

# DOCUMENTACION

## DEFINICION DE LA SOLUCION

### **Objetivos generales**

- Desarrollar un sistema de software a través de los conceptos de análisis y diseño de sistemas.
- Entender las tecnologías implementadas en el sistema de software diseñado.
- Conectar el back-end con el front-end
- Conexión de base de datos con el servidor.
- Lograr crear el diseño de la aplicación

### **Objetivos 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.

# ALCANCE DEL PROYECTO

Se realizó una aplicación web donde las tecnologías usadas se adaptan al modelo que se estará utilizando durante el desarrollo de la aplicación web donde se desarrollara todo el análisis y el diseño del sistema de una forma minimalista y comprensible para el cliente.

Se conectó a una base de datos donde esa base de datos se conecta con el servidor donde el cliente realiza una petición al servidor y le devuelve los datos que le han solicitado el gestor de base de datos utilizado fue SQL SERVER.

Se entendieron las tecnologías que se han estado utilizando durante el proyecto  
Se logró satisfactoriamente conectar el lado del front-end con el lado del back-end  
Donde el back-end está hecho con c# y el fron-end se realizó con .net

Se logró crear el diseño de la aplicación teniendo un diseño minimalista y amigable con la vista.

Se logró diseñar el torneo dando un modelo relacional donde relaciona la tabla torneo con la tabla partida esto esta relación ayuda a tener un mejor control de las partidas de cada torneo.

Se realizó el login usando las tecnologías necesarias y se llegó al objetivo a cumplir, este objetivo es la funcionalidad del login.

# REQUERIMIENTOS INICIALES DEL SISTEMA

## REQUERIMIENTOS FUNCIONALES

LOGIN

REGISTRO

CONECTAR A BASE DE DATOS

PARTIDAS

TABLERO

FICHAS

NOMBRE DE TORNEO

TORNEOS

HISTORIAL

REPORTES

## ATRIBUTOS DEL SISTEMA

SQL SERVER (18.6)

C#

NAVEGADOR

ASP.NET

VISUAL STUDIO

PROCESADOR CORI 3

# GLOSARIO

**1) Tablero:**

donde el usuario colocara sus fichas a la hora de que empiece el juego.

**2) Desarrollo:**

es quien realiza esta labor, normalmente sólo se preocupa por el funcionamiento del software, es tarea del diseñador web preocuparse del aspecto final(layout) de la página y del webmaster

**3) reservi:**

es un juego entre dos personas, que comparten 64 fichas iguales, de caras distintas, que se van colocando por turnos en un tablero dividido en 64 escaques. Las caras de las fichas se distinguen por su color y cada jugador tiene asignado uno de esos colores, ganando quien tenga más fichas sobre el tablero al finalizar la partida. Se clasifica como juego de tablero, abstracto y territorial; al igual que el go y las amazonas.

**4) Partida:**

donde dos jugadores se encuentran para poder enfrentarse.

**5) Guardar:**

Conservar una cosa o retenerla, evitando que desaparezca, se pierda o se altere.

**6) AJAX:**

Proviene del inglés Asynchronous JavaScript And XML (JavaScript y XML con ejecución asíncrona). Al fin y al cabo AJAX es una forma de desarrollo o programación de aplicaciones web que combina: - Una presentación basada en los estándares XHTML y CSS - Utiliza el DOM para interactuar con la página web - Para el intercambio de datos se basa en XML y XSLT - Solicita datos a un servidor de forma asíncrona utilizando XMLHttpRequest - Junta todo y lo presenta al usuario utilizando JavaScript

**7) Ancla (anchor)**

Es una abreviatura para referirse a los enlaces internos de una página web. Un ancla es un enlace que no nos cambia de página sino que enlaza a una zona del mismo documento de una página web.

**8) API**

Del inglés Application Programming Interface. Interfaz de Programación de Aplicaciones. Una serie de rutinas usadas por una aplicación para gestionar

generalmente servicios de bajo nivel, realizados por el sistema operativo de la computadora.

**9) ASP**

Active Server Pages. Páginas activas de servidor creadas mediante una tecnología de páginas dinámicas que permite alterar su contenido o incluso su forma, sin necesidad de editar o rescribir su código. Se trata de páginas con etiquetas de lenguaje HTML normales a las que se añaden una serie de programas.

**10) Atributo**

Es una parte de un elemento que añade información adicional a ese elemento.

**11) Avatar**

Personalidad o identidad visual que se atribuyen algunos usuarios de Internet, ya sea en algún chat, juegos, etc. Un avatar es un facsímil gráfico que se puede utilizar en las habitaciones de discusión virtual o chat rooms y que permite jugar un rol e interactuar con gente en línea. El término se hizo popular en la novela Snow Crash de Neal Stephenson.

**12) Base de datos**

Datos almacenados en una computadora que pueden ser accedidos y manipulados fácilmente; piense en un gran gabinete físico con muchas etiquetas con archivos de información relevante.

**13) Browser (navegador web)**

Browser, web browser o navegador es un programa de ordenador utilizado para ver páginas web e interactuar con ellas. Ejemplos de navegadores son: Firefox, Google Chrome, Opera, Konqueror, Safari, Midori, Epiphany, Amaya, Links, Internet explorer, etc.

**14) Chrome**

Navegador creado por Google. Según sus creadores, empezaron desde cero sin seguir patrones, e hicieron un navegador mejor adaptado a tecnologías más recientes para aplicaciones web.

**15) Class**

Selector CSS que nos permite especificar propiedades de estilo para un grupo de elementos. En HTML se representa como class, en CSS se representa con un símbolo de punto ".".

**16) Cliente**

Ordenador que requiere los servicios de otro ordenador. También, programa que requiere los servicios de otros programas. Normalmente, el navegador es un cliente de un servidor de datos.

**17) Cookie**

Es un fragmento de información que se almacena en el disco duro del visitante de una página web a través de su navegador, a petición del servidor de la página. Esta información puede ser luego recuperada por el servidor en posteriores visitas.

**18) CSS**

Las hojas de estilo en cascada (Cascading Style Sheets, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

**19) Desarrollador web (programador web)**

Un desarrollador web es un creador de páginas web que enfoca su trabajo en el desarrollo interno del funcionamiento de una página web.

**20) Diseño**

El diseño web es una actividad que consiste en la planificación, diseño e implementación de sitios web y páginas web. No es simplemente una aplicación del diseño convencional, ya que requiere tener en cuenta cuestiones tales como navegabilidad, interactividad, usabilidad, arquitectura de la información y la interacción de medios como el audio, texto, imagen y vídeo. Se lo considera dentro del diseño multimedial.

**21) DOM**

Document Object Model o DOM es el API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

**22) Debug**

Los programadores también hacen errores, así que después de hacer un programa, el programador debe probarlo, para ver si encuentra "bugs" en el sistema, por eso el término "debug".

### **23)Esquema**

Más conocido en inglés por XML Schema, es un documento XML que describe la especificación esperada entre aplicaciones para otros documentos XML. Es una alternativa más avanzada al uso de un DTD.

### **24)float**

Propiedad CSS muy utilizada para alinear un elemento a la izquierda o a la derecha "flotándolo", es decir, de tal manera que el resto de los elementos continúe con un flujo normal a menos que floten de manera similar o se especifique lo contrario en el documento CSS.

### **25)<html>**

El elemento <html> es el primero de cualquier documento HTML. Envuelve el resto del conjunto. Si origina una página Web normal, el elemento <html> contiene dos ingredientes esenciales más: el elemento <head>, que define el título, los metadatos y las hojas de estilo vinculadas, y el elemento <body>, que alberga el contenido real.

### **26)HTTP**

Hyper Transfer Protocol. Protocolo base de la Web.y que ofrece un conjunto de instrucciones para que los servidores y navegadores funcionen. Es el lenguaje usado para escribir documentos para servidores World Wide Web. Es una aplicación de la ISO Standard 8879:1986.

### **27)Herencia**

La herencia en CSS o hojas de estilo en cascada, es la forma en que los elementos que no tienen estilos predefinidos toman los estilos de sus padres en el árbol del documento.

### **28)HTTPS**

Protocolo de Transferencia de Hipertexto sobre SSL (Secure Socket Layer). Es un protocolo TCP/IP utilizado por los servidores Web para transferir y mostrar contenido Web de forma segura. Los datos transmitidos son encriptados de manera que no se pueden leer por nadie a excepción del receptor. HTTPS es usado por cualquier sitio Web que este recopilando información sensible de sus clientes como información bancaria o de compra. Si estas haciendo una transacción online, deberías asegurarte de que esta se lleva a cabo sobre HTTPS para que los datos permanezcan seguros y se transmitan de forma segura.

### **29)<input>**

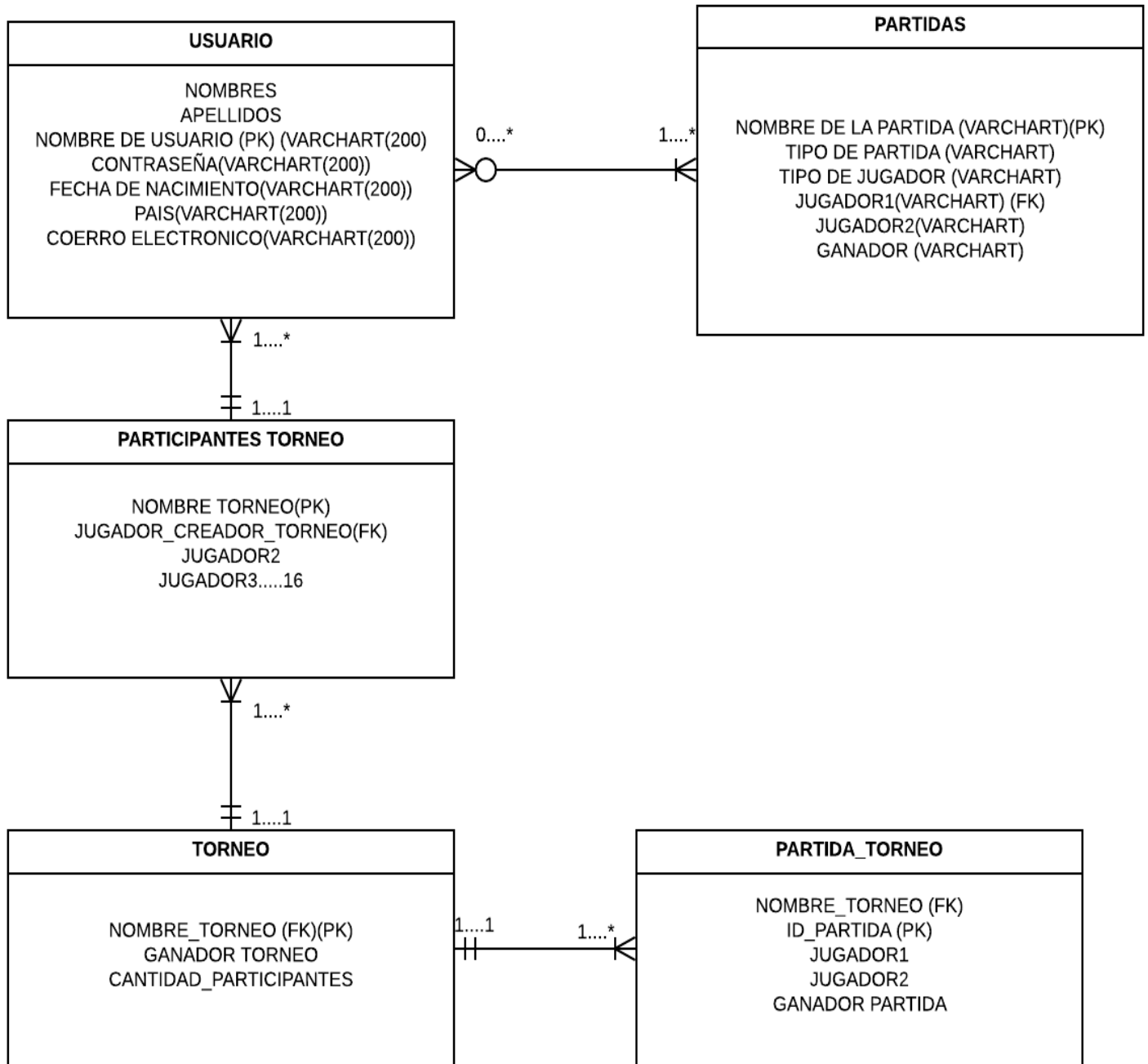
El elemento <input> es el ingrediente más común de un formulario HTML (representado por el elemento <form>). Puede representar distintos artilugios en pantalla (llamados controles) que recogen la información de un visitante Web.

### **30) Servidor**

Es el computador en el que se ejecutan los programas que realizan alguna tarea en beneficio de otras computadoras. Algunos servicios habituales son los servicios de archivos, que permiten a los usuarios almacenar y acceder a los archivos de un sitio web, así como ejecutar sus aplicaciones asociadas, en beneficio directo del usuario final.



# MODELO RELACIONAL



# DIAGRAMA DE CASOS DE USO

## DIAGRAMA DE USO GENERAL

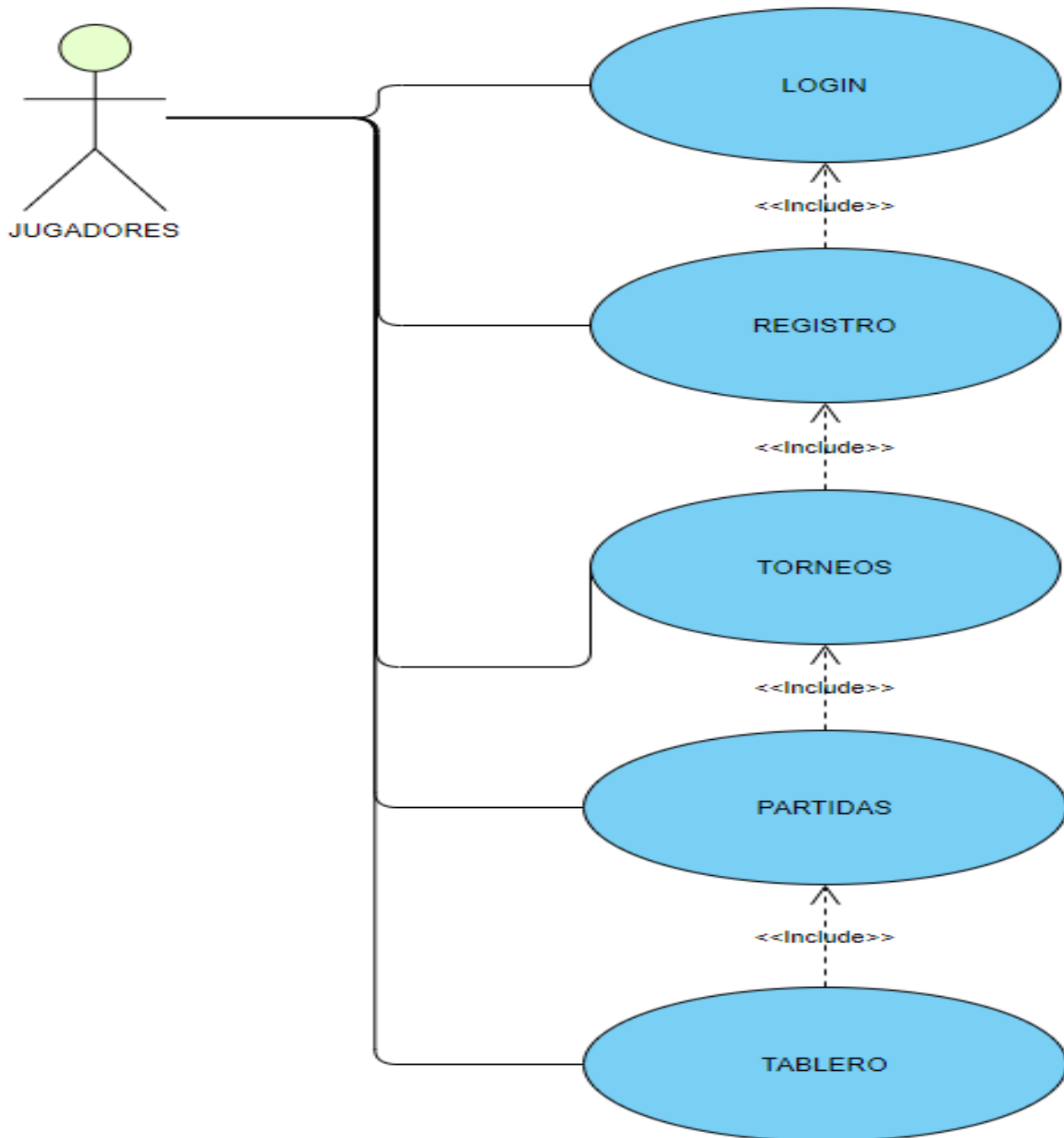


DIAGRAMA DE USO LOGIN NO.2

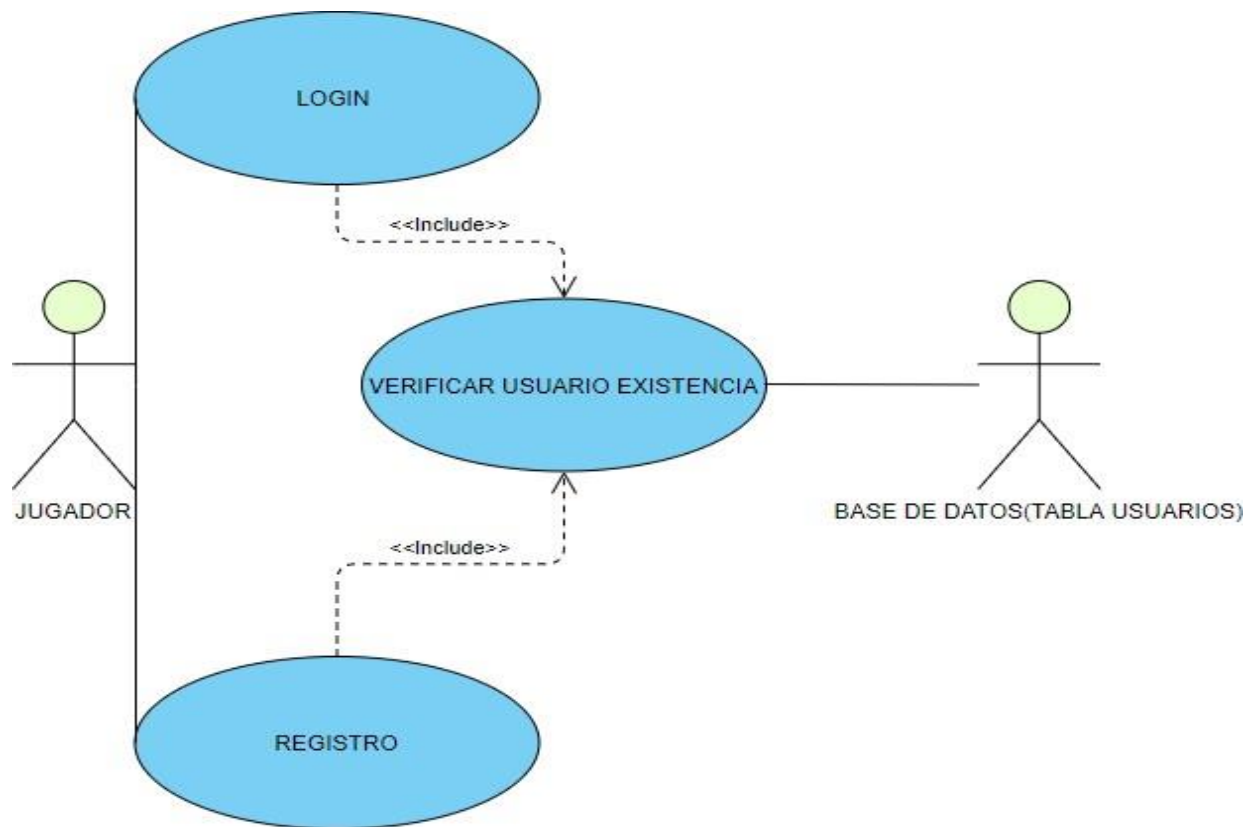
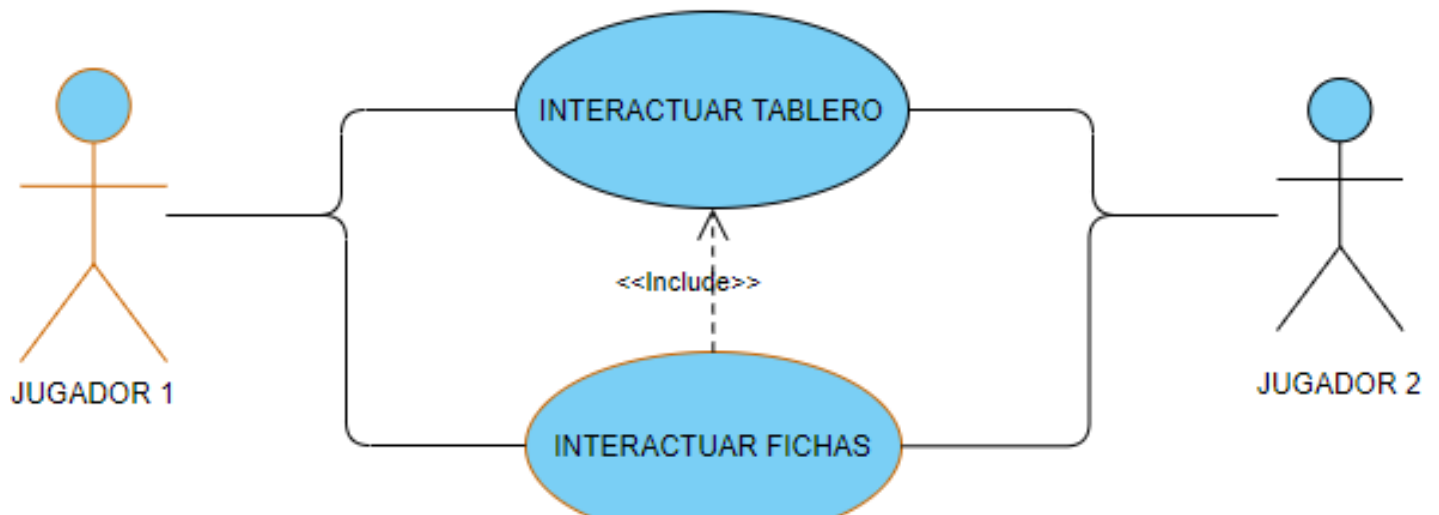


DIAGRAMA DE USO (TABLERO) NO. 3



### CASO DE USO ALTO NIVEL 1

CASO DE USO	Funcionalidad de la aplicación
ACTORES	Jugador
RESUMEN	El jugador ingresa a la url del juego donde le saldrá el login si no tiene alguna cuenta abajo hay un botón de registrarse donde se podrá registrar y entrar al juego othello donde podrá colocar fichas dentro de un tablero.
TIPO	primario

### CASO DE USO EXPANDIDO 1

CASO DE USO	Funcionalidad de la aplicacion
ACTORES	Jugador
PROPOSITO	Crear un usuario , crear una partida y jugar othello
RESUMEN	se creara un usuario para poder ingresar a la partida y poder jugar con otro contrincante a othello.
TIPO	Principal y esencial
<b>CURSO NORMAL DE LOS EVENTOS</b>	
El jugador ingresa a la pagina	
El usuario presionara el botón registro y ira a la página de registro	
El usuario colocara sus datos	
Luego se verificara si el usuario no está creado	
El usuario no está creado satisfactorio	
Se re direcciona a el juego othello	
se mostrara el tablero de juego	
Empezara la partida	
Los usuario pueden colocar las fichas	
<b>CURSOS ALTERNOS</b>	
Si la dirección url está mal no ingresara al juego	
Si el usuario no coloca los datos saldrá un error y no se guardaran	
Si el nombre de usuario esta creado no se guardaran los datos	
No se re direccionara al juego	

## CASO DE USO ALTO NIVEL 2

CASO DE USO	Registro
ACTORES	Jugador, base_de_datos
RESUMEN	Crear un usuario en la base de datos llenado los campos solicitados para luego tener un usuario y colocarlo en la página de login y poder ingresar a el juego
TIPO	primario

## CASO DE USO EXPANDIDO 2

CASO DE USO	Registro
ACTORES	Jugador, base_de_datos
PROPOSITO	Crear un usuario y guárdalo en una base de datos
RESUMEN	Guardar registros en una base de datos donde después esos datos se puedan consultar para la página login y poder ingresar al juego de una forma adecuada
TIPO	Primario y esencial
<b>CURSO NORMAL DE LOS EVENTOS</b>	
El jugador interactuara con la base de datos si tiene algún usuario y contraseña ya creada	
El jugador nuevo se ira al botón registrarse	
El jugador colocara sus datos en los input text	
El jugador le dará al botón registrase ese botón le pedirá a la base de datos que guarde ese registro	
Luego se re direcciona al login	
El jugador tiene que colocar el usuario y contraseña en el login para poder entrar al juego	
<b>CURSOS ALTERNOS</b>	
Si el usuario y contraseña no existe en el login se recibirá un mensaje de error que no existe el usuario ni la contraseña	
si en la página registro los input text no están llenos se marcara como un error	
Al dar un error la página la base de datos no guardara los datos ingresados en los input text	
Se re direccionara al login	
Si no coloca el usuario y el login correctamente marcara un error de los datos no concuerdan	

### CASO DE USO ALTO NIVEL 3

CASO DE USO	Tablero
ACTORES	Jugador 1, jugador 2
RESUMEN	Jugador 1 y jugador 2 podrán interactuar con el tablero y colocar las fichas donde la cantidad de fichas son 64 donde el tablero tiene 64 posiciones donde poder colocarlas.
TIPO	Primario y esencial

### CASO DE USO EXPANDIDO 3

CASO DE USO	tablero
ACTORES	Jugador 1, jugador 2
PROPOSITO	Interactuar con el tablero y poder colocar fichas en el tablero
RESUMEN	Poder colocar fichas blancas y negras en el tablero para poder interactuar con el y poder descargar el archivo con las posiciones de las fichas
TIPO	Primario y esencial

### CURSO NORMAL DE LOS EVENTOS

Los jugadores podrán interactuar con el tablero
El jugador 1 y el jugador 2 podrán colocar las fichas
Podrán descargar las condenas de donde colocaron las fichas al final del juego o en el momento que quieran
Podrán cargar de nuevo la partida donde la dejaron pendiente
Podrán salir del juego si lo desean con el botón salir del juego
El xml se guardara en la el directorio (escritorio) donde se podrá ver las posiciones de las fichas

### CURSO ALTERNO

Si los jugadores no han colocado ninguna ficha el documento xml estará vacío sin datos
Si el jugador no presiona el botón de descargar xml al cerrar el juego no se podrá cargar su partida de nuevo

### CASO DE USO ALTO NIVEL 4

CASO DE USO	LOGIN
ACTORES	Jugador
RESUMEN	Ingresa al juego si ya contiene un usuario por medio de los campos usuario y contraseña.
TIPO	Primario y esencial

#### CASO DE USO EXPANDIDO 4

CASO DE USO	LOGIN
ACTORES	Jugador
PROPOSITO	El usuario ingrese al juego
RESUMEN	El jugador podrá ingresar su usuario y su contraseña y podrá ingresar al juego
TIPO	Primario y esencial
CURSO NORMAL DE LOS EVENTOS	
El usuario ingresa a la pagina	
El usuario ingresa sus datos usuario y contraseña	
El usuario ingresara al juego si el usuario y la contraseña son iguales	
CURSO ALTERNO	
Si el usuario coloca mal la url no mostrara la pagina	
Si el usuario ingresa mal los datos le mostrara un error	
Si no son iguales los datos no lo redireccionara a la pagina principal	

#### CASO DE USO ALTO NIVEL 5

CASO DE USO	TORNEO
ACTORES	Jugadores
RESUMEN	Los jugadores se pueden enfrentar entre si, la máxima cantidad de jugadores que puede ingresar es de 16
TIPO	Primario y esencial

#### CASO DE USO ALTO NIVEL 5

CASO DE USO	TORNEO
ACTORES	Jugadores
PROPOSITO	Los jugadores interactúen entre si y tener un ganador en el torneo
RESUMEN	Tener los datos de todos los usuarios que juegan en el torneo
TIPO	Primario y esencial
CURSO NORMAL DE LOS EVENTOS	
El jugador crea un torneo	
El jugador envía el nombre del torneo a los otros jugadores	
Ya ingresados los 16 jugadores se iniciara el torneo	
Se inician las llaves los jugadores se enfrentaran aleatoriamente	
Conforme el torneo avance se eliminaran los jugadores y no tendrán acceso al torneo	
El terminar el torneo se mostrara al ganador	
CURSO ALTERNO	
Si el jugador pasa mal el nombre del torneo no le funcionara a los otros usuarios	
Si el torneo no se llena en una hora se eliminara el torneo automáticamente	
Si algún jugador se sale del juego el juego seguirá y el otro jugador gana automáticamente	

## CASO DE USO ALTO NIVEL 6

CASO DE USO	PARTIDAS
ACTORES	Jugador 1 y jugador 2
RESUMEN	Los dos jugadores interactúen entre si moviendo fichas dentro del tablero
TIPO	Primario y esencial

## CASO DE USO ALTO NIVEL 6

CASO DE USO	PARTIDAS
ACTORES	Jugador 1 y jugador 2
PROPOSITO	Los jugadores se puedan enfrentar uno a uno o puede que un jugador con la maquina
RESUMEN	Los dos jugadores ingresan al sistema para poder jugar entre ellos podrán colocar fichas
TIPO	Primario y esencial
<b>CURSO NORMAL DE LOS EVENTOS</b>	
Los dos jugadores ingresan a la para poder enfrentarse	
El primer jugador colocara una ficha es donde el juego comenzara	
El segundo jugador colocara la segunda ficha	
Si el usuario ya no tiene posibilidades el otro jugador ganara la partida	
<b>CURSO ALTERNO DE LOS EVENTOS</b>	
Si solo un jugador ingresa a la partida el juego no comenzara	
Si uno de los dos jugadores no coloca una ficha el juego no comenzara	
Si ninguno de los no puede ganar será un empate	
Si un jugador sale de la partida el otro jugador ganara automáticamente	



MODELO CONCEPTUAL

