



Documento de Arquitectura de Software

Alumnos-

Alejandro Galindo Covarrubias 00000181388

José Francisco Felix Romero 00000181332

Fecha-

03 de diciembre 2019

Materia-

Arquitectura de Software

Profesor-

Gilberto Borrego Soto

Tabla de contenidos

1.- Introducción.	1
1.1. Propósito.	1
1.2. Alcance.	1
1.3. Definiciones, Acrónimos, y Abreviaciones	1
2. Representación Arquitectónica.	1
3. Objetivos Arquitectónicos y Restricciones de Diseño.	2
4. Vista de Escenarios.	3
4.1. Diagrama de casos de uso.	3
4.2. CU1 Registrar Jugador.	3
4.3. CU2 Iniciar Juego.	3
4.4. CU3 Jugar.	3
4.5. CU4 Configurar Tablero.	3
4.6. CU5 Cerrar Timbiriche.	3
4.7. CU6 Salir al menú principal.	3
5. Vista Lógica	4
5.1. Diagrama de dominio	4
5.2. Diagramas de secuencia a nivel de análisis.	5
5.2.1. CU Iniciar Juego	5
5.2.2. CU Jugar	6
6. Vista de implementación.	7
6.1. Diagrama de componentes.	7
Tabla 6.2. Responsabilidades de los componentes.	7
6.2. Diagrama de clases de cada componente.	8
6.2.1. Cliente	8
6.2.2. SckServer	9
6.2.3. DominioDTO	9
6.2.4. PipesAndFilters	10
6.3. Realización de casos de uso.	11
6.3.1. CU Iniciar Juego	11
6.3.2. CU Jugar	12
7. Vista de despliegue	13
7.1. Diagrama de despliegue.	13
Tabla 7.2. Descripción de los elementos del diagrama de despliegue	13

Documento de Arquitectura de Software

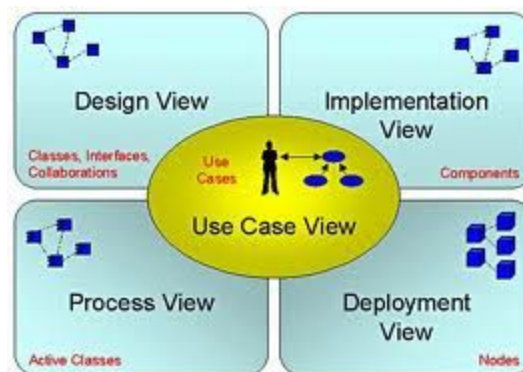
1.- Introducción.

Este documento proporciona una descripción general de alto nivel y explica la arquitectura del Juego de Timbiriche.

El documento define los objetivos de la arquitectura, los casos de uso admitidos por el sistema, los estilos arquitectónicos y los componentes que se han seleccionado. El documento proporciona una justificación para las decisiones de arquitectura y diseño tomadas desde la idea conceptual hasta su implementación.

1.1. Propósito.

Este documento proporciona una descripción arquitectónica completa del Juego de Timbiriche. Presenta varias vistas arquitectónicas diferentes para representar los diferentes aspectos del sistema. Para representar el software con la mayor precisión posible, la estructura de este documento se basa en la vista de arquitectura de Arquitectura "4 + 1" de Philippe Kruchten.



1.2. Alcance.

El alcance de este documento es explicar la arquitectura del juego Timbiriche; describe los diversos aspectos del diseño del sistema, que se consideran arquitectónicamente significativos. Estos elementos y comportamientos son fundamentales para guiar la construcción del sistema y para comprender este proyecto en su conjunto.

1.3. Definiciones, Acrónimos, y Abreviaciones

Acrónimo	Definición
Timbiriche.	Traducción en español del juego Dots And Boxes.
PAF.	Pipes And Filters - Estilo Arquitectónico.

2. Representación Arquitectónica.

Este documento detalla la arquitectura utilizando las vistas definidas en el modelo "4 + 1" [Kruchten]. Las vistas utilizadas para documentar el sistema son:

- **Vista de escenarios.**
 - **Audiencia:** Todas las partes interesadas del sistema, incluidos los usuarios finales.

- **Área:** Describe el conjunto de escenarios y / o casos de uso que representan alguna funcionalidad central significativa del sistema.
- **Artefactos relacionados:** Modelo de caso de uso, documentos de caso de uso.
- **Vista lógica.**
 - **Audiencia:** Diseñadores/Programadores.
 - **Área:** Describe las entidades clave.
 - **Artefactos relacionados:** Diagramas de clases
- **Vista de implementación.**
 - **Audiencia:** Diseñadores/Arquitectos.
 - **Área:** Describe la organización en componentes del sistema, así como la comunicación entre ellos.
 - **Artefactos relacionados:** Diagrama de componentes.
- **Vista de despliegue.**
 - **Audiencia:** Gerentes de implementación.
 - **Área:** Describe el mapeo del software en el hardware y muestra los aspectos distribuidos del sistema. Describe las posibles estructuras de implementación, al incluir escenarios de implementación conocidos y anticipados en la arquitectura, permitimos que los implementadores hagan ciertas suposiciones sobre el rendimiento de la red, la interacción del sistema, etc.
 - **Artefactos relacionados:** Diagrama de despliegue.

3. Objetivos Arquitectónicos y Restricciones de Diseño.

El patrón arquitectónico Pipes And Filters es descrito como un patrón estructural, pues detalla la manera en la que distintos objetos se comunican unos con otros, formando consigo estructuras más grandes y/o complejas que puedan implementar nueva funcionalidad, más concretamente, el objetivo de PAF es el de separar el flujo de un proceso en acciones individuales que, usualmente reciben una entrada y generan una salida; Estos son llamados Filters. Los Filters son unidos entre sí por Pipes o vías que solo sirven para dirigir de la manera que sea necesaria, la salida generada por un Filter hacia la entrada de otro, para que al concluir el proceso se tenga una salida, que terminará en un “Sink”, o un objetivo común donde el resultado de todo el proceso puede terminar.

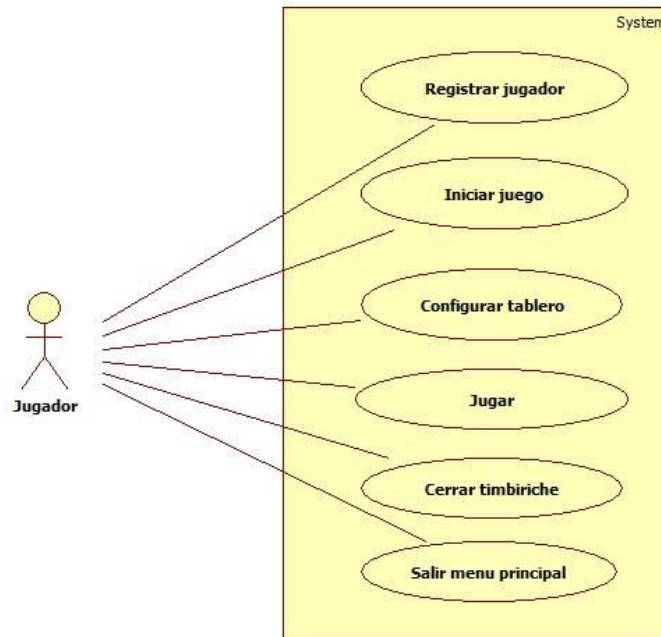
Por lo tanto, la arquitectura PAF, apoyada en gran parte del primer principio de SOLID, el “Principio de Responsabilidad Única” restringe al diseño a ser separado según las funcionalidades del sistema, y consecuentemente las responsabilidades de cada función en pasos más pequeños que constituyan a una sola acción, y que estas lleguen a un mismo punto.

El uso de sockets en el diseño del sistema involucra además que el diseño arquitectónico sea complejo, y que se vaya acompañado de los demás principios SOLID, sobre todo del quinto, el de “Inversión de dependencias”, donde, al usar sockets, dependemos de abstracciones, no de clases concretas.

4. Vista de Escenarios.

El propósito de esta vista es proporcionar un contexto adicional que rodee el uso del sistema y las interacciones entre sus componentes. A continuación, se presenta el diagrama de casos de uso, así como la descripción de los mismos, que son arquitectónicamente significativos del sistema.

4.1. Diagrama de casos de uso.



4.2. CU1 Registrar Jugador.

Este caso de uso permite al Jugador registrarse para ingresar al juego.

4.3. CU2 Iniciar Juego.

Este caso de uso permite a los Jugadores dentro de una sala empezar la partida.

4.4. CU3 Jugar.

Este caso de uso permite a los Jugadores jugar “Timbiriche”.

4.5. CU4 Configurar Tablero.

Este caso de uso permite a los Jugadores configurar los colores de su tablero.

4.6. CU5 Cerrar Timbiriche.

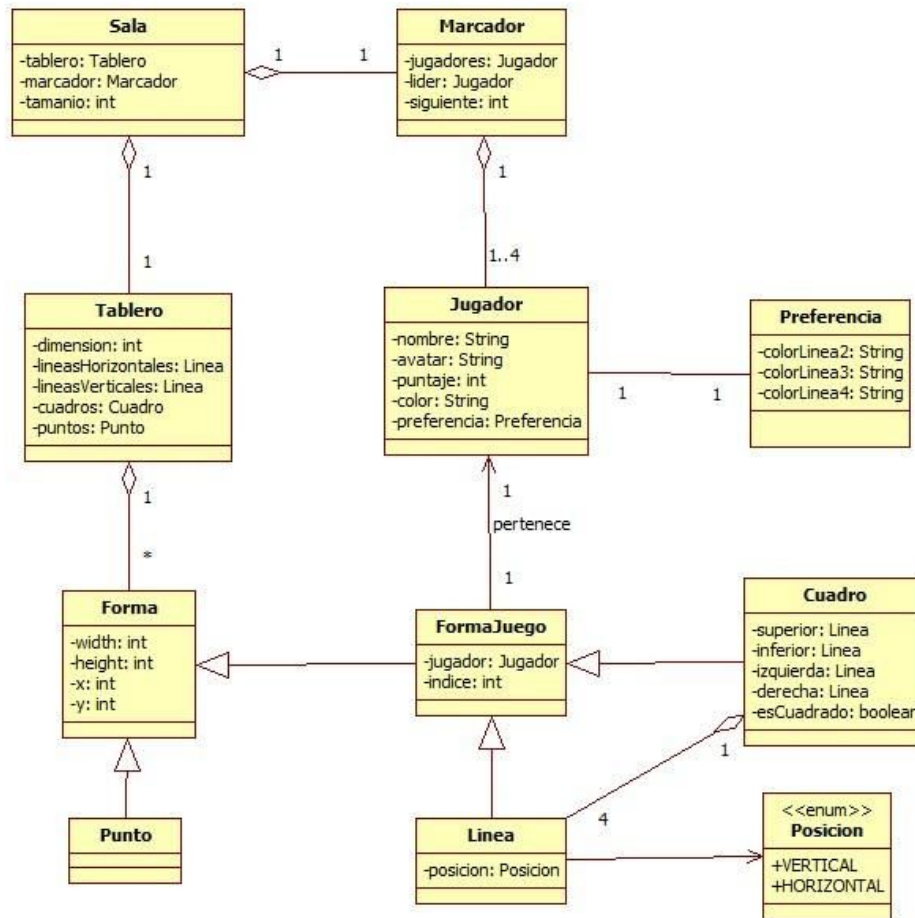
Este caso de uso permite al usuario cerrar el juego.

4.7. CU6 Salir al menú principal.

Este caso de uso permite al usuario salir al menú principal del juego.

5. Vista Lógica

5.1. Diagrama de dominio

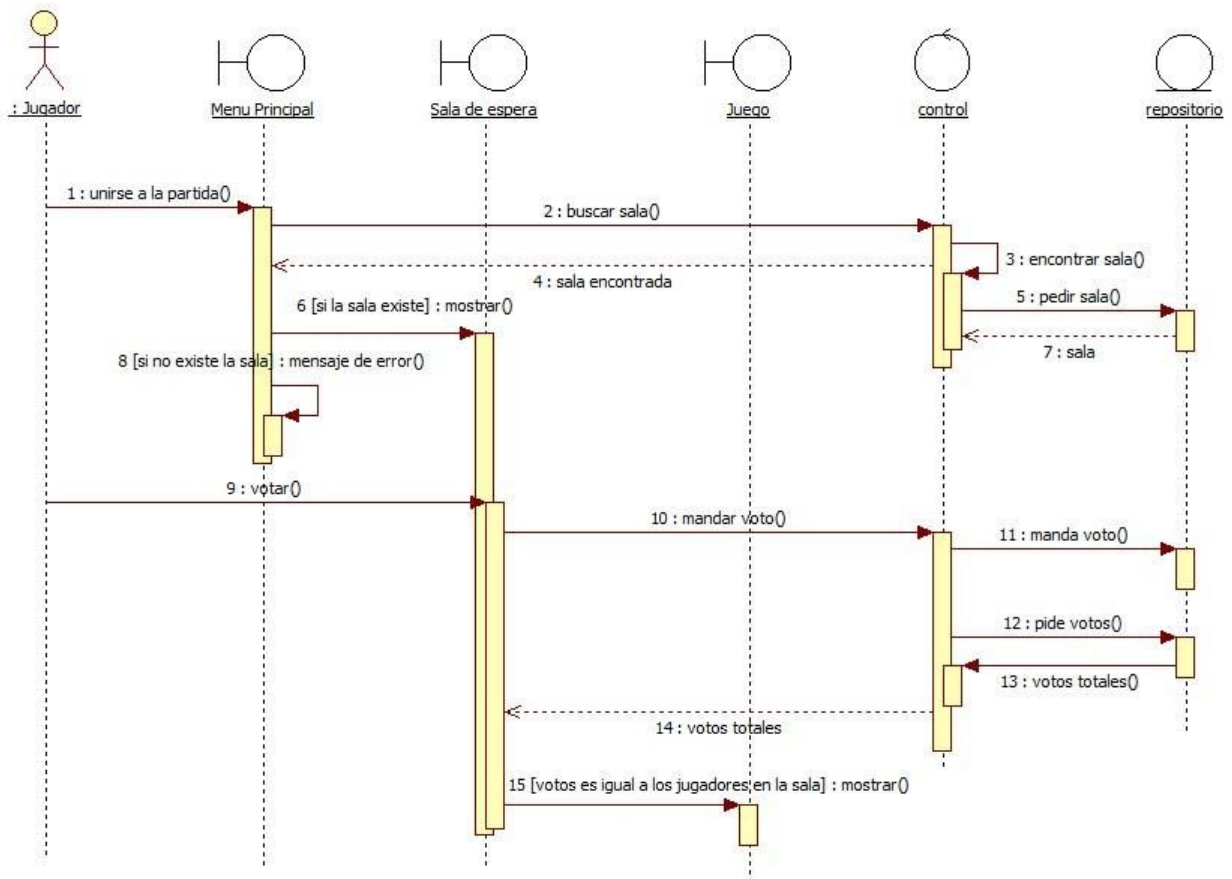


El Dominio del sistema es representado por una clase **Sala** como clase central y representación de todo el “juego”, la cual se compone de un **Marcador**, encargado de dictaminar al líder del juego y al siguiente en turno, además de representar la totalidad de 1 a 4 jugadores en partida. Cada **Jugador** tiene un nombre, una ruta hacia su avatar, el puntaje, el color preferente para sí mismo y además de una **Preferencia**, que representa los colores para el resto de jugadores en partida.

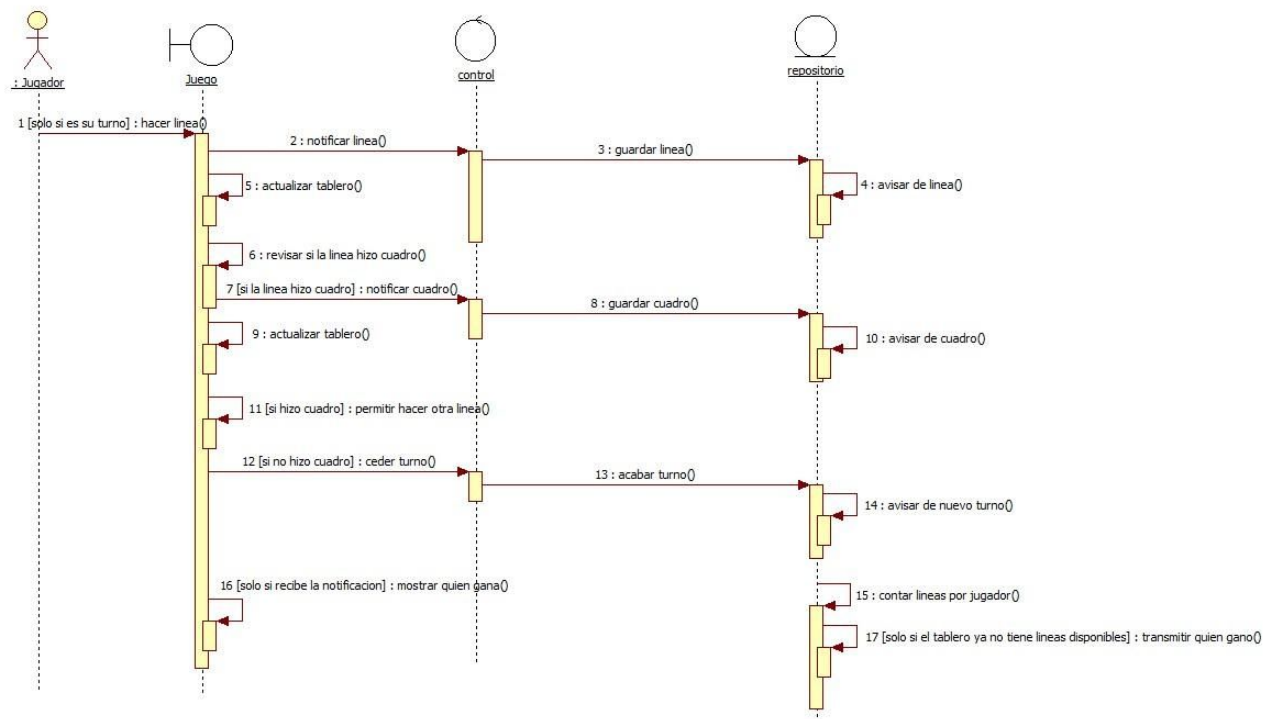
Por otro lado la sala se compone también de un **Tablero**, dicho tablero se compone de todas las **Forma** posibles para su posterior representación en el sistema, como los puntos del tablero determinados por su dimensión. En esta parte del Dominio se marca una diferencia entre las **Forma** pertenecientes al despliegue del **Tablero** y a las **Forma** pertenecientes al proceso del juego, creando así una “doble herencia” con **FormaJuego**, la cual, a diferencia de **Forma**, pertenece a un **Jugador**. De **FormaJuego** heredan **Linea**, que además de sus inherentes atributos de **Forma** y **FormaJuego**, su colocación en el tablero es determinada por su **Posicion**, una clase que guarda constantes de las posibles posiciones de una **Linea**, y por último también hereda **Cuadro**, que se compone de dos **Linea** de **Posicion VERTICAL** y dos **Linea** de **Posicion HORIZONTAL**, y un atributo que determina si está terminado o no, es decir, que las cuatro **Linea** tienen un **Jugador** asignado.

5.2. Diagramas de secuencia a nivel de análisis.

5.2.1. CU Iniciar Juego



5.2.2. CU Jugar



6. Vista de implementación.

6.1. Diagrama de componentes.

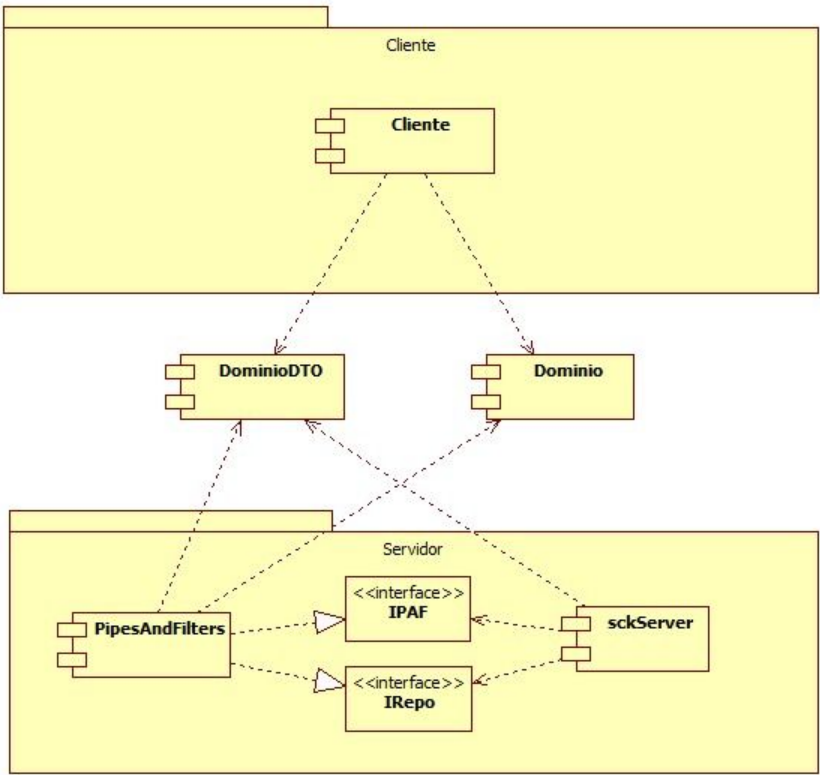
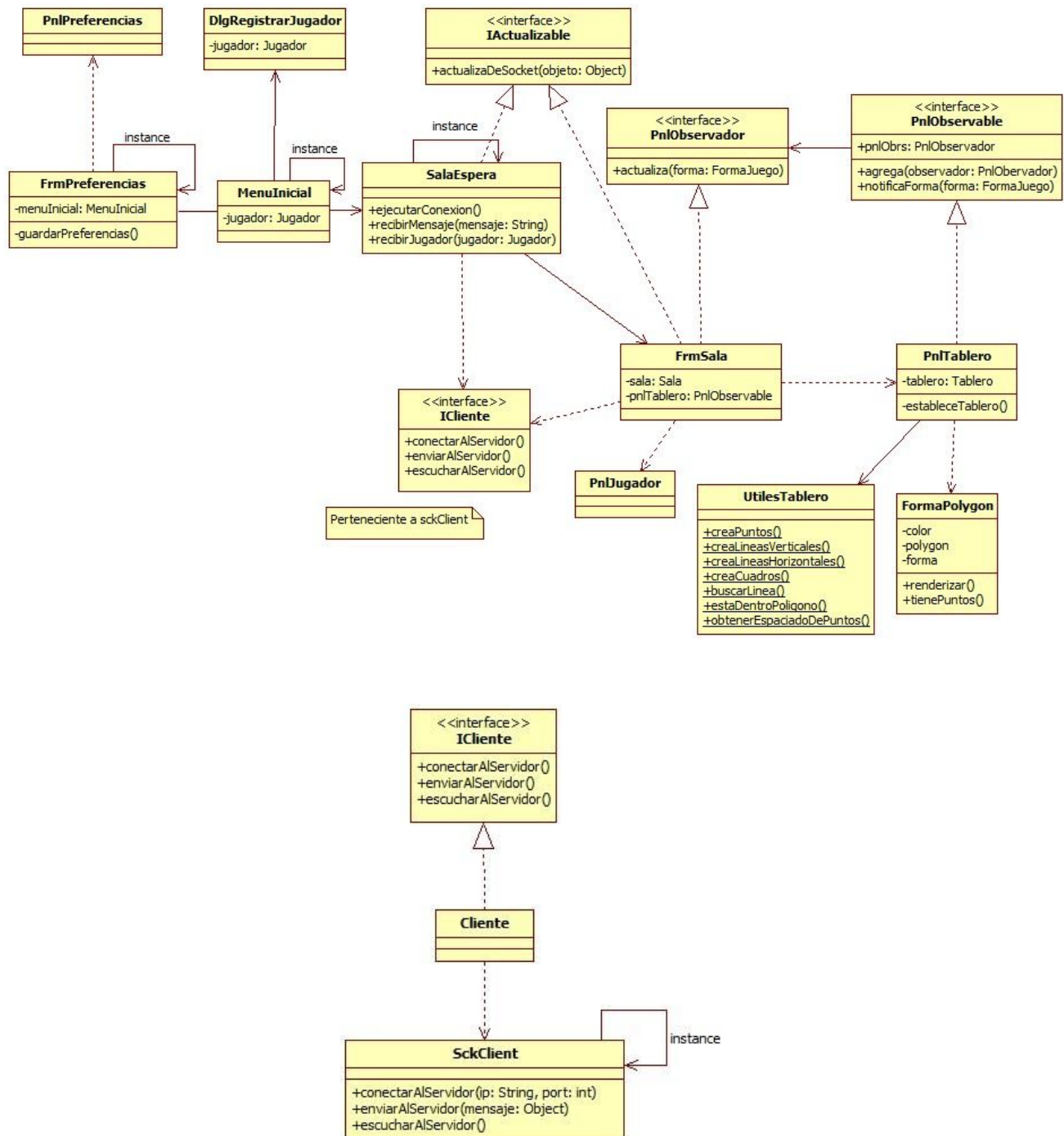


Tabla 6.2. Responsabilidades de los componentes.

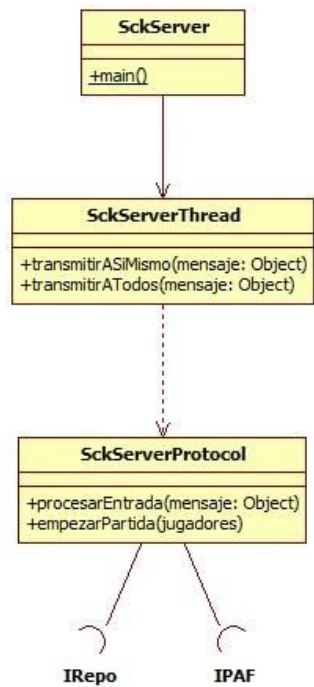
Componente	Responsabilidades
Dominio	Abstraer las funcionalidades del sistema y la solución al problema
DominioDTO	Abstraer la abstracción de las funcionalidades del sistema para usarse exclusivamente como objetos de comunicación entre sckCliente y sckServer
cliente	Desplegar la interfaz de usuario, comunicar de los cambios a sckCliente y recibir actualizaciones del Servidor.
	Establece la conexión por parte del Cliente vía sockets con el Servidor.
sckServer	Establece la conexión por parte del Servidor vía sockets con los Cliente. Se encarga de recibir conexiones simultáneas y decidir qué hacer con los mensajes entrantes.
PipesAndFilters	Se encarga de procesar datos recibidos, y convertirlos de objetos de DominioDTO a objetos de Dominio para después guardarlos en un repositorio común expuesto fuera del componente.

6.2. Diagrama de clases de cada componente.

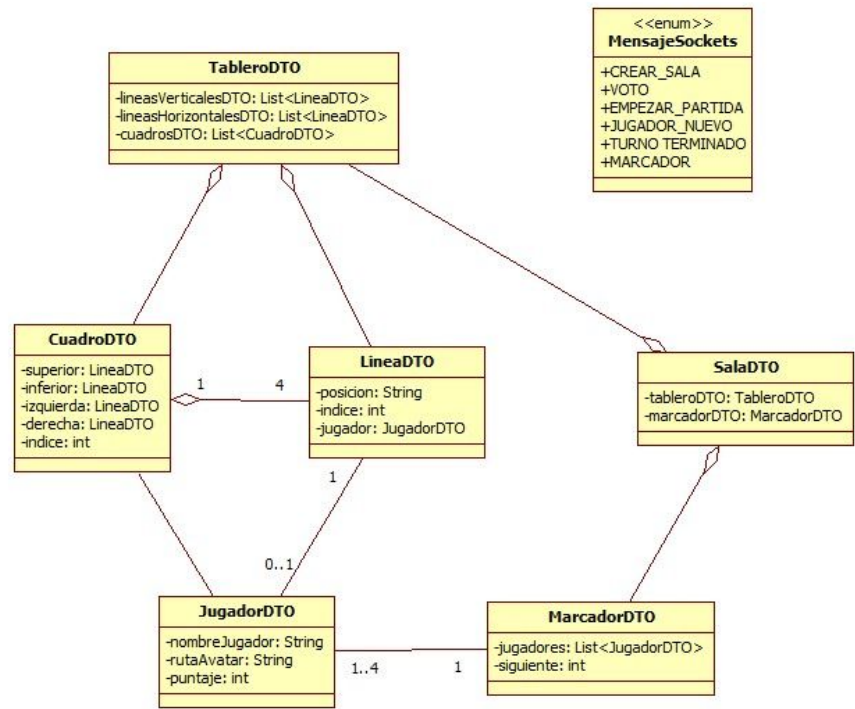
6.2.1. Cliente



6.2.2. SckServer



6.2.3. DominioDTO



Documento de Arquitectura de Software

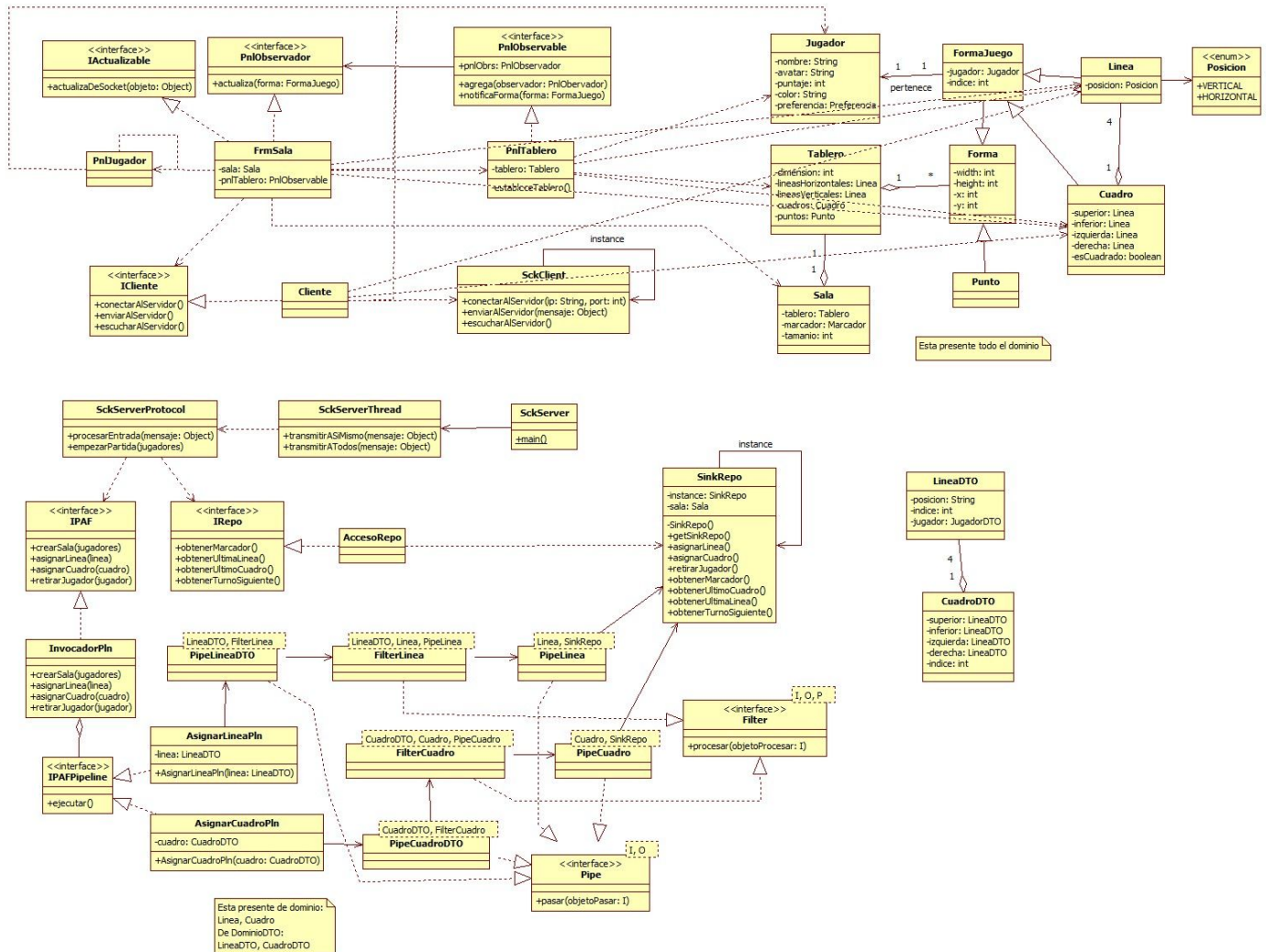
Timbiriche



6.3.1. CU Iniciar Juego

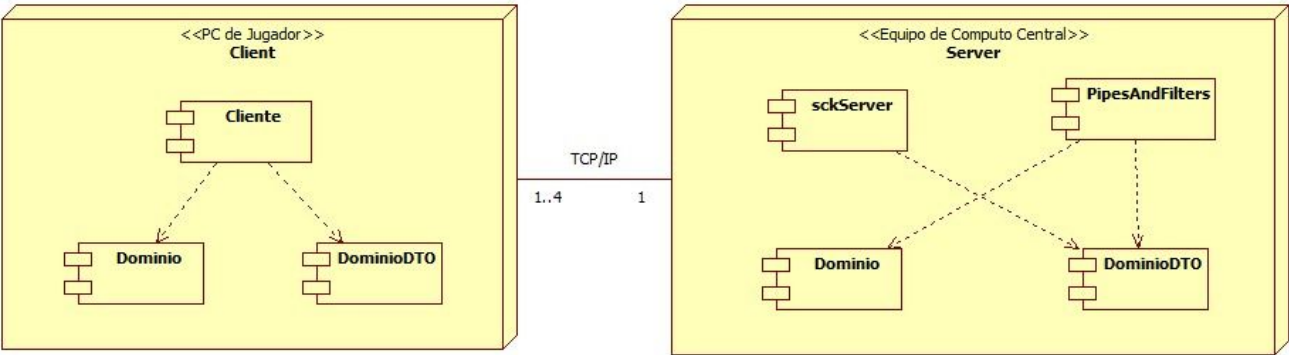


6.3.2. CU Jugar



7. Vista de despliegue

7.1. Diagrama de despliegue.



El sistema será dividido en dos partes, una parte en un Servidor o equipo central que a través de tecnología de Sockets permitirá la interacción con el sistema desplegado en uno de los Cliente. Dentro del Servidor serán llevadas las tareas del manejo de conexiones simultáneas con clientes e hilos, la creación de objetos de Dominio y DominioDTO en un componente PAF y de repositorio del sistema como un Sink. En la parte de Cliente se desplegará las interfaces gráficas de usuario y la conectividad con tecnología socket del cliente al servidor.

Tabla 7.2. Descripción de los elementos del diagrama de despliegue

Elemento	Descripción
Cliente	Una PC perteneciente a un jugador, donde será posible jugar Timbiriche y conectarse a un Servidor
Servidor	Un servidor central o equipo de computo concreto que da soporte a la partida de Timbiriche en la cual interactúan los jugadores