Plataforma para la gestión de viajes

Software Architecture Document

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/sept/2015 | 1.0 | Diligenciamiento del documento | Luis Felipe Hernandez Gomez |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

1.5 Overview

2. Architectural Representation

3. Architectural Goals and Constraints

4. Use-Case View

5. Logical View

5.1 Overview

5.2 Architecturally Significant Design Packages

5.3 Use-Case Realizations

6. Process View

7. Deployment View

8. Implementation View

8.1 Overview

8.2 Layers

9. Data View (optional)

10. Size and Performance

11. Quality

Software Architecture Document

# Introduction

Este documento se enfoca en la arquitectura global del Sistema utilizando herramientas a muy alto nivel sobre patrones arquitectonicos, enfoques como el capability, rup, estandares de calidad bajo la norma iso 27000. Ademas de utilizar framework basado en el modelo mvc para optimizar el rendimiento del desarrollo del Proyecto web.

## Purpose

Este documento proporciona una visión general de arquitectura global del sistema, utilizando una serie de puntos de vista diferentes de arquitectura para representar diferentes aspectos del sistema. Se tiene la intención de captar y transmitir las decisiones arquitectónicas significativas que se hayan introducido en el sistema.

## Scope

El alcance de este documento involucra a todos los demás documentos de esta fase inicial, además de que puede estar sujeto a modificaciones durante el desarrollo de diferentes fases siempre será aprobado por el cliente y servirá de guía y referente de alto nivel del proyecto.

El alcance de este documento se ve afectado por los requerimientos funcionales de la aplicación para el proceso de solicitud de comisión de servicios del area de contabilidad, este documento aplica para la fase de ejecución del Proyecto, asi como una guia para desarrollar el sistema en gestión.

## Definitions, Acronyms, and Abbreviations

Stakeholders: Involucrados, interesados y/o afectados por el proyecto tanto directa como indirectamente.

Framework de desarrollo:  es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos desoftware concretos, que puede servir de base para la organización y desarrollo de [software](https://es.wikipedia.org/wiki/Software).

## References

*[This subsection provides a complete list of all documents referenced elsewhere in the* ***Software Architecture Document****. Identify each document by title, report number (if applicable), date, and publishing organization. Specify the sources from which the references can be obtained. This information may be provided by reference to an appendix or to another document.]*

## Overview

En este documento se encontrará los patrones seleccionados para el Sistema, casos usos, procesos de desarrollo, se especifica detalladamente los requerimientos no funcionales del sistema, tambien se propone el trato de los datos en función de la organización y se vera como se manejara la aplicación en las diferentes vistas que se enfoca el 4+1.

# Architectural Representation

El Sistema debera proveer a los usuarios un criterio de aceptacion para las solicitudes que estos envien al respectivo funcionario, es requisito que la información detallada en el sistema solo lo podra manejar el rol encargado de dar aval de dichas solicitudes, el Sistema debe aprobar las solicitudes cuyos atributos estan bien diligenciados de lo contrario no se debe dejar mandar la solicitud ademas que debera estar disponible en la plataforma durante las horas laboradas en la semana de lunes a Viernes.

Puntos de vista a tener en cuenta:

* El formulario debera ser diligenciado en un plazo de antelacion de 15 dias habiles del viaje
* El Sistema tendra diferentes roles entre los cuales estan los administrativos, los docentes, la administradora, el rector, los cuales tienen que dar el aval de la solicitud de comision de servicios entre los cuales tambien se define la contable y la tesorera quienes son las encargadas de aprobar los viaticos de cada viaje.
* El docente o administrativo debera poder gestionar la solicitud de comisión de servicios
* Es importante contar con la aceptación o firma del jefe inmediato del gestionante de la solicitud, el rector y del area de contabilidad para poder aceptar los viaticas. Solo los docentes y administrativos de planta podran obtener viaticos para los viajes
* Cada viaje debera poseer descripción,fecha,lugar,valor para llevarse acabo.
* La vista logica esta conformada por solicitud,viajes, administradora,pagos
* La vista de despliegue tendremos components de autenticación, pruebas, reportes para cada uno de las entidades core del Sistemas y por ultimo un componente de pagos para llevar un control sobre la cantidad de viaticos y gastos.
* La vista de procesos, se tendra en cuenta la importancia del manejo de las solicitudes ya que el Sistema no puede dejar mandar aquellas solicitudes que no tengan todos los datos ademas es importante que el administrador pueda revisarla y darle el visto bueno de la Universidad, el Sistema debe estar disponible en los horarios de trabajos asignados para los trabajadores publicos de lunes a Viernes
* La vista física, en nuestro entorno se necesitara equipos con procesadores amd o intel con una freccuencia mayor a 1.7 ghz, se utilizara el lenguaje de programación php con el framework symfony y utilizacion de herramientas como css y javascript, etc.

# Architectural Goals and Constraints

La plataforma para gestion de viajes de la Universidad del Quindío es un software que debe cumplir con los siguientes requisites:

* Poder diligenciar la solicitud de comision de servicios
* Ver la aprobación de la solicitud diligenciada
* La administradora debe aprobar las solicitudes diligenciadas
* El rector debera aprobar las solicitudes diligenciadas
* El decano de la facultad en la cual esta adscrito el docente debera aprobar las solicitudes diligenciadas
* Generar reportes referentes a los pagos de cada facultad

La plataforma debera permitir aprobar las solicitudes dependiendo si es profesor o administrative de acuerdo a la aprobacion del jefe inmediato, rector y del area de contabilidad de la Universidad y despues debera mostrar al solicitante la aprobación de la Universidad para el viaje. El Sistema debe estar disponible en los horarios establecidos para los administrativos de lunes a Viernes las 24 horas del dia.

Dicho todo esto es fundamental que el entorno de desarrollo del Proyecto cuente con herramientas de reutilización de codigo muy potentes que ayuden a optimizar el tiempo y beneficio de la organización tal como es el caso del framework symphony el cual provee de varias herramientas necesarias para el Proyecto tales como pruebas, generación de reportes, etc. El equipo cuenta con dos desarrolladores uno encargado para la parte de generación de reportes, monitoreo y pruebas y el otro encargado de los modulos de pagos,gds y autenticación

# Use-Case View

Los casos de uso son los siguientes:

* El usuario debe loguearse para entrar a la plataforma
* El usuario podra diligenciar el formato de comision de servicios
* El area de contabilidad podra aprobar la solicitud de comision de servicios
* El Sistema debera verificar el jefe inmediato del usuario logueado
* El Sistema debera mandar la solicitud al jefe inmediato,rector y al area de contabilidad

# Logical View

La arquitectura debera soportar la gran cantidad de solicitudes de comision de servicios para la aprobación de este, asi mismo deberá aprobar aquellas que estan totalmente diligenciadas, debe poder mostrar al usuario el estado de su solicitud. Al momento de la aprobación de la solicitud se debe verificar los viaticos pedidos para el viaje y aprobarlos por medio de tesoreria Este Proyecto esta compuesto por los siguientes paquetes y clases:

* Solicitud
* Contabilidad
* Viajes
* Tesoreria
* Administrativos
* Docentes
* Reportes

## Overview

A continuación se muestra detalladamente cada paquete y su funcion dentro del Proyecto:

* Solicitud: Esta es la encargada de tener toda la información sobre los formatos diligenciados y enviados a sus respectivas areas
* Contabilidad: Es la encargada de aprobar y hacer la gestion sobre la solicitud registrada y es la de mayor jerarquia en el sistema
* Viajes: Esta es la encargada de llevar a cabo toda la información referente a los viajes que realicen los solicitantes, asi como el transporte aereo que de destino
* Tesoreria: Es la encargada de dar soporte a los viaticos repartidos en los diferentes viajes y solicitudes
* Administrativos: Es el encargado de diligenciar el formato de comision de servicios para su aprobación, el Sistema guardara estos datos en la base de datos
* Docentes: Este diligenciará el formato para su aprobación.
* Reportes: Este debe mostrar todos los reportes referente a los docentes y administrativos que viajaron, asi como el pago de los viaticos, etc.

## Architecturally Significant Design Packages

Los paquetes mas importantes de la arquitectura son los siguientes:

* Solicitud: Este paquete hace referencia a la solicitud que hacen los docentes o administrativos para viajar, es importante que tengan en cuenta los objetivos del viaje y si necesitan viaticos o no, para este apartado solo los docentes o administrativos de planta podran hacer uso de los viaticos
* Contabilidad: Este paquete hace referencia a la encargada de aprobar las solicitudes, tambien ella es la que decide si hacer uso de los viaticos o no.
* Viajes: Este paquete es el encargado de obtener toda información acerca de los viajes, fecha, destino, eventos en el cual se va a participar entre otras cosa muy importantes
* Administrativos: Este paquete tendra toda la informacion de los administrativos que estan en viajes
* Docentes: Este paquete tendra toda la informacion de los docentes que estan en viajes

## Use-Case Realizations

* El usuario debera ingresar a la plataforma mediante un usuario y contraseña
* El Sistema debera reconocer que rol cumple (administrativo, docente,contabilidad,rector,tesorero)
* El usuario podra solicitar una comision de servicios presionando el boton de solicitud de comision
* Al diligenciar la solicitud y presionar el boton de enviar el Sistema debe mostrar un mensaje diciendo solicitud enviada si esta bien formulada, de lo contrario este debera mostrar que hizo mal
* Contabilidad podra ver el listado de solicitudes que le han llegado al Sistema, entrando al Sistema y presionando ver listado de solicitudes
* Contabilidad podra aprobar la solicitud mediante el boton de aprobar solicitud y el Sistema tiene que verificar el aval de todos los implicados del proceso (rector,administrativo,docente,contabilidad)

# Process View

El Sistema debe de entender quien se loguea, su cargo y funciones, en el caso de un docente el Sistema debe entender quien es el jefe inmediato de el asi como si es de planta o no, es necesario que al momento de diligenciar la solicitud de comisión de servicios los datos sean lo mas claros posibles y el envio de este solo sea de accuerdo a la aceptación del Sistema, el Sistema solo debe estar disponible de lunes a Viernes para el buen funcionamiento de este

# Deployment View

Es importante que el Sistema este configurado en un servidor debian con php y mysql lo que permite mayor estabilidad de la plataforma, con una ram de 16 gb y un disco duro mayor a 500 gb. Ya que los reportes en general se deben hacer diariamente es por esto que se debe tener amplio espacio en gb para la informacion generada, ademas de soportar la gran cantidad de solicitudes diligenciadas.

# Implementation View

La arquitectura necesita que se puedan interactuar los datos de una forma para diligenciar el formulario, el modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado de aprobación (habitualmente desde el controlador).

## Overview

Los componentes principales de la arquitectura son:

Modelo: Se encarga que la plataforma no deje enviar aquellas solicitudes mal diligenciadas o mal redactadas, asi como aprobar las solicitudes de cada persona

Vista: Maneja la visualización de la información, los listados de las solicitudes, los viaticos aprobados, etc.

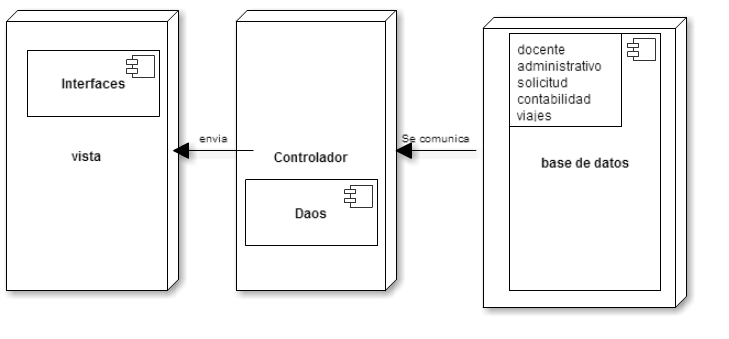
Controlador: El controlador depende del modelo para poder funcionar adaptarse a los diferentes requerimientos de la plataforma.

## Layers

La arquitectura posee las siguientes capas con sus respectivas componentes:

* Modelo: solicitud, contabilidad, viajes, pagos, docentes, administrativos,
* Vista: Isolicitud,Icontabilidad,Iviajes,Ipagos,Idocentes,Iadministrativos
* Controlador: todo lo referente con la logica del negocio

**MODELO DE COMPONENTES**



# Data View (optional)

Los datos que ingresen a la plataforma se guardaran en una base de datos relacional, la cual nos permitira tener control sobre aspectos de reportes, registros, solicitudes enviadas , aprobadas etc. Los datos deben tener una persistencia que permitan ser entendibles para la persona encargada

# Size and Performance

El software debera poder identificar el usuario logueado y poder diligenciar formato de comisión de servicios para poder aprobar la solicitud es necesario que el formulario este completamente diligenciado ademas que debe permitir ver el estado actual de la solicitud, contabilidad debera poder aprobar las solicitudes que cumplan con todo lo estipulado en el proceso de la Universidad, ademas de asignarle los viaticos de acuerdo con lo estipulado, el Sistema no podra aprobar solicitudes sin el aval del rector ni del jefe inmediato al funcionario que diligencie la solicitud, solo los funcionarios de planta podran acceder al recurso de viaticos y solo se aceptara la solicitud si este es llenado con 15 dias de antelación de acuerdo con el cronograma de la programacion del evento a asistir.

# Quality

El Sistema estara disponible los dias laborales de la semana 24 horas del día, ademas de estar desarrollado para ahorrarle tiempo al funcionario de rechazar solicitudes mal diligenciadas, sin objetivos claros, sin especificar el sitio del evento entre otros factores que ayudaran a la persona encargada a poder verificar solo aquellas solicitudes que fueron aceptadas y enviadas a revision