

Instituto Tecnológico de Costa Rica

Algoritmos y Estructuras de Datos I

Leonardo Araya Martínez

Proyecto #1

Connect Dots

Emanuel Rojas Fernandez

Gabriel Fernandez Vargas

Jimmy Feng Feng

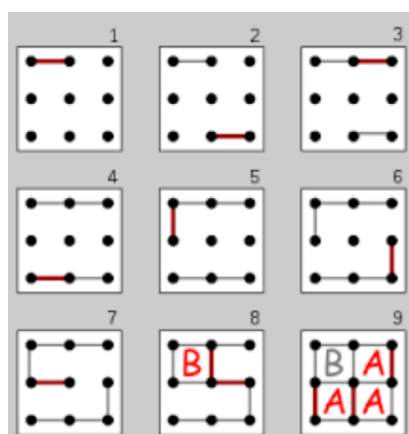
II Semestre 2023

## Índice

<i>1. Descripción del Problema:.....</i>	<i>3</i>
<i>2. Diagrama de Clases:.....</i>	<i>5</i>
<i>3. Descripción de Estructuras de Datos Desarrolladas:.....</i>	<i>6</i>
<i>4. Corridas de ejemplo con capturas de pantalla y explicaciones en prosa .....</i>	<i>7</i>

## 1. Descripción del Problema:

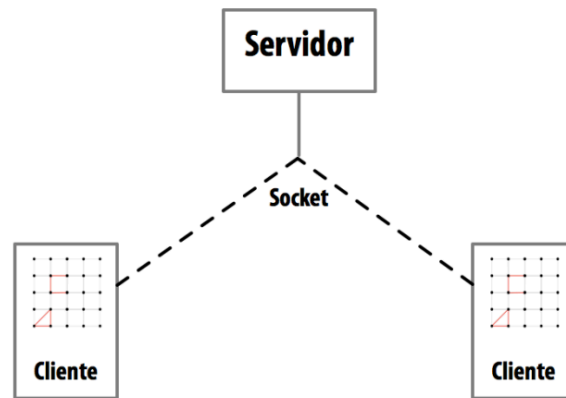
Connect dots es un juego multiplayer de  $n$  jugadores. El juego consiste en una malla de puntos donde el jugador puede unir dichos puntos mediante líneas por turno. Si el jugador logra cerrar un cuadrado en su turno, puede seguir jugando hasta que agregue una línea que no cierre un cuadrado. El objetivo del juego es cerrar cuadrados para obtener puntos. El jugador que logre cerrar más cuadrados al final de la partida, será el ganador.



Todos los cuadrados tienen el mismo valor. Cuando un jugador cierra un cuadrado, a este se le asignará algún identificador que permita saber cuál fue el jugador que lo cerró.

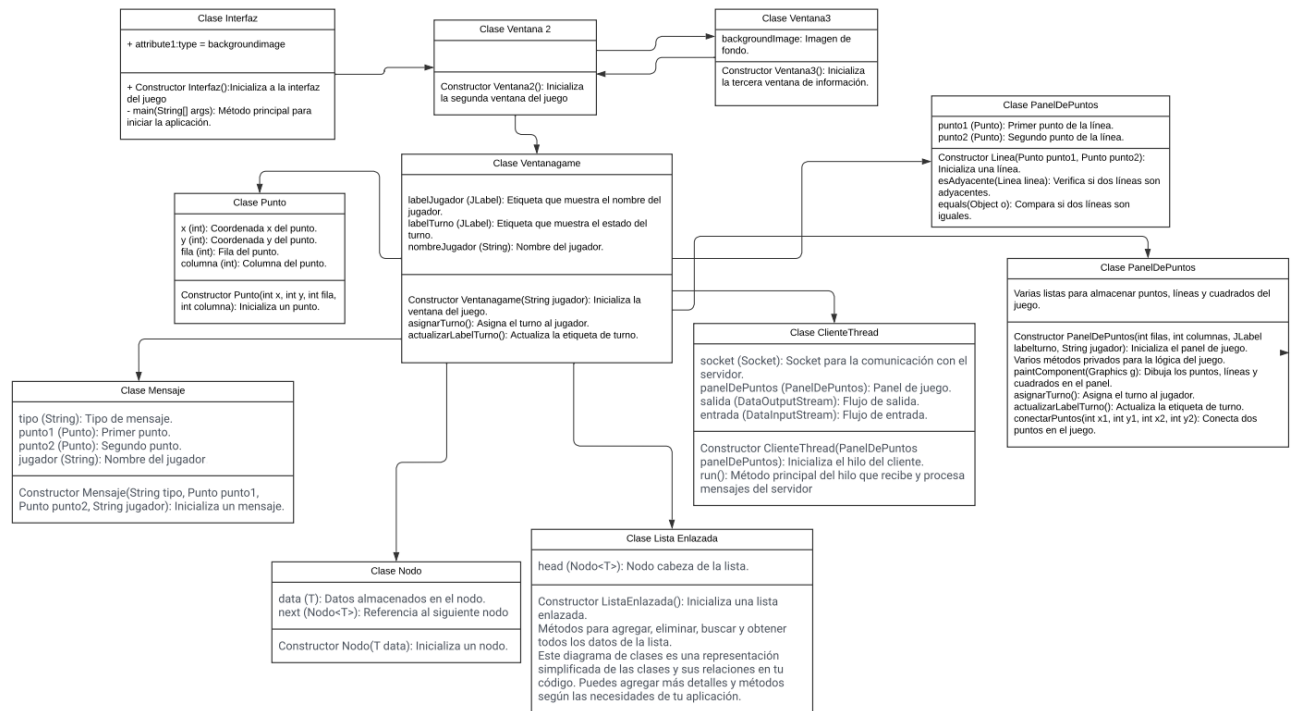
Dots es un juego multiplayer. Hay un servidor central que consiste en una aplicación en Java que escucha las conexiones entrantes por un Socket. Cada aplicación cliente se ejecuta en una computadora y se conecta por socket al servidor. Cuando el juego va a iniciar, el servidor recibe la petición del cliente y lo ingresa en una cola. En el momento que hayan dos o más jugadores, empieza el juego. Pueden jugar varios jugadores a la vez.

El servidor lleva el control completo del juego. Los clientes únicamente grafican lo que el servidor les envía. Los clientes a su vez envían las acciones que realicen al servidor, el cual se encarga de mantener el estado del juego completo. Toda la comunicación entre cliente y servidor es en formato JSON.



Adicionalmente, un cliente tendrá un control elaborado por los estudiantes que permitirá al jugador seleccionar dónde colocará las líneas.

## 2. Diagrama de Clases:



### 3. Descripción de Estructuras de Datos Desarrolladas:

- ☐ Lista Enlazada: Una lista enlazada utilizada para almacenar objetos. Esta estructura de datos se utiliza para mantener un registro de los puntos del juego y las líneas dibujadas entre ellos. Permite agregar, eliminar y buscar elementos en la lista enlazada de manera eficiente.
- ☐ Clase Punto: Representa un punto en el tablero del juego. Cada objeto 'Punto' tiene dos propiedades principales: las coordenadas 'x' e 'y', que indican la posición del punto en el plano. Estas coordenadas permiten ubicar el punto en el tablero y establecer relaciones espaciales entre puntos.
- ☐ Clase Línea: Representa una línea entre dos puntos en el tablero del juego. Está diseñada para administrar las relaciones entre los puntos y verificar las propiedades de las líneas, como si son adyacentes o perpendiculares.
- ☐ Clase Queue: La clase "Queue es una implementación de una estructura de datos de cola genérica. Una cola es una estructura de datos que sigue el principio "primero en entrar, primero en salir" (FIFO, por sus siglas en inglés), lo que significa que el primer elemento que se agrega a la cola es el primero en ser eliminado de ella. Es utilizado para implementar la funcionalidad de turnos de los jugadores.

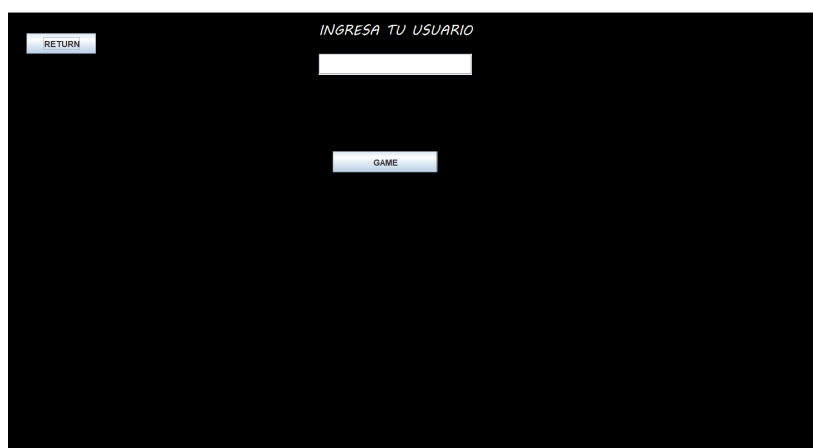
#### 4. Corridas de ejemplo con capturas de pantalla y explicaciones en prosa



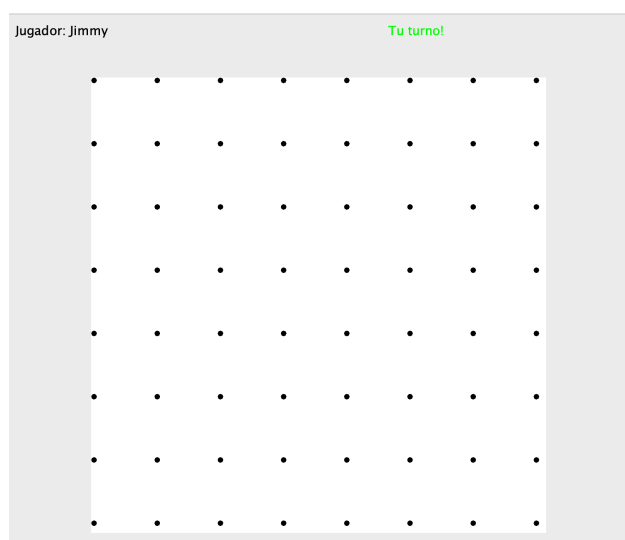
En esta captura de pantalla, podemos ver la interfaz principal del juego "Connect Dots". El jugador tiene la opción de "PLAY" (Jugar) o "INFO" (Información).



En esta otra captura de pantalla podemos ver lo que pasa al estripar el botón INFO (información), se pueden ver las instrucciones del juego, el control y los nombres de los desarrolladores

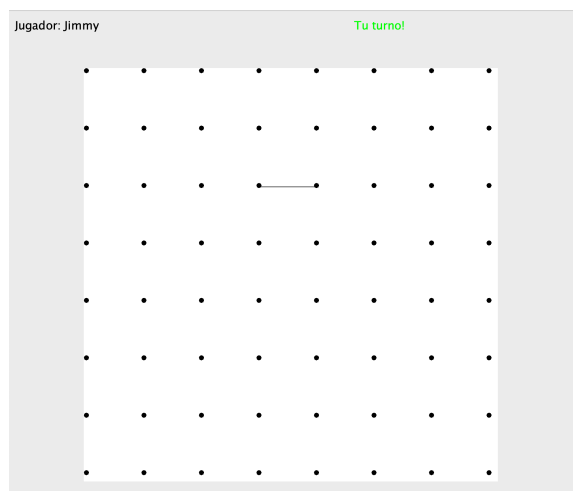


En esta imagen podemos ver o que pasa al estripar el botón de PLAY (jugar), sale un cuadro de texto para poner el nombre de usuario y empezar el juego.



En esta imagen podemos ver el tablero después de ingresar el nombre de usuario y darle el botón de GAME (Juego). Aquí se muestra el nombre y el estado de los turnos.

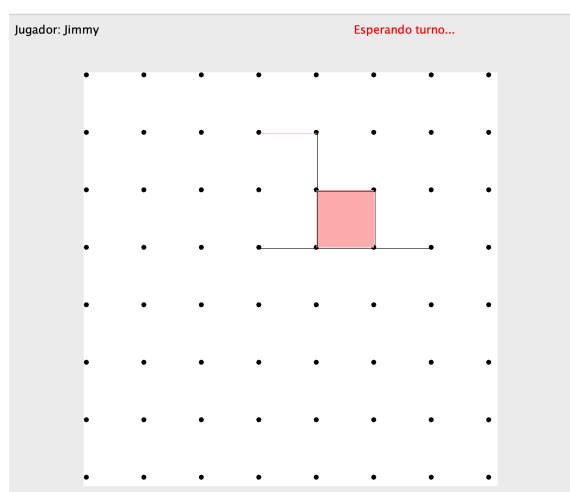




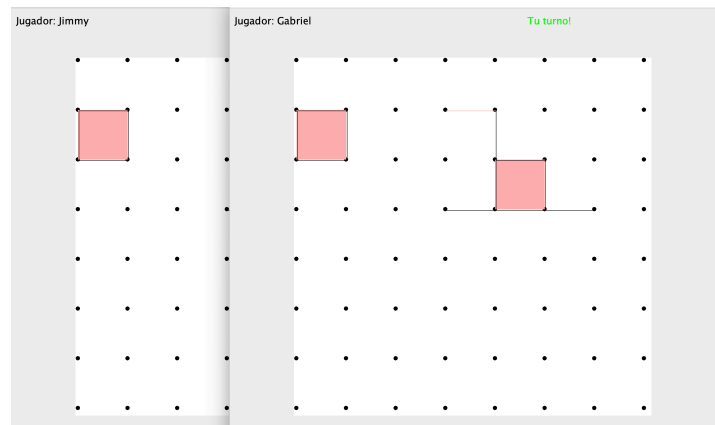
En esta imagen, se puede observar cuando se selecciona dos puntos distintos en el tablero, lo cual genera una línea horizontal o vertical.

[Server] Es mi turno. Puedes escribir ahora.  
 {"jugador": "Jimmy", "y1": 205, "x1": 405, "y2": 205, "x2": 305}

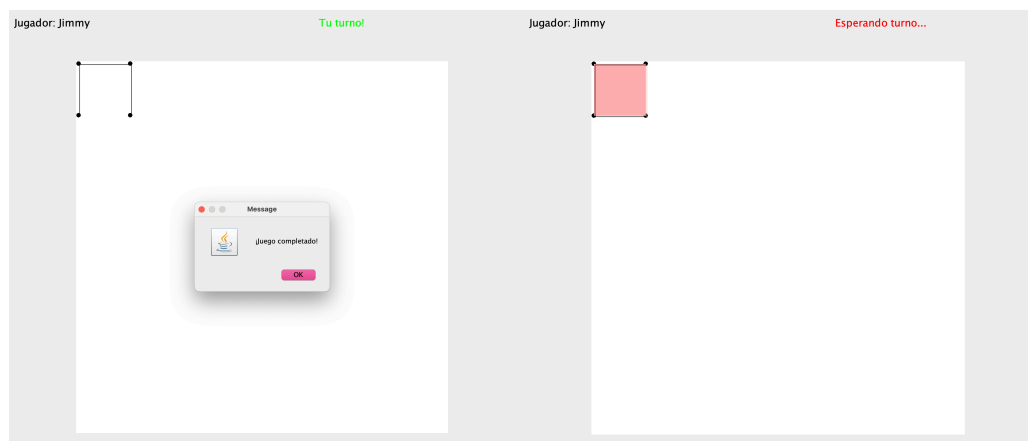
Cuando se traza una línea, el cliente manda un mensaje al servidor en formato JSON, y lo recibe como lo muestra en la imagen anterior.



Cuando se conectan 4 líneas para formar un cuadrado, rellena de color el cuadrado como se muestra en la imagen adjunta anterior.



Aquí se puede observar la interacción que tiene cliente con cliente.



Lo anterior es un ejemplo simple de lo que pasaría si se completara todos los cuadros, al trazar la última línea, aparece una indicación de que se terminó el juego y posteriormente muestra como quedaría al final del juego.