*Documento de Arquitectura*

Versión 1.0

Historial de Revisiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Historial de revisiones | | | | |
| Ítem | Fecha | Versión | Descripción | Equipo |
| 1 | 9/05/2013 | 1.0 | Versión inicial. | Eysenck Gómez  José Manuel Rios  Katherine Velásquez |
|  |  |  |  |  |

Tabla de Contenido

1. Introducción 3

1.1. Propósito 4

1.2. Alcance 4

1.3. Definiciones, acrónimos y abreviaturas 4

1.3.1. Definiciones 4

1.3.2. Acrónimos 4

1.4. Referencias 4

1.5. Visión General del documento 5

2. Representación de la arquitectura 5

3. Metas y Restricciones de la arquitectura 5

3.1. Metas 5

3.2. Restricciones del Sistema 5

4. Vista Lógica 7

4.1. Diseño en Capas 7

4.1.1. Capa Template 8

4.1.2. Capa View 8

4.1.3. Capa Modelo 8

5. Vista de Despliegue 8

6. Tamaño y performance de la Base de Datos 9

Documento de Arquitectura de Software

# Introducción

Uno de las fases más importantes dentro de la construcción del software es el diseño de la arquitectura de software, que permite representar la estructura del sistema sirviendo de comunicación entre las personas involucradas en el proceso de desarrollo y ayudando a realizar diversos análisis que orienten la toma de decisiones.

Este documento provee al usuario especializado una vista de la arquitectura del Sistema de Gestión de Tareas.

La plantilla de este documento se basó en las especificaciones de RUP (Rational Unified Process) para el documento de arquitectura de software.

## Propósito

El propósito general del sistema es brindar una solución para el desenvolvimiento diario de las empresas tipo 24/7, estas empresas tendrán en el sistema la manera más fácil de organizar sus tareas y repartirlas a su personal de acuerdo a las habilidades, tiempos y criticidad de la tarea, además de poder ver estadísticas del trabajo realizado.

## Alcance

Este documento presenta la arquitectura en la cual estará basado el Sistema de Gestión de Tareas, definiendo de manera detallada la distribución de los paquetes del sistema en las diversas capas que éste presenta, así como una descripción de las clases a utilizar.

## Definiciones, acrónimos y abreviaturas

Se brindan definiciones y acrónimos de términos usados en el presente documento que necesiten de alguna explicación para su correcta interpretación.

### Definiciones

|  |  |
| --- | --- |
| **Paquete** | Agrupaciones de casos de uso y actores por la funcionalidad que proveen. |
| **Actor** | Alguien o algo externo al sistema y que interactúa con él. |
| **Caso de Uso** | Secuencia de acciones que el sistema realiza, la cual proporciona un resultado de valor observable. |
| **Rational Software Architect** | Es una herramienta de diseño y desarrollo integrados que potencia el desarrollo orientado al modelado con UML para la creación de aplicaciones y servicios con buena arquitectura.. |

### Acrónimos

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

## Referencias

Las referencias aplicables son:

1. Documento de Especificaciones de CUS del Proyecto Sistema de Gestión de Tareas (SGT-ECU)
2. Documento Project Charter del Proyecto Proyecto Sistema de Gestión de Tareas (SGT-PC).

## Visión General del documento

En las secciones de este documento se detalla la arquitectura del software a desarrollar. Para ello se presenta de manera clara el caso de uso que más representa la arquitectura del sistema, empleando un lenguaje sencillo y directo, así como gráficos y vistas de acuerdo a la metodología utilizada.

# Representación de la arquitectura

La Arquitectura a utilizar será Cliente-Servidor con un estilo arquitectónico MTV (Modelo Template View), una aproximación del modelo MVC en la cuál las vistas no representan la capa de interfaz de usuario sino la lógica del negocio y los Templates cumplen el rol común de Vista en el modelo MVC. El cliente es la aplicación que se ejecutará via web desde un lugar remoto por el usuario. Se emplea un servidor principal.

La arquitectura se muestra como una serie de vistas empleando UML.

# Metas y Restricciones de la arquitectura

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

1. El sistema de Gestion de Tareas permitirá a los usuarios de tipo administrador crear solicitudes de tarea, usar un algoritmo de asignación de estas tareas a los empleados, y a los usuarios empleados a poder tener una cola de tareas e ir marcándolas como resueltas.
2. Para poder acceder al el sistema de Gestion de Tareas se necesitará contar con credenciales de usuario administrador o empleado, un nombre de usuario y una contraseña.
3. Los requerimientos de rendimiento estipulados como requisitos no funcionales, se consideran objetivos críticos al momento de elegir la arquitectura a implementar.

## Restricciones del Sistema

1. El sistema de Gestion de Tareas usará como motor de Base de Datos PostgreSQL por su naturaleza Open Source y su capacidad para procesar una gran cantidad de data.
2. Los navegadores web que deben usarse para acceder a la aplicación web deben ser como mínimo Chrome desde su versión 17, Firefox desde su versión 6, o Internet Explorer desde su versión 9, en otro caso de uso de un navegador alternativo, este debería poder soportar los puntos básicos de HTML5.
3. El sistema en donde se haga el paso a producción deberá contar con Python en su versión mínima de 2.6.

# Vista Lógica

## Diseño en Capas

En la siguiente figura podemos observar la distribución de los paquetes en cada una de las capas del sistema.

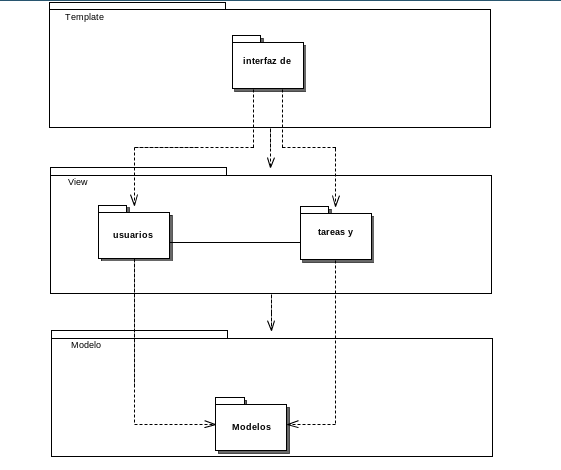


Figura .1: Diagrama de Distribución de Capas del Sistema

### Capa Template

Esta capa contiene las vistas con las cuales interactúa el usuario.

### Capa View

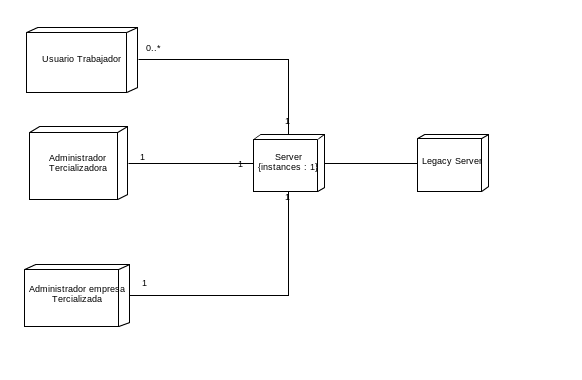
Esta capa contiene la lógica para el manejo de las operaciones del proceso de Gestión de Tareas.

### Capa Modelo

Esta capa contiene el paquete de Objetos de Acceso a Base de Datos, los cuales permiten leer y escribir entidades en la Base de Datos.

# Vista de Despliegue

El presente Diagrama de Despliegue muestra de manera gráfica los nodos que conforman el sistema, además los describe indicando la localización de las tareas en los nodos físicos. Cada nodo representa un recurso de ejecución como una computadora o servidor y se encuentra conectado con otros mediante un enlace de comunicación. En este caso, la red será de tipo LAN.

Figura 7.1: Diagrama de Despliegue del Sistema. Servidor (Base de Datos) y Cliente (Computadora usuario).

# Tamaño y performance de la Base de Datos

La arquitectura de software elegida garantiza el cumplimiento de los requisitos de capacidad de disco y latencia de la Base de Datos.

**9.1 Tiempo de respuesta en el acceso a la Base de Datos**

El sistema proveerá acceso a la información de la base de datos con un tiempo de respuesta no mayor de 10 segundos.

**9.2 Tiempo de respuesta de transacciones**

El programa no demorará más de 40 segundos en registrar una solicitud de tarea.

**9.3 Espacio en disco para el servidor de Base de datos**

El espacio en disco necesario en el Servidor para garantizar el correcto funcionamiento de la Base de Datos deberá ser como mínimo 5 GB.