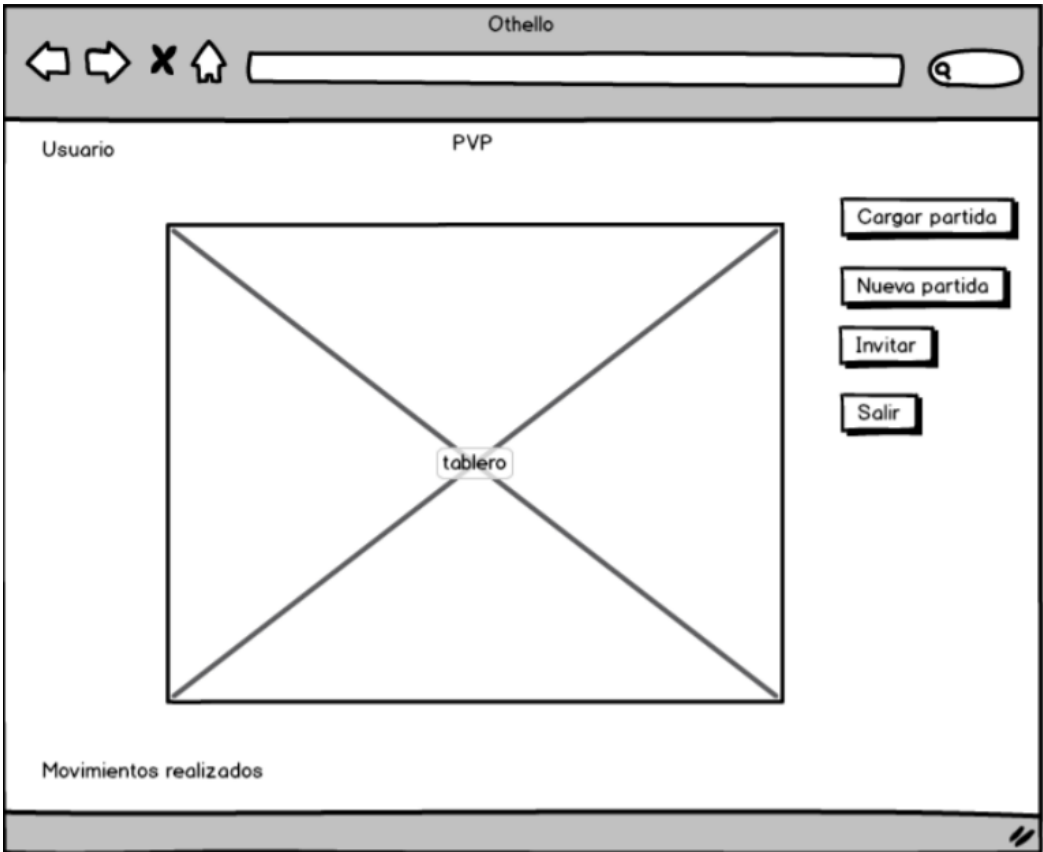
Aceptar

Regresar

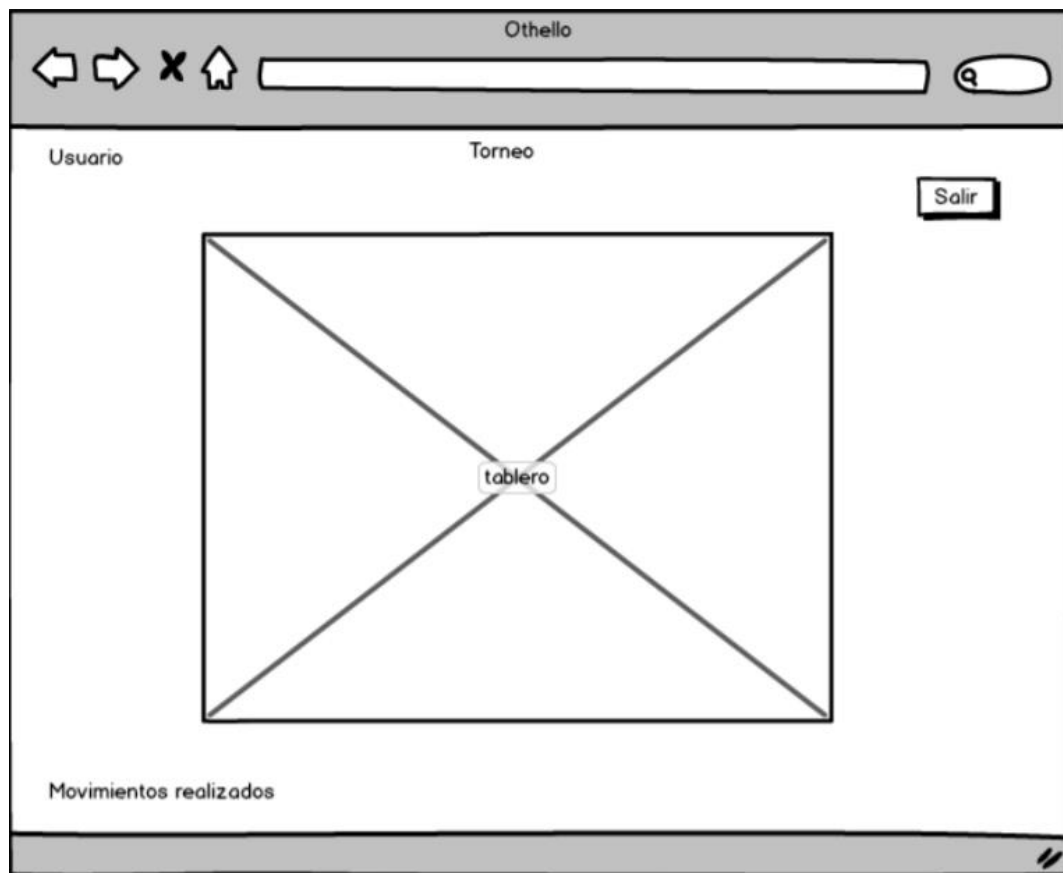
Opciones de Jugabilidad



Jugador vs Jugador



Página de Torneo



XII. Planificación del Proyecto

[illegible]

Id	Modo de	Nombre de tarea	Duración	Comienzo	Fin	S	D	19 oct '20	26 oct '20
								L M X J V S D	L M X J V S
18	1	Ventana del Perfil del Usuario	2 días	mié 21/10/20	jue 22/10/20				
19		Codificación							
20		Ancho y Alto del tablero para Othello Xstream	2 días	dom 18/10/20	lun 19/10/20				
21		Colores por Jugador	2 días	dom 18/10/20	lun 19/10/20				
22		Validación de los colores de cada Jugador	2 días	dom 18/10/20	lun 19/10/20				
23		Validación del Reto Inverso	2 días	dom 18/10/20	lun 19/10/20				
24		Tablero de Apertura Personalizada	2 días	dom 18/10/20	lun 19/10/20				
25		Validación del Tablero de Apertura Personalizada	2 días	dom 18/10/20	lun 19/10/20				
26		Validación de Funcionalidad del Archivo XML	2 días	dom 18/10/20	lun 19/10/20				
27		Validaciones del Archivo XML	2 días	dom 18/10/20	lun 19/10/20				
28		Crear Cronometro para Othello y Othello Xstream	1 día	dom 18/10/20	dom 18/10/20				
29		Creación de Campeonato	5 días	jue 22/10/20	mié 28/10/20				
30		Validación de los datos del Campeonato	5 días	jue 22/10/20	mié 28/10/20				
31		Validación de Equipos	5 días	vie 23/10/20	jue 29/10/20				
32		Validación de Puntos por Partida	4 días	vie 23/10/20	mié 28/10/20				
33		Validación de Empate	4 días	sáb 24/10/20	mié 28/10/20				
34		Creación del Esquema del torneo	4 días	lun 26/10/20	jue 29/10/20				
35		Validación del Archivo XML para los Campeonatos	4 días	lun 26/10/20	jue 29/10/20				
36		Validación de Siguiete Ronda	4 días	mar 27/10/20	vie 30/10/20				
37		Validaciones del Perfil de Usuario	3 días	mié 28/10/20	vie 30/10/20				

XIII. Script de la base de datos

```
create database Proyecto_IPC2
use Proyecto_IPC2
```

```
create table Admin(
id_admin int IDENTITY(1,1) primary key,
Nombreadmin varchar(45),
Nombre varchar(45),
Apellido varchar(45),
Contra varchar(45),
Fecha_nac Date,
Pais varchar(45),
Correo varchar(50));
```

```
create table Usuario(
id_us int IDENTITY(1,1) primary key,
Nombre_usuario varchar(45),
Nombre varchar(45),
```

```
Apellido varchar(45),  
Contra varchar(45),  
Fecha_nac Date,  
Pais varchar(45),  
Correo varchar(50));
```

```
create table Torneo(  
id_torneo int IDENTITY(1,1) primary key,  
Nombre varchar(45),  
id_admin int,  
foreign key(id_admin) REFERENCES Admin(id_admin));
```

```
create table Equipo(  
id_equipo int IDENTITY(1,1) primary key,  
NombreEquipo varchar (15),  
NombreParticipante1 varchar(30) NULL,  
NombreParticipante2 varchar(30) NULL,  
NombreParticipante3 varchar(30) NULL,  
id_us int,  
foreign key (id_us) REFERENCES Usuario(id_us)  
);
```

```
create table ListaTorneo(  
id_listaTorneo int IDENTITY(1,1) primary key,  
id_torneo int,  
id_equipo int,  
foreign key(id_torneo) REFERENCES Torneo(id_torneo),  
foreign key (id_equipo) REFERENCES Equipo(id_equipo)  
);
```

```
create table Invitacion(  
id_invitacion int IDENTITY(1,1) primary key,  
contenido varchar(100),  
id_us int,  
foreign key (id_us) REFERENCES Usuario(id_us)  
);
```

```
create table Solitario(  
id_solitario int IDENTITY(1,1) primary key,  
id_us int,  
foreign key(id_us) REFERENCES Usuario(id_us)  
);
```

```
create table ModoJuegoSol(  
id_modosol int IDENTITY(1,1) primary key,  
Modo_normal int null,  
Modo_inverso int null,  
Modo_xtream int null,  
id_solitario int,  
foreign key(id_solitario) REFERENCES Solitario(id_solitario)
```

);

```
create table Detalle_torneo(  
id_detalle_torneo int IDENTITY(1,1) primary key,  
id_torneo int,  
id_us int,  
foreign key(id_torneo) REFERENCES Torneo(id_torneo),  
foreign key (id_us) REFERENCES usuario(id_us)  
);
```

```
create table Versus(  
id_versus int IDENTITY (1,1) primary key,  
id_invitacion int,  
id_us int,  
foreign key(id_invitacion) REFERENCES Invitacion(id_invitacion),  
foreign key(id_us) REFERENCES Usuario(id_us)  
);
```

```
create table ModoJuegoVer(  
id_Modo int IDENTITY(1,1) primary key,  
Modo_normal int null,  
Modo_inverso int null,  
Modo_xstream int null,  
id_versus int,  
foreign key(id_versus) REFERENCES Versus(id_versus)  
);
```

```
create table Detalle_Solitario(  
id_detalle_solitario int IDENTITY (1,1) primary key,  
movimiento int null,  
ganado int null,  
perdido int null,  
empatado int null,  
id_modo_sol int,  
foreign key (id_modo_sol) REFERENCES ModoJuegoSol(id_modo_sol)  
);
```

```
create table Partido(  
id_partido int IDENTITY (1,1) primary key,  
ronda int,  
movimineto int,  
id_detalle_torneo int,  
foreign key (id_detalle_torneo) REFERENCES  
Detalle_torneo(id_detalle_torneo)  
);
```

```
create table Tipo(  
id_tipo int IDENTITY (1,1) primary key,  
ganado int null,  
perdido int null,
```

```
empatado int null,  
id_partido int,  
foreign key (id_partido) REFERENCES Partido(id_partido)  
);
```

```
create table Detalle_Versus(  
id_detalle_versus int IDENTITY(1,1) primary key,  
movimiento int null,  
ganado int null,  
perdido int null,  
empatado int null,  
id_Modo int,  
foreign key (id_Modo) REFERENCES ModoJuegoVer(id_Modo)  
);
```