

## Historia del documento

Date	Versión	Descripción	Autor
27/09/2018	1	Inicio Documento	

## Aprobaciones

<div>.....</div>
------------------

## Contenido

1.	Introducción	4
1.1	Propósito de este documento	4
1.2	Documentos de referencia	5
1.3	Definición acrónimos y abreviaturas	5
1.4	Referencias	5
2.	Modelo de diseño	6
2.1	Componentes de software	7
3.	Modelo de arquitectura	8
3.1	Realización de las capas	<b>Error! Bookmark not defined.</b>
3.2	Realización de las capas	<b>Error! Bookmark not defined.</b>
3.3	Realización de las capas	10
3.3.1	Capa Cliente	10
3.3.2	Capa Presentación	10
3.3.3	Capa Lógica de Negocio	10
3.3.4	Acceso a datos	10
3.3.5	Integración	10
3.3.6	Recursos	11
3.4	Componentes y frameworks	12
3.5	Modelo entidad relación	13

## TABLAS

Tabla 1 Documentos de referencia .....	5
Tabla 3 Descripción servidores .....	<b>Error! Bookmark not defined.</b>
Tabla 4 Frameworks , librerías y componentes .....	12

## FIGURAS

Figura 2 Diagrama de secuencia .....	6
Figura 3 Diagrama de despliegue detallado .....	9

# 1. Introducción

## 1.1 Propósito de este documento

El documento de diseño detallado de Software contiene los modelos necesarios y suficientes para la construcción del sistema de información Sapiens. Aquí se encuentra detallado, todos los componentes, servidores y demás artefactos requeridos tanto para la construcción del sistema de información, como para su instalación y mantenimiento.

Este documento es construido teniendo en cuenta los diferentes requisitos funcionales a partir de los cuales el software será desarrollado, velando que se cumplan a cabalidad todas las historias de usuario que se han identificado en las iteraciones de análisis.

## 1.2 Documentos de referencia

Titulo	Versión	Autor

Tabla 1 Documentos de referencia

## 1.3 Definición acrónimos y abreviaturas

- http: HypertextTransfer Protocol
- tcp/ip: Protocolo de transferência por IP

## 1.4 Referencias

[KRU41]: The "4+1" view model of software architecture, Philippe Kruchten, November 1995,  
<http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>

[RSA]: IBM Rational Software Architect  
<http://www-306.ibm.com/software/awdtools/architect/swarchitect/index.html>

[RUP]: The IBM Rational Unified Process  
<http://www-306.ibm.com/software/awdtools/rup/index.html>

[RUPRSA]: Developing a J2EE Architecture with Rational Software Architect using the Rational Unified Process®, IBM DeveloperWorks, Jean-Louis Maréchaux, Mars 2005,  
[http://www-128.ibm.com/developerworks/rational/library/05/0816\\_Louis/](http://www-128.ibm.com/developerworks/rational/library/05/0816_Louis/)

## 2. Modelo de diseño

En el siguiente diagrama de secuencia, se detallan los componentes y la secuencia de llamados que se deben ejecutar, con el fin de cumplir todas y cada una de las acciones que un actor usa en cualquiera de las funcionalidades dentro del sistema de información.

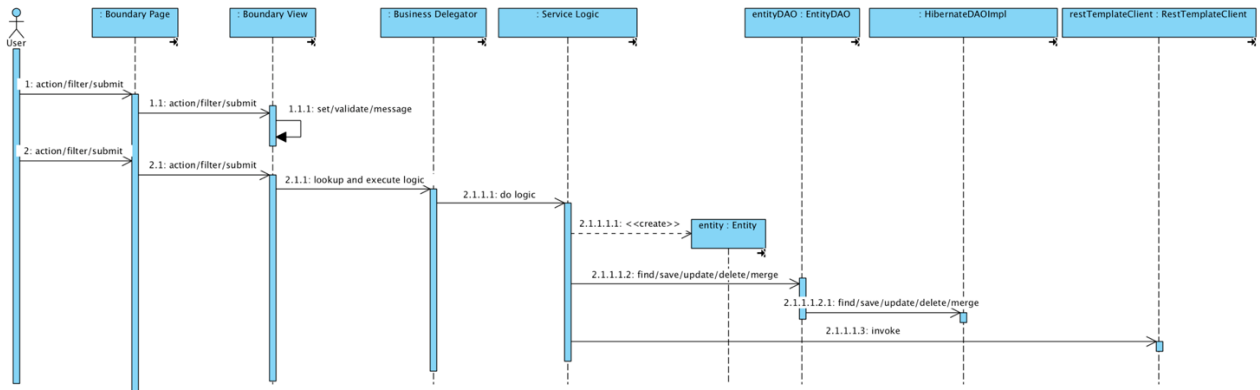


Figura 1 Diagrama de secuencia

## 2.1 Componentes de software

**Boundary Page:** Página Web, construida en lenguaje XHTML.

**Boundary View:** Lógica de presentación, construida en un componente Controlador de pantalla.

**Rest Controller:** Componente que encapsula y desacopla las invocaciones entre la capa de presentación y la capa de lógica de negocio.

**Service Logic:** Componente que implementa la lógica de negocio del sistema de información. Aquí se encuentran los algoritmos que llevan a cabo las funcionalidades identificadas en las iteraciones de análisis.

**Entity Dao:** Componente encargado de la lógica de persistencia objeto/relacional.

**Hibernate DAO Impl:** Componente con la lógica de integración a la base de datos relacional

**Rest Template:** En caso de existir integraciones con sistemas externos, las invocaciones se realizarían por medio de Servicios Rest. Este componente se encargaría de desacoplar y traducir las invocaciones http a los componentes remotos.



### 3. Modelo de arquitectura

El Modelo Físico/de Despliegue provee un modelo detallado de la forma en la que los componentes se desplegarán a lo largo de la infraestructura del sistema. Detalla las capacidades de red, las especificaciones del servidor, los requisitos de hardware y otra información relacionada al despliegue del sistema propuesto.

En el siguiente diagrama, se detallan los nodos de hardware, los componentes y la relación entre ellos, en una distribución física:



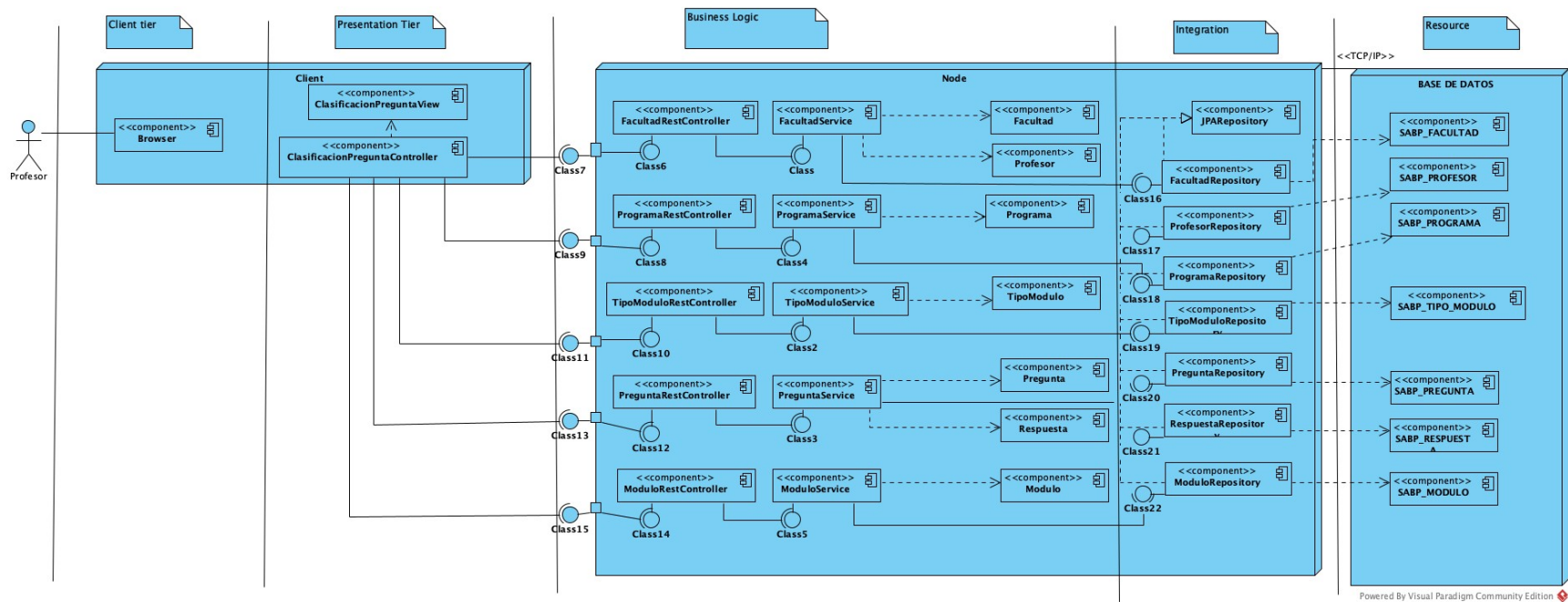


Figura 2 Diagrama de despliegue detallado

## 3.1 Realización de las capas

### 3.1.1 *Capa Cliente*

Para la capa cliente no es necesario hacer desarrollo de ningún tipo debido a que la aplicación será web sin usar ningún tipo de plugin del lado del cliente. El usuario debe contar con un navegador Web de alguna de las siguientes versiones Firefox 21 o superior, Google Chrome 26 o superior, Safari 6 o superior.

### 3.1.2 *Capa Presentación*

Encargada de la visualización de la aplicación. Esta capa esta compuesta por los componentes de presentación de las aplicaciones los cuales serán desarrollados usando una implementación de Angular 9.

### 3.1.3 *Capa Lógica de Negocio*

Encargada de establecer las restricciones propias de la lógica de negocios o los aspectos particulares de la aplicación. Contiene la lógica de negocio.

La lógica de negocio será implementada en clases planas (pojo) usando el framework de Spring con la implementación de anotaciones.

### 3.1.4 *Acceso a datos*

Encargada del almacenamiento de los datos en el sistema de base de datos los cuales serán implementados usando HibernateCore.

### 3.1.5 *Integración*

Encargada de la comunicación con los sistemas externos como Servicios web, encoladores de transacciones, Mensajería, etc.

Para la integración con sistemas externos tanto exposición de servicios web como consumo se usará JAX-WS para servicios SOAP y JAX-RS para servicios REST.

En los casos de integración con los sistemas de mensajería se debe usar JMS.

### **3.1.6 Recursos**

Encargada del gestor de base de datos, FileSystem. En esta capa no se hará ningún desarrollo simplemente se debe usar una base de datos relacional y sistema de almacenamiento según sea el caso.

## 3.2 Componentes y frameworks

A continuación, se hace una descripción de los diferentes frameworks, componentes y librerías a usar en el proyecto por cada una de las capas.

Capa	Framework o Librería	Funcionalidad	Versión
Presentación	Angular	Componentes de presentación UI	9
Presentación	Apache POI	Generación de Archivos Excel, Power Point, Word	3.9
Presentación	iText	Generación de PDF	5.5.12
Lógica de Negocio	Spring	Contenedor de inyección de dependencias e inversión de control	5.0.7
Persistencia	Hibernate	Persistencia de datos Mapeador Objeto relacional	5.3.1
Integración	JAX-WS	Servicios Web SOAP	2.2
Integración	JAX-RS	Servicios Web Rest full	1.1
Reportes	JasperReport	Generación de reportes en diferentes formatos. Se aconseja usar iReport para diseñar los reportes.	6.3.1
Logs	Log4j	Manejo de logs de la aplicación	2.11
Auditoria	Spring AOP	Manejo de logs de auditoria en las diferentes capas de la aplicación	3.2
Seguridad	Spring Security	Spring Security es un marco poderoso y altamente personalizable de autenticación y control de acceso. Es el estándar para asegurar aplicaciones basadas en Spring.	5.0.6

Tabla 2 Frameworks , librerías y componentes

### 3.3 Modelo entidad relación

Ver archivos:

- sapiens - MER.jpg

