Sistema Integrado de Información

o Arquitectura Software

Versión 1.0

Andres Fernando López Avila

Juan Esteban Moreno Rodríguez

Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Dia** | **Versión** | **Descripción** | **Autor** |
| 06/09/2014 | 1.0 | Se inició la organización de la plantilla realizando su traducción al español y la redacción de cada uno de los ítems. | Andres Fernando López Avila. |
| 07/09/2014 | 1.1 | Se da continuidad con el documento y se empieza la realización de las vistas. | Andres Fernando López Avila.  Juan Esteban Moreno. |
|  |  |  |  |

Tabla de contenido

1. Introducción 3

1.1 Objetivo 3

1.2 Alcance 3

1.3 Definiciones, acrónimos y abreviaturas 3

1.4 Referencias 3

1.5 Información general 3

2. Representación arquitectónica 3

3. Objetivos arquitectónicos y limitaciones 3

4. Vista Caso-Uso 3

5. Vista lógica 3

5.1 Información general 3

5.2 Paquetes de Diseño Arquitectónico significativos 3

5.3 Realización Casos de uso 3

6. Vista de Despliegue 3

7. Vista Implementación 3

7.1 Información general 3

7.2 Capas 3

8. Tamaño y Rendimiento 3

9. Calidad 3

Documento Arquitectura Software

# Introducción

El proceso del diseño de la aplicación del sistema integrado de información para Colciencias debe ser un proceso con criterio no realizado de manera improvisada ni empírica ya que cada sistema posee características distintas, gracias a las especificaciones obtenidas en el documento proporcionado por Colciencias podemos reconocer cada uno de los procesos propios del sistema de información y poder realizar una solución arquitectónica del proyecto.

## Objetivo

El objetivo principal del documento es analizar el caso de estudio y los lineamientos con los cuales ha de plantearse la arquitectura del sistema encargado de la aplicación Sistema integrado de Información (SII) para Colciencias, la idea fundamental de este documento es presentar las decisiones arquitectónicas que se tome dentro del grupo de desarrollo, partiendo de patrones y estilos estándares y adaptándolos a las necesidades propias del proyecto.

## Alcance

El sistema Integrado de Información (SII) de Colciencias es una aplicación que se utilizara para la gestión de usuarios, proyectos de investigación, comunidades, grupos, convocatorias, seguimientos de proyectos y para disminuir la carga operativa de la entidad en la prestación de sus servicios.

El presente documento hace referencia al diseño realizado para el SII de Colciencias, el cual ha sido producto de un análisis elaborado por el grupo de trabajo sobre los requerimientos del sistema, como estos pueden ser satisfechos con las tecnologías y características definidas en las Especificaciones Funcionales y Técnicas.

El documento está estructurado alrededor de tres puntos principales:

* Las características generales del diseño.
* Realización de casos de uso.
* Modelos y vistas que lo detallan.

Los modelos son utilizados tanto para el análisis de requisitos, como para el diseño de la solución, así como para la especificación, construcción y despliegue del sistema en su ambiente.

Los modelos son presentados por vistas o diagramas, generalmente utilizando notaciones gráficas como UML.

## Definiciones, acrónimos y abreviaturas

**SII**: sistema integrado de información que hace uso intensivo y extensivo de las tecnologías de la información y las comunicaciones.

**UML:** Lenguaje Unificado de Modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

**SAD:** Arquitectura Software Documento

## Referencias

Las referencias utilizadas para la ayuda y realización de este documento están dadas a continuación:

* [Especificaciones Funcionales y Técnicas – Comunidad Colciencias.](../../Proyecto/DA_PROCESO_14-6-12678_000000000_10671393.pdf)
* [IEEE-1471 - Rich Hilliard](http://www.enterprise-architecture.info/Images/Documents/IEEE%201471-2000.pdf)
* [“4+1” View Model of Software Architecture - Philippe Kruchten](http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf)

## Información general

Este documento detalla claramente la arquitectura sencilla y confiable que se implementará al proyecto. Con esta se muestran los principales casos de uso que conlleva el software y la forma como el usuario interactúa con estos, el objetivo que pretende y la meta que se debe cumplir

* **Sección 1:** Representación arquitectónica esta es la que describe la arquitectura de software del SII su representación, su distribución e implementación de vistas así como también sus objetivos arquitectónicos y las limitaciones que este sistema presenta.
* **Sección 2:** Vista de caso de uso, a través de esta vista realizaremos una definición del alcance funcional que tendrá la aplicación y cada uno de los subsistemas que lo componen.
* **Sección 3:** En la sección de la vista lógica observaremos la información que corresponde a las clases con las cuales se realizara la implementación del sistema y cada uno de los casos de uso a realizar.
* **Sección 4:** Esta sección se compone de la vista de procesos organizar la sección por grupos de procesos que se comunican o interactúan. Describir los principales modos de comunicación entre procesos.
* **Sección 5:** Diagrama de Despliegue muestra de manera gráfica los componentes hardware que conforman el sistema, además los describe indicando la localización de las tareas de los componentes físicos.
* **Sección 6:** En esta sección se describe la estructura completa del Modelo de Implementación, la descomposición del software en capas y subsistemas en el Modelo de Implementación, y cualquier componente arquitectónicamente significativo.
* **Sección 7:** En la sección de Tamaño y Rendimiento es donde se describe cuáles serán las características mínimas de la tecnología en la que se desplegara la aplicación.
* **Sección 8:** LaCalidad será la sección final y en donde se describe cada uno de los requerimientos de calidad que implementaremos para tener un software óptimo.

# Representación arquitectónica

El proyecto se basara en un aspecto de arquitectura principalmente, el cual es Modelo- Vista – Controlador, con esto pretendemos construir una arquitectura de buena calidad y que nos permita realizar cambios en el modelo utilizado, se desarrollara una sola aplicación integrada en la que se permitirá el acceso a los usuarios que se encuentren registrados y a las áreas a las cuales él tiene acceso autorizado.

La arquitectura la mostraremos como una serie de vistas usando la aplicación visual-paradigm y UML, además de esto nos basaremos en el modelo 4+1, en el cual se contienen la vista lógica, de implementación, de procesos e implementación y la vista central de los casos de uso.

# Objetivos arquitectónicos y limitaciones

El objetivo principal de la arquitectura de la aplicación es mostrar e informar cómo está estructurado el proyecto, como actuara un cambio al momento que se requiera y nos demostrara la importancia de realizar un análisis y un diseño previo a la implementación.

Se tomaran en cuenta los siguientes objetivos y limitaciones para el diseño de la arquitectura del sistema:

**Objetivos:**

* La aplicación permitirá a los usuarios acceder desde cualquier terminal o computadora.
* Para poder acceder al sistema se necesita poseer un código y una contraseña de usuario valida y dependiendo del rol del usuario se le habilitaran las opciones correspondientes, que en nuestro caso será administrador.
* Permitirá al administrador configurar y editar noticias y publicaciones dentro del portal.
* Configurar plantillas de notificaciones y alertas.
* Registrar un investigador como evaluador.
* Administrar los parámetros, atributos y listas del sistema.

**Limitaciones:**

* Como requisito indispensable suministrado por el cliente se deberá hacer uso de uno o más frameworks para la realización e implementación del software.
* No se podrá hacer uso de del lenguaje de programación JAVA.

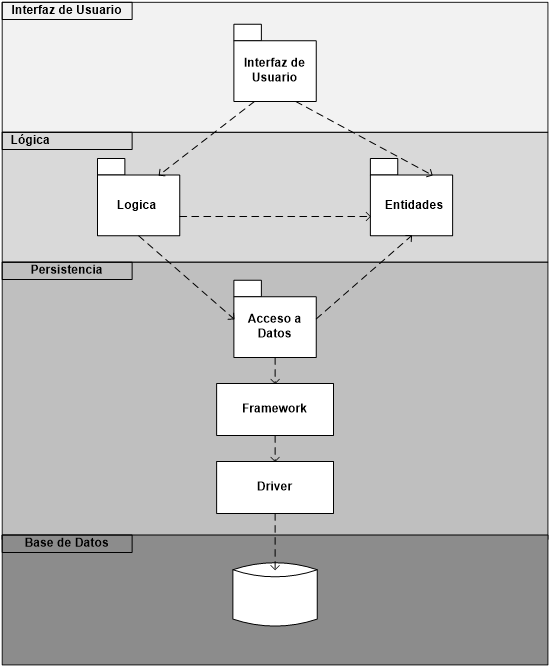
# Vista Caso-Uso

* Ver: [DA\_PROCESO\_14-6-12678\_000000000\_10671393 – Sección Modulo de Administración – desde la página 80 hasta la 96.](https://www.dropbox.com/s/hvc5nbpp793j7wh/DA_PROCESO_14-6-12678_000000000_10671393.pdf?dl=0)

# Vista lógica

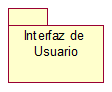
## Información general

El framework Yii implementa el paradigma modelo-vista controlador (model-view-controller MVC), el cual está adoptado ampliamente en la programación Web. MVC tiene como objetivo separar la lógica del negocio de las consideraciones de la interfaz de usuario a fin de que los desarrolladores puedan modificar cada parte más fácilmente sin afectar a la otra. En MVC el modelo representa la información (los datos) y las reglas del negocio; la vista contiene elementos de la interfaz de usuario como textos, formularios de entrada; y el controlador administra la comunicación entre la vista y el modelo.

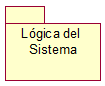


## Paquetes de Diseño Arquitectónico significativos

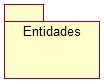
En el primer paquete del sistema se encuentra la interfaz de usuario o GUI el cual contiene las clases para la visualización de la aplicación y son las encargadas de crear una vista web que utilizara el usuario para comunicarse con el sistema.



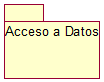
El segundo paquete será la lógica del sistema, en el que se encontraran las clases que permiten llevar a cabo los servicios del negocio, como son: Controlar el mantenimiento de los usuarios, administrar los parámetros, etc. Contiene la lógica para el manejo de las operaciones del sistema.



El paquete Entidades del Negocio contiene las clases que utilizará el sistema.



Por ultimo tendremos el paquete de acceso a datos en el cual se contienen las clases que el sistema utiliza para manejar la persistencia de los objetos con la Base de Datos.



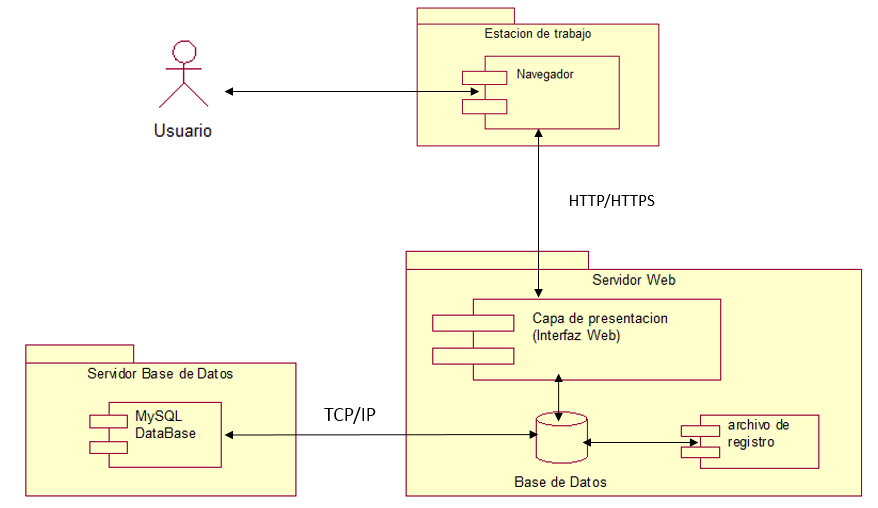
## Realización Casos de uso

[Esta sección ilustra cómo el software funciona realmente dando algunos casos de uso seleccionado (o escenario) realizaciones, y explica cómo los diferentes elementos del modelo de diseño contribuyen a su funcionalidad.]

Lo que está haciendo esteban

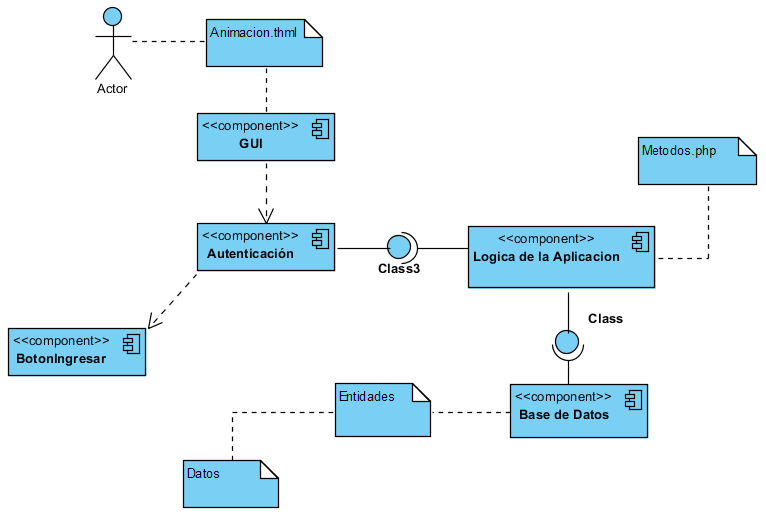
# Vista de Despliegue

El presente Diagrama de Despliegue muestra de manera gráfica los componentes hardware que conforman el sistema, además los describe indicando la localización de las tareas de los componentes físicos. Cada componente representa un recurso de ejecución como una computadora o servidor y se encuentra conectado con otros mediante un enlace de comunicación.



# Vista Implementación

## Información general



## Capas

* **Capa GUI:**

La interfaz gráfica que en nuestro caso es web, maneja las clases que permiten la interacción del usuario con el sistema, en pocas palabras será el medio por el cual el usuario utilizara las funciones de la aplicación.



* **Capa Lógica del Negocio:**

Este componente permite manejar la lógica del sistema de todas las áreas involucradas.



* **Capa Base de Datos:**

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información.



# Tamaño y Rendimiento

Al elegir esta arquitectura de software nos encontramos con ciertas características que deben ser tenidas en cuenta ya que afectan el rendimiento del software.

* **Peso aproximado archivos anexos convocatorias**:

El peso de los archivos que sean anexados por los usuarios de las convocatorias no puede ser mayor a 500 MB.

* **Máximo factor de almacenamiento de disco:**

Espacio utilizado vs espacio libre será de 80 %.

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

El espacio en disco necesario para la parte del Servidor de Base de datos deberá tener como mínimo 1 GB libres en disco para su correcto funcionamiento.

# Calidad

Para aprovechar de la mejor manera la arquitectura planteada debemos seguir los siguientes requerimientos de calidad:

* **Disponibilidad:**

La solución debe contar con una disponibilidad de 99.95%. Con un tiempo máximo de disponibilidad de (24 horas X 7 días) por semana.

* **Escalabilidad:**

La solución debe ser capaz de adaptarse y reaccionar sin perder la calidad o bien manejar el crecimiento continuo de trabajo de manera fluida.

* **Seguridad:**

Se debe garantizar la autenticación de usuarios e ingresaran a diferentes funcionalidades dependiendo del rol que desempeñen, permitirá la clasificación de información y el acceso a las funcionalidades de administración y configuración.

* **Respaldo:**

Se debe generar periódicamente copias de seguridad de la información.

* **Usabilidad:**

El sistema permitirá un manejo intuitivo por parte del usuario, tendremos usuarios de todas las edades y de diferentes ramas de la educación por lo que no puede ser accesible fácilmente solo para personas con conocimiento en tecnología si no para cualquier persona que tenga una conexión a internet.