**[Año]**

**[2009]**

ANDRÉS MARÍN

ALEXANDRA ARDILA

KAREM ADRIANA MORENO

[Versión 3.0]

[4 DE NOVIEMBRE de 2009]

**[WORDOMINATION]**

**[SKY SOLUTION]**

[Equipo SKY SOLUTION]

**[DESCRIPCIÓN DE DISEÑO DE SOFTWARE]**

[Equipo SKY SOLUTION]



**PÁGINA DE FIRMAS**

El presente documento es aprobado por las personas referenciadas a continuación:

**Firma Clientes:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­­­­­\_\_\_\_\_**

JUAN CARLOS ALDANA MIGUEL TORRES

**Firma Director de Proyecto:**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ANDRÉS ROGELIO MARÍN CASTELBLANCO

**Firma Director de Calidad y Manejo de Riesgos:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

ALEXANDRA ARDILA SARMIENTO

**Firma Administrador de Configuración y Documentación:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

KAREM ADRIANA MORENO PACHECO

**Firma Arquitectos:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

ALEXANDRA ARDILA SARMIENTO  KAREM ADRIANA MORENO PACHECO

**Firma Director de Desarrollo:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

ANDRÉS ROGELIO MARÍN CASTELBLANCO

HISTORIAL DE CAMBIOS

En esta sección se presenta una tabla que describe la evolución y los cambios que se le realizan al documento desde que se inicia hasta que se haya llegado a la versión base que es entregada al cliente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Versión** | **Fecha**  **Entrega** | **Sección del documento modificada** | **Descripción de cambios (corta)** | **Responsable (S)** |
| 1.0 | 21 de Octubre | Diseño Plantilla  SDD v1.0 | Realización del diseño y modificación | **Karem Moreno** |
| 1.0 | 21 de Octubre | Consideraciones de diseño, secciones: 2.1,2.2,2.3,2.4,2.5 | Realización de estas secciones | **Karem Moreno** |
| 1.1 | 24 de Octubre | Consideraciones de diseño, secciones: 2.1,2.2,2.3,2.4,2.5 | Realización calidad de estas secciones | **Andrés Marín** |
| 1.0 | 21 de Octubre | Arquitectura, secciones: 3.1,3.2,3.2.1,3.2.1.1 | Realización de estas secciones | **Alexandra Ardila** |
| 1.1 | 24 de Octubre | Arquitectura, secciones: 3.1,3.2,3.2.1,3.2.1.1 | Realización calidad de estas secciones | **Karem Moreno** |
| 2.0 | 31 de Octubre | Diseño de alto nivel, secciones : 4.1,4.1.1,4.1.2 | Realización de estas secciones | **Karem Moreno** |
| 2.0 | 31 de Octubre | Introducción, secciones: 1.1,1.2,1.3,1.4 | Realización de estas secciones | **Alexandra Ardila** |
| 1.0 | 2 de Noviembre | Diagrama de clases | Realización del diagrama con su documentación | **Alexandra Ardila**  **Andrés Marín** |
| 4.0 | 3 de Noviembre | Prototipo IV  (Justificación) | Entrega del Prototipo | **Andrés Marín** |

Tabla : Historial de cambios

Contenido

[Lista de Tablas 4](#_Toc180071410)

[Lista de Ilustraciones 5](#_Toc180071411)

[1. Introducción 6](#_Toc180071412)

[1.1 Descripción del Sistema 6](#_Toc180071413)

[1.2 Mapa del Diseño 7](#_Toc180071414)

[1.3 Referencias y Documentos de Apoyo 7](#_Toc180071415)

[1.4 Definiciones, Acrónimos y Abreviaciones 8](#_Toc180071416)

[2. Consideraciones de Diseño 11](#_Toc180071417)

[2.1 Suposiciones 11](#_Toc180071418)

[2.2 Restricciones 11](#_Toc180071419)

[2.3 Entorno del Sistema 12](#_Toc180071420)

[2.4 Metodología de Diseño 13](#_Toc180071421)

[2.5 Riesgos 14](#_Toc180071422)

[3. Arquitectura 15](#_Toc180071423)

[3.1 Apreciación Global 16](#_Toc180071424)

[3.2 Diagrama de Componentes 17](#_Toc180071425)

[3.2.1 Subsistema 1 19](#_Toc180071426)

[3.2.1.1 Componente 1 19](#_Toc180071427)

[3.3 Estrategias de Diseño 19](#_Toc180071428)

[4. Diseño de Alto Nivel 21](#_Toc180071429)

[4.1 Diagrama de despliegue 21](#_Toc180071430)

[4.1.1 Nodo 1 *“Nombre del Nodo”* 23](#_Toc180071431)

[4.1.2 Conector 1 “Nombre del Nodo” 23](#_Toc180071432)

[4.2 Diagrama de Comportamiento e Interacción 23](#_Toc180071433)

[4.2.1.1 Diagrama de Actividad 24](#_Toc180071434)

[4.2.1.1 Diagrama de Secuencia 24](#_Toc180071435)

[5. Diseño de Bajo Nivel 26](#_Toc180071436)

[5.1 Subsistema 1 29](#_Toc180071437)

[5.1.1 Componente 1 29](#_Toc180071438)

[6. Diseño de Interfaces de Usuario 32](#_Toc180071439)

[6.1. Diseño general de la aplicación 32](#_Toc180071440)

[6.2. Árbol de navegabilidad 33](#_Toc180071441)

[Anexos 34](#_Toc180071442)

Lista de Tablas

[**Tabla 1**: Historial de cambios 1](#_Toc180071443)

[**Tabla** **2:** Ventajas y Desventajas de Estilos Arquitectónicos 16](#_Toc180071444)

[**Tabla** **3:** Subsistema 1 19](#_Toc180071445)

[**Tabla** **4:** Componente 1 19](#_Toc180071446)

[**Tabla** **5:** Estrategias de Diseño 20](#_Toc180071447)

[**Tabla** **6:** Nodo 1 23](#_Toc180071448)

[**Tabla** **7:** Conector 1 23](#_Toc180071449)

[**Tabla** **8:** Documentación de clases 29](#_Toc180071450)

[**Tabla** **9:** Descripción de entradas y salidas 32](#_Toc180071451)

Lista de Ilustraciones

[Ilustración 1: Descripción del sistema 6](#_Toc180071452)

[Ilustración 2: Mapa de Diseño 7](#_Toc180071453)

[Ilustración 3: Suposiciones generales 11](#_Toc180071454)

[Ilustración 4: Restricciones 12](#_Toc180071455)

[Ilustración 5: Entorno del Sistema 13](#_Toc180071456)

[Ilustración 6: Administración de Riesgos 14](#_Toc180071457)

[Ilustración 7: Apreciación Global 16](#_Toc180071458)

[Ilustración 8: Capas Arquitectónicas 17](#_Toc180071459)

[Ilustración 9: Diagrama de componentes 17](#_Toc180071460)

[Ilustración 10: Subsistema Usuario 7 Texas Poker 18](#_Toc180071461)

[Ilustración 11: Estrategias de Diseño 20](#_Toc180071462)

[Ilustración 12: Descripción de Nodos 21](#_Toc180071463)

[Ilustración 13: Descripción Conexiones 22](#_Toc180071464)

[Ilustración 14: Nodo Usuario 7 Texas Poker 22](file:///C:\Documents%20and%20Settings\Juan%20David\Desktop\Proyecto%20Especial\SDD\SDD%201.0.docx#_Toc180071465)

[Ilustración 15: Diagrama de Actividad 24](#_Toc180071466)

[Ilustración 16: Diagrama de secuencia 25](#_Toc180071467)

[Ilustración 17: Diseño de bajo nivel 26](#_Toc180071468)

[Ilustración 18: Niveles de diseño 27](#_Toc180071469)

[Ilustración 19: Ejemplo diseño de bajo nivel 28](#_Toc180071470)

[Ilustración 20: Ejemplo Descripción de Clase 29](#_Toc180071471)

[Ilustración 21: Ejemplo Resumen de Métodos 30](#_Toc180071472)

[Ilustración 22: Ejemplo descripción de método 31](#_Toc180071473)

[Ilustración 23: Diseño general de la aplicación 32](#_Toc180071474)

[Ilustración 24: Ejemplo árbol de navegabilidad 33](file:///C:\Documents%20and%20Settings\Juan%20David\Desktop\Proyecto%20Especial\SDD\SDD%201.0.docx#_Toc180071475)

# Introducción

## Descripción del Sistema

De acuerdo a los anteriores documentos de nuestro proceso de desarrollo del producto ***WorDomination***, y en necesidad de resolver lo planteado, desarrollamos entonces este documento, que nos narrará de una forma más precisa acerca de la arquitectura en alto y bajo nivel, planteados y utilizados por los diferentes integrantes del equipo **SKY SOLUTION**, para resolver y satisfacer todos los requerimientos funcionales y no funcionales de el producto en cuestión, y dar así la mayor satisfacción al cliente.

Narraremos y mostraremos entonces, cómo y por qué se tomo la decisión de utilizar una arquitectura cliente-servidor, en la cual cada uno de estos subsistemas se dividirá en distintos componentes, y como al ser estos integrados, darán como resultado el cumplimiento de las debidas labores de cada uno de los subsistemas, que a su vez, al ser también integrados resolverán todas las funciones y peticiones de los jugadores y clientes en ***WorDomination***, al tiempo que permitirán también una mayor organización para los desarrolladores, y la posibilidad de reutilización de código, o modificaciones del programa.



Ilustración : Descripción del sistema

## Mapa del Diseño

Se decidió dividir en dos grupos principales los distintos esquemas de nuestro documento, los cuales se muestra en la siguiente gráfica; cada uno con sus diferentes componentes.

* Diseño de alto nivel donde se incluye:
* Estilo arquitectónico
* Diagrama de despliegue
* Diagrama de componentes
* Diagramas de actividad y de secuencia
* Diseño de bajo nivel:
* Diagramas de clase
* Diagrama entidad relación



Ilustración : Mapa de Diseño

## Referencias y Documentos de Apoyo

***[1]*** *Bruegge B, Dutoit AH. Ingeniería de Software orientada a objetos. 1st ed. Trujano G.México: Pearson Educación; 2002*

***[2]*** *Ian Sommerville. Ingeniería de software. Séptima edición. Pearson Education.* ***2005***

***[3]*** *Wiegers, Karl. , Sample requirements documents. Process Goodies 2002, Disponible en* [*http://www.processimpact.com/goodies.shtml*](http://www.processimpact.com/goodies.shtml)

***[4]*** *Larman Craig. UML y Patrones, Una introducción al análisis y diseño orientado a objetos y al proceso Unificado. 2st ed. Madrid, España. Pearson, Prentice Hall Education. 2002*

***[5]*** *Tortoise SVN, Repositorio de Archivos. Disponible en:*

*http://www.tortoisesvn.tigris.org/*

***[6]*** *Daniele Marcela.Teoria\_11\_UML\_componentes e interfaces.pdf. UNRC. 2007*

***[7]*** *Construx Software, Software Requirements Specification, “Cx CheckList Requierements”. CXOne Standard, Construx Software Builder, Inc, Noviembre 2002.*

***[8]*** *Scrabble. Disponible en:*

[*http://www.mattelscrabble.com/es/adults/history/index.html*](http://www.mattelscrabble.com/es/adults/history/index.html)

***[9]*** *Asociación Uruguaya de Scrabble. Abril 13 de 2009, disponible en:*

[*http://www.scrabbel.org.uy/reglas/reglas.htm*](http://www.scrabbel.org.uy/reglas/reglas.htm)

***[10]***  *Página de Miguel Torres [homepage de Internet]. Bogotá. Ing. Miguel Eduardo Torres Moreno MSc. Copyright - Miguel Torres 2008. [Actualizado el 2008]. Materias Ingeniería de Software(2009-3) – Plantilla SDD V1.0 (Línea Base), IRONWORKS. Disponible en:* [*http://sophia.javeriana.edu.co/~metorres/*](http://sophia.javeriana.edu.co/~metorres/)

***[11]*** *Diccionario de la Real Academia Española. Disponible en: http://www.rae.es/rae.html*

***[12]*** *Página de Miguel Torres [homepage de Internet]. Bogotá. Ing. Miguel Eduardo Torres Moreno MSc. Copyright - Miguel Torres 2008. [Actualizado el 2008]. Materias Análisis y diseño de objetos – Plantilla Casos de uso. Disponible en:*

[*http://sophia.javeriana.edu.co/~metorres/*](http://sophia.javeriana.edu.co/~metorres/)

***[13]*** *Página de Miguel Torres [homepage de Internet]. Bogotá. Ing. Miguel Eduardo Torres Moreno MSc. Copyright - Miguel Torres 2008. [Actualizado el 2008]. Materias Análisis y diseño de objetos –Diapositivas Introducción a UML. Disponible en:*

[*http://sophia.javeriana.edu.co/~metorres/*](http://sophia.javeriana.edu.co/~metorres/)

***[14]*** *Página de Miguel Torres [homepage de Internet]. Bogotá. Ing. Miguel Eduardo Torres Moreno MSc. Copyright - Miguel Torres 2008. [Actualizado el 2008]. Materias Análisis y diseño de objetos – Diapositivas Diagrama de clases. Disponible en:*

[*http://sophia.javeriana.edu.co/~metorres/*](http://sophia.javeriana.edu.co/~metorres/)

***[15]*** *Página de Miguel Torres [homepage de Internet]. Bogotá. Ing. Miguel Eduardo Torres Moreno MSc. Copyright - Miguel Torres 2008. [Actualizado el 2008]. Materias Análisis y diseño de objetos – Diapositivas Secuencia y colaboración. Disponible en:*

[*http://sophia.javeriana.edu.co/~metorres/*](http://sophia.javeriana.edu.co/~metorres/)

***[16]*** *Página de Miguel Torres [homepage de Internet]. Bogotá. Ing. Miguel Eduardo Torres Moreno MSc. Copyright - Miguel Torres 2008. [Actualizado el 2008]. Materias Análisis y diseño de objetos – GRASP.pdf. Disponible en:*

[*http://sophia.javeriana.edu.co/~metorres/*](http://sophia.javeriana.edu.co/~metorres/)

***[17]*** *Página de Miguel Torres [homepage de Internet]. Bogotá. Ing. Miguel Eduardo Torres Moreno MSc. Copyright - Miguel Torres 2008. [Actualizado el 2008]. Materias Análisis y diseño de objetos – Design Patterns.GoF. Disponible en:*

[*http://sophia.javeriana.edu.co/~metorres/*](http://sophia.javeriana.edu.co/~metorres/)

***[18]*** *IEEE (Institute of Electrical and Electronics Engineers), ), IEEE Recommended Practice* for Software Design Descriptions*, IEEE-SA Standards Board, Junio 1998.*

***[19]*** *Arquitectura cliente – servidor, disponible en:*

[*http://www.csae.map.es/csi/silice/Global71.html*](http://www.csae.map.es/csi/silice/Global71.html)

***[20]*** *Larry Ullman, MySQL, 2nd Edition.* [*http://www.mysql.com/downloads/*](http://www.mysql.com/downloads/)

***[21]*** *Java SE Technologies – Java Database Connectivity (JDBC) [homepage de Internet]. Copyright 1994-2009 Sun Microsystems, Inc. Disponible en:* [*http://java.sun.com/javase/technologies/database/index.jsp*](http://java.sun.com/javase/technologies/database/index.jsp)

**[22]** *Introduction to TCP/IP [homepage de Internet]. Copyright 1995 PCLT. Disponible en:* [*http://www.yale.edu/pclt/COMM/TCPIP.HTM*](http://www.yale.edu/pclt/COMM/TCPIP.HTM)

**[23]** *MySQL AB :: Developer Zone [homepage de Internet]. © 1995-2008 MySQL AB. [citado 2008 - 2009]. Disponible en:http://www.mysql.com*

**[24]**Ing. Evans Balcázar Veizaga. *Memoria Persistente.* Universidad Autónoma Gabriel René Moreno.

**[25]** Barcia, N.; Fernández, C.; Frutos, S; López, G.; Mengual, L.; Riano, F. J.; Yagüez, F. J. *Redes de computadores y arquitecturas de comunicaciones*. Prentice Hall. Madrid; 2006.

**[26]** Definición Arquitectura cliente – servidor, disponible en:

<http://www.csae.map.es/csi/silice/Global71.html>

**[27]** *Andrea Barraza, Luisa Barrera, Natalia Bejarano, José Guzmán, Oscar López, Saúl Martínez, DOCUMENTO DE DISEÑO DE SOFTWARE – SDD. Pontificia Universidad Javeriana. Año 2009.*

## Definiciones, Acrónimos y Abreviaciones

En esta sección mostraremos aquellas definiciones de palabras y de acrónimos que serán utilizados en el documento con el fin de aclarar el contenido y asegurar su óptimo entendimiento.

|  |  |
| --- | --- |
| **DEFINICIONES y ACRÓNIMOS** | |
| API | (Aplication Programming Interface) Interfaz de Programación de Aplicaciones, es un conjunto de especificaciones de comunicación entre componentes software. |
| Aplicación WorDomination | Es la aplicación completa, la cual es ejecutada desde el computador del jugador |
| Armar | Construir una palabra en el tablero de juego |
| commit | Instrucción de manejo de bases de datos que guardad todos los cambios realizados en ella. |
| Cronómetro | Aquel que llevará el tiempo restante del jugador que tiene el turno actual, para completar su jugada. |
| CRUD | Create, Retrieve, Update, Delete; se refiere sobre todo a las operaciones en bases de datos. |
| DBMS | (DATA BASE MANAGEMENT SYSTEM): sistema dedicado a servir de interfaz entre las bases de datos y las aplicaciones que las utilizan*.* |
| Dinámico | Que se va reflejando casi al mismo momento en que va sucediendo. |
| Ficha | Componente del juego conformada por un letra del abecedario asignada por el juego. |
| framework | Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, librerías y un lenguaje de scripting entre otros. |
| Ganador | Aquel jugador que al final de la partida, haya conseguido el mayor puntaje. |
| GMail | Servicio gratuito de correo electrónico que brinda Google por Internet, el cual está disponible en la página web: <http://mail.google.com> |
| GNU | Sistema operativo totalmente libre es compatible con la arquitectura UNIX. |
| GPL | Licencia de Software usada por la mayoría de programas de GNU y muchas aplicaciones de Software libre. |
| GUI | En el contexto del proceso de interacción persona-ordenador, la interfaz gráfica de usuario (IGU), provee una forma amigable de interacción con un sistema informático. |
| Interfaz | Parte de una aplicación para permitir el flujo de información entre la aplicación y el usuario. |
| *JDBC* | *Java DataBase Connectivity* |
| JRE | *Java Runtime Environment* |
| Jugador | Es aquel usuario registrado que se encuentre activo dentro de una ***partida***. |
| JVM | *Java Virtual Machine* |
| LAN | *Local Area Network*; Red de área local, conexión entre varios ordenadores, en un área determinada. |
| Loguear | Escribir nombre y contraseña en el sistema para poder ingresar como jugador. |
| Microsoft Project | Herramienta de software desarrollado por Microsoft, especializado en la gestión de proyectos, en lo referente planeación y gestión de recursos humanos. |
| Microsoft Office | Conjunto de aplicaciones de software para oficina que ha desarrollado Microsoft, en donde se destacan los siguientes productos: Microsoft Access, Microsoft PowerPoint, Microsoft Word y Microsoft Excel. |
| Modelo ER | Modelo Entidad Relación |
| Orden Turno | Orden sucesivo en el que ejecutarán las jugadas cada uno de los jugadores, dependiendo del orden de llegada. |
| Partida | Es el conjunto de turnos que los jugadores pueden realizar, en el cual, se llevarán a cabo los diferentes movimientos de las fichas y que al final se elegirá u ganador. |
| Persistencia | Guardar información de datos para poder reutilizarlos una vez se salga del sistema. |
| Primary Key | Llave primaria para la base de datos, que permitirá de manera única identificar una tupla. |
| Progressbar | Barra para medir el progreso de una tarea o proceso en el programa. |
| Puntaje | Los puntos acumulados por un jugador en una partida. |
| Puntuación | Los puntos acumulados por un jugador a través de todas sus partidas jugadas. |
| RFC | Serie de notas sobre Internet, donde cada una de ellas es un documento con un protocolo de Internet sin ambigüedades. |
| Ronda | Vuelta de un turno de todos los jugadores presentes en el juego. |
| Scrabble | Juego original en el cual se baso WorDomination, para su construcción. |
| SDD | Descripción de Diseño de Software. |
| SPMP | Documento que plantea los planes a seguir en un proyecto de software. El SPMP define las actividades, productos de trabajo, indicadores de avance y recursos asignados al proyecto. También están definidos los procedimientos administrativos y convenciones aplicables al proyecto, como el reporte de estado, la gestión de riesgos, y la administración de configuración y planes para llevar el control y aseguramiento de la calidad. |
| SRS | Especificación de requerimientos de Software. |
| SKY SOLUTION | Equipo de trabajo cuyos integrantes son: Andrés Marín, Alexandra Ardila y Karem Moreno, quienes son estudiantes de la asignatura Ingeniería de Software, de la facultad de Ingeniería de la Pontificia Universidad Javeriana. |
| SQL | *Structured Query Language* |
| Stakeholder | Cualquier persona relacionada directa o indirectamente con el uso del sistema, cualquiera a quien afecte o lo utilice en algún momento. |
| SVN | Es un sistema de control de versiones que se utiliza para llevar la administración de la configuración en los documentos y código fuente. Para mayor información, puede consultar la siguiente página web: <http://subversion.tigris.org/> |
| *Tablero* | Espacio de 15x15 cuadros o celdas, en la cual se irán organizando las diferentes letras que a su vez armarán las palabras. |
| *TCP/IP* | Protocolo de red común utilizado para que todos los ordenadores conectados a Internet puedan comunicarse entre si, es compatible con cualquier sistema operativo. |
| TortoiseSVN | Herramienta de software que se utiliza para el control y gestión de versiones, el cual permite administrar los documentos y el código fuente de un proyecto de software que se esté desarrollando y utilice un repositorio SVN. |
| Transacción | Conjunto de operaciones que conforman una unidad de trabajo. |
| Transacción Atómica | Transacción donde se garantiza se han cumplido todos los pasos o ninguno. |
| **Turno** | Momento en el cual el jugador podrá hacer su respectiva jugada |
| UDP | Es un protocolo que no esta orientado a la conexión de la capa de transporte, no detecta errores. |
| Usuario Anónimo | Aquel que todavía no está registrado en el sistema y por lo tanto hará su previa inscripción para poder acceder al sistema. |
| Usuario Registrado | Aquel que haya hecho previo registro de sus datos, nombre y contraseña, en el sistema |
| ventana de Registro | Es la ventana donde un usuario anónimo, intentará registrarse con su nombre o nick, y contraseña. |
| ventana mesa de jugadores | Es la ventana siguiente a la del login y permite ir sentando a los jugadores mientras van llegando. |
| Ventana WorDomination | Es la ventana donde se juega y arman las palabras según el turno de cada jugador |
| WLAN | *Wireless Local Area* Network, Es un sistema de comunicación de datos inalámbrico, utilizado como alternativa a las redes de área local cableadas o como extensión de éstas. Utiliza tecnología de radiofrecuencia que permite mayor movilidad a los usuarios al minimizar las conexiones cableadas. |
| WorDomination | Es el nombre del juego como del proyecto, que se va a realizar. |
| ***802.11*** | *El protocolo IEEE 802.11 o WI-FI es un estándar de protocolo de comunicaciones de la IEEE que especifica las normas de funcionamiento en una WLAN.* |

Tabla - Definiciones, Acrónimos y abreviaturas

# Consideraciones de Diseño

## Suposiciones

Se identifican tres grupos de suposiciones que serán utilizados en la aplicación de **SKY SOLUTION**:

**El proyecto**

* Las suposiciones del proyecto han sido señaladas en el documento SKYSOLUTION\_SPMP\_v3.3 sección 1.1.2

**El sistema y equipo de desarrollo**

* Las suposiciones del producto han sido señaladas en el documento SKYSOLUTION\_SRS\_v5.0 sección 2.6
* Las perspectivas del producto han sido señaladas en el documento SKYSOLUTION\_SRS\_v5.0 sección 2.1 con la sección 3.1 Requerimientos de interfaces externas.
* Cuenta con las herramientas de hardware y software necesarias para llevar a cabo el proceso de diseño e implementación del sistema.

**El cliente**

* Las suposiciones del cliente-usuario han sido señaladas en el documento SKYSOLUTION\_SRS\_v5.0 sección 2.3
* Las suposiciones del proyecto han sido señaladas en el documento SKYSOLUTION\_SPMP\_v3.3 sección 1.1.2

## Restricciones

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| * ***WorDomination*** estará disponible sólo en idioma español. * ***WorDomination*** no tendrá tolerancia a fallos. * El tiempo para el desarrollo del proyecto será de 4 meses. * ***WorDomination*** deberá funcionar en la sala de Ingeniería de Sistemas primer piso. * Restricciones del PROYECTO ver documento SKYSOLUTION\_SPMP\_ v3.3 sección 1.1.2 * Restricciones del PRODUCTO ver documento SKYSOLUTION\_SRS\_ v5.0 sección 2.4 * Restricciones del DISEÑO ver documento SKYSOLUTION\_SRS\_ v5.0 sección 3.4 | * *Los usuarios deben cumplir con las siguientes características ver documento*  *SKYSOLUTION\_SRS\_ v5.0 sección 2.3.* | * *Las restricciones de software deben cumplir con la sección 2.1.4 y 3.1.3 del documento*  *SKYSOLUTION\_SRS\_ v5.0 .* * *Las restricciones de memoria deben cumplir con la sección 2.1.6 del documento*  *SKYSOLUTION\_SRS\_ v5.0 .* * *Las restricciones de bases de datos deben cumplir con la sección 3.6 del documento*  *SKYSOLUTION\_SRS\_v5.0 .* * *Las interfaces con el usuario deben cumplir con la sección 2.1.2 del documento*  *SKYSOLUTION\_SRS\_v5.0 .* * Restricciones del PRODUCTO ver documento SKYSOLUTION\_SRS\_v5.0 sección 2.4 | * *Las restricciones de hardware deben cumplir con la sección 2.1.3 y 3.1.2 del documento SKYSOLUTION\_SRS\_ v5.0 .* * *Requerimientos del desempeño deben cumplir con la sección 3.3 del documento SKYSOLUTION\_SRS\_v5.0 .* * *Las restricciones de comunicación deben cumplir con la sección 2.1.5 del documento SKYSOLUTION\_SRS\_v5.0 .* * Restricciones del PRODUCTO ver documento SKYSOLUTION\_SRS\_v5.0 sección 2.4 |

En la siguiente ilustración se muestran las restricciones.

Ilustración : Restricciones

## Entorno del Sistema

Se describe el entorno a nivel de software y hardware con el que interactuará el sistema es el siguiente:

* **Software**
  + Es un producto nuevo (ver SKYSOLUTION\_SRS\_v5.0 sección 2.1.1)
* **Hardware:**
  + Deberá cumplir con las siguientes características de hardware (ver SKYSOLUTION\_SRS\_v5.0 sección 2.1.3)



* + Deberá cumplir con las siguientes Interfaz con el usuario (ver SKYSOLUTION\_SRS\_v5.0 sección 2.1.2)
  + Deberá cumplir con las características de memoria (ver SKYSOLUTION\_SRS\_v5.0 sección 2.1.6)



* + Deberá cumplir con las características del servidor (ver SKYSOLUTION\_SRS\_v5.0 sección 2.1.4)



* Deberá cumplir con las características del conexión (ver SKYSOLUTION\_SRS\_v5.0 sección 2.1.5)

Ilustración 3: Entorno del Sistema

## Metodología de Diseño

La metodología usada para el diseño del software es basada en la Orientada a objetos como se muestra a continuación:

|  |  |
| --- | --- |
| ***REQUERIMIENTOS*** | *METODOLOGIAS* |
| * *Recolección de requerimientos* * *Análisis y especificación de requerimientos* * *Identificar casos de uso* |
| *IMPORTANCIA* |
| *Permite obtener todas aquellas características que exige el cliente para el desarrollo del producto.* |
| *JUSTIFICACIÓN DE USO* |
| *Se debe realizar para mostrar más específico el sistema y cuales requerimientos deben cumplirse para cumplir con el producto a desarrollar.* |
| ***DISEÑO GENERAL*** | *METODOLOGIAS* |
| * *Diagrama de casos de uso* * *Modelo del dominio* * *Modelo de clases* * *Identificación de la arquitectura* |
| *IMPORTANCIA* |
| *Permite identificar aquellos objetos importantes que harán que el sistema funcione al llegar hacer desarrollados.* |
| *JUSTIFICACIÓN DE USO* |
| *Se debe realizar para tener un correcto orden de diseño permitiendo obtener las características más importantes al desarrollar el sistema.* |
| ***DISEÑO ESPECIFICO*** | *METODOLOGIAS* |
| * *Modelos más detallados de los realizados anteriormente.* * *Documentación casos de uso* * *Documentación del Modelo del dominio* * *Documentación de la arquitectura realizada.* |
| *IMPORTANCIA* |
| *Permite que se detalle mejor los modelos que se van a usar para guía de implementación y se sepa a que se refiere cada uno.* |
| *JUSTIFICACIÓN DE USO* |
| *Se debe realizar para conocer con detalle como debe ser el proceso de desarrollo basándose con los diagramas creados.* |
| ***PRUEBAS E IMPLEMENTACION*** | *METODOLOGIAS* |
| * *Implementación de acuerdo a los modelos realizados.* * *Prueba unitarias* * *Prueba del sistema* |
| *IMPORTANCIA* |
| *Permite establecer el código fuente del producto que se esta desarrollando y solucionar los errores que se presentan en este.* |
| *JUSTIFICACIÓN DE USO* |
| *Se debe realizar para cumplir lo desarrollado con las características pedidas por el cliente.* |

Tabla - Metodología de Diseño

## Riesgos

Los riesgos presentados en el proceso del proyecto se presentan en el documento SKYSOLUTION\_SPMP\_v3.3 sección 5.4. Para el control de requerimientos los podemos encontrar en el documento SKYSOLUTION\_ SPMP\_v3.3 sección 5.3.1

# Arquitectura

## Apreciación Global

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 al servidor, y este deberá darle la respectiva respuesta. Decidimos utilizar esta arquitectura, ya que nos proporcionará separación de responsabilidades, la cual nos permitirá mayor centralización y claro un mejor diseño del sistema.

Nuestra arquitectura contará 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, éstas 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 válida 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 funcionará 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 gráfica 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 ejecutará.

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 comunicará directamente, sino interactuará 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.

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 número de problemas que pueden ser resueltos con esta misma.

Vemos entonces que nuestro problema presenta similitudes tales como:

* Interfaz gráfica 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 cálculo 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

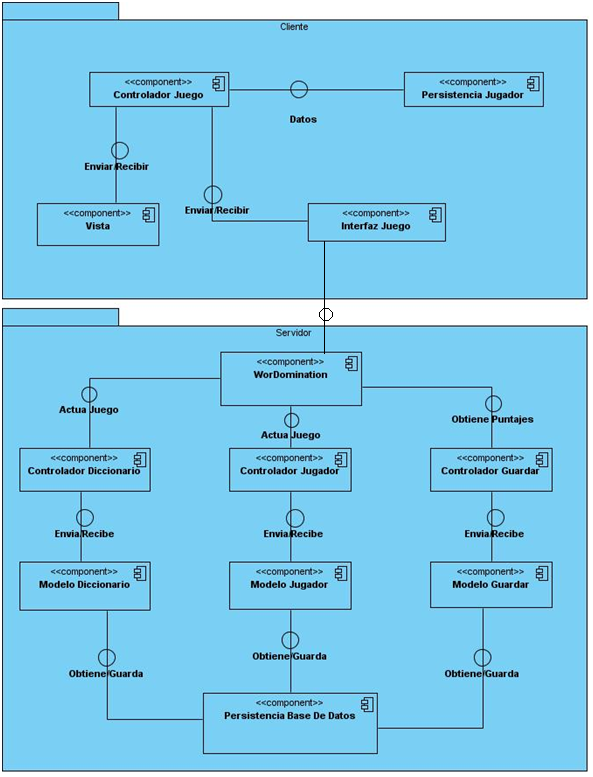


Ilustración 4: Diagrama de componentes

### Subsistema 1 – 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 |

Tabla : Subsistema 1 - servidor

### Componente 1 – Persistencia Base de datos

### bnvmnv

|  |  |  |  |
| --- | --- | --- | --- |
| **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* |
| *---* | *---* | *---* |

Tabla : Componente 1 – Persistencia Base de datos

*Esta sección finaliza cuando se hayan documentado todos los subsistemas y sus componentes asociados; para permitir la trazabilidad del sistema (con el uso de este documento) se sugiere utilizar algún tipo de convención, por ejemplo asignar a cada subsistema un color y dejar las tablas con la explicación de los componentes asociados con ese mismo color, de esta forma es más fácil relacionar los subsistemas y sus componentes con su módulo (sección 3.1) correspondiente.*

## Estrategias de Diseño

*Esta sección es un resumen de las diferentes estrategias de diseño que serán utilizadas por el equipo de desarrollo para la implementación de la aplicación, entre estas se pueden incluir [7]:*

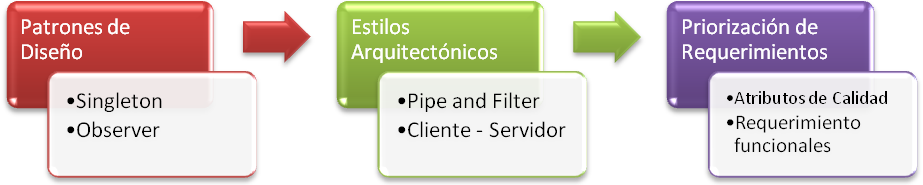


Ilustración : Estrategias de Diseño

*Un posible formato que se puede usar para la especificación de las Estrategias de Diseño es el descrito en la 5: Estrategias de Diseño*

|  |  |
| --- | --- |
| **Estrategias de diseño** | **Descripción** |
| **Estrategia 1** | * *Razones para la escogencia de la estrategia* * *Utilización de las estrategias dentro del contexto del proyecto* |

Tabla : Estrategias de Diseño

# Diseño de Alto Nivel

## Diagrama de despliegue

*El diagrama de despliegue describe la arquitectura física del sistema en ejecución, es decir, presenta los elementos hardware y software que ejecuta cada uno de ellos.*

*Un Diagrama de Despliegue describe una* ***arquitectura de ejecución****, esto es la configuración de hardware y software que define cómo el sistema estará configurado y además cómo operará (identificación de procesos). El propósito principal de un diagrama de este tipo es presentar una vista estática o “foto” del ambiente de implementación [6].*

*Una forma de explicar o documentar el diagrama de despliegue es haciendo la división por nodos y por conexiones:*



Ilustración : Descripción de Nodos

* Para los Nodos :
  1. Nombre del nodo: Este es el estereotipo que se le asigna al nodo.
  2. Especificación: Hace referencia a las características que debe tener la máquina.
  3. Ubicación: Lugar en donde se encuentra ubicada la máquina.
  4. Componentes: Enumera los componentes software que se ejecutará en el nodo.
  5. Comentarios adicionales: Notas.

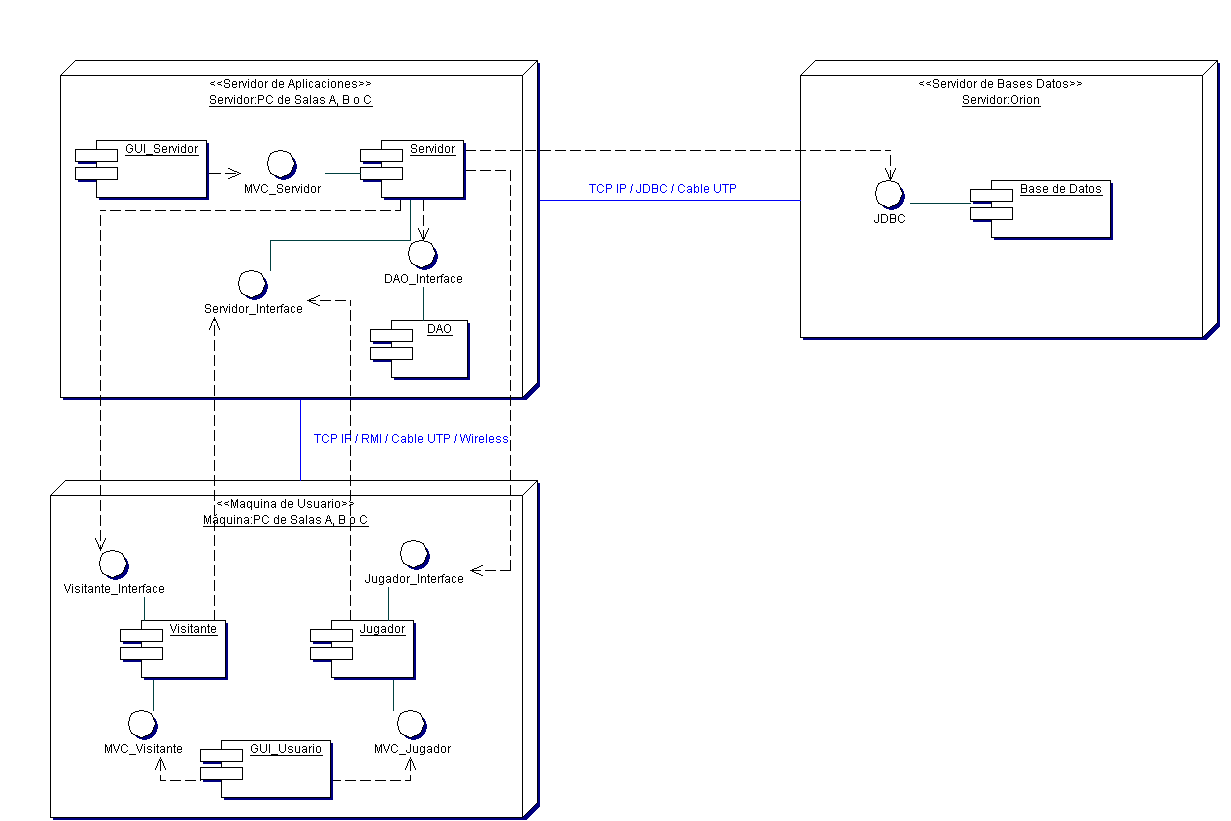


Ilustración : Descripción Conexiones

* Para las Conexiones :
  1. Comunicación: Indica el tipo y los nodos que está comunicando dicha conexión.
  2. Protocolos y herramientas: Como su nombre lo indica, describe el tipo de protocolo que se utiliza en la comunicación y algunas herramientas de software necesarias.
  3. Medio: Indica el medio físico por el cual se realiza la conexión.
  4. Comentarios adicionales: Notas.

Un ejemplo de un diagrama de despliegue es el encontrado en [3], de ese trabajo se tomó el Nodo Usuario, , que tenía conexión con el Nodo Servidor de Aplicaciones, del cual se observa una sección. Es posible ver los nodos que son los cubos en el diagrama y las conexiones que son las líneas continuas de color azul.

Ilustración : Nodo Usuario 7 Texas Poker



### Nodo 1 *“Nombre del Nodo”*

*En la 6: Nodo 1 se muestra un formato que puede ser usado para describir cada uno de los subsistemas de la aplicación final.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre del nodo** | *Estereotipo <<Nombre >>* | | |
| **Especificación** | *Remitirse a la sección 2.3 Entorno del Sistema, para la especificación del equipo* | **Ubicación** | *Espacio físico y real del equipo que está siendo descrito* |
| **Componentes** | | **Comentarios adicionales** | |
| * *Componente 1* * *Componente 3* | | * *Subsistemas o componentes albergados* | |

Tabla : Nodo 1

### Conector 1 “Nombre del Nodo”

*En la 7: Conector 1 muestra un formato que puede ser usado para describir cada uno de los subsistemas de la aplicación final.*

|  |  |  |
| --- | --- | --- |
| **Comunicación** | *Nombre de la asociación* | |
| **Protocolos y herramientas** | **Nombre** | **Descripción** |
| *Protocolo 1* | *Descripción del Protocolo 1* |
| *Herramienta 1* | *Descripción de la Herramienta 1* |
| **Medio** | *Elemento de conexión física ( Wireless- Cable UTP)* | |
| **Comentarios adicionales** | - | |

Tabla : Conector 1

## Diagrama de Comportamiento e Interacción

*Los diagramas de comportamiento de UML permiten tener un mejor acercamiento a lo que sucederá cuando la aplicación realice alguna funcionalidad específica, mientras que los diagramas de interacción añaden a esto el flujo de control y de datos [8].*

*Para estos diagramas lo ideal es escoger uno de los Casos de Uso principales del análisis realizado y hacer los diagramas en base a la elección. Una vez se ha seleccionado el caso de uso, también se hace la traza con los requerimientos asociados y se procede a la construcción de los diagramas.*

*Esta sección se divide en Diagramas de Actividad y de Secuencia*

### Diagrama de Actividad

*Una actividad es un paso en la ejecución de un programa, en la cual un trabajo se ejecuta. Esto puede incluir un cálculo, una consulta, manipulación de datos o simplemente verificación de información [6].*

*Los diagramas de actividad generalmente se utilizan para especificar [9]:*

* *Un método*
* *Un caso de uso*
* *Un proceso de negocio (Workflow)*

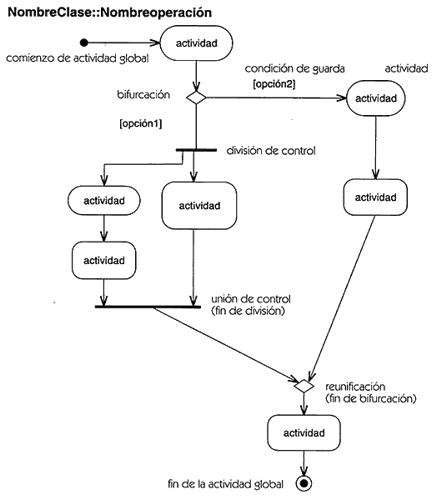


Ilustración : Diagrama de Actividad

### Diagrama de Secuencia

*Los diagramas de secuencia (al igual que los de colaboración) son usados para ilustrar la interacción entre objetos, por lo tanto estos diagramas modelan objetos y paso de mensajes entre objetos. Estos diagramas se hacen visualización basada en el tiempo [6].*

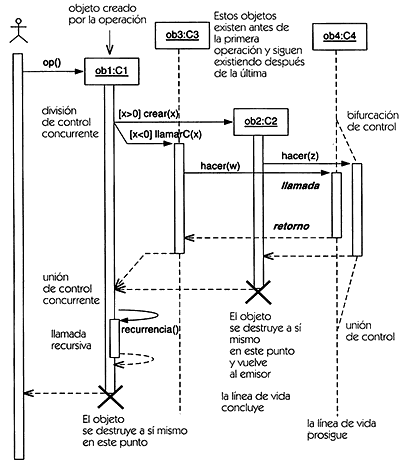


Ilustración : Diagrama de secuencia

# Diseño de Bajo Nivel

*El diseño de bajo nivel describe de forma detallada cómo van a ser implementados los subsistemas, paquetes, módulos y/o componentes que se hayan definido en el diseño de alto nivel [10], para cada uno de estos se deben especificar las clases, interacciones y demás estructuras necesarias para su construcción [11] (ver ilustración 17).*

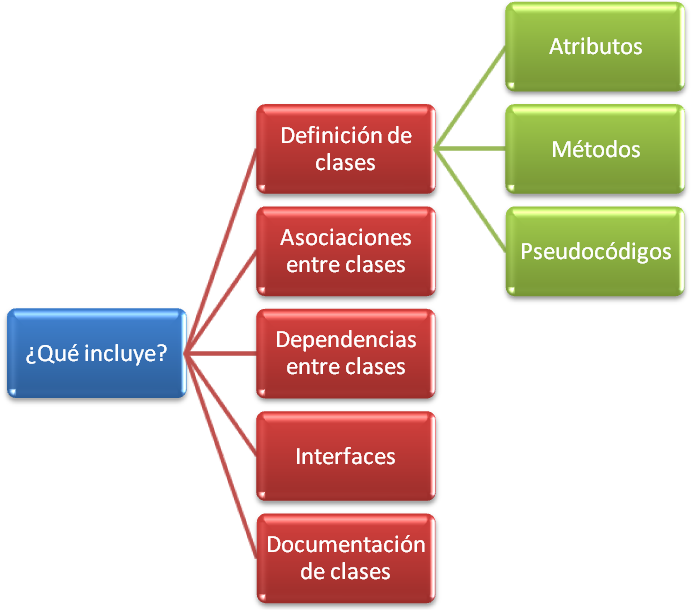


Ilustración : Diseño de bajo nivel

*Para realizar esta sección se deben llevar a cabo diferentes iteraciones, con las cuales se obtenga un mayor nivel de detalle hasta llegar a uno lo suficientemente claro para una fácil implementación desde el punto de vista del arquitecto. En la ilustración 18 se presentan los diferentes niveles de detalle del sistema de software, indicando cómo después del diseño de alto nivel se empieza con niveles más específicos como lo son la definición de clases con sus métodos y atributos. [12]*

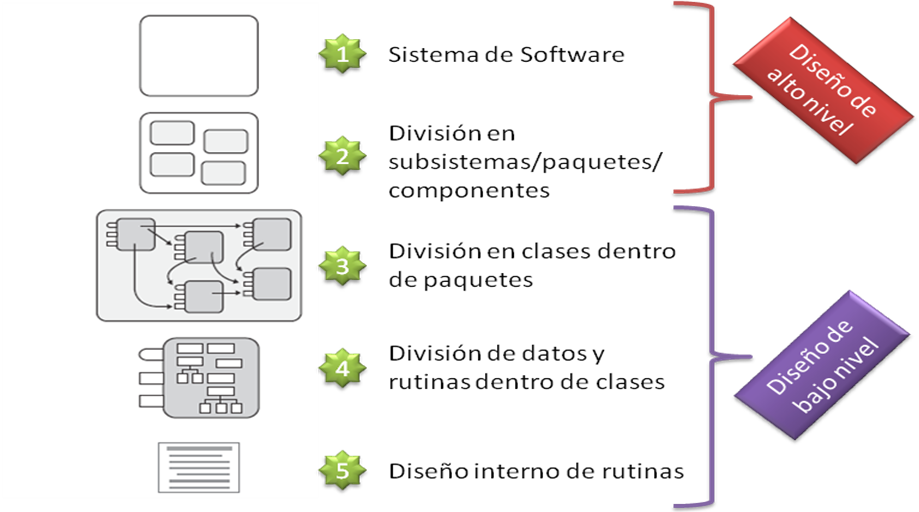


Ilustración : Niveles de diseño

*Para la descripción de bajo nivel se sugiere manejar una estructura jerárquica de los niveles que se hayan definido, en la figura 19 se presenta un ejemplo en el cual se han definido inicialmente subsistemas y estos a su vez se organizan en componentes.*

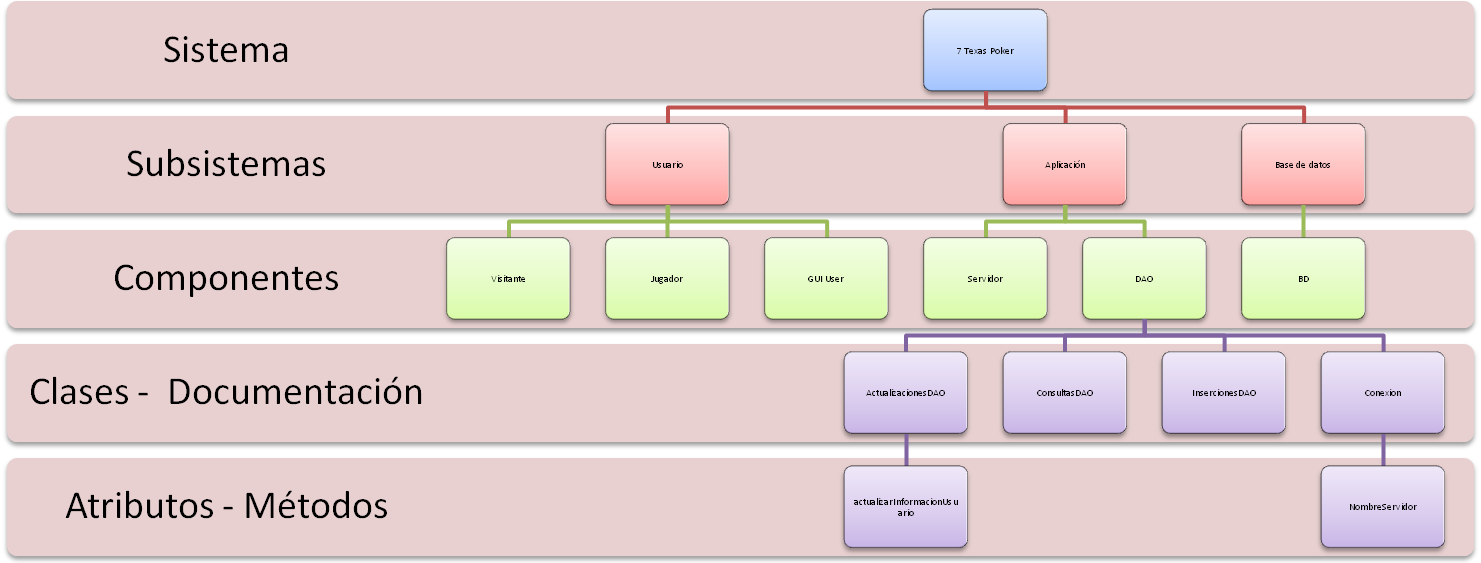


Ilustración : Ejemplo diseño de bajo nivel

## Subsistema 1

*En esta sección se deben listar todos los componentes, paquetes o módulos que contenga el subsistema 1.*

### Componente 1

*Se debe presentar el diagrama de clases del Componente 1 con su respectiva documentación, sin embargo, para comodidad se sugiere dejar en el presente documento la imagen del diagrama y realizar la documentación en el IDE seleccionado para la implementación con el fin de obtener la caracterización de las clases como un API, por ejemplo, utilizando JavaDoc.*

|  |  |
| --- | --- |
| Descripción de la clase | *Indica el objetivo que cumple la clase y las características propias de esta* |
| Descripción de atributos | *Por cada uno se debe especificar: el tipo de atributo, rango y una breve descripción.* |
| Descripción de métodos | *Por cada uno se debe incluir una breve descripción, la precondición para que el método pueda ser invocado, la poscondición indicando el resultado obtenido después de ser invocado, los parámetros que recibe (de estos debe estar especificado el rango y una breve descripción), el retorno indicando el tipo de dato que se retorna y el rango que posee; por último, se deben incluir las excepciones que va a manejar el método, describiendo en qué casos ocurriría.* |

Tabla : Documentación de clases

*En las ilustraciones 20, 21 y 22 se presenta un ejemplo de documentación de la clase “ConsultasDao” del proyecto 7 Texas Poker.*



Ilustración : Ejemplo Descripción de Clase



Ilustración : Ejemplo Resumen de Métodos



Ilustración : Ejemplo descripción de método

# Diseño de Interfaces de Usuario

## Diseño general de la aplicación

Se debe especificar cada una de las interfaces, indicando el diseño tanto de las entradas como de las salidas que generan información valiosa a los usuarios del sistema.

Para cada una de las interfaces de debe indicar:

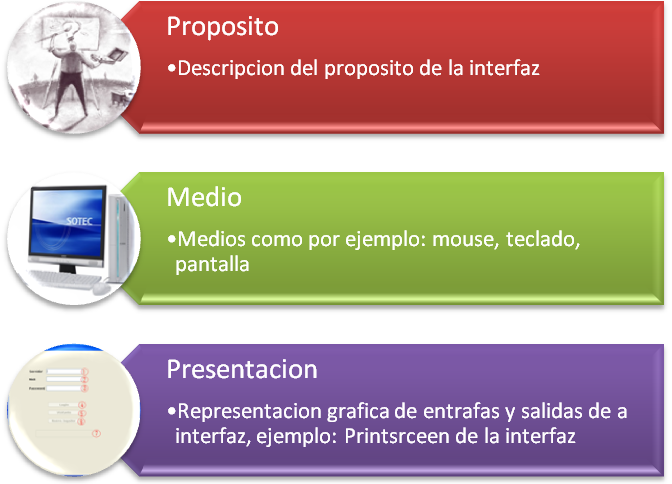


Ilustración : Diseño general de la aplicación

Además se debe manejar el siguiente formato para cada una de las entradas y salidas de la interfaz

|  |  |
| --- | --- |
| **Convención** | *Identificador de la entrada/salida dentro de la interfaz* |
| **Nombre** | *Nombre de la entrada/salida dado en las documentaciones* |
| **Alias (Interfaz)** | *Nombre o Label con el que identifican la entrada/salida* |
| **Alias (Código)** | *Nombre a nivel del código fuente del programa de la entrada/salida* |
| **Propósito** | *Propósito de funcionalidad de la entrada/salida* |
| **Tipo de Entrada** | *Por ejemplo: Campo de Texto, Tabla, Botón, etc.* |
| **Tipo de Dato** | *Por ejemplo: Numérico, Alfanumérico o Alfabético* |
| **Longitud mínima** | *Longitud mínima asignada a la entrada/salida* |
| **Longitud máxima** | *Longitud máxima asignada a la entrada/salida* |
| **Valor por defecto** | *Valor por defecto que se encuentra en la entrada/salida* |
| **Rango** | *Rango de los valores que puede tener la entrada/salida* |
| **Validación** | *Como se verifica que la entrada/salida es valida* |
| **Secuencia de Entrada** | *Orden en la secuencia de entradas/salidas de la interfaz* |
| **Comentarios** | *Comentarios como por ejemplo: obligatoriedad, medio por el cual entra o sale* |

Tabla : Descripción de entradas y salidas

## Árbol de navegabilidad

En el árbol de navegabilidad de las interfaces gráficas, cada nodo representa el título de dicha interfaz y las flechas los posibles caminos



Ilustración : Ejemplo árbol de navegabilidad

# Anexos