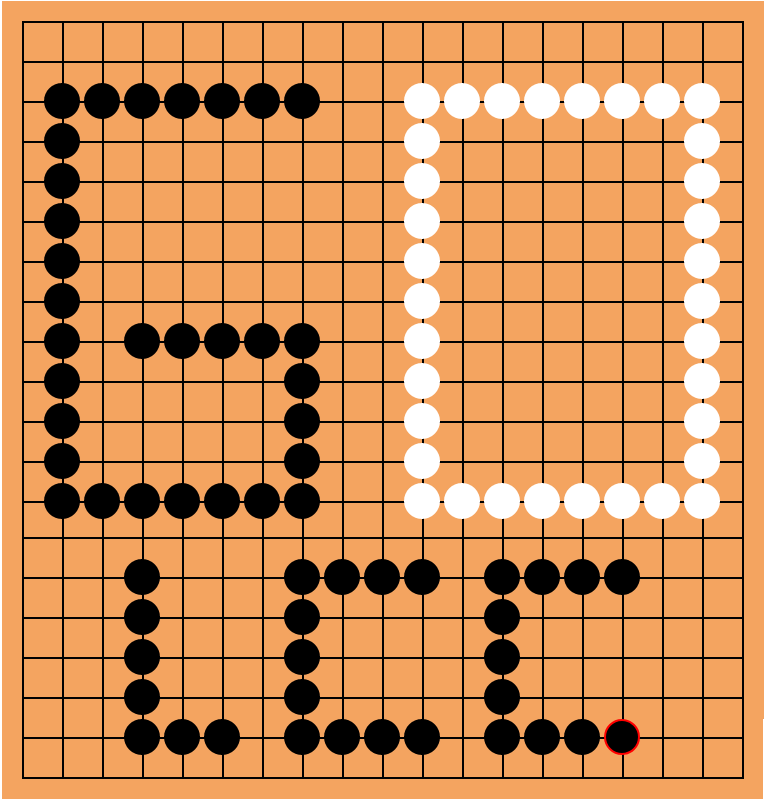
**Lógica para Ciencias de la Computación**

**Proyecto N°1**

****

**Autores:**

**Carlos Loza – LU: 94399**

**Javier Fernández Tierno – LU: 106243**

Índice

[Descripción general del software 2](#_Toc8983464)

[Introducción 2](#_Toc8983465)

[Inicio de la partida 2](#_Toc8983466)

[Captura 2](#_Toc8983467)

[Suicidio 2](#_Toc8983468)

[Pasar el turno 2](#_Toc8983469)

[Final de la partida 2](#_Toc8983470)

[Descripción de la implementación 3](#_Toc8983471)

[Documentación para el cliente 4](#_Toc8983472)

[Documentación para el desarrollador 7](#_Toc8983473)

[Implementación en Prolog 7](#_Toc8983474)

# Descripción general del software

## Introducción

El GO es un juego de tablero de estrategia para dos personas que consiste de un tablero de 19 líneas horizontales (filas) por 19 líneas verticales (columnas) y piezas llamadas piedras (fichas) de color blanco y negro. Se originó en China hace más de 2500 años. El objetivo del juego es controlar una cantidad de territorio mayor a la del oponente. Para controlar un área, debe rodearse con las piedras.​ Gana el jugador que controla la mayor cantidad de territorio al finalizar la partida. El juego consiste en colocar las piedras en las intersecciones del tablero. Las negras inician la partida y una vez colocada una piedra, no se puede volver a mover. Se puede capturar una piedra o un conjunto de piedras y eliminarlas del tablero si están completamente rodeadas por piedras de otro color.

## Inicio de la partida

Al inicio de la partida el tablero está vacío. Las negras juegan primero. Después ambos jugadores juegan alternadamente. Una jugada consiste en poner una piedra en una intersección vacía.​ En general, el jugador más fuerte juega con blancas, puesto que las negras tienen la ventaja de iniciar primero.

Captura

Cuando un jugador hace una jugada que priva de su última libertad a una piedra del oponente, debe sacar las piedras rodeadas del tablero y guardarlas separadamente hasta el final de la partida.​ Las piedras que solo poseen una libertad se dicen que están en atari.

## Suicidio

No se permite hacer una jugada ocupando una posición libre en el interior de una formación enemiga, a no ser que esta jugada capture piedras enemigas. Si la formación que uno quiere capturar está rodeada, entonces se puede jugar en su interior, si al hacerlo gana dicha formación.

## Pasar el turno

En lugar de poner una piedra, un jugador puede pasar, es decir pierde un turno. Pasar un turno casi nunca es buena idea. Solo es bueno pasar de turno al final de la partida, cuando los territorios ya están definidos y no hay jugadas por hacer. Cuando los dos jugadores pasan consecutivamente, se acaba la partida.

## Final de la partida

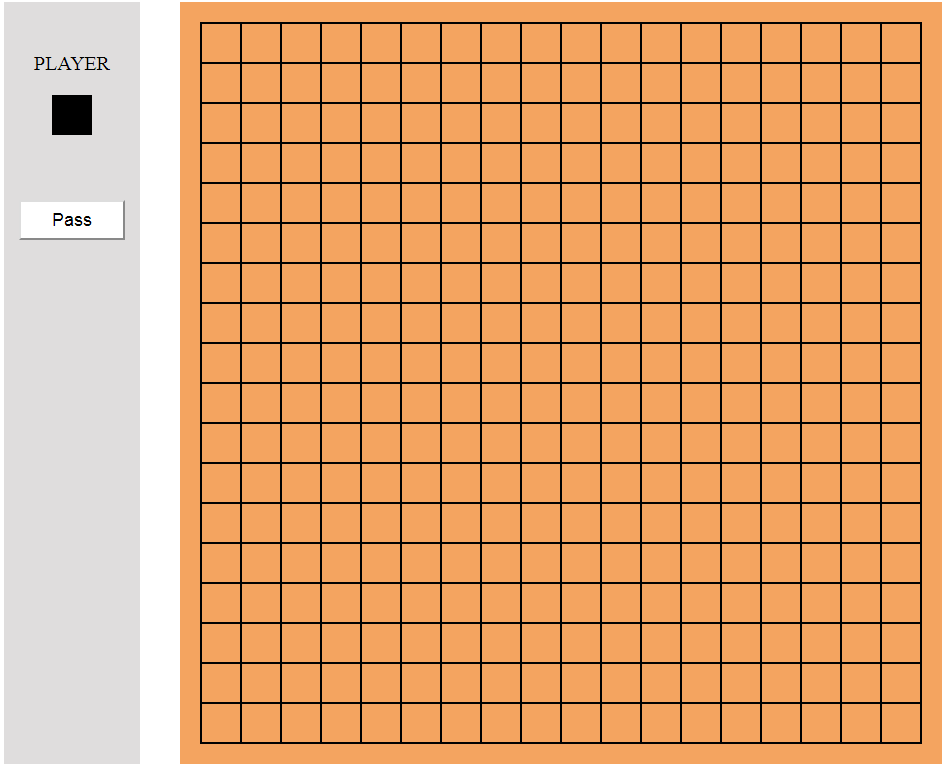
Una vez finalizada la partida, se retiran las piedras muertas de cada bando añadiéndolas a las capturadas. Se conocen como piedras muertas a aquellas que se encuentran rodeadas y en grupos con mala forma. Entonces, los jugadores cuentan los puntos de cada territorio, teniendo en cuenta los espacios encerrados por las fichas de cada jugador y se decide quién ganó la partida.​

# Descripción de la implementación

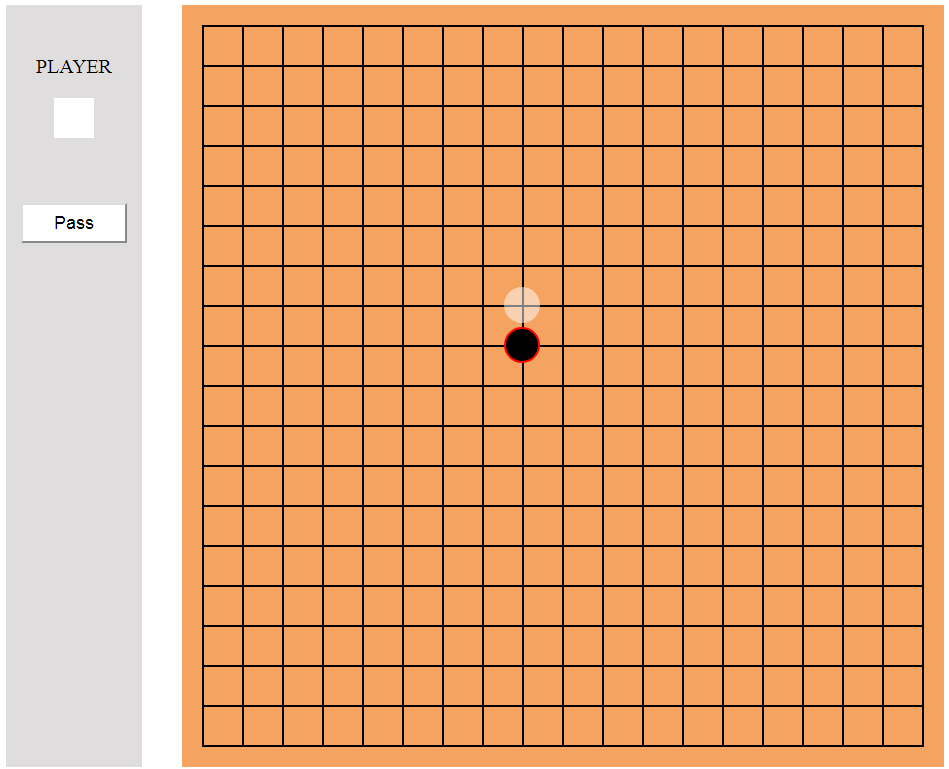
Se utilizó el lenguaje de programación PROLOG para llevar a cabo la implementación lógica del videojuego. También se utilizaron tres lenguajes básicos de desarrollo web: HTML, CSS y JavaScript.

# Documentación para el cliente

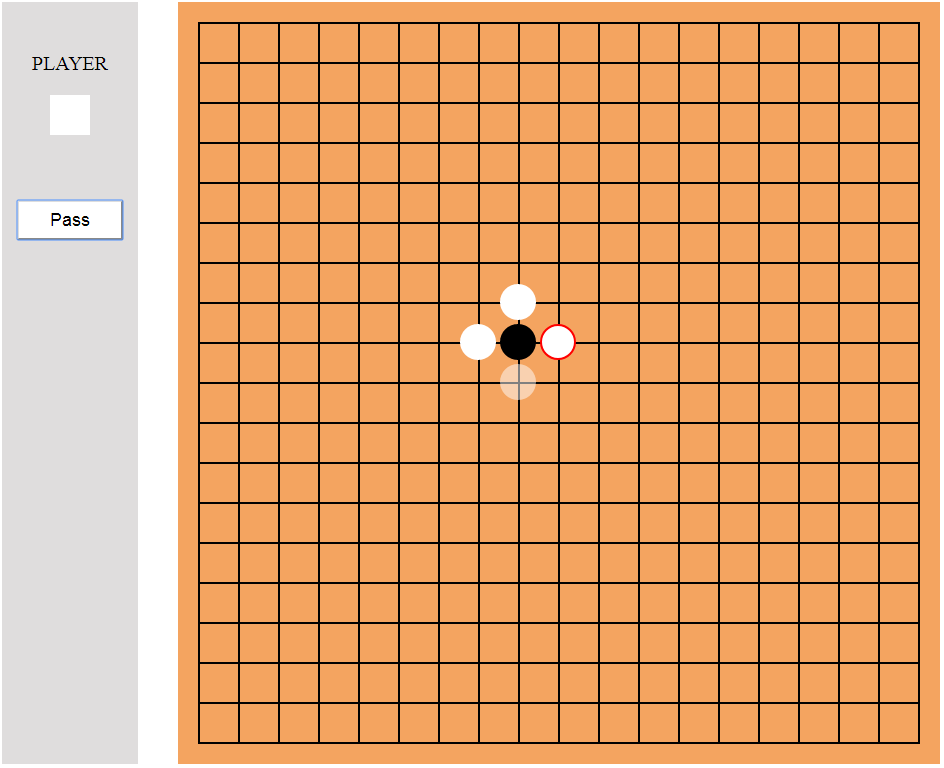
El juego consta de una grilla de 19x19 vacía al inicio de la partida. También cuenta con un botón “Pass” que permite pasar de turno. Por último, siempre se empieza colocando una ficha negra.

****

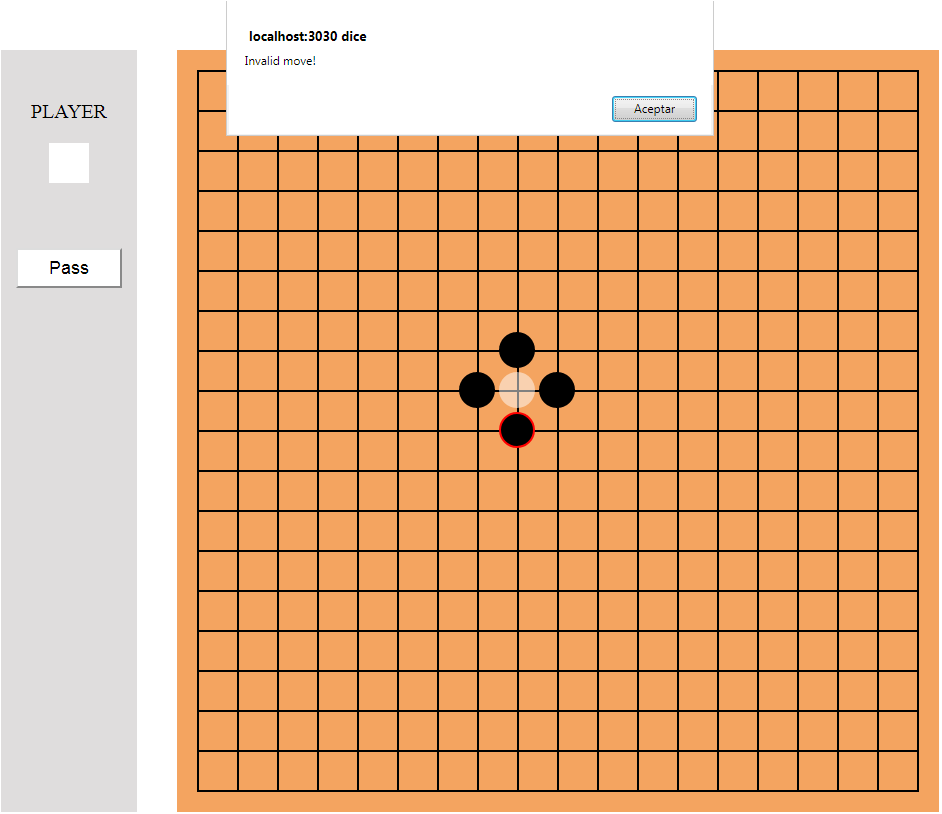
Cuando se coloca la primera ficha negra, es el turno del jugador con fichas blancas colocar una ficha en el tablero.



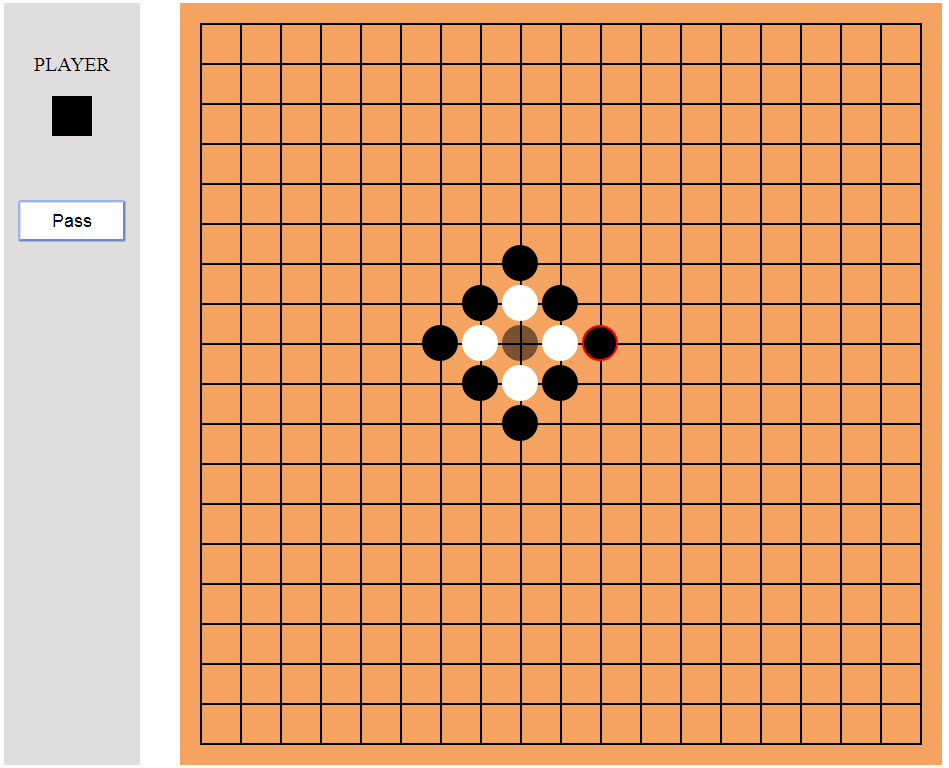
A continuación, se presenta una situación llamada “atari” que consta de una ficha negra siendo encerrada por fichas blancas, cuando el jugador coloque una ficha blanca en la parte inferior de la ficha negra, esta quedara eliminada y retirada del juego.



La siguiente situación es considerada suicidio, ya que el jugador con fichas blanca intenta colocar una ficha entre medio de fichas negras, quedando así encerrada automáticamente al colocarse. Esto es ilegal para el videojuego y no se permite colocar fichas en estos casos.



Sin embargo, la siguiente jugada no es considerada suicidio, ya que el jugador con fichas negras coloque una ficha en el centro de esta formación, quedaran eliminadas todas las fichas blancas, ya que cada ficha blanca quedo encerrada por 4 fichas negras compartidas. Las mismas son eliminadas y retiradas de la partida.



# Documentación para el desarrollador

A continuación se describen detalles acerca de funciones implementadas, relaciones entre funciones y relaciones entre interfaces.

## Implementación en Prolog

Predicados que provee la cátedra:

* **emptyBoard(Board):** Provee un tablero formado por una lista que contiene listas como elementos.
* **goMove(Board, Color, [Fila,Col], RRBoard):** Dado un tablero, el color de una ficha y una posición, se retorna el tablero con una nueva ficha colocada en la posición especificada.
* **replace(X, XIndex, Y, [Xi|Xs], [Xi|XsY]):** Este predicado funciona de la siguiente manera, primero se obtiene la nueva fila del tablero y luego se inserta la nueva fila en el tablero, modificándolo con la nueva ficha.

Predicados creados por los autores:

* **suicidio(Board, Fila, Col, Color):**
* **reemplazarBoard(Ant ,Board, F, C, Color, RBoard):**
* **limpiarAlrededor(Board, Fila, Col, Color, RBoard):**
* **invertirColor(Color, ColorI):**
* **cascaraLimpiarEncerrado(Board, Fila, Col, Color, Board):**
* **encerrado(Board, Fila, Col, \_Color, Board):**
* **limpiar(Board, Fila, Col, Board):**
* **contarFichas(Board, CantBlancas, CantNegras):**
* **contarCol([F|Bs], Color, Rta):**
* **contarFila([Color|Ls], Color, Rta):**