3. Arquitectura

3.1 Apreciación Global

En esta sección se especificará la Arquitectura del Sistema que fue escogido para la implementación final de la aplicación. Antes de escoger la arquitectura adecuada es necesario evaluar las ventajas y desventajas de cada una y de esta forma determinar la arquitectura que prestará mayores funcionalidades de desempeño, disponibilidad (y otro atributos de calidad). La muestra un pequeño resumen de las principales características de los estilos arquitectónicos más utilizados.

Para nuestro proyecto de Sky Solution decidimos utilizar una arquitectura de cliente/servidor debido a que debemos hacer una conexión del usuario con el lugar donde estará registrada la información del juego, o mejor dicho la base de datos, será entonces una arquitectura basada en dos capas, donde el cliente hará una petición a el servidor, y este deberá darle la respectiva respuesta.

Decidimos utilizar esta arquitectura, ya que nos proporcionara separación de responsabilidades, la cual nos permitirá mayor centralización y claro un mejor diseño del sistema.

Nuestra arquitectura contara entonces con unas características principales tales como:

* Cuenta con dos tipos distintos de capas, que en este caso vendrían siendo los dos nodos diferentes del sistema, cliente como primero, y servidor el segundo tipo.
* Múltiple concurrencia de usuarios, debió a que existe una correcta distribución, estos podrán hacer sus peticiones de manera paralela y obtener una respuesta correcta.
* El cliente será quien inicie las solicitudes, estas deberán llegar al servidor, y cuando sean recibidas, este será el encargado de procesarlas y enviar la respuesta correspondiente.
* El aumentar la capacidad de clientes o servidores, será una característica valida del sistema, es decir que en cualquier momento, se podrá agregar uno de los dos tipos diferentes de nodo, a la red.

La lógica con la que funcionara esta arquitectura, será básicamente la narrada en las características anteriores, sin embargo para detallar más a fondo, veremos entonces detalles sobre el funcionamiento interno de esta arquitectura:

Como deberes o características del nodo cliente, encontramos además de las pasadas, también el esperar las respuestas de un servidor, y la capacidad de poderse conectar con varios de ellos a la vez, así como utilizar a una interfaz grafica final, como el medio de comunicación directo, sabiendo también que esta interfaz estará ubicada en el cliente, y no en el servidor, con el fin de que siendo el componente principal, no haya necesidad de cargarla a este desde el servidor, cada vez que se inicie nuestra aplicación, sino tenerla ubicada en el mismo lugar desde donde se ejecutara.

Para el servidor tendremos también que podrá recibir múltiples conexiones al tiempo, desde diferentes clientes, que en nuestro caso, por el momento, no estará limitado, y también contaremos con que el servidor, no se comunicara directamente, sino interactuara por medio de mecanismos como interfaces, descritas en el punto **6. Diseño de Interfaces de Usuario**, sabemos también que el servidor debe estar conectado al cliente, por medio de algún tipo de red.

*Básicamente toda aplicación tiene 3 módulos principales 1) Presentación que se encarga de mostrar la información, 2) Lógica el cual realiza el manejo interno de todos los datos y en general del proceso interno de la aplicación y 3) Datos al cual se le delega el almacenamiento, la modificación y la eliminación de la información guardada ya sea en un motor de bases de datos o en un sistema de archivos.*

*Para el proyecto 7 Texas Hold’em del periodo 2007-01 la representación de los módulos [3] (que en ese caso fueron capas, debido a la elección de la arquitectura) puede verse en la .*

En nuestro desarrollo de la aplicación WorDomination contamos con un gran número de características que se asemejan a la forma y lógica que maneja la arquitectura de cliente/servidor, así como un buen numero de problemas que pueden ser resueltos con esta misma.

Vemos entonces que nuestro problema presenta similitudes tales como:

* Interfaz grafica principal a disposición del cliente, ubicada precisamente en el lugar de ejecución o localización del cliente.
* Peticiones del cliente para que se resuelvan externamente, tales como calculo de puntajes sobre palabras, o persistencia de información sobre puntajes finales de los jugadores.
* Servidor central que cuente con los debidos datos para hacer posible la comunicación entre jugadores para cada partida.
* Múltiples jugadores conectados al mismo tiempo para poder iniciar una partida.

Después de esto vimos que las ventajas de utilizar esta arquitectura serian muy grandes, por lo cual tomamos en definitiva la decisión de utilizarla al desarrollar nuestra aplicación.

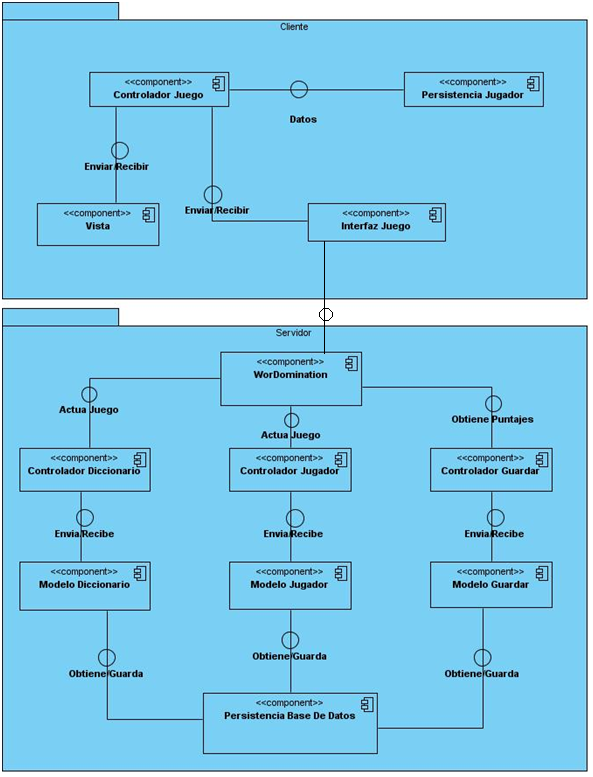
## Diagrama de Componentes

*Este diagrama presenta los componentes del sistema, sus interfaces y conectores (ilustración 9), incluyendo la organización y dependencia lógica entre ellos; además, se muestran los subsistemas que conforman la aplicación [4].*

*Una forma de explicar o documentar el diagrama de componentes es haciendo la división por subsistemas y para cada uno explicar los componentes e interfaces asociados. Posibles características que se pueden tener en cuenta en cada descripción son [5]:*

Ilustración : Diagrama de componentes

1. *Composición de los subsistemas: Indica los componentes que se encuentran en el subsistema.*
2. *Propósito de subsistemas y componentes: Describe por qué existen, su importancia y características generales.*
3. *Interfaces disponibles: Esta es sólo para los componentes e indica las interfaces que tiene cada uno disponibles para los demás, las cuales permiten realizar la comunicación entre dichos componentes y utilizar los servicios prestados por cada uno. Esta incluye:*
   1. *Nombre de la interfaz.*
   2. *Componentes que la utilizan.*
   3. *Servicios prestados.*
4. *Dependencias: Esta es sólo para componentes, muestra las interfaces y componentes de los que depende utilizando sus servicios. Esta incluye:*
   1. *Nombre de la interfaz.*
   2. *Componente que la ofrece (el “oferente” de la interfaz).*
   3. *Descripción de la dependencia.*



* + 1. Subsistema Servidor

|  |  |
| --- | --- |
| **Propósito del Subsistema** | *Con este subsistema buscamos brindar toda la parte de Interfaces que necesita el servidor para ser operado y actualizado, además de contar con toda la persistencia de datos, enviados desde nuestro subsistema cliente.*  *Este subsistema debe permitir también la consulta y el envió de datos almacenados en él, y de los distintos cálculos del juego.* |
| **Composición del Subsistema** | * Persistencia Base De datos * Modelo Diccionario * Modelo Jugador * Modelo Guardar * Controlador Diccionario * Controlador Jugador * Controlador Guardar * WorDomination |

* + - 1. Persistencia Base De datos

|  |  |  |  |
| --- | --- | --- | --- |
| **Propósito del**  **Componente** | *Guardar toda la información de puntuación acumulada sobre todos los jugadores registrados.* | | |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Salida/Entrada Datos* | *Modelo Diccionario*  *Modelo Jugador*  *Modelo Guardar* | *Nuestros modelos acudirán a este componente al momento de guardar la información de las partidas, y sumársela al acumulado que se lleva justo antes de que se invoque esta.* |
| **Dependencias** | *Nombre de la Interfaz* | *Componente que la ofrece* | *Descripción de la dependencia* |
| *---* | *---* | *---* |

* + - 1. Modelo Diccionario

|  |  |  |  |
| --- | --- | --- | --- |
| **Propósito del**  **Componente** | *El propósito de este componente es gestionar la interacción entre las partidas y las utilidades del diccionario que serán prestadas* | | |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Salida/Entrada De Datos* | *Persistencia Base De Datos* | *Nuestro Componente Modelo Diccionario utiliza la persistencia de la base de datos para intercambiar información correspondiente a* |
| **Dependencias** | *Nombre de la Interfaz* | *Componente que la ofrece* | *Descripción de la dependencia* |
| *Gestión* | *WorDomination* | *Necesitamos que WorDomination realice una correcta interacción para que así proporcione los datos correspondientes sobre la validación o acción del diccionario, al cliente* |

* + - 1. Modelo Jugador

|  |  |  |  |
| --- | --- | --- | --- |
| **Propósito del**  **Componente** | *El propósito de este componente es gestionar la interacción entre las partidas y las utilidades y los requerimientos del Jugador o Cliente* | | |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Salida/Entrada De Datos* | *Persistencia Base De Datos* | *Descripción de la utilidad o función de la interfaz mencionada. ¿Qué servicios presta a los componentes que la utilizan?* |
| **Dependencias** | *Nombre de la Interfaz* | *Componente que la ofrece* | *Descripción de la dependencia* |
| *Gestión* | *WorDomination* | *Descripción de la utilidad o función de la interfaz mencionada. ¿Qué servicios presta a este componente?* |

* + - 1. Modelo Guardar

|  |  |  |  |
| --- | --- | --- | --- |
| **Propósito del**  **Componente** | *Con este modelo buscamos persistir los datos de las partidas ejecutadas por los clientes.* | | |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Guardar Perfil* | *Persistencia* | *Se encarga de la comunicación entre el cliente y la persistencia a las bases, donde guardara la información necesitada en los requerimientos* |
| **Dependencias** | *Nombre de la Interfaz* | *Componente que la ofrece* | *Descripción de la dependencia* |
| *Guardar Perfil* | *WorDomination* | *Requiere de este modelo para poder hacer una correcta persistencia a la base, es el medio de comunicación o traducción para que la persistencia entienda al cliente.* |

* + - 1. WorDomination

|  |  |  |  |
| --- | --- | --- | --- |
| **Propósito del**  **Componente** | *Componente principal encargado de la gestión directa con el cliente y servir de camino a los demás modelos.* | | |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Gestión* | *Modelo Diccionario* | *Este componente nos permite la comunicación al Diccionario, donde le posibilita las palabras a ser consultadas* |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Gestión* | *Modelo Jugador* | *Este componente nos permite el envió de mensajes de consulta de un Jugador, así como sus debidas respuestas.* |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Comunicación* | *Interfaz Juego* | *Nos da la comunicación más directa entre cliente y servidor, debido a que contacta estos dos puntos a través de funciones de nuestro sistemas y subdivididas en los distintos componentes.* |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Guardar Perfil* | *Modelo Guardar* | *Lleva por este camino los datos suficientes para poder ser guardados y mas adelante tener la completa persistencia.* |
| **Dependencias** | *Nombre de la Interfaz* | *Componente que la ofrece* | *Descripción de la dependencia* |
| *---* | *---* | *---* |

* + 1. Subsistema Cliente

|  |  |
| --- | --- |
| **Propósito del Subsistema** | *Este subsistema como primera instancia es el encargado de dar al cliente todas las interfaces que este necesite, ya sea tanto para ejecutar el juego, como para consultar datos anteriores (puntuaciones) de algún jugador.*  También está encargado de mostrar a cada jugador como se van desempeñando las jugadas de los otros competidores, o mejor dicho de ir actualizando el juego de todos los jugadores cada vez que este cambie. |
| **Composición del Subsistema** | * *Controlador Juego* * *Persistencia Jugador* * *Interfaz Juego* * *Vista* |

* + - 1. *Controlador Juego*

|  |  |  |  |
| --- | --- | --- | --- |
| **Propósito del**  **Componente** | *Este componente se encarga de mantener actualizado el juego así como de prestar los servicios necesarios para que todas las pantallas se vayan refrescando al tiempo que se modifique cualquier detalle de la partida.* | | |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Actualizar* | *Modelo* | *Envía la información necesaria sobre el estado actual de la partida, para que esta se vea reflejada en las pantallas de todos los jugadores* |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componente que la utilizan* | *Servicios prestados* |
| *Salida/Entrada Datos* | *Persistencia Jugador* | *Gracias a la persistencia se podrá descifrar la información grafica de la aplicación, y así poder utilizar esto para las debidas acciones pedidas por el cliente* |

* + - 1. *Persistencia Jugador*

|  |  |  |  |
| --- | --- | --- | --- |
| **Propósito del**  **Componente** | *Con este componente buscamos obtener y acceder a la información relacionada con las jugadas ejecutadas por el jugador, para poder continuar con el procesamiento del juego.* | | |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *--* | *--* | *--* |
| **Dependencias** | *Nombre de la Interfaz* | *Componente que la ofrece* | *Descripción de la dependencia* |
| *--* | *--* | *--* |

* + - 1. *Interfaz Juego*

|  |  |  |  |
| --- | --- | --- | --- |
| **Propósito del**  **Componente** | *Es la interfaz que se encargara de hacer la respectiva comunicación directa a el servidor, y la respectiva traducción a los demás componentes del cliente, de esta forma gestionara el envió y la recepción de mensajes entre ellos.* | | |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Comunica Cliente* | *WorDomination* | *Transmite los diferentes mensajes y claves entre el cliente y el servidor* |
| **Dependencias** | *Nombre de la Interfaz* | *Componente que la ofrece* | *Descripción de la dependencia* |
| *Actualizar* | *Controlador Juego* | *Es necesario controlar la misma información para todos los jugadores, en el mismo momento, para que no se produzcan incoherencias, por lo cual es necesario conocer la información por parte del cliente.* |

* + - 1. *Vista*

|  |  |  |  |
| --- | --- | --- | --- |
| **Propósito del**  **Componente** | *Sera el contacto directo al cliente, quien se encargue de gestionar el correcto orden entre los diferentes clientes, y de mostrar la información correcta del desarrollo del juego a ellos, así como tomar las distintas peticiones de estos.* | | |
| **Interfaces Disponibles** | *Nombre de la Interfaz* | *Componentes que la utilizan* | *Servicios prestados* |
| *Actualizar* | *Controlador Juego* | *Se encargara de todas las Interfaces o vistas que se proporcionan al usuario para que este pueda interactuar de manera sencilla con la aplicación, interactuara con el controlador por medio de Actualizar, para que así las vistas estén correctamente actualizadas a través del tiempo.* |
| **Dependencias** | *Nombre de la Interfaz* | *Componente que la ofrece* | *Descripción de la dependencia* |
| *--* | *--* | *--* |