Documento de Arquitectura del Software

Proyecto: Sistema de Trámite Documentario

Versión: 1.0

Historial de Revisiones

| **Versión** | **Fecha** | **Autor** | **Descripción** |
| --- | --- | --- | --- |
| 1.0 | 04/10/13 | Añaños Reyes, Javier  Pérez Álvarez, Alex  Rodríguez Flores, Jorge | Documento de Arquitectura de Software |

Índice de Contenido

1 Introducción 5

1.1 Alcance 5

1.2 Documentos relacionados 5

1.3 Definiciones 5

1.4 Acrónimos 5

2 Objetivos 6

2.1 Objetivos Específicos 6

2.2 Requerimientos especiales 6

3 Metas y restricciones 7

3.1 Metas 7

3.2 Restricciones 7

4 Vista de Casos de Uso 7

4.1 Vista de Casos de Uso: Paquete Trámite 9

4.2 Vista de Casos de Uso: Paquete Usuarios 9

4.3 Vista de Casos de Uso: Paquete Reportes 9

4.4 Vista de Casos de Uso: Paquete Seguridad 9

5 Vista Procesos 10

6 Vista Lógica 10

7 Vista de Implementación 12

8 Vista de Implantación 12

9 Vista de Datos 13

10 Aseguramiento de la Calidad 13

10.1 Objetivos de Calidad 13

10.1.1 Esenciales 13

10.1.2 Esperados 13

10.1.3 Deseados 14

11 Glosario 15

12 Anexos 15

**Documento de Arquitectura del Software**

# Introducción

La definición de una Arquitectura de Software aporta en particular una visión abstracta de alto nivel al realizarse el diseño, que pospone cada uno de los módulos definidos a los pasos posteriores del mismo.

Por lo que al esbozar el diseño, se debe tomar mucha preponderancia en el momento de fijar una arquitectura y así tomarla como algo clave a la hora de diseñar una solución.

## Alcance

Este documento se concreta, de manera puntual, en el desarrollo de la vista lógica y de implementación correspondientes que comprenderán de cierta manera el resto de vistas a elaborarse en el proceso. Así como también a la identificación de los diferentes componentes pertenecientes a cada vista, para el desarrollo del Sistema de Tramite Documentario.

## Documentos relacionados

Para poder visualizar las referencias a otros documentos, se debe de llenar la tabla que se muestra a continuación:

| **Título** | **Identificador del documento** |
| --- | --- |
| Documento de Procesos de Negocio | [STD\_PRO\_NEG.docx](https://github.com/jorgedison/Proyecto-STD/blob/master/METODOLOGIA%20RUP/PROYECTO%20STD/01%20MODELO%20DE%20NEGOCIO/STD_PRO_NEG.docx) |
| Listado de requisitos del Sistema | [STD\_LIST\_REQ.xlsx](https://github.com/jorgedison/Proyecto-STD/blob/master/METODOLOGIA%20RUP/PROYECTO%20STD/02%20REQUISITOS/STD_LIST_REQ.xlsx) |
| Matriz de Trazabilidad Requisitos del Sistema – Casos de uso | [STD\_MATR\_TRAZ.xlsx](https://github.com/jorgedison/Proyecto-STD/blob/master/METODOLOGIA%20RUP/PROYECTO%20STD/02%20REQUISITOS/STD_MATR_TRAZ.xlsx) |

## Definiciones

|  |  |
| --- | --- |
| **Definición** | **Descripción** |
| Paquetes | Agrupaciones de casos de uso y actores por funcionalidad que proveen. |
| Actor | Alguien o algo externo al sistema que interactúa con él. |
| Caso de Uso | Secuencia de acciones que el sistema realiza, la cual proporciona un resultado de valor observable. |
| StartUML | Se refiere a la herramienta que permite realizar el modelado de los diagramas presentados en este documento. |

## Acrónimos

|  |  |
| --- | --- |
| **Acrónimo** | **Significado** |
| ERS | Especificación de Requisitos de Software |
| RUP | Rational Unified Process |
| UML | Unified Modeling Language |

# Objetivos

El proyecto Sistema de Tramite Documentario será desarrollado por etapas en base a los diferentes modelos principalmente dados por la Vista de Casos de Uso, Lógica, implementación, implantación y Datos.

## Objetivos Específicos

* Diseñar una arquitectura que sea necesaria y suficiente, que cuente con un funcionamiento efectivo, es decir, que provea las funciones y propiedades especificadas.
* Elaborar un diseño conciso, claro, con alto grado de cohesión y bajo acoplamiento entre cada uno de los componentes, para así, permitir la manipulación y reemplazo de los mismos con facilidad que anticipe mudanzas posibles y permita el trabajo a futuro.
* Desarrollar un bosquejo inicial de la aplicación, que integre distintos sistemas informáticos, los cuáles darán soporte a tareas a realizarse en el Sistema de Tramite Documentario.

## Requerimientos especiales

Al desarrollar el proyecto de Sistema de Tramite Documentario, se debe tener en cuenta las siguientes consideraciones acerca del mismo.

## Usabilidad.

## La interfaz del usuario deberá ser tan familiar, práctica e intuitiva como sea posible a los usuarios que han usado otras aplicaciones Web.

## Escalabilidad.

## La aplicación Web tiene que ser completamente escalable sin que un aumento de los recursos dedicados a la misma suponga modificación alguna en su comportamiento o capacidades.

## Seguridad

## El acceso debe ser controlado con nombres de usuario y contraseñas. Solo los usuarios con derechos de administrador podrán acceder las funciones administrativas mientras los usuarios normales no.

## Portabilidad

## La aplicación debe poder adaptarse a las distintas posibles arquitecturas físicas susceptibles de ser empleadas para el despliegue del paquete, limitándose en la medida de lo posible el impacto de tal adaptación a tareas de configuración, y evitándose así la necesidad de modificar el código de la misma ante dichas situaciones.

## Importación y exportación de datos

## El sistema deberá almacenar todos los datos en una base de datos SQL estándar, donde pueda ser accesado por otros programas.

# Metas y restricciones

La meta principal de la arquitectura del sistema es mostrar los aspectos principales que influirán en la etapa de desarrollo.

Se tomarán en cuenta las siguientes metas y restricciones para el diseño de la arquitectura del sistema:

# Metas

* El Sistema de Tramite documentario permitirá a los usuarios acceder al sistema desde los browsers Internet Explorer 7 o superior, Mozilla Firefox 4 o superior, Chrome y Opera.
* El tiempo de procesamiento para todas las consultas no deberá ser mayor a 10 segundos.
* Para el caso de los registros, modificaciones y eliminaciones no deberá ser mayor a 5 segundos. El uso del sistema para los trabajadores de la empresa no se limita a dentro de la organización.
* Tiempo de respuesta de transacciones tales como registrar, actualizar y eliminar no demorará más de 5 minutos en generar una distribución óptima para los cortes haciendo uso del algoritmo elegido y guardándolo en base de datos.
* Espacio en disco necesario para el servidor de Base de datos deberá tener como mínimo 1 GB libre en disco para su correcto funcionamiento.

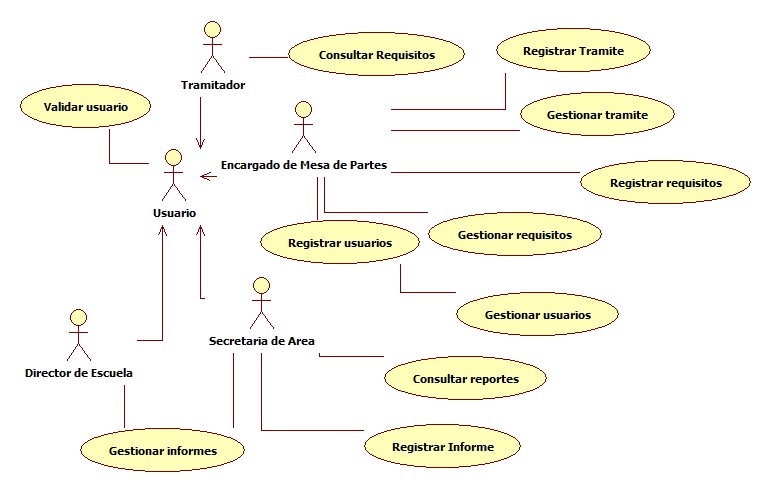
# Restricciones

* Se deberá contar con un servidor web y de aplicaciones para poder poner la aplicación en ejecución.
* Necesariamente el sistema usará como motor de Base de Datos MySQL 5.0. Siendo necesario la elaboración de una copia de rescate de las tablas para no afectar la información almacenada antes de poner en total funcionamiento al sistema.
* Las computadoras clientes deberán contar como mínimo con procesadores Pentium III a más, memoria RAM mínima de 1 GB y sistema operativo Microsoft o Linux.
* El servidor web y de aplicaciones requerirá al menos 5GB de disco duro y 2GB de RAM. Debe contar con un mínimo de 500 MB de espacio libre en el disco y el sistema operativo debe ser Linux.
* Todos los nodos que hagan uso del sistema, al estar este desarrollado en Java, deberán de tener instalado el JRE 1.6 o superior
* Se requerirá la implementación de una interfaz que se comunique con el módulo de seguridad interno de la empresa donde se desplegará la solución.

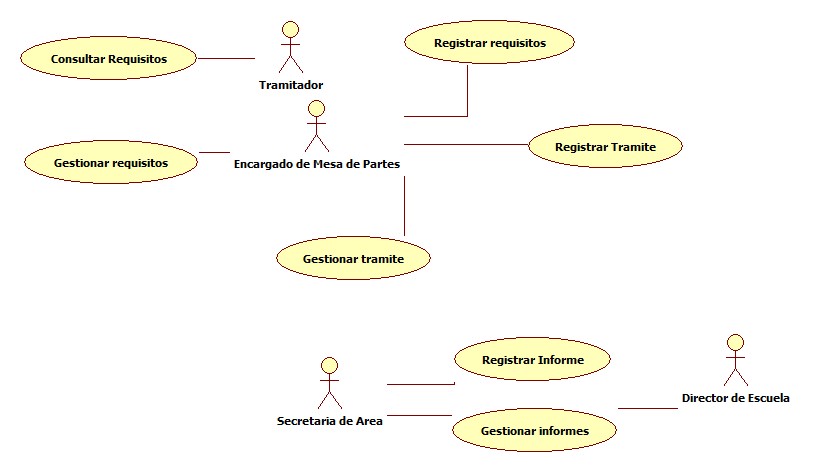
# Vista de Casos de Uso

El diagrama de casos de uso representa la forma de cómo un cliente opera con el sistema en desarrollo, además de la forma tipo y orden de los elementos que interactúan y así obtener el conjunto de escenarios que contendrá cada iteración.

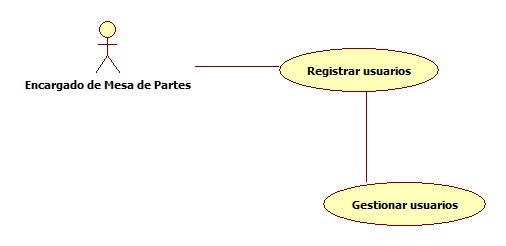
|  |  |
| --- | --- |
| **N CUS** | **Caso de Uso** |
| CU-01 | Consultar Requisitos |
| CU-02 | Registrar tramite |
| CU-03 | Gestionar tramite |
| CU-04 | Registrar requisitos |
| CU-05 | Gestionar requisitos |
| CU-06 | Registrar Informe |
| CU-07 | Gestionar informes |
| CU-08 | Validar usuario |
| CU-09 | Registrar usuarios |
| CU-10 | Gestionar usuarios |
| CU-11 | Consultar reportes |



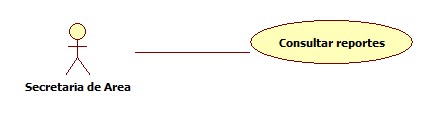
# Vista de Casos de Uso: Paquete Trámite



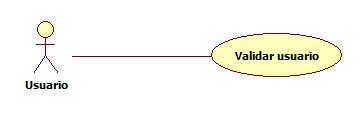
# Vista de Casos de Uso: Paquete Usuarios



# Vista de Casos de Uso: Paquete Reportes



# Vista de Casos de Uso: Paquete Seguridad



# Vista Procesos

Es el diagrama principal para el análisis y diseño. En este diagrama se presentan las clases del sistema con sus relaciones estructurales y de herencia. Para establecer las clases, objetos atributos y operaciones entre clases se tomó como base a los diagramas de casos de uso.



# Vista Lógica

La vista lógica del sistema de trámite documentario está compuesta de tres paquetes principales: Interfaz de usuario, servicios del negocio y datos de acceso.

El primer paquete, Interfaz de usuario, contiene las clases visuales de la aplicación, las cuales crean las ventanas que utiliza el usuario para comunicarse con el sistema para una adecuada y fácil utilización de los paquetes que se encuentran en las siguientes capas.



El paquete Lógica del Negocio contiene las clases que permiten llevar a cabo los servicios del negocio, como son: Controlar el mantenimiento de usuarios, atender los requerimientos a cortar, etc.



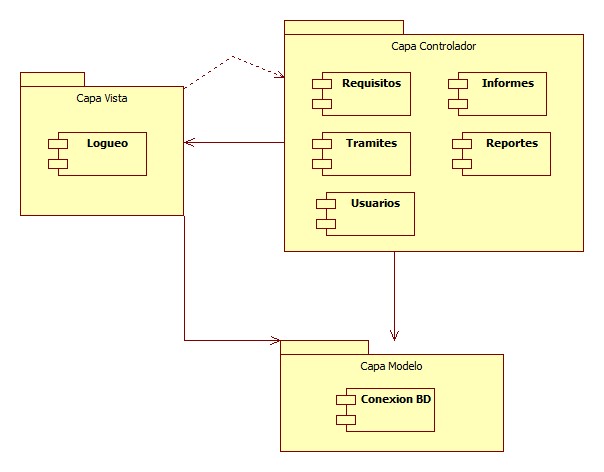
El paquete de Objetos de Acceso de Datos contiene las clases que el sistema utiliza para manejar la persistencia de los objetos con la Base de Datos.





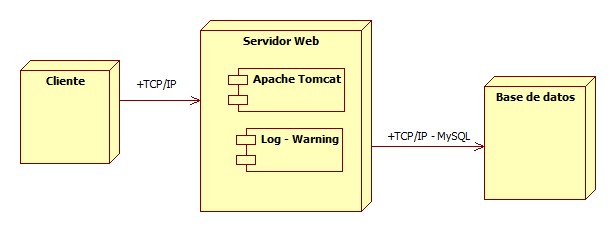
# Vista de Implementación

En esta vista se muestra en general las dependencias y cómo se implementan los componentes físicos del sistema, agrupándolos en subsistemas organizados en capas y jerarquías.



# Vista de Implantación

En esta vista se describe las configuraciones de las redes físicas (hardware) sobre las cuales el software será desarrollado. Se debe indica los nodos físicos para cada configuración que ejecutan el software, y sus interconexiones.



# Vista de Datos



# Aseguramiento de la Calidad

## Objetivos de Calidad

### Esenciales

* Funcionalidad > Corrección
* Funcionalidad > Robustez

### Esperados

* Funcionalidad > Exactitud
* Funcionalidad > Compatibilidad
* Funcionalidad > Corrección medible
* Usabilidad > Comprensibilidad y Legibilidad
* Usabilidad > Apoyo para tareas
* Usabilidad > Eficiencia
* Usabilidad > Seguridad
* Usabilidad > Consistencia y Familiaridad
* Usabilidad > Satisfacción Subjetiva

### Deseados

* Confiabilidad > Consistencia en carga
* Confiabilidad > Consistencia bajo concurrencia
* Confiabilidad > Disponibilidad bajo carga
* Confiabilidad > Longevidad
* Eficiencia
* Escalabilidad
* Escalabilidad > Desempeño bajo carga
* Escalabilidad > Grandes volúmenes de datos
* Operabilidad
* Capacidad de mantenimiento > Comprensibilidad
* Capacidad de mantenimiento > Capacidad de evolución
* Capacidad de mantenimiento > Capacidad de prueba

# Glosario

*Arquitectura de Software*; estructura del sistema, que incluye componentes de software, las propiedades visibles externas de esos componentes, y las relaciones entre estos.

# Anexos

Software Architecture Document, MedBiquitous project

<http://www.medbiq.org/technology/tech_architecture/softwarearchitecture.pdf>

Software Architecting: How is Software Architecture Created?

<http://www.bredemeyer.com/howto.htm>

Software Architecture , Action Guide from Bredemeyer Consulting

<http://www.bredemeyer.com/pdf_files/action_guide.pdf>

También se agregará como anexo páginas con ejemplos de los diferentes diagramas elaborados:

http://www.agilemodeling.com/artifacts/componentDiagram.htm

http://www.netbeans.org/kb/60/uml/why-model.html

http://www.extex.org/development/state.html