# Portada

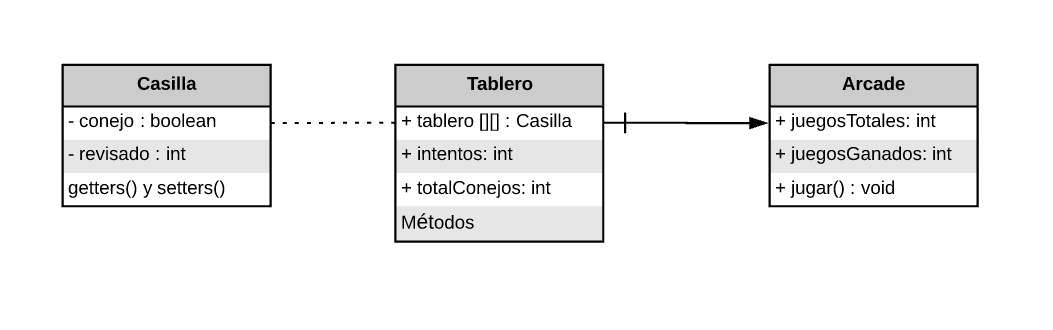
### Indice:

1. Portada
2. Indice
3. Introduccion
4. Desarrollo
5. Conclusiones
6. Resultados
7. Referencias Bibliograficas

## Introducccion

## Desarrollo

Para poder desarollar el Videojuego se hizo un analisis en el cual la mejor manera fue la del siguiente diagrama UML:



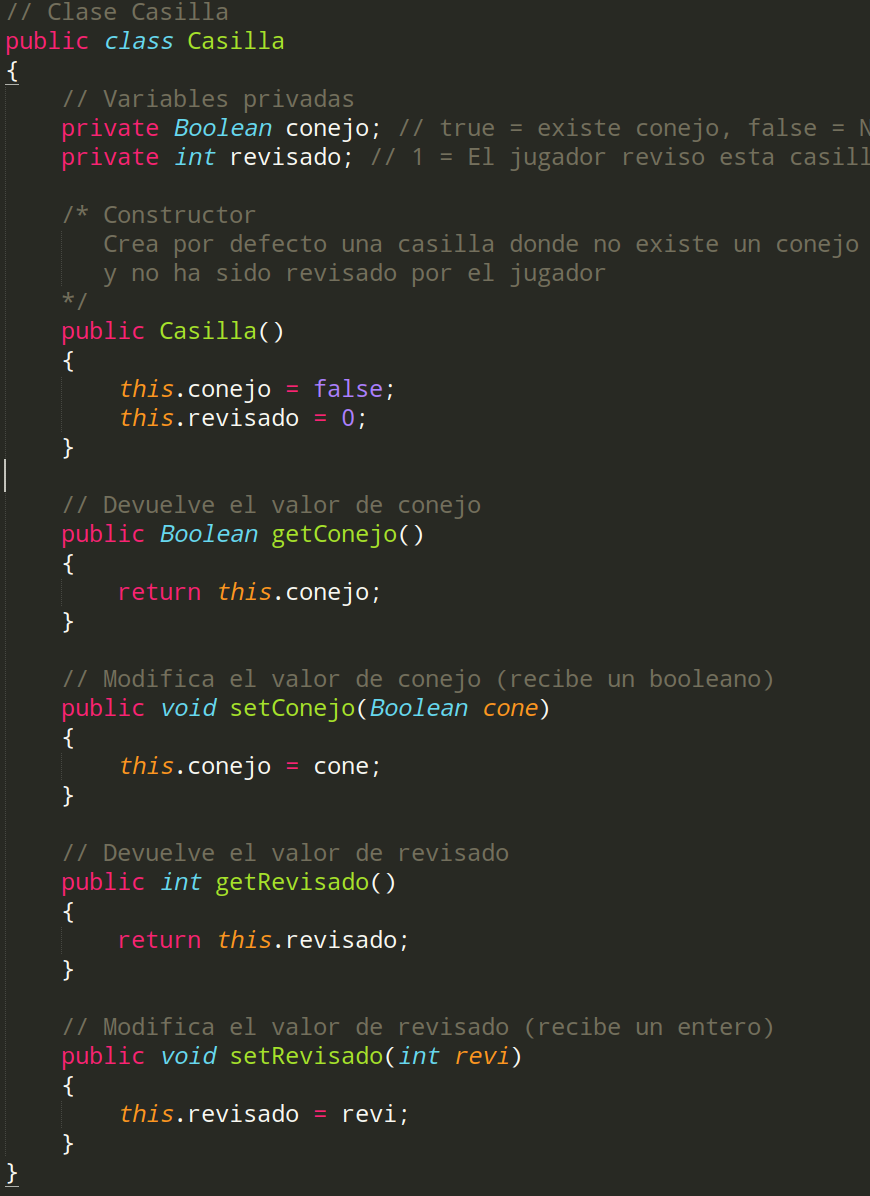
Donde:

* Casilla: Clase que nos ayudara a saber si hay conejo o no en dicha casilla, a su vez de saber si ya se reviso o no dicha casilla.
* Tablero: Clase que definira un Tablero de objeto casillas, las cuales pueden tener o no conejos. Aqui tambien definiremos la logica del Videojuego.
* Arcade: Clase que hereda de Tablera la cual nos permitira manejar las estaditicas y todas las partidas que desee jugar el usuario.

**Clase Casilla**

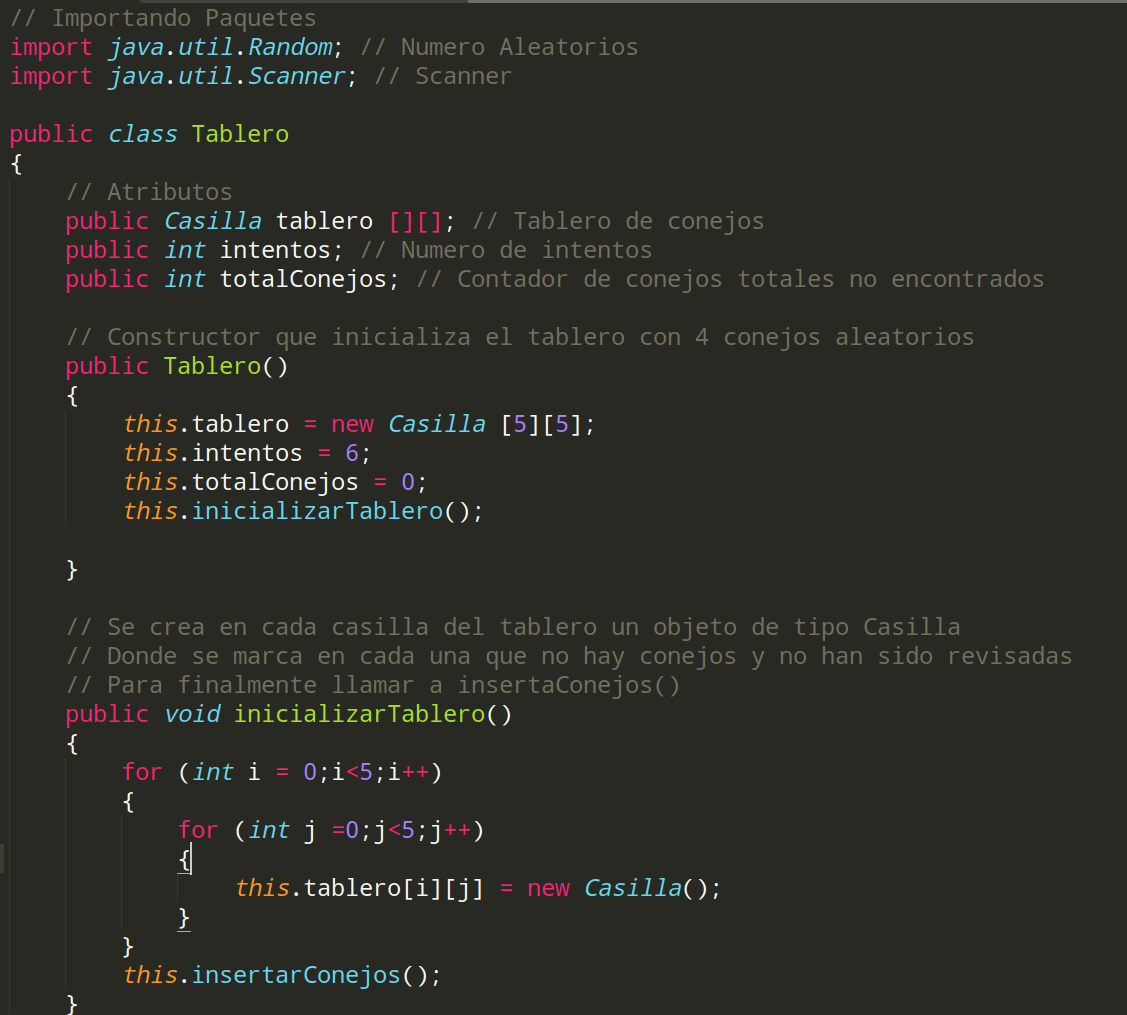
La clase mas sencilla de las tres, aqui solo se define que tiene como atributos un boolean llamado conejo, el cual si es false, no existira conejo en dicha casilla y existira si su valor es true, mientras que atributo es un int que valdra 1 si ha sido revisada la casilla o 0 en caso contrario.

Codigo Fuente



**Clase Tablero**

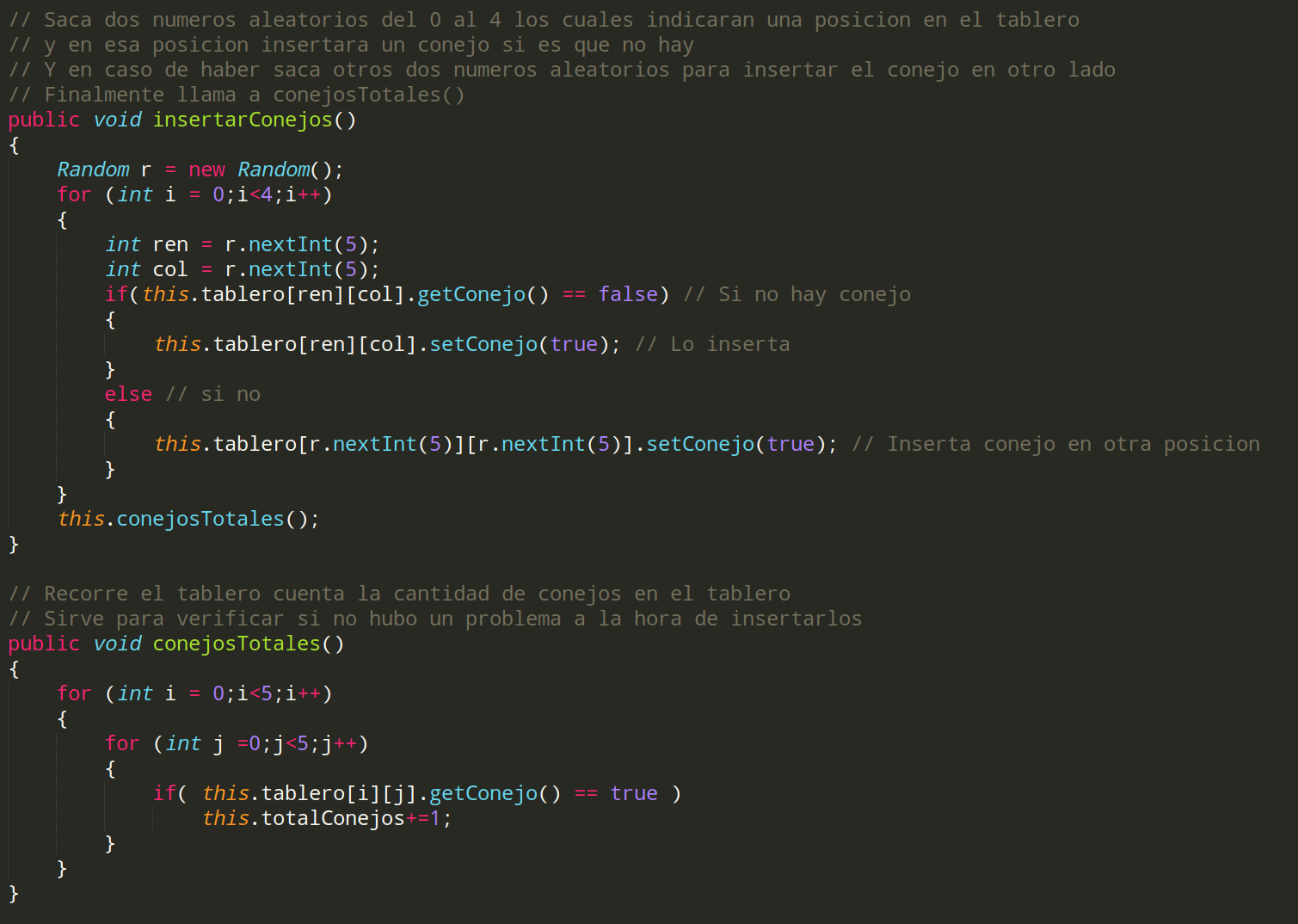
La clase que definio la logica del juego, esta se explicara por partes.



Se define los contadores intentos y totalConejos, los cuales respectivamente nos marcaran el limite de intentos en cada partida y la cantidad de conejos a encontrar. Tambien se define una matriz de Casillas de 5x5 en la cual se le inidicara posteriormente donde hay conejos ocultos y donde no.

**Constructor():** Instancia el tablero e inicializa variables y el tablero.

**inicializarTablero() :** Se recorre el tablero para que en cada posicion se cree una casilla que no se haya revisado y no tenga conejo. Para finalmente llamar al metodo insertarConejos.



**insertarConejos():** Es el encargado de ingresar a 4 casillas aleatorias un conejo, solamente cambiando el valor de conejo a true de las cuatro casillas. Para luego llamar al metodo conejosTotales.

**conejosTotales():** Recorre todo el tablero (matriz) para contar cuantos conejos hay en el y poder asignar el numero a totalConejos.



**imprimirTablero():** Con un formato donde muestra numero de intentos y conejos a encontrar, imprime el tablero en pantalla y en cada casilla mostrara algo diferente segun se haya hecho con ella:

* ? -> En caso de que haya o no conejo y no se haya revisado la casilla.
* \* -> En caso de que no haya conejo y se haya revisado la casilla (No encontrado un conejo).
* X -> En caso de que haya conejo y se haya revisado la casilla (Encontrado un conejo)

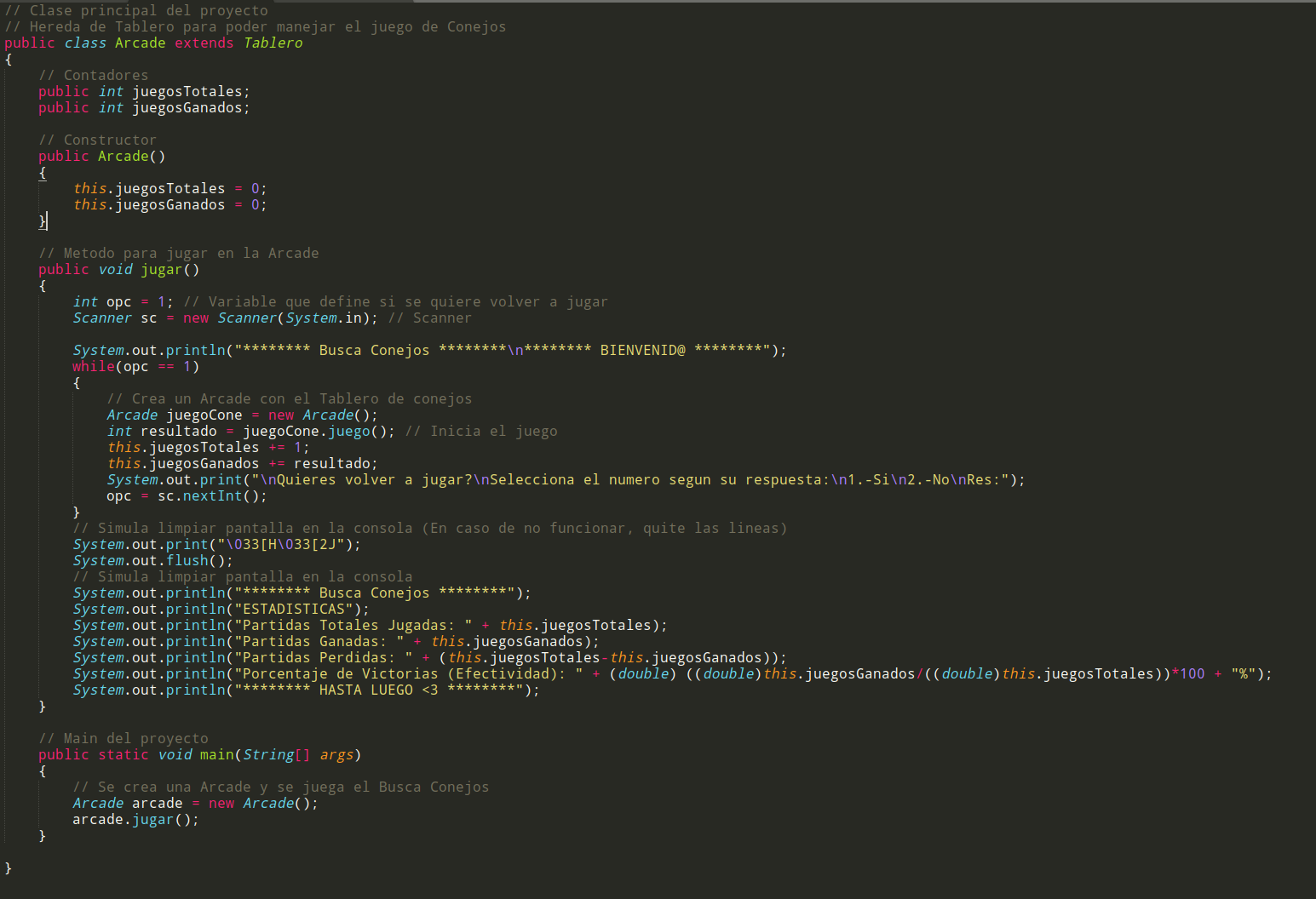
**respuestaTablero():** Imprime el tablero con la ubicacion de los conejos;

* ? -> En caso de que no haya conejo.
* X -> En caso de que haya conejo.



**juego():** Su funcion es que mientras el usuario no gaste sus intentos o encuentre todos los conejos, le muestre el tablero con su situacion actual dentro del juego, le pida que inidique en que casilla revisar, verifica que la casilla sea valida, para luego ir a la casilla y si hay conejo decremente uno los conejos a buscar (totalConejos) y si no decrementa uno sus intentos. En caso de ganar te lo notifica y devuelve un 1 y en caso de perder devuleve un 0.

**Clase Arcade**

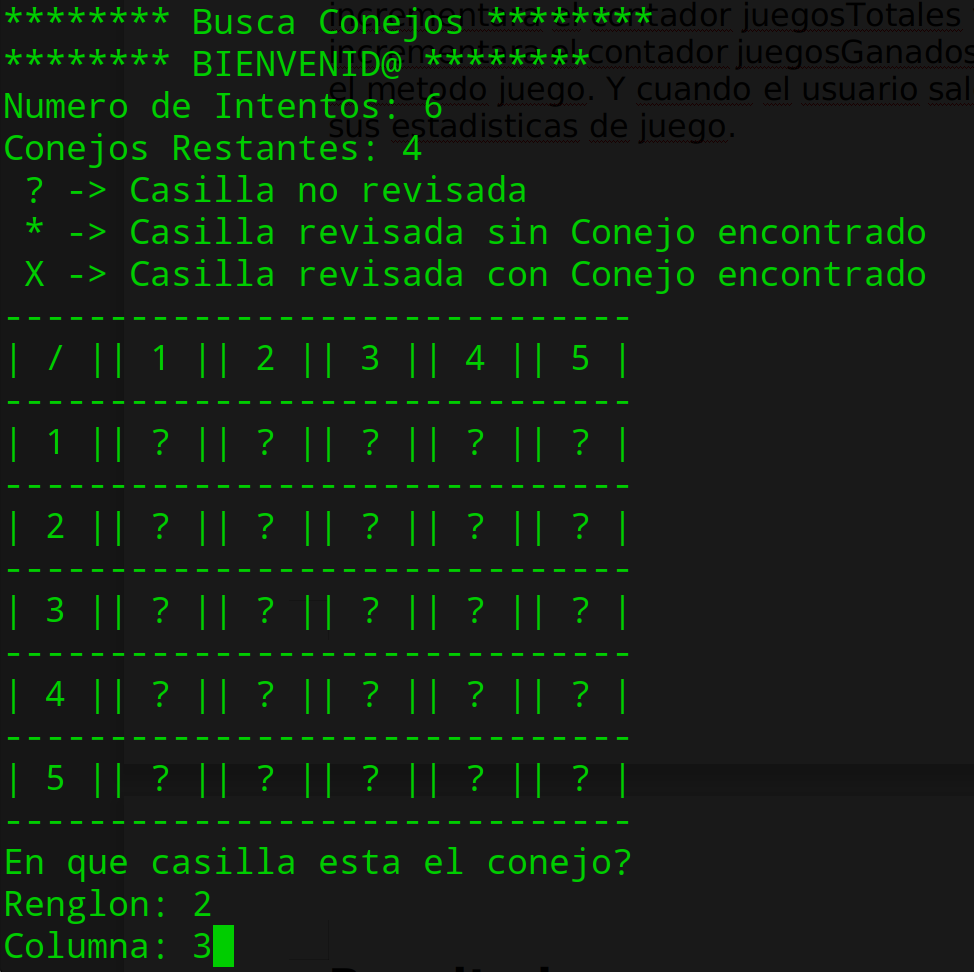


Hereda todo lo de la clase Tablero, tiene dos contadores juegosTotales y juegosGanados que definiran los juegos totales y ganados que juege el usuario respectivamente.

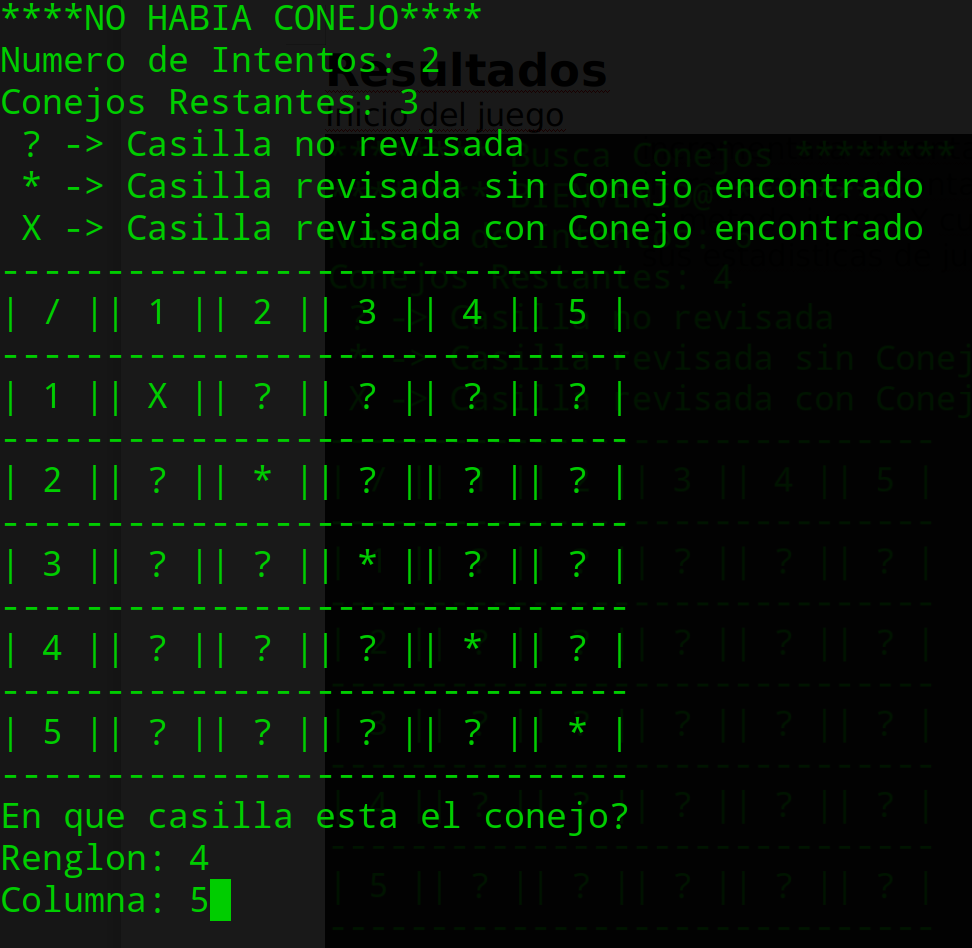
jugar(): Mientras el desee podra jugara todar las partidas que quiera y distintas ya que en cada partida se vuelve a crear un objeto de tipo Arcade, donde cada partida incrementara el contador juegosTotales y si se gana incrementara el contador juegosGanados con lo que mande el metodo juego. Y cuando el usuario salga se le muestre sus estadisticas de juego.

**Resultados**

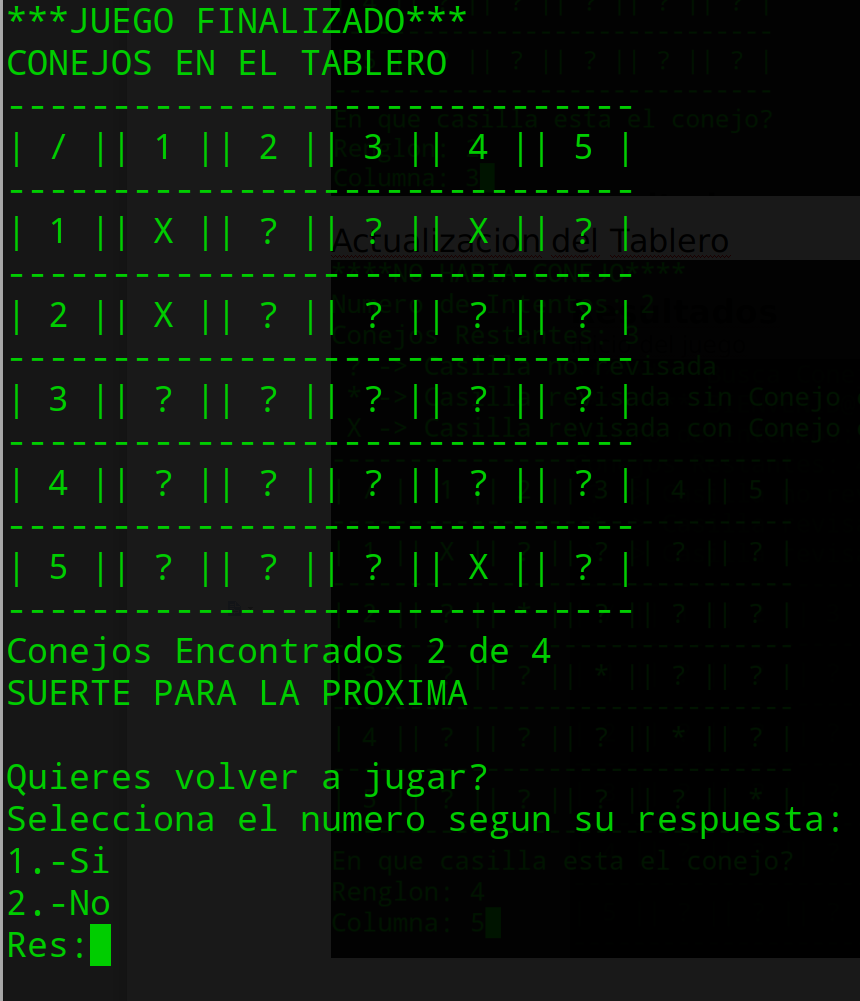
Inicio del juego



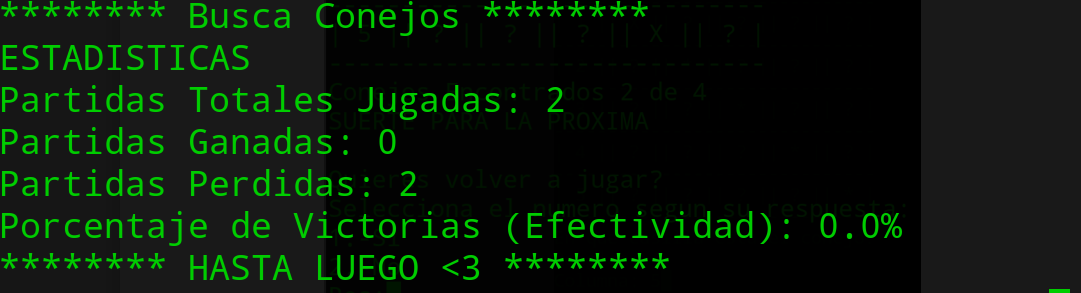
Actualizacion del Tablero



Fin de una Partida



Fin del Juego



**Conclusiones**

**Referencias Bibliograficas**