REPÚBLICA DEL ECUADOR
Escuela Politécnica Nacional
"E SCIENTIA HOMINIS SALUS"

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO E IMPLEMENTACION DE UN PROTOTIPO DE SEGURIDAD WEB PARA LA GESTION Y ADMINISTRACION DE DOCUMENTOS PARA EL SISTEMA ALFRESCO

PROYECTO DE TITULACION PREVIO A LA OBTENCION DEL TITULO DE INGENIERO EN ELECTRONICA Y REDES DE LA INFORMACION

FERNANDO VINICIO OCHOA CARRIÓN

heavyvinicio@hotmail.com

DIRECTOR: ING. FLORES CIFUENTES WILLIAMS FERNANDO

fflores@epn.edu.ec

Quito, Marzo 2015

DECLARACIÓN

Yo, Fernando Vinicio Ochoa Carrión, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Fernando Vinicio Ochoa Carrión

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Fernando Vinicio Ochoa Carrión, bajo mi supervisión.

Ing. Fernando Flores

DIRECTOR

AGRADECIMIENTOS

A Dios, por darme la oportunidad de poseer tan valioso regalo como es la educación, la sabiduría y los dones para aplicar todo este conocimiento generado por mis maestros, amigos y familiares a lo largo de mi carrera estudiantil en beneficio de la humanidad.

Mis más sinceros agradecimiento a todas aquellas personas que hicieron de este proyecto una realidad, en especial énfasis a la Srta. Lourdes Custodio, de cuya gestión se logró obtener el trabajo de la mejor manera.

A mis padres, con especial énfasis a mi madre por haberme dado todo su apoyo incondicional en todos esos momentos difíciles que se han dado a lo largo de la carrera.

A mi director, por guiarme en todos los aspectos esenciales para que este trabajo se consolide como un proyecto de calidad útil para la universidad y la sociedad.

Vinicio Ochoa

DEDICATORIA

Dedico este proyecto de titulación y toda mi carrera estudiantil, a Dios y a mi familia ya que gracias a ellos hoy por hoy mantengo por sobre todo los valores y la constancia que me inculcaron en su momento.

A mis amigos, que jamás dejaron de creer en este proyecto y del cual recibí muchos consejos que ayudaron a complementar este trabajo.

A Jessica, que jamás dejo de creer en mí y que siempre estuvo a mi lado apoyándome y motivándome hasta alcanzar la culminación del objetivo final.

A mis maestros, pues ellos me inculcaron todos los conocimientos necesarios para que se pueda estructurar y desarrollar este proyecto de la mejor manera posible.

Vinicio Ochoa

CONTENIDO

ÍNDICE DE FIGURAS. XIII

ÍNDICE DE TABLAS..... XVII

RESUMEN..... XIX

ABSTRACT..... XX

PRESENTACIÓN XXI

CAPÍTULO I 1

MARCO TEÓRICO 1

1.1 INTRANET Y EXTRANET 1

1.1.1 RED 1

1.1.2 INTRANET 1

1.1.2.1 Beneficios 2

1.1.2.2 Características 2

1.1.3 EXTRANET 2

1.1.3.1 Beneficios 2

1.1.3.2 Características 3

1.1.4 INTERNET 3

1.2 SEGURIDAD DE LA RED 4

1.2.1 MECANISMOS DE SEGURIDAD 4

1.2.1.1 Seguridad Física 4

1.2.1.2 Seguridad Lógica 4

1.2.1.3 Seguridad Activa 5

1.2.1.4 Seguridad Pasiva 5

1.2.2 SERVICIOS DE SEGURIDAD 5

1.2.2.1 Confidencialidad 5

1.2.2.2 Autenticación 5

1.2.2.3 Integridad.....5

1.2.2.4 No Repudio.....5

1.2.2.5 Control de Acceso 6

1.2.2.6 Disponibilidad 6

1.2.3 MÉTODOS DE SEGURIDAD MÁS USADOS ACTUALMENTE 6

1.2.3.1 Firma Digital 6

1.2.3.2 Firma electrónica 6

1.2.3.2.1 Diferencias entre la firma digital vs firma electrónica 7

1.2.3.3 Certificados Digitales.....7

1.2.3.3.1 Formato 7

1.2.3.3.2 Tokens de Seguridad 8

1.2.3.3.3 Características 9

1.2.3.4 CryptoApplet.....9

1.2.3.4.1 Formatos de firma que permite CryptoApplet 9

1.3 ENTERPRISE CONTENT MANAGEMENT – ECM 11

1.3.1 DEFINICIÓN 11

1.3.2 TIPOS DE ECM 11

1.3.2.1 Plataforma Windows.....12

1.3.2.1.1 SharePoint 12

1.3.2.2 Plataforma Linux.....13

1.3.2.2.1 Alfresco 13

1.3.2.2.2 Nuxeo 15

1.3.2.2.3 Comparativa de Alfresco vs Nuxeo 16

1.3.3 ALFRESCO 16

1.3.3.1 Arquitectura de Alfresco 16

1.3.3.1.1 Capa General 17

1.3.3.1.2 Alfresco Client 18

1.3.3.1.3 Alfresco Content Application Service 19

1.3.3.1.4 Alfresco Content Repository 20

1.3.3.2 Extendiendo la funcionalidad de Alfresco 23

1.3.3.2.1 Web Services 23

1.3.3.2.2 SOAP – Simple Object Access Protocol 24

1.3.3.2.3 RESTFULL Web Services 27

1.3.4 NORMATIVA DE DOCUMENTACIÓN VIGENTE 28

1.3.4.1 Nacional 28

1.3.4.1.1 Norma NTE INEN 2410 29

1.3.4.1.2 Ley de Comercio Electrónico, firmas electrónicas y mensajes de datos 30

1.3.4.2 Internacional 31

1.3.4.2.1 ISO/IEC 15489 Un marco sistemático para las buenas prácticas de gestión documental en organizaciones 31

1.3.4.2.2 ISO 9075 Tecnologías de la Información – Lenguajes de Bases de Datos SQL. 31

CAPÍTULO II 34

DISEÑO DEL SISTEMA WEB 34

2.1 QUÉ ES LA COMIEX? 34

2.1 ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA COMIEX..... 35

2.1.1 ANÁLISIS DE RECURSOS UTILIZADOS..... 35

2.1.1.1 Documentos Procesados 35

2.1.1.2 Materiales de oficina.....38

2.1.1.3 Relación Tiempo – Dinero 38

2.2 FUNCIONAMIENTO DE LA COMIEX..... 39

2.2.1 DEPARTAMENTOS 39

2.2.2 PROCESOS ADMINISTRATIVOS..... 40

2.3 PROCESOS QUE SE GESTIONAN EN LA COMIEX..... 42

2.4 ANÁLISIS DE REQUERIMIENTOS PARA EL SERVIDOR..... 43

2.4.1 CAPACIDAD DE ALMACENAMIENTO 43

2.4.2 DISPOSITIVOS DE RED 44

2.4.3 CAPACIDAD DE PROCESAMIENTO..... 45

2.4.4 CAPACIDAD DE MEMORIA 45

2.4.5 SERVIDOR ACTUAL DE LA COMIEX..... 46

2.5 ANÁLISIS DE LOS MODELOS DE GESTIÓN PARA ALFRESCO..... 47

2.5.1 ADQUISICIÓN DE LOS DATOS 47

2.5.2 ADMINISTRACIÓN DE LA INFORMACIÓN 48

2.5.3 CATEGORIZACIÓN DE LA INFORMACIÓN 48

2.5.4 ANÁLISIS Y DISEÑO DEL MODELO DE AUTENTICACIÓN Y AUTORIZACIÓN..... 49

2.5.4.1 Autenticación.....49

2.5.4.2 Autorización.....51

2.5.4.3 Identificación de los usuarios.....51

2.5.5 ANÁLISIS Y DISEÑO DE MODELOS DE CONTENIDO..... 52

2.5.5.1 Modelo de contenido (Alfresco, 2014).....52

2.5.5.1.1 Tipos de datos para representaciones de contenido..... 53

2.5.5.1.2 Asociaciones..... 54

2.5.5.2 Modelo de contenido para un memorando 54

2.5.5.3 Presentación de proyectos 56

2.5.5.4 Informes.....60

2.6 DISEÑO DEL SISTEMA WEB 62

2.6.1 QUE ES UNA ARQUITECTURA 62

2.6.2 QUE ES UN MODELO..... 63

2.6.3 MODELO CLIENTE – SERVIDOR 63

2.6.3.1 Ventajas.....63

2.6.3.2 Desventajas.....64

2.6.4 ARQUITECTURA DE N CAPAS 64

2.6.4.1 Capa de datos 66

2.6.4.2 Capa de negocios.....66

2.6.4.3 Capa de presentación 66

2.6.5 METODOLOGÍA UWE..... 66

2.6.5.1 ¿Qué es la metodología UWE y que permite? 66

2.6.5.2 Fases de metodología UWE.....66

2.6.5.3 Planificación: Análisis de requerimientos para la COMIEX 68

2.6.5.4 Diseño 71

2.6.5.4.1 Actores 71

2.6.5.4.2 Relaciones entre actores 72
2.6.5.4.3 Modelos de Casos de Uso y de paquetes 73
2.6.5.4.4 Modelos de Contenido 75
2.6.5.4.5 Modelo de navegación 77
2.6.5.4.6 Modelo de presentación 77
2.7 DISEÑO DE REPORTES 79
2.7.1 JASPERREPORT 79
2.7.1.1 Características técnicas 79
2.7.1.2 Beneficios de usar JasperReport 80
2.7.1.3 Diseño de reportes usando JasperReport 81
2.7.1.3.1 Estado de los trámites 81
2.7.1.3.2 Resumen financiero por proyecto 82
2.7.1.3.3 Informe final del proyecto 82
CAPÍTULO III 85
3 IMPLEMENTACIÓN DEL SISTEMA 85
3.1 INTRODUCCIÓN 85
3.2 ANÁLISIS DEL SISTEMA DESARROLLADO 85
3.2.1 ARQUITECTURA 85
3.2.1.1 Capa de datos y de negocios 86
3.2.1.1.1 Java Persistence API - JPA 87
3.2.1.2 Capa de servicios 91
3.2.1.2.1 Delegación de Servicios 92
3.2.1.3 Capa de presentación 93
3.2.2 JSF2 94

3.2.2.1 Funcionamiento 94
3.2.2.1.1 Modelo visor controlador 94
3.2.3 PRIMEFACES (Primefaces, 2014) 95
3.2.3.1 Características 96
3.2.4 SISTEMA GESTOR DE PROYECTOS DE INVESTIGACIÓN – SIGPI . 96
3.2.4.1 Control de Acceso 97
3.2.4.1.1 Proceso de autenticación 101
3.2.4.1.2 Gestión de usuarios 105
3.2.4.1.3 Integración con Alfresco 105
3.2.5 ANÁLISIS DE LA FIRMA DIGITAL CON CRYPTOAPPLET 114
3.2.5.1 Generación de certificados digitales 114
3.2.5.2 Compilación de CryptoApplet 115
3.2.5.3 Proceso para firmar 116
3.2.6 ANÁLISIS DE REPORTES 119
3.2.6.1 JXLS 119
3.2.6.1.1 Descripción 119
3.2.6.1.2 Implementación 120
3.2.6.2 JasperReport – IReport Designer 121
3.2.6.2.1 IReport Designer 121
3.3 INSTALACIÓN 125
3.3.1 CARACTERÍSTICAS DEL SERVIDOR WEB 125
3.3.1.1 IPTABLES 125
3.4 PRUEBAS DE FUNCIONAMIENTO 126
3.4.1 PRUEBAS FUNCIONALES DE LOS MÓDULOS 126

3.4.2 PRUEBAS DE NAVEGABILIDAD 126

3.4.3 PRUEBAS DE FIRMA DIGITAL 127

3.4.3.1 Firefox 127

3.4.3.2 Chrome 127

3.4.3.3 Internet Explorer 128

3.5 ANÁLISIS DE COSTOS 129

CAPÍTULO IV 131

CONCLUSIONES Y RECOMENDACIONES 131

4.1 CONCLUSIONES 131

4.1.1 Limitaciones 132

4.1.2 Ampliaciones del sistema 132

4.2 RECOMENDACIONES 133

REFERENCIAS BIBLIOGRÁFICAS 134

ANEXOS 147

ÍNDICE DE FIGURAS

Figura 1.1. Red Computadoras 1

Figura 1.2. Formato del Certificado Digital 8

Figura 1.3. Token de seguridad..... 9

Figura 1.4. Cuadro comparativo de ECM's..... 16

Figura 1.5. Arquitectura Alfresco. 17

Figura 1.6. Alfresco Content Service. 19

Figura 1.7. Alfresco Content Repository. 20

Figura 1.8. Funcionamiento del estándar CMIS 21

Figura 1.9. Protocolo SOAP 25

Figura 1.10. Funcionamiento SOAP..... 25

Figura 1.11. Formato Mensaje SOAP 26

Figura 2.1. Número de trámites promedio..... 37

Figura 2.2. Número de hojas promedio 37

Figura 2.3. Flujo de trabajo genérico para la COMIEX..... 43

Figura 2.4. Relación en MB de los documento escaneados 44

Figura 2.5. Velocidad de transmisión 45

Figura 2.6. Categorización de la documentación 48

Figura 2.7. Arquitectura de autenticación de Alfresco. 50

Figura 2.8. Modelo cliente - servidor 63

Figura 2.9. Arquitectura de N capas..... 65

Figura 2.10. Arquitectura de N capas..... 65

Figura 2.11. Fases del Desarrollo UWE 67

Figura 2.12. Estructura básica del funcionamiento de la aplicación 68

Figura 2.13. Caso de uso: Actores 72
Figura 2.14. Relación entre actores 73
Figura 2.15. Modelo de paquetes 74
Figura 2.16. Modelos de casos de uso para el administrador 74
Figura 2.17. Modelo de caso de uso para el coordinador 75
Figura 2.18. Modelo de caso de uso para el usuario general 75
Figura 2.19. Modelo de Contenido para la verificación de la firma digital 76
Figura 2.20. Modelo de Contenido para la firma digital y los reportes 76
Figura 2.21. Modelo de Navegación del menú genérico 77
Figura 2.22. Vista de login para todos los usuarios 78
Figura 2.23. Vista de inicio para el usuario general / investigador 78
Figura 2.24. Plantilla de reporte de tareas 81
Figura 2.25. Reporte de estado financiero del proyecto 82
Figura 2.26. Reporte final del proyecto 83
Figura 3.3. Arquitectura JEE – JPA. 87
Figura 3.4. Arquitectura de JPA. 88
Figura 3.5. Conversión modelo / objeto relacional 89
Figura 3.6. Relación Proyecto – Persona 89
Figura 3.7. Implementación de los servicios 92
Figura 3.8. Publicación de servicios 93
Figura 3.9. Modelo Visor – Controlador 95
Figura 3.10. Pantalla de inicio de SIGPI 96
Figura 3.11. Proceso de autenticación para un usuario 101
Figura 3.12. Configuración del archivo shiro.xml (1) 102

Figura 3.13. Configuración del archivo shiro.xml (2) 103

Figura 3.14. Jerarquía de directorios según los roles y urls asignadas 104

Figura 3.15. Interfaz de administración para un Investigador 105

Figura 3.16. Esquema general para autenticación del Sistema Alfresco 106

Figura 3.17. Método de autenticación utilizando los servicios web de Alfresco . 107

Figura 3.18. Categorías 107

Figura 3.19. Etiquetas 107

Figura 3.20. Servicios web ligeros publicados por Alfresco 108

Figura 3.21. Método para consultar mediante RestFull las tareas ejecutadas en Alfresco 109

Figura 3.22. Arquitectura de OpenCMIS 110

Figura 3.23. Método para obtener la estructura de directorios de Alfresco mediante CMIS 112

Figura 3.24. Obtención del repositorio del Alfresco 112

Figura 3.25. Interfaz de mantenimiento para los documentos extraídos desde Alfresco. Proyecto Semilla PIS-14-01 113

Figura 3.26 Generación de la CA 114

Figura 3.27. Creación del certificado 114

Figura 3.28. Firma del certificado generado con la CA 115

Figura 3.29. Exportación del certificado autofirmado 115

Figura 3.30. Comando para exportar un certificado con extensión PKCS12 115

Figura 3.31. Compilación de CryptoApplet 116

Figura 3.32. Interfaz para firmar y verificar la firma digital 116

Figura 3.33. Selección del certificado para firmar el documento 117

Figura 3.34. Firma Digital exitosa con CryptoApplet 117

Figura 3.35. Documento firmado en Alfresco 118
Figura 3.36. Verificación de la firma digital..... 118
Figura 3.37. Verificación de la firma digital..... 119
Figura 3.38. Segmento de código para generar un reporte en base a plantillas 120
Figura 3.39. Acceso a atributos de los objetos en el template 120
Figura 3.40. Definición de operaciones en el template..... 120
Figura 3.41. Conexión con la base de datos desde iReport 122
Figura 3.42. Obtención de tablas desde el diseñador de consultas del iReport. 122
Figura 3.43. Obtención de columnas..... 123
Figura 3.44. Diseño del reporte. 123
Figura 3.45. Vista previa del reporte. 124
Figura 3.46. Exportación del reporte a PDF. 124
Figura 3.47. Comparación de Chrome e Internet Explorer 126
Figura 3.48. Firma Digital exitosa con CryptoApplet 127
Figura 3.49. Verificación de la firma digital..... 127
Figura 3.50. Calculo de horas estimadas para el desarrollo del proyecto 130
Figura 3.51. Costo estimado del sistema 130
Figura D-6. Vista de Home 152
Figura D-7. Vista de reportes 152
Figura D-8. Vista de firma digital 153
Figura D-9. Vista de Home 153
Figura D-10. Vista de proyectos 154
Figura D-11. Vista de reportes 154
Figura D-12. Vista de firma digital 155

ÍNDICE DE TABLAS

Tabla 1.1. Comparación entre Internet, Intranet y Extranet..... 3

Tabla 2.1. Número de trámites promedio 36

Tabla 2.2. Gastos en insumos de oficina / mes..... 38

Tabla 2.3. Relación de tiempo / minutos 38

Tabla 2.4. Departamentos relacionados con la COMIEX 39

Tabla 2.5. Procesos desarrollados por departamento..... 41

Tabla 2.6. Análisis de la capacidad de almacenamiento del servidor 44

Tabla 2.7. Análisis de requerimientos de memoria para el servidor 46

Tabla 2.8. Características del servidor adquirido por la COMIEX 46

Tabla 2.9. Roles predefinidos en Alfresco..... 51

Tabla 2.10. Perfiles de usuarios para Alfresco. 52

Tabla 2.11. Tipos de datos..... 53

Tabla 2.12. Modelo de contenido para un memorando. 55

Tabla 2.13. Modelo de contenido para la presentación de proyectos de investigación..... 56

Tabla 2.14. Modelo de contenido para la presentación de informes de avance y final de la investigación 60

Tabla 2.15. Roles de los usuarios para la aplicación web..... 69

Tabla 2.16. Requerimientos generales para los usuarios 69

Tabla 2.17. Especificación Técnica de JasperReport..... 80

Tabla 3.1 Representación de columnas con JPA..... 90

Tabla 3.2 Relación Url - Rol 103

Tabla 3.3. Resumen de puertos y sentencias a ejecutar..... 125

Tabla 3.4. Tabla comparativa de compatibilidad entre navegadores web 128

Tabla 3.5. Tabla referencial de meses ocupados para la creación del SIGPI 129

RESUMEN

Este proyecto de titulación tiene la finalidad de aplicar un prototipo de seguridad web que permita gestionar la documentación con el ECM Alfresco y la firma digital con la ayuda de CryptoApplet. Este trabajo fue aplicado con éxito sobre la ex Comisión de Investigación y Extensión – COMIEX y ahora nuevo Vicerrectorado de Investigación y Proyección Social – VIPS, sobre el cual se implanto los modelos de datos necesarios para la gestión de: i) memorandos, ii) proyectos de investigación, iii) informes parciales y finales, iv) prometeos, entre otros generados por el VIPS. Adicionalmente, se hace un análisis de las tecnologías aplicadas para conectividad remota como son los Web Services, Rest Full y CMIS sin dejar de lado la metodología web UWE, que se utilizó para el modelamiento y diseño estructural y funcional del aplicativo.

El prototipo web fue desarrollado en base a la arquitectura JEE con JSF como la tecnología MVC propia del estándar de java para proyectos empresariales y con JPA para gestionar la persistencia con la base de datos de los registros generados desde el aplicativo. Adicionalmente, se utilizó para la capa de presentación web el framework PrimeFaces como el ente principal que permitirá gestionar peticiones ajax, controles y componentes dinámicos para mejorar la interacción del usuario final con el sistema. Por otro lado, se analiza el funcionamiento de la firma digital, generación de certificados digitales con capacidad de firma sobre documentos en formato PDF con la ayuda de CryptoApplet.

Para finalizar, se concluyen con un análisis de resultados obtenidos de las pruebas realizadas por los usuarios finales y por el desarrollador en cuanto a funcionalidad y navegabilidad.

Palabras claves: CMIS, ECM, firma digital, java

ABSTRACT

This titling project aims to implement a prototype of web security that allows managing documents with Alfresco ECM and digital signature with the help of CryptoApplet. This work was successfully applied on the former Commission Research and Extension - COMIEX and now new Vice-Principalship for Research and outreach - VIPS, on which data models needed for management was implanted: i) memorandums, ii) projects research, iii) partial and final reports iv) promethean, among others generated by the VIPS. Additionally, an analysis of the technologies applied for remote connectivity becomes such as Web Services, Rest Full and CMIS without leaving the web UWE methodology, which was used for modeling and structural and functional design of the application.

The web prototype was developed based on the JEE architecture with JSF as a java technology standard for business projects and JPA to manage the persistence database of records generated from the application. Additionally, for the presentation layer was used PrimeFaces framework as the main technology that allow you to manage ajax requests, controls and dynamic components to improve end-user interaction with the system. Moreover, in this document is analized the operation of the digital signature, digital certificates generation with digital signature capacity for PDF files with the help of CryptoApplet.

Finally, it concludes with an analysis of results of tests performed by end users and the development in terms of functionality and navigability

Keywords: CMIS, ECM, Digital Signature, Java

PRESENTACIÓN

Actualmente en el Ecuador existen innumerables repositorios digitales que se los ha enfocado a distintas actividades y áreas como por ejemplo al almacenamiento de textos digitales o documentos legales, etc. Sin embargo, aún carece de repositorios que gestionen y auditen la información que se tiene, es aquí donde entra a formar parte en esto el repositorio Alfresco. En donde a pesar de que varias empresas públicas ya mantienen un sistema Alfresco, este no está adecuado a las necesidades propias de la organización por lo que no se lo utiliza eficientemente en cuanto a requerimientos (Workflows), ingreso de metadatos (modelos de contenido), etc.

Considerando todas estas falencias, se ha gestionado este prototipo que permitirá inicialmente, gestionar la documentación en base a modelos personalizados que ayudaran a mejorar y controlar el ingreso y el mantenimiento de la información, así como también, la indexación de la misma y adicionalmente, se ha insertado la firma digital como un mecanismo de seguridad que permita garantizar la validez de un documento digital almacenado en el repositorio de Alfresco.

Este trabajo pretende dar a conocer cómo y cual serían los pasos a seguir para gestionar de mejor manera el repositorio de Alfresco y cuál sería el procedimiento adecuado para brindar seguridad a la información que se desee ingresar al repositorio digital.

CAPÍTULO I

MARCO TEÓRICO

1.1 INTRANET Y EXTRANET

1.1.1 RED

Una red se la puede definir como el conjunto de dos o más equipos de hardware y software interconectados (alámbricamente o inalámbricamente) para permitir la compartición de información, recursos y servicios dentro de la red.

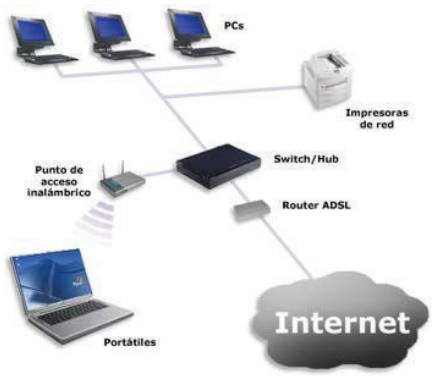


Figura 1.1. Red Computadoras. Fuente: (MADO S.A., 2013)

1.1.2 INTRANET

Es el conjunto de una o varias redes TCP/IP, las cuales tienen como objetivo fundamental, la compartición de recursos y servicios dentro de una empresa, organización o institución, pero dentro de un entorno privado.

1.1.2.1 Beneficios

- Incremento de la eficiencia y productividad de la organización
- Construye una cultura colaborativa
- Reducción de costos, ahorrando tiempo y dinero
- Posibilita la comunicación con un sin número de individuos y organizaciones
- Posibilidad de compartir un gran número de recursos y servicios

1.1.2.2 Características

- Están basadas en una arquitectura cliente – servidor
- Independiente de la plataforma en la que se esté operando
- Está construida bajo la familia de protocolos TCP/IP
- Es fácil de usar
- Escalable

1.1.3 EXTRANET

Es un conjunto de redes privadas TCP/IP interconectadas, las cuales permiten el acceso a los recursos y servicios de la red, pero solamente a determinados usuarios que estén debidamente autorizados a hacer uso de los mismos.

1.1.3.1 Beneficios

- Reducción de costos de comunicación y el papeleo en el proceso de los pedidos de sus clientes.
- Acceso a recursos e información de la red sin necesidad de estar ubicado físicamente dentro de la red privada.

- Acceso a los servicios brindados por la empresa a través de diversos medios de comunicación como el internet o modem telefónico.

1.1.3.2 Características

- Permite el acceso controlado y restringido a los recursos y servicios dentro de la red privada.
- Totalmente basada en el protocolo de internet.

1.1.4 INTERNET

Es un conjunto de redes que se manejan bajo el modelo de TCP/IP pero a nivel mundial, también es conocida como ***"la red de redes",*** debido a que esta red engloba todas las redes existentes en el mundo.

En la Tabla 1.1., se muestra una comparativa entre las características de la intranet, extranet e internet.

Tabla 1.1. Comparación entre Internet, Intranet y Extranet

	Internet	Intranet	Extranet
Acceso	Público	Privado	Restringido
Usuarios	Todo Público	Miembros de la institución o empresa	Usuarios de otras empresas que mantienen relación con la empresa dueña de la intranet
Información	Fragmentada y distribuida ampliamente	Privada	Restringida solo a ciertos servicios y recursos.

1.2 SEGURIDAD DE LA RED

Se entiende por **seguridad informática**, al conjunto de normas, procedimientos, métodos y técnicas orientadas a la protección de recursos, servicios o información dentro de una red y que tiene como fin, generar políticas que ayuden a proteger el sistema ante intrusiones indebidas o que ayuden a salvaguardar los recursos en caso de algún incidente accidental.

1.2.1 MECANISMOS DE SEGURIDAD

1.2.1.1 Seguridad Física

Tiene que ver con los mecanismos físicos que se utilizan para proteger el sistema, por ejemplo:

- Copias de seguridad
- Control de incendios
- Video vigilancia

1.2.1.2 Seguridad Lógica

Son los mecanismos y herramientas que permiten proteger la información de manera directa, por ejemplo:

- Control de acceso: Permite el acceso a la información solo a los servicios y/o usuarios autorizados
- Antivirus
- Firewalls
- Etc.

1.2.1.3 Seguridad Activa

Contempla el conjunto de medidas destinadas a reducir los riesgos de amenazas ante el sistema.

1.2.1.4 Seguridad Pasiva

Son las medidas que se toman para tratar de reducir el impacto y los efectos originados por algún incidente fortuito o fraudulento en el sistema.

1.2.2 SERVICIOS DE SEGURIDAD

1.2.2.1 Confidencialidad

Consiste en garantizar que la información solo sea accesible para los usuarios autorizados y para nadie más.

1.2.2.2 Autenticación

Proporciona al sistema la garantía de que el individuo que accedió al recurso o que generó algún contenido dentro del mismo, es quien dice ser, esto lo hace en base a la identificación del usuario, usando clave y password, tokens, certificados digitales, firma electrónica, tarjeta de coordenadas, etc.

1.2.2.3 Integridad

Asegura que los datos que han sido enviados no han sido modificados por algún tercero fuera del sistema.

1.2.2.4 No Repudio

Consiste en el hecho de no poder negar tanto el emisor como el receptor el haber

emitido o recibido algún tipo de información, esto es muy utilizado en los servicios de mensajería.

1.2.2.5 Control de Acceso

Permite controlar que tipos de usuarios tienen acceso al sistema y que permisos tienen sobre este en base a roles, departamentos o grupos.

1.2.2.6 Disponibilidad

Garantiza que la información, servicios o recursos estén disponibles para los usuarios autorizados que los requieran.

1.2.3 MÉTODOS DE SEGURIDAD MÁS USADOS ACTUALMENTE

1.2.3.1 Firma Digital

La firma digital es un proceso de cifrado matemático que permite proteger un mensaje antes de ser enviado al destinatario y garantiza fundamentalmente:

- Integridad del mensaje
- Confidencialidad de la información
- Autenticidad del contenido
- Garantiza que ninguna de las partes puede negar haber enviado o recibido el mensaje

1.2.3.2 Firma electrónica

Busca un objetivo similar a la firma digital, pero con la diferencia de que esta se asemejarse a la firma manuscrita del firmante.

1.2.3.2.1 Diferencias entre la firma digital vs firma electrónica

- La firma digital trabaja con un sistema de llave secreta y llave pública mientras que la firma electrónica la puede representar una imagen JPG[1].
- En firma digital se da un proceso de cifrado matemático, mientras que en la firma electrónica se genera un Timestamp para sellar el documento.

1.2.3.3 Certificados Digitales

El certificado digital es un documento virtual que permite identificar inequívocamente a un individuo o institución, en base a una autoridad certificadora (http://hitachi-id.com, 2014), que es una entidad de confianza, la cual de fe de la identidad del usuario o institución ante un sistema y garantiza además, que las transacciones que se realicen tengan las siguientes características:

- Autenticidad de la identidad del usuario
- La integridad de los datos transmitidos
- La confidencialidad de las transacciones

1.2.3.3.1 Formato

Los certificados digitales son similares a cualquier documento de identificación único como el pasaporte y tienen una función específica la cuál es ayudar a que el propietario de ese documento pueda ingresar a cualquier lugar para el que tenga acceso. De igual manera el pasaporte presenta un conjunto de datos importantes que permiten identificar al usuario en varios sitios, el certificado digital presenta cuatro partes importantes:

---

[1] JPG (*Joint Photographic Experts Group*) es un estándar de compresión y codificación de imágenes.

- Una clave publica
- Los datos personales del propietario
- La firma privada de la autoridad certificadora cuya función es validar la clave pública.
- El tiempo de validez del certificado digital.

Todos estos datos se pueden apreciar en la Figura 1.2:

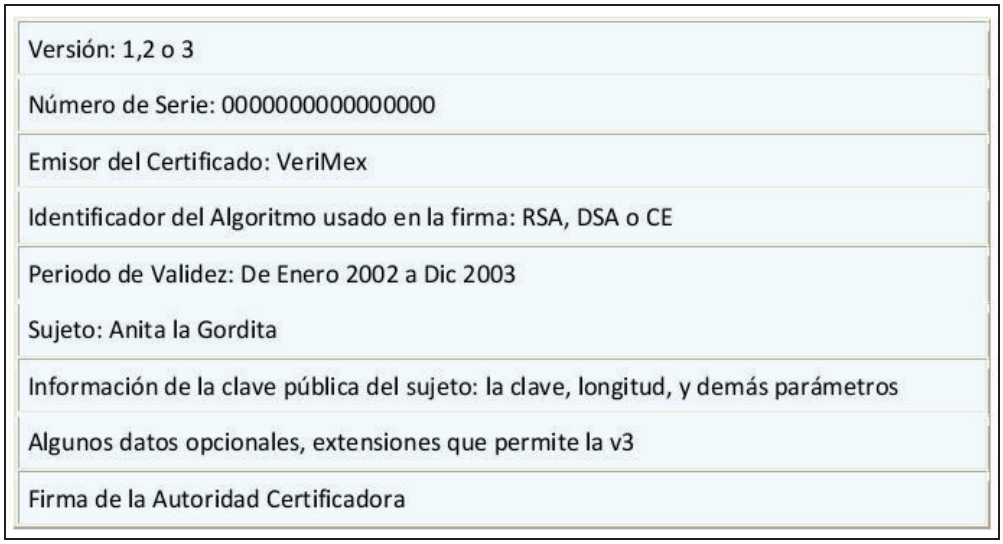


Figura 1.2. Formato del Certificado Digital. Fuente: (Castilla-La Mancha, 2013)

1.2.3.3.2 Tokens de Seguridad

Un token de seguridad es un dispositivo electrónico que posibilita la autenticación de un usuario ante un sistema, aplicación o redes informáticas y a su vez permite utilizar los datos del propietario para firmar y cifrar digitalmente mensajes que se envíen a través de la red.

1.2.3.3.3 Características

- Es un dispositivo electrónico que está conformado por un chip criptográfico y una interfaz USB para la conexión con el computador.
- Presenta una capacidad mínima de 32 Kb.
- Almacena internamente un certificado digital del propietario del token, con el cual se autentica el usuario ante la realización de cualquier tipo de transacción. Véase la Figura 1.3.

Figura 1.3. Token de seguridad. Fuente: (Vialgroup, 2013)

1.2.3.4 CryptoApplet

CryptoApplet es un proyecto de código abierto desarrollado por la Universidad de Jaume, la cual tiene la finalidad facilitar la firma digital usando certificados digitales o tokens de seguridad.

1.2.3.4.1 Formatos de firma que permite CryptoApplet

CryptoApplet permite firmar documentos en los siguientes formatos:

Firma en bruto: Habitualmente usado para firmar varios documentos a la vez y

generalmente el formato de estos documentos es PDF[2].

CMS (https://tools.ietf.org, 2009)/PKCS#7: Permite firmar documentos completos o parciales usando los repositorios de certificados de Microsoft Windows o Linux con los navegadores web.

XMLDsign (http://www.w3.org, 2013): Este formato está definido por la W3C (http://www.w3c.es, 1994) y esta define la sintaxis que debería de tener el documento XML para que se pueda realizar la firma digital.

XADES-X-L en formato DigiDoc: Es una extensión a la firma sobre XML-Dsign, la cual permite generar una "***firma electrónica reconocida en el sentido de la directiva 1999/93/EC de la Unión Europea***", (w3c, 2003). Esta firma electrónica permite tener una validez mayor, dado que almacena toda la información de validación (embebida en documento firmado) por la que haya cursado en el tiempo posibilitando la verificación de la misma por varios años aunque la CA ya no exista.

Factura-e: Esta es una funcionalidad adicional que tiene este aplicativo, el cual originalmente fue creado para firmar facturas electrónicas en toda España, su funcionamiento radica en la generación de un archivo XML con los datos rubrica de la persona o entidad. Este documento XML deberá estar siempre presente para que se pueda validar como correcta la firma.

TSA (Timestamp): Esta técnica permite agregar datos de tiempo a la firma digital.

---

[2] PDF (Portable Document Format) es un formato desarrollado por Adobe para el intercambio de documentos electrónicos. Fuente: (Adobe Systems, s.f.)

1.3 ENTERPRISE CONTENT MANAGEMENT – ECM[3]

1.3.1 DEFINICIÓN

Una definición sería: *"La gestión de contenido empresarial o Enterprise Content Management (ECM) es una de las estrategias y tecnología empleadas en la industria de la tecnología de la información para manejar la captura, almacenamiento, seguridad, control de versionado, recuperación, distribución, conservación y destrucción de documentos y contenidos. Evidentemente se debe entender que los Documentos también son contenidos y se deben ser gestionados como tales."* (Bernal, s.f.)

Otros autores independientes definen a un ECM como: *"Un ECM es un set integrado de tecnologías que administran el contenido y su ciclo de vida, mejorando la productividad del negocio y mitigando riesgos y gastos excesivos en los volúmenes de contenido"* (OPENTEXT, s.f.).

Se podría concluir como definición de un Enterprise Content Management (ECM), como una infraestructura tecnológica que se encarga de gestionar y administrar el contenido empresarial de una organización y que además posibilita mantener un control total sobre estos, por ejemplo:

- Almacenamiento de contenidos
- Control de versionado
- Control de acceso a los contenidos
- Destrucción de contenidos, etc.

1.3.2 TIPOS DE ECM

Debido al incremento de la documentación en las distintas organizaciones, sean

---

[3] ECM (*Enterprise Content Management*) es una plataforma para la gestión de contenido empresarial.

estas, públicas o privadas, se hace imposible gestionar grandes volúmenes de contenido. Para tratar de mejorar esta situación se ha implementado varias soluciones, siendo la principal la compartición de contenido por la intranet, sin embargo, esta solución no fue muy aceptada por la ineficacia en la gestión del contenido compartido, esto debido a varias situaciones:

- Gestión compleja
- Falta de backups
- Insuficiencia de control de acceso a los contenidos compartidos
- No hay concurrencia
- No hay indexación ni un control de versionado del contenido

Por estos y otros motivos, la intranet dejó de ser utilizada para estos fines y se integraron nuevas soluciones documentales a estas organizaciones dependiendo de la plataforma que se maneje.

1.3.2.1 Plataforma Windows

1.3.2.1.1 SharePoint

SharePoint nace como alternativa de Microsoft fácil de implementar y manejar debido a una interfaz ágil y dinámica de administración de contenidos, esta plataforma atiende varias deficiencias del mercado documental.

Características

- Gestión mediante bibliotecas
- Estructura basada en carpetas
- Atributos en carpetas y ficheros

- Control de versiones y backups
- Retención de información
- Colaboración con usuarios
- Seguridad con Active Directory

Ventajas

- Portal Interno de acceso para los usuarios
- Gestión de permisos mediante Active Directory
- Reducción de costos de administración de contenido
- Acceso y consulta rápida de contenidos
- Mejora productividad
- Fomenta el trabajo colaborativo.

Desventajas

- Licencias costosas
- Software propietario

1.3.2.2 Plataforma Linux

1.3.2.2.1 Alfresco

Alfresco es un ECM que nació como una iniciativa de un grupo de desarrolladores de software en el año 2005 y cuyo objetivo es el de gestionar de una manera eficiente todo el contenido involucrado dentro de una organización.

Características

- Gestión de documentos
- Gestión de contenido web
- Versionado a nivel de repositorio
- Gestión de registros
- Gestión de imágenes
- Flujos de trabajo basados en BPM[4].
- Indexación y búsquedas inteligentes de documentos
- Soporte multiplataforma.

Ventajas

- Tiene una solución Open Source completa
- Soporta varios lenguajes de programación entre ellos: Java, PHP, Visual Net, etc.
- Fácil de usar
- Alta disponibilidad
- Soporta fórums, chat, etc.

Desventajas

- Tiene un inicio lento de los servicios al iniciar el servidor
- Mantienen bloqueadas el uso de las APIS para el desarrollo en la nube.

---

[4] BPM(*Business Process Management*) es una técnica para la gestión empresarial para el manejo de flujos de trabajo orientados a procesos propios de las organizaciones.

1.3.2.2.2 Nuxeo

Nuxeo al igual que Alfresco es una solución libre multiplataforma, basado en estándares abiertos, de escala empresarial y que permite la gestión de documentos de cualquier tipo.

Características

- Todos las características de Alfresco
- Auditoria de documentación

Ventajas

- Fácil manejo
- Robusto
- Posibilita el arrastrar y soltar documentos dentro de Nuxeo
- Fácil colaboración
- Pre visualización de contenido
- Se puede extender su funcionalidad con Java.

Desventajas

- Inconvenientes técnicos cuando se desea almacenar documentos de gran tamaño.
- Problemas con la edición de documentos en línea
- Difícil de configurar la plataforma en el servidor.

1.3.2.2.3 Comparativa de Alfresco vs Nuxeo

En la Figura 1.4 se puede apreciar un cuadro comparativo entre las dos tecnologías de documentación mejor valoradas a nivel mundial en cuanto a sus principales características.

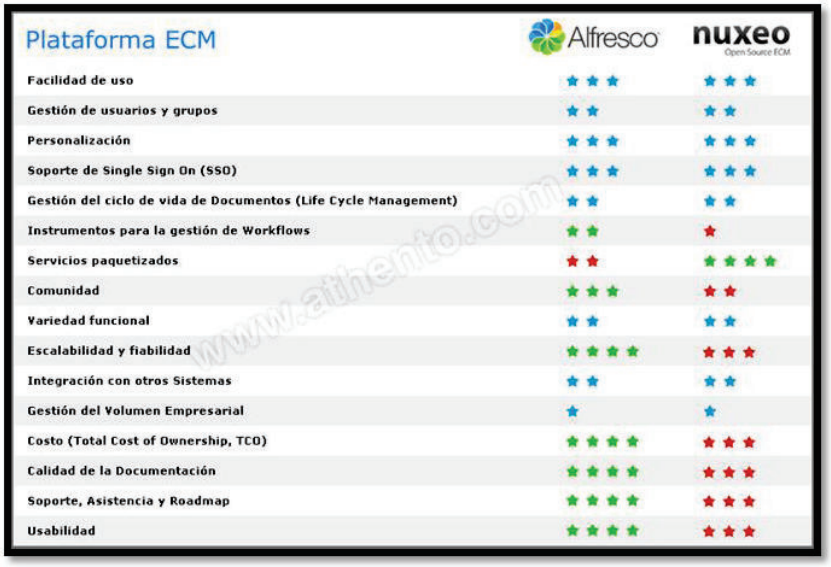


Figura 1.4. Cuadro comparativo de ECM's. Fuente: (athento, 2009)

1.3.3 ALFRESCO

Como se mencionó anteriormente, Alfresco es un gestor documental que ofrece excelentes características en comparación con otros gestores documentales, sin embargo, presenta una amplia gama de herramientas que facilitan la extensión de sus funcionalidades para adaptarse a los requerimientos de la empresa.

1.3.3.1 Arquitectura de Alfresco

Alfresco está compuesto por un conjunto de capas como se puede apreciar en la Figura 1.5, las cuales posibilitan un funcionamiento empresarial adecuado para la gestión de la documentación. A continuación se presenta un breve resumen de

las principales características que ofrece cada capa de Alfresco.

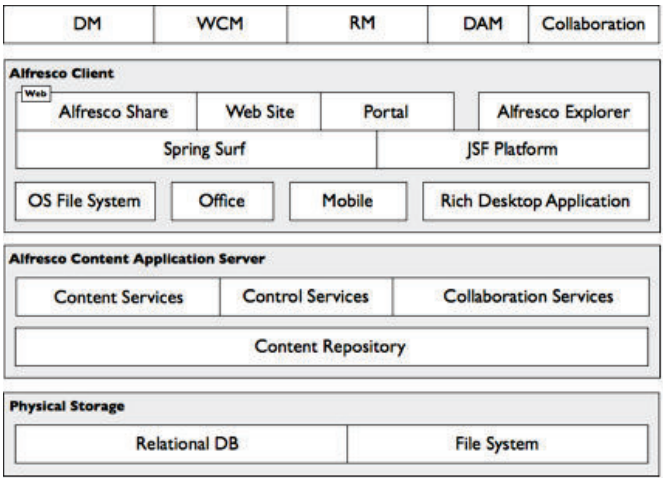


Figura 1.5. Arquitectura Alfresco. Fuente: (Alfresco, 2012)

1.3.3.1.1 Capa General

Esta capa está encargada de la gestión documental y está compuesta por las siguientes capas:

Document Management (DM)

- Esta encargada de la recolección, organización y la compartición de los documentos que son generados dentro de una organización.
- Su principal objetivo es la administración de la documentación.

Web Content Management (WCM)

- Esta posibilita la administración del contenido documental a través de un navegador web.
- Su principal característica es la gestión de documentación interna que no

será publicada fuera de la organización.

Digital Asset Management (DAM)

- Encargado de la administración de gráficos, videos y audio.
- Es similar al DM pero con la característica principal que ofrece funcionalidades especializadas para trabajar con contenido multimedia.

Record Management (RM)

- Esta encargado de la gestión legal de los registros dentro de Alfresco.
- Una de las principales características es el manejo de políticas, planes de registro y auditoría de los documentos que se gestionen en Alfresco.

1.3.3.1.2 Alfresco Client

Presenta un set de herramientas que se pueden personalizar para ser usadas dentro de la organización en un portal web, así tenemos:

Alfresco Explorer

- Es una interfaz de usuario que posibilita una rápida personalización de los modelos que se aplicarán dentro de una empresa.
- Su principal característica es la facilidad de expandir la funcionalidad de Alfresco usando la plataforma de java web denominada Java Server Faces (ORACLE, s.f.)

Alfresco Share

- Se caracteriza por ser más fácil para administrar y gestionar la

documentación que ingresen los usuarios, además posibilita las mismas características que Alfresco Explorer, pero con una interfaz más rica para el usuario final.

- Alfresco Share se puede extender usando diferentes tecnologías, por ejemplo se puede utilizar la plataforma Net para el caso de Windows como una plataforma de pago y para el caso de software libre, se puede usar java, php, perl, entre otros.
- Al igual que en Alfresco Explorer se podía extender su funcionalidad con JSF, en Alfresco Share, también se podría o no usar esta tecnología, debido a que resulta más fácil de modelar las necesidades de la empresa usando Spring (Spring, s.f.).
- Spring es una tecnología modular WEB que se puede implementar usando Java y cuya principal característica es la posibilidad de trabajar en un mismo proyecto, pero con la diferencia de que se puede asignar a varios grupos de personas el desarrollo de diferentes módulos y cuyo resultado final, es la unión de todos estos módulos para generar la aplicación final.

1.3.3.1.3 Alfresco Content Application Service

Se caracteriza por ser el corazón de Alfresco debido a que está encargada de mantener y gestionar el contenido que se encuentra almacenado en el repositorio de datos.

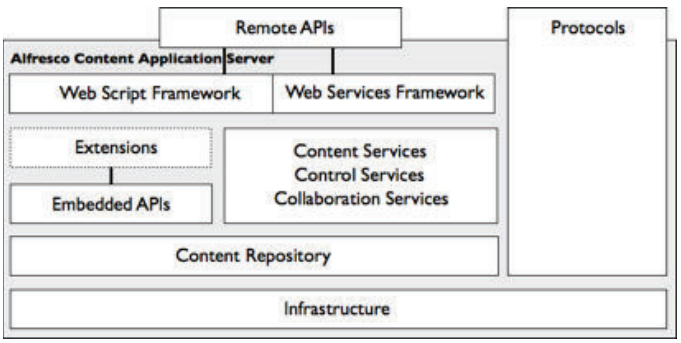


Figura 1.6. Alfresco Content Service. Fuente: (Alfresco, 2013)

Como se puede visualizar en la Figura 1.6, Alfresco presenta un set de interfaces remotas públicas que permiten la comunicación con el cliente remoto y además permiten extender su funcionalidad, así tenemos:

Remote API

Es un set de interfaces que son usadas principalmente para extender las funcionalidades de Alfresco usando un lenguaje de programación.

Protocol Bindings

- Están encargados de mapear los servicios que se manejara entre el servidor y el cliente, es decir, son los encargados de gestionar la comunicación entre ambas partes para ejecutar alguna acción específica.
- La unión de estas dos interfaces permite enriquecer las funcionalidades que puede brindar Alfresco para la modelación y puesta en marcha de los requerimientos de una empresa.

1.3.3.1.4 Alfresco Content Repository

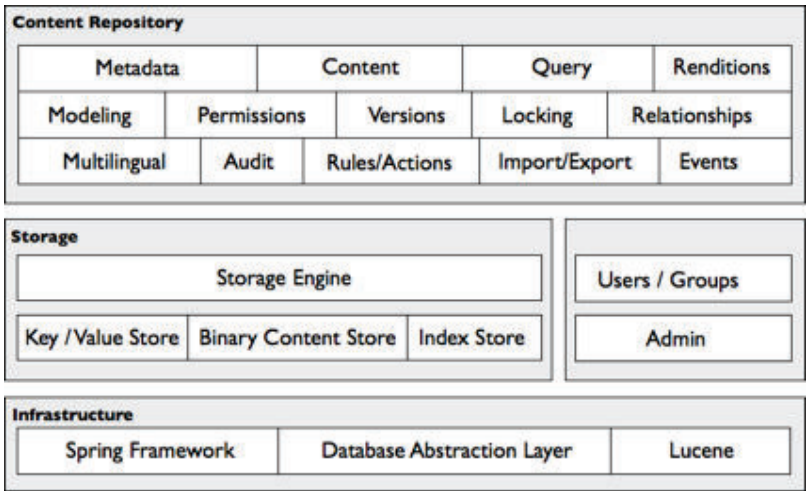


Figura 1.7. Alfresco Content Repository. Fuente: (Alfresco, 2013)

Esta capa está encargada del almacenamiento de la información en el repositorio y de las configuraciones y profiles de los usuarios, documentos, reglas, actividades, etc. Véase la Figura 1.7.

El repositorio de Alfresco implementa los estándares CMIS[5] (Chemistry Apache, s.f.) y JCR (Java Dzone, 2010).

Estándar OASIS CMIS

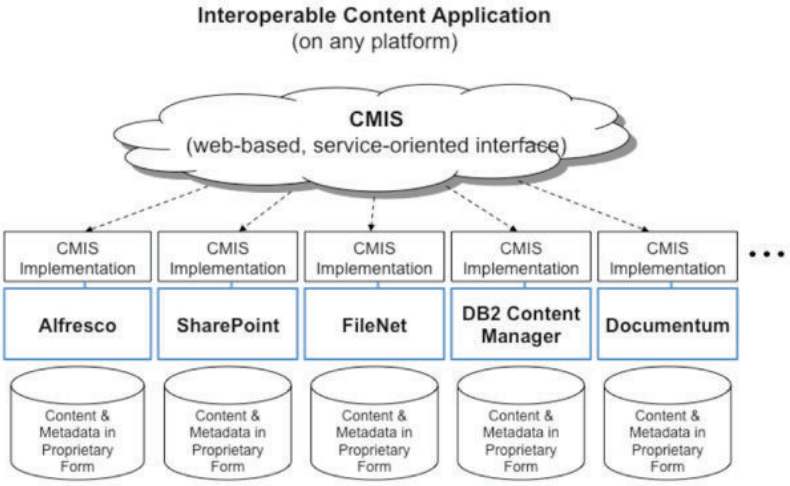


Figura 1.8. Funcionamiento del estándar CMIS. Fuente: (blyx, 2013)

El estándar OASIS CMIS, define una serie de reglas que se encuentran diseñados por los principales fabricantes de gestores de contenido ECM y cuya principal función es la de brindar un servicio orientado a interfaces que sea interoperable y que no esté limitado solamente a un sistema ECM en concreto, como se muestra en la Figura 1.8.

---

[5] CMIS(*Content Management Interoperability Services*) es un framework OpenSource para la gestión de repositorios empresariales de contenido documental.

Alfresco al estar basado sobre el estándar CMIS, permite la gestión y administración del contenido que se almacenará en el repositorio a un nivel granular, de la misma manera define una serie de reglas y acciones que se pueden ejecutar como por ejemplo:

- Consultas de contenido
- Control de acceso
- Versionado de la documentación
- Eventos
- Auditorias

Estándar JCR[6] (JSR– 170)

El estándar JCR define los procedimientos a seguir para la implementación de un modelo de repositorio usando Java. Actualmente Alfresco está definido sobre este estándar para la gestión del repositorio y se ha usado la especificación JSR-170 (Jcp, s.f.) para la implementación del mismo.

Este estándar define dos conceptos importantes que se manejan en Alfresco:

- **Nodo**: Esta encargado de proveer la metadata y la estructura del contenido almacenado.
- **Contenido**: Es el archivo o documento.

Alfresco Content Services

Alfresco presenta una serie de servicios tanto públicos como internos, muchos de los cuales pueden ser sobreescritos o personalizables para ejecutar una acción más compleja.

---

[6] JCR(*Java Content Repository*) es el gestor de repositorios de contenido propio de Java

Esta subcapa permite la administración y gestión de la documentación, así como también, posibilita la incrementación de las funcionalidades de Alfresco usando diversos recursos como por ejemplo:

- Java
- Scripting
- RestFull[7]
- Web Services
- Interfaces de Usuario, tales como agregación de datos para ingresar en la metadata de los documentos que se ingresen en Alfresco Share o Alfresco Explorer.

1.3.3.2 Extendiendo la funcionalidad de Alfresco

1.3.3.2.1 Web Services

Web Services es una tecnología relativamente nueva y muy versátil, debido a su interoperabilidad con distintas tecnologías y a la facilidad de desarrollo que está presenta a las organizaciones.

La principal característica que presentan los Web Services, es la amplia gama de protocolos y estándares libres que posibilitan la interoperabilidad con las distintas tecnologías, así podemos citar algunas de las más importantes:

- SOAP[8]

---

[7] RestFull (*Representational State Transfer*) es una tecnología remota que utiliza servicios web asociados al protocolo JSON.

[8] SOAP (*Simple Object Access Protocol*) es un protocolo de acceso a objetos planos utilizado en servicios web.

- HTML
- WDSL[9]
- XML
- UDDI[10]

Todos estos protocolos y estándares se unen para implementar una infraestructura fuerte tanto para generar como para consumir estos recursos, por este motivo, los Web Services han tomado rápidamente un gran renombre al momento de ser usados en el campo empresarial.

Antes de pasar a otro tema, es necesario definir cuál es el funcionamiento de los servicios web, es por esta razón que se empezará describiendo el protocolo SOAP y su interrelación con el resto de protocolos.

1.3.3.2.2 SOAP – Simple Object Access Protocol

Es un protocolo basado en XML (Ietf, 2007) que permite el intercambio de información entre dos o más entidades mediante el envío de mensajes serializados a través del internet usando el protocolo HTML (Ietf, 1995). Veasé la Figura 1.9.

---

[9] WSDL (*Web Service Definition Language*) es un contrato que se entrega para poder interactuar con servicios web.

[10] UDDI (*Universal Description, Discovery and Integration*) es un estándar que trabaja en conjunto con el contrato del servicios web y que busca brindar las información necesaria para operar con los mensajes SOAP.

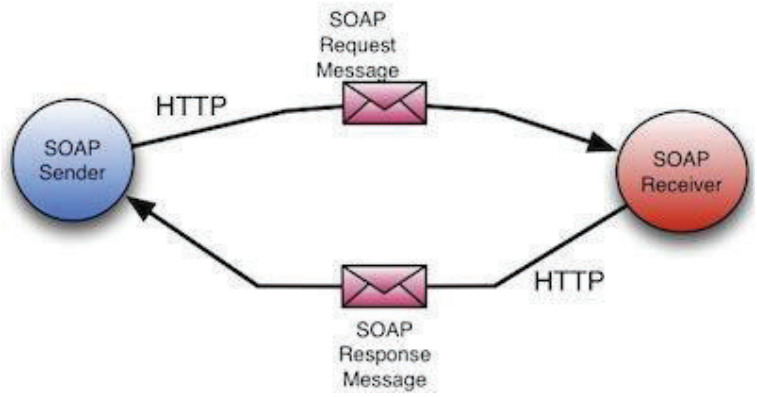


Figura 1.9. Protocolo SOAP. Fuente: (java-forums, 2012)

Funcionamiento del protocolo SOAP

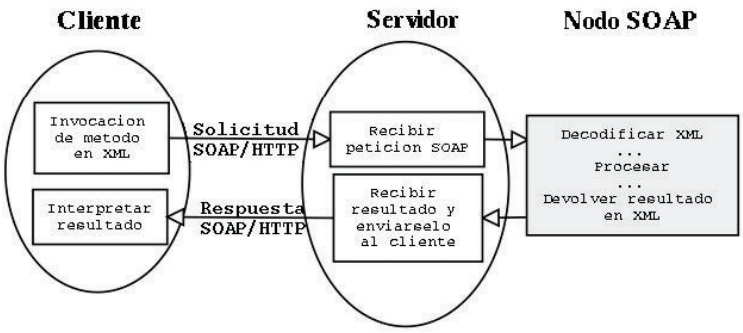


Figura 1.10. Funcionamiento SOAP. Fuente: (Franco, 2001)

En la Figura 1.10 se puede apreciar cuando una entidad solicita determinada información o solicita el acceso a un método de un servicio web, esta genera una serie de peticiones que están encapsuladas en mensajes XML. Para identificar la petición se realiza una serie de operaciones básicas que se detallan a continuación:

1. Se identifica el servicio al que se quiere tener acceso usando una base de

datos básica que almacena la información de todos los Web Services registrados en el servidor, denominada UDDI (Universal Descriptor, Discovery and Integration).

2. Una vez localizado el web services, se analiza las funcionalidades que presenta apoyándose en un archivo XML, que está definido por la notación WSDL (Web Services Description Language) dada por la W3C. Este archivo XML define los métodos SOAP que se pueden intercambiar, los protocolos que se deben usar y la localización de los recursos necesarios para su adecuado funcionamiento.
3. Una vez localizado los métodos que se solicitan, se ejecutan los procedimientos correspondientes y se genera la respuesta.
4. Esta respuesta a la solicitud generada, es encapsulada usando el protocolo XML, es serializada y por último enviada de regreso al cliente que la solicito.

Partes del mensaje SOAP

Un mensaje SOAP está compuesto por tres partes como se muestra en la Figura 1.11, las cuales definen las propiedades y características principales que serán almacenadas como parte del mensaje al momento de ser enviado usando algún protocolo.

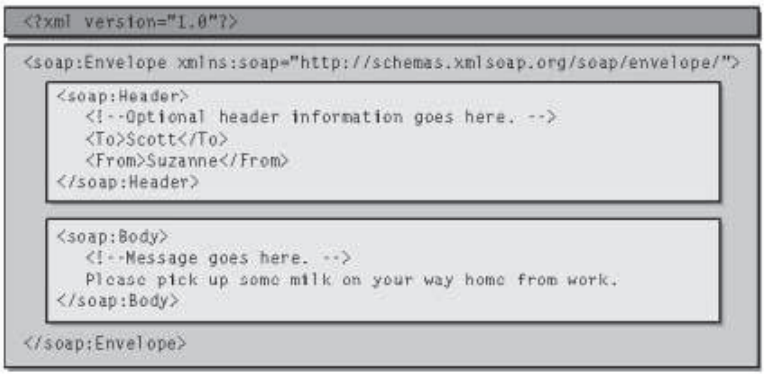


Figura 1.11. Formato Mensaje SOAP. Fuente: (Gonzáles, 2004)

Estas partes se definen a continuación de una manera más detallada:

- **Envelope o Envoltura:** Define la estructura que contendrá el mensaje y está compuesto por el Header y el Body.
- **Header o Cabecera del mensaje:** El Header define datos opcionales del mensaje, en la Figura 1.11., se muestra dos parámetros: el destinatario y el emisor.
- **Body o Cuerpo:** Define los datos a ser enviados en el mensaje.

1.3.3.2.3 RESTFULL Web Services

Rest (Representational State Transfer) es un estilo de programación que se fundamenta en las características que hicieron de la arquitectura web una de las redes más exitosos y extensibles del mercado, añadiendo restricciones sobre las interacciones de los diferentes componentes que contiene.

En comparación con los Web Services, se podría decir que es una implementación ligera y que se adapta de mejor manera a la arquitectura web, tanto pasada como presente. Sin embargo, a pesar de ser un concepto nuevo, este ha tenido una buena acogida por parte de los desarrolladores debido a la facilidad con la que pueden implementar sistemas híbridos interoperables usando diferentes tecnologías que ofrece el mercado como son JavaScript o XML.

Ventajas de RestFull vs Web Services

- La implementación de RestFull resulta más ligera que implementar Web Services con SOAP
- No se requiere ninguna herramienta de desarrollo para su implementación
- Fácil de Implementar
- Es escalable.
- Se puede usar el estándar WSDL en ambos casos.

Desventajas de RestFull vs Web Services

- Es más fácil consumir servicios web que servicios generados con RestFull
- Los Web Services soportan un mayor número de estándares que los servicios RestFull
- Al ser RestFull un estilo nuevo, hay pocos sistemas que usan esta técnica en su desarrollo.

Como conclusión se puede decir que al trabajar con Web Services, lo que importará al final, es la manera como se desee implementar la solución de software que se requiera. Alfresco mantiene ambas tecnologías disponibles para su desarrollo, sin embargo, en este tema de titulación se usaran ambas tecnologías debido a sus características y prestaciones que brindan para brindar soluciones escalables a largo plazo.

1.3.4 NORMATIVA DE DOCUMENTACIÓN VIGENTE

1.3.4.1 Nacional

En el Ecuador actualmente se maneja en cuanto a normativas de gestión documental y archivística, las siguientes normas:

- NTE INEN 2410:2010. Documentación, elaboración de oficios, oficios circulares, memorandos, memorandos circulares y circulares. Requisitos.
- Ley Nº 2001 – 67. Ley de Comercio Electrónico, firmas electrónicas y mensajes de datos y su reforma.

A pesar de que el estado ha tratado de mejorar la situación precaria de las entidades estatales con la ayuda de un sistema gestor documental denominado QUIPUX, este no ha podido satisfacer la demanda debido a la baja concurrencia

de usuarios que puede manejar. Sin embargo, este sistema está implementando la normativa nacional de lo que en resumen se podría obtener es:

1.3.4.1.1 Norma NTE INEN 2410

Esta norma define una serie de acrónimos y abreviaturas para identificar los siguientes puntos relevantes en la elaboración de memos, oficios y circulares:

- Anexo: Documentos adjuntos
- Asunto: Síntesis del contenido del texto
- Clasificación del documento: Nivel de seguridad
- Circular: Textos de igual contenido dirigidos a un grupo de personas para dar a conocer algo.
- Código de Referencia: Números y letras que identificación el documento
- Elaborador: Persona responsable que redacta.

Posteriormente se definen los requisitos para la elaboración de los documentos, entre los cuales se especifican:

- Diagramación del documento (márgenes, bordes y texto)
- El formato pertinente para codificación del documento
    - IESS-DF-2010-001-OF
    - Guayaquil, 09 de enero de 2010
- Datos del destinatario
    - Señor, Señora, Señora Licenciada, Señor Magister
- Cargo
    - Formato con negrita
        - **Gerente General de Operaciones**

BANCO CENTRAL ECUADOR

Para finalizar se anexan algunos ejemplos de formatos que se deberían de seguir para la realización de los mismos dentro de cualquier institución pública.

1.3.4.1.2 Ley de Comercio Electrónico, firmas electrónicas y mensajes de datos

Esta ley publicada el 17 de abril del 2002 en el registro oficial Nº 577 especifica una serie de reglas que se deben de seguir para regular el comercio electrónico, firmas electrónicas y mensajes de datos, para esto, se especifican los siguientes bloques de acción:

De las firmas electrónicas

- "*La firma electrónica tendrá igual validez y se le reconocerán los mismos efectos jurídicos que a una firma manuscrita en relación con los datos consignados en documentos escritos, y será admitida como prueba en juicios*" (CONGRESO NACIONAL, 2002)
- Adicionalmente especifica las obligaciones que posee el titular de la firma electrónica.
- De los certificados de firma electrónica
    - Especifica los requisitos legales para la obtención de este certificado, la duración del mismo.
- De las entidades de certificación
    - Especifica los requisitos que deben de cumplir las entidades, tanto estatales como internacionales, para ser consideradas entidades de certificación dentro del Ecuador.
- De las infracciones informáticas
    - Se estipula las clausulas penales en caso de infringir en acciones fraudulentas, por ejemplo:

- Uso indebido de la firma electrónica
- Acceso a datos de usuarios con fines ilícitos y sin permiso.
- Falsificación electrónica.

- Del organismo encargado de hacer cumplir todas estas disposiciones
  - El organismo encargado de hacer ejecutar este reglamento es el Consejo Nacional de Telecomunicaciones (CONATEL)

1.3.4.2 Internacional

1.3.4.2.1 ISO/IEC 15489 Un marco sistemático para las buenas prácticas de gestión documental en organizaciones

Esta norma fue creada en Ottawa, Canadá, en septiembre de 2000, y define lo siguiente:

- Las directrices para la gestión y administración de registros
- Define los formatos a seguir para estandarizar la documentación dentro de la organización
- Ofrece una terminología relacionada con la gestión documental
- Especifica las revisiones a realizarse una vez implementada esta norma.

De esta norma se generó una segunda parte que se centra más concretamente en la gestión metadática de los registros y en las maneras de facilitar las búsquedas en base a Tesauros, control de vocabulario, etc.

1.3.4.2.2 ISO 9075 Tecnologías de la Información – Lenguajes de Bases de Datos SQL.

Esta norma define la estructura del motor de bases de datos que se debería usar

de acuerdo a las necesidades de los usuarios para la gestión de las tecnologías de la información, está compuesta de 14 partes:

- Framework
- Foundation
- Interfaces CLI
- Módulo de persistencia
- Administración de datos externos
- Vinculación de lenguajes de objetos
- Definición de esquemas
- Rutinas SQL y tipos usados en JAVA
- XML

Se ha incluido a esta norma ISO[11], debido a que se usará Alfresco como el sistema ECM gestor de documentación, esta norma regula el funcionamiento base de Alfresco en conjunto con la norma ISO 15489.

En conclusión, se tiene que Ecuador está muy cercano a aplicar estas normativas en el país, sin embargo, al utilizar QUIPUX como el sistema definitivo de documentación estatal, se incurre en un grave error debido a la falta de mantenimiento del sistema y a la insuficiencia de capacidad para gestionar un gran número de usuarios simultáneos.

Alfresco ha venido posicionándose en Ecuador de una manera rápida, a tal nivel que ya se usa en diferentes instituciones públicas como son el INEN, SUPERTEL,

---

[11] ISO(*International Organization for Standardization*) es una entidad encargada de ;a generación de normas a nivel de fabricación, comercio y comunicación a nivel mundial.

etc., esto es por el soporte y flexibilidad para la aplicación de nuevos estándares tanto nacionales como internacionales y por su gran soporte para gestionar usuarios concurrentes debido a que trabaja con Java.

CAPÍTULO II

DISEÑO DEL SISTEMA WEB

2.1 QUÉ ES LA COMIEX?

La COMIEX es el organismo institucional designado por Consejo Politécnico para la gestión de proyectos de investigación y extensión que se puedan generar dentro de la EPN. Las principales funciones que desempeña son:

a. Seguimiento a los proyectos de investigación que se desarrollen.
b. Gestión de todos los procesos administrativos para facilitar los materiales e insumos que se requieran para la ejecución de los proyectos de investigación.
c. Generación de capacitaciones que incentiven a los docentes a realizar investigación dentro de la EPN.

Cabe destacar que la EPN gestiona tres tipos de proyectos de investigación:

- **Proyectos de Investigación Semilla:** Son proyectos financiados por la EPN que se crearon con la única finalidad de incentivar al personal docente para que incurra en la investigación. Estos proyectos tienden a ser la base para generar proyectos de mayor envergadura.

- **Proyectos de Investigación Internos:** Son proyectos que se manejan a nivel interno, dentro de la EPN, cuya principal diferencia con respectos a los proyectos de Investigación Semilla, es que estos no tienen ningún financiamiento por parte de la EPN.

- **Proyectos de Investigación Externos:** Estos proyectos se desarrollan con organismos gubernamentales primordialmente debido a que estos financian prácticamente toda la investigación.

1.4 ANÁLISIS DE LA SITUACIÓN ACTUAL DE LA COMIEX

1.4.1 ANÁLISIS DE RECURSOS UTILIZADOS

1.4.1.1 Documentos Procesados

Actualmente, la COMIEX gestiona más de 60 proyectos de investigación, de los cuales todos deben presentar varios tipos de documentos para la realización de las siguientes actividades:

- Solicitud de disponibilidades presupuestarias
- Solicitud de modificación de rubros
- Solicitud de rembolsos de facturas
- Solicitud de pagos de facturas
- Solicitud de pago de pasantes
- Solicitud de adquisiciones de equipos
- Entre otros

De todos estos trámites el número de peticiones en promedio desde el mes de enero hasta el mes de octubre de 2013 se puede apreciar en la Tabla 2.1.

Tabla 0.1. Número de trámites promedio

Solicitudes Gestionadas por la COMIEX	# Peticiones	# hojas/trámite	#P*#Hojas/T
Modificación de rubros	15	2	30
Disponibilidades presupuestarias	110	5	550
Anulaciones de disponibilidades presupuestarias	8	2	16
Adquisiciones de equipos	43	7	301
Pago de facturas	79	7	553
Rembolsos de facturas	6	4	24
Otros	38	8	304
TOTAL			1778

De estos valores se obtiene en promedio 118 hojas que se usarían mensualmente.

En la Figura 2.1 se puede apreciar las peticiones que más se gestionan en la COMIEX son:

1. Solicitudes de disponibilidades presupuestarias
2. Pago de facturas
3. Otros
4. Modificaciones presupuestarias
5. Anulaciones de disponibilidades presupuestarias
6. Rembolsos

Figura 0.1. Número de trámites promedio, Fuente: COMIEX

En Figura 2.2 se puede observar la línea de tendencia que se maneja en un periodo de tiempo de 10 meses, sin embargo, esto cambiará debido a las nuevas convocatorias para la presentación de proyectos en el presente año.

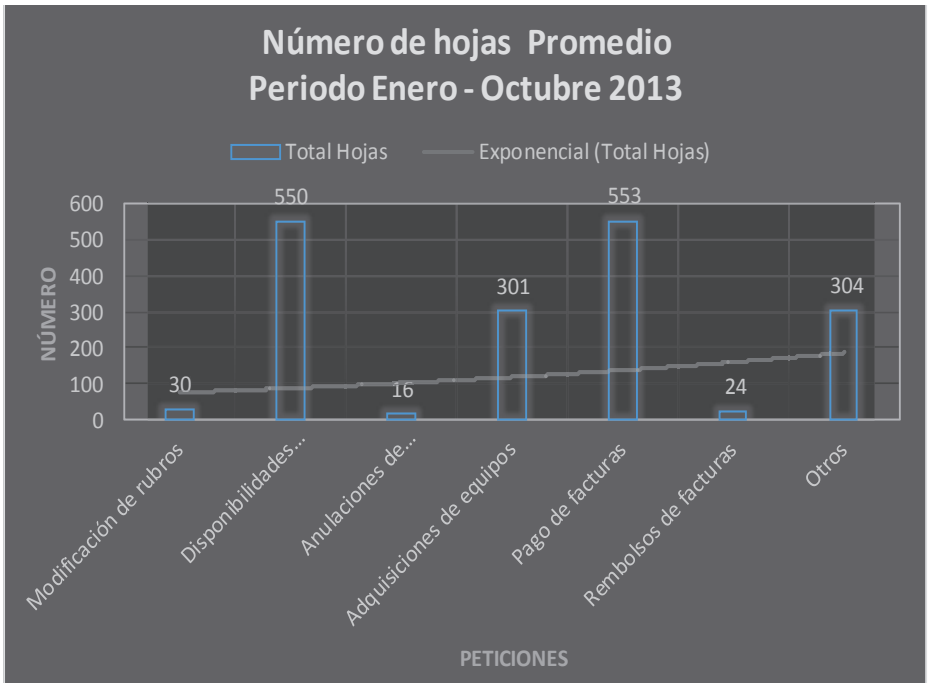


Figura 0.2. Número de hojas promedio, Fuente COMIEX

1.4.1.2 Materiales de oficina

El procesamiento de cada trámite requiere de varios materiales e insumos mínimos por mes tal como se puede Tabla 2.2.

Tabla 0.2. Gastos en insumos de oficina / mes

Materiales	Número Promedio/mes	Costos
Hojas	780	$ 7.50
Clips	150	$ 3.50
Marcadores	2	$ 2.00
Tinta de impresora	2 tóner	$ 80.00
Otros insumos (esferos, lápices, resaltadores, etc.)	10	$ 8.00
TOTAL		**$ 101.00**

1.4.1.3 Relación Tiempo – Dinero

Tabla 0.3. Relación de tiempo / minutos

Proceso	Tiempo estimado/minutos	Tiempo Estimado/mes
Digitalización	1	22
Gestión del tramite	10	220
Impresión	3	66
Mensajería	5-60	110 - 1320
Total	19-60	418 - 1320

Todos estos documentos incurren en algún gasto, sea este de tiempo o dinero es así que, para gestionar un trámite el promedio de tiempo calcular mediante la Tabla 2.3.

Y considerando que al día en promedio existen 5 trámites esto nos da un total de 300 minutos, sin considerar el periodo de tiempo que demora en generarse una respuesta por el receptor, que aproximadamente puede durar entre dos a cuatro días y en algunos casos hasta varias semanas. Adicionalmente, el costo que se genera por este proceso depende del valor hora de la persona que colabora en la gestión, por lo que este valor es variable y al momento no es contabilizable.

1.5 FUNCIONAMIENTO DE LA COMIEX

1.5.1 DEPARTAMENTOS

La COMIEX trabaja conjuntamente con varios departamentos para la gestión de los proyectos de investigación, dichos departamentos se listan en la Tabla 2.4.

Tabla 0.4. Departamentos relacionados con la COMIEX

Departamento	Función que ejecuta ante la COMIEX
Vicerrectorado	Encargada de aprobar los documentos que se gestionarán para la ejecución de los proyectos de investigación.
Dirección Administrativa	Encargada de gestionar los pagos y adquisiciones de equipos, suministros de oficina, reactivos,

	insumos, etc.
Dirección Financiera	Encargada de gestionar la emisión de Certificaciones Presupuestarias para las adquisiciones que se requieran para los proyectos y adicionalmente, es la delegada de realizar los pagos.
Unidad de Gestión de Proyectos	Esta dependencia está vinculada directamente con la Dirección Financiera y es la encargada de gestionar los proyectos externos de la EPN.
Dirección de Recursos Humanos	Encargada de gestionar la contratación de personal para los proyectos de investigación.
Dirección de Planificación	Encargada de gestionar la planificación de los proyectos de investigación, entre otras actividades.

1.5.2 PROCESOS ADMINISTRATIVOS

Un proceso es un conjunto de análisis detallados en base a fases o etapas de las actividades que se desarrollan en una organización.

Conjuntamente con cada departamento se gestionan varios procesos administrativos los mimos que se detallan en la Tabla 2.5.

Tabla 0.5. Procesos desarrollados por departamento

Proceso	Departamentos Involucrados	Descripción
Aprobación General	Vicerrectorado	Firma de memorandos, oficios o circulares.
Disponibilidad de Certificación Presupuestaria (CP)	Vicerrectorados y Unidad de Gestión de Proyectos (UGP)	Mediante memorando firmado por el Sr. Vicerrector, se procede a solicitar a la UGP la CP correspondiente para la adquisición de equipos, reactivos, etc. Para los proyectos de investigación.
Adquisiciones de materiales, insumos y equipos de laboratorio.	Vicerrectorado y la Dirección Administrativa	Mediante memorando firmado por el Sr. Vicerrector, se procede a solicitar la adquisición de los equipos e insumos que se requieran para ejecutar un proyecto en base a la CP solicitada con anterioridad.
Pago de Facturas	Vicerrectorado y Dirección Administrativa	Mediante memorando firmado por el Sr. Vicerrector, se procede a

		solicitar la cancelación de los haberes correspondientes a los equipos e insumos adquiridos para el proyecto.
Modificación de Rubros	Vicerrectorado y la Unidad de Gestión de Proyectos	

1.6 PROCESOS QUE SE GESTIONAN EN LA COMIEX

Como ya se mencionó, la COMIEX gestiona varios procesos, de los cuales muchos presentan un mismo procesamiento pero con una finalidad distinta. Así podemos clasificar a los procesos por los departamentos que deberán ingresar los distintos trámites y cuál debería ser el flujo a seguir para completar el proceso.

A continuación se muestra un proceso genérico que se fundamenta en lo siguiente:

- **Proceso General**

  La COMIEX enviará el memorando para firma al Señor Vicerrector

  Sí se aprueba, el memorando regresa con la firma respectiva

  Caso contrario es devuelto para las modificaciones pertinentes

- **Disponibilidad Presupuestaria**

  La COMIEX enviará el memorando firmado más los documentos correspondientes para el tramite

  Sí se aprueba, el tramite regresa con la disponibilidad presupuestaria adjunta

  Caso contrario, regresa el trámite para las modificaciones correspondientes

- **Pago de Facturas**

  La COMIEX enviará el memorando firmado más los documentos correspondientes para el tramite

  Sí se aprueba, el tramite regresa se procesa en la dirección

  Caso contrario, regresa el trámite para las modificaciones correspondientes

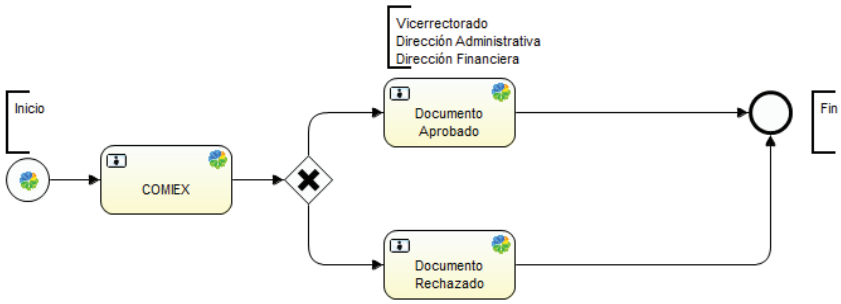


Figura 0.3. Flujo de trabajo genérico para la COMIEX

Como se puede apreciar en la Figura 2.3, el trámite es genérico y de fácil aplicación para todos los departamentos involucrados, adicionalmente, genera un control más amplio sobre cómo se gestionan los procesos y sobre cuáles son las fallas para su posterior análisis y corrección; facilitando de esta manera la optimización de recursos para todo el proceso.

1.7 ANÁLISIS DE REQUERIMIENTOS PARA EL SERVIDOR

En base al resultado obtenido del análisis de la situación actual de la COMIEX, se puede deducir los requerimientos para el servidor.

1.7.1 CAPACIDAD DE ALMACENAMIENTO

Considerando que en promedio se generan 750 hojas al mes y que se digitalizan los documentos a color principalmente se obtiene la Tabla 2.6.

Tabla 0.6. Análisis de la capacidad de almacenamiento del servidor

Número de hojas	Tamaño en MB Digitalización a blanco y negro	Tamaño en MB Digitalización a color	Total MB (BN/Color)	Total GB (BN/Color)
750/mes * 12	5	10	45000/90000	44/88

Dado que se necesita de almacenamiento 88 GB para almacenar en promedio 750 hojas digitalizadas a color anualmente y considerando una holgura del 10% se obtienen 97 GB que redondeando nos daría 100 GB anuales en promedio y considerando que se debería de tener el disco hábil por lo menos para un periodo de 5 años, la capacidad mínima requerida es de 500 GB.

1.7.2 DISPOSITIVOS DE RED

Considerando que en promedio diariamente trabajan tres personas subiendo información al repositorio y que cada persona ingresa 10 hojas con un promedio 5 MB se obtiene el total de los datos subidos que puede ser observado en el cálculo de la Figura 2.4.

$$Total\ MB\ subidos = 3\ usuarios\ x\ 10\ hojas\ x\ 5\ Mb$$

$$Total\ MB\ subidos = 150MB$$

Figura 0.4. Relación en MB de los documento escaneados

Adicionalmente, para determinar la velocidad de transmisión mínima necesaria se debe considerar que al escanear el documento este se almacenan en el

repositorio y posteriormente llenan la información requerida mínima en el sistema, el periodo de tiempo que toma en promedio esta operación es de 10 minutos, así tenemos el cálculo en la Figura 2.5.

$$Velocidad\ de\ transmisión = \frac{150MB}{10\ minutos}\ x\frac{1\ minutos}{3600\ segundos}$$

$$Velocidad\ de\ transmisión = 0.041\frac{MB}{s}$$

Figura 0.5. Velocidad de transmisión

Por lo tanto con la interfaz Ethernet propia del servidor puede solventar la velocidad mínima necesaria de 0.041 MB/s para trabajar con el repositorio.

1.7.3 CAPACIDAD DE PROCESAMIENTO

No existe una formula exacta para el cálculo de que procesador y cuál podría ser mejor para las aplicaciones que se requieran por lo que se tomara como buena práctica trabajar con un procesador que brinde las siguientes características:

- Arquitectura de 64 Bits
- Frecuencia mínima de 2 GHz
- Por lo menos 2 núcleos(core i3 o superior)

1.7.4 CAPACIDAD DE MEMORIA

Considerando que el servidor debe mantener la base de datos, el servidor web y la máquina virtual de java, se calculan los siguientes valores en base a las recomendaciones de los fabricantes:

Tabla 0.7. Análisis de requerimientos de memoria para el servidor

Servicio	Memoria (GB)
Tomcat	0.15 GB(mínimo) (doc.3tera.com, 2011)
JAVA	1 GB(mínimo) (ORACLE, 2010)
MySQL	0.2 GB(mínimo)
Otros	0.5 GB
Total	1.85 GB

Tomando en cuenta los valores de la Tabla 2.7, se determina que la memoria mínima necesaria para que funcione adecuadamente el servidor es de 2 GB en promedio estándar.

1.7.5 SERVIDOR ACTUAL DE LA COMIEX

La COMIEX ha adquirido una PC que cumple con los requerimientos antes planteados, tal y como se muestra en la Tabla 2.8.

Tabla 0.8. Características del servidor adquirido por la COMIEX

Especificación Técnica	
Memoria RAM	8 GB
Disco duro	1 TB
Tarjeta de red	10/100 Mb
Procesador	Core i7 de 2.4GHz
Periféricos	Sí

Por lo que se mantendrá el equipo para la implementación de los servicios necesarios.

1.8 ANÁLISIS DE LOS MODELOS DE GESTIÓN PARA ALFRESCO

Alfresco está desarrollado bajo el modelo visor controlador (MVC) en conjunto con Spring Surf como el framework que soporta el portal web. Como ya se mencionó, la arquitectura de Alfresco busca dar todas las facilidades a los usuarios para que este pueda ser extendido usando:

- Web Services
- Web Scripts
- CMIS

De estas tecnologías, se van a usar las dos primeras dado que abarcan casi todo el framework de desarrollo en Alfresco, pero antes de proceder con la implementación del sistema, se desarrollarán los siguientes tópicos:

1.8.1 ADQUISICIÓN DE LOS DATOS

Los datos serán digitalizados usando escáner de alta velocidad, estos posteriormente serán ingresados al repositorio a través de la interfaz web de Alfresco.

El formato por defecto para ingresar los documentos al repositorio será en formato de documento portátil (PDF) ya que estos podrán ser firmados digitalmente sin ningún problema, sin embargo, Alfresco soporta una amplia gama de formatos entre imágenes, documentos de office (Excel, Word y PowerPoint) y documentos de formato no tan común.

1.8.2 ADMINISTRACIÓN DE LA INFORMACIÓN

Los documentos serán gestionados dependiendo de los requerimientos de cada área administrativa, es así, que para la COMIEX los documentos serán ingresados en carpetas virtuales para su fácil categorización e indexación, así como también, cada documento tendrá varias propiedades de metadatos que permitirán una localización y una descripción más detallada de los contenidos buscados y almacenados.

1.8.3 CATEGORIZACIÓN DE LA INFORMACIÓN

Todos los documentos serán categorizados dependiendo de su contenido y valor, por lo que para la COMIEX se tendría lo siguiente:

- Categorización por proyectos (Semilla, internos y externos)
- Categorización por años (2010, 2011, 2012, 2013, etc.)
- Categorización por palabras claves

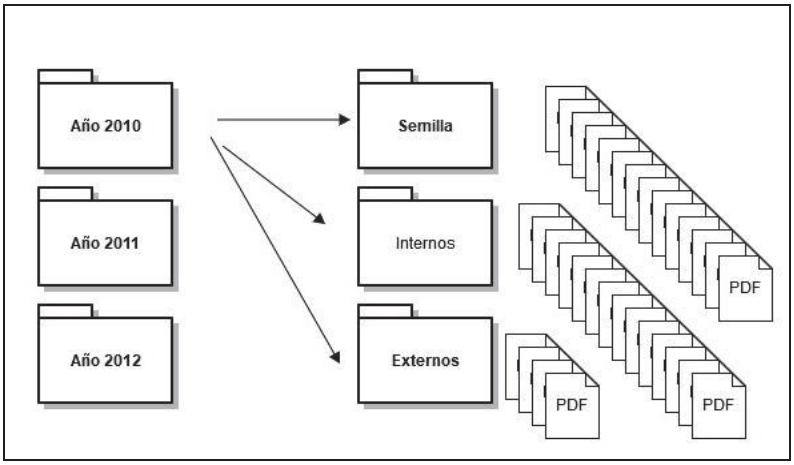


Figura 0.6. Categorización de la documentación

Para el caso de otras instancias administrativas, la categorización dependerá de cuales sean sus requerimientos tal como lo muestra la Figura 2.6.

1.8.4 ANÁLISIS Y DISEÑO DEL MODELO DE AUTENTICACIÓN Y AUTORIZACIÓN

1.8.4.1 Autenticación

Alfresco permite autenticarnos contra el sistema haciendo uso de varias herramientas comunes como son:

- CIFS[12] (Ietf, 1997)
- Web Client
- FTP
- WebDAV[13] (Webdav, 2009)
- Web Services
- Spring Beans exposed as public services in Java(Clase Java)

Dependiendo de la manera con la que se desee autenticar se podría hacer un SSO (Single Sign On) con el servicio de Active Directory o LDAP o de algún cliente web como Alfresco Share o Explorer según se muestra en la Figura 2.7.

Considerando que la COMIEX no presenta un servicio de Active Directory o LDAP[14] del cual se pueda hacer uso, se usará la autenticación vía cliente web.

---

[12] CIFS (Common Internet File System) es un estándar de compartición de ficheros de Microsoft Windows.

[13] WebDAV (*Web Distributed Authoring and Versioning*) es una extensión del HTTP que permite gestionar operaciones de autorización de contenidos a través de la web.

[14] LDAP(*Lightweight Directory Access Protocol*) es un protocolo que permite gestionar un directorio activo en plataformas Linux y Windows similar al Active Directory.

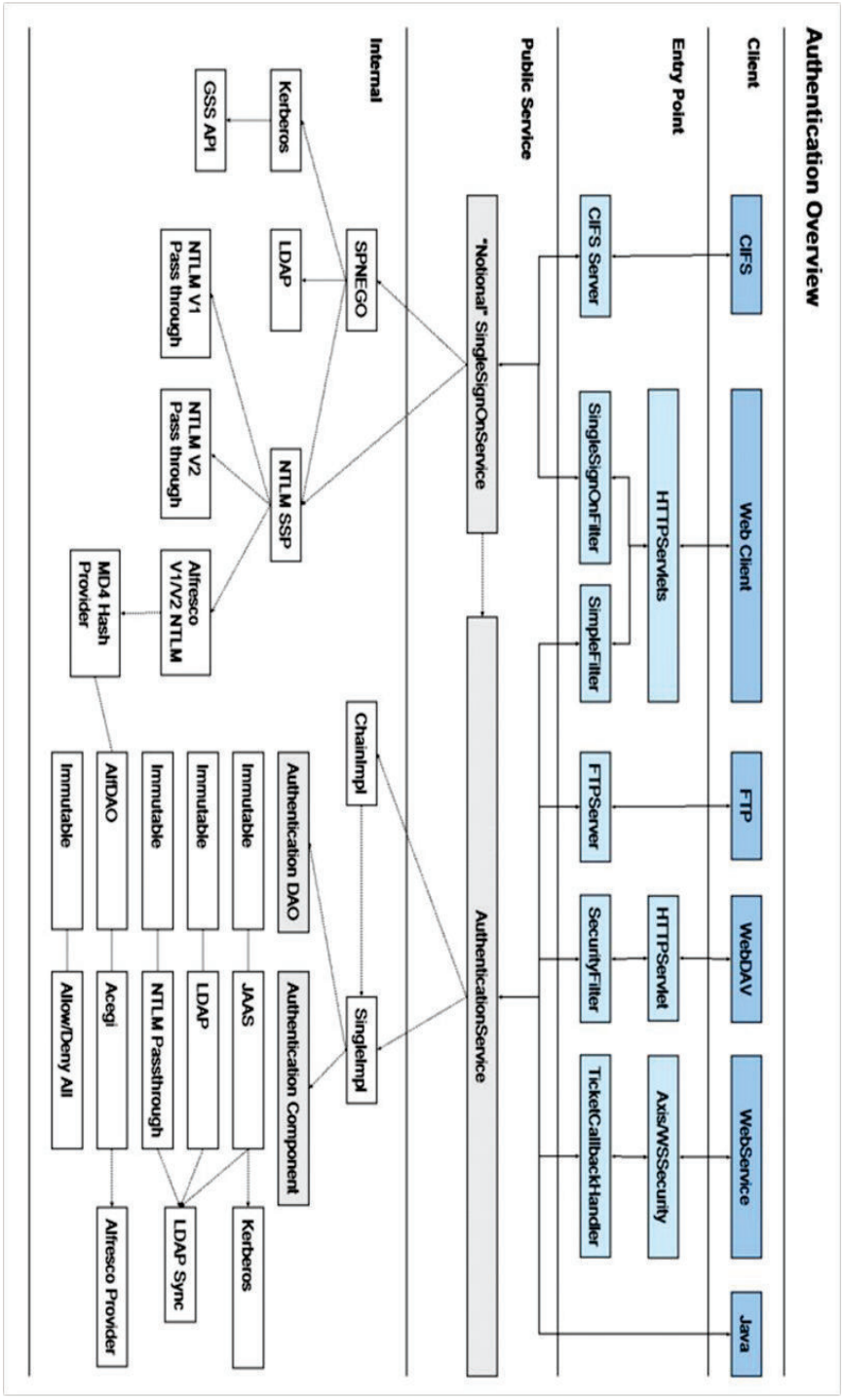


Figura 0.7. Arquitectura de autenticación de Alfresco. Fuente: (Alfresco, s.f.)

1.8.4.2 Autorización

Para el proceso de autorización, Alfresco tiene predefinido 5 roles que se muestran en la Tabla 2.9, los cuales se adaptan a los requerimientos de la mayoría de departamentos y personas, adicionalmente, Alfresco permite la creación de grupos de trabajo, por lo que la asignación de roles puede ser a nivel de grupo, subgrupo o usuario dentro del sistema.

Tabla 0.9. Roles predefinidos en Alfresco. Fuente: (Alfresco, s.f.)

Rol	Lectura	Edición	Agregación	Eliminación
Consumidor	✓			
Editor	✓	✓		
Contribuidor	✓		✓	
Colaborador	✓	✓	✓	
Coordinador	✓	✓	✓	✓

De estos roles, la COMIEX hará uso de tres roles: **Consumidor**, **Colaborador** y **Coordinador**, dado que se ajusta perfectamente a los perfiles requeridos por el área administrativa.

1.8.4.3 Identificación de los usuarios

Considerando que la COMIEX coordina esfuerzos con varios departamentos para la realización de cualquier petición, se enumerará los actores que intervendrán en el proceso por cada departamento.

Tabla 0.10. Perfiles de usuarios para Alfresco.

Nombre del espacio	Grupo	Roles
Comisión de Investigación y Extensión	Administradores Colaboradores Coordinadores Secretarias Autoridades	Administrador Colaborador Coordinador Consumidor Consumidor
Dirección de Recursos Humanos	Administradores Secretarias Autoridades	Administrador Consumidor Consumidor
Planificación	Autoridades	Consumidor
Rectorado	Colaboradores	Colaborador

En la Tabla 2.10 se muestran los roles planteados que podrían ser utilizados por las diferentes áreas.

Teniendo en cuenta que se debe de trabajar también con la Dirección Financiera con el sistema, la unidad ha expresado que al momento todos sus trámites deben ser realizados con documentos físicos, por lo que no se los ha agregado al sistema como tal.

1.8.5 ANÁLISIS Y DISEÑO DE MODELOS DE CONTENIDO

1.8.5.1 Modelo de contenido (Alfresco, 2014)

Un modelo de contenido es la representación de un prototipo de objeto que

presenta varios atributos, estos atributos pueden ser guardados en un archivo XML. El objetivo de estos documentos es ayudar a describir de una mejor manera el documento que posee estos atributos y de esta manera se facilite el proceso de indexación para cuando se realicen búsquedas.

1.8.5.1.1 Tipos de datos para representaciones de contenido

Al igual que en cualquier lenguaje de programación, los tipos de datos permitidos son los mismos, en la Tabla 2.11 se muestra los tipos de datos permitidos para la representación en XML de las distintas propiedades.

Tabla 0.11. Tipos de datos. Fuente: (Alfresco, s.f.)

Tipo	Descripción	Equivalente JAVA
text		java.lang.String
content	Content Descriptor (includes content store URL, MIME type, etc.)	org.alfresco.service.cmr.repository.ContentData
int		java.lang.Integer
long		java.lang.Long
float		java.lang.Float
double		java.lang.Double
date		java.util.Date
datetime		java.util.Date

boolean		java.lang.Boolean
qname	Qualified Name	org.alfresco.service.namespace.QName
category	Reference to a category within a classification	java.lang.String
noderef	Node Reference	org.alfresco.service.cmr.repository.NodeRef
period	Period/duration/cron/...	org.alfresco.service.cmr.repository.Period
path	Path	org.alfresco.service.cmr.repository.Path
any	Any of the above	java.lang.Object

1.8.5.1.2 Asociaciones

Las asociaciones representan documentos adjuntos dentro de algún registro.

De acuerdo a los documentos que se manejan en la COMIEX y considerando que estos tienen a la vez documentos adjuntos, tenemos los siguientes modelos de contenido:

1.8.5.2 Modelo de contenido para un memorando

Este prototipo describirá todos los documentos relacionados con peticiones y requerimientos como por ejemplo:

- Memorandos enviados y recibidos
- Certificaciones presupuestarias (Memorando recibido)

En la Tabla 2.12 se muestra las propiedades que se utilizaran para describir el documento gestionado:

Tabla 0.12. Modelo de contenido para un memorando.

Propiedad	Tipo de dato	Observaciones
Nombre	String	Código del memorando (CP – XXX - AÑO)
Asunto	String	Tipo de memorando
Descripción	String	Descripción del trámite
Datos de envío		
De	String	Nombre de la persona que emite el memorando
Para	String	Nombre de la persona que recibe el memorando
Fecha de envió	Datetime	Fecha de envío
Fecha de recepción	Datetime	Fecha de recepción
Datos de recepción		
Sumillador	Person	Persona que recibe el documento
Con copia a	Person	Persona a la que se informa adicionalmente del trámite en cuestión

Documentación adjunta		
Anexos	Association	
Palabras clave		
Etiquetas	Category	Palabras claves

1.8.5.3 Presentación de proyectos

Este modelo representa al documento que se entrega al momento de convocar a la presentación de algún proyecto de investigación:

- Semilla
- Interno
- Externo

La Figura 2.13 detalla los parámetros necesarios para la representación del documento.

Tabla 0.13. Modelo de contenido para la presentación de proyectos de investigación

Propiedad	Tipo de dato	Observaciones
Nombre	String	Código del proyecto (PIS – 13 – 01)
Titulo	String	Nombre del proyecto
Descripción	String	Resumen del proyecto
Datos generales		

Área del proyecto	String	Área del proyecto (Control de procesos)
Línea de Investigación	String	Línea de investigación (Control de procesos)
Departamento	Category	Conjunto de departamentos categorizados (facultad, departamento)
Tipo de proyecto	Category	Conjunto de proyectos por código y tipo (semilla – PIS – 13 - 01)
Entidad Auspiciante	Category	Conjunto de categorías de posibles entidades auspiciantes (EPN, FLACSO, etc.)
Aplica a	List	ComboBox que contendrá: Ing. y tecnología Ciencias médicas Ciencias agrícolas Ciencias sociales Humanidades
Fecha de Presentación	Date	Fecha de presentación del proyecto
Fecha de aprobación	Date	Fecha de aprobación del proyecto
Fecha de inicio	Date	Fecha de inicio del proyecto
Fecha de prorroga	Date	Fecha de aplazamiento de entrega del proyecto
Fecha de finalización	Date	Fecha de entrega del proyecto(informe

		final, presentación y artículo)
Datos del director del proyecto		
Nombres	String	Datos generales para el director del proyecto
Apellidos	String	
Tel. casa	String	
Tel. oficina	String	
Celular	String	
Extensión EPN	String	
Email	String	
Primero, segundo y tercer colaborador		
Tipo de colaborador	List	Puede ser: Profesor Profesor visitante Pasante
Nombres	String	Datos generales para los colaboradores del proyecto
Apellidos	String	
Tel. casa	String	
Tel. oficina	String	
Celular	String	
Extensión EPN	String	

Email	String	
Datos presupuestarios		
Convenios de becas	Double	Datos financieros estimados al momento de la presentación del proyecto
Equipos	Double	
Reactivos y materiales de laboratorio	Double	
Literatura especializada	Double	
Viajes técnicos y de muestreo	Double	
Ponencias en congresos internacionales	Double	
Monto total	Double	
Otros datos		
Observaciones	String	Observaciones adicionales
Documentación adjunta		
Informes	Association	Informes de presentación o informe final
Memorandos	Association	Memorandos generales
Palabras claves		
Etiquetas	Category	Palabras claves

1.8.5.4 Informes

Este modelo representa a los informes que son presentados al momento de arrancar con la investigación, estos informes son generados dependiendo del tipo de investigación:

- Informes de avance – 2 informes anuales
- Informe final – 1 informe al finalizar la investigación

En la Tabla 2.14 se muestra las propiedades que deberán ingresar para poder dar seguimiento a la investigación.

Tabla 0.14. Modelo de contenido para la presentación de informes de avance y final de la investigación

Propiedad	Tipo de dato	Observaciones
Nombre	String	Código del informe
Titulo	String	Título del proyecto
Descripción	String	Resumen del proyecto
Datos generales		
Tipo de informe	List	Puede ser de tipo: Avance Final
Periodo	List	Puede ser: Enero – Junio (A) Julio – Diciembre (B)

Objetivo General	String	Objetivo general
Objetivos específicos(5 objetivos)		
Objetivo específico	String	Objetivos y resultados específicos
Resultados Obtenidos	String	
Actividades desarrolladas / mes (Periodo)		
Mes 1,2,3,4,5,6,7	String	
Publicaciones desarrolladas		
Nacionales	String	Listado de publicaciones desarrolladas durante toda la investigación
Internacionales	String	
Indexadas	String	
Memorias	String	
Libros	String	
Artículos	String	
Otros	String	
Tesis desarrolladas (tres tesis)		
Tesis	String	Datos generales relacionadas con las tesis desarrolladas durante la investigación
Estado	List	
Otros datos		

Conclusiones generales/ parciales	String	Conclusiones generales / parciales de la investigación
Recomendaciones generales / parciales	String	Recomendaciones generales / parciales de la investigación
Observaciones generales	String	Observaciones adicionales
Porcentaje de avance(1-10)	int	Porcentaje de desarrollo de la investigación
Documentación adjunta		
Anexos	Association	Documentos adjuntos con el informe
Palabras claves		
Etiquetas	Etiquetas	Etiquetas

Como se puede apreciar, los modelos de contenido son una herramienta básica del repositorio de Alfresco y por lo tanto, es necesario realizar un buen análisis y diseño para que se pueda representar de mejor manera los trámites y procesos en la realidad.

1.9 DISEÑO DEL SISTEMA WEB

1.9.1 QUE ES UNA ARQUITECTURA

Es un conjunto de patrones que permite definir un marco funcional sobre el cual

se desarrollara el software y que adicionalmente garantizara otros factores como son: el mantenimiento, la auditabilidad, el testeo y la interacción con otros sistemas informáticos.

1.9.2 QUE ES UN MODELO

Un modelo permite una representación abstracta de un determinado problema, así como también, la interacción que puede presentarse cuando se lo expone a diferentes factores externos como internos.

Un modelo permite diseñar de mejor manera un sistema de n capas.

1.9.3 MODELO CLIENTE – SERVIDOR

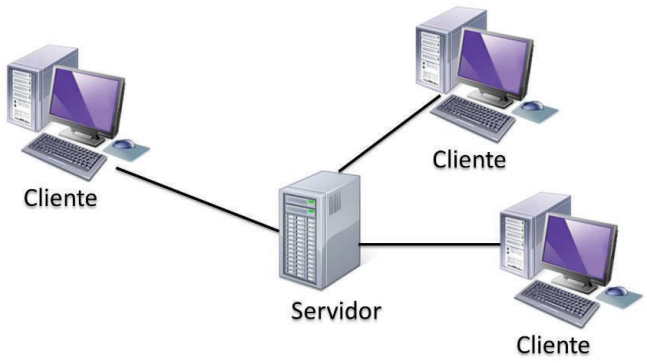


Figura 0.8. Modelo cliente - servidor

Este modelo define la interacción que se expone entre un maquina servidor (entorno remoto) y un cliente (entorno local). Por ejemplo: Figura 2.8.

1.9.3.1 Ventajas

- Toda la información se almacena y gestiona dentro del servidor remoto.
- Facilidad de manejo para el usuario final

- Garantiza interoperabilidad entre clientes
- Brinda cierto grado de escalabilidad al sistema informático desarrollado

1.9.3.2 Desventajas

- Al existir un gran número de usuarios concurrentes puede generarse una saturación al servidor lo que causaría la caída del servicio.
- Si el servicio dentro del servidor esta caído ningún cliente podrá interactuar con el servidor.

Este modelo ha dado paso a la construcción de diferentes sistemas multiplataforma que garantizan la interoperabilidad entre servicios como es el caso de arquitectura SOA[15] (Microsoft, s.f.) en donde se comparten servicios a través de un bus de datos y estos pueden ser accedidos por cualquier cliente para el cual haya sido desarrollado el sistema como por ejemplo: dispositivos móviles, computadoras de escritorio, tablet, etc.

1.9.4 ARQUITECTURA DE N CAPAS

El modelo de n capas es una arquitectura que ha predominado el diseño de sistemas multiplataforma y cuyo objetivo es separar los distintos entornos por lo que se puede desarrollar un sistema informático.

Es así que un sistema puede ser dividido en tres capas. Ver Figura 2.9:

- Capa de datos
- Capa de lógica del negocio

---

[15] SOA(*Service Oriented Architecture*) es una arquitectura que permite interoperación entre sistemas con diferentes arquitecturas.

- Capa de presentación.

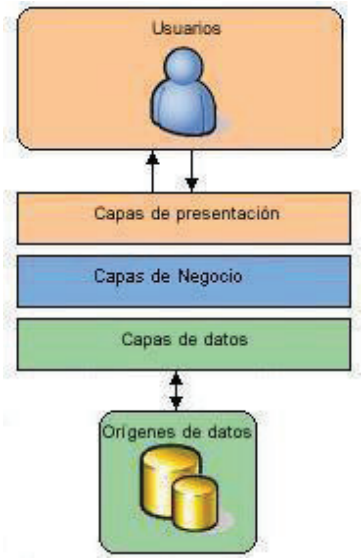


Figura 0.9. Arquitectura de N capas Fuente:(http://juanlaracarrion.blogspot.com)

Cada una de estas capas puede verse inmersas en n capas intermedias que ayuden a satisfacer los requerimientos iniciales. Véase la Figura 2.10.

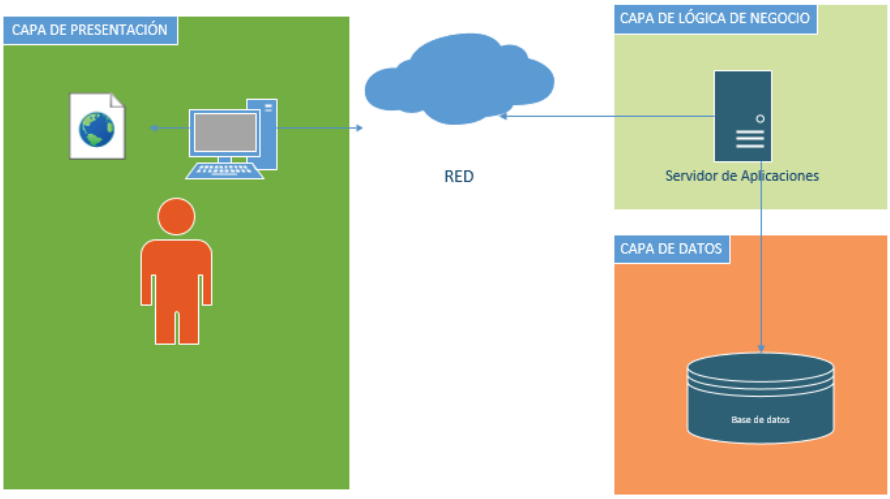


Figura 0.10. Arquitectura de N capas

1.9.4.1 Capa de datos

Esta capa se encarga de gestionar la comunicación entre la base de datos y el sistema como tal. Adicionalmente gestiona las transacciones y operaciones SQL.

1.9.4.2 Capa de negocios

Esta capa se encarga de gestionar la lógica de clases, excepciones y restricciones que definen al modelo de negocio.

1.9.4.3 Capa de presentación

Esta capa define los formularios que permitirán al usuario final interactuar con la aplicación desarrollada.

1.9.5 METODOLOGÍA UWE

1.9.5.1 ¿Qué es la metodología UWE y que permite?

UML Based Web Enginnering – UWE, es una metodología de ingeniería de software enfocado en el uso de UML (Ietf, 2003) para el desarrollo de arquitecturas web, cuyo principal objetivo es la realización de modelos que permitan trabajar con el lenguaje orientado a objetos.

1.9.5.2 Fases de metodología UWE

Como muestra la Figura 2.11, el modelo UWE se encuentra fundamentalmente integrada por estas cuatro etapas, sin embargo existen etapas adicionales como las que se presentan a continuación:

- **Planificación**: Se presenta la captura, análisis y especificaciones de requisitos para la aplicación.

- **Diseño:** Se basa fundamentalmente en el análisis de los requisitos generados en la fase de planificación, aquí también se analizan y generan los diagramas UML respectivos para una correcta codificación del sistema.

- **Codificación:** Consiste en pasar el diseño a código incluyendo la generación de interfaces amigables con el usuario.

- **Pruebas:** Se utilizan para corregir posibles bugs dentro de la aplicación desarrollada.

- **Instalación:** Es el proceso mediante el cual el sistema desarrollado es transferido a una maquina destino o host en el cual los usuarios podrán hacer uso del sistema.

- **Mantenimiento**: Es el proceso de control, mejora y optimización del sistema generado.

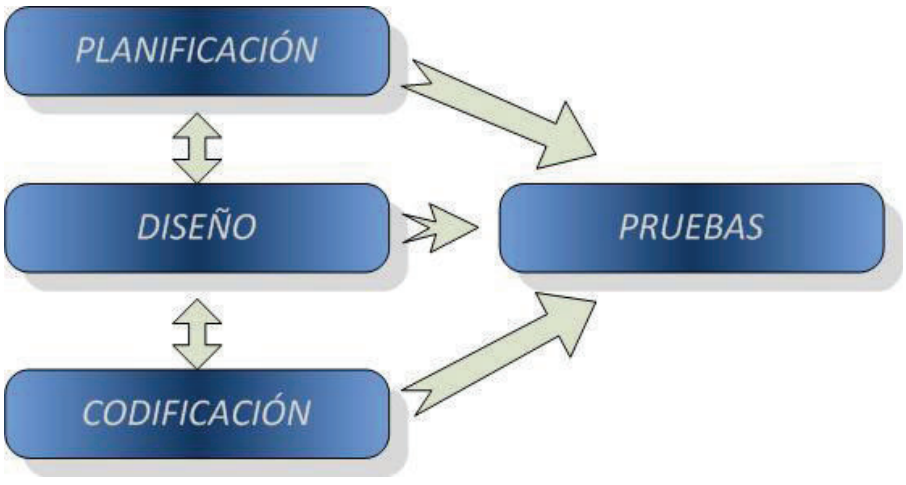


Figura 0.11. Fases del Desarrollo UWE, fuente: (**Galeano, 2013)**

Cabe destacar que las etapas de codificación, pruebas, instalación y mantenimiento del sistema, se analizarán en la implementación de la aplicación en el capítulo siguiente.

1.9.5.3 Planificación: Análisis de requerimientos para la COMIEX

La implementación de la aplicación web implica que se maneje la estructura mostrada en la Figura 2.12:

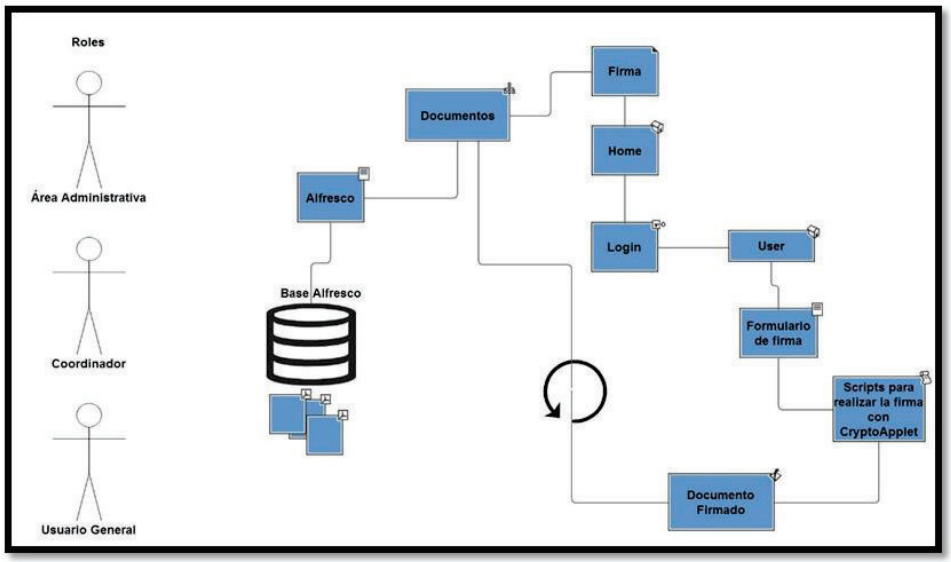


Figura 0.12. Estructura básica del funcionamiento de la aplicación

Para la aplicación del modelo UWE en el sistema que se va a desarrollar para la COMIEX presenta los siguientes requisitos:

- **Requerimientos de los roles para los usuarios**

Cada una de las acciones que pueden desarrollar los usuarios se detalla en la Tabla 2.15 y Tabla 2.16:

Tabla 0.15. Roles de los usuarios para la aplicación web

Requerimientos para los Usuarios	
Rol	**Acciones**
Área Administrativa	Creación y generación de reportes Firma digital de documentos (solo memorandos) Verificación de la validez de la firma digital (solo memorandos) Gestión de proyectos de investigadores
Coordinador	Gestión de proyectos de investigadores Visualización de Reportes Firma digital (solo memorandos) Verificación de la Firma digital (solo memorandos)
Usuario General(Investigador)	Visualización de reportes(estados del proyecto)
Nota: *Estos requerimientos deberán ser ejecutados a través del sistema a desarrollar para la COMIEX*	

Tabla 0.16. Requerimientos generales para los usuarios

Requerimientos Generales	
Rol	Acciones
Todos los roles	Edición de su perfil dentro del sistema. Ingreso y eliminación de documentos.

	Capacidad de aplicación de modelos de contenido para la gestión de la información de metadatos en los documentos. Actualización de modelos de contenido.
Nota*:* Estos requerimientos deberán ser ejecutados a través de la interfaz gráfica del CMS Alfresco	

A continuación se presenta el proceso a seguir para el usuario que disponga de cualquiera de los tres roles que se presentaron:

- El usuario accederá a la plataforma web de Alfresco, se autenticará e ingresará al sistema
- En este punto puede subir documentos, eliminarlos, actualizarlos, dependiendo de los permisos de los que disponga su cuenta
- Para el proceso de la firma digital se ingresará a la plataforma web de firma digital, con los datos de su cuenta
- El proceso de autenticación se llevará a cabo con el user y password que maneja Alfresco para proporcionar acceso a su plataforma web
- Una vez autenticado, el usuario será re-direccionado a la página del usuario(dependiendo del rol), en esta página se establecerán los siguientes parámetros:
    - **Ingreso de los documentos que se desea firmar**: Este documento será seleccionado de una carpeta que se desee para el almacenaje de documentos recién cargados.
    - **Ingreso de la carpeta de salida**: Ubicación de la carpeta donde se guardará él o los documentos firmados.
    - Asunto
    - Ubicación

- Otros.

- Posteriormente se procederá a enviar los documentos a firmar al gestor de firmas digitales, se seleccionará el certificado a usar, se firmará el documento y por último se los enviará al repositorio de Alfresco.

Cabe destacar que la aplicación se encargará de gestionar la firma digital, para esto se usarán procedimientos remotos como son los Web Services y los WebScripts que proporciona Alfresco, dichos servicios permiten la manipulación de todos los servicios que Alfresco brinda al usuario final.

1.9.5.4 Diseño

En esta parte de aplicarán algunos de los modelos UML[16] que define la metodología UWE, dichos modelos permitirán realizar diagramas para el funcionamiento de la aplicación, definiendo:

- Actores
- Relaciones entre actores
- Casos de uso para cada actor
- Modelos de contenido
- Modelos de navegación
- Modelos de presentación

1.9.5.4.1 Actores

En la aplicación web existirán 3 tipos de actores:

---

[16] UML(*Unified Modeling Language*) es un lenguaje de modelado enfocado en el diseño de sistemas informáticos.

- Usuario administrador
- Usuario coordinador
- Usuario general o usuario investigador

Cada uno de estos usuarios presentados en la Figura 2.13 contará con los roles citados en la Tabla 2.15

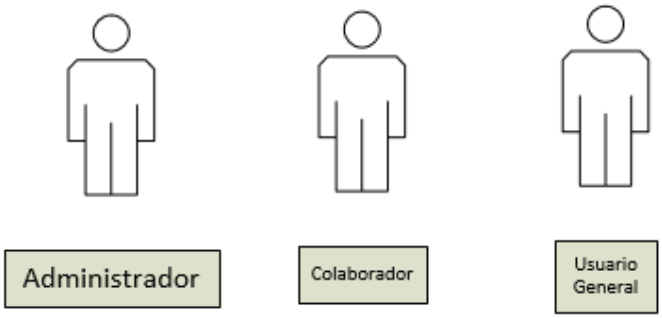


Figura 0.13. Caso de uso: Actores

1.9.5.4.2 Relaciones entre actores

Las relaciones se las presentan modeladas de la siguiente manera:

- El usuario administrador podrá realizar todas las actividades que realiza el usuario general y el usuario coordinador.
- El usuario administrador realiza las siguientes actividades:
    - Genera reportes
    - Crea reportes
    - Gestiona proyectos
    - Firma digital

- El usuario coordinador realiza las siguientes actividades:
    - Genera reportes
    - Firma digital
- El usuario general realiza las siguientes actividades:
    - Genera reportes

En la Figura 2.14 se representa visualmente la relación entre los roles de usuario.

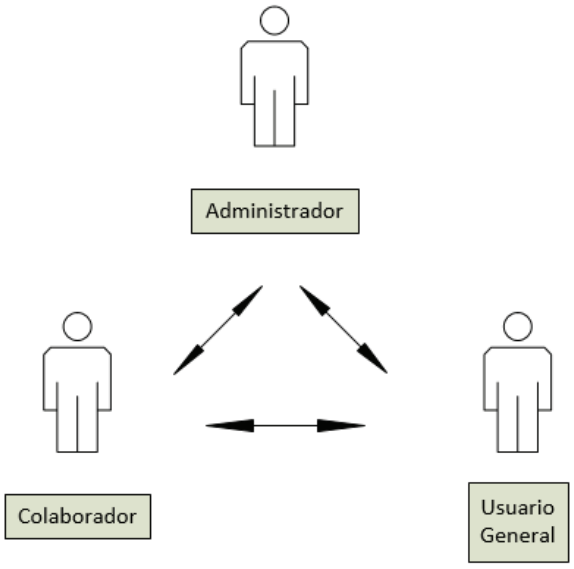


Figura 0.14. Relación entre actores

1.9.5.4.3 Modelos de Casos de Uso y de paquetes

Este modelo define el comportamiento que se tendrá ante diversas situaciones que se presenten en el sistema, pretende modelar la interacción del usuario ante el sistema para la ejecución de una determinada tarea.

Elementos Básicos del modelo:

- **Actores:** Representan el rol que desempeña el usuario ante el sistema.

- **Casos de uso:** Son las tareas que pueden llevarse a cabo.
- **Asociaciones:** Representan la interacción entre el actor y la tarea.

En las Figuras 2.15, 2.16, 2.17, 2.18 se presenta la aplicación de los modelos.

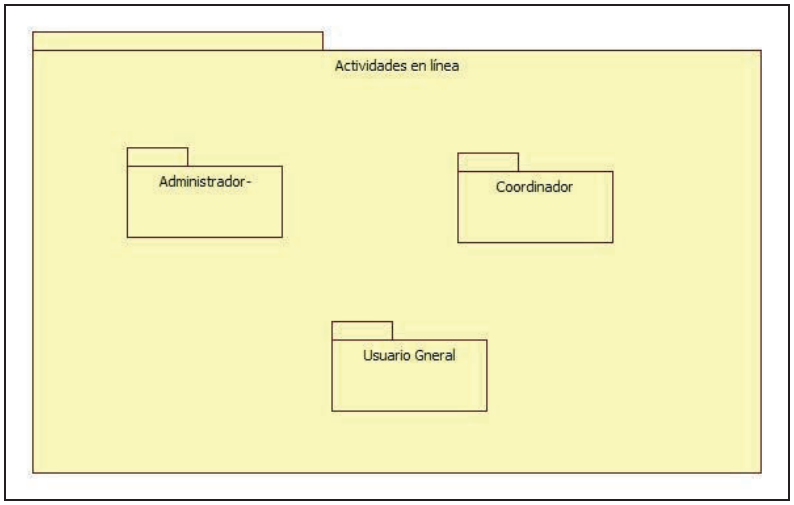


Figura 0.15. Modelo de paquetes

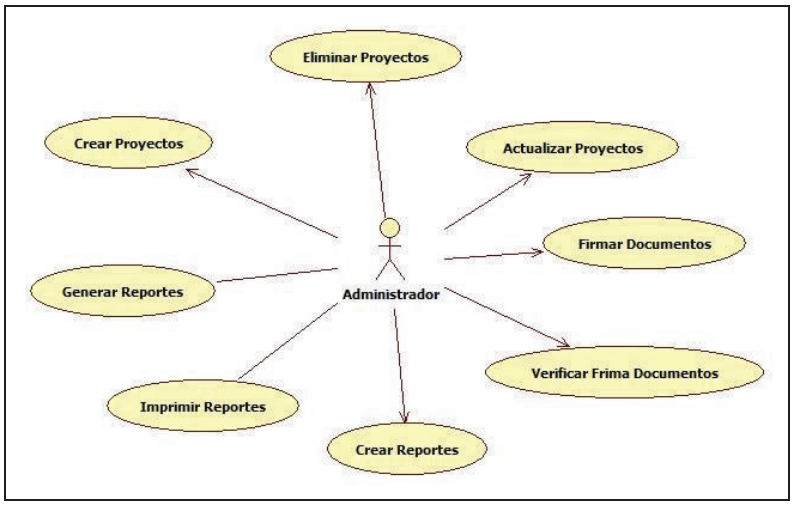


Figura 0.16. Modelos de casos de uso para el administrador

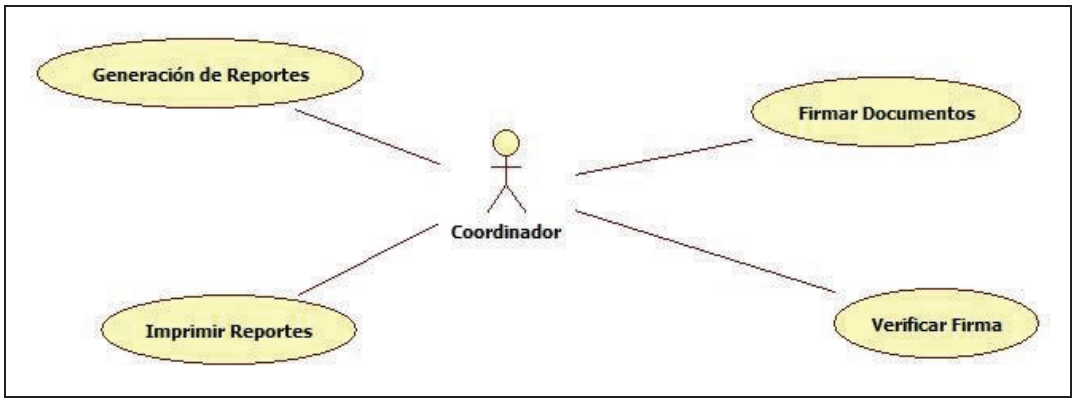


Figura 0.17. Modelo de caso de uso para el coordinador

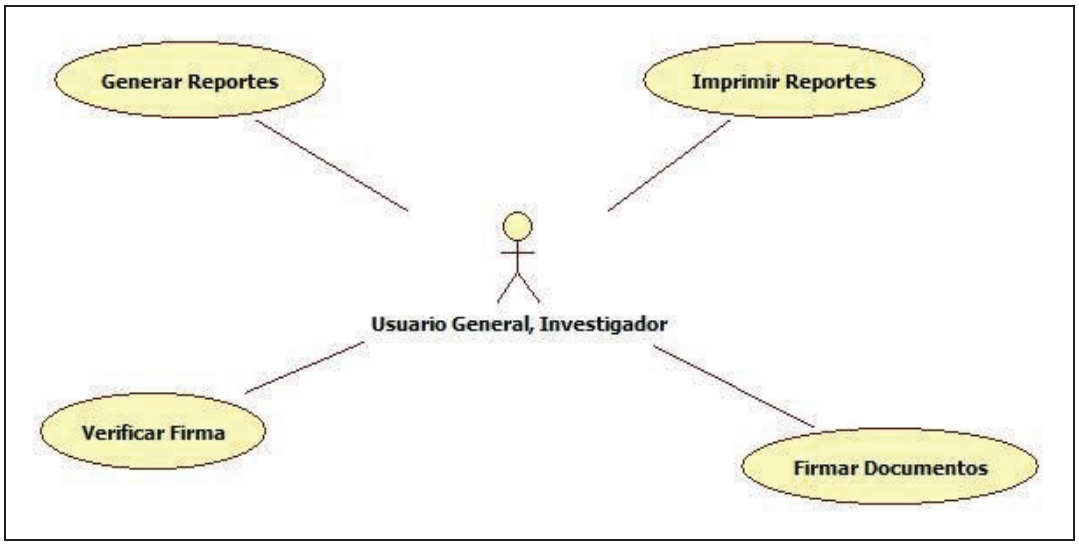


Figura 0.18. Modelo de caso de uso para el usuario general

1.9.5.4.4 Modelos de Contenido

Para este modelo se debe considerar que los datos se obtendrán en su mayoría de la base de Alfresco mediante servicios web, por lo que no se requerirá la generación de una base de datos especializada para la aplicación. Sin embargo, se debe gestionar la parte financiera de los proyectos, así como también, la firma digital por lo que se realizarán las clases pertinentes para este cometido. Véase

las Figuras 2.19 y 2.20.

Figura 0.19. Modelo de Contenido para la verificación de la firma digital

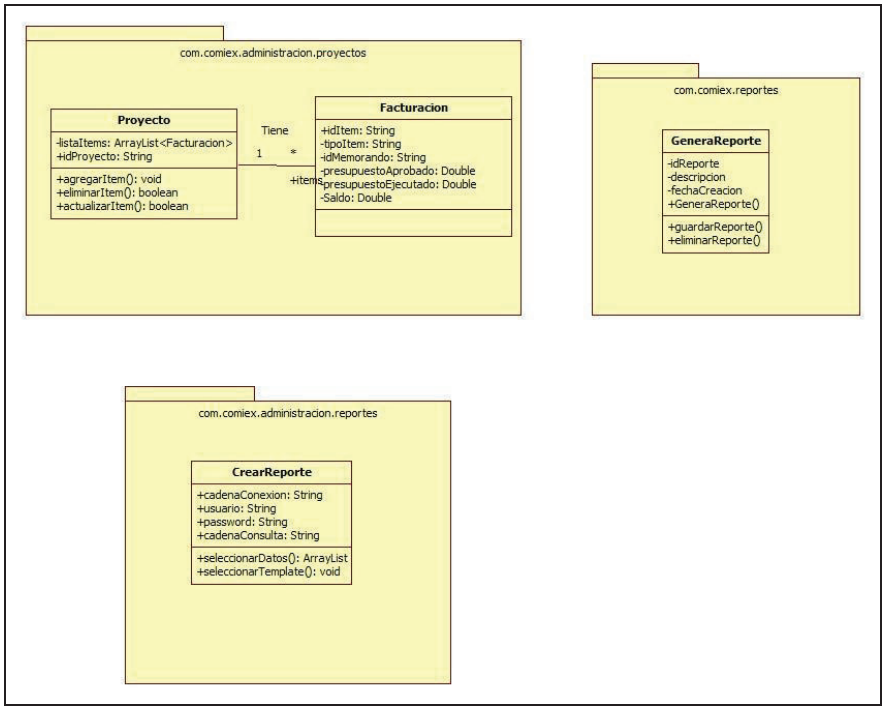


Figura 0.20. Modelo de Contenido para la firma digital y los reportes

1.9.5.4.5 Modelo de navegación

Este modelo se encarga de manejar la interactividad del usuario con el sitio web. En la Figura 2.21, se muestra el flujo que se presenta en el sitio web.

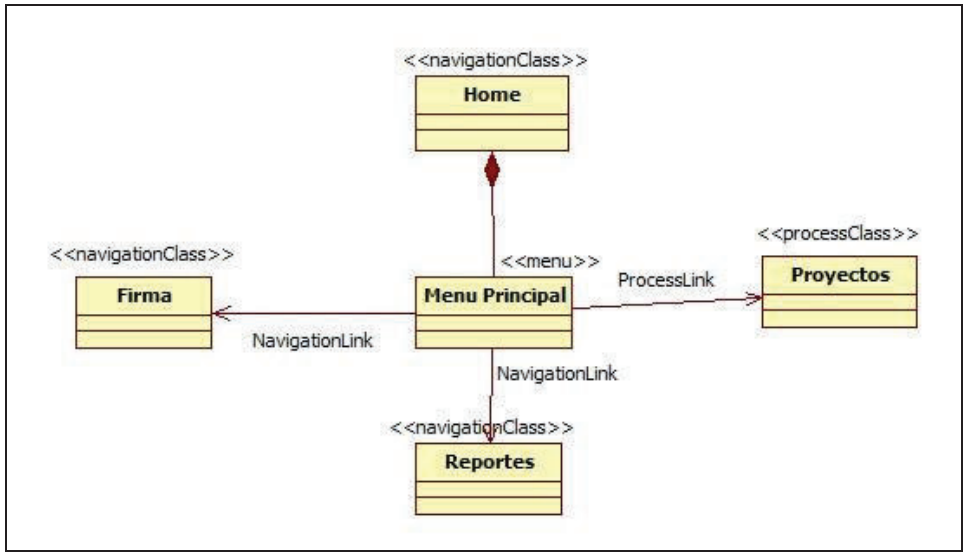


Figura 0.21. Modelo de Navegación del menú genérico

1.9.5.4.6 Modelo de presentación

Este modelo diagrama las vistas a ser presentadas al usuario a través de un navegador web. Adicionalmente, se debe especificar todos los elementos correspondientes a ser usados en el sistema como son:

- Botones
- Diálogos
- Etiquetas(label)
- Etc.

Para la modelación de estás vistas se ha usado el programa Pencil (Pencil, s.f.),

el cual nos permite fácilmente diagramar las distintas ventanas que requiramos. En la Figuras 2.22 y la Figura 2.23 permite apreciar el resultado de usar Pencil en nuestras vistas.

Figura 0.22. Vista de login para todos los usuarios

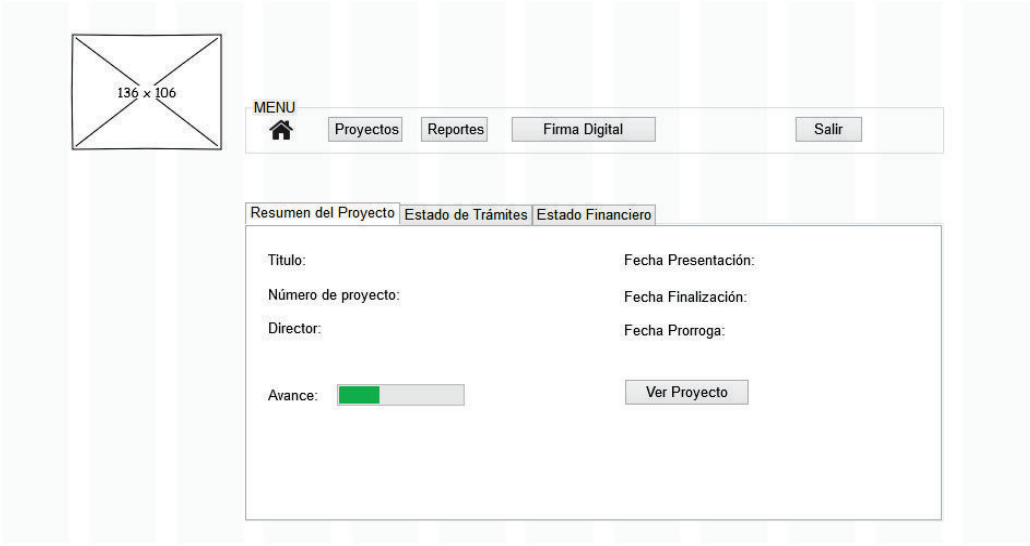


Figura 0.23. Vista de inicio para el usuario general / investigador (ANEXO D)

1.10 DISEÑO DE REPORTES

Los reportes que se gestionaran en la COMIEX se diseñarán usando el sistema JasperReport, el mismo que nos permitirá muy fácilmente generar los reportes en varios formatos de acuerdo a los requerimientos del departamento.

1.10.1 JASPERREPORT

JasperReport es una suite para la generación de reportes avanzados, construido sobre Java y que puede ser embebida en cualquier aplicación que web.

1.10.1.1 Características técnicas

- Permite exportar documentos en formato PDF, HTML, XLS, CSV y XML
- Es un software OpenSource
- Puede ser embebido en aplicaciones Java, J2EE[17] y aplicaciones web.
- Presenta un servidor dedicado para gestionar los reportes en caso de requerirlo.
- Trabaja en conjunto con iReport Designer para generar los reportes usando una interfaz muy intuitiva.
- Puede trabajar en conjunto con eclipse.
- Presenta soporte para cualquier base de datos
- Es independiente de la plataforma.
- Puede implementarse usando cualquier contenedor web como apache Tomcat, web logic, etc.

---

[17] J2EE(*Java Enterprise Edition*) es un estándar de Oracle que define la arquitectura para la generación de software empresarial.

En la Tabla 2.17 se muestra un resumen de las especificaciones técnicas de JasperReport

Tabla 0.17. Especificación Técnica de JasperReport, Fuente: (JasperReport, s.f.)

Resumen de Especificaciones Técnicas	
Reference Operating Systems:	Windows Vista/XP/7/2003/2008*, Red Hat Enterprise Linux (RHEL)*, Novell SUSE Linux*, Sun Solaris, OpenSolaris*, Fedora Enterprise Linux, Apple Mac OS X, Debian Linux, HP-UX, FreeBSD, IBM AIX, CentOS, Ubuntu. Note: Jaspersoft is 100% pure Java, so should work in any Java 1.5 or 1.6 environment.
Application servers:	Apache Tomcat*, JBoss, IBM WebSphere, Oracle WebLogic, Sun GlassFish, SpringSource tc Server
Data Sources:	Any RDBMS compliant with JDBC 2.1 and SQL-92 (includes most relational databases). Other data sources: Bean, JNDI, JasperAnalysis (Mondrian), XML/A, Custom (e.g. Hibernate, XML, etc.)
Web Browsers:	Microsoft Internet Explorer, Mozilla Firefox, Apple Safari, Google Chrome
Portal servers:	Liferay

1.10.1.2 Beneficios de usar JasperReport

- Capacidad de diseño de reportes de una manera gráfica e intuitiva

- Arquitectura escalable
- Puede ser implementado fácilmente
- Existe una buena documentación
- Existe una versión libre y una pagada.
- Puede ser embebida en las aplicaciones mediante JasperReport Server.

1.10.1.3 Diseño de reportes usando JasperReport

Para el desarrollo del diseño de los reportes, se considerará primero cuales son los procesos principales que se requieren en la COMIEX.

1.10.1.3.1 Estado de los trámites

Al gestionarse los trámites en conjunto con Workflows, Alfresco permite usar WebScripts para retornar mediante consulta el estado de los procesos. Para lo cual la plantilla a usar se muestra en la Figura 2.24.

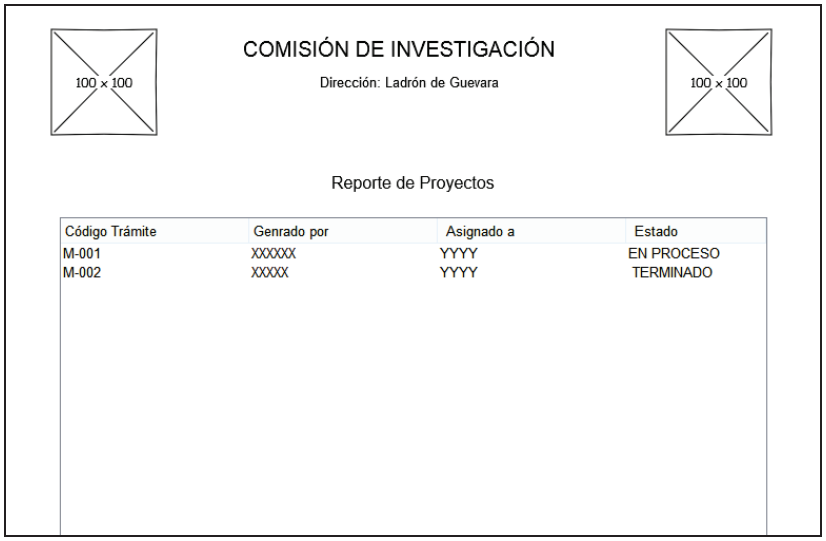


Figura 0.24. Plantilla de reporte de tareas

1.10.1.3.2 Resumen financiero por proyecto

El objetivo de la plantilla diagramada en la Figura 2.25 es que cada investigador acceda al reporte correspondiente al proyecto que ejecuta.

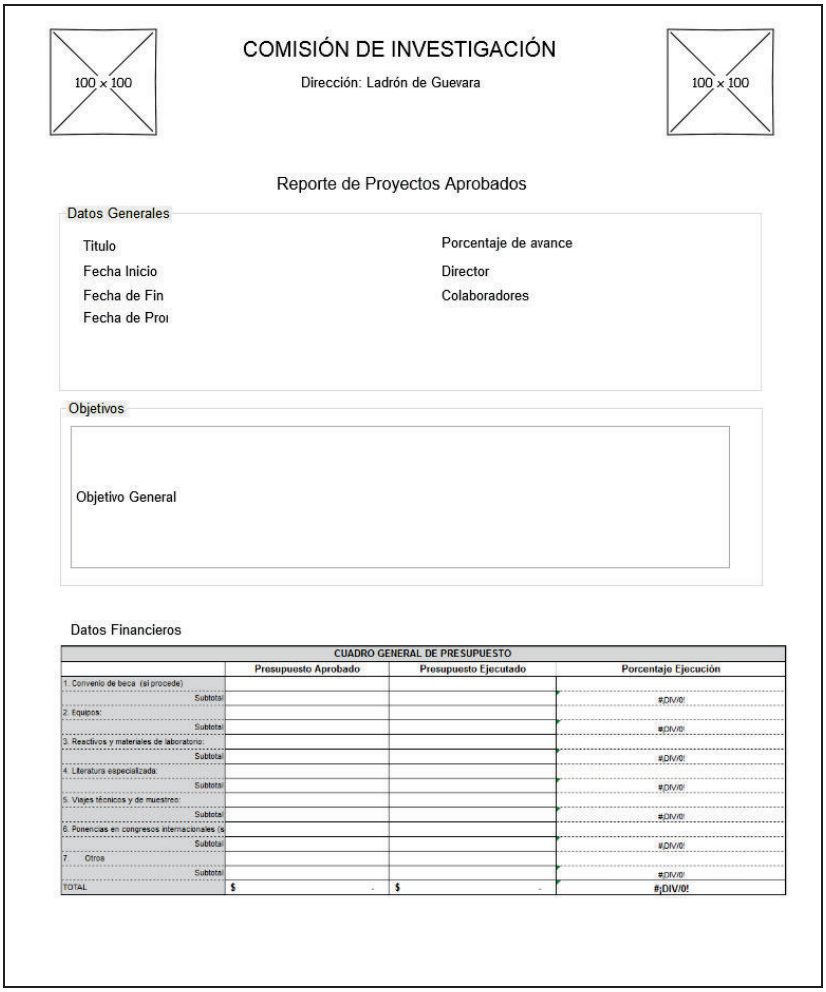


Figura 0.25. Reporte de estado financiero del proyecto

1.10.1.3.3 Informe final del proyecto

Este reporte mostrado en la Figura 2.26 tiene como principal función dar a conocer cuál fue el porcentaje de cumplimiento de los objetivos, el estado financiero y los resultados obtenidos de la ejecución del mismo.

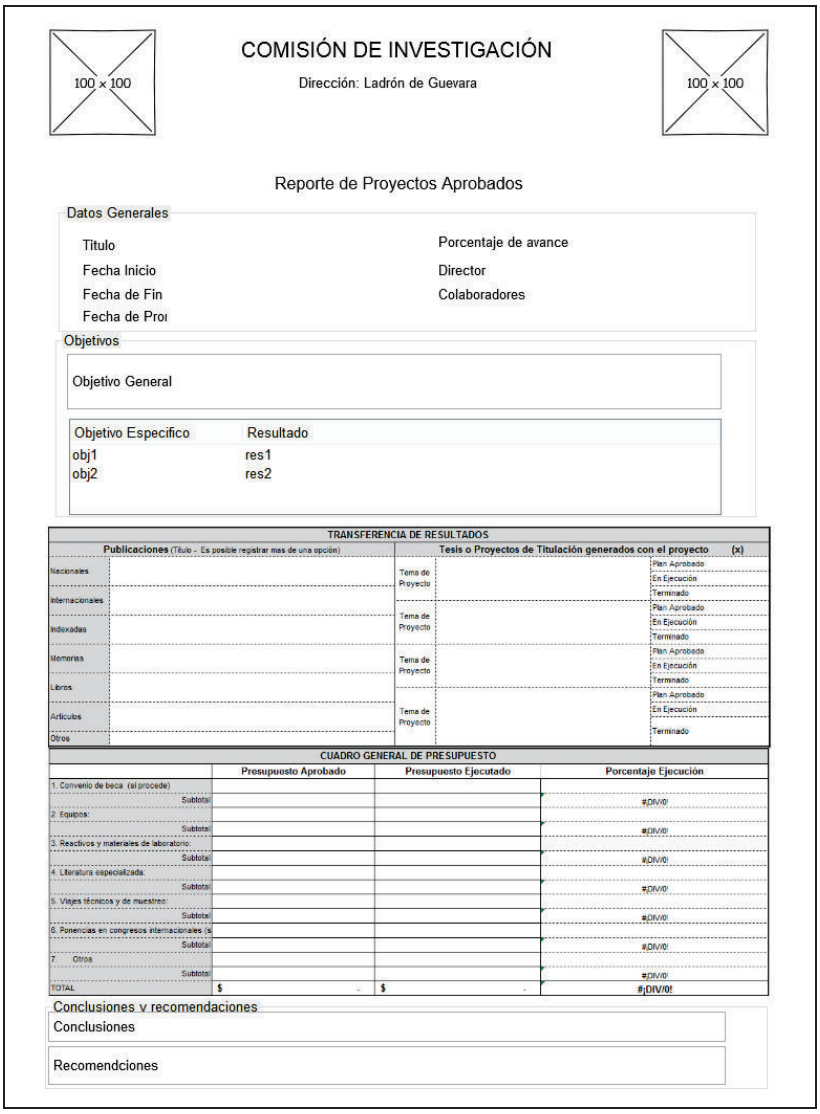


Figura 0.26. Reporte final del proyecto

Considerando que el diseño de las distintas plantillas tanto para la aplicación web como para los reportes, están terminados, en el siguiente capítulo se tratará el tema de la implementación respectiva para cada caso y adicionalmente, se realizarán las pruebas correspondientes para verificar su correcto funcionamiento.

Hay que acotar que a pesar de que está utilizando una metodología para el diseño web, los diseños que se manejarán podrían cambiar, pero de haber cambios, los mismos serían mínimos.

CAPÍTULO III

IMPLEMENTACIÓN DEL SISTEMA

1.11 INTRODUCCIÓN

Este capítulo abarcará todo el proceso que se desarrolló para la implementación, test y puesta en marcha del prototipo planteado tomando en consideración el modelo UWE, JSF (Oracle, Enero), JEE (Java Enterprise Edition), Alfresco y CryptoApplet como el gestor de firma digital.

1.12 ANÁLISIS DEL SISTEMA DESARROLLADO

1.12.1 ARQUITECTURA

El prototipo implementado está basado en una arquitectura de 3 capas, mismas que se presenta en la Figura 3.1:

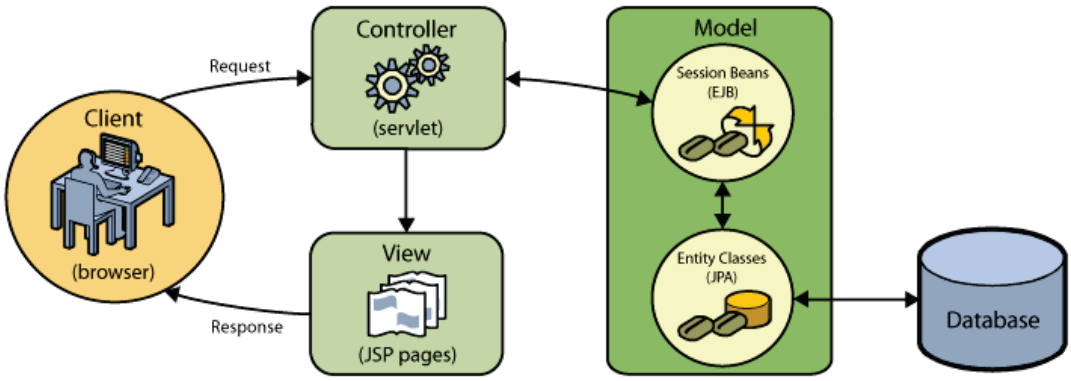


Figura 0.1. Arquitectura JEE. Fuente: (ORACLE, s.f.)

1.12.1.1 Capa de datos y de negocios

Estas capas se definen de acuerdo a la arquitectura JEE y presentan:

- La estructura de base de datos y la generación de objetos relacionales a partir de tablas relacionales y,
- Los componentes que se utilizarán para acceder a los objetos relacionales para posteriormente realizar operaciones contra la base de datos.

Para este análisis se partirá del esquema de datos generado para la base de datos presentada en la Figura 3.2.

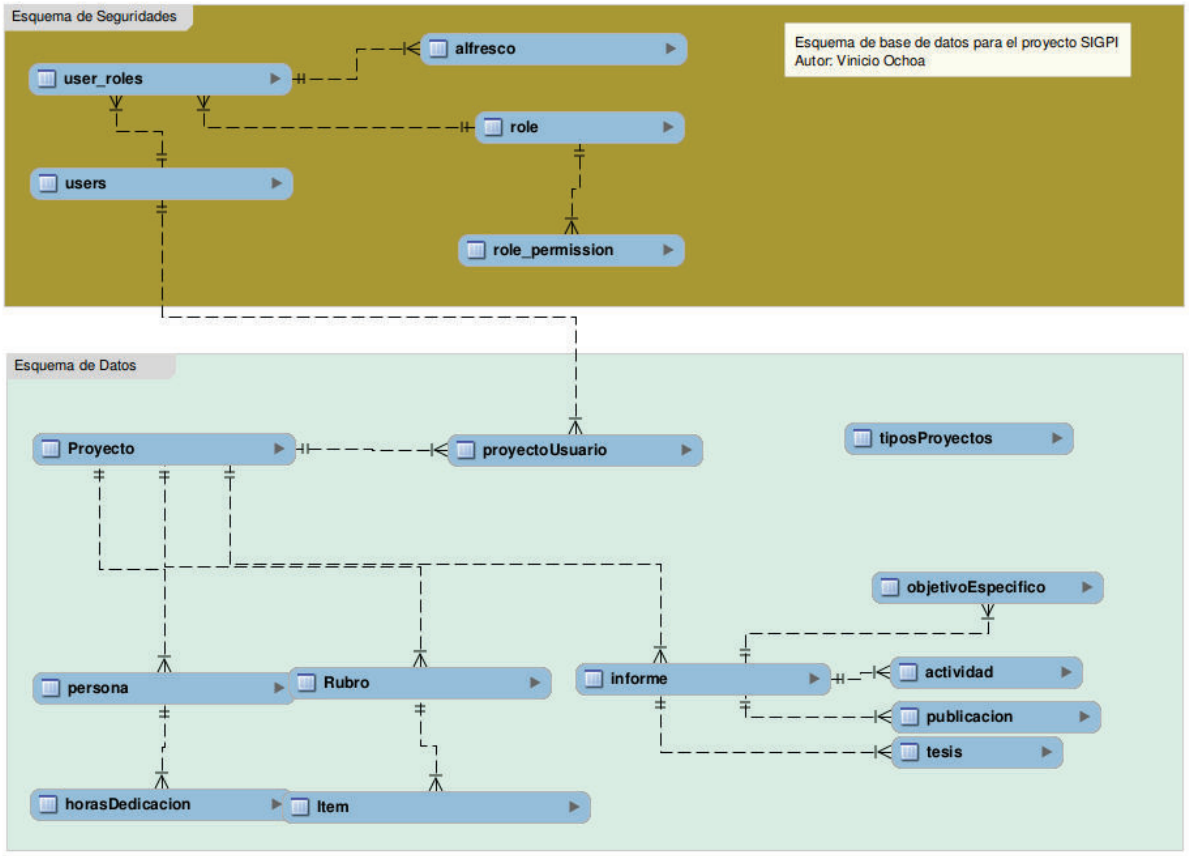


Figura 0.2. Esquema de base de datos (Anexo F)

Para gestionar las distintas operaciones SQL se ha utilizado una estructura JEE, la cual define una subcapa que permite integrar la base de datos con la capa de persistencia, misma que presenta los elementos presentados en la Figura 3.3:

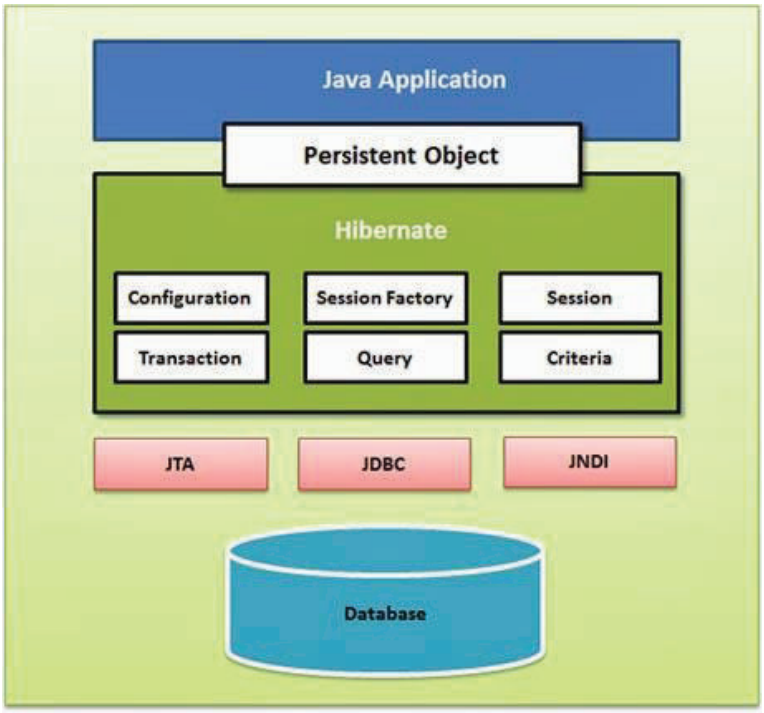


Figura 0.3. Arquitectura JEE – JPA. Fuente: (http: //javajeecastellano.files.wordpress.com, 2013)

1.12.1.1.1 Java Persistence API - JPA

Esta API está definida por ORACLE y considera cuatro secciones que permiten la interacción modelo / objeto relacional, véase la Figura 3.4:

- Java Persistence API
- Lenguaje de consulta
- Criteria API
- Mapeo Objeto/relación

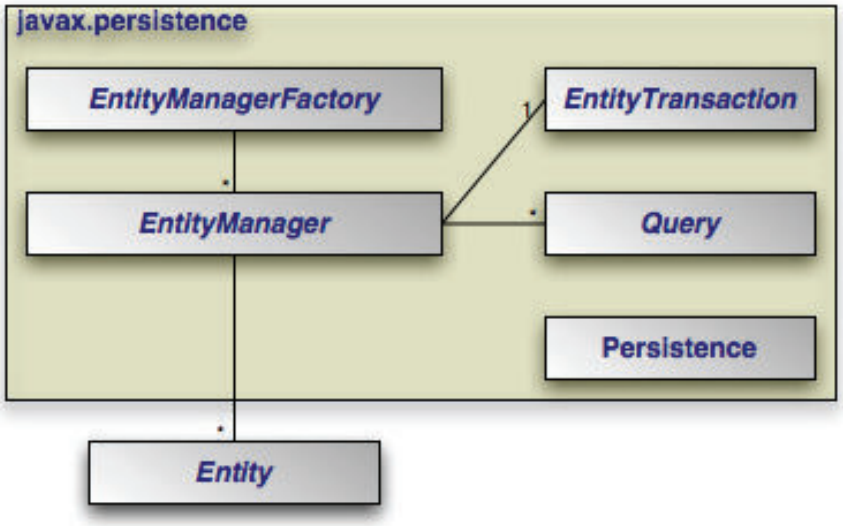


Figura 0.4. Arquitectura de JPA. Fuente: (OpenJPA, s.f.)

Estas secciones se abarcan en cuatro tipos de objetos básicos, así tenemos:

Entity: Son clases que representan una tabla completa en el modelo relacional de la base de datos. Todas las entidades se referencian en un documento **Persistence.xml**, mismo que mantiene una unidad de persistencia (PU) que permitirá instanciar objetos de las distintas entidades para realizar operaciones SQL.

Cada entidad presenta notaciones predefinidas que permiten interpretar los distintos tipos de datos dependiendo de su naturaleza, véase Figura 3.5:

- La clave primaria – @Id
- Los atributos propios de la clase - @Column(name = “xxx”)
- Las relaciones bidireccionales
    - Esta notación se coloca siempre y cuando exista una relación de uno a muchos o de muchos a muchos, por ejemplo:

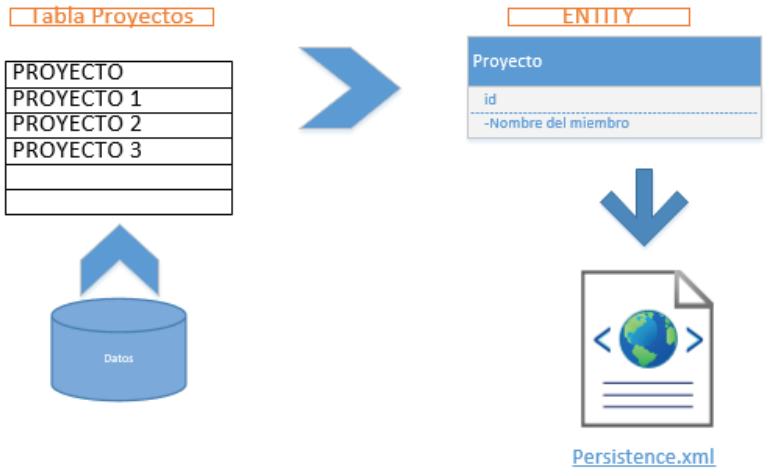


Figura 0.5. Conversión modelo / objeto relacional

En la Figura 3.6 se puede apreciar un modelo de ejemplo para el mapeo de objetos a entidad de las tablas Proyecto y Persona.

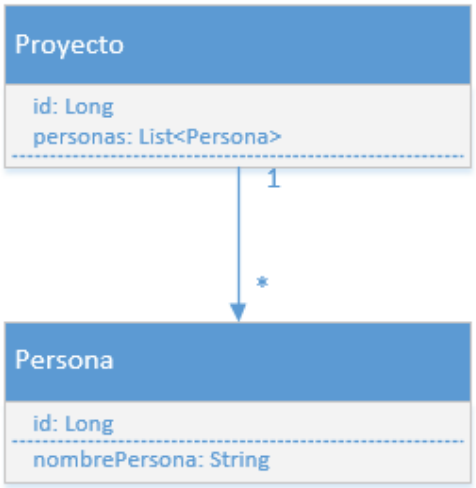


Figura 0.6. Relación Proyecto – Persona

Las notaciones más importantes para la representación de los campos y de las

relaciones de las tablas se presentan en la Tabla 3.1:

Tabla 0.1 Representación de columnas con JPA

Base de datos	Clase		
Columna	Persona	Proyecto	Representación
id_Proyecto	Relación ManyToOne: Se especifica en el atributo la notación @JoinColumn(name ="id_Proyecto")		Lista de objetos de tipo Persona
id_Persona		Relación OneToMany: Se especifica en el atributo la notación OneToMany(mappedBy="id_Persona")	Un objeto de tipo Persona
Fecha	Se especifica la notación @Temporal(Date)		Objeto de tipo Date
Columnas eventuales (No existen en la tabla)	Se especifica la notación @Transient Estos objetos jamás se persisten dado que no existen en la tabla de la base de datos.		Objetos temporales dinámicos de cualquier tipo de Java

EntityManager: Permite instanciar un contexto o una PU especificada en el documento Persistence.xml y presenta las siguientes propiedades:

- Gestión de los hilos para abrir o cerrar la conexión a la base de datos.

- Gestiona la transaccionalidad de operaciones.
- Gestiona las operaciones SQL (insert, delete, update, merge).

Permite definir las consultas mediante:

- Lenguaje de consulta orientados a objetos - JPQL[18] (Oracle, 2012)
- Lenguaje de consulta en base a criterios - Criteria API (Oracle, 2012)

La notación asignada es nombreClase – DAO (Oracle, 2002), por ejemplo: ProyectoDAO.

Para el proyecto se han creado tantos DAOs como clases se tienen en el modelo de bases de datos.

1.12.1.2 Capa de servicios

Esta capa define la forma como se interactúa con los objetos y la base de datos mediante servicios (DAOs). En la figura 3.7 se muestra un ejemplo de la implementación de un servicio.

Beneficios de esta implementación

- Permite centralizar las acciones a realizarse sobre determinada entidad
- Se pueden crear tantos servicios como se deseen
- Transparente para el desarrollador
- Facilidad para consumir servicios por otros usuarios ajenos al aplicativo.

---

[18] JPQL(*JPA Query Language*) es un lenguaje propio de la arquitectura JEE que permite realizar consultas desde un objeto a la base de datos.

A continuación se muestra la Figura 3.7. la cual especifica los parámetros más importantes sobre el modelo de interfaces utilizadas.

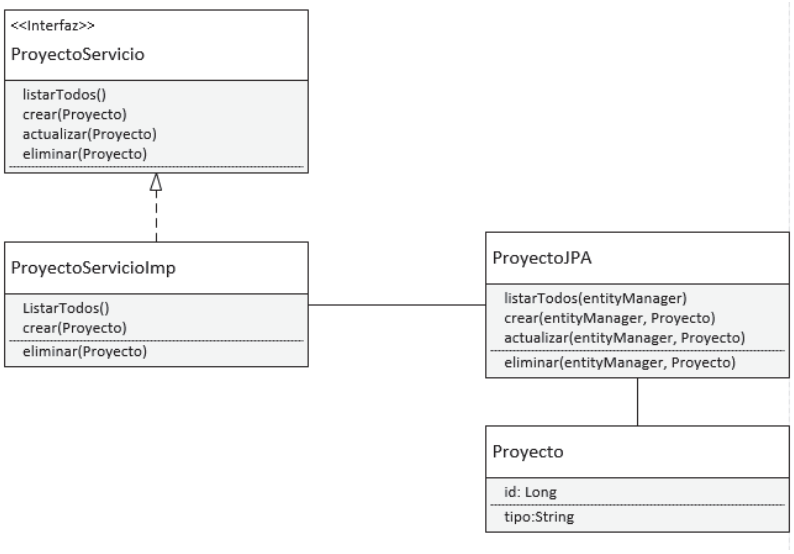


Figura 0.7. Implementación de los servicios

1.12.1.2.1 Delegación de Servicios

El modelo utilizado presenta la creación de un delegado que permitirá gestionar los servicios que serán publicados mediante:

- Web Services
- RestFull (Web Services Ligeros)

Ventajas de usar un delegado

- Brinda control de acceso a los servicios
- Los servicios pueden ser consumidos por cualquier tipo de dispositivo, sean estos: computadoras de escritorio o dispositivos móviles
- Oculta servicios que no se requiere que sean visibles para otros usuarios.

En la Figura 3.8 se presenta una implementación del delegado utilizado en el prototipo:

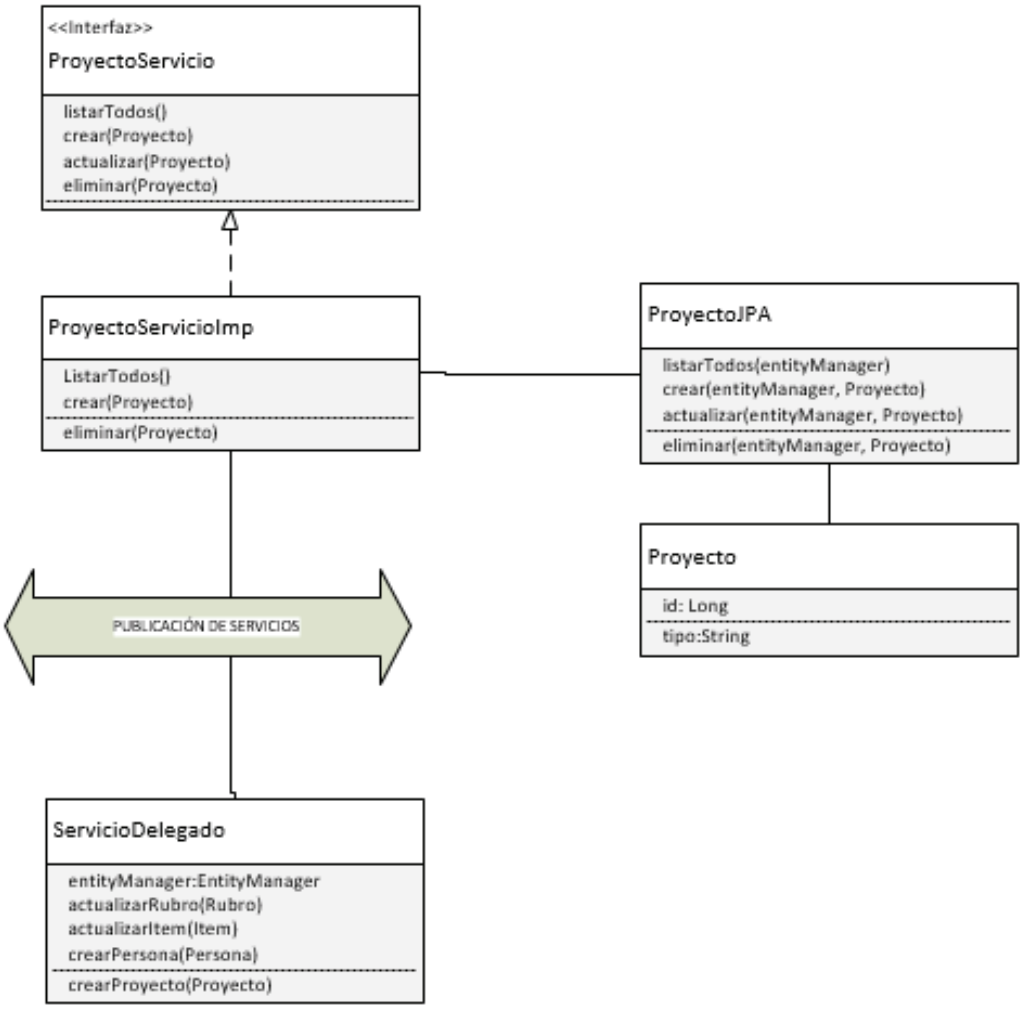


Figura 0.8. Publicación de servicios

1.12.1.3 Capa de presentación

Una vez que se han definido los delegados a usar para la interacción entre la capa de presentación con la capa de negocios y de datos, procederá a realizar un

análisis de las tecnologías utilizadas para la visualización del contenido en el navegador web.

1.12.2 JSF2

JSF2 (Oracle, 2012), es una tecnología del lado del servidor desarrollada para la capa de presentación y brinda las siguientes características:

- Separa la lógica de presentación de las aplicaciones web
- Permite manejar los estados(comportamiento) en la sesión de los componentes
- Maneja eventos y control de validación
- Reutilización de código
- Permite la utilización de un lenguaje de etiquetado XML
- Gestiona fácilmente las reglas de navegación
- Mantiene componentes UI que pueden ser extendidos según la necesidad
- Trabaja con el modelo visor controlador – MVC
- Permite trabajar con plantillas predefinidas(Facelets)

1.12.2.1 Funcionamiento

1.12.2.1.1 Modelo visor controlador

Es un patrón arquitectónico de diseño web que permite descentralizar el control y reduce la duplicidad de código, véase la Figura 3.9.

JSF trabajar con el modelo MVC facilitando el desarrollo de los aplicativos de manera sustancial dado que ahora el desarrollador se enfoca únicamente en una sección específica, mientras otros desarrolladores pueden trabajar en otras

secciones sin que esto entorpezca el trabajo.

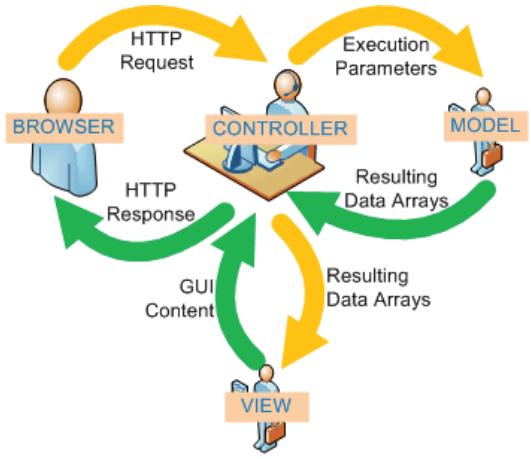


Figura 0.9. Modelo Visor – Controlador Fuente: (metodologiasdesistema, 2010)

Modelo: Define la lógica asociada atrás de una página web, es el contenido que se envía al controlador para procesar una petición.

Controlador: Es el encargado de procesar los requerimientos generados en la aplicación.

Vista: Es una página web que permite representar los valores enviados por el controlador.

1.12.3 PRIMEFACES (Primefaces, 2014)

Es una librería del lado del cliente que permite renderizar componentes HTML para mejorar la interfaz gráfica del usuario.

1.12.3.1 Características

- Soporte para Ajax: Permite realizar actualizaciones inmediatas de componentes con un mínimo código JavaScript
- Se integra fácilmente con entornos de desarrollo Java especialmente con JSF
- Soporte para dispositivos móviles
- Es un componente OpenSource

En el aplicativo web desarrollado se utilizó este framework para la visualización de páginas HTML con actualizaciones dinámicas – Ajax y componentes PrimeFaces para la generación de formularios.

1.12.4 SISTEMA GESTOR DE PROYECTOS DE INVESTIGACIÓN – SIGPI

Figura 0.10. Pantalla de inicio de SIGPI

SIGPI es un sistema desarrollado para la gestión técnica – financiera de los

proyectos de investigación que se desarrollan en la EPN, este sistema es una extensión del Sistema Gestor Documental Alfresco y cuya finalidad es facilitar al investigador información sobre la evolución financiera de los proyectos que se encuentren desarrollando.

Considerando los frameworks analizados en este capítulo se procede a explicar el funcionamiento del sistema desarrollado para el Vicerrectorado de Investigación y Proyección Social de la EPN, para lo cual, a continuación se indicará algunas de las características más relevantes:

- Soporte básico para dispositivos móviles: Desde tablets hasta computadoras de escritorio.
- Presenta una interfaz gráfica gestionada mediante PrimeFaces como framework visual de componentes.
- Gestión de reportes
- Firma digital usando CryptoApplet.
- Control de acceso mediante roles.
- Modificación de documentos gestionados por Alfresco mediante Web Services y RestFull.

A continuación se detallarán algunas características del sistema para entender mejor el funcionamiento del mismo.

1.12.4.1 Control de Acceso

Para sistema de seguridad del aplicativo se implementó considerando tres roles de acceso con diferentes permisos cada uno, por lo que se utilizó un framework de seguridad conocido como **Apache Shiro** (Apache Shiro, 2014)**,** el cual brinda las siguientes características a entornos web desarrollados con Java:

Autenticación: Permite la autenticación de usuarios mediante diversas formas siendo las más importantes:

- LDAP
- Base de datos
- Realms

Criptografía: Apache Shiro brinda la posibilidad de aplicar métodos criptográficos para la generación o encriptación de códigos sensibles como son claves de acceso. Entre los métodos de encriptación y cifrado que permite Apache Shiro se tiene:

- **Cifrado:** Permite encriptar datos utilizando variadas técnicas como: AES[19], DES[20], Base64[21] y Hex[22].
- **Hashing**: Técnicas que se aplican a un código mediante las cuales se obtienen un valor diferente que puede ser utilizado para verificar las credenciales de usuarios, entre algunas técnicas de hashing se tiene: MD5 [23]y SH4256 [24]

---

[19] AES (*Advanced Encryption Standard*) es un estándar criptográfico de llave simétrica.

[20] DES (*Data Encryption Standard*) es un estándar de llave simétrica por bloques que en la actualidad se considera inseguro y se ha dejado de usar. Se lo remplazo con el algoritmo AES.

[21] BASE64: Es un sistema de numeración posicional que usa 64 caracteres como base.

[22] HEX: Codificación de caracteres en formato hexadecimal.

[23] MD5 (*Message-Digest Algorithm 5)* es un algoritmo hash que genera una cadena de 32 dígitos con codificación ASCII.

[24] SHA256 (*Secure Hash Algorithm*) es un algoritmo hash de longitud de 256 códigos ASCII.

Integración web: Posibilita la integración a otros sistemas mediante:

- **Spring Framework**: Es un framework de desarrollo no estandarizado por Java que permite trabajar con Spring Security para asegurar o extender la seguridad de aplicaciones informáticas.

- **Guice Dependency Injection Framework**: Es un framework desarrollado por Google que posibilita la **inyección de dependencias** con aplicativos desarrollados en Java utilizando **Google Guice** (Google, s.f.).

- **CAS SSO server**: permite proteger un aplicativo mediante una **Central de Autenticacion – CAS** (Jasig, 2014), la cual brinda la posibilidad de centralizar la seguridad en un único lugar permitiendo distribuir los aplicativos en diferentes lugares dejando de lado la seguridad para enfocarse en el desarrollo de los mismos. CAS permite loguear a los usuarios mediante una autenticación **Single Sign On – SSO**[25].

Autorización: El control de acceso se puede manejar de las siguientes maneras:

- **Permisos:** Representan las acciones que un usuario puede realizar en un determinado lugar. Existen tres tipos de permisos en Apache Shiro:

[25] SSO (*Single Sign On*) es un procedimiento de autenticación que permite mediante una clave autenticarse a varios servicios automáticamente.

- **Nivel de Recurso**: Permite interactuar directamente con un recurso.
- **Nivel de Instancia**: Permite interactuar directamente con un objeto específico.
- **Nivel de Atributo**: permite interactuar directamente con un atributo de un objeto especifico

Roles

- **Roles Implícitos:** Son roles mantienen permisos establecidos implícitamente. Por ejemplo si un usuario presenta el rol de administrador obtendrá automáticamente todos los permisos del sistema para realizar cualquier tipo de operación.

- **Roles Explícitos:** Son roles que presentan permisos explícitos definidos mediante programación o en un archivo de configuración.

Manejo de sesiones

Permite gestionar de una manera inteligente y ágil las sesiones que se inician por los usuarios. Es así que presenta las siguientes características:

- **Session Timeout:** Gestiona las sesiones expiradas por inactividad del usuario frente al sistema web.

- **Session Listener:** Permite gestionar las sesiones que se generan por los usuarios al momento de iniciar sesión en un aplicativo web

- **Session Valdation:** Permite validar si una sesión esta expirada o detenida por algún error para luego restituirla con una nueva.

1.12.4.1.1 Proceso de autenticación

Para este aplicativo se ha desarrollado el proceso de autenticación mostrado en la Figura 3.11.

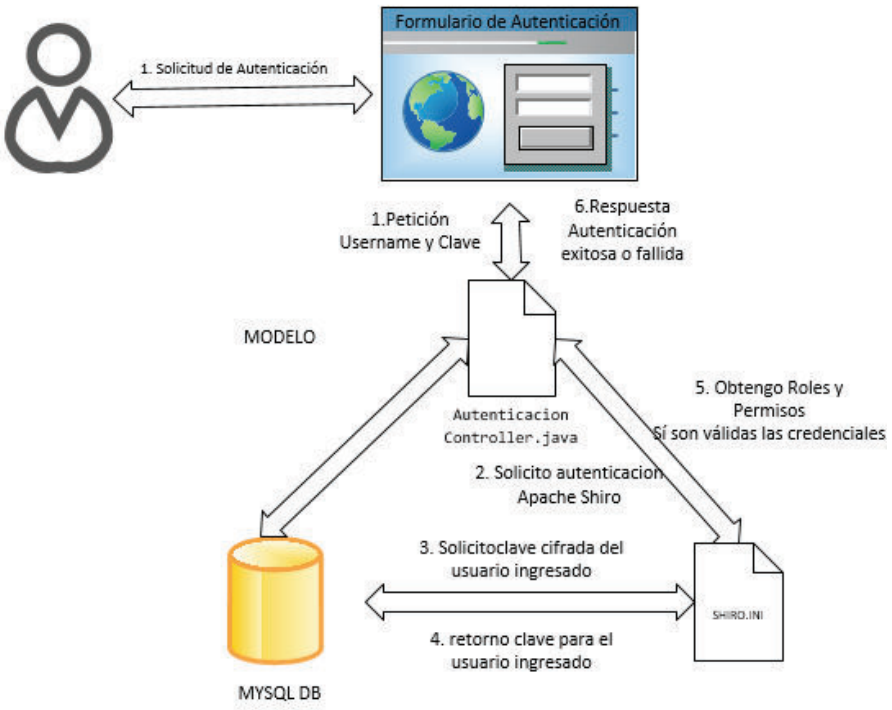


Figura 0.11. Proceso de autenticación para un usuario

La autenticación se la gestiona mediante 2 archivos (uno java y otro de configuración) y la base de datos. A continuación se detalla el proceso:

AutenticacionController.java: Esta clase es la encargada de obtener el username y la clave ingresada por el usuario.

Shiro.ini: Archivo de configuración para Apache Shiro y permite especificar lo siguiente:

- Parámetros de acceso a la base de datos, como son user, password y esquema de la base de datos y consultas SQL, véase Figura 3.12.

```
[main]
# listener = org.apache.shiro.config.event.LoggingBeanListener
jdbcRealm = org.apache.shiro.realm.jdbc.JdbcRealm
jdbcRealm.permissionsLookupEnabled = true
jdbcRealm.authenticationQuery = select password from users where username = ? and activo = 1
jdbcRealm.userRolesQuery = select role_name from user_roles, users where userId = users_userId AND username = ?
jdbcRealm.permissionsQuery = select permission from role_permission, role where role_roleId = roleId AND name = ?
ds = com.mysql.jdbc.jdbc2.optional.MysqlDataSource
ds.url = jdbc:mysql://127.0.0.1:3306/appVicerrectorado
```

Figura 0.12. Configuración del archivo shiro.xml (1)

- Definición de roles y permisos: Para este caso no se han definido los roles en este archivo dado que se utiliza la base de datos como fábrica de este ítem, sin embargo, y considerando que se realizó un análisis previo se han creado tres tipos de roles:

    - Administrador,
    - Colaborador e
    - Investigador.

- Definición de urls: cada usuario tendrá acceso dependiendo del rol a una url, es así, que se han definido las urls presentadas en la Figura 3.13.

```
[urls]
# The /login.jsp is not restricted to authenticated users (otherwise no one could log in!), but
# the 'authc' filter must still be specified for it so it can process that url's
# login submissions. It is 'smart' enough to allow those requests through as specified by the
# shiro.loginUrl above.
/login.jsf = authc
/logout.jsf = logout
/adm/** = authc
/col/** = authc
/web/** = authc
```

Figura 0.13. Configuración del archivo shiro.xml (2)

Considerando que cada usuario que se registre entrará en un rol establecido por el administrador del sistema SIGPI, los usuarios únicamente accederán a la url asignada a cada rol según se indica en la Tabla 3.2.

Tabla 0.2 Relación Url - Rol

URL	ROL
/adm	Administrador
/col	Colaborador
/web	Investigador

Base de datos: Contiene los roles y permisos asociados a cada usuario registrado en el sistema. El proceso sería el siguiente:

1. El usuario ingresa su username y clave de ingreso
2. El sistema envía mediante una petición post al modelo(Controlador JSF)
3. El controlador intenta autenticar mediante Apache Shiro
    a. Se busca el archivo shiro.ini y se obtiene los parámetros de conexión a la base de datos, para posteriormente solicitar mediante una

operación SELECT la clave del usuario ingresado para hacer el match. Si las credenciales coincide con la clave ingresada se considera autenticado.

4. Si se ha autenticado correctamente, se devuelven los roles y permisos obtenidos desde la base.
5. Se verifica en el controlador los roles y se re direcciona hacia la url predefinida por cada rol.

Cada url presenta una estructura de directorios para la presentación de hojas HTML, tal como se muestra Figura 3.14.

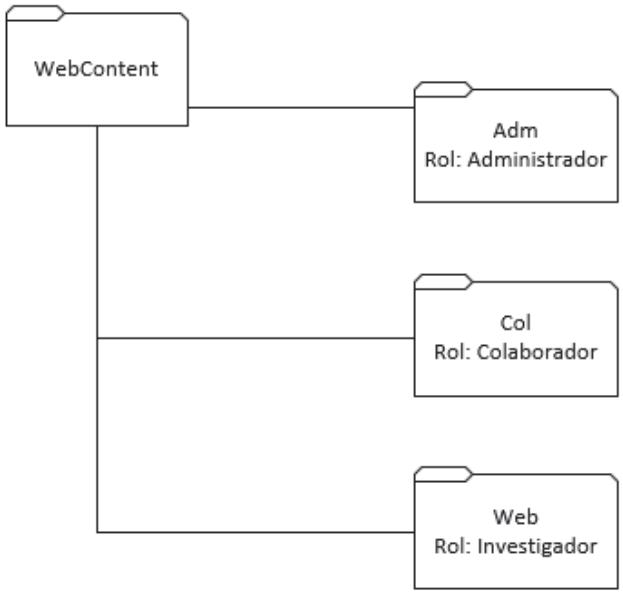


Figura 0.14. Jerarquía de directorios según los roles y urls asignadas

Dentro de cada carpeta se definen las paginas HTML que serán accedidas por cada usuario.

1.12.4.1.2 Gestión de usuarios

El proceso de gestionar los usuarios será ejecutado únicamente por las personas que mantengan el rol de **Administrador,** por lo que se generó una página web que permita lo siguiente:

- Creación y modificación de usuarios
- Asignación de un rol
- Si el rol asignado es de tipo Investigador, se puede asignar el o los proyectos vigentes para el usuario seleccionado.

En la Figura 3.15. se muestra la interfaz de administración generada para el caso de tener rol de Investigador:

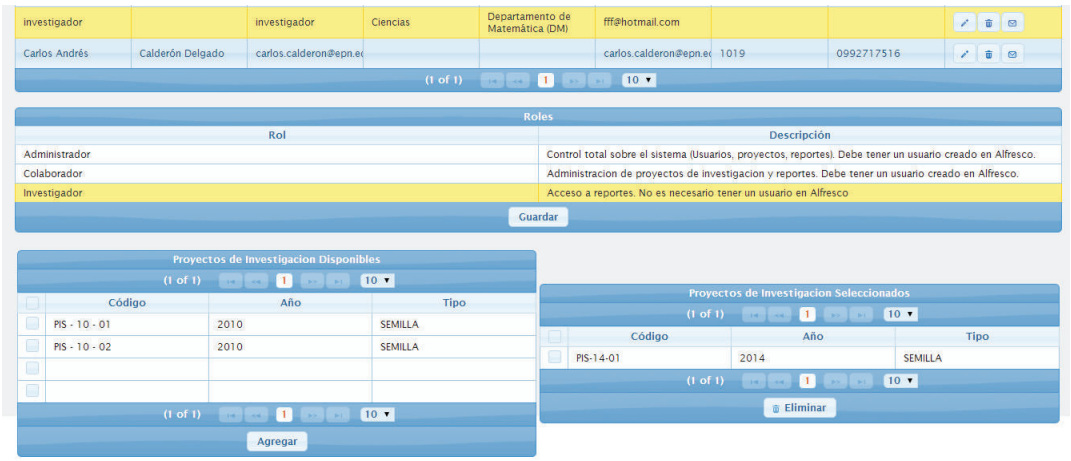


Figura 0.15. Interfaz de administración para un Investigador

1.12.4.1.3 Integración con Alfresco

Funcionamiento de los servicios web

Como ya se indicó en el capítulo II acerca del funcionamiento de los web Services, se procederá a hacer unas breves pautas con lo que tiene que ver con:

Autenticación

Para este caso, Alfresco presenta un API que facilita el proceso de autenticación. El proceso para autenticar es el siguiente:

1. El usuario ingresa una única vez, salvo que cambie las credenciales en el Alfresco, el usuario y clave.
2. Dicha credencial se envía al Alfresco para su verificación.
3. Sí las credencias son correctas se emite un token, mismo que servirá para realizar cualquier operación que tenga que ver con la manipulación u obtención de recursos desde Alfresco hacia el aplicativo desarrollado.

En la Figura 3.16 se muestra el esquema general para autenticar el sistema con Alfresco:

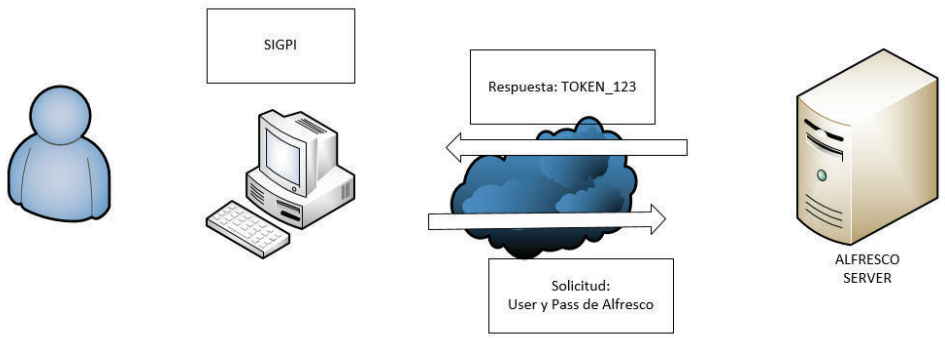


Figura 0.16. Esquema general para autenticación del Sistema Alfresco

Adquisición de etiquetas y categorías

Considerando que el usuario posea credenciales válidas y por ende un token expedido por Alfresco, se procederá a obtener las etiquetas y categorías asociadas a cada documento almacenada en el repositorio digital, para lo cual se utilizó el código mostrado en la Figura 3.17.

```
private void cargarEtiquetas() {
    authAlfrescoWS.setUserName(this.userName);
    authAlfrescoWS.setPassword(this.password);

    if (authAlfrescoWS.autentica()) {
        try {
            tag.categorias();
        } catch (RemoteException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

}
```

Figura 0.17. Método de autenticación utilizando los servicios web de Alfresco

Dicho código permite autenticar al usuario y obtener las secciones del Alfresco que se presentan en las Figuras 3.18 y 3.19 respectivamente.

Figura 0.18. Categorías

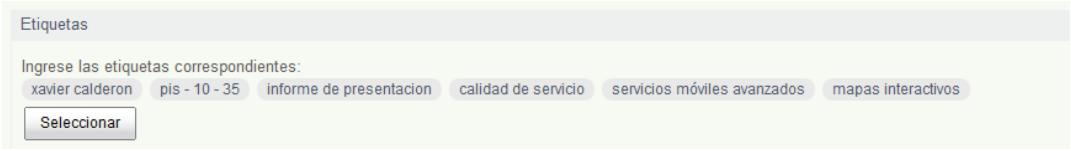


Figura 0.19. Etiquetas

Funcionamiento de los web scripts (RestFull)

En el caso de los servicios web ligeros se utilizó el token obtenido de la autenticación con web services propios de Alfresco para en base a este token obtener las tareas mediante peticiones post generadas desde el aplicativo desarrollado. Para esto Alfresco presenta una sección específica para gestionar peticiones con los servicios web ligeros que se encuentran alojados en la URL: http://documentacion.epn.edu.ec:8080/alfresco/service/. Véase la Figura 3.20.

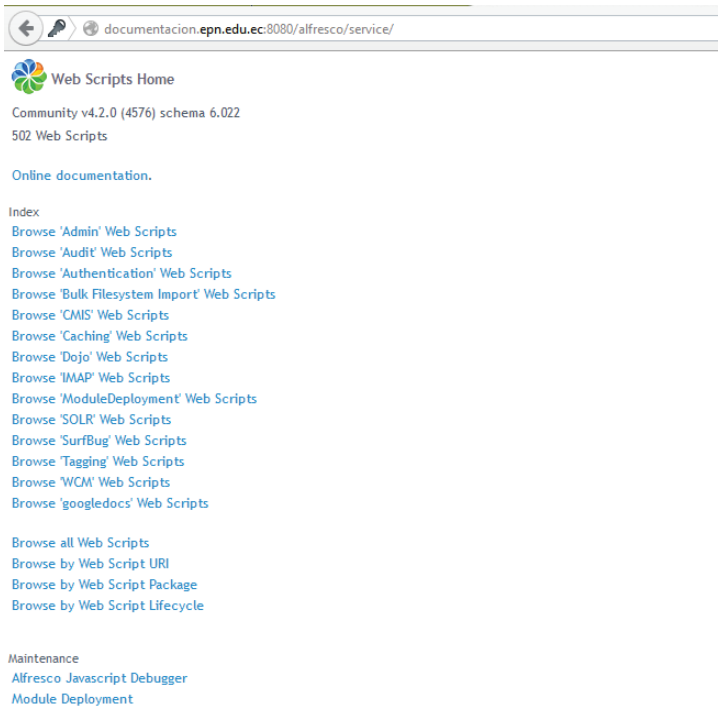


Figura 0.20. Servicios web ligeros publicados por Alfresco

Adquisición de tareas

Conociendo que Alfresco presenta una API de servicios de flujos de trabajo, se utilizó esta sección para gestionar las tareas asignadas a otros usuarios. Estas peticiones se procesan de la siguiente manera:

1. Se obtienen las credenciales del usuario autenticado en el sistema SIGPI.
2. En base a las credenciales se obtiene el token de autenticación de Alfresco.
3. Con el token se generan las peticiones con los servicios web ligeros y se obtiene la información requerida para mostrarla en el sistema.

Un ejemplo de esta implementación se puede apreciar en la Figura 3.21.

```
public void obtenerTareasCompletadas() {
    tareasCompletadas.clear();
    try {

        DefaultHttpClient httpClient = new DefaultHttpClient();
        HttpGet getRequest = new HttpGet(
                properties.getProperty("WEB_SCRIPT") + "?alf_ticket="
                        + au.getToken() + "&state=completed");
        getRequest.addHeader("accept", "application/json");

        HttpResponse response = httpClient.execute(getRequest);
        if (response.getStatusLine().getStatusCode() != 200) {
            throw new RuntimeException("fallo: HTTP error code: "
                    + response.getStatusLine().getStatusCode());
        }
        InputStreamReader isr = new InputStreamReader(response.getEntity()
                .getContent());
        cargarTareas(isr, true);

    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
```

Figura 0.21. Método para consultar mediante RestFull las tareas ejecutadas en Alfresco

Adquisición de documentos desde Alfresco

OpenCMIS: Alfresco permite interactuar con los documentos mediante

OpenCMIS[26], que es el framework aceptado para trabajar con el repositorio mediante WebScripts y Web Services.

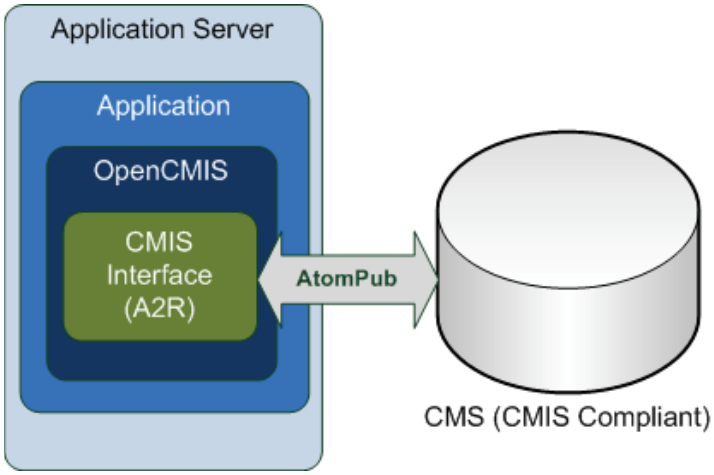


Figura 0.22. Arquitectura de OpenCMIS Fuente: **(gleenders, 2010)**

Al trabajar con este framework, se define un modelo de archivo en texto claro con una estructura de árbol básico que permite navegar sobre el repositorio de Alfresco. Para obtener este documento se puede acceder a la URL: http://documentacion.epn.edu.ec:8080/alfresco/cmisatom.

Autenticación de OpenCMIS

OpenCMIS define el siguiente procedimiento de autenticación para obtener los documentos del repositorio:

- Ingresar las credenciales de un usuario válido en Alfresco al sistema SIGPI

[26] OpenCMIS: Es un framework OpenSource que permite interactuar con repositorios de documentos. Para más información véase la URL: http://chemistry.apache.org/java/opencmis.html

- La URL de Alfresco que brinda el servicio está ubicada en la url: http://documentacion.epn.edu.ec:8080/alfresco/cmisatom

- En base a la dirección web se especifica que se guarde la sesión mediante la gestión de cookies para que el usuario final interactúe con los documentos y carpetas según sea requerido.

- Se habilita la compresión para mejorar la velocidad de recuperación y envió de contenido desde y hacia el repositorio digital de Alfresco.

- Se trabaja con el repositorio en base a una URI[27] especifica dentro del sistema, para el caso del Vicerrectorado de Investigación se creó un directorio especifico que mantiene todos los proyectos de investigación denominado:
    - CMISPATHPROYECTOS: /Sitios/comiex/documentLibrary/Proyectos de Investigación.

- A partir de esta uri se procederá a trabajar con los documentos de los distintos proyectos de investigación.

En la Figura 3.23 se puede apreciar el procedimiento generado para recuperar el repositorio de Alfresco en base a la URI específica generada para el Vicerrectorado de Investigación.

---

[27] URI (Uniform Resource Identifier), permite identificar a un recurso unívocamente en el repositorio de Alfresco.

```java
@Override
public boolean conectarRepositorio() {
    boolean auth = false;
    try {
        Map<String, String> parameter = new HashMap<String, String>();
        // user credentials
        parameter.put(SessionParameter.USER, this.userName);
        parameter.put(SessionParameter.PASSWORD, this.password);

        // connection settings
        parameter.put(SessionParameter.ATOMPUB_URL, CMISPATH);
        parameter.put(SessionParameter.BINDING_TYPE,
                BindingType.ATOMPUB.value());

        // set the alfresco object factory
        parameter.put(SessionParameter.OBJECT_FACTORY_CLASS,
                "org.alfresco.cmis.client.impl.AlfrescoObjectFactoryImpl");
        parameter.put(SessionParameter.AUTH_HTTP_BASIC, "true");
        parameter.put(SessionParameter.COOKIES, "true");
        parameter.put(SessionParameter.COMPRESSION, "true");
        parameter.put(SessionParameter.PROXY_USER, this.userName);
        parameter.put(SessionParameter.PROXY_PASSWORD, this.password);

        // create session
        SessionFactory factory = SessionFactoryImpl.newInstance();
        this.session = factory.getRepositories(parameter).get(0)
                .createSession();

        cargarEtiquetas();
        auth = true;
    } catch (Exception e) {
        Utils.mensaje("error", "Usuario o password incorrectos");

    }
```

Figura 0.23. Método para obtener la estructura de directorios de Alfresco mediante CMIS

En la Figura 3.24 se presenta el resultado de la obtención de los documentos.

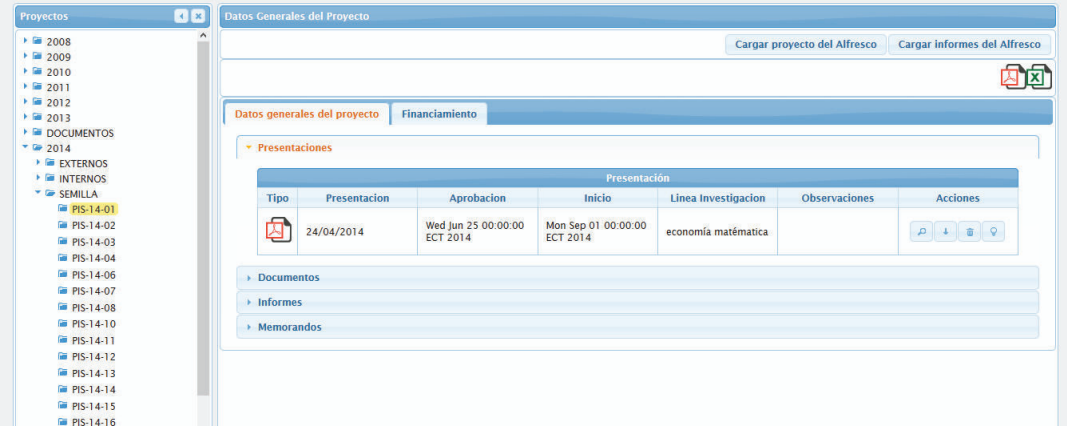


Figura 0.24. Obtención del repositorio del Alfresco

En base a los distintos modelos que se generaron para clasificar y organizar de manera correcta a los documentos ingresados en cada proyecto, se realiza una búsqueda mediante OpenCMIS hasta obtener todos los documentos correspondientes a:

- Presentaciones
- Documentos generales: presentaciones, videos, otros, etc.
- Informes de avance y final
- Memorandos enviados y recibidos

Por cada documento se obtiene las propiedades de metadatos correspondientes al modelo que se le haya asignado inicialmente, permitiendo al usuario la modificación o eliminación del documento según sea requerido. En la Figura 3.25 se presenta una ventana del modelo de presentación del proyecto semilla.

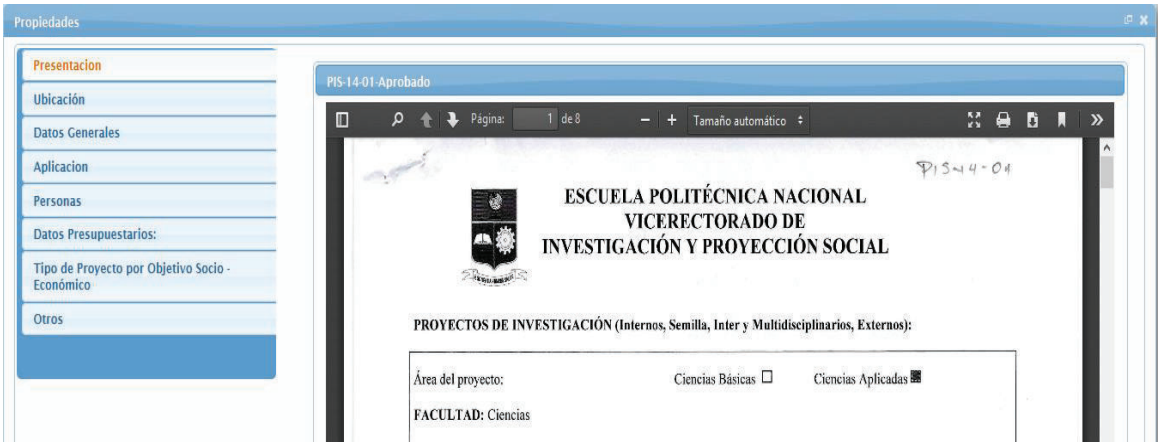


Figura 0.25. Interfaz de mantenimiento para los documentos extraídos desde Alfresco. Proyecto Semilla PIS-14-01

1.12.5 ANÁLISIS DE LA FIRMA DIGITAL CON CRYPTOAPPLET

Para este tópico se utilizó OpenSSL (OpenSSL, 2014) el cual es un framework OpenSource que permite la gestión de certificados de seguridad para cualquier fin específico como es el caso de la firma digital o certificados para proteger sitios web.

1.12.5.1 Generación de certificados digitales

El proceso para generar certificados digitales con capacidad de firma digital consiste del siguiente procedimiento:

- Crear una autoridad certificadora (CA[28]): **Trusted Root Certificate Authority,** la cual será usada para firmar los certificados que se generen para cada usuario. Véase Figura 3.26.

```
openssl req -new -x509 -keyout ca.key -out ca.crt -config openssl.cnf -extensions ca_ext
```

Figura 0.26 Generación de la CA

Para ello crea un certificado auto firmado con la siguiente sentencia:

- Se crea un certificado para firmar en base a la CA. Véase Figura 3.27.

```
openssl req -new -out mail.csr -keyout mail.key -config openssl.cnf
```

Figura 0.27. Creación del certificado

[28] **Certification Authority (CA):** Es una entidad de confianza encargada de revocar o emitir certificados digitales o cualquier dispositivo de seguridad como tokens o tarjetas de coordenadas.

- Se firma el certificado con el certificado raíz. Vease Figura 3.28.

```
openssl x509 -req -in mail.csr -out mail.crt -CA ca.crt -CAkey ca.key -CAserial serial.txt -extfile openssl.cnf -extensions mail_ext
```

Figura 0.28. Firma del certificado generado con la CA

- Exportamos el certificado con la clave pública y privada con la extensión PKCS#12 (OpenSSL, s.f.). Véase Figura 3.29.

```
openssl pkcs12 -export –clcerts -in mail.crt -out mail.p12 -inkey mail.key
```

Figura 0.29. Exportación del certificado autofirmado

- Con esto ya se tiene un certificado digital para firmar documentos.

1.12.5.2 Compilación de CryptoApplet

Para realizar la firma digital con CryptoApplet (UJI, 2014) se necesita agregar al KeyStore (Oracle, 2014) de CryptoApplet la entidad de certificación root para que pueda validar el certificado que se generó anteriormente, para ello se exporta el certificado raíz con la herramienta keytool de Java. Véase la Figura 3.30.

```
keytool -import -alias vice -file ca.ctr -keystore cryptoapplet -storepass cryptoapplet
```

Figura 0.30. Comando para exportar un certificado con extensión PKCS12

Con el certificado raíz en el KeyStore de CryptoApplet se procede a compilar en base al comando especificado en la documentación oficial.

```
mvn -DskipTests clean install
```

Figura 0.31. Compilación de CryptoApplet

El comando de la Figura 3.31 genera los componentes que serán guardados en el sistema SIGPI para firmar los documentos extraídos desde Alfresco.

1.12.5.3 Proceso para firmar

La firma digital puede realizarse por cualquiera de los tres roles. La interfaz se muestra en la Figura 3.32.

Figura 0.32. Interfaz para firmar y verificar la firma digital

El proceso de firma se ejecuta de la siguiente manera:

- El usuario ingresa el documento en formato PDF
- Sube el documento a la memoria del sistema
- Firma el documento en base al certificado generado.

- Para esto se iniciará CryptoApplet y se seleccionará el certificado digital desde la ubicación donde se encuentre almacenado. Véase la Figura 3.33.

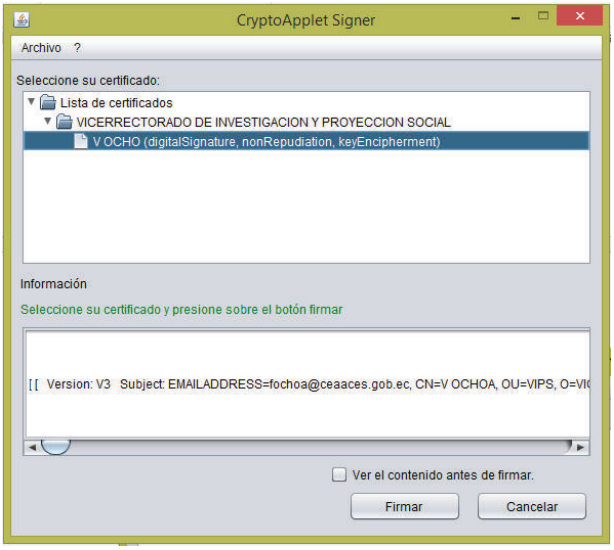


Figura 0.33. Selección del certificado para firmar el documento

- Sí la firma se ha generado correctamente se presentará la Figura 3.34.

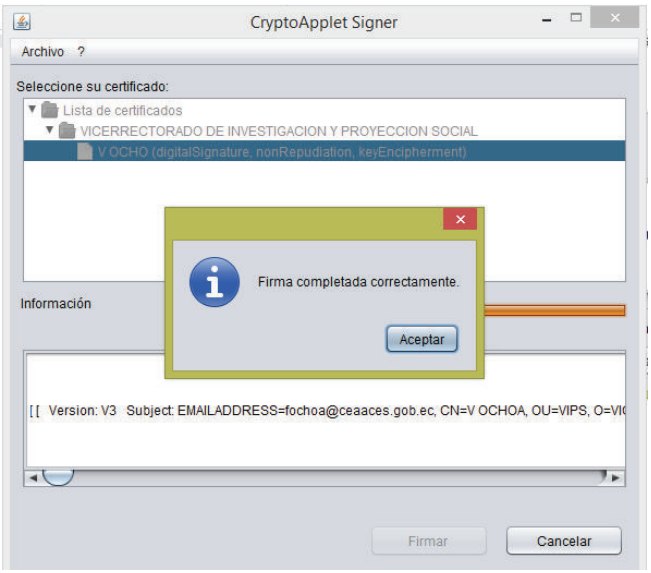


Figura 0.34. Firma Digital exitosa con CryptoApplet

- Dado que se firmó el documento, se almacena una copia en Alfresco para posteriormente permitirle al usuario descargar su documento para respaldo o para verificar la firma del mismo.
- En Alfresco el documento se generará en la ruta ***Documentos/Documentos_Investigadores***, misma que será revisada cada vez que se firme un nuevo archivo. Véase la Figura 3.35.

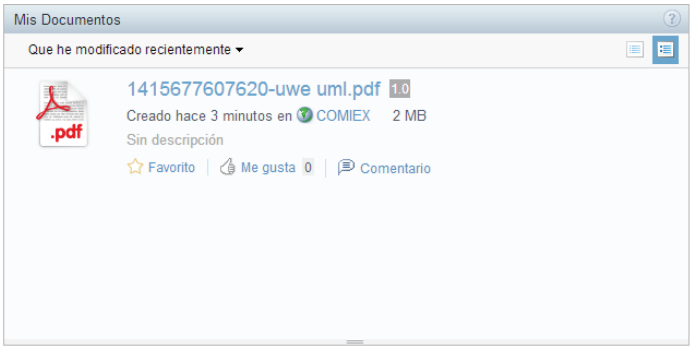


Figura 0.35. Documento firmado en Alfresco

Para el caso de verificar la firma digital se la puede hacer de dos maneras:

- La primera opción se presenta cuando se ha firmado el documento, el sistema da la opción de descargarlo y opcionalmente verificar la firma digital tal como se muestra en la Figura 3.36.

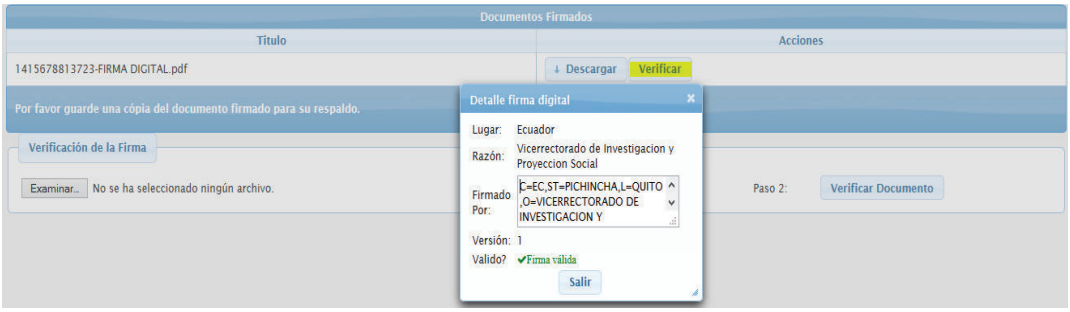


Figura 0.36. Verificación de la firma digital

- La segunda opción es cargar el documento firmado y verificar la firma digital. Véase la Figura 3.37.

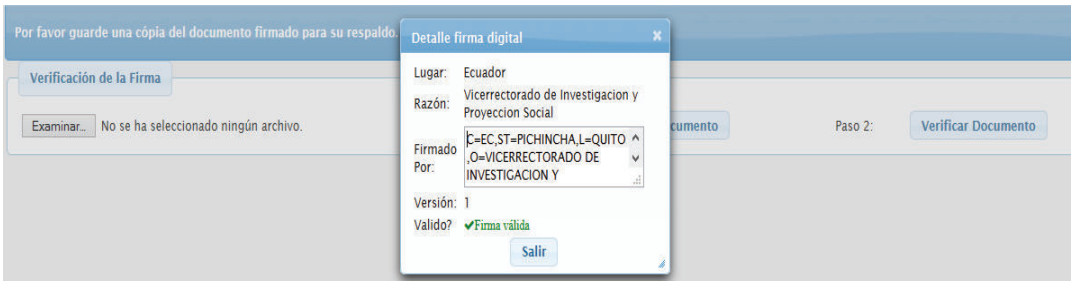


Figura 0.37. Verificación de la firma digital

1.12.6 ANÁLISIS DE REPORTES

La generación de reportes se realizó utilizando las herramientas JXLS (JXLS, 2014) y IReport Designer (JasperSoft, s.f.) de la empresa JasperReport los cuales permitirán la generación de informes personalizados y dinámicos en base a plantillas y consultas a una base de datos.

1.12.6.1 JXLS

1.12.6.1.1 Descripción

Es un framework OpenSource que está diseñado para generar reportes en base a plantillas XLS con la ayuda de listas dinámicas de objetos generadas desde Java. Para poder ejecutarlas se utilizó el siguiente segmento de código:

```
Map beans = new HashMap();
beans.put("proyecto", listaProyectos);
descargarReporte(beans, "reporte-general.xls",
        "Reporte General Proyectos.xls");
```

Figura 0.38. Segmento de código para generar un reporte en base a plantillas

1.12.6.1.2 Implementación

Templates

El acceso a los objetos se genera mediante programación orientada a objetos, es decir para acceder a los atributos se usa el operador punto (.), de igual manera se lo realiza en la hoja de Excel, misma que viene a ser el prototipo del reporte a generar cuando se ingresen las listas con los objetos a reportar. Véase la Figura 3.39.

N.	Código	Nombre del Proyecto	Director	Departamento
	${proyecto.codigo}	${proyecto.nombre}	${proyecto.director.nombrePersona} ${proyecto.director.apellidoPersona}	${proyecto.departamento}

Figura 0.39. Acceso a atributos de los objetos en el template

Así mismo, se pueden realizar operaciones matemáticas con el framework simplemente definiendo la operación a realizar en inglés.

Informe 1	Informe 2	Presupuesto Aprobado	Presupuesto Modificado	Presupuesto Ejecutado	% Ejecución Financiera	Saldos
		${proyecto.totalPresupuestoAprobado}	${proyecto.totalPresupuestoModificado}	${proyecto.totalPresupuestoEjecutado}	$[Q7/P7]//proyecto	$[P7-Q7]//proyecto
	TOTAL:	$[SUM(O7)]	$[SUM(P7)]	$[SUM(Q7)]	**$[SUM(R7)]**	**$[SUM(S7)]**

Figura 0.40. Definición de operaciones en el template

Como se puede observar en la Figura 3.40., si se desea hacer la operación suma, se lo puede realizar colocando el operador $[SUM (P7)], el mismo que sumará todos los valores desde la posición P7 hasta el valor de objetos Pn.

Sí se desea hacer divisiones, se puede especificar la operación $[Q7/P7]//proyecto mediante el operador especial (//), el cual indica que se debe realizar la operación de división por todos los objetos que existan en la lista (Qn/Pn).

Considerando los requerimientos del Vicerrectorado de Investigación y Proyección Social se han generado tres tipos de reportes, así tenemos:

- Reporte general de proyectos
- Reporte parcial de proyectos: Reporte solo para investigadores con los valores financieros totales que se mantienen a la fecha.
- Reporte de rubros y de hora de trabajo del docente.
- Otros repositorios

1.12.6.2 JasperReport – IReport Designer

1.12.6.2.1 IReport Designer

Descripción

Esta herramienta OpenSource permite al usuario administrador generar reportes adicionales para la comisión simplemente utilizando la conexión con la base de datos del sistema SIGPI.

Creación de reportes dinámicos

Para crear los reportes se hace necesario instalar esta herramienta desde la fuente https://community.jaspersoft.com/project/ireport-designer/releases, en donde se debe seleccionar la distribución que se desee. Existen tanto para Windows como para Linux y Mac.

Se instalará el sistema en una máquina que posea conexión con la base de datos de SIGPI.

Una vez instalado iReport se procede con los siguientes pasos:

- Crear una conexión con la base de datos. Véase la Figura 3.41.

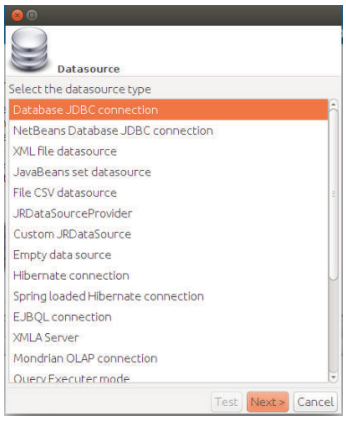


Figura 0.41. Conexión con la base de datos desde iReport

- Se crea un nuevo reporte y mediante el diseñador de consultas se obtienen todas las tablas que se requieran para el reporte. Véase la Figura 3.42.

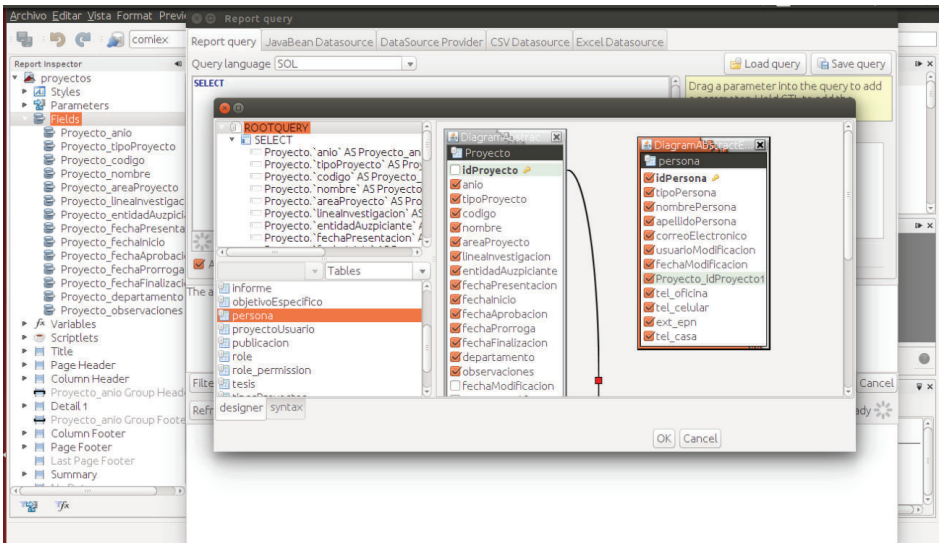


Figura 0.42. Obtención de tablas desde el diseñador de consultas del iReport.

- Se extraen las columnas de las tablas. Véase la Figura 3.43.

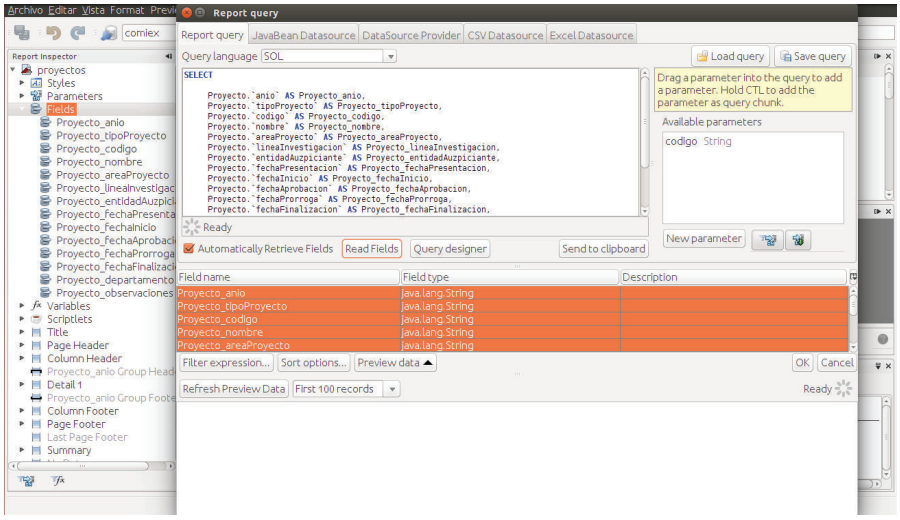


Figura 0.43. Obtención de columnas.

- Se arrastran las etiquetas que representan a cada columna según como se desee el reporte. Véase la Figura 3.44.

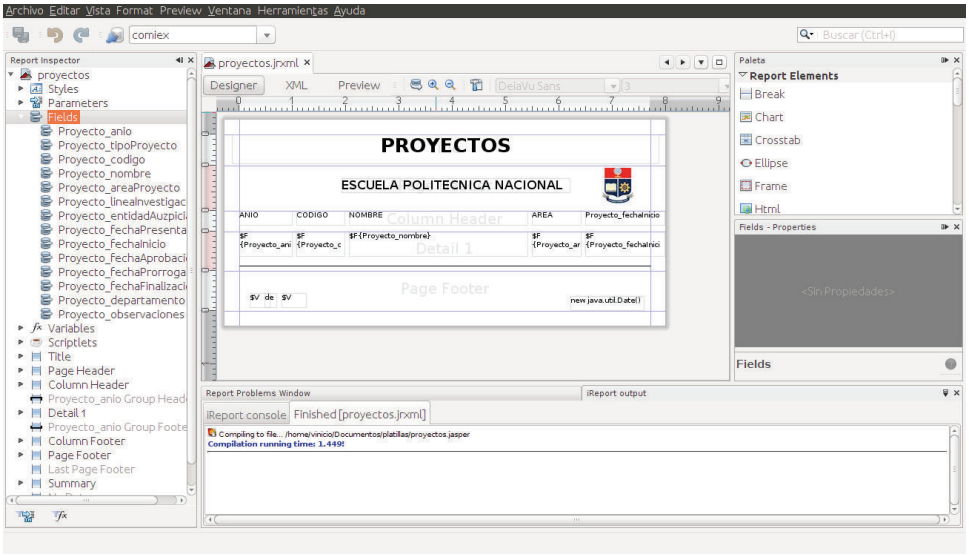


Figura 0.44. Diseño del reporte.

- Se revisa la vista previa. Véase la Figura 3.45.

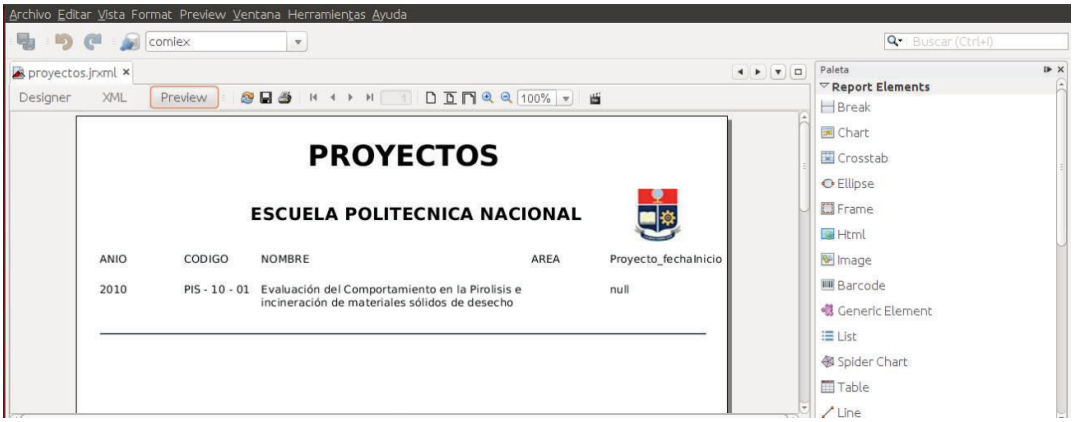


Figura 0.45. Vista previa del reporte.

- Para finalizar se exporta al formato deseado. Véase la Figura 3.46.

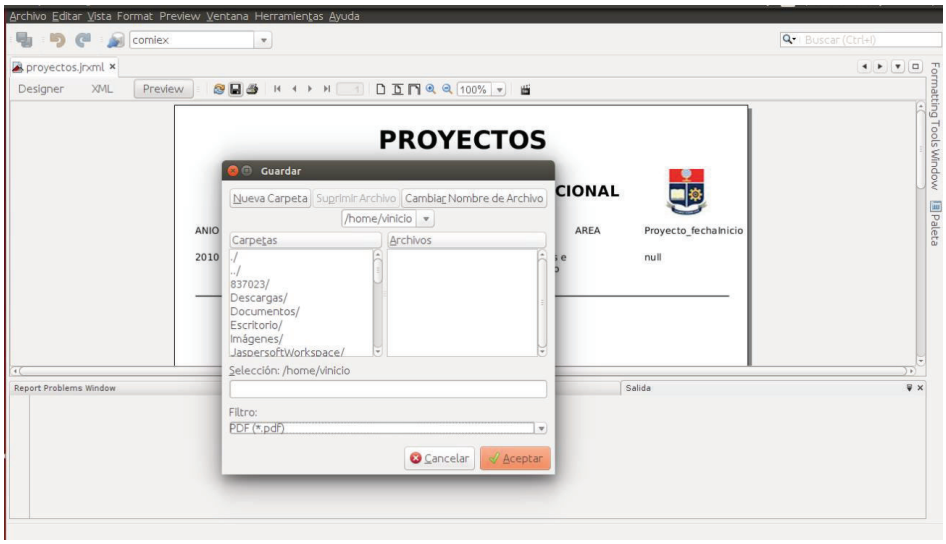


Figura 0.46. Exportación del reporte a PDF.

Mediante este procedimiento el usuario con rol de administrador estará en la

capacidad de crear los reportes que considere necesarios adicionales a los que el sistema ya exporta.

1.13 INSTALACIÓN

La instalación de SIGPI se lo realizará en el mismo servidor donde se encuentra Alfresco, y este será colocado en un servidor Tomcat independiente.

1.13.1 CARACTERÍSTICAS DEL SERVIDOR WEB

1.13.1.1 IPTABLES

Iptables define un conjunto de reglas que permiten controlar el flujo del tráfico de entrada y salida mediante la apertura o cierre de puertos o IPs.

En este caso iptables nos permitirá habilitar el acceso para el servidor web Tomcat que estará esperando peticiones por el puerto 8090, así como también, se deberá habilitar el puerto para él envió de correo.

Resumiendo se tendría que habilitar los puertos listados en la Tabla 3.3:

Tabla 0.3. Resumen de puertos y sentencias a ejecutar

Servicio	Puerto	Comando
Tomcat - SIGPI	8090	iptables -I INPUT -p tcp --dport 8090 -j ACCEPT
Alfresco	8080, 80	iptables -I INPUT -p tcp --dport 8080 -j ACCEPT
Correo	465	iptables -I INPUT -p tcp --dport 465 -j ACCEPT

1.14 PRUEBAS DE FUNCIONAMIENTO

1.14.1 PRUEBAS FUNCIONALES DE LOS MÓDULOS

Las pruebas del funcionamiento de cada módulo se realizaron con las personas que colaboran con el Vicerrectorado de Investigación y Proyección Social, por lo que se corrigieron todos los inconvenientes hasta llegar al sistema estable que se encuentra disponible en la web.

1.14.2 PRUEBAS DE NAVEGABILIDAD

El sistema funciona correctamente con todos los navegadores web, pudiendo apreciarse simplemente unos pequeños inconvenientes en cuanto a resolución y tamaños de objetos que se hacen imperceptibles en ciertos navegadores. Véase la Figura 3.47.

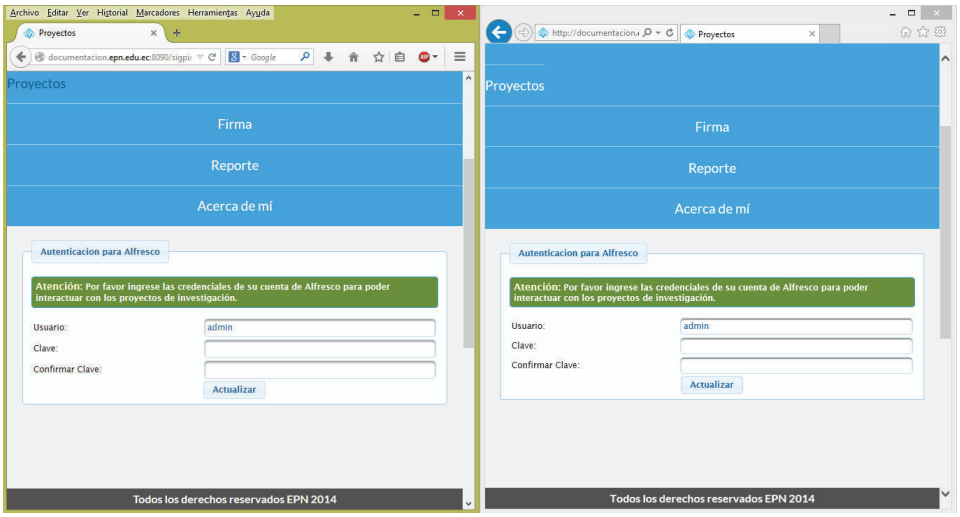


Figura 0.47. Comparación de Chrome e Internet Explorer

1.14.3 PRUEBAS DE FIRMA DIGITAL

1.14.3.1 Firefox

El sistema firma y verifica la firma correctamente, tal y como se muestra en las siguientes figuras 3.48. y 3.49.

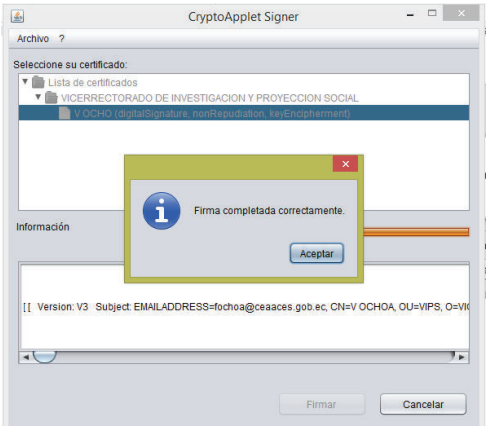


Figura 0.48. Firma Digital exitosa con CryptoApplet

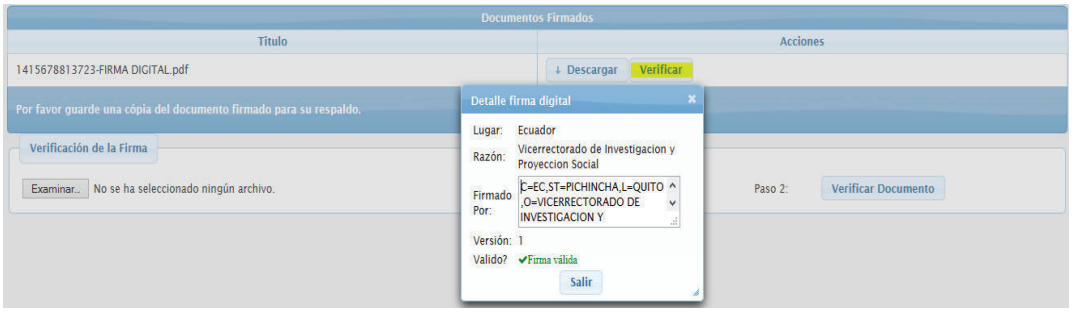


Figura 0.49. Verificación de la firma digital

1.14.3.2 Chrome

Para el caso de firmar los documentos, el sistema funciona correctamente, pero para verificar la firma no se ejecuta adecuadamente por lo que no se recomienda utilizar este navegador para realizar la firma digital.

1.14.3.3 Internet Explorer

Al igual que con Chrome, la firma digital se ejecuta correctamente, pero para verificar la firma existen problemas relacionados con el navegador, por lo que se recomienda que no se utilice este navegador.

Resumiendo los resultados de las pruebas se presenta la Tabla 3.4.

Tabla 0.4. Tabla comparativa de compatibilidad entre navegadores web

	CHROME	FIREFOX	INTERNET EXPLORER 8 O SUPERIOR
Navegabilidad	Muy buena	Muy buena	Muy buena
Firma digital	Muy buena	Muy buena	Muy buena
Verificación de la firma digital	Malo	Muy Buena	Muy Buena
Otros servicios generales: reportería, ingreso de datos y gestión de usuarios	Muy buena	Muy buena	Muy buena

Se puede concluir que para utilizar el aplicativo se puede utilizar cualquier navegador web, sin embargo, se debe tener presente que Chrome presenta problemas al momento de verificar la firma digital al igual que Internet Explorer.

1.15 ANÁLISIS DE COSTOS

El sistema desarrollado fue elaborado en base a la Tabla 3.5.

Tabla 0.5. Tabla referencial de meses ocupados para la creación del SIGPI

Ítem	Tiempo/mes	Observaciones
Investigación	3	Como interactuar con Alfresco y Java. Cuáles son los procedimientos del VIPS
Implementación	6	Implementación de los modelos y el sistema SIGPI para el consumo de los servicios y modelos planteados. Diseño gráfico del SIGPI
Configuraciones del servidor	0.2	Configuraciones del servidor para que soporte Alfresco y hardening del mismo.
TOTAL	**9.2 meses**	

Considerando que se dedicó cada día 4 horas al menos en promedio, esto nos da un equivalente en horas de:

$$\frac{9.2\ meses}{1} \times \frac{30\ dias}{1\ mes} = \mathbf{276\ dias}$$

$$276\ dias \times \frac{8\ horas}{1\ dia} \approx \mathbf{2200\ horas}$$

Figura 0.50. Calculo de horas estimadas para el desarrollo del proyecto

De estas 2200 horas, considerando que cada hora se la avalúa en 12 dólares; el sistema tendría un costo aproximado de:

$$\boldsymbol{Costo\ del\ sistema} \approx \mathbf{\$26500}$$

Figura 0.51. Costo estimado del sistema

Este sería el costo del software desarrollado con todos los componentes y la investigación que se generó para su construcción.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

1.16 CONCLUSIONES

- El sistema de gestión documental Alfresco, presenta grandes ventajas con respecto a la gestión que se le da a los documentos ingresados, esto gracias a que este ECM cuenta con la norma ISO 9001, misma que define las normas de calidad que deberían de tener los documentos.

- El repositorio de Alfresco permitió la adopción de las normativas internas que se gestionan en el VIPS, de una manera medianamente fácil, es así, que ya se han adoptado más modelos para la gestión de docentes prometeos y recursos humanos.

- La gestión documental en el país aún tiene falencias grandes esto debido a la falta de cultura organizacional y documental en las personas en cuanto al ingreso de documentación a repositorios específicos, esto se apreció en el VIPS.

- Se observó que si no se especifica adecuadamente los valores de memoria con los que trabajara el servidor web de Alfresco, este tiende a consumir toda la memoria disponible, ocasionando fallos de memoria y la caída del servidor web. Esto se solucionó con la agregación de un script que libera la memoria cache que ya no se está usando, manteniendo de esta manera la memoria consumida en un rango estable y seguro para ejecutar el resto de aplicaciones que se llevan en el equipo.

1.16.1 Limitaciones

- SIGPI no cuenta con el soporte necesario para dispositivos móviles pequeños como celulares o tablets, por lo que la resolución se ve afectada y la gestión de los proyectos se hace ineficiente.

- La firma digital solo se la puede ejecutar en el sistema SIGPI, aunque los documentos firmados se almacenan directamente en Alfresco para su posterior gestión y validación.

- Al no poseer una entidad certificadora de confianza avalada por todos los navegadores y por el KeyStore de java, los certificados generados deberán ser agregados al KeyStore de CryptoApplet y al de la máquina de java (JRE) para que se pueda realizar tanto la firma como la verificación de la firma digital.

1.16.2 Ampliaciones del sistema

Si bien es cierto, SIGPI es un sistema para computadoras de escritorio, se puede extender las funcionalidades a dispositivos móviles haciendo uso del api de web services o RestFull que brinda Alfresco por lo que en base a este prototipo se facilitaría la construcción de nuevas herramientas que faciliten la automatización de tareas en las diferentes dependencias de la Escuela Politécnica Nacional.

1.17 RECOMENDACIONES

- Se debe de ingresar todos los campos correspondientes a cada tipo de documento con la finalidad de que al final cuando se desee gestionar los proyectos de investigación, el sistema esté en la capacidad de generar reportes confiables y completos alineados a la realidad.

- Se sugiere que se adquiera certificados avalados por una entidad certificadora de confianza reconocida por el KeyStore de Java, pudiendo ser: DigiCert y VeriSign algunas opciones muy buenas.

- Es deseable que se adquiera un servidor dedicado para que se encargue únicamente de la gestión del repositorio Alfresco y del sistema SIGPI y otro que solo gestione la base de datos ya que el ingreso concurrente de información puede ocasionar que el servidor colapse por insuficiencia de recursos tanto de memoria como de disco duro.

REFERENCIAS BIBLIOGRÁFICAS

REFERENCIAS WEB

Vialgroup. (20 de 05 de 2013). *Vialgroup*. Obtenido de Vialgroup: http://vialgroup.com.ec/site/index.php/es/temas-de-interes/16-token. Consultado en mayo 2013.

Adobe Systems. (s.f.). *http://www.adobe.com/*. Obtenido de Adobe Systems: http://www.adobe.com/es/products/acrobat/adobepdf.html. Consultado en julio 2013.

Alfresco. (20 de 06 de 2012). *Alfresco*. Obtenido de Alfresco: http://docs.alfresco.com/4.0/index.jsp. Consultado en Agosto 2013.

Alfresco. (20 de 05 de 2013). *Alfresco Content Repository*. Obtenido de Alfresco Content Repository: http://docs.alfresco.com/4.0/index.jsp?topic=%2Fcom.alfresco.enterprise.doc%2Fconcepts%2Fweb-app-framework.html. Consultado en Agosto 2013.

Alfresco. (16 de 04 de 2013). *Alfresco Framework*. Obtenido de Alfresco Framework: http://docs.alfresco.com/4.0/index.jsp?topic=%2Fcom.alfresco.enterprise.doc%2Fconcepts%2Fweb-app-framework.html. Consultado en Agosto 2013.

Alfresco. (9 de Junio de 2014). *Alfresco*. Obtenido de Alfresco Wiki: http://wiki.alfresco.com/wiki/Data_Dictionary_Guide. Consultado en Agosto 2013.

Alfresco. (s.f.). *Alfresco Wiki*. Obtenido de Alfresco Wiki: http://wiki.alfresco.com/wiki/Data_Dictionary_Guide. Consultado en Agosto 2013.

Alfresco. (s.f.). *Alfresco Wki*. Obtenido de Alfresco Wki: http://wiki.alfresco.com/wiki/Security_and_Authentication. Consultado en Septiembre 2013.

athento. (14 de 12 de 2009). *athento*. Obtenido de athento:

http://blog.athento.com/2009/02/alfresco-vs-nuxeo-gestion-documental.html/. Consultado en Septiembre 2013.

http://juanlaracarrion.blogspot.com. (14 de 12 de 2010). *juanlaracarrion*. Obtenido de juanlaracarrion: http://juanlaracarrion.blogspot.com/2010/12/arquitectura-de-3-capas-en-c.html. Consultado en septiempre 2013.

Galeano, L. (10 de 2013). *elproyectodeluisgaleano*. Obtenido de elproyectodeluisgaleano: http://elproyectodeluisgaliano.blogspot.com/2012/11/metodologia-uwe-aplicada-mi-solucion.html. Consultado en Diciembre 2013.

Bernal, I. V. (s.f.). *SlideShare*. Obtenido de SlideShare: http://www.slideshare.net/isra00/introduccin-al-software-de-gestion-documental-ecm. Consultado en Septiembre 2013.

java-forums. (15 de 02 de 2012). *java-forums*. Obtenido de java-forums: http://www.java-forums.org/ocpjwsd/55587-tutorial-review-soap-web-services-web-service-developer-exam.html. Consultado en septiembre 2013.

w3c. (20 de 02 de 2003). *Xades*. Obtenido de Xades: http://www.w3.org/TR/XAdES/. Consultado en septiembre 2013.

Webdav. (1 de Enero de 2009). *Webdav*. Obtenido de Webdav: http://www.webdav.org/specs/. Consultado en septiembre 2013.

Wikipedia. (16 de 05 de 2013). *Wikipedia.* Obtenido de http://es.wikipedia.org/wiki/Servicio_web. Consultado en septiembre 2013.

Wikipedia. (02 de 09 de 2013). *Wikipedia*. Obtenido de Wikipedia: http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado. Consultado en septiembre 2013.

http://users.dcc.uchile.cl. (s.f.). *http://users.dcc.uchile.cl*. Obtenido de http://users.dcc.uchile.cl:

http://users.dcc.uchile.cl/~psalinas/uml/casosuso.html. Consultado en diciembre 2013.

http://www.w3.org. (11 de 04 de 2013). *w3*. Obtenido de w3: http://www.w3.org/TR/xmldsig-bestpractices/. Consultado en diciembre 2013.

http://www.w3c.es. (1 de Octubre de 1994). *W3C*. Obtenido de W3C: http://www.w3c.es/Consorcio/. Consultado en diciembre 2013.

https://tools.ietf.org. (1 de Septiembre de 2009). *IETF*. Obtenido de IETF: https://tools.ietf.org/html/rfc5652. Consultado en diciembre 2013.

Ietf. (1 de Noviembre de 1995). *Ietf*. Obtenido de Ietf: https://tools.ietf.org/html/rfc1866. Consultado en diciembre 2013.

Ietf. (19 de Diciembre de 1997). *Ietf*. Obtenido de Ietf: https://tools.ietf.org/html/draft-leach-cifs-v1-spec-01. Consultado en diciembre 2013.

Ietf. (1 de Enero de 2003). *Ietf*. Obtenido de Ietf: https://tools.ietf.org/html/rfc3444. Consultado en diciembre 2013.

Ietf. (1 de Mayo de 2007). *Ietf*. Obtenido de Ietf: https://tools.ietf.org/html/rfc4825. Consultado en diciembre 2013.

Java Dzone. (1 de Abril de 2010). *Java Dzone*. Obtenido de Java Dzone: http://java.dzone.com/articles/java-content-repository-best. Consultado en diciembre 2013.

Franco, J. M. (10 de 15 de 2001). *trevinca*. Obtenido de trevinca: http://trevinca.ei.uvigo.es/~txapi/espanol/proyecto/superior/memoria/node173.html. Consultado en Enero 2014.

metodologiasdesistema. (10 de 2010). *metodologiasdesistema*. Obtenido de metodologiasdesistema: http://metodologiasdesistemas.blogspot.com/2010/10/los-frameworks-son-nuestro-silver.html. Consultado en enero 2014.

Apache Shiro. (2014). *Apache Shiro*. Obtenido de Apache Shiro: http://shiro.apache.org/. Consultado en Febrero 2014.

blyx. (18 de 02 de 2013). *blyx*. Obtenido de blyx: http://blyx.com/. Consultado en Febrero 2014

Castilla-La Mancha. (20 de 05 de 2013). *Adaptatic*. Obtenido de Adaptatic: http://www.cocemfeclm.org/documentos/FirmaDigital.pdf. Consultado en Febrero 2014.

gleenders. (14 de 07 de 2010). *gleenders*. Obtenido de gleenders: http://gleenders.blogspot.com/2010/07/implementing-cmis.html. Consultado en Marzo 2014.

Gonzáles, B. (7 de 07 de 2004). *desarrolloweb*. Obtenido de desarrolloweb: http://www.desarrolloweb.com/articulos/1557.php. Consultado en Marzo 2014.

Chemistry Apache. (s.f.). *Chemistry Apache*. Obtenido de http://chemistry.apache.org/java/opencmis.html. Consultado en Marzo 2014

Google. (s.f.). *Google Guice*. Obtenido de Google Guice: https://code.google.com/p/google-guice/. Consultado en Mayo 2014.

Jasig. (27 de Mayo de 2014). *CAS Jasig*. Obtenido de CAS Jasig: https://wiki.jasig.org/display/CAS/Home. Consultado en abril 2014.

programacion.net. (s.f.). *programacion.net*. Obtenido de programacion.net: http://programacion.net/articulo/integracion_de_jsf-_spring_e_hibernate_para_crear_una_aplicacion_web_del_mundo_real_307/3. Consultado en julio 2014.

Spring. (s.f.). *Spring Open Source*. Obtenido de Spring Open Source: https://spring.io/. Consultado en julio 2014.

doc.3tera.com. (2011). Obtenido de doc.3tera.com: http://doc.3tera.com/AppLogic30/es/Catalog_Ref/index.htm?toc.htm?CatApplicationServersTomcat.html. Consultado en Julio 2014.

http://hitachi-id.com. (20 de 01 de 2014). *hitachi-id*. Obtenido de hitachi-id: http://hitachi-id.com/concepts/certificate_authority.html. Consultado en Julio 2014.

OpenJPA. (s.f.). *openjpa*. Obtenido de openjpa: http://openjpa.apache.org/builds/1.0.0/apache-openjpa-1.0.0/docs/manual/jpa_overview_arch.html. Consultado en julio 2014.

OpenSSL. (2014). *OpenSSL*. Obtenido de OpenSSL: https://www.openssl.org/. Consultado en julio 2014.

OpenSSL. (s.f.). *PKCS12 OpenSSL*. Obtenido de PKCS12 OpenSSL: https://www.openssl.org/docs/apps/pkcs12.html. Consultado en julio 2014.

OPENTEXT. (s.f.). *OPENTEXT*. Obtenido de SlideShare: http://www.opentext.com/2/global/products/enterprise-content-management.htm. Consultado en julio 2014.

Oracle. (2002). *DAO Model Oracle*. Obtenido de DAO Model Oracle: http://www.oracle.com/technetwork/java/dataaccessobject-138824.html. Consultado en julio 2014.

ORACLE. (2010). *docs.oracle.com*. Obtenido de http://docs.oracle.com/cd/E19900-01/820-1800/gbxbk/. Consultado en julio 2014.

Oracle. (1 de Enero de 2012). *JPQL Oracle*. Obtenido de JPQL Oracle: http://docs.oracle.com/html/E24396_01/ejb3_langref.html. Consultado en julio 2014.

Oracle. (Enero de 2012). *JSF Oracle*. Obtenido de JSF Oracle: http://docs.oracle.com/javaee/6/tutorial/doc/bnaph.html. Consultado en julio 2014.

Oracle. (Enero de 2013). *JPA Oracle*. Obtenido de JPA Oracle: http://docs.oracle.com/javaee/6/tutorial/doc/bnbpz.html. Consultado en julio 2014.

Oracle. (2014). *KeyTool Oracle*. Obtenido de KeyTool Oracle: http://docs.oracle.com/javase/tutorial/security/toolsign/wstep2.html. Consultado en julio 2014.

Oracle. (2013 de Enero). *JEE Oracle*. Obtenido de JEE Oracle: http://docs.oracle.com/javaee/6/tutorial/doc/. Consultado en julio 2014.

ORACLE. (s.f.). *netbeans*. Obtenido de netbeans: https://netbeans.org/kb/docs/javaee/ecommerce/entity-session.html. Consultado en julio 2014.

ORACLE. (s.f.). *ORACLE*. Obtenido de ORACLE: http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html. Consultado en julio 2014.

Pencil. (s.f.). *Pencil*. Obtenido de Pencil: http://pencil.evolus.vn/. Consultado en julio 2014.

Primefaces. (22 de Diciembre de 2014). *Primefaces*. Obtenido de Primefaces: http://primefaces.org/. Consultado en julio 2014.

http://javajeecastellano.files.wordpress.com. (15 de 12 de 2013). *javajeecastellano*. Obtenido de javajeecastellano: http://javajeecastellano.files.wordpress.com/2013/12/df92f-hibernate_arquitectura.jpg. Consultado en Julio 2014.

JXLS. (2014). *JXLS*. Obtenido de JXLS: http://jxls.sourceforge.net/. Consultado en julio 2014.

JasperReport. (s.f.). *JasperReport*. Obtenido de JasperReport: http://www.jaspersoft.com/supported-platforms. Consultado en agosto 2014.

JasperSoft. (s.f.). *Ireport JasperSoft*. Obtenido de Ireport JasperSoft: https://community.jaspersoft.com/project/ireport-designer. Consultado en agosto 2014.

UJI. (21 de Enero de 2014). *CryptoApplet UJI*. Obtenido de CryptoApplet UJI:

http://jira.uji.es/confluence/pages/viewpage.action?pageId=26739320. Consultado en agosto 2014.

Microsoft. (s.f.). *Microsoft*. Obtenido de Microsoft: http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf. Consultado en septiembre 2014.

Jcp. (s.f.). *Jcp*. Obtenido de Jcp: https://www.jcp.org/en/jsr/overview. Consultado en diciembre 2014.

LIBROS Y TEXTOS

CONGRESO NACIONAL. (17 de 04 de 2002). Ley De Comercio Electrónico, firmas electrónicas y mensajes de datos. Quito, Pichincha, Ecuador.

Hofacker, A. (2008). *Rapid lean construction - quality rating model.* Manchester: s.n.

Koskela, L. (1992). *Application of the new production philosophy to construction.* Finland: VTT Building Technology.

ANEXOS

ANEXO A
GLOSARIO DE TÉRMINOS

Sigla	Definición
API (Application Programming Interface)	Define un conjunto de procedimientos que pueden ser consumidos por uno o varios sistemas informáticos.
BPM (Business Process Management)	Permite la modelamiento e implementación de procesos ya sea mediante un sistema informático o simplemente en un papel.
CIFS (Como Internet File System)	Es un protocolo que define el acceso remoto a un sistema de archivos mediante varios equipos a la vez sin al necesidad de instalar ningún software
CMIS (Content Management Interoperability Services)	Definido en e RFC 1189, define una capa de abstracción para trabajar con repositorios de contenido utilizando web Services o servicios Rest.
CMS (Content Management System)	Herramienta informática para la gestión de contenido digital
CONATEL (Consejo Nacional de Telecomunicaciones)	Organismo gubernamental encargado de la administración y regulación de las telecomunicaciones en el Ecuador
CryptoApplet	Herramienta informática desarrollada por la Universidad de Jaume para gestionar la firma digital en diferentes formatos
DAM (Digital Asset Management)	Componente de la arquitectura de Alfresco que permite la administración de contenido multimedia como: gráficos, videos y audio
DAO (Data Access	Patrón y componente de software acceso de una aplicación

Object)	con la capa de datos
DM (Document Management)	Componente de la arquitectura de Alfresco que permite la recolección, organización y compartición de documentos.
ECM (Enterprise Content Management)	Es una forma de gestionar y estructurar la información dentro de una organización. Se puede vale de un sistema informático para ejecutarlo.
FTP (File Transfer Protocol)	Protocolo de transferencia de archivos.
INEN (Instituto Ecuatoriano de Normalización)	Ente gubernamental encargado de la generación y administración de normas a nivel del Ecuador.
ISO (International Organización for Standardization)	Ente internacional encargado de la creación y administración de normas para fabricación e productos o servicios en todas las ramas de la industria
J2EE (Java 2 Plataform, Enterprise Edition)	Es un conjunto de tecnologías mantenidas por ORACLE y utilizadas comúnmente para la generación e proyectos empresariales con el lenguaje de programación JAVA.
JCR (Java Content Repository)	Especificación de Java para trabajar con repositorios
JPQL (Java Persistence Query Language)	Lenguaje orientado a la estructuración se sentencias SQL con JAVA
JSF (JAVASERVER FACES)	Es una tecnología desarrollada por SUN y posteriormente adoptada por ORACLE que permite la gestión de ventanas web utilizando el patrón MVC
JSON (JavaScript Object Notación)	Formato de ligero para intercambio de datos. Utilizado principalmente con servicio web de tipo ligeros

JXLS	Librería que permite interactuar a java con documentos en formato XLS
KeyStore	Almacén de llaves que mantiene JAVA
KeyTool	Herramienta propia de JAVA para la generación y gestión de almacenes de llaves.
LDAP (Lightweight Directory Access Protocol)	Protocolo que define el acceso a un servicio de directorio. Generalmente se instala en sistemas operativos de tipo Linux. Muy similar a Active Directory en windows.
MVC (Modelo Visor Controlador)	Es un patrón de arquitectura que define independientemente el modelo(capa de negocios), la vista(interfaz de usuario) y el controlador(controla el flujo entre peticiones web)
Nodo	Es un puntero que utiliza Alfresco para proveer la metadata y la estructura del contenido almacenado
OpenCMIS	Librería que permite interactuar con la especificación CMIS
ORACLE	Compañía multinacional e software que proporciona productos enfocados en: base de datos, sistemas operativos, desarrollo, etc.
PKCS (Public Key Cryptography Standars)	Grupo de estándares que permiten definir la encriptación que llevara la clave pública.
QUIPUX	Herramienta informática desarrollada y administrada por la Sub Secretaría de Gobierno Electrónico del Ecuador. La función principal de este sistema es la generación y seguimiento de documentos legales como son: memorandos, informes, etc.
RestFull Web Services (Representational State	Web Service de naturaleza ligera que utilizan el protocolo JSON

Transfer)	
RM (Record Management)	Componente que permite la gestión legal de los documentos.
RSA (Rivest, Shamir y Adleman)	Es una metodología criptográfica de clave pública que se utiliza para cifrar o firmar documentos digitalmente.
SIGPI (Sistema Gestor de Proyectos de Investigación de la EPN)	Herramienta informática encargada de la gestión y administración de los proyectos de investigación que se desarrollan en al EPN.
SOAP (Simple Object Access Protocol)	Protocolo que permite la comunicación entre dos objetos mediante el intercambio de registros XML
SSO (Single Sign On	Procedimiento de Autenticacion que permite a un usuario mantenerse logueado con sus credenciales a un conjunto de aplicaciones y servicios sin la necesidad de estar ingresándolas por cada uno.
SUPERTEL (Superintendencia de telecomunicaciones)	Ente gubernamental encargado de hacer cumplir normativas y reglamentos vigentes a nivel de ecuador en el sector de las comunicaciones que interactúen con el espectro radioeléctrico
UDDI (Universal Description, Discovery and Integration)	Es utilizado principalmente en web services y su rol es identificar de manera univoca los servicio que este brinda.
UML (Unified Modeling Language)	Lenguaje de modelado unificado para la generación de aplicaciones estructuradas.
UWE (UML Based Web Enginnering)	Define un conjunto de procedimientos para la generación de proyectos web utilizando la metodología UML.
W3C (World Wide Web Consortium)	Consorcio internacional que genera recomendaciones para la web.

Web Content Management (WCM)	Componente de la arquitectura de Alfresco que permite la administración de contenido
Web Services	Conjunto de tecnologías que permiten el intercambio de información en una red de aplicaciones
WebDAV (Web Distributed Authoring and Versioning)	Protocolo que permite la edición, creación y eliminación de contenido a través de la web.
WorkFlow	Abarca el estudio de los procesos que se gestionan en una organización
WSDL (Web Services Description Language)	Documento que define la estructura y parámetros de un web Service
XADES (XML Advanced Electronic Signatures)	Recomendación que permite la firma digital avanzada en documentos XML. Se dice avanzada porque se puede embeber en el documento firmado referencias o los certificados, clave pública u otros datos que se utilizaron para firmar y validar el documento
XML (Extensible Markup Language)	Lenguaje de marcas desarrollado por la W3C para almacenar lenguaje de etiquetado extensible utilizado principalmente para el intercambio de información en aplicaciones web
XMLDSIGN (XML Digital Signature)	Recomendación de la W3C que define la sintaxis que se debería de mantener para firmar documentos en formato XML

ANEXO B
NORMATIVAS

NACIONALES

- NTE INEN 2410 (2011): □DOCUMENTACIÓN. ELABORACIÓN DE OFICIOS, OFICIOS CIRCULARES, MEMORANDOS, MEMORANDOS CIRCULARES Y CIRCULARES. REQUISITOS.
- REGLAMENTO A LA LEY DE COMERCIO ELECTRÓNICO, FIRMAS ELECTRÓNICAS Y MENSAJES DE DATOS□□□□□□□□□□□□□□□

INTERNACIONALES

- ISO-15489: INFORMACIÓN Y DOCUMENTACIÓN. GESTIÓN DE DOCUMENTOS

(CD ADJUNTO)

ANEXO C
MANUAL DE USUARIO

El manual de usuario en formato PDF puede ser encontrado en el CD adjunto.

(CD ADJUNTO)

ANEXO D
MODELOS DE LA CAPA DE PRESENTACIÓN

- Pantalla Inicial - login

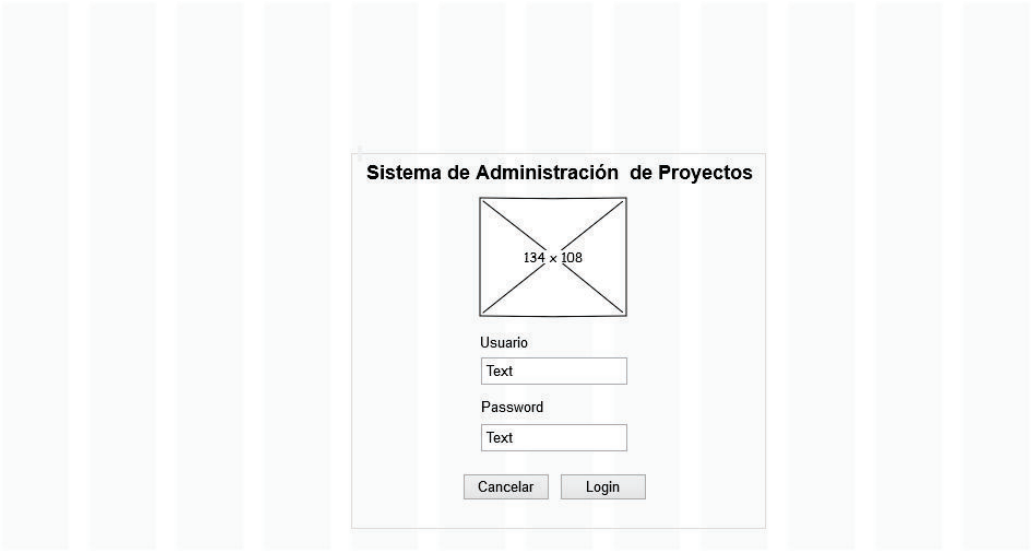


Figura D-1. Vista de login para todos los usuarios

- **USUARIO GENERAL**
  - Inicio

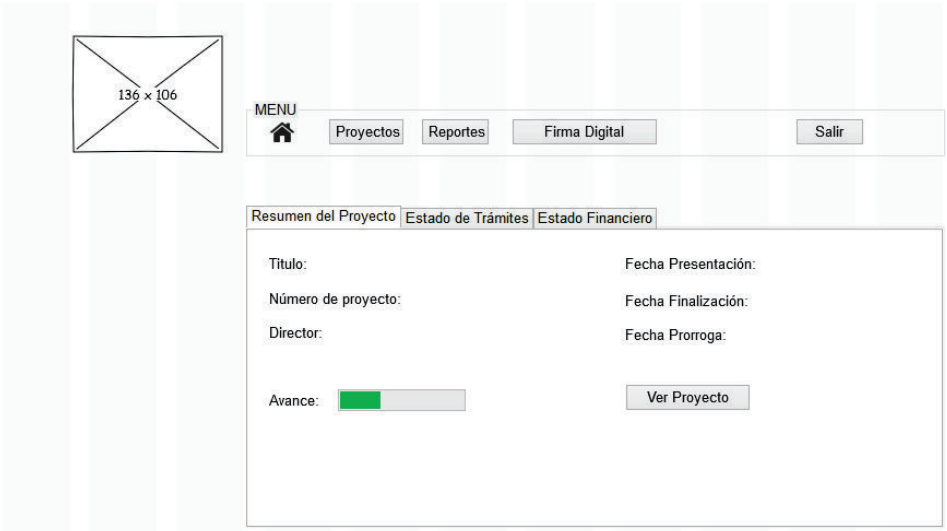


Figura D-2. Vista para el Home

  - Proyectos

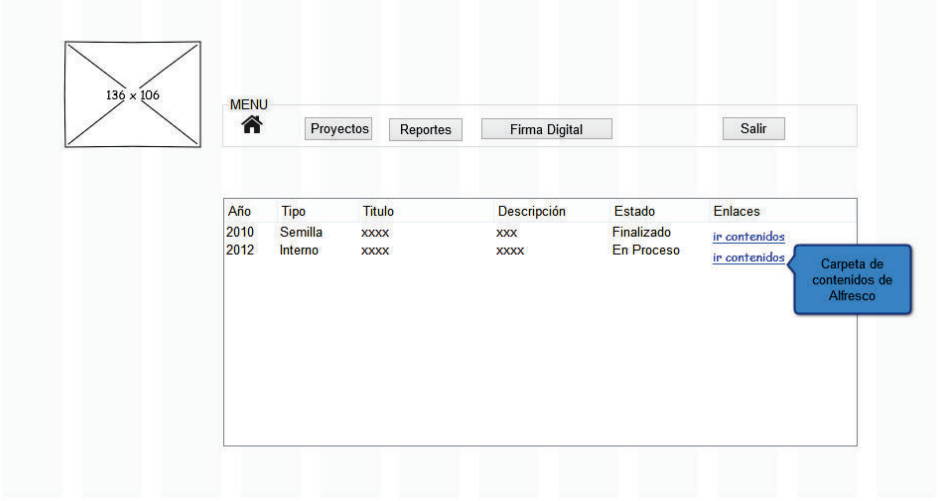


Figura D-3. Vista de proyectos

- Reportería

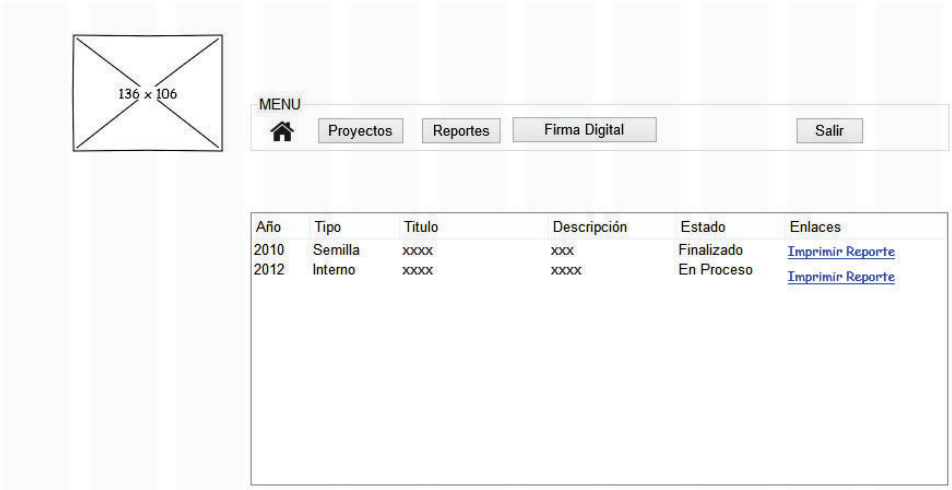


Figura D-4. Vista de reportes

- Firma Digital

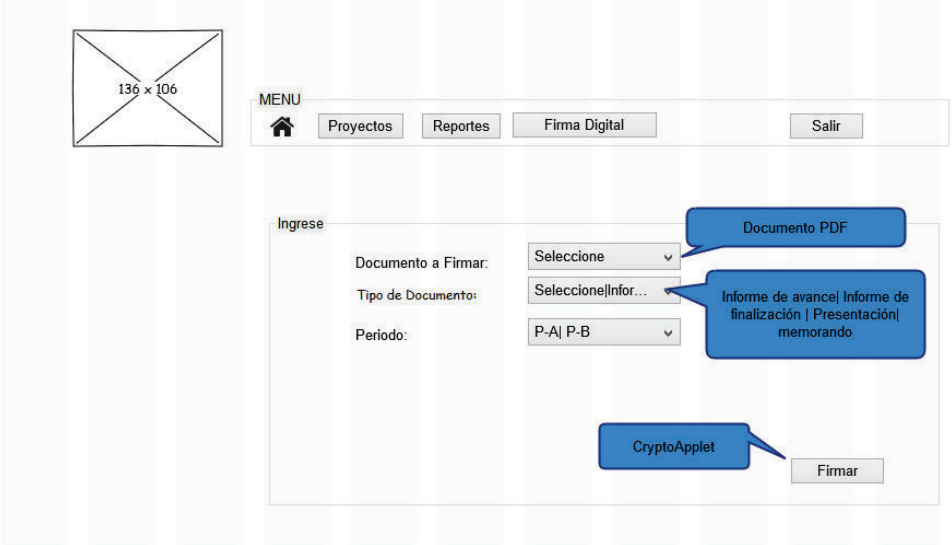


Figura D-5. Vista de firma digital

- **USUARIO COORDINADOR**
    - Inicio

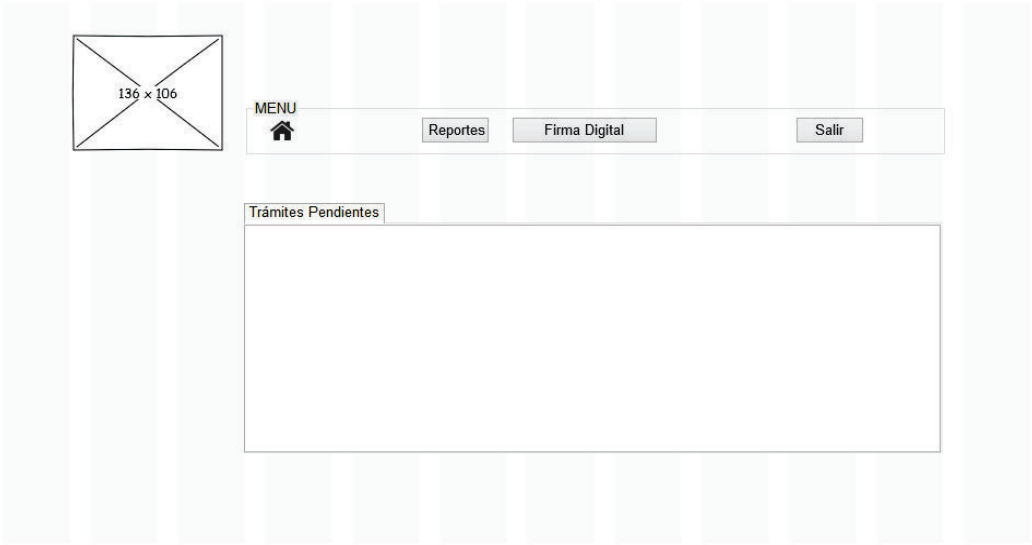


Figura D-1. Vista de Home

    - Reportería

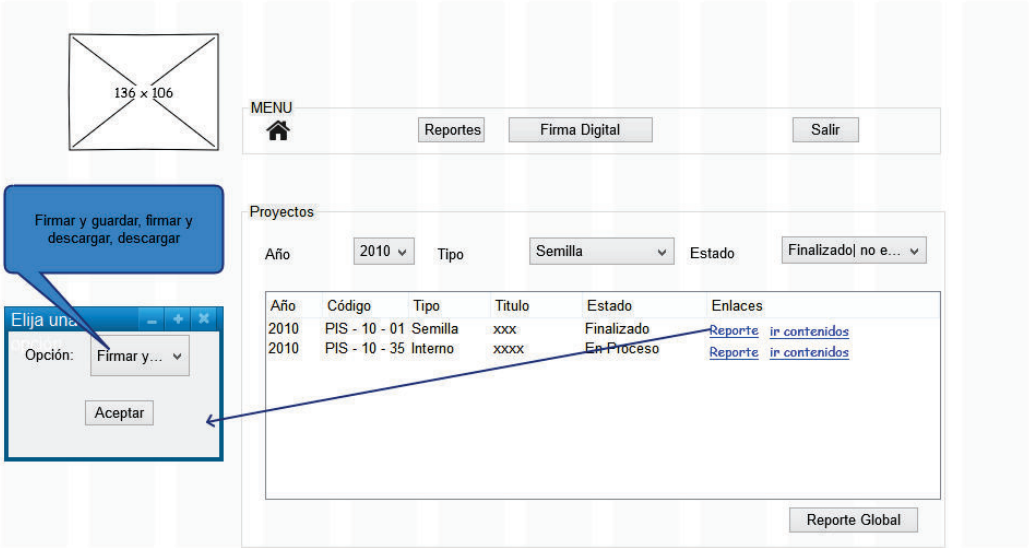


Figura D-2. Vista de reportes

- Firma Digital

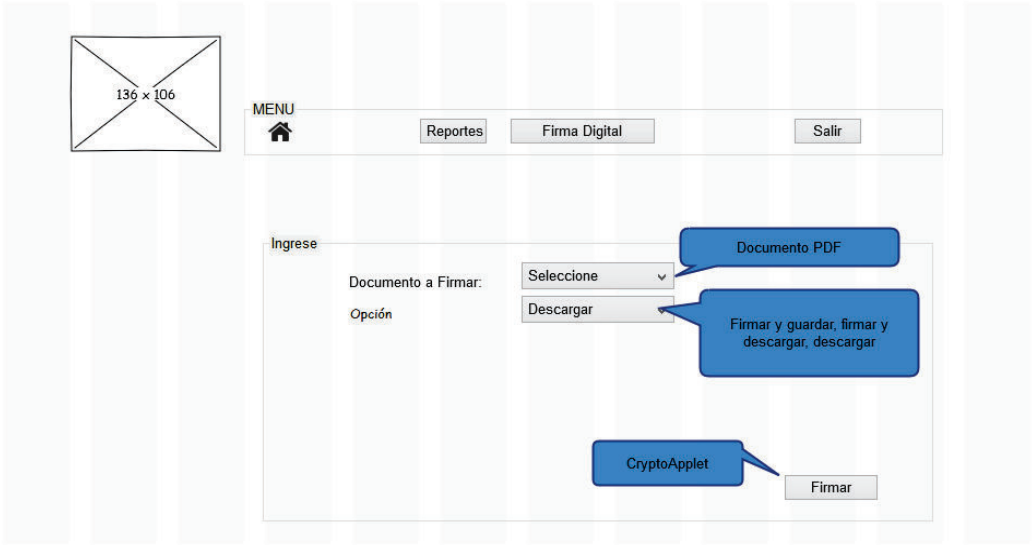


Figura D-3. Vista de firma digital

- **USUARIO ADMINISTRADOR**
  - Inicio

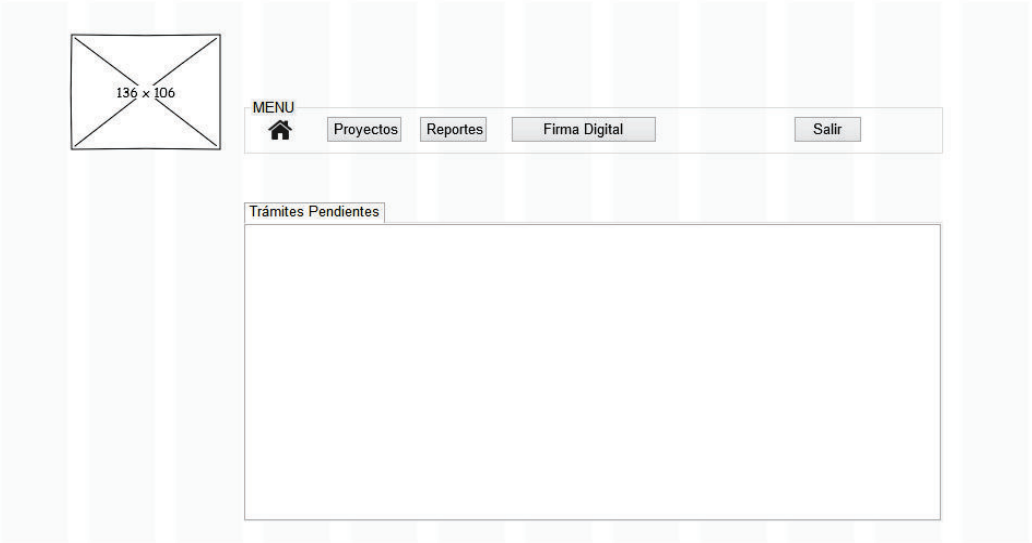


Figura D-4. Vista de Home

- Proyectos

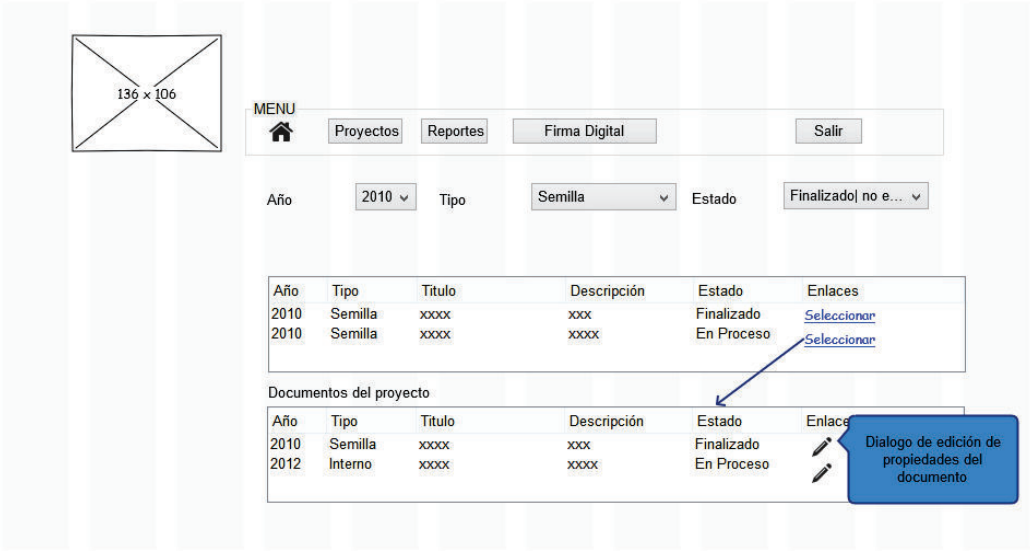


Figura D-5. Vista de proyectos

- Reportería

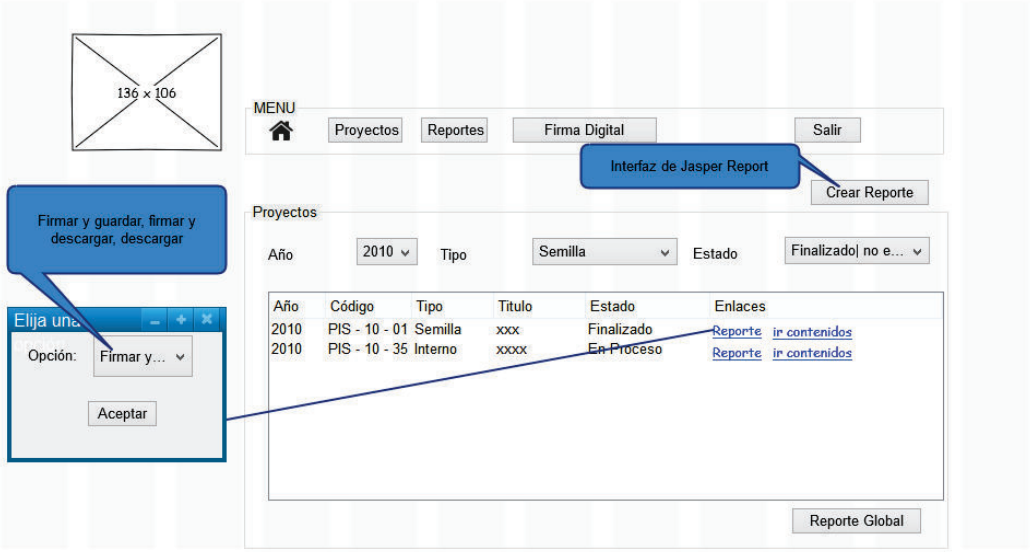


Figura D-6. Vista de reportes

- Firma Digital

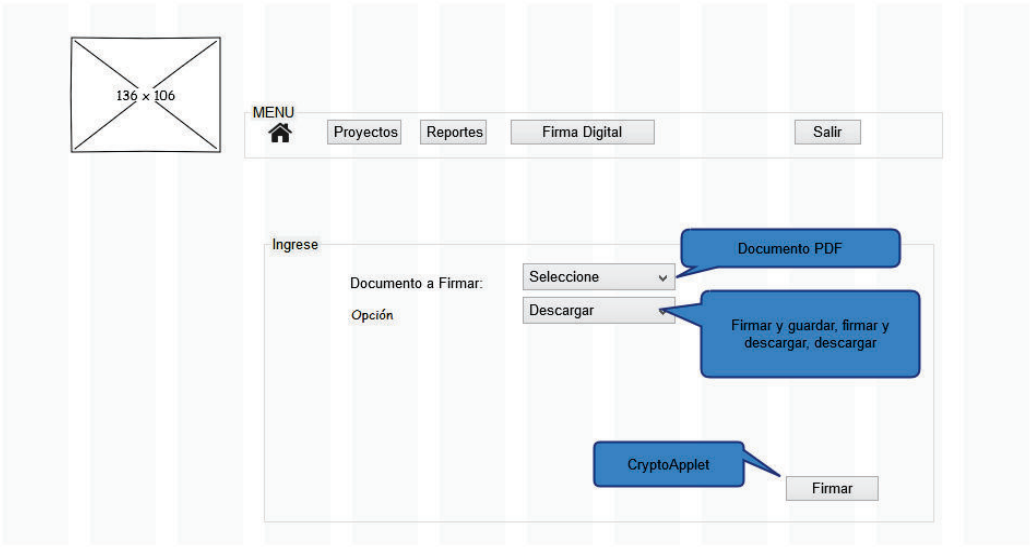


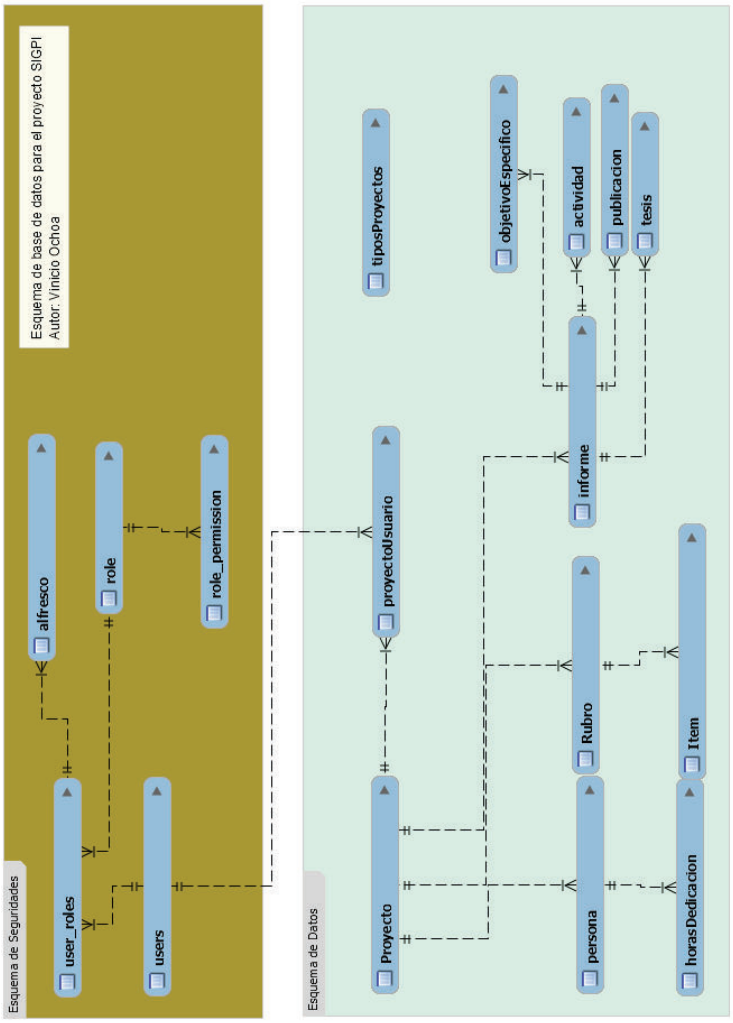
Figura D-7. Vista de firma digital

ANEXO E
CÓDIGO FUENTE DEL SISTEMA

El código fuente del Sistema de Gestión de Proyectos de Investigación – SIGPI, puede encontrarlo en el cd del proyecto.

(CD ADJUNTO)

ANEXO F
ESQUEMA DE LA BASE DE DATOS



El script SQL generado del esquema se adjunta en el proyecto desarrollado.

(CD ADJUNTO)

ANEXO G
CERTIFICADO DE CONFORMIDAD