

**Proyecto final de**

**Arquitectura de Software**

**Alumnos-**

Cesar Alan Acatictla Medina

**Fecha-**

13 de diciembre 2019

**Materia-**

Arquitectura de Software

**Profesor-**

Gilberto Borrego Soto

INSTITUTO TECNOLÓGICO DE SONORA

**Tabla de contenidos**

[**1.- Introducción** 1](#_Toc11368)

[1.1. Propósito 1](#_Toc11369)

[1.2. Alcance 1](#_Toc11370)

[1.3. Definiciones, Acrónimos, y Abreviaciones 1](#_Toc11371)

[**2. Representación Arquitectónica** 1](#_Toc11372)

[**3. Objetivos Arquitectónicos y Restricciones de Diseño** 2](#_Toc11373)

[**4. Vista de Escenarios** 3](#_Toc11374)

[4.1. Diagrama de casos de uso 3](#_Toc11375)

[4.2. CU1 Registrar Jugador 3](#_Toc11376)

[4.3. CU2 Iniciar Juego 3](#_Toc11377)

[4.4. CU3 Jugar 3](#_Toc11378)

[4.5. CU4 Configurar Tablero 3](#_Toc11379)

[4.6. CU5 Cerrar Timbiriche 3](#_Toc11380)

[4.7. CU6 Salir al menú principal 3](#_Toc11381)

[5. Vista Lógica 4](#_Toc11382)

[5.1. Diagrama de dominio 4](#_Toc11383)

[5.2. Diagramas de secuencia a nivel de análisis 5](#_Toc11384)

[5.2.1. CU Iniciar Juego 5](#_Toc11385)

[5.2.2. CU Jugar 6](#_Toc11386)

[**6. Vista de implementación** 7](#_Toc11387)

[6.1. Diagrama de componentes 7](#_Toc11388)

[Tabla 6.2. Responsabilidades de los componentes 7](#_Toc11389)

[6.2. Diagrama de clases de cada componente 8](#_Toc11390)

[6.2.1. Presentación 8](#_Toc11391)

[6.2.2. SckClient y SckServer 8](#_Toc11392)

[6.2.3. DominioDTO 9](#_Toc11393)

[6.2.4. PipesAndFilters 10](#_Toc11394)

[6.3. Realización de casos de uso 11](#_Toc11395)

[6.3.1. CU Iniciar Juego 11](#_Toc11396)

[6.3.2. CU Jugar 12](#_Toc11397)

[**7. Vista de despliegue** 13](#_Toc11398)

[7.1. Diagrama de despliegue 13](#_Toc11399)

[Tabla 7.2. Descripción de los elementos del diagrama de despliegue 13](#_Toc11400)

**Documento de Arquitectura de Software**

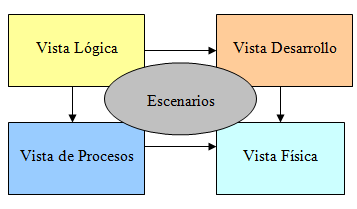
# 1.- Introducción.

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

En el documento se definen los objetivos de la arquitectura, entre los cuales podremos observar que tenemos los casos de uso, los estilos arquitectónicos y los componentes seleccionados.

### Propósito.

En el presente documento proporciono diferentes vistas arquitectónicas para representar los diferentes aspectos los cuales el sistema tiene. Para poder representar el software con la mayor calidad posible.



### 

### 1.2. Alcance.

Para este documento presentado se estableció un alcance en base al tiempo de desarrollo y diseño los cuales se establecieron en un inicio, por lo tanto, se considera que se llevó el trabajo a un punto el cual se considera completado y en el cual se describen los diversos diseños implementados en el sistema.

### 1.3. Definiciones, Acrónimos, y Abreviaciones

**Acrónimo:** Timbiriche

**Definición** : Timbiriche es la traducción al español del juego Dots And Boxes que significa puntos y cajas.

# 2. Representación Arquitectónica.

En el presente documento se detalla la arquitectura la cual fue utilizada, utilizando las vistas las cuales fueron definidas en el modelo 4 + 1.

Para esta parte mencionare las vistas las cuales fueron utilizadas para el sistema fueron:

* **Escenarios.**
* **Lógica.**
* **Implementación.**
* **Despliegue**
* 3. Objetivos Arquitectónicos y Restricciones de Diseño.

Con el uso del patrón arquitectónico Pies and filters se describe como un patrón estructural, ya que este detalla la manera en la cual distintos objetos se comunican unos con otros, formando consigo estructuras mas grandes y también mas complejas las cuales pueden implementar nuevas funciones. Y en donde tenemos los llamados Filters.

Los Filters como su nombre lo dice representan un filtro y el cual puede ser unido entre sí por Pipes que solo sirven para dirigir de la manera que sea necesaria.

Asi que el uso de la arquitectura pies and filters, apoya en gran parte de lo que se conoce como principio de Solid, el cual restringe al diseño a ser separado y con ello consecuentemente las responsabilidades de las funciones del sistema.

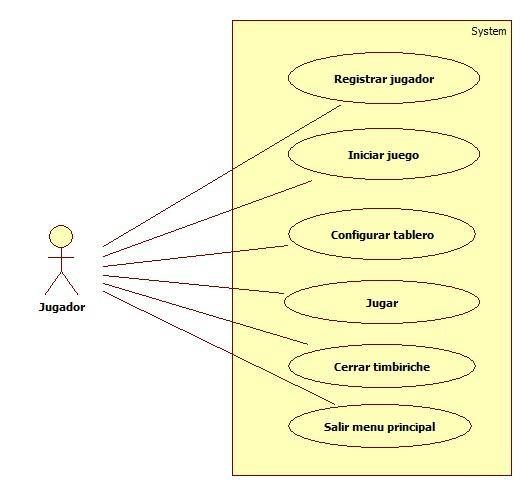
Y por último el uso de sockets es un plus que se le da al sistema para la conexión de multiple usuarios los cuales pueden interactuar con el sistema para pode hacer uso de el y el juego timbiriche.

# 4. Vista de Escenarios.

El propósito de la vista de escenario es da un mejor contexto del uso del sistema y la interacción entre los componentes.

### 

### 4.1. Diagrama de casos de uso.



### 4.2. CU1 Registrar Jugador.

Solo se permite al Jugador registrarse para iniciar el juego.

### 4.3. CU2 Iniciar Juego.

Solo se le permite a los Jugadores dentro de una sala comenzar la partida.

### 4.4. CU3 Jugar.

Permite a los Jugadores jugar el juego.

### 4.5. CU4 Configurar Tablero.

Los jugadores se les permite editar y configurar los colores de su tablero.

### 4.6. CU5 Cerrar Timbiriche.

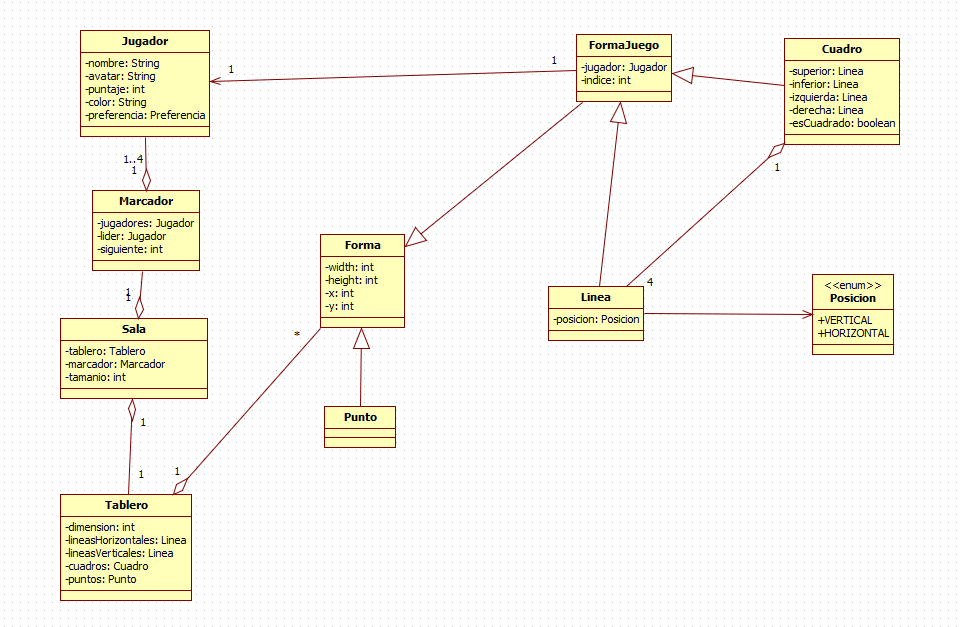
Se le permite al jugar poder cerrar el juego.

### 4.7. CU6 Salir al menú principal.

Se le permite al jugar poder salir al menú principal del juego.

## 5. Vista Lógica

### 5.1. Diagrama de dominio

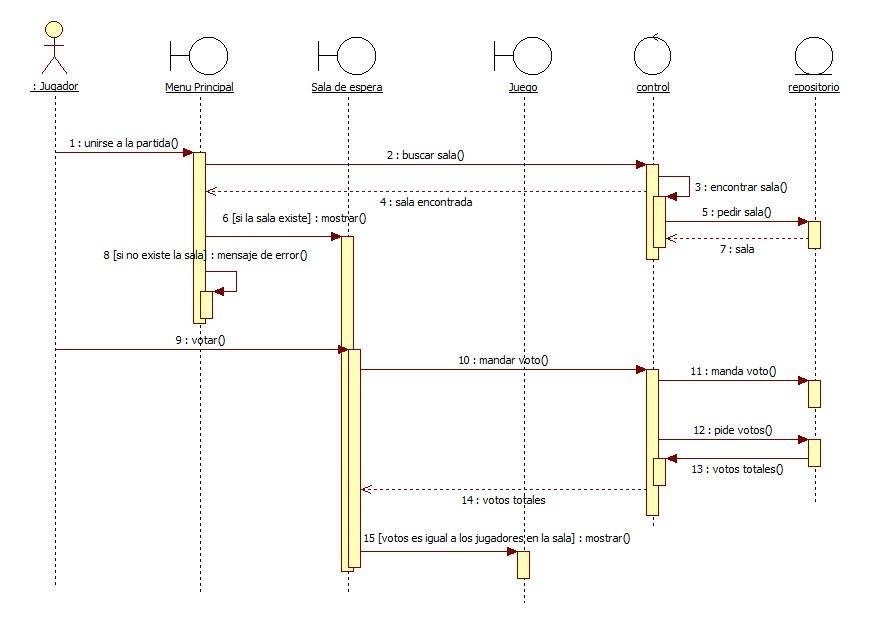


En la parte del dominio del sistema este se le da una representación por medio de una clase como clase central y representación de todo el juego y en la cual estos se componen de un marcador, encargado de dictaminar al host del juego y al siguiente turno, además de representar la totalidad de 1 a 4 jugadores en una partida.

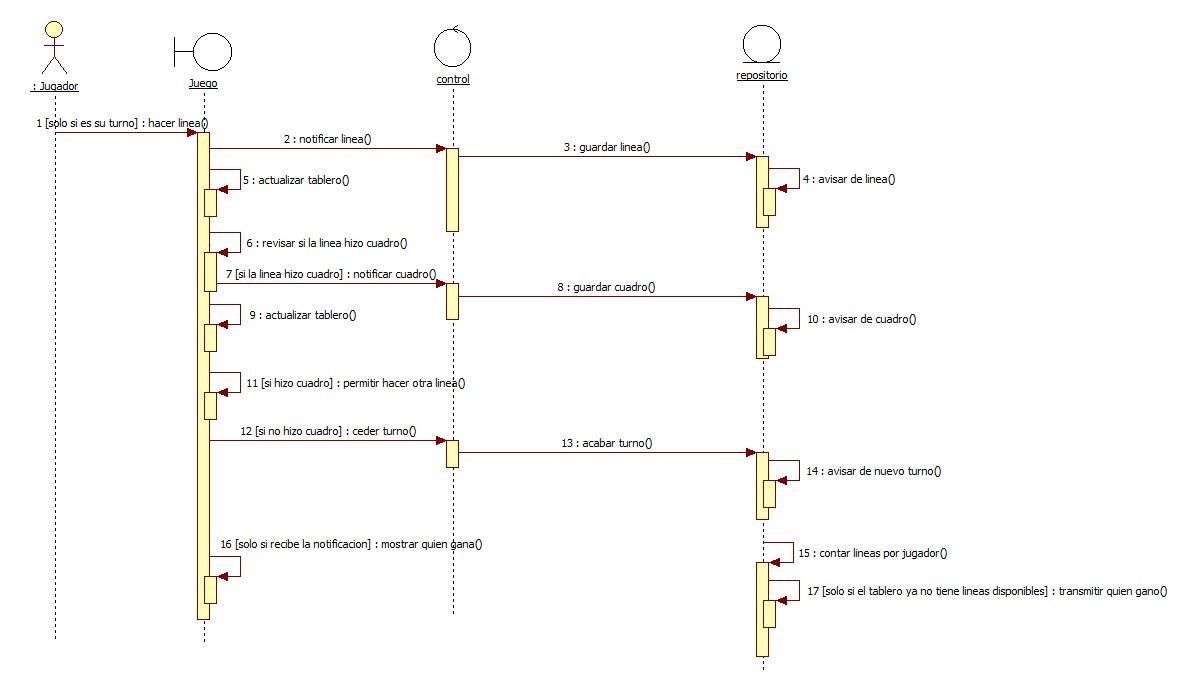
Cada uno de los jugadores tiene su nombre, una ruta hacía su avatar, el puntaje, el color preferente para sí mismo y además de una **Preferencia**, que representa los colores para el resto de jugadores en partida.

### 5.2. Diagramas de secuencia a nivel de análisis.

#### 5.2.1. CU Iniciar Juego

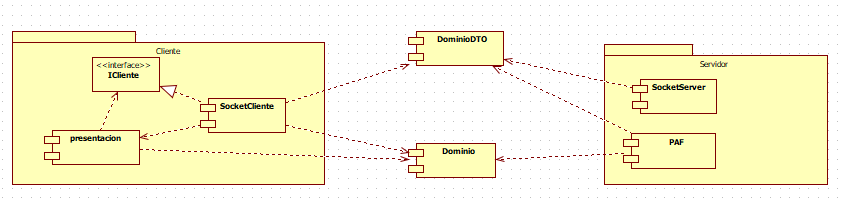


#### 5.2.2. CU Jugar



# 6. Vista de implementación.

### 6.1. Diagrama de componentes.



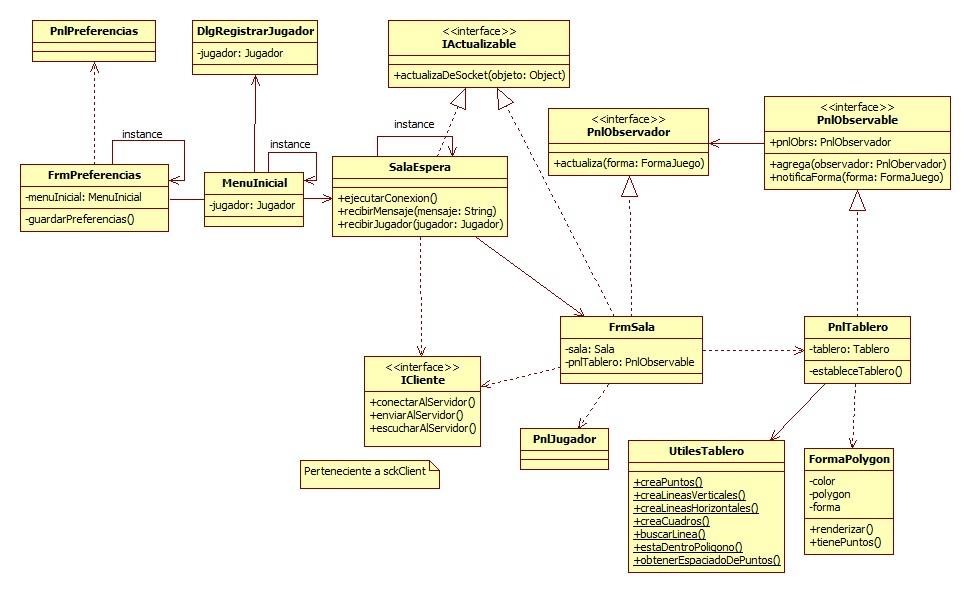
## 

## Tabla 6.2. Responsabilidades de los componentes.

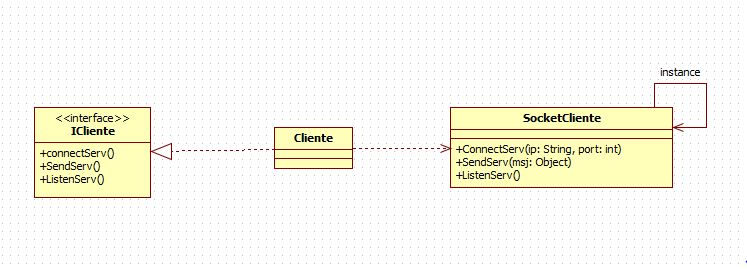
|  |  |
| --- | --- |
| **Componente** | **Responsabilidades** |
| **Dominio** | Obtiene las funcionalidades del sistema y la solución al problema. |
| **DTODominio** | Obtiene las funcionalidades del sistema para usarse como objetos de comunicación entre SocketCliente y también el SocketServer |
| **presentacion** | Muestra la interfaz del usuario y comunica los cambios al SocketCliente y recibir actualizaciones del Servidor. |
| **SocketCliente** | Establece la conexión del Cliente vía sockets con el Servidor. |
| **SocketServer** | Se encarga de recibir conexiones simultáneas y decidir qué hacer con los mensajes entrantes, el cual recibe la conexión simultánea y decide que hacer con el mensaje. |

### 6.2. Diagrama de clases de cada componente.

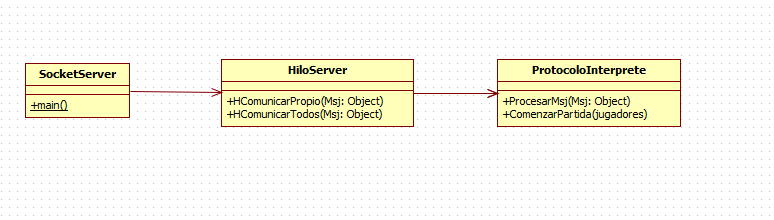
#### 6.2.1. Presentación



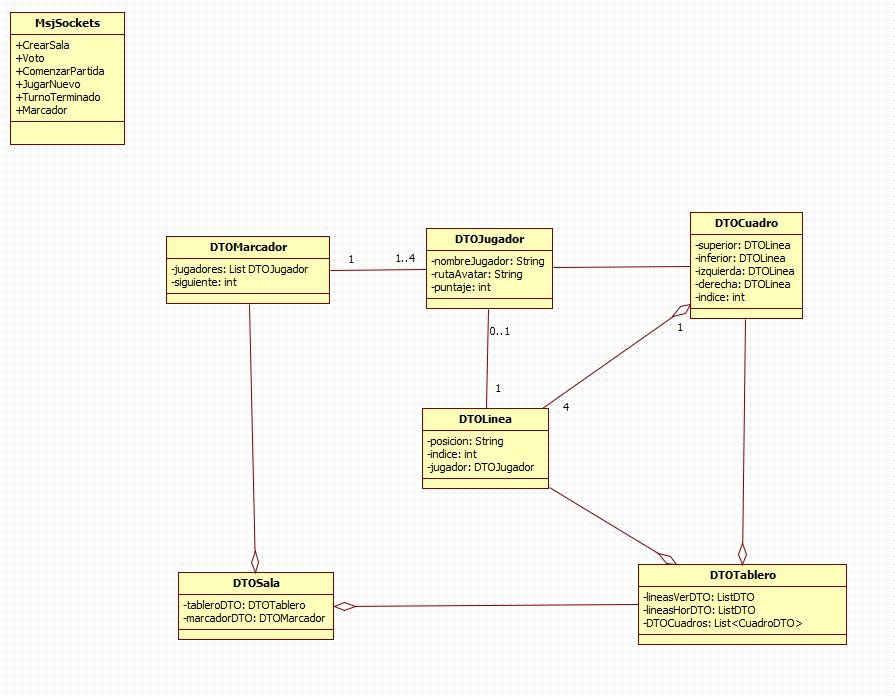
#### 6.2.2. SckClient



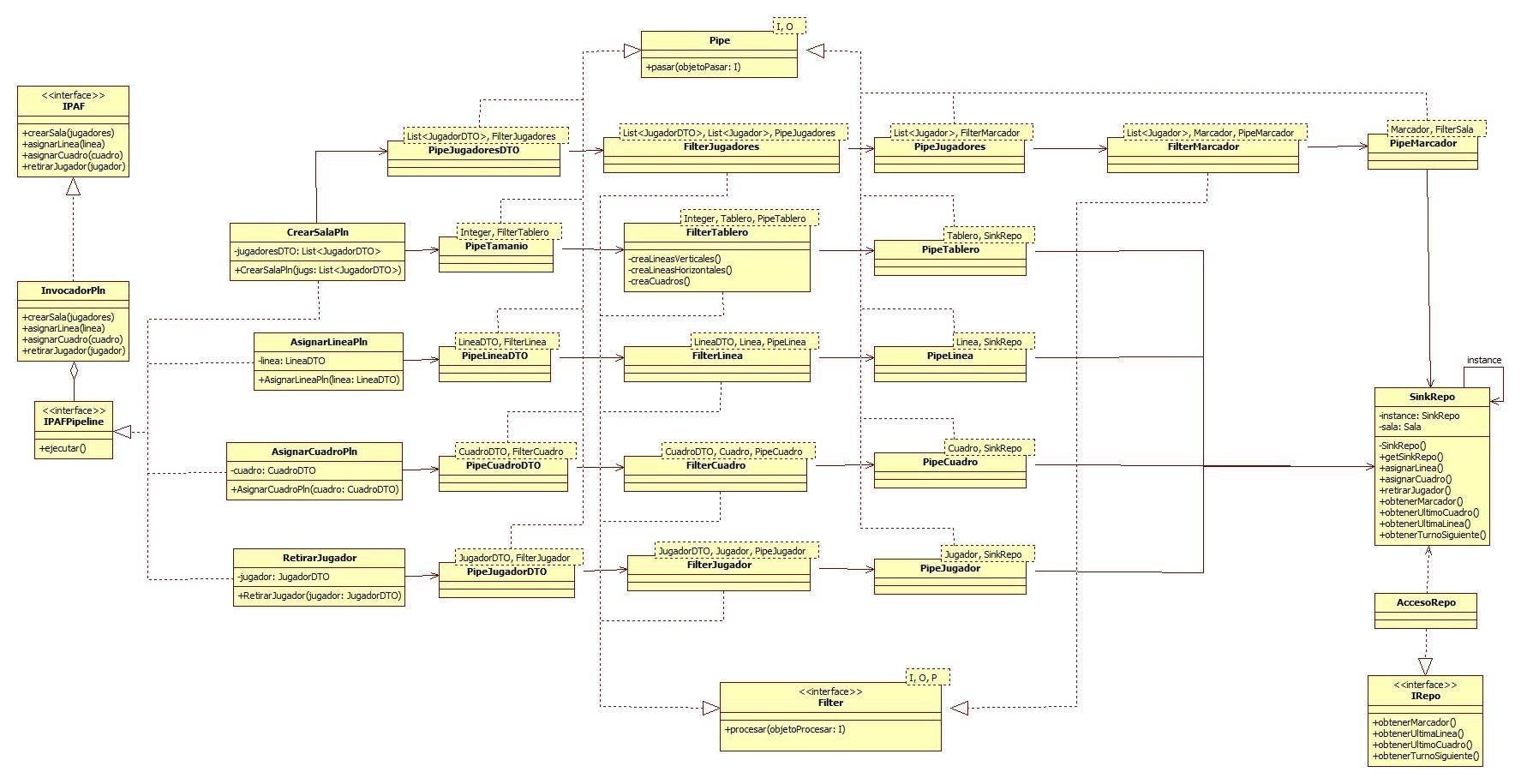
#### SckServer



#### 6.2.3. DominioDTO



#### 6.2.4. PipesAndFilters

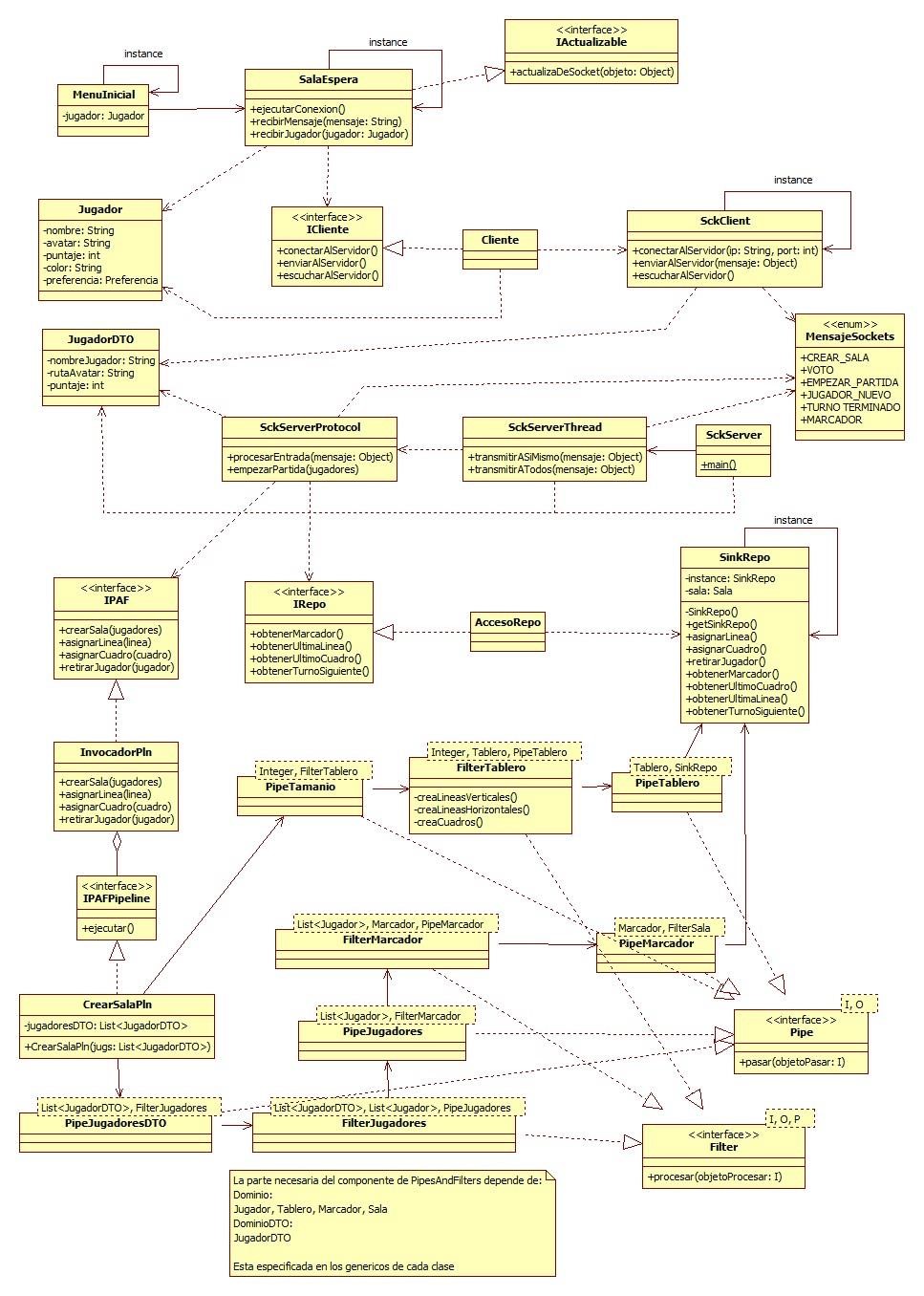


**Documento de Arquitectura de Software 10 Juego:**

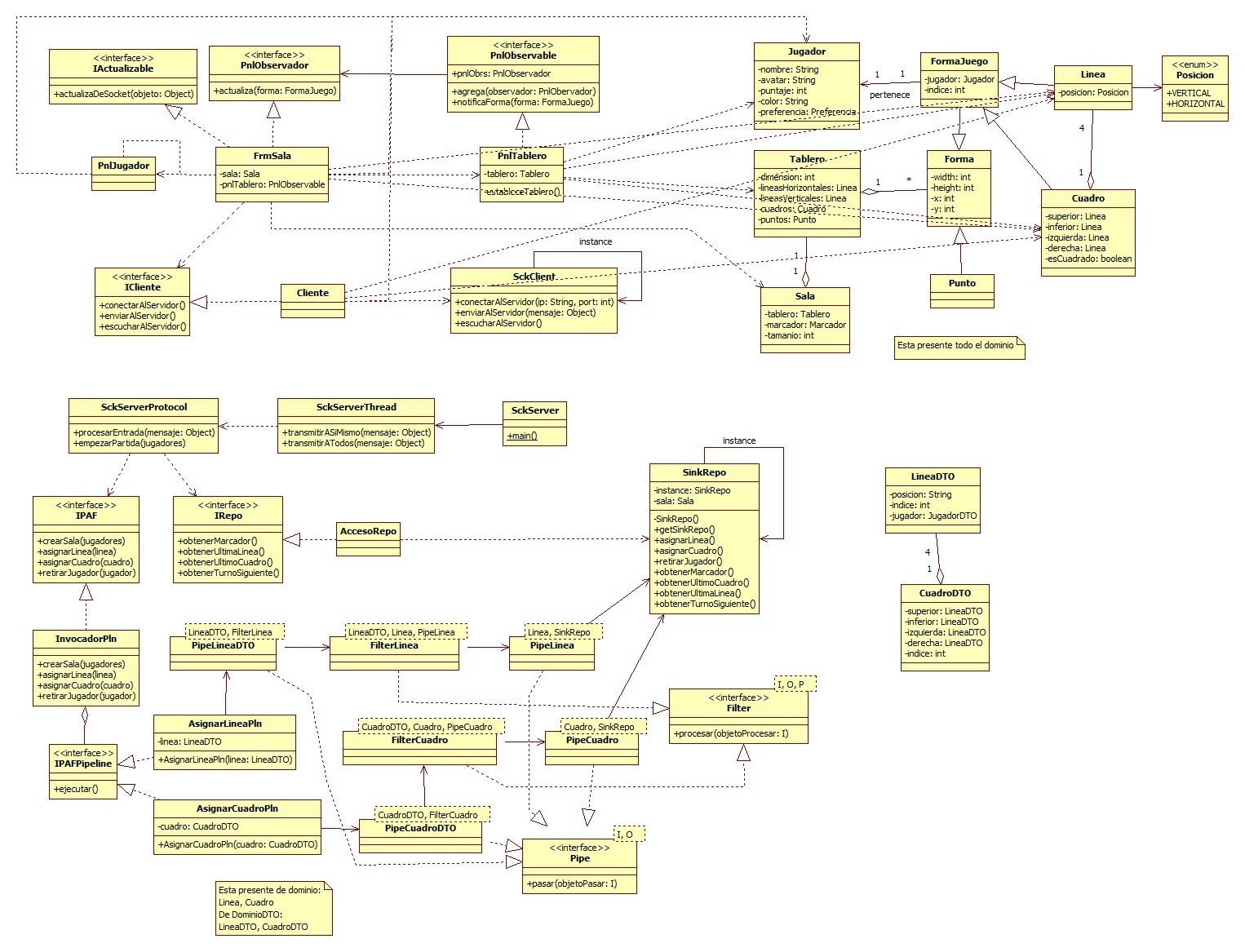
**Timbiriche**

### 6.3. Realización de casos de uso.

#### 6.3.1. CU Iniciar Juego

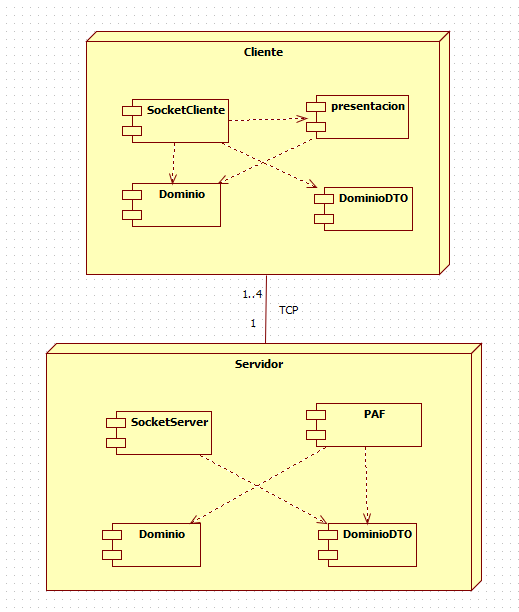


#### 6.3.2. CU Jugar



# 7. Vista de despliegue

### 7.1. Diagrama de despliegue.



En el diagrama de despliegue se diseño de tal forma que se dividió en 2 partes, una de estas partes es el servidor en el cual a través del uso e implementación de Sockets nos permite la interacción de un sistema el cual puede ser desplegado en uno de los clientes.

**Servidor:** Lo que ocurre dentro del servidor es que se llevan a cabo tareas de manejo de conexiones las cuales son de manera simultánea con los clientes por hilos, también la creación de objetos de tipo dominio y DTOdominio todo en un componente Pipes and Filters y usando un repositorio de sistema como un sink.

**Cliente:** En la parte del cliente es donde se despliega la interfaz del usuario y la conexión con el servidor mediante sockets.