Sistema de Información Integrado Colciencias – Modulo de Seguimiento

Software Architecture Document

Versión 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/Sep/2014 | 1.0 | Release Inicial | Nelson Andres Castillo Saza Gustavo Adolfo Lopez |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. [Introducción](#_Introducción) 4

1.1 [Purpose](#_Propósito) 4

1.2 [Scope](#_Alcance) 4

1.3 References 4

1.4 [Overview](#_Generalidades) 4

2. [Architectural Representation](#_Architectural_Representation) 4

3. [Architectural Goals and Constraints](#_Metas_Arquitectónicas_y) 4

4. [Use-Case View](#_Use-Case_View) 4

5. [Logical View](#_Logical_View) 4

5.1 [Overview](#_Overview) 4

5.2 [Architecturally Significant Design Packages](#_Architecturally_Significant_Design) 4

5.3 [Use-Case Realizations](#_Use-Case_Realizations) 4

6. [Process View](#_Process_View) 4

7. [Deployment View](#_Deployment_View) 4

8. [Implementation View](#_Implementation_View) 4

8.1 [Overview](#_Overview_1) 4

8.2 [Layers](#_Layers) 4

9. [Size and Performance](#_Size_and_Performance) 4

10. [Quality](#_Quality) 4

Software Architecture Document

# Introduction

Este documento provee una visión de alto nivel sobre la arquitectura de software del módulo de seguimiento perteneciente al sistema de información integrado de la comunidad Colciencias, describiéndose las tecnologías a ser utilizadas para la correcta verificación de la validez de los convenios de Colciencias así como de los contratos y el cumplimiento de las obligaciones de las partes claves en este proceso, además de la comprobación de los resultados para la financiación que se otorga.

Este documento además provee una descripción de alto nivel para las metas de la arquitectura, los casos de uso que soportan el sistema y los elementos arquitectónicos elegidos para el mejor alcance de los casos de uso. Debido a esto se nos permite entonces el diseño de criterios y documentos que definen los estándares a ser usados en el desarrollo del módulo de seguimiento.

## Purpose

El propósito de este documento es brindar una visión comprensible de la arquitectura general del sistema debido a que el módulo de seguimiento abarca las funcionalidades que son necesarias para dar soporte a cada uno de los proyectos beneficiarios de una convocatoria mediante seguimiento técnico y financiero. Esto se hace con la intención de captar y transmitir las decisiones significativas que hayan sido introducidas o se vayan a introducir en el sistema.

## Scope

El módulo de seguimiento contempla las funcionalidades de Contratación mediante el soporte a las actividades necesarias para la legalización de un contrato y la liberación de recursos financieros, Seguimiento técnico, seguimiento financiero y la definición del seguimiento más apropiado para un proyecto. Por este motivo decimos entonces que el modulo se relaciona con los procesos de convocatorias, informes y proyectos.

## References

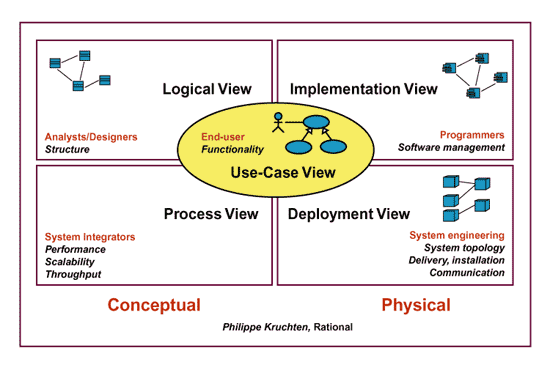
* Sistema de Información Integrado Especificaciones funcionales y técnicas – Comunidad Colciencias. Abril de 2014. Departamento administrativo de Ciencia, Tecnología e Innovación COLCIENCIAS.

## Overview

Este documento cuenta con una breve descripción de los subsistemas con los que cuenta el módulo de seguimiento, describiendo los diferentes diagramas utilizados en su modelado.

# Architectural Representation

La arquitectura de la aplicación es representada pos las vistas definidas en el modelo “4+1” pero siguiendo las recomendaciones de nomenclatura Rational Unified Process (RUP) dividiéndola en las siguientes cinco vistas, cada una con sus respectivos diagramas UML aplicados según sea el caso:



* **Use-case View:** Describe los actores y casos de uso para el sistema, esta vista presenta las necesidades del usuario y es elaborada a un nivel de diseño para describir los flujos y las limitaciones en más detalle.
* **Logical View:** La vista lógica muestra una visión general de las partes más significativas del sistema mediante la descomposición en subsistemas y paquetes. Aquí también es posible observar las relaciones entre los diversos componentes y sus dependencias.
* **Process View:** La vista de procesos describe las tareas que participan en la ejecución del sistema, sus interacciones y configuraciones. También describe la asignación de objetos y clases a las tareas.
* **Deployment View:** La vista de despliegue muestra la asignación de los elementos de la vista lógica en nodos de procesamiento físico, además de la configuración de la red física entre los nodos. Contiene el modelo de despliegue.
* **Implementation View:** La vista de implementación muestra los componentes principales y sus asociaciones, además de las librerías dependientes. Contiene el modelo de componentes.

# Architectural Goals and Constraints

El Sistema de Información integrado Comunidad Colciencias tiene definidos varios requisitos no funcionales que deben ser alcanzados y que también aplican para el módulo de seguimiento Ver “Tabla: Requerimientos no funcionales”, *Sistema de Información Integrado. Especificaciones funcionales y técnicas – Comunidad Colciencias*, pág. 796.   
Para el módulo de seguimiento en particular consultar la Pág. 43 del documento anteriormente mencionado.

Adicionalmente a esto, el modulo debe ser construido completamente bajo el patrón de arquitectura de software Modelo-Vista-Controlador (MVC).

Otra característica importante para este módulo es que debido al constante manejo de formularios es importante que estos cuenten con seguridad contra ataques de tipo Cross-site Scripting.

# Use-Case View

Los casos de uso clave para el módulo de seguimiento en particular se encuentran definidos en el documento “Diagrama general”, *Sistema de Información Integrado. Especificaciones funcionales y técnicas – Comunidad Colciencias*, pág. 521.

Los casos de uso a ser tratados específicamente en este equipo de trabajo son 7 (SEG10 – SEG16) y se encuentran detallados desde la página 546 hasta la página 561 del documento anteriormente mencionado.

# Logical View

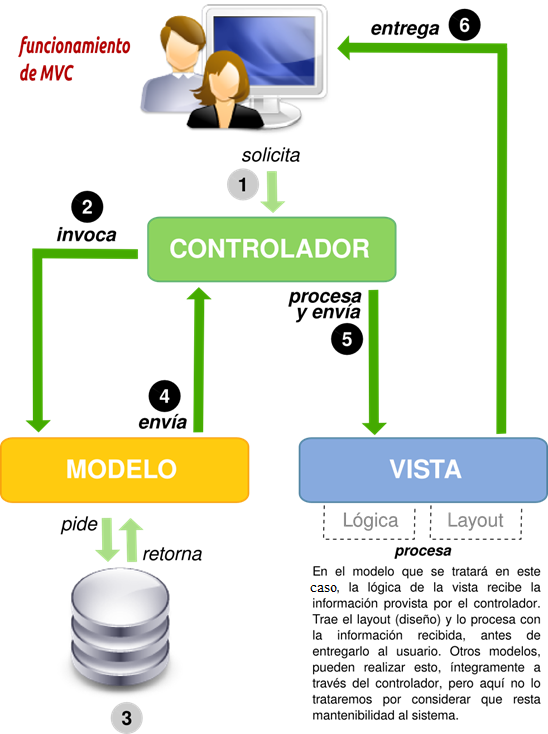
La organización conceptual del sistema es presentado en la vista lógica en términos de capas, paquetes, subsistemas y clases.

## Overview

La aplicación basada en el módulo de seguimiento será desarrollada bajo el patrón de arquitectura Modelo Vista Controlador (MVC).

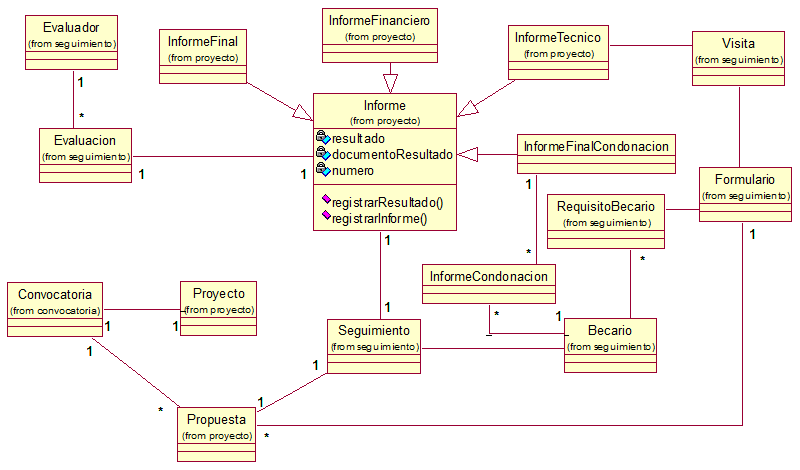
Este patrón ha sido elegido debido a esta orientado a la reutilización de código y la separación de conceptos, lo cual facilita el desarrollo, además del mantenimiento, extensibilidad y comunicación con otros módulos del sistema.

* El **Modelo** es la representación de los datos con los que el sistema interactúa. Por tanto controla los accesos a esta información y se encarga de facilitarla para que pueda ser presentada al usuario mediante la vista. Las peticiones al modelo se hacen por medio del controlador.
* El **Controlador** atiende peticiones. responde a eventos, solicita y procesa información. Es un mediador entre el modelo y la vista.
* La **Vista** presenta la información proveniente del modelo y el controlador en un formato adecuado para la interacción con los datos. Generalmente hace referencia a interfaces graficas de usuario como plantillas en HTML.



**Diagramas de clases:**

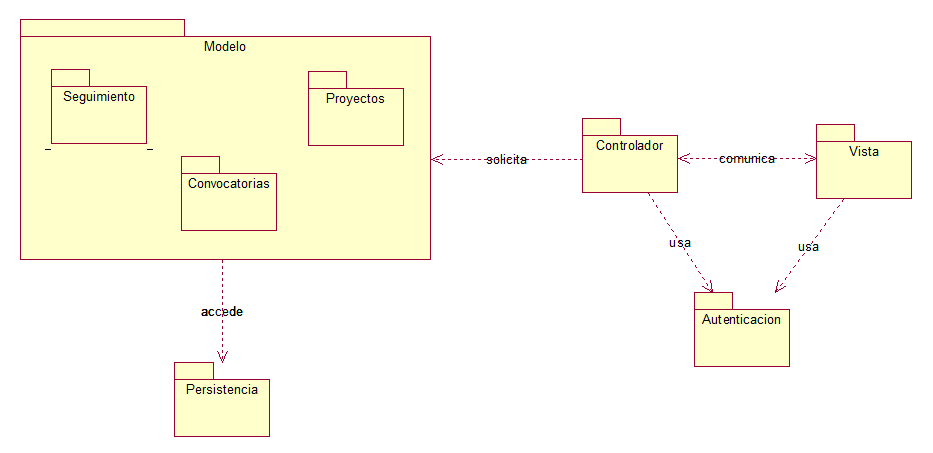
Por otro lado el diagrama de clases correspondiente al módulo de seguimiento ha sido definido en el documento “Seguimiento”, *Sistema de Información Integrado. Especificaciones funcionales y técnicas – Comunidad Colciencias*, pág. 783. Sin embargo este diagrama de clases es muy amplio debido a que contempla todas las clases necesarias para el desarrollo de los 20 casos de uso contenidos dentro del módulo de seguimiento. Es por este motivo que se hecho un rediseño a muy alto nivel del diagrama de clases existente contemplando únicamente los 7 casos de uso correspondientes (SEG10 – SEG16). Para ver más detalladamente los atributos y métodos que conforman cada clase por favor ver el diagrama de clases del documento anteriormente dicho.



*Diagrama de clases a muy alto nivel para los 7 casos de uso particulares del módulo de seguimiento en este grupo de trabajo.*

Es importante aclarar que este diagrama de clases no es el único al que se debe hacer referencia desde el módulo de seguimiento pues algunos casos de usos, específicamente los casos de uso SEG11 (“Registrar informe visita”, *Sistema de Información Integrado. Especificaciones funcionales y técnicas – Comunidad Colciencias*, pág. 548) y SEG12 (“Tabla: Requerimientos no funcionales”, *Sistema de Información Integrado. Especificaciones funcionales y técnicas – Comunidad Colciencias*, pág. 553), cuya complejidad es alta y tienen un alcance demasiado amplio debido a que se conectan con otros módulos externos de la aplicación.; así que es para estos casos de uso es importante apoyarse en las estructuras propuestas en los diagramas de clases para “Convocatorias y Proyectos”, *Sistema de Información Integrado. Especificaciones funcionales y técnicas – Comunidad Colciencias*, pág. 762.

## Architecturally Significant Design Packages



La imagen anterior representa el modelo de los paquetes más significativos para el módulo de seguimiento y sus las relaciones entre ellos.

1. **Vista:** El paquete de la vista es el encargado de almacenar todos los elementos que serán renderizados en interfaces graficas de usuario como lo son elementos HTML, CSS y Javascript.
2. **Autenticación:** Este paquete es el responsable de la lógica para el manejo de usuarios, desde su creación hasta el manejo de sus sesiones.
3. **Controlador:** En el controlador se almacenan todas las clases encargadas de la lógica del negocio, además de los elementos que se encargan de atender y redirigir las peticiones realizadas por la vista.
4. **Modelo:** El paquete modelo almacena a su vez más paquetes para distribuir el esquema de clases.
   1. **Seguimiento:** Este paquete contiene todos los objetos que son propios del módulo de seguimiento.
   2. **Convocatorias:** Este paquete contiene todos los objetos que pertenecen al módulo externo de convocatorias pero que se relacionan directamente con el módulo de seguimiento y son necesarios para llevar a cabo los casos de uso.
   3. **Proyectos:** Este paquete contiene todos los objetos que pertenecen al módulo externo de proyectos pero que se relacionan directamente con el módulo de seguimiento y son necesarios para llevar a cabo los casos de uso.
5. **Persistencia:** En este paquete se almacena el esquema de la base de datos que para este módulo estará basada en PostgreSQL.

## Use-Case Realizations

La realización de los 7 casos de uso correspondientes al módulo de seguimiento se encuentra definidos en los siguientes documentos adjuntos:

[SEG10 - Evaluar Informes Técnicos Financieros](use_case_SEG10_Evaluar_Informes_Tecnicos_Financieros.docx)

[SEG11 - Registrar Informe Visita](use_case_SEG11_Registrar_Informe_Visita.docx)

[SEG12 - Registro Seguimiento Becarios](use_case_SEG12_Registro_Seguimiento_Becarios.docx)

[SEG13 - Verificar Requisitos Becarios](use_case_SEG13_Verificar_Requisitos_Becarios.docx)

[SEG14 - Registrar Informes Condonación](use_case_SEG14_Registrar_Informes_Condonacion.docx)

[SEG15 - Asignar Evaluador Para Condonación](use_case_SEG15_Asignar_Evaluador_Para_Condonacion.docx)

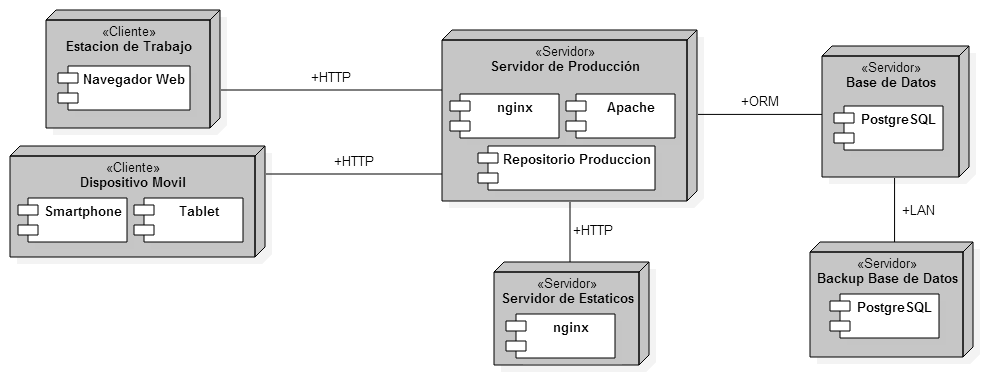
[SEG16 Evaluar Informes Condonación](use_case_SEG16_Evaluar_Informes_Condonacion.docx)

# Process View

# Deployment View

En esta sección se describen las configuraciones en las que el software es desplegado y puesto en marcha. Estas configuraciones son visualizadas mediante el diagrama de despliegue. Estos diagramas indican nodos físicos con capacidad de procesamiento que ejecutan software y sus interconexiones a través de diferentes medios como LAN, p2p, bus, etc.

Debido a que el modulo debe estar soportado como aplicación web se usara el modelo de cliente/servidor como podemos observar en el siguiente diagrama:



*Diagrama de Componentes*

Los elementos que intervienen en este diagrama de despliegue son:

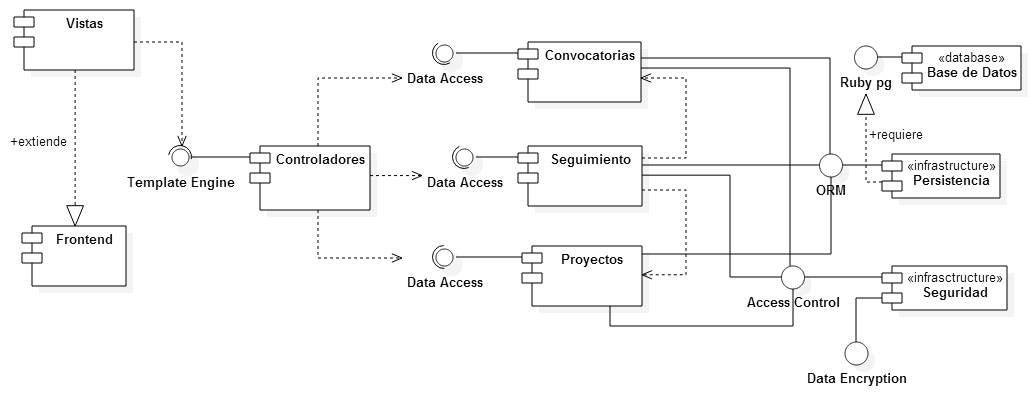
* **Estación de trabajo:** Este nodo es un cliente que accede al servidor por medio de un navegador web a través del protocolo HTTP.  
  Sus características son:
  + **Procesador:** Inter core i3 1.8 GHz, 2M cache
  + **Memoria RAM:** DDR3 2 GB
  + **Disco duro:** SATA 220 GB
* **Dispositivo móvil:** Este nodo representa un cliente que accede al servidor de producción por medio de un Smartphone o Tablet y se comunica con él a través del protocolo HTTP.  
  Sus características son:
  + **Sistema operativo:** Android 4.4 Kitkat
  + **Memoria RAM:** 1 GB
  + **Almacenamiento:** 8 GB
* **Servidor de producción:** Este nodo es donde se tendrá funcionando la aplicación para uso final. Las tecnologías usadas en este nodo son un servidor apache complementado con un servidor nginx en caso de que se requiera y un repositorio local en el que se almacena el código de producción. Se comunica con el servidor de la base de datos a través del ORM.  
  Sus características son:
  + **Servidor:** Dell PowerEdge R720xd
  + **Procesador:** Intel Xeon serie E5-2600 v3, 2.8 GHz, 15M cache, 10 cores
  + **Memoria RAM:** RDIMM DDR4 32 GB
  + **Disco duro:** SSD 960 GB
  + **Sistema operativo:** Ubuntu Server 14.04 LTS x64
* **Servidor de estáticos:** Este nodo se encarga únicamente de servir archivos estáticos al servidor principal debido a que entre los requisitos de la aplicación se tiene en cuenta un gran rendimiento y una disponibilidad alta. Esta técnica es usada con el servidor nginx que ya ha demostrado que es bastante eficiente en esta tarea. Se conecta al servidor de producción mediante HTTP.  
  Sus características son:
  + **Servidor:** Dell PowerEdge T110
  + **Procesador:** Intel Xeon E5-2400 2.2 GHz, 10M cache
  + **Memoria RAM:** DDR3 8 GB
  + **Disco duro:** SATA 1TB 7.2K RPM 3Gbps
  + **Sistema operativo:** Ubuntu Server 14.04 LTS x64
* **Servidor de base de datos:** En este nodo se almacena únicamente todo lo relacionado con la base de datos que en nuestro caso es Postgresql versión 9.3. Se comunica a través de LAN con otro servidor que le sirve de backup para cumplir el requerimiento exigido en el proyecto de tener copias de seguridad y un plan de recuperación de la información.  
  Sus características son:
  + **Servidor:** Dell Competent SC8000
  + **Procesador:** Dos procesadores Intel de 6 nucleos a 2.5 GHz
  + **Memoria RAM:** DDR3 12 GB
  + **Disco duro:** SCSI 150 TB
  + **Sistema operativo:** Ubuntu Server 14.04 LTS x64 con PostgreSQL 9.3
* **Backup de base de datos:** El propósito de este nodo es servir como estrategia de recuperación de la información de la base de datos principal y almacenar las copias de respaldo generadas periódicamente. Se conecta mediante LAN con la base de datos central.  
  Sus características son:
  + **Servidor:** IBM System x3690 X5
  + **Procesador:** Intel Xeon E7, 2.0 GHz, 4 cores
  + **Memoria RAM:** DDR3 12 GB
  + **Disco duro:** SCSI 150 TB
  + **Sistema operativo:** Ubuntu Server 14.04 LTS x64 con PostgreSQL 9.3

# Implementation View

La vista de implementación define el modelo de implementación, la descomposición del software en capas y subsistemas y los componentes de gran importancia arquitectónica.

## Overview

A continuación se definirán las capas y los componentes incluidos en ellas, además de mostrar las interacciones entre capas y componentes en un diagrama de componentes.



*Diagrama de componentes*

## Layers

* **Capa de la vista:** En esta capa se agrupan todas las plantillas HTML y elementos CSS propios del módulo. Además también contiene el componente Frontend quien es el que se comunica con estas vistas para extender sus funcionalidades a nivel de interfaz gráfica de usuario. Las vistas se comunican con los controladores a través de un motor de plantillas que facilita la renderizacion de las mismas.
* **Capa del modelo:** Esta capa contiene los componentes Convocatorias, Seguimiento y Proyectos que están relacionados con las clases y entidades mapeadas de la base de datos.
* **Capa del controlador:** En esta capa se almacenan componentes controladores, es decir se agrupan n distintos controladores responsables comunicarse con los modelos y realizar procesamiento con su información.
* **Capa de elementos comunes:** Esta capa contiene los componentes que son de uso común por otros componentes como lo son por ejemplo el sistema de seguridad y autenticación, asi como el de persistencias y manejo de la base de datos.

# Size and Performance

La aplicación tiene definidos unos lineamientos claros en cuanto a rendimiento y manejo de recursos en el documento “Requisitos no funcionales”, *Sistema de Información Integrado. Especificaciones funcionales y técnicas – Comunidad Colciencias*, pág. 796, que aplican para todos los módulos, y para el atributo de calidad referente al rendimiento se encuentran las condiciones en “Desempeño”, *Sistema de Información Integrado. Especificaciones funcionales y técnicas – Comunidad Colciencias*, pág. 797.

En este caso, el módulo de seguimiento debe hacer especial énfasis en las siguientes características:

* **Peso aproximado de archivos anexos:** Los documentos anexos a los formularios para los distintos procesos de seguimiento no deben superar las 500 Mb.
* **Máximo factor de almacenamiento en disco:** La cantidad de espacio utilizado en la base de datos nunca debe ser superior al 80%.
* **Desempeño:** Las transacciones del módulo no deben superar una duración de 5 segundos con 500 usuarios concurrentes.

Adicional a esto se tiene también se cuenta con un escenario para el atributo de calidad del desempeño o rendimiento indicado en la siguiente tabla:

|  |  |
| --- | --- |
| Parte del escenario | Posible Valor |
| Fuente del Estimulo | Los Usuarios |
| Estimulo | Inicio de 400 transacciones por minuto |
| Artefacto | El Sistema |
| Ambiente | Modo Normal |
| Respuesta | Se procesan las transacciones |
| Medida de Respuesta | Latencia de 4 segundos |

**Tácticas para la mejora del rendimiento:**

Para asegurar un mejor rendimiento para el sistema en este módulo, se tendrá como estrategia el uso de 3 tácticas que permiten disminuir el consumo de recursos y el tiempo de bloqueo:

* **Demanda de recursos:** El consumo de recursos puede ser disminuido cuando se define que un recurso puede ser tratado por otro. Teniendo esto en cuenta se delegara entonces la función de servir los archivos estáticos de la aplicación a otro servidor (nginx) para que así el servidor central se quite esa carga de procesamiento. Otra estrategia podría ser la priorización de las peticiones entrantes de manera que se evite la saturación del servidor.
* **Gestión de recursos:** Si las peticiones pueden ser manejadas de manera paralela entonces podría implementarse la técnica de concurrencia mediante el uso hilos con la ayuda de un balanceador de carga. Mantener copias de los datos en cache y sincronizarlos posteriormente con la base de datos es otra estrategia que podría funcionar bastante bien.
* **Arbitraje de recursos:** Establecer una estrategia de manejo de los procesos entrantes como por ejemplo la técnica de “Primero en entrar, primero en salir” que básicamente atiende los procesos como si fueran una cola. También se podría usar un método como el Round Robin que asigna prioridades dinámicamente a los procesos, o cualquiera otra política de planificación.

# Quality

Los atributos de calidad de software para esta arquitectura están definidos claramente en el documento “Requisitos no funcionales”, *Sistema de Información Integrado. Especificaciones funcionales y técnicas – Comunidad Colciencias*, pág. 796. Sin embargo para este módulo en específico consideramos que los requisitos de calidad que son más fuertemente necesarios en los casos de uso asignados son:

* Desempeño
* Usabilidad
* Seguridad
* Respaldo
* Disponibilidad