

---

**<Company Name>**

---

**Ultimate Tic Tac Toe  
Software Architecture Document**

**Version 1.0**

Ultimate Tic Tac Toe	Version: 1.0
Software Architecture Document	Date: 03/03/15
<document identifier>	

## Revision History

Date	Version	Description	Author
03/03/15	1.0	Primera entrega del documento de arquitectura de software (1, 3, 4, 5, 10).	David Puga Mendivil A01332391 Dennis Omar Lugo A01214271 Cynthia Garcia Velasco A01332329

Ultimate Tic Tac Toe	Version: 1.0
Software Architecture Document	Date: 03/03/15
<document identifier>	

# Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	5
1.5	Overview	5
2.	Architectural Representation	5
3.	Architectural Goals and Constraints	5
4.	Use-Case View	6
5.	Logical View	7
5.1	Overview	7
5.2	Architecturally Significant Design Packages	8
5.3	Use-Case Realizations	10
6.	Process View	11
7.	Deployment View	11
8.	Implementation View	12
8.1	Overview	12
8.2	Layers	12
9.	Data View (optional)	12
10.	Size and Performance	12
11.	Quality	12

Ultimate Tic Tac Toe	Version: 1.0
Software Architecture Document	Date: 03/03/15
<document identifier>	

# Software Architecture Document

## 1. Introduction

Este documento proporciona una visión general de la arquitectura de un sistema web desarrollado por los miembros del equipo. En este documento se describen las bases que los miembros usaran para la creación de una aplicación web, que estará basada en el clásico juego de gato, llamada Ultimate Tic Tac Toe. Este proyecto ayudará a los miembros a poner en práctica sus conocimientos para desarrollar el sistema previamente descrito.

El documento proporciona una descripción de alto nivel detallando los objetivos que tiene el desarrollo del proyecto, así como los casos de uso, capas, paquetes y subsistemas, que serán utilizados. De igual forma describe las restricciones y requerimientos necesarios para que la aplicación funcione de forma adecuada.

### 1.1 Purpose

La realización de nuestro proyecto tiene como objetivo demostrar los conocimientos adquiridos a lo largo del semestre en la materia de Diseño y Arquitectura de Software, aplicando los distintos tipos de patrones Gof en una aplicación web basada en el clásico juego de gato. A lo largo de este documento se describen las partes relacionadas con la estructura, creación e implementación del este sistema.

### 1.2 Scope

Este documento de arquitectura de software aplica al sistema web Ultimate Tic Tac Toe, el cuál será desarrollado a lo largo del semestre Enero-Mayo 2015. Esta aplicación estará basada en el clásico juego de gato para dos jugadores donde se tendrán nueve subtableros de gato que representarán cada casilla de un tablero de gato general. Para poder ganar la partida el jugador tendrá que ganar los respectivos subtableros del gato general que formen las líneas horizontales, verticales y diagonales, así como en el juego de gato clásico.

### 1.3 Definitions, Acronyms, and Abbreviations

Web	Conjunto de información que se encuentra en una dirección determinada de la web.
Servlets	Clase en el lenguaje de programación Java, utilizada para ampliar las capacidades de un servidor.
Java	Máquina virtual que usa Java Virtual Machine la cual contiene bibliotecas de la plataforma.
Base de datos	Programa capaz de almacenar gran cantidad de datos, relacionados y estructurados que pueden ser consultados rápidamente.
MySql	Sistema de gestión de base de datos relacional, multihilo y multiusuario.
Gof	Gang of Four. Se consideran los patrones de diseño fundamentales para orientación a objetos.
Servidor	Aplicación en ejecución capaz de atender las peticiones de un cliente y devolver una respuesta en concordancia.
Patrones de diseño	Son la base de las soluciones a problemas comunes en el desarrollo de software. Brindan una solución probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares.

Ultimate Tic Tac Toe	Version: 1.0
Software Architecture Document	Date: 03/03/15
<document identifier>	

## 1.4 References

Tbd

## 1.5 Overview

Este documento de arquitectura de software se divide en once secciones dentro de las cuales se describen las partes que conforman la arquitectura de la aplicación Ultimate Tic Tac Toe.

- En la primera sección se describen los objetivos de la creación y desarrollo de este documento, así como de la aplicación misma. De igual forma se describen las definiciones y referencias que serán usadas y consultadas a lo largo del desarrollo de este sistema.
- En la segunda sección se describe la representación arquitectónica del sistema, mencionando las vistas necesarias para el sistema (Use-Case, Logical, Process, Deployment, and Implementation Views), así como el modelo utilizado por cada una de éstas.
- En la tercera sección se describen los objetivos que se pretenden alcanzar el sistema, así como sus correspondientes restricciones y requerimientos.
- En la cuarta sección se representará el modelo de casos de uso donde se describirá el comportamiento del sistema.
- En la quinta sección se describe la parte lógica del sistema, la funcionalidad que será proporcionada a los usuarios. Aquí se describen las partes más significantes de modelo, la forma en la que se encuentra compuesto y sus paquetes. Así mismo se definirán las clases que componen cada paquete con sus respectivas relaciones, operaciones y responsabilidades.
- En la sexta sección se describe el sistema en procesos, la forma en que estos se encontrarán agrupados y la forma en la que se comunicarán entre ellos.
- En la séptima sección se describen las características físicas necesarias para que el sistema pueda ser implementado.
- En la octava sección se describe la estructura del modelo de implementación en capas y subsistemas, incluyendo los componentes más significativos de éste.
- En la novena sección se describe forma en la cual se mantendrá la persistencia de datos, en caso de que el sistema lo requiera.
- En la décima sección se describen las características más importantes que tendrán un impacto dentro de la arquitectura del sistema, así como sus respectivas restricciones.
- En la décimo novena sección se describe como la arquitectura de software contribuye a las funcionalidades del sistema y la importancia de estas funcionalidades para el sistema.

## 2. Architectural Representation

*[This section describes what software architecture is for the current system, and how it is represented. Of the Use-Case, Logical, Process, Deployment, and Implementation Views, it enumerates the views that are necessary, and for each view, explains what types of model elements it contains.]*

## 3. Architectural Goals and Constraint

Los requerimientos de nuestra aplicación web Ultimate Tic Tac Toe:

- Contar con un navegador web (Preferentemente Chrome y Firefox).
- Contar con una conexión a internet.
- Tener java instalado.

Objetivos:

- Funcionamiento completo de la aplicación web Ultimate Tic Tac Toe.
- Uso de patrones de diseño Gof aprendidos en clase en la implementación del sistema.
- Mantener la aplicación portable a través de los navegadores web.
- Mantener la conexión entre los jugadores en un mismo juego.

Ultimate Tic Tac Toe	Version: 1.0
Software Architecture Document	Date: 03/03/15
<document identifier>	

Restricciones importantes a considerar del sistema:

- Sólo soporta dos jugadores simultáneamente.
- Tener instalado plugins de java.
- Conexión a internet.

Estrategia de diseño e implementación:

- Uso de patrones de diseño Gof (Gang of Four).
- Uso metodología cliente-servidor para conexión del juego.

Herramientas de desarrollo:

- Lenguaje de programación Java.
- Ambiente de desarrollo Netbeans.
- Java servlets para conexión entre computadoras.
- JavaFX para diseño de interfaz gráfica.
- MySql para conexión a base de datos.

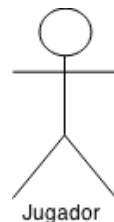
*[This section describes the software requirements and objectives that have some significant impact on the architecture; for example, safety, security, privacy, use of an off-the-shelf product, portability, distribution, and reuse. It also captures the special constraints that may apply: design and implementation strategy, development tools, team structure, schedule, legacy code, and so on.]*

#### 4. Use-Case View

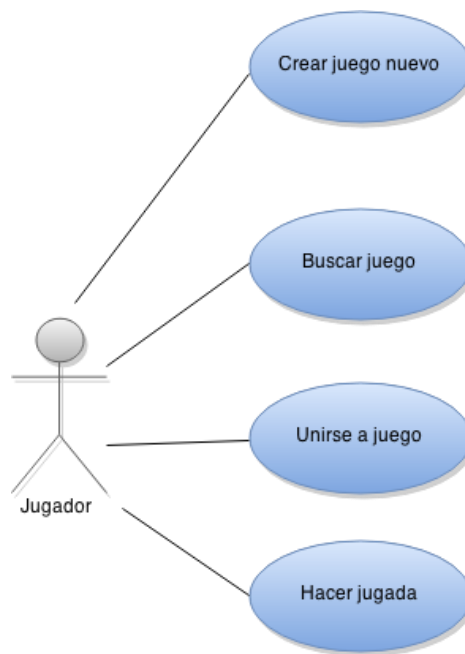
Casos de uso:

- Crear juego nuevo: este caso de uso describe como un usuario puede crear un nuevo juego de Ultimate Tic Tac Toe. El actor de este caso de uso es el jugador.
- Buscar juego: este caso de uso describe como un usuario puede buscar un juego en el cuál puede participar. El actor de este caso de uso es el jugador.
- Unirse a juego: este caso de uso describe como un usuario puede unirse a un juego existente, es decir, que haya sido iniciado por otro jugador, con la restricción de que sólo puede hacer dos jugadores en un solo juego. El actor de este caso de uso es el jugador.
- Hacer jugada: este caso de uso describe como un usuario puede hacer una jugada, es decir, escoger entre X o O para colocar en algunas de las casillas del tablero. El actor de este caso de uso es el jugador.

Actor:



Ultimate Tic Tac Toe	Version: 1.0
Software Architecture Document	Date: 03/03/15
<document identifier>	



## 5. Logical View

### 5.1 Overview

En esta sección se describen las clases más importantes del sistema, organizados en subsistemas y paquetes. De igual forma describe gráficamente los casos de uso más importantes del sistema. Se incluye también el diagrama de las clases para exponer las relaciones más significativas entre las clases y los paquetes.

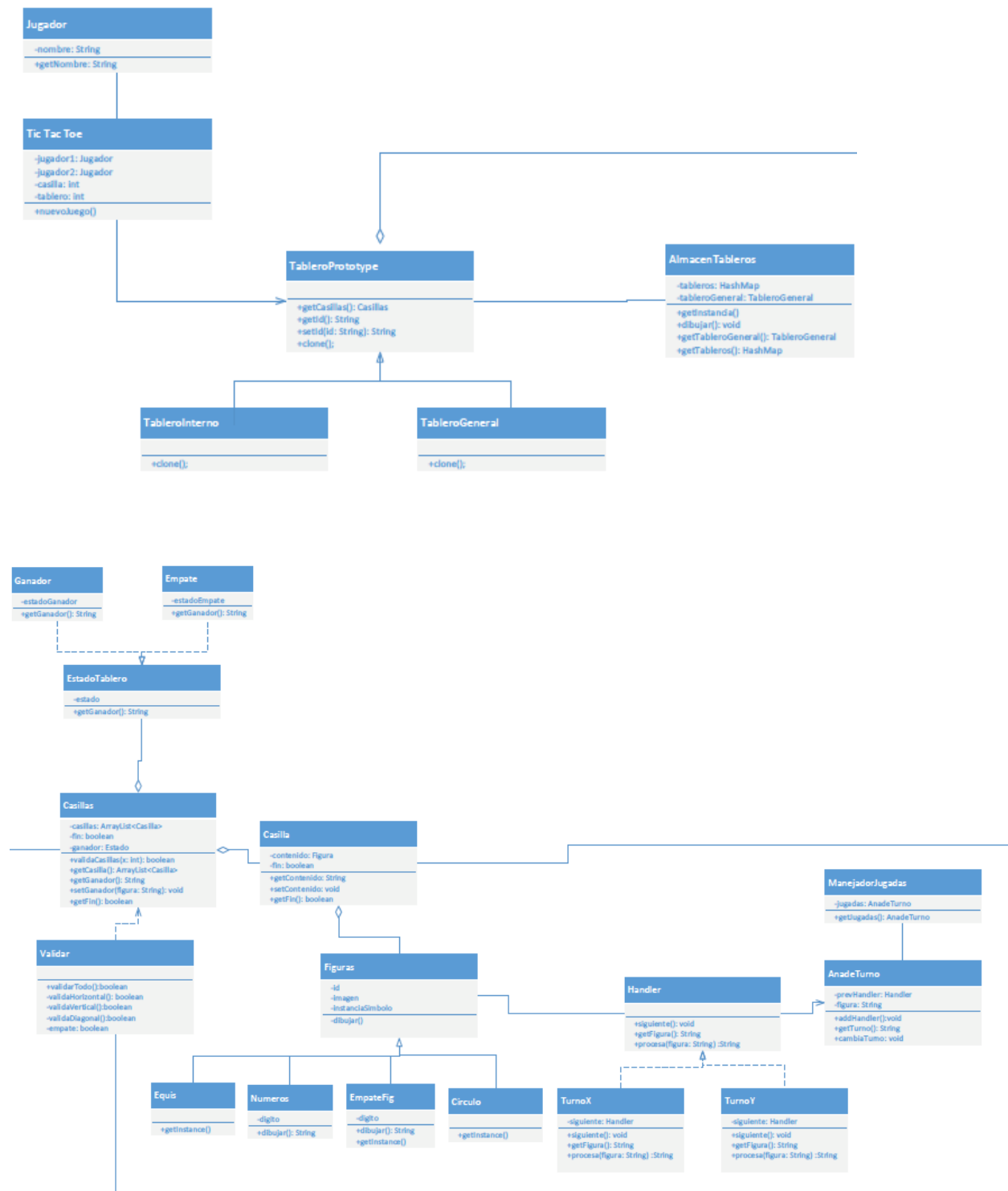
En esta vista lógica el sistema tiene tres capas principales: Presentación, Negocio y Datos.

- Dentro de la capa de presentación se incluye el paquete de la interfaz de usuario, que representa la forma en la cual el actor, en este caso el jugador, interactuará con el sistema.
- Dentro de la capa de negocio se contienen las clases encargadas del control del sistema y las entidades que controlan.
- Dentro de la capa de negocio se incluyen los paquetes encargados de la interacción con la base de datos donde se almacenará la información necesaria para la persistencia de datos.



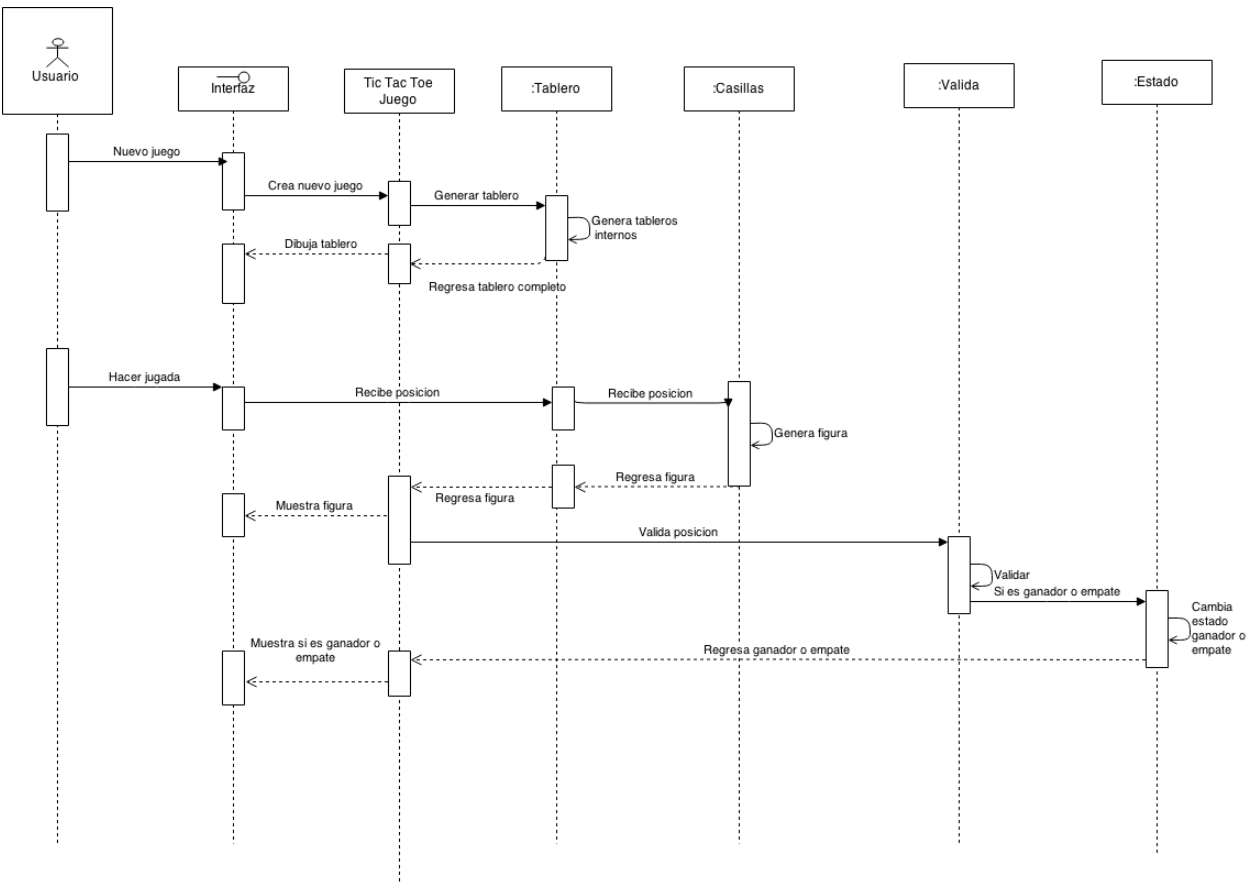


Ultimate Tic Tac Toe	Version: 1.0
Software Architecture Document	Date: 03/03/15
<document identifier>	

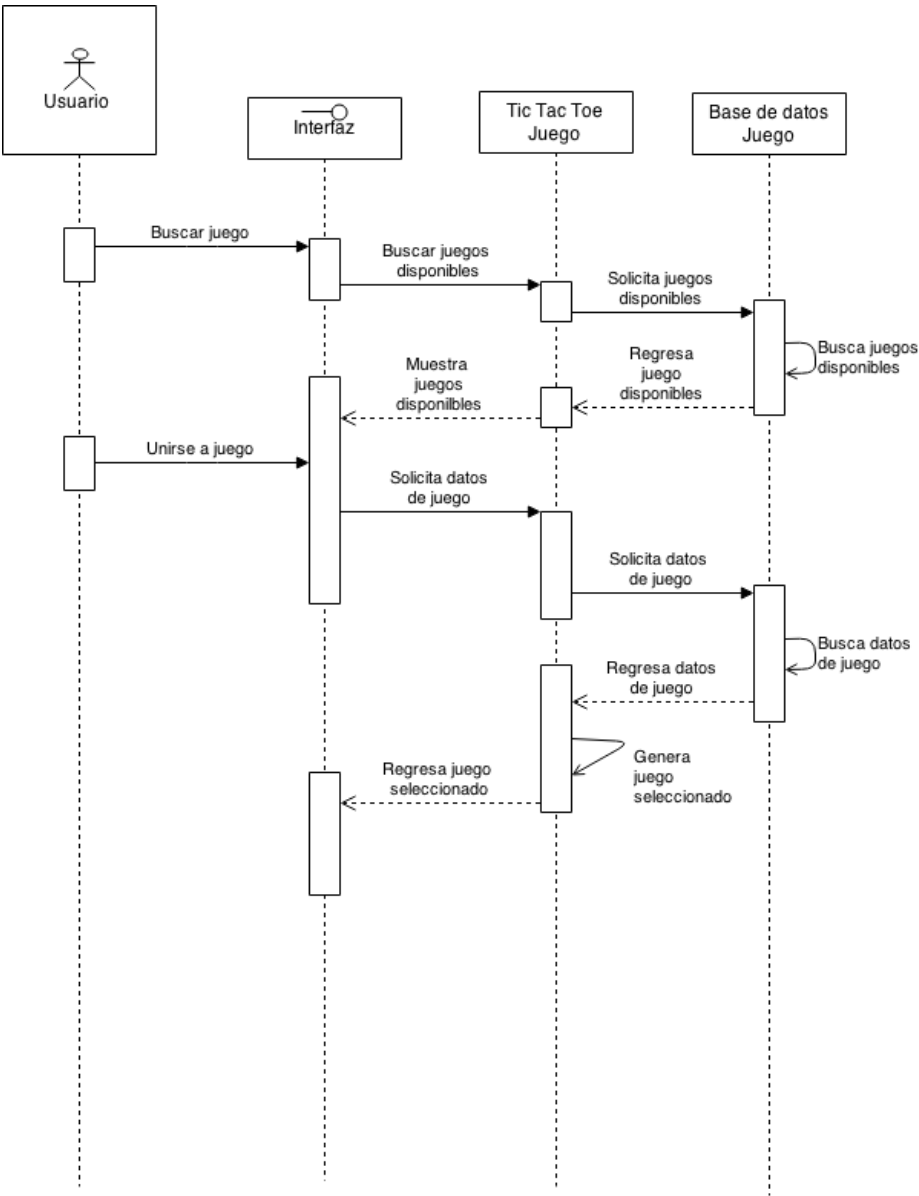


\*\*Diagrama completo en anexo

5.3 Use-Case Realizations



Ultimate Tic Tac Toe	Version: 1.0
Software Architecture Document	Date: 03/03/15
<document identifier>	



6. Process View

[This section describes the system's decomposition into lightweight processes (single threads of control) and heavyweight processes (groupings of lightweight processes). Organize the section by groups of processes that communicate or interact. Describe the main modes of communication between processes, such as message passing, interrupts, and rendezvous.]

7. Deployment View

[This section describes one or more physical network (hardware) configurations on which the software is deployed and run. It is a view of the Deployment Model. At a minimum for each configuration it should indicate the physical nodes (computers, CPUs) that execute the software and their interconnections (bus, LAN, point-to-point, and so on.) Also include a mapping of the processes of the Process View onto the physical nodes.]

Ultimate Tic Tac Toe	Version: 1.0
Software Architecture Document	Date: 03/03/15
<document identifier>	

## 8. Implementation View

*[This section describes the overall structure of the implementation model, the decomposition of the software into layers and subsystems in the implementation model, and any architecturally significant components.]*

### 8.1 Overview

*[This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers. ]*

### 8.2 Layers

*[For each layer, include a subsection with its name, an enumeration of the subsystems located in the layer, and a component diagram.]*

## 9. Data View (optional)

*[A description of the persistent data storage perspective of the system. This section is optional if there is little or no persistent data, or the translation between the Design Model and the Data Model is trivial.]*

## 10. Size and Performance

Este diseño de arquitectura de software suporta los siguientes requerimientos:

- El sistema puede soportar como máximo a 2 usuarios conectados simultáneamente al servidor
- El sistema puede proporcionar la persistencia de datos mientras exista la conexión al servidor y a la base de datos.
- El sistema puede correr en cualquier computadora, mientras tenga instalado un navegador web y tenga conexión a internet.
- El cliente debe de tener instalados los plugins de java para su navegador web.
- Se requiere una conexión a internet mínima de 5 MB para que los usuarios puedan usar la aplicación.
- El cliente requiere un espacio en disco duro mínimo 600 MB libres, necesarios para la instalación de un navegador web y los plugins de java.

La arquitectura seleccionada es compatible con los requisitos de tamaño y de sincronización a través de la implementación de una arquitectura cliente-servidor. Los componentes han sido diseñados para garantizar que se necesitan requisitos mínimos de disco y de memoria por parte de la computadora del cliente.

## 11. Quality

*[A description of how the software architecture contributes to all capabilities (other than functionality) of the system: extensibility, reliability, portability, and so on. If these characteristics have special significance, such as safety, security or privacy implications, they must be clearly delineated.]*