Sistema Integrado de Información

o Arquitectura Software

Versión 1.0

Historial de revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| **Dia** | **Versión** | **Descripción** | **Autor** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
| 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.  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 Procesos 3

7. Vista de Despliegue 3

8. Vista Implementación 3

8.1 Información general 3

8.2 Capas 3

9. Tamaño y Rendimiento 3

10. 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

[Esta sección describe lo que la arquitectura de software es que el sistema actual, y cómo se representa. De los casos de uso, lógica, proceso, distribución e implementación Vistas, enumera los puntos de vista que son necesarios, y para cada vista, explica qué tipos de elementos de modelo que contiene.]

# Objetivos arquitectónicos y limitaciones

[Esta sección describe los requisitos de software y los objetivos que tienen algún impacto significativo en la arquitectura; por ejemplo, la seguridad, la seguridad, la privacidad, el uso de un producto off-the-shelf, la portabilidad, la distribución y la reutilización. También captura las limitaciones especiales que puedan aplicarse:. Diseño y Estrategia de aplicación, herramientas de desarrollo, estructura de equipo, programación, código de legado, y así sucesivamente]

# Vista Caso-Uso

[Esta sección lista los casos de uso o escenarios del modelo de casos de uso si representan algunas funciones importantes, el centro del sistema final, o si tienen una gran cobertura arquitectónica-que ejercer muchos elementos arquitectónicos o si el estrés o ilustran un procedimiento específico, delicado punto de la arquitectura.]

# Vista lógica

[En esta sección se describen las partes arquitectónicamente significativos del modelo de diseño, tales como su descomposición en subsistemas y paquetes. Y para cada paquete significativo, su descomposición en clases y utilidades de clase. Debe introducir clases de gran importancia arquitectónica y describir sus responsabilidades, así como unos muy importantes relaciones, operaciones y atributos.]

## Información general

[Esta sección describe la descomposición general del modelo de diseño en términos de su jerarquía y capas paquete.]

## Paquetes de Diseño Arquitectónico significativos

[Para cada paquete significativo, incluir un apartado con su nombre, su breve descripción y un diagrama con todas las clases y paquetes significativos contenidos en el paquete.

Para cada clase significativa en el paquete, incluya su nombre, breve descripción, y, opcionalmente, una descripción de algunos de sus principales responsabilidades, operaciones y atributos.]

## 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.]

# Vista de Procesos

[Esta sección describe la descomposición del sistema en procesos ligeros (hilos individuales de control) y los procesos de peso pesado (agrupaciones de procesos ligeros). Organizar la sección por grupos de procesos que se comunican o interactúan. Describir los principales modos de comunicación entre procesos, como el paso de mensajes, alarmas, y de encuentro.]

# Vista de Despliegue

[Esta sección describe una o más redes físicas (hardware) configuraciones en que el software se implementa y ejecuta. Es una visión del modelo de liberación. Como mínimo para cada configuración debe indicar los nodos físicos (computadoras, CPUs) que ejecutan el software y sus interconexiones (bus, LAN, punto a punto, y así sucesivamente.) Incluye también un mapeo de los procesos del Proceso Ver en los nodos físicos.]

# Vista Implementación

[Esta sección describe la estructura general del modelo de ejecución, la descomposición del software en capas y subsistemas en el modelo de implementación, y cualquier componente de gran importancia arquitectónica.]

## Información general

[Esta subsección nombres y define las diferentes capas y su contenido, las normas que rigen la inclusión de una capa determinada, y los límites entre capas. Incluya un diagrama de componentes que muestra las relaciones entre capas. ]

## Capas

[Para cada capa, incluir una subsección con su nombre, una enumeración de los subsistemas situados en la capa, y un diagrama de componentes.]

# Tamaño y Rendimiento

[Una descripción de las principales características de dimensionamiento del software que afectan a la arquitectura, así como las limitaciones de rendimiento objetivo.]

# Calidad

[Una descripción de cómo la arquitectura de software contribuye a todas las capacidades (excepto la funcionalidad) del sistema: extensibilidad, fiabilidad, portabilidad, y así sucesivamente. Si estas características tienen un significado especial, como las implicaciones de seguridad, de seguridad o privacidad, deben estar claramente delineados.]