

**Reto 2: Actualización de diseño e implementación del Triqui Distribuido**

**Antecedentes del Reto2 del Triqui:**

Durante el desarrollo de las clases y como material de apoyo en Eafit Interactiva, se ha entregado dos (2) versiones del triqui. La versión 1, es un Triqui Standalone y la Version 2 es un Triqui Simple distribuido cliente/servidor bajo Sockets TCP.

La versión 1 obviamente, permite jugar un triqui al tiempo y los 2 jugadores deben estar en la misma máquina. La interface de usuario es modo Consola.

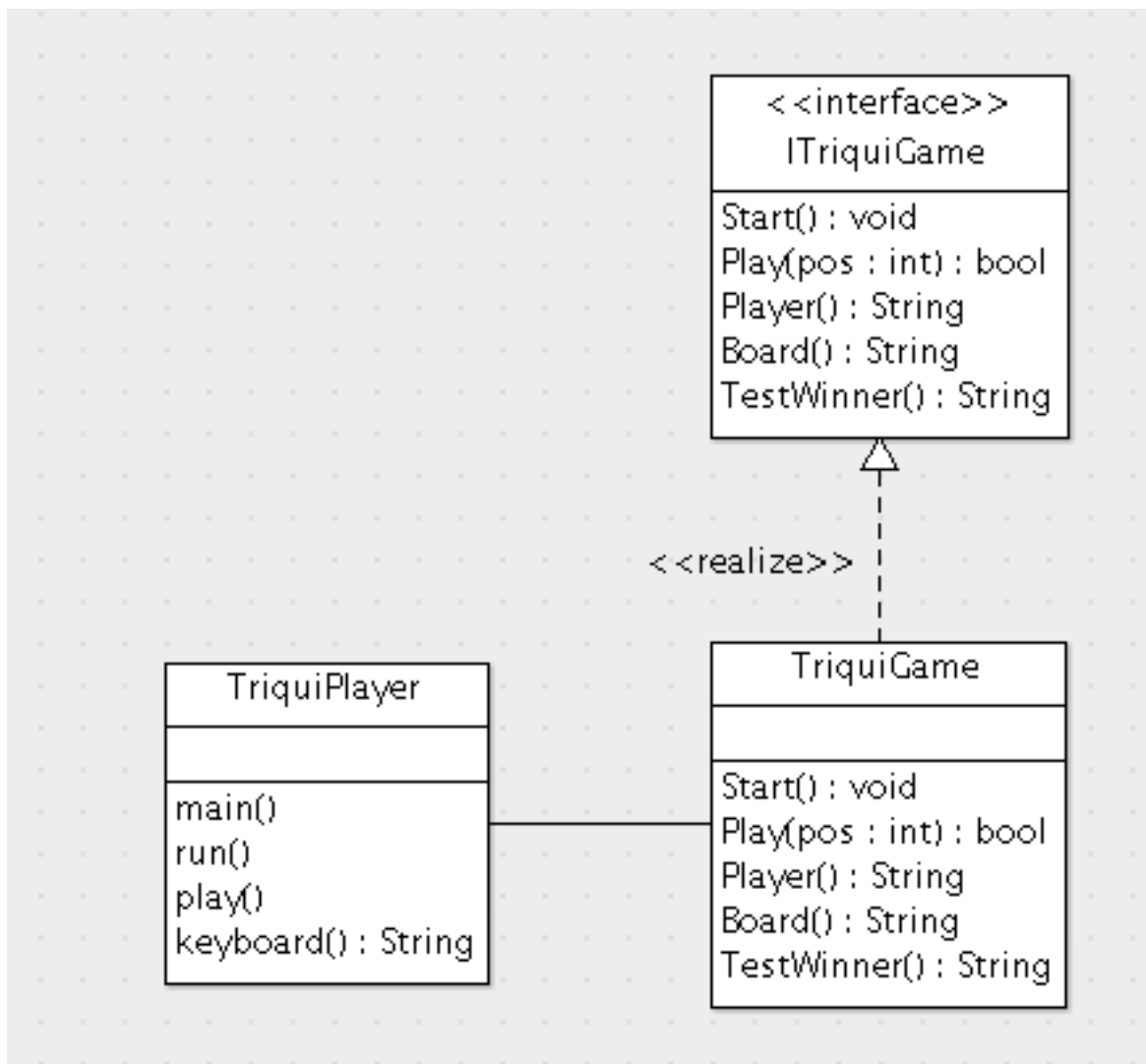


Figura 1: Diagrama de clases versión standalone

La versión 2, implementa un servidor de triqui, el cual si bien permite varios juegos remotos, NO los permite al tiempo, solo atiende un juego al tiempo. Del lado del cliente, el triqui permite que los 2 jugadores estén en la misma interface y máquina, es decir, los jugadores tienen que estar en el mismo sitio.

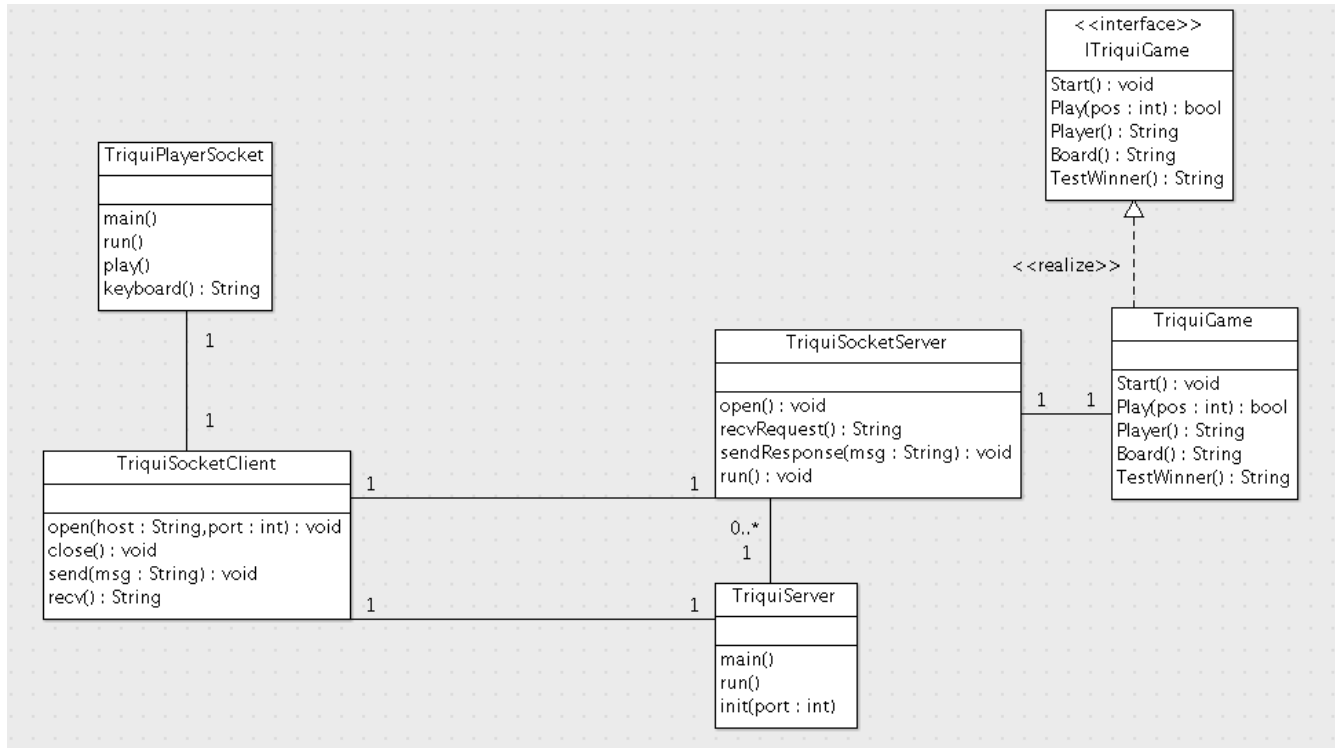


Figura 2. Diagrama de clases versión Sockets

Más detalles del diseño e implementación de estas 2 versiones, se encuentran en este documento y el materia publicado en Eafit Interactiva.

### Requerimientos del Reto2:

A continuación se presenta los requerimientos del reto 2, los cuales se sintetizan en los siguientes 3:

1. El Servidor debe atender a los diferentes clientes en modo concurrente, es decir, que permita atender varios juegos simultáneamente varias sesiones de juego remotas.
2. Realizar una modificación al diseño e implementación que permita a los 2 jugadores, ESTAR EN DOS CLIENTES O MAQUINAS DISTRIBUIDAS. Es decir, ya el TriquiPlayer solo conecta a un Jugador. Esto implica, que se deben adicionar funcionalidades al servidor para que permita GESTIONAR la conexión y los juegos entre los JUGADORES distribuidos. Este requerimiento es nuevo, y no existía en las versiones 1 y 2 del Triqui.
3. Realizar el diseño y la implementación en una arquitectura P2P, en la cual, el Cliente posea la lógica del juego, e implemente algún mecanismo para coordinar las acciones del juego (un solo juego, y los 2 clientes se conectan, uno local y el otro remoto. O dos juegos, los cuales deben coordinar las acciones). Se recomienda utilizar una arquitectura P2P híbrida basada en Servidor, en el cual se implementen los servicios de registro, búsqueda y localización de Clientes. La

funcionalidad de los servicios ofrecidos por el Servidor **DEBEN SER IMPLEMENTADOS POR INVOCACIÓN REMOTA**, no por Sockets.

Entregables:

- Documento de Análisis y Diseño Global o de Arquitectura.
- Código Fuente
- Aplicación desplegada en el servidor sistemas.eafit.edu.co

Fecha de entrega:

Febrero 26 de 2014