



UNIVERSIDAD DEL BÍO-BÍO

Análisis Juego Del Gato Con Sockets

Integrantes:

Deivid Sandoval

Dilan Toro

Índice

Introducción	3
Propósito del sistema	4
Descripción general	5
Metas del sistema	6
Características del usuario	7
Restricciones generales.....	8
Requisitos específicos	9
Atributos generales	10
Atributos relacionados con cada función	10
Caso de uso	11
Diagrama de secuencia	12
Diagrama de clases.....	13
Diagrama de colaboración	14

Introducción

En el mundo de la programación, la creación de juegos ha sido una práctica esencial para comprender los conceptos fundamentales y potenciar habilidades. En este informe, exploraremos el universo del juego del gato, también conocido como Tic Tac Toe, implementado en el lenguaje de programación Java haciendo uso de sockets. Este proyecto no solo nos permite sumergirnos en las complejidades de la programación de juegos, sino que también destaca la importancia de la comunicación entre diferentes componentes a través de sockets.

Propósito del sistema

El propósito fundamental de implementar el juego del gato en Java utilizando sockets es explorar la comunicación entre diferentes entidades a través de una red. En este proyecto, los sockets actúan como canales de comunicación que permiten a dos jugadores interactuar en tiempo real, aunque estén en ubicaciones diferentes. La elección del juego del gato no solo se debe a su simplicidad, sino también a la riqueza de interacciones que presenta.

Descripción general

Cada jugador ejecuta una instancia del programa, y a través de la red, se establece una conexión mediante sockets para facilitar la interacción.

El juego sigue las reglas tradicionales del tic-tac-toe, donde dos jugadores alternan turnos para colocar sus fichas en un tablero de 3x3. La innovación radica en la capacidad de los jugadores para realizar sus movimientos de forma remota, lo que implica una comunicación constante entre las instancias del juego en ambos extremos de la conexión.

Además, el programa gestiona la lógica del juego, asegurándose de que se respeten las reglas y determinando cuándo un jugador ha ganado, perdido o si el juego termina en empate. Esta implementación ofrece una introducción práctica y educativa a los conceptos de programación de juegos y comunicación a través de sockets en Java.

Metas del sistema

1. Implementación del Juego Funcional: Desarrollar un juego del gato completamente funcional en Java que siga las reglas tradicionales del Tic Tac Toe.
2. Comunicación a Través de Sockets: Establecer una comunicación bidireccional efectiva entre las instancias del juego en dos dispositivos diferentes utilizando sockets, permitiendo que los jugadores interactúen en tiempo real.
3. Sincronización de Movimientos: Garantizar la sincronización adecuada de los movimientos entre los jugadores para mantener la integridad del juego, evitando conflictos y asegurando que ambos jugadores tengan una experiencia coherente.
4. Detección de Ganadores o Empates: Implementar la lógica necesaria para determinar cuándo un jugador ha ganado, cuándo hay empate y actualizar la interfaz de usuario en consecuencia.
5. Interfaz de Usuario Intuitiva: Crear una interfaz de usuario clara e intuitiva que permita a los jugadores realizar sus movimientos fácilmente y recibir retroalimentación instantánea sobre el estado del juego.
6. Manejo de Errores y Excepciones: Implementar mecanismos para manejar posibles errores y excepciones que puedan surgir durante la ejecución del programa, proporcionando mensajes informativos para una experiencia de usuario mejorada.
7. Documentación Clara y Concisa: Proporcionar documentación detallada del código fuente, explicando la estructura del programa, el propósito de cada componente y cómo se logra la comunicación a través de sockets.
8. Pruebas Exhaustivas: Realizar pruebas exhaustivas del programa para asegurar su estabilidad, identificar y corregir posibles errores, y garantizar un rendimiento óptimo incluso en situaciones de uso intensivo.
9. Facilidad de Uso y Despliegue: Diseñar el programa de manera que su ejecución y despliegue sean sencillos para que cualquier persona interesada pueda disfrutar del juego sin complicaciones.

Características del usuario

Jugador 1:

Inicia la conexión y escucha en un puerto.

Controla el flujo del juego.

Envía información actualizada al jugador cliente.

Jugador 2:

Se conecta al servidor mediante IP y puerto.

Realiza movimientos en el juego.

Recibe información actualizada del jugador servidor.

Restricciones generales

Conectividad de Red: El programa depende de la conectividad de red entre los dispositivos del jugador servidor y el jugador cliente. La falta de una conexión estable podría afectar negativamente la experiencia de juego.

Puertos Disponibles: El servidor debe tener acceso a un puerto disponible para escuchar conexiones entrantes. Restricciones en la disponibilidad de puertos podrían interferir con el funcionamiento correcto del programa.

Compatibilidad de Versiones: La compatibilidad entre las versiones del programa en el servidor y el cliente es esencial. Cambios no compatibles podrían resultar en comportamientos inesperados o fallos en la comunicación.

Firewalls y Configuraciones de Red: Configuraciones de firewall u otras restricciones de red pueden interferir con la comunicación entre los jugadores. Es esencial garantizar que las configuraciones permitan el tráfico necesario para el juego.

Recursos del Sistema: El rendimiento del programa puede verse afectado por limitaciones en los recursos del sistema, como la capacidad de procesamiento y la memoria disponible, especialmente al manejar múltiples conexiones concurrentes.

Entrada del Usuario: La interacción del usuario, especialmente en la introducción de datos, está sujeta a restricciones de entrada. Se debe implementar una gestión de errores robusta para manejar posibles entradas incorrectas o maliciosas.

Requisitos específicos

Funciones básicas de los jugadores

Referencia	Función	Categoría
R1.1	Mostrar tablero	Evidente
R1.2	Leer la entrada de la fila	Evidente
R1.3	Leer la entrada de la columna	Evidente

Funciones del sistema

Referencia	Función	Categoría
R1.4	Conectarse a un socket con otro jugador	Oculto
R1.5	Selecciona quién comienza a jugar	Evidente
R1.6	Marcar una O en el tablero (Jugador 1)	Evidente
R1.7	Inicializar el tablero vacío	Oculto
R1.8	Marcar una X en el tablero (Jugador 2)	Evidente
R1.9	Verificar que la casilla de la entrada ingresada esté vacía	Oculto
R1.10	Verificar si hay un ganador	Oculto
R1.11	Verificar si el tablero está lleno sin ganadores (empate)	Oculto
R1.12	Limpiar terminal	Oculto
R1.13	Mostrar resultado de la partida	Evidente
R1.14	Terminar conexión entre ambos jugadores	Oculto

Atributos generales

Atributo	Detalle y restricciones
Interfaz de usuario	Información mostrada en la terminal a los jugadores.
Restricciones de acceso	Atributo que restringe el acceso a un jugador.

Atributos relacionados con cada función

Ref.	Función	Categoría de la función	Atributo	Detalles y restricción	Categoría del atributo
1	Mostrar tablero	Evidente	Interfaz de usuario		Obligatoria
2	Leer la entrada de la fila y Columna	Evidente			Obligatoria
3	Conectarse a un socket con otro jugador de forma online	Oculto			Obligatoria
4	Seleccionar quien comienza a jugar	Evidente	Interfaz de usuario	Solo el Servidor	Obligatoria
5	Marcar una O en el tablero (Jugador 1)	Evidente	Sin restricción		Obligatoria
6	Marcar una X en el tablero (Jugador 2)	Evidente	Sin restricción		Obligatoria
7	Verificar que la casilla de la entrada ingresada esté vacía	Oculto			Obligatoria
8	Verificar si hay un ganador	Oculto			Obligatoria
9	Verificar si el tablero está lleno sin ganadores (empate)	Oculto			Obligatoria
10	Inicializar el tablero vacío	Oculto			Obligatoria
11	Mostrar resultado de la partida	Evidente	Interfaz de usuario		Obligatoria
12	Terminar conexión entre ambos jugadores	Oculto			Obligatoria

Caso de uso

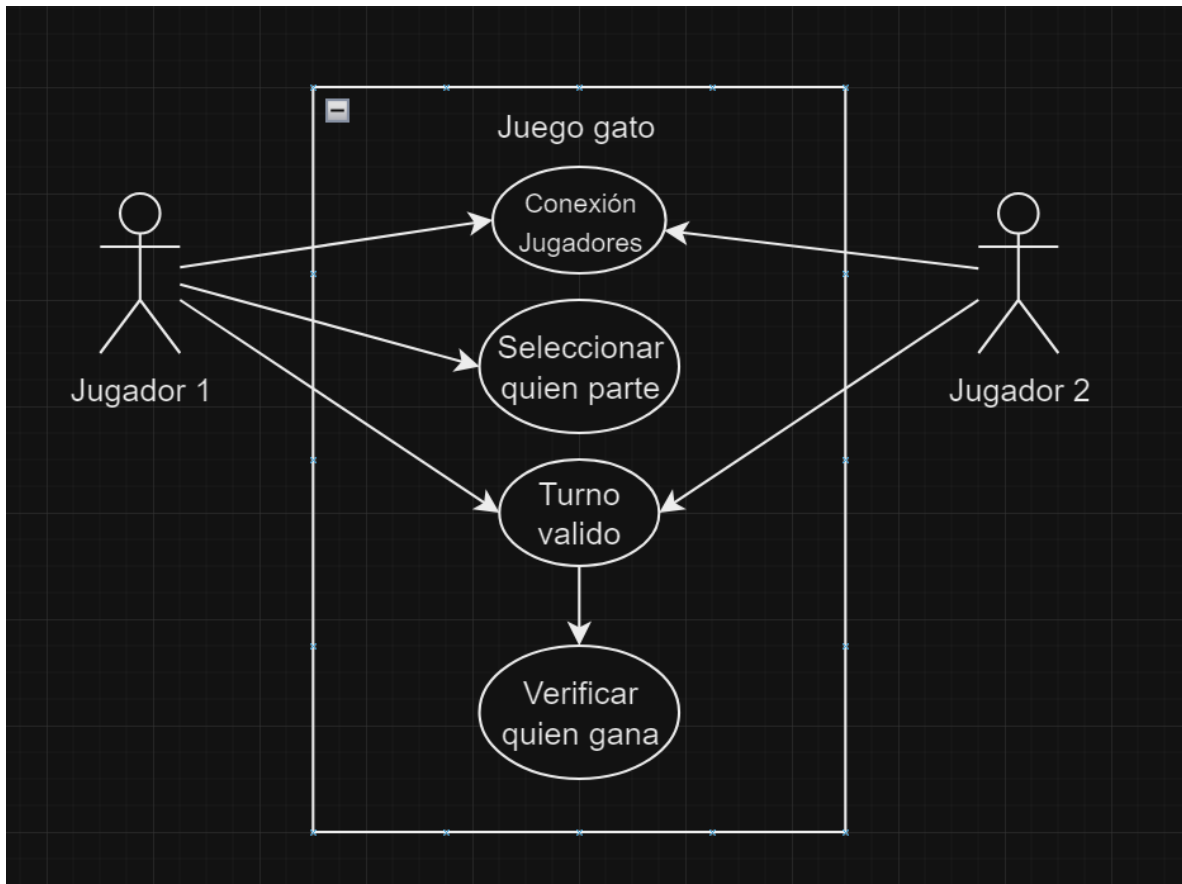


Diagrama de secuencia

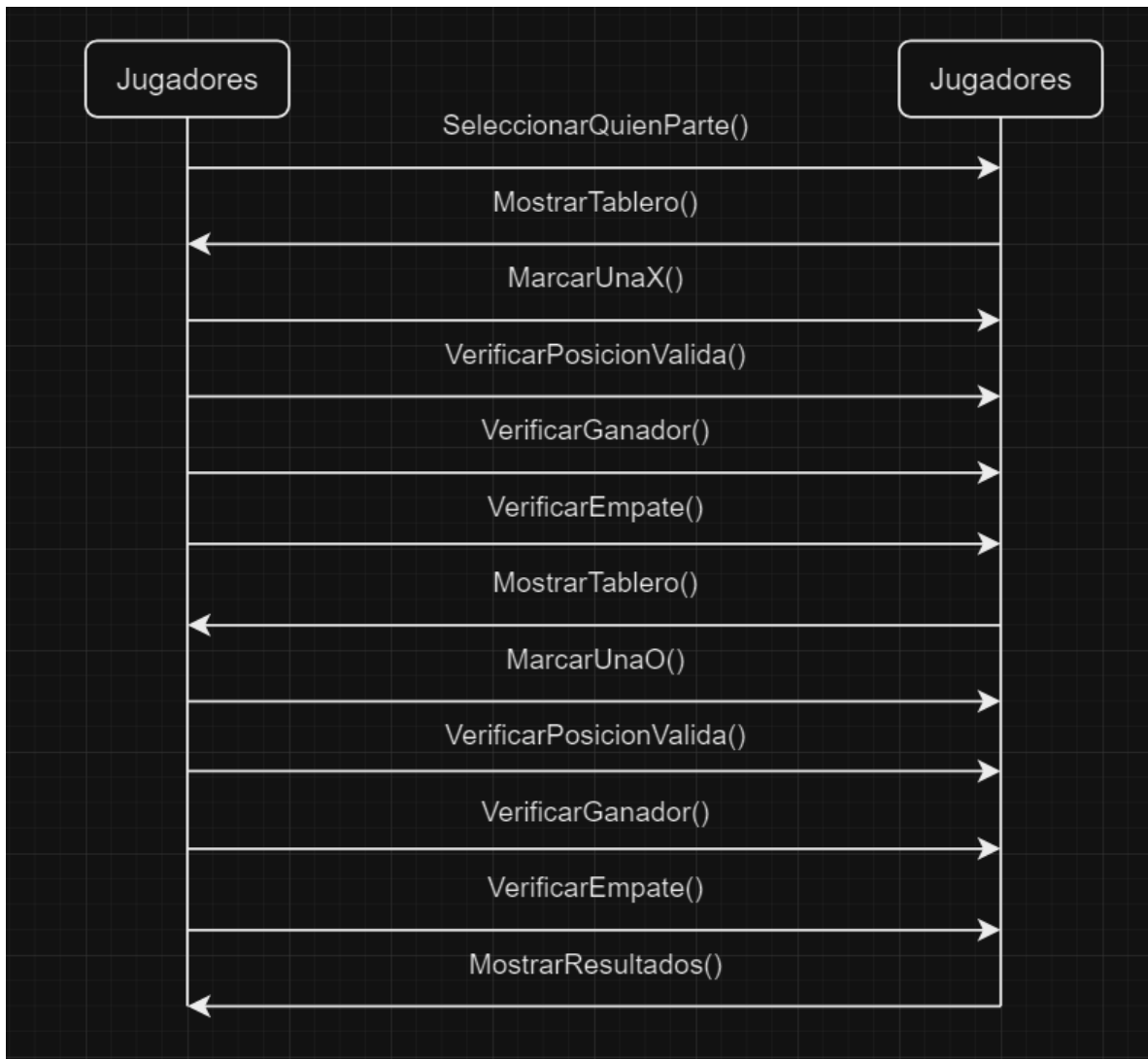


Diagrama de clases



Diagrama de colaboración

