Plataforma para la gestión de viajes

Software Architecture Document

Version 2.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/sept/2015 | 1.0 | Diligenciamiento del documento | Luis Felipe Hernandez Gomez |
| 29/sept/2015 | 2.0 | Cambios y mejoras al documentos | Luis Felipe Hernández Gómez, Dwan Córdoba |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Introduction 4

1.1 Purpose 4

1.2 Scope 4

1.3 Definitions, Acronyms, and Abbreviations 4

1.4 References 4

1.5 Overview 4

2. Architectural Representation 4

3. Architectural Goals and Constraints 4

4. Use-Case View 4

5. Logical View 4

5.1 Overview 4

5.2 Architecturally Significant Design Packages 4

5.3 Use-Case Realizations 4

6. Process View 4

7. Deployment View 4

8. Implementation View 4

8.1 Overview 4

8.2 Layers 4

9. Data View (optional) 4

10. Size and Performance 4

11. Quality 4

Software Architecture Document

# Introduction

Este documento se enfoca en la arquitectura global del Sistema utilizando herramientas a muy alto nivel sobre patrones arquitectónicos, enfoques como el CAPABILITY, RUP, estándares de calidad bajo la norma ISO 27000. Además 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 área de contabilidad, este documento aplica para la fase de ejecución del Proyecto, así como una guía 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 de software 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, también se propone el trato de los datos en función de la organización y se verá cómo se manejara la aplicación en las diferentes vistas que se enfoca el 4+1.

# 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.]

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

Vista lógica: En la vista lógica está ubicada todo lo que son entidades CORE y relaciones entre las entidades, entre las cuales tenemos las siguientes entidades:

* Solicitante
* Solicitud
* Superior Inmediato al solicitante
* Comisión
* Pagos
* Contabilidad
* El formulario deberá ser diligenciado en un plazo de antelación de 15 días hábiles del viaje
* El Sistema tendrá diferentes roles entre los cuales están los administrativos, los docentes, la administradora, el rector, los cuales tienen que dar el aval de la solicitud de comisión de servicios entre los cuales también se define la contable y la tesorera quienes son las encargadas de aprobar los viáticos de cada viaje.
* El docente o administrativo deberá poder gestionar la solicitud de comisión de servicios
* Es importante contar con la aceptación o firma del jefe inmediato del solicitante, el rector y del área de contabilidad para poder aceptar los viaticas. Solo los docentes y administrativos de planta podrán obtener viáticos para los viajes
* Cada viaje deberá poseer descripción, fecha, lugar, valor.
* La vista lógica está conformada por solicitud, viajes, administradora, pagos
* La vista de despliegue tendremos componentes 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 viáticos y gastos.
* La vista de procesos, se tendrá 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 además 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 públicos de lunes a Viernes
* La vista física, en nuestro entorno se necesitara equipos con procesadores AMD o INTEL con una frecuencia mayor a 1.7 GHz, se utilizara el lenguaje de programación PHP con el FRAMEWORK SYMFONY y utilización de herramientas como CSS y JAVASCRIPT, etc.

# Architectural Goals and Constraints

[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.]

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

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

La plataforma deberá permitir aprobar las solicitudes dependiendo si es profesor o administrativo de acuerdo a la aprobación del jefe inmediato, rector y del área de contabilidad de la Universidad y después deberá 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 día.

Dicho todo esto es fundamental que el entorno de desarrollo del Proyecto cuente con herramientas de reutilización de código 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 módulos de pagos, GDS y autenticación

# Use-Case View

[This section lists use cases or scenarios from the use-case model if they represent some significant, central functionality of the final system, or if they have a large architectural coverage—they exercise many architectural elements or if they stress or illustrate a specific, delicate point of the architecture.]

Los casos de uso son los siguientes:

* El usuario debe iniciar sesión para entrar a la plataforma
* El usuario podrá diligenciar el formato de comisión de servicios
* El área de contabilidad podrá aprobar la solicitud de comisión de servicios
* El Sistema deberá verificar el jefe inmediato del usuario conectado.
* El Sistema deberá mandar la solicitud al jefe inmediato, rector y al área de contabilidad

# Logical View

[This section describes the architecturally significant parts of the design model, such as its decomposition into subsystems and packages. And for each significant package, its decomposition into classes and class utilities. You should introduce architecturally significant classes and describe their responsibilities, as well as a few very important relationships, operations, and attributes.]

Diagrama de clases



## Overview

[This subsection describes the overall decomposition of the design model in terms of its package hierarchy and layers.]

Diagrama de Paquetes



## Architecturally Significant Design Packages

[For each significant package, include a subsection with its name, its brief description, and a diagram with all significant classes and packages contained within the package.

For each significant class in the package, include its name, brief description, and, optionally, a description of some of its major responsibilities, operations, and attributes.]

Los paquetes más importantes de la arquitectura son los siguientes:

* Login: Todos los actores del sistema incluyendo tesorera, el rector, el docente o administrativo debe ingresar al sistema mediante un usuario y una contraseña
* Sistema de Solicitudes: 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 viáticos o no, para este apartado solo los docentes o administrativos de planta podrán hacer uso de los viáticos
* Sistema de aprobación: Este paquete hace referencia a la aprobación las solicitudes diligenciadas
* Sistema de aprobación de pagos: La tesorera debe de llevar un registro de todos los viáticos que se han pedido para las diferentes comisiones.

## Use-Case Realizations

[This section illustrates how the software actually works by giving a few selected use-case (or scenario) realizations, and explains how the various design model elements contribute to their functionality.]

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

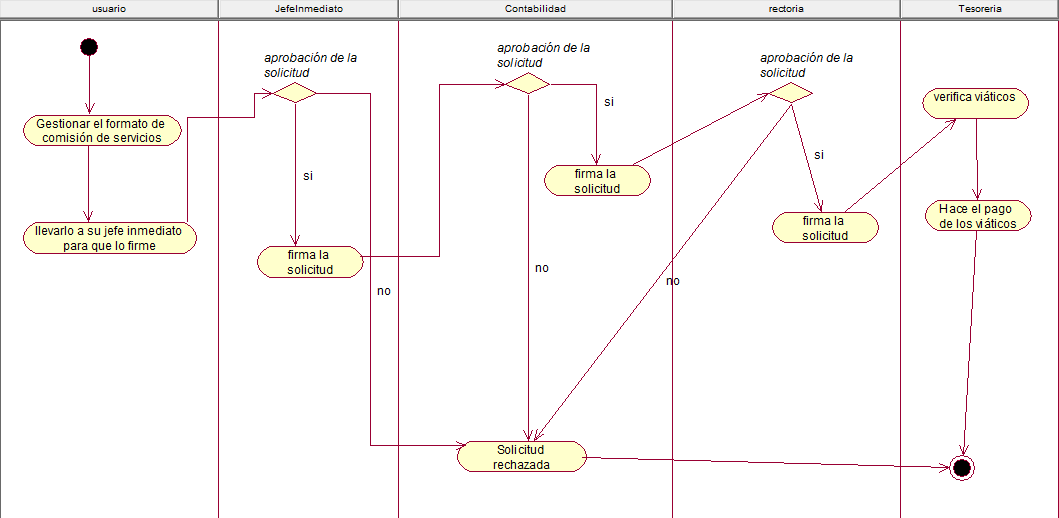
# Process View

[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.]

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

Diagrama de actividades:

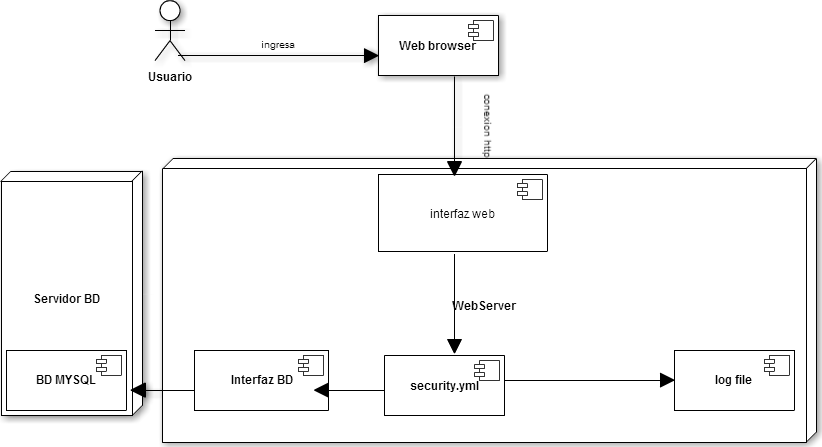
Actividad Solicitar la comisión de servicios de permanencia



# 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.]

Diagrama de components:



# 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.]

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

[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. ]

Los componentes principales de la arquitectura son:

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

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

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

## Layers

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

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 lógica del negocio

# 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.]

Los datos que ingresen a la plataforma se guardaran en una base de datos relacional, la cual nos permitirá 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

[A description of the major dimensioning characteristics of the software that impact the architecture, as well as the target performance constraints.]

El software responde al proceso de gestión de comisión de servicios de la universidad del Quindío, dispone de un sistema que en promedio será capaz de atender por el momento de 10 usuarios en línea.

# 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.]

El Sistema estará disponible los días laborales de la semana 24 horas del día, además 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 revisión