Trabajo Práctico Nº 2

Documento de Arquitectura

12 de septiembre de 2013

**U**niversidad **T**ecnológica **N**acional – **F**acultad **R**egional **S**anta **F**e

**Integrantes**

De Filippis, Darío

Greca, Daniela Romina

Martínez, Juan Agustín

**Profesor:**

Castellaro, Marta

**Docentes Auxiliares:**

Domínguez, Martín Alejandro

Vrancken, Lisandro

Wasinger, Román

**Historial de Versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Autor | Principales Cambios |
| 29/08/2013 | 1.0 | De Filippis, Greca, Martinez | Creación inicial del documento. |
| 05/09/2013 | 1.1 | De Filippis, Greca, Martinez | Generación de los diagramas. |
| 11/09/2013 | 1.2 | De Filippis, Greca, Martinez | Revisión y corrección del documento. |
| 12/09/2013 | 1.3 | De Filippis, Greca, Martinez | Compleción de las secciones finales del documento. |

Contenido

[Introducción 4](#_Toc366778072)

[Propósito 4](#_Toc366778073)

[Términos y Abreviaturas 4](#_Toc366778074)

[Referencias 4](#_Toc366778075)

[Requerimientos que afectan a la Arquitectura 5](#_Toc366778076)

[Requisitos Funcionales 5](#_Toc366778077)

[Requisitos de Calidad – Priorizados 5](#_Toc366778078)

[Arquitectura de la Aplicación 6](#_Toc366778079)

[Módulos del Sistema 6](#_Toc366778080)

[Vista de Alto Nivel 6](#_Toc366778081)

[Patrones de Diseño 7](#_Toc366778082)

[Tecnologías y Criterios de Diseño Utilizados 8](#_Toc366778083)

[Capa de Presentación 8](#_Toc366778084)

[Capa de Servicios 8](#_Toc366778085)

[Capa de Acceso a Datos 8](#_Toc366778086)

[Capa de Integración 9](#_Toc366778087)

[Aspectos aplicables a toda la aplicación 9](#_Toc366778088)

[Arquitectura Tecnológica 10](#_Toc366778089)

[Arquitectura Tecnológica 10](#_Toc366778090)

[Software de base 11](#_Toc366778091)

[Arquitectura de servidores y comunicaciones 11](#_Toc366778092)

[Herramientas y Tecnologías 11](#_Toc366778093)

# Introducción

## Propósito

El presente documento tiene como objetivo presentar la especificación de arquitectura del Sistema de Gestión Organizacional de Operaciones (SiGOO) de una empresa de la región, con el fin de servir como guía para el diseño y desarrollo del mismo, y presentar su estructura general.

## Términos y Abreviaturas

Los siguientes términos y abreviaciones se utilizan en el documento:

* GUI: Interfaz Gráfica de Usuario.
* Login: Se utiliza para referir al ingreso a un sistema que maneja autorización y autenticación de usuarios.
* HTTP: Protocolo de Transferencia de Hipertexto.
* HTML: Lenguaje de Marcado de Hiper-Texto.
* PCs: Computadoras personales.

## Referencias

Los siguientes documentos contienen información relevante para este documento:

* Documento de Constitución del Proyecto.
* Documento de Alcance Inicial.

# Requerimientos que afectan a la Arquitectura

## Requisitos Funcionales

* Interfaz web accesible en forma remota desde diferentes dispositivos: PCs, laptops, tablets, smartphones.
* Gestión de solicitudes de gastos (por transporte, hoteles y otros viáticos).
* Consulta de solicitudes pendientes.
* Consulta de gastos dados de alta.
* Login diferenciado de ingreso al sistema (Supervisor/Empleado).

## Requisitos de Calidad – Priorizados

* Disponibilidad del servicio del 99,99%.
* Escalabilidad: Facilidad proporcionada por el sistema del 95% para que puedan agregarse o quitarse componentes del software manteniendo su confiabilidad.
* Performance: Respuesta acorde a la funcionalidad requerida del sistema sin bajar el nivel de servicio.
* Confiabilidad: Capacidad para evitar fallos y tolerar cualquier defecto, manteniendo las prestaciones ante la ocurrencia de estos.
* Usabilidad: El sistema responde al ritmo de trabajo del usuario, las opciones son dispuestas claramente y encontradas por el usuario de forma rápida evitando demoras.
* Modificabilidad: En la medida que el desarrollo posterior de las funcionalidades previstas se realicen sin introducir efectos secundarios.
* Seguridad: Refiere a la disponibilidad solo para usuarios autorizados.

# Arquitectura de la Aplicación

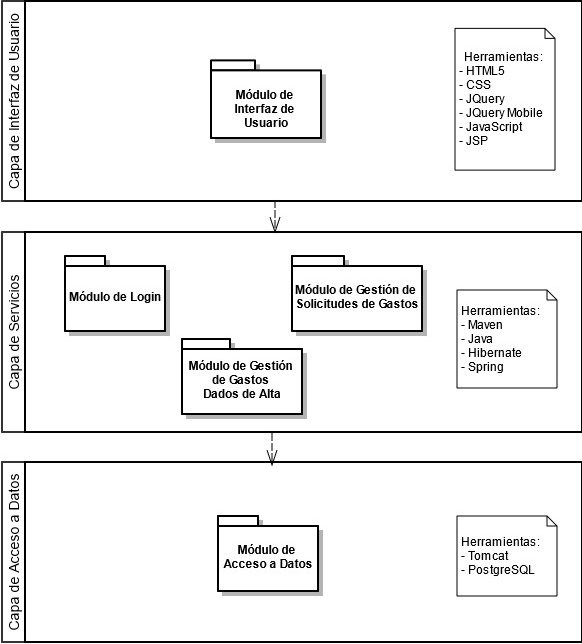
## Módulos del Sistema

* Login.
* Gestión de solicitudes de gastos.
  + Aprobación de Gastos.
  + Consulta de solicitudes pendientes.
* Interfaz gráfica.
* Gestor de Base de Datos.

## Vista de Alto Nivel

Para la construcción de este sistema se utilizará una arquitectura de 3 capas. Este estilo de arquitecturas permite generar aplicaciones flexibles, reusables y de fácil agregado de cambios, siendo una buena elección debido a la necesidad de extender la aplicación en el corto plazo para proveer otras funcionalidades para la empresa.

Los diferentes módulos de la aplicación serán distribuidos de la siguiente manera:



* **Capa de interfaz de usuario:** Brinda al usuario la interfaz del sistema a través de la cual va a interactuar. El usuario se conecta a esta capa por medio del protocolo HTTP y esta le provee la respuesta en formato HTML.
* **Capa de lógica de negocio:** Es la encargada de realizar todo el procesamiento relativo a las reglas del negocio a implementar. Provee los servicios necesarios para responder a las solicitudes a la capa de interfaz de usuario.
* **Capa de acceso a datos:** Posee la lógica para obtener los datos de las bases de datos y transferirlos a la capa de lógica de negocio para su procesamiento.

Cada una de estas posee una implementación independiente; proveen servicios a la capa superior y utilizan los servicios de la inferior a través de interfaces bien definidas, lo que nos da la posibilidad de modificar cada una de ellas sin afectar a las demás.

## Patrones de Diseño

* Singleton: [A determinar.]
* DTO: Este patrón tendrá como objetivo la transferencia de datos entre los distintos subsistemas de la aplicación, y serán usados en conjunto con los DAO.
* DAO: Se utilizará este patrón para proveer acceso a datos a través de un módulo que maneje el acceso a la base de datos y los mecanismos de persistencia.
* Facade: [A determinar.]
* Factory: [A determinar.]
* Inyección de dependencias: [A determinar.]
* AOP: Para manejo de transacciones y excepciones a lo largo de la aplicación.

# Tecnologías y Criterios de Diseño Utilizados

## Capa de Presentación

Funciones que proporciona:

* Resolver la interacción con el usuario: Provee un conjunto de interfaces gráficas de usuario (GUI) consistentes, tanto para empleados como para supervisores, con diferenciación visual de cada tipo de usuario.
* Facilitar el acceso a datos de parte del usuario.
* Para cada tipo de dispositivo (PCs, Laptops, Tablets, Smartphones), se diseña una interfaz que:
  + Se adecúe al tipo de interacción del usuario con el mismo.
  + Presente los datos en cantidad y disposición acorde a las dimensiones de este.

Interacción con Otras Capas:

* Capa de Servicios: Hace uso de los servicios proporcionados delegando la ejecución de operaciones a la misma, recibiendo los datos que deben ser mostrados en pantalla al usuario.

## Capa de Servicios

Funciones que Proporciona:

* Proveer un conjunto de operaciones genéricas, llamadas “servicios”, para la implementación de los requerimientos representados a través de casos de uso.
* Ofrecer estos servicios a través de interfaces simples y semánticas, haciendo que su implementación sea transparente para los usuarios.
* Realizar la manipulación de los datos a través de múltiples y complejas operaciones sobre un superconjunto de datos que encapsula, obteniendo un resultado refinado para el usuario.

Interacción con Otras Capas:

* Capa de Presentación: Provee los servicios del sistema mediante una interfaz que encapsula los procedimientos.
* Capa de Acceso a Datos: Utiliza la interfaz de esta capa para acceder a la información almacenada en la base de datos.

## Capa de Acceso a Datos

Funciones que Proporciona:

* El acceso simplificado a los datos almacenados en la base de datos. Encapsula totalmente la persistencia, permitiendo este manejo simple y de fácil uso.
* Provee el soporte para las transacciones realizadas sobre los objetos del modelo: inserción, modificación, eliminación, listado, etc.
* Simplifica el modelo de datos.
* Soporta diferentes motores de bases de datos y distintas versiones.

Interacción con Otras Capas:

* Capa de Servicios: Provee una interfaz para el acceso a los datos almacenados.

## Capa de Integración

Funciones que proporciona:

* Comunicación explícita con sistemas externos.
* Servicios web: [A determinar.]

## Aspectos aplicables a toda la aplicación

En esta sección se especificarán en detalle todas aquellas cuestiones que involucren a toda la aplicación.

**Estrategias de test**

Se procederá a la realización de pruebas automatizadas mediante la utilización de la herramienta JUnit.

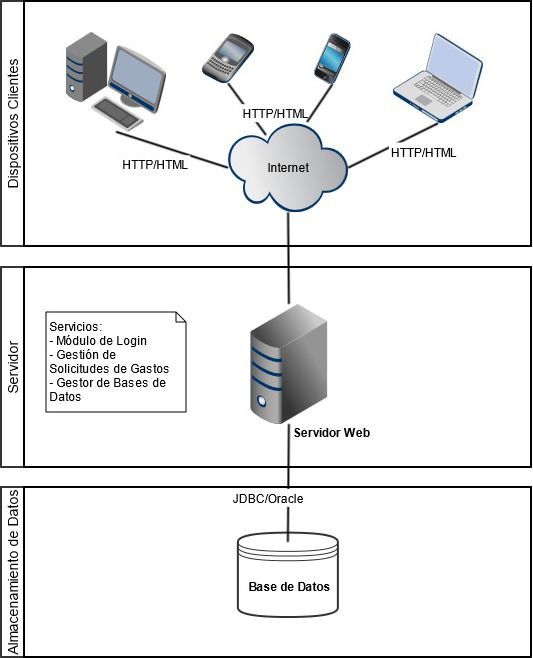
**Solución para las funcionalidades típicas del sistema**

[A determinar.]

# Arquitectura Tecnológica

## Arquitectura Tecnológica

La arquitectura se basará en un modelo **Cliente-Servidor** como se muestra en la imagen a continuación. En ella tenemos separada la lógica de presentación de datos (lado cliente), de la lógica de procesamiento y almacenamiento de datos (lado servidor).



## Software de base

**Sistema operativo**

Linux.

Android.

iOS.

Blackberry.

Windows pone.

**Navegadores Web soportados**

Chrome 29.0.

Mozilla Firefox 23.0.1

**Plugins**

[A determinar.]

## Arquitectura de servidores y comunicaciones

**Motor de base de datos**

La base de datos a utilizar será PostgreSQL 9.3, que es el motor de base de datos que actualmente se está utilizando en la empresa.

**Servidor de aplicación**

El servidor de aplicación será Apache Tomcat 7.0.

**Protocolos de comunicación**

Se utilizará JDBC/Oracle para la comunicación con la base de datos.

La interacción con los usuarios será a través del protocolo HTTP.

## Herramientas y Tecnologías [A determinar]

* SVN.
* Maven.
* Java.
* JUnit.
* Hibernate.
* Spring.
* Servlets y JSP.
* Tomcat.
* Servidor de integración continua.
* PostgreSQL.
* Herramientas UML.