



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES

**PROYECTO PREVIO LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**“SISTEMA DE ACOPIO DE INFORMACIÓN INTERINSTITUCIONAL Y
REPORTE DE MEDIOS DE COMUNICACIÓN”**

APLICATIVO:

**“DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ACOPIO DE
INFORMACIÓN INTERINSTITUCIONAL Y REPORTE DE MEDIOS DE
COMUNICACIÓN, PARA LA GOBERNACIÓN IMBABURA (MINISTERIO
DEL INTERIOR)”**

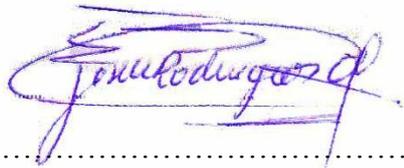
AUTOR: MARLON SILVERIO ARICHÁVALA VALLEJOS

DIRECTOR: ING. JOSÉ LUIS RODRÍGUEZ

**IBARRA-ECUADOR
2013**

CERTIFICACIÓN DIRECTOR

Certifico que la tesis **“Sistema de acopio de información interinstitucional y reporte de medios de comunicación”** con el aplicativo **“desarrollo e implementación de un sistema de acopio de información interinstitucional y reporte de medios de comunicación, en la Gobernación Imbabura”**, ha sido realizada en su totalidad por el egresado: Marlon Silverio Arichávala Vallejos, portador de la cédula de identidad número: 1003436225.



.....
Ing. José Luis Rodríguez
Director de la Tesis



REPÚBLICA DEL ECUADOR
GOBERNACIÓN IMBABURA

El Ecuador ha sido, es y será país Amazónico.



Ministerio del Interior

Ibarra, 14 de Junio de 2013

Señores
UNIVERSIDAD TÉCNICA DEL NORTE
Presente

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del egresado MARLON SILVERIO ARICHÁVALA VALLEJOS con CI: 1003436225, quien desarrolló su trabajo con el tema "**SISTEMA DE ACOPIO DE INFORMACIÓN INTERINSTITUCIONAL Y REPORTE DE MEDIOS DE COMUNICACIÓN**", me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del egresado MARLON SILVERIO ARICHÁVALA VALLEJOS. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra institución.

El Sr. MARLON SILVERIO ARICHÁVALA VALLEJOS puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,

Dr. Marcelo Villamarín Carrascal.
Gobernador
Gobernación Imbabura



062 950 815
062 952 094

gobernador.imbabura@ministeriodelinterior.gob.ec

Ibarra - Ecuador



**UNIVERSIDAD TÉCNICA DEL NORTE
BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR
DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD	1003436225
APELLIDOS Y NOMBRES	ARICHÁVALA VALLEJOS MARLON SILVERIO
DIRECCIÓN	Barrio La Florida, Claveles 3-12 y Amapolas
EMAIL	marlon_arichavala@yahoo.es
TELÉFONO FIJO	06 2631913
TELÉFONO MÓVIL	0986983530

DATOS DEL TRABAJO DE GRADO	
TÍTULO	“DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA DE ACOPIO DE INFORMACIÓN INTERINSTITUCIONAL Y REPORTE DE MEDIOS DE COMUNICACIÓN, PARA LA GOBERNACIÓN IMBABURA (MINISTERIO DEL INTERIOR)”
AUTOR	ARICHÁVALA VALLEJOS MARLON SILVERIO
FECHA	23 DE SEPTIEMBRE DE 2013
PROGRAMA	PREGRADO
TÍTULO	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	ING. JOSÉ LUIS RODRÍGUEZ

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, MARLON SILVERIO ARICHÁVALA VALLEJOS, con cédula de identidad Nro. 1003436225, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 143.



Nombre: ARICHÁVALA VALLEJOS MARLON SILVERIO

Cédula: 1003436225

Ibarra a los 23 días del mes de septiembre de 2013



UNIVERSIDAD TÉCNICA DEL NORTE

CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

Yo, MARLON SILVERIO ARICHÁVALA VALLEJOS, con cedula de identidad Nro. 1003436225, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, articulo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **SISTEMA DE ACOPIO DE INFORMACIÓN INTERINSTITUCIONAL Y REPORTE DE MEDIOS DE COMUNICACIÓN**”, que ha sido desarrollado para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte

Nombre: MARLON SILVERIO ARICHÁVALA VALLEJOS

Cédula: 1003436225

Ibarra a los 23 días del mes de septiembre de 2013

CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrollo sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derecho patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra a los 23 días del mes de septiembre de 2013

A handwritten signature in blue ink, appearing to read 'Marlon', is written on a light-colored rectangular piece of paper. The signature is fluid and cursive.

.....
NOMBRE: MARLON SILVERIO ARICHÁVALA VALLEJOS

CEDULA: 1003436225

DEDICATORIA

A Dios fuente de vida y esperanza inagotables, por concederme el don de la existencia.

A mi Madre Marianita, por todo el apoyo, comprensión y amor que ella siempre brinda incondicionalmente, gracias a ella aprendí el valor de la vida, preparación y sobre todo el deseo de superación y crecimiento tanto profesional como espiritual.

A mi hermano Manolo, gracias por la oportunidad de crecer profesionalmente, pero sobre todo gracias por ser mi hermano y con tu ejemplo enseñarme valores como la bondad, esfuerzo, sacrificio, fuerza y el amor desinteresados.

A mi hermana Diana por su apoyo, consejo y sobre todo gracias por demostrarme que con dedicación y esfuerzo los sueños y anhelos se pueden cumplir.

AGRADECIMIENTO

A mi familia, la mayor bendición y regalo de Dios y la vida para conmigo.

Al Ing. José Luis Rodríguez por su orientación y conocimientos otorgados para el desarrollo del presente proyecto.

A la Gobernación Imbabura por darme la oportunidad, y todas las facilidades para el desarrollo del presente proyecto.

A la Universidad Técnica del Norte, institución de prestigio y reconocida por la calidad en su enseñanza, gracias por darnos la oportunidad de formarnos y aprender en sus Aulas.

A mis profesores, compañeros y amigos por brindarme sus conocimientos consejos y experiencias de vida.

Y a todas aquellas personas, que han formado parte de mi vida universitaria.

RESUMEN

El presente proyecto ha sido realizado gracias a la iniciativa y para solventar las necesidades que tiene la Gobernación Imbabura de automatizar los procesos y el trabajo conjunto entre Instituciones.

El desarrollo del Sistema de acopio de Información Interinstitucional y Reporte de Medios de comunicación “AIRCOM” ha sido realizado con el fin de integrar y automatizar la información generada por las Instituciones pertenecientes a la Gobernación de Imbabura, además la necesidad de registrar lo expuesto por los principales medios de comunicación de la provincia mediante un aplicativo de entorno de web de fácil acceso y manipulación.

La metodología empleada para el análisis, implementación y documentación en el desarrollo del sistema ha sido mediante RUP (RATIONAL UNIFIED PROCESS), la misma que consta de cuatro fases: Inicio, Elaboración, Construcción y Transición permitiendo así, mantener un orden mediante el proceso de desarrollo del proyecto.

Para la implementación y desarrollo del Sistema Web se utilizó postgresSQL como servidor de bases de datos, el framework de desarrollo symfony en la versión 1.4 y el cliente de interface RIA Extjs en la versión 4.1.

Todas las herramientas empleadas en la elaboración e implementación del sistema, son herramientas libres u Open Source y orientadas a la web.

El desarrollo del presente proyecto se lo realizó gracias a la ayuda e información proporcionada por la Gobernación de Imbabura y especialmente por el equipo Político y de Comunicación.

SUMMARY

This project has been realized thanks to the initiative of the "Gobernación Imbabura" that is addressing the needs to automate the communication processes and joint efforts to work together with different institutions that belong to Imbabura's Government.

The development of Information Systems Collection Agency and Media Report "AIRCOM" has been accomplished in order to integrate and automate the information generated by the institutions belonging to the "Gobernación Imbabura". In addition it was realized to record the statement made by the mainstream media in the province through a web application environment that has an easy access and manipulation.

The methodology used for the analysis, implementation and documentation in the development of the system uses RUP (Rational Unified Process). It consists of four phases: Beginning, Elaboration, Construction and Transition. These four steps allow to keep the order through the process of the project development.

For the implementation and development of the Web system was used PostgreSQL as database server, symfony development framework in version 1.4 and RIA client Extjs interface in version 4.1.

All the tools used in the development and implementation of the system are free or open source and they are web oriented.

The development of this project was made with the help and information provided by the "Gobernación Imbabura" especially by the Political and Communication Team.

ÍNDICE

CERTIFICACIÓN DIRECTOR.....	II
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN.....	VI
CONSTANCIAS.....	VII
DEDICATORIA	VIII
AGRADECIMIENTO	IX
RESUMEN	X
ÍNDICE.....	XII
ÍNDICE DE ILUSTRACIONES.....	XIX
ÍNDICE DE TABLAS	XXI
1 INTRODUCCIÓN	2
1.1 ANTECEDENTES	2
1.2 SITUACIÓN ACTUAL	6
1.3 EL PROBLEMA:	7
1.4 OBJETIVOS DEL PROYECTO	7
1.4.1 Objetivo General.....	7
1.4.2 Objetivos Específicos	8
1.5 ALCANCE.....	8
1.6 JUSTIFICACIÓN.....	9
2 MARCO TEÓRICO.....	13
2.1 SOFTWARE LIBRE Y ESTÁNDARES UTILIZADOS.....	13
2.1.1 SOFTWARE LIBRE	13
2.1.2 HTML(<i>HyperTextMarkupLanguage</i>)	13
2.1.3 CSS.....	13
2.1.4 SERVIDOR WEB	14
2.1.5 PHP.....	14
2.2 SERVIDOR DE APLICACIONES APACHE.....	14
2.2.1 Introducción a Apache.....	14
2.2.2 Estructura y funcionamiento del servidor Apache	14
2.2.3 Publicación de aplicaciones Symfony PHP.....	16
2.3 BASE DE DATOS POSTGRESQL	17
2.3.1 Introducción a PostgreSQL.....	17
2.3.2 Arquitectura de Postgres	18
2.4 FRAMEWORK SYMFONY.....	19
2.4.1 Introducción a Symfony	19
2.4.2 El entorno de desarrollo de Symfony	20
2.4.3 Arquitectura de Symfony	21
2.4.4 Plugins Symfony 1.4	23
2.5 EXTJS 4.1	24
2.5.1 Introducción a Extjs	24
2.5.1.1 RIA	25

2.5.1.2	JAVA SCRIPT	25
2.5.1.3	AJAX.....	26
2.5.1.4	El objeto XMLHttpRequest.....	27
2.5.1.5	JSON	27
2.5.2	ExtJs es Asíncrono	28
2.5.3	Funcionalidades.....	28
2.5.4	Ventajas.....	29
2.5.5	Desventajas	30
3	DESCRIPCIÓN Y FUNCIONAMIENTO DE LOS MÓDULOS	32
3.1	Funcionamiento del sistema	32
3.1.1	Vista general.....	32
3.1.2	Características del sistema	33
3.1.3	Dependencias para la Implementación.....	33
3.1.3.1	Costos.....	33
3.1.3.2	Licenciamiento	34
3.1.3.3	Instalación	34
3.1.4	Funcionamiento	34
3.2	Descripción y Funcionamiento de los Módulos	35
3.2.1	Descripción del módulo de Administración	35
3.2.2	Funcionamiento del Módulo de Administración del Sistema	36
3.2.3	Descripción del Módulo de Auditoría.....	37
3.2.4	Funcionamiento del Módulo de Auditoría.	37
3.2.5	Descripción del Módulo de Recursos e Infraestructura de la Institución.	37
3.2.6	Funcionamiento del Módulo de Recursos e Infraestructura de la Institución..	37
3.2.7	Descripción del Módulo de Localizaciones Geográficas.....	38
3.2.8	Funcionamiento del módulo de Localización Geográfica.....	39
3.2.9	Descripción del Módulo de Instituciones y Actores Institucionales.....	39
3.2.10	Funcionamiento del Módulo de Instituciones y Actores Institucionales.	39
3.2.11	Descripción del Módulo de Información y Medios de Comunicación.....	40
3.2.12	Funcionamiento del Módulo de Información y Medios de Comunicación.	40
3.2.13	Descripción del Módulo de Reportes e Informes.....	42
3.2.14	Funcionamiento del Módulo de Reportes e Informes.	42
4	DISEÑO Y DESARROLLO DEL APLICATIVO.....	44
4.1	Introducción.....	44
4.2	FASE DE INCEPCIÓN	45
4.2.1	Documento de Visión	45
4.2.1.1	Introducción.....	45
4.2.1.1.1	Propósito.....	45
4.2.1.1.2	Alcance	45
4.2.1.1.3	Definiciones, Siglas y Abreviaturas	45
4.2.1.1.4	Referencias.....	45
4.2.1.2	Posicionamiento.....	46

4.2.1.2.1	Oportunidad de negocio.....	46
4.2.1.2.2	Definición del problema	47
4.2.1.3	Descripción de los interesados y usuarios.....	48
4.2.1.3.1	Resumen de los interesados.....	48
4.2.1.3.2	Resumen de los usuarios	49
4.2.1.3.3	Entorno actual de los usuarios	50
4.2.1.3.3.1	Coordinador de Proyecto	51
4.2.1.3.3.2	Responsable del Proyecto	52
4.2.1.3.3.3	Responsable Funcional	52
4.2.1.3.4	Perfiles de usuario.....	53
4.2.1.3.4.1	Administrador del Sistema.....	53
4.2.1.3.4.2	Administrador Funcional del Sistema.....	54
4.2.1.3.4.3	Usuario del Sistema	54
4.2.1.3.5	Necesidades de los interesados y Usuarios.....	55
4.2.1.3.6	Alternativas y Competencias	56
4.2.1.4	Vista General del Producto.....	56
4.2.1.4.1	Perspectiva del Producto.....	57
4.2.1.4.2	Resumen de capacidades	57
4.2.1.4.3	Suposiciones y Dependencias.....	58
4.2.1.4.4	Costos y Precios	58
4.2.1.4.5	Licenciamiento e Instalación	59
4.2.1.5	Características del producto.....	60
4.2.1.5.1	Facilidad de acceso y Uso	60
4.2.1.5.2	Unificación de la Información.....	60
4.2.1.5.3	Conexión Fácil y Dinámica	60
4.2.1.5.4	Base de datos Centralizada	60
4.2.1.6	Rangos de Calidad	60
4.2.1.7	Precedencia y Prioridad.....	61
4.2.1.8	Otros requerimientos del Producto	61
4.2.1.8.1	Sistema Multiplataforma.....	61
4.2.1.8.2	Acceso a Internet.....	61
4.2.1.8.3	Servidor Web	61
4.2.1.8.4	Sistemas Operativos	61
4.2.1.8.5	Protocolo de Comunicaciones.....	61
4.2.1.8.6	Plataforma de Redes	61
4.2.2	Plan de Desarrollo de Software	61
4.2.2.1	Introducción	62
4.2.2.1.1	Propósito	62
4.2.2.1.2	Alcance	62
4.2.2.2	Vista General del Proyecto.....	63
4.2.2.2.1	Propósito, Alcance y Objetivos.....	63
4.2.2.2.2	Suposiciones y restricciones.....	65

4.2.2.2.3	Entregables del Proyecto	65
4.2.2.2.4	Evolución del Plan de Desarrollo de Software	67
4.2.2.3	Organización del Proyecto	67
4.2.2.3.1	Participantes del Proyecto	67
4.2.2.3.2	Interfaces Externas.....	68
4.2.2.3.3	Roles y Responsabilidades	68
4.2.2.4	Gestión del Proceso	68
4.2.2.4.1	Plan del Proyecto	68
4.2.2.4.2	Plan de las Fases.....	69
4.2.2.4.2.1	Objetivos de las Iteraciones	70
4.2.2.4.2.2	Calendario del Proyecto.....	72
4.2.2.4.3	Seguimiento y control del Proyecto.....	75
4.2.2.4.3.1	Gestión de Requisitos	75
4.2.2.4.3.2	Control de Plazos.....	75
4.2.2.4.3.3	Control de Calidad.....	75
4.2.2.4.3.4	Gestión de Riesgos	75
4.2.2.4.3.5	Gestión de Configuración.....	75
4.2.2.5	Referencias.....	76
4.2.3	Actas de Trabajo	76
4.3	FASE DE ELABORACIÓN	76
4.3.1	Arquitectura	76
4.3.1.1	Introducción.....	76
4.3.1.1.1	Propósito.....	76
4.3.1.1.2	Alcance	76
4.3.1.2	Representación de la Arquitectura	77
4.3.1.3	Objetivos y Restricciones de la Arquitectura	77
4.3.1.4	Vista de Casos de Uso	77
4.3.1.4.1	Procesos	77
4.3.1.4.1.1	Proceso Modulo de Administración y Seguridad	77
4.3.1.4.1.2	Proceso Modulo de Auditoria	78
4.3.1.4.1.3	Módulo de Recursos e Infraestructura de la Institución.....	79
4.3.1.4.1.4	Módulo de Información y Medios de Comunicación:.....	79
4.3.1.4.1.5	Módulo de Instituciones y Actores Institucionales	80
4.3.1.4.1.6	Módulo de Localización Geográfica	80
4.3.1.4.1.7	Módulo de Reportes e Informes.	81
4.3.1.4.2	Actores	81
4.3.1.4.3	Modelos de Casos de Negocio	82
4.3.1.4.4	Modelos de Casos de Uso	83
4.3.1.4.4.1	Caso de Uso Administración del Sistema	83
4.3.1.4.4.2	Caso de Uso Auditoria de Sistema	84
4.3.1.4.4.3	Caso de Uso Periodista.....	85
4.3.1.4.4.4	Caso de Uso Usuario Representante Institucional (Cliente)	86

4.3.1.4.5	Especificación de casos de Uso	87
4.3.1.4.5.1	Asignación de Usuarios al Sistema	88
4.3.1.4.5.2	Registrar Responsable Institucional	89
4.3.1.4.5.3	Registrar Institución	91
4.3.1.4.5.4	Enviar Proyecto.....	92
4.3.1.4.5.5	Crear una Etapa de Proyecto.....	94
4.3.1.4.5.6	Registrar un Medio de Comunicación	96
4.3.1.4.5.7	Registrar un Programa.....	97
4.3.1.4.5.8	Registrar un Periodista	99
4.3.1.4.5.9	Crear un Reporte de Prensa	101
4.3.1.4.5.10	Reporte de Apreciaciones Negativas de Prensa	102
4.3.1.4.5.11	Reporte de Apreciaciones Positivas de Prensa.....	104
4.3.1.4.5.12	Crear un reporte de Programa	105
4.3.1.4.5.13	Reporte de Apreciaciones Negativas de Programa	107
4.3.1.4.5.14	Reporte de Apreciaciones Positivas de Programa.....	108
4.3.1.4.5.15	Registro de productos para Inventario.....	109
4.3.1.4.5.16	Reporte de Auditoría	111
4.3.2	Vista Dinámica	112
4.3.2.1	Introducción	112
4.3.2.2	Diagramas de Secuencia.....	112
4.3.3	Vista Lógica	113
4.3.3.1	Introducción	113
4.3.3.2	Descomposición de Subsistemas.....	114
4.3.4	Vista de Despliegue.....	114
4.3.4.1	Servidor de Aplicaciones	115
4.3.4.2	Computadores Internos	115
4.3.4.3	Computadores Externos.....	115
4.3.5	Diseño en Capas.....	116
4.3.5.1	Capa Interfaz de Usuario	116
4.3.5.2	Capa Lógica del Negocio.....	117
4.3.6	Vista de Datos	117
4.3.6.1	Modelo Relacional.....	117
4.3.6.2	Modelo Físico	117
4.3.7	Tamaño y Desempeño	117
4.3.7.1	Tiempo de respuesta en el acceso a la base de datos	117
4.3.7.2	Tiempo de respuesta de transacciones.....	117
4.3.7.3	Espacio en disco para Servidor de Base de Datos	118
4.3.8	Calidad	118
4.3.8.1	Usabilidad	118
4.3.8.2	Eficiencia.....	118
4.3.8.3	Seguridad.....	118
4.3.8.4	Confiabilidad.....	118

4.3.8.5	Mantenimiento	118
4.3.9	Arquitectura del Proyecto	119
4.3.9.1	Introducción	119
4.3.9.2	Paquetes de Análisis	119
4.3.10	Lista de Riesgos	124
4.3.10.1	Introducción	124
4.3.10.1.1	Propósito	124
4.3.10.1.2	Alcance	124
4.3.10.1.3	Revisión General	124
4.3.10.2	Riesgos	124
4.3.10.2.1	El cliente EXT JS y framework symfony1.4 son herramientas poco conocidas por el desarrollador	124
4.3.10.2.2	La Gobernación Imbabura no cuenta con Servidores y equipos tecnológicos donde instalar la Aplicación.....	125
4.3.10.2.3	Computador de acceso es atacado por virus.....	126
4.3.10.2.4	Modelo Vista controlador utilizado por Symfony no se acople a la metodología RUP.	127
4.3.10.2.5	Incompatibilidad con navegadores de internet y configuración en los clientes de acceso a la aplicación.....	128
4.3.10.2.6	Desconocer los procesos manejados por la Gobernación de Imbabura.	129
4.3.10.2.7	Resumen de Riesgos.....	130
4.3.11	Estrategia de Pruebas.....	131
4.3.11.1	Pruebas de Integridad de datos.....	131
4.3.11.2	Pruebas del Sistema	131
4.3.11.3	Pruebas del ciclo de Negocio.	132
4.3.11.4	Pruebas de Interfaz de Usuario.....	133
4.3.11.5	Pruebas de Desempeño.	133
4.3.11.6	Pruebas de Seguridad y Control de Acceso.	134
4.3.11.7	Herramientas	135
4.3.11.8	Recursos.....	135
4.3.12	Prototipo de Interfaz de Usuario.....	136
4.3.13	Interfaces de Usuario	136
4.3.13.1	Inicio.....	136
4.3.13.2	Menú Principal Backend (Administrador, Periodista, Auditor).....	136
4.3.13.3	Menú principal Frontend (Cliente).....	137
4.3.13.4	Ejemplo de Edición, Formulario de Ingreso de Proyectos	137
4.3.13.4.1	Sección de datos proyecto.	137
4.3.13.4.2	Sección de ubicación del proyecto.....	138
4.3.13.4.3	Sección de asignación de objetivos.....	138
4.3.14	Estándares de Programación.....	139
4.3.14.1	Introducción.....	139

4.3.14.1.1	Propósito	139
4.3.14.1.2	Descripción	139
4.3.14.2	Estandarización de la Base de Datos	139
4.3.14.2.1	Normas a Aplicarse	139
4.3.14.3	Estandarización del Lenguaje de Programación.....	140
4.3.14.4	Resumen de Tecnologías.....	140
4.4	FASE DE CONSTRUCCIÓN.....	142
4.4.1	Modelo de Datos	142
4.4.1.1	Introducción	142
4.4.1.1.1	Propósito	142
4.4.1.1.2	Alcance	142
4.4.1.1.3	Definición de Siglas y Abreviaturas	142
4.4.1.1.4	Referencias	143
4.4.1.2	Modelo de Datos	143
4.5	FASE DE TRANSICIÓN.....	144
4.5.1	Manual de Instalación.....	144
4.5.2	Introducción.....	144
4.5.2.1	Copiar los archivos necesarios para la instalación	144
4.5.2.2	Instalación y Configuración de herramientas.....	144
4.5.2.2.1	Apache	144
4.5.2.2.2	PHP	145
4.5.2.2.3	PostgresSql	147
4.5.2.2.4	NetBeans	148
4.5.2.2.5	Framework Symfony 1.4.....	153
4.5.2.2.6	ExtJs 4.1	156
4.5.2.2.7	PgAdmin III.....	157
4.5.2.2.8	PowerArchitect.....	158
5	CONCLUSIONES Y RECOMENDACIONES	161
5.1	Conclusiones.....	161
5.2	Recomendaciones	162
5.3	GLOSARIO	163
5.3.1	Introducción.....	163
5.3.2	Propósito.....	163
5.3.3	Referencias	163
5.3.4	Glosario	163
6	REFERENCIAS BIBLIOGRÁFICAS.	166
7	ANEXOS.....	167

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Modelo Vista Controlador.....	10
Ilustración 2: Fases de la Metodología RUP.....	11
Ilustración 3: Estructura del servidor Apache2.....	15
Ilustración 4: Configuración de la aplicación Symfony 1.4 en Apache.	16
Ilustración 5: Logo de la BDD PostgreSQL.....	17
Ilustración 6: Arquitectura de la bdd Postgres	19
Ilustración 7: Logo del Framework Symfony.....	19
Ilustración 8: Entorno de Producción Symfony1.4	20
Ilustración 9: El patrón MVC.	21
Ilustración 10: Estructura del proyecto Symfony 1.4.....	22
Ilustración 11: Formulario de ingreso de sfDoctrineGuardPlugin.	24
Ilustración 12: Logo Extjs.	24
Ilustración 13: Funcionamiento Ajax.	27
Ilustración 14: Web2.0.....	28
Ilustración 15: Entorno (Formulario Extjs)	29
Ilustración 16: Prospectiva del Sistema.	32
Ilustración 17: Vista General del Sistema.	35
Ilustración 18: Prospectiva del Producto.	57
Ilustración 19: Calendario de las principales tareas del proyecto.	72
Ilustración 20: Procesos del Módulo Administración y Seguridad.	78
Ilustración 21: Proceso Modulo de Auditoria.	78
Ilustración 22: Módulo de Recursos e Infraestructura de la Institución.	79
Ilustración 23: Módulo de Información y Medios de Comunicación.	79
Ilustración 24: Módulo de Instituciones y Actores Institucionales.....	80
Ilustración 25: Módulo de Localización Geográfica.	80
Ilustración 26: Módulo de Reportes e Informes.	81
Ilustración 27: Actores.	81
Ilustración 28: Modelos de Casos de Negocio.	82
Ilustración 29: Casos de Uso del Módulo de Administración y Seguridad.	83
Ilustración 30: Casos de uso del módulo Auditoría Informática.....	84
Ilustración 31: Caso de Uso Periodista.....	85
Ilustración 32: Diagramas de Secuencia.	113
Ilustración 33: Vista de Despliegue.....	115
Ilustración 34: Diseño en Capas.	116
Ilustración 35: Paquetes de Análisis.	123
Ilustración 36: Formulario de Inicio.	136
Ilustración 37: Menú Principal Backend.	136
Ilustración 38: Menú principal Frontend.	137
Ilustración 39: Sección de Ingreso de proyecto.	137
Ilustración 40: Sección de Ubicación del Proyecto.	138

Ilustración 41: Sección de asignación de objetivos.....	138
Ilustración 42: Instalación de Apache.	145
Ilustración 43: Instalación de PHP.....	146
Ilustración 44: Instalación NetBeans.....	149
Ilustración 45: Licencia NetBeans.....	149
Ilustración 46: Ubicación JDK.	150
Ilustración 47: Ubicación IDE NetBeans.	150
Ilustración 48: Iniciar el proceso de instalación NetBeans.....	151
Ilustración 49: Progreso de Instalación NetBeans.....	151
Ilustración 50: Finalización de Instalación NetBeans.	152
Ilustración 51: Entorno NetBeans	152
Ilustración 52: Instalación Symfony 1.4	154
Ilustración 53: Configuración de Symfony con NetBeans	155
Ilustración 54: Integración Symfony con NetBeans.	156
Ilustración 55: Integración ExtJs con Symfony	157
Ilustración 56: Entorno PgAdminIII.	158
Ilustración 57: Iniciar PowerArchitect.....	158
Ilustración 58: Entorno Power Architect.....	159
Ilustración 59: Configuración Power Architect.	159

ÍNDICE DE TABLAS

Tabla 1: Herramientas de Desarrollo	10
Tabla 2: Libertades del Software Libre	13
Tabla 3: Limitaciones de PostgreSQL	18
Tabla 4: Características del Sistema.....	33
Tabla 5: Funcionamiento del Módulo de Administración.....	36
Tabla 6: Funcionamiento del Módulo de Auditoria.....	37
Tabla 7: Funcionamiento del Módulo de Recursos e Infraestructura de la Institución.....	38
Tabla 8: Funcionamiento del módulo de Localización Geográfica	39
Tabla 9: Funcionamiento del Módulo de Instituciones y Actores Institucionales.....	40
Tabla 10: Funcionamiento del Módulo de Información y Medios de Comunicación.....	42
Tabla 11: Funcionamiento del Módulo de Reportes e Informes.....	42
Tabla 12: Definición del Problema	48
Tabla 13: Resumen de Interesados.....	49
Tabla 14: Resumen de los Usuarios.....	50
Tabla 15: Perfil de Coordinador del Proyecto.....	52
Tabla 16: Perfil Responsable del Proyecto.....	52
Tabla 17: Perfil responsable Funcional.....	53
Tabla 18: Perfil Administrador del Sistema.....	53
Tabla 19: Perfil Administrador Funcional del Sistema	54
Tabla 20: Perfil del Usuario del Sistema.....	54
Tabla 21: Necesidades de los interesados y usuarios.....	56
Tabla 22: Resumen de capacidades.....	58
Tabla 23: Costos del Proyecto.....	59
Tabla 24: Roles y Responsabilidades.....	68
Tabla 25: Planificación de fases.....	69
Tabla 26: Hitos Finales de cada Fase.....	70
Tabla 27: Objetivos de las Iteraciones.....	71
Tabla 28: Calendario Fase de Inicio.....	73
Tabla 29: Calendario Fase de Elaboración.....	74
Tabla 30: Descripción de Clientes.....	82
Tabla 31: Caso de Uso Administración del Sistema.....	84
Tabla 32: Caso de Uso Auditoria de Sistema.....	85
Tabla 33: Caso de Uso Periodista.....	86
Tabla 34: Casos de uso Usuario Cliente (Responsable Institucional).....	87
Tabla 35: Casos de uso Usuario Cliente (Responsable Institucional).....	87
Tabla 36: Perfil del Usuario del Sistema.....	89
Tabla 37: Registrar Responsable Institucional.....	90
Tabla 38: Registrar Institución	92
Tabla 39: Enviar Proyecto	94
Tabla 40: Crear una Etapa de Proyecto.....	95

Tabla 41: Registrar un Medio de Comunicación	97
Tabla 42: Registrar un Programa.....	99
Tabla 43: Registrar un Periodista.....	100
Tabla 44: Crear un Reporte de Prensa.....	102
Tabla 45: Reporte de Apreciaciones Negativas de Prensa.....	103
Tabla 46: Reporte de Apreciaciones Positivas de Prensa.....	105
Tabla 47: Crear un Reporte de Programa.....	107
Tabla 48: Reporte de Apreciaciones Negativas de Programa.....	108
Tabla 49: Reporte de Apreciaciones Positivas de Programa.....	109
Tabla 50: Registro de productos para Inventario.....	111
Tabla 51: Reporte de Auditoria.....	112
Tabla 52: Resume de Riesgos.....	130
Tabla 53: Pruebas de Integridad de Datos.....	131
Tabla 54: Pruebas del Sistema.....	131
Tabla 55: Pruebas del Ciclo de Negocio.....	132
Tabla 56: Pruebas de Interfaz de Usuario.....	133
Tabla 57: Pruebas de Desempeño.....	134
Tabla 58: Pruebas de Seguridad y Control de Acceso.....	135
Tabla 59: Herramientas.....	135
Tabla 60: Recursos.....	135
Tabla 61: Tecnologías empleadas.....	142

Capítulo 1: Introducción



RESUMEN DEL CAPÍTULO

- *Antecedentes.*
- *Situación actual.*
- *El Problema.*
- *Objetivos del Proyecto.*

1 INTRODUCCIÓN

1.1 ANTECEDENTES

Las instituciones políticas son los sujetos de creación de información sumamente relevante en la sociedad y la comunicación es indispensable para que estas se vuelvan un vínculo de interacción social. La Gobernación de Imbabura como Institución política y pública, perteneciente al Ministerio del Interior, recibe y genera información de un grado de importancia sumamente alto, la cual está destinada a cumplir metas de desarrollo y mejoramiento en la calidad de vida de la población.

La **Gobernación de Imbabura** trabaja en conjunto con diversas Instituciones Nacionales y Provinciales, Cantonales e incluso Parroquiales, para el logro de sus objetivos, tiene como misión, Direccionar y orientar la política del Gobierno Nacional en la provincia, los planes y proyectos promovidos por el Ministerio del Interior a nivel provincial, a través de una gestión eficiente, eficaz, efectiva, transparente y pública, para el fortalecimiento de la gobernabilidad y la seguridad interna, en su gran mayoría buscando cumplir con los objetivos del Plan Nacional del Buen Vivir, para este trabajo conjunto es necesaria la articulación de los procesos entre Instituciones para lo cual se hace uso de todo tipo de recursos económicos, humanos y tecnológicos.

GOBERNACIÓN DE IMBABURA

Misión:

Direccionar y orientar la política del Gobierno Nacional en la provincia, los planes y proyectos promovidos por el Ministerio del Interior a nivel provincial, a través de una gestión eficiente, eficaz, efectiva, transparente y pública, para el fortalecimiento.

Atribuciones y Responsabilidades:

1. Dirigir y direccionar la aplicación de la política pública sobre gobernabilidad y seguridad interna a nivel provincial.
2. Asegurar la implementación de estrategias de desarrollo provincial del Gobierno Nacional.
3. Supervisar el correcto desempeño de las unidades administrativas dependientes de la Gobernación.

4. Controlar el buen manejo de los bienes de dominio público y la conservación de los edificios destinados a funcionamiento de los establecimientos públicos.
5. Supervisar que los funcionarios y empleados públicos desempeñen cumplidamente sus deberes.
6. Diseñar mecanismos e implementación para la ejecución de las políticas nacionales y locales; emanadas del Gobierno Nacional.
7. Informar sobre el seguimiento y evaluación de la implementación de las políticas en la provincia.
8. Presidir el Gabinete Provincial.
9. Presidir el Consejo Provincial de Seguridad Interna.
10. Presidir el Comité de Operaciones de Emergencias en la provincia.
11. Nombrar bajo su responsabilidad a los Jefes Políticos, Intendente y Subintendente de Policía, Comisarios y Tenientes Políticos.
12. Supervisar las instituciones de derecho privado que reciban permanente apoyo económico del Estado.
13. Disponer, en los casos de declaración de estado de emergencia nacional que los recaudadores de impuestos y contribuciones entreguen a los pagadores de la provincia las sumas correspondientes, sin perjuicio del control posterior que ejercerá la Contraloría General del Estado.
14. Formular mecanismos de coordinación con el régimen seccional autónomo, dependiente y organizaciones sociales.
15. Delegar y articular planes de concertación social y propender a nuevas formas de intervención política.
16. Establecer el diálogo y la concertación para la solución de conflictos en el ámbito de su competencia y localidad.
17. Controlar y gestionar por la tranquilidad y el orden público exigiendo para ello el auxilio de la fuerza pública, garantizar la seguridad de las personas y de los bienes; prevenir delitos y combatir la delincuencia.
18. Gestionar la planificación del desarrollo provincial a través de los órganos del Régimen Provincial Dependiente.
19. Aprobar los instructivos e impartir órdenes necesarias para proteger el medio ambiente en los casos de emergencia.

20. Presentar anualmente al/la Ministro/a del Interior, un plan de trabajo, con el respaldo de la Comisión Ejecutiva Provincial, hasta el 30 de septiembre para el año siguiente en el cual contemplen las soluciones a los problemas de la provincia.
21. Las demás atribuciones y deberes que le fueren asignados por la Constitución, leyes y reglamentos vigentes, así como las políticas emitidas por el/la Ministro/a del Interior o su delegado.

Estructura Básica: A continuación se muestra las principales Instituciones que tienen responsabilidades con la Gobernación de Imbabura.

DIRECCIÓN DE GESTIÓN POLÍTICA Y MANEJO DE CONFLICTOS

Misión: Implementar y evaluar directrices para que, a través de los procesos de diálogo concertados con la sociedad, fortalezcan la ejecución de la política de gobernabilidad en la provincia.

UNIDAD DE GESTIÓN POLÍTICA Y MANEJO DE CONFLICTOS

Misión: Ejecutar e instrumentar políticas, planes, programas, proyectos y acciones en la provincia que mejoren la seguridad ciudadana, prevengan el delito y la violencia y protejan la población, fomentando su participación, control y cooperación en coordinación con los diferentes órganos e instituciones de la provincia.

JEFATURAS POLÍTICAS

Misión: Representar al ejecutivo provincial, controlar el accionar administrativo de la organización pública cantonal para el eficiente cumplimiento de los objetivos institucionales, además articular a la Gobernación con los niveles de Gobierno y la ciudadanía en el ámbito local, promoviendo procesos de participación social, política, manteniendo la gobernabilidad.

Responsable: Jefe/a Político/a.

TENENCIAS POLÍTICAS

Misión: Representar al ejecutivo provincial y cantonal, suministrando servicios de calidad a la ciudadanía a través del cumplimiento de las disposiciones legales que lo rigen y propender a la eficiencia y eficacia de sus atribuciones y responsabilidades.

Responsable: Teniente/a Político/a;

DIRECCIÓN DE GARANTÍAS DEMOCRÁTICAS

Misión: Apoyar y coordinar la ejecución e implementación de directrices emitidas por autoridad competente, a fin de desarrollar capacidades de respuesta en el ejercicio de las garantías constitucionales.

UNIDAD DE PROTECCIÓN DE DERECHOS

Misión: Coordinar, dirigir y apoyar la ejecución e implementación en la provincia de políticas internas sobre derechos así como, garantizar la transversalización de derechos fundamentales de la población.

COMISARIAS DE LA MUJER Y LA FAMILIA

Misión: Prevención, detección, administración de justicia especializada en violencia intrafamiliar, remisión y coordinación de atención integral de personas víctimas de violencia de género brindando un tratamiento integral e interdisciplinario por medio del servicio legal, psicológico y social, y promoción de una vida libre de violencia.

Responsable: Comisario/a de la Mujer y la Familia.

DIRECCIÓN DE SEGURIDAD CIUDADANA

Misión: Coordinar con su apoyo técnico y respuesta inmediata con información consolidada de los procesos desconcentrados para retroalimentar con información actualizada y servir de asesoramiento para la toma de decisiones.

UNIDAD DE CONTROL Y GESTIÓN DE LA SEGURIDAD CIUDADANA Y AL ORDEN PÚBLICO.

Misión: Ejecutar políticas, planes, programas, proyectos y acciones en la provincia que mejoren la seguridad ciudadana, prevengan el delito y la violencia, y protejan a la población.

INTENDENCIAS GENERALES DE POLICÍA

Misión: Administrar justicia en bien de la ciudadanía que lo solicitare en el ámbito de la provincia y cantón, prestando servicios legales, eficientes y permanentes, propendiendo al mejoramiento de sus competencias con mayor calidad, agilidad y respeto.

Responsable: Intendente/a General de Policía.

SUB-INTENDENCIA GENERAL DE POLICÍA

Atribuciones y responsabilidades: Ejercerá las mismas atribuciones y responsabilidades que el Intendente General, dentro del ámbito de su circunscripción territorial.

Responsable: Subintendente/a General de Policía.

COMISARIAS NACIONALES DE POLICÍA

Misión: Administrar justicia en bien de la provincia y cantón, prestando servicios legales, eficientes y permanentes, propendiendo el mejoramiento de sus competencias con mayor calidad, agilidad y respeto.

Responsable: Comisario/a Nacional de Policía.

1.2 SITUACIÓN ACTUAL

En la Gobernación de Imbabura no existe la suficiente Articulación Institucional y Sectorial.

Articulación Institucional: *Son las redes de trabajo, la cooperación y el esfuerzo de diversas actividades con otros organismos del Estado, Organizaciones de la Sociedad e Instituciones, que son fundamentales a la hora de implementar acciones en conjunto. La articulación institucional nos permite ampliar los conocimientos, aprovechar de forma eficiente los recursos y potenciar las capacidades de cada una de las partes involucradas.*

Acopio de Información: *Es la recopilación y almacenamiento de información de fuentes documentales: libros, periódicos, revistas, archivos (texto, audio, video), Internet, etc.*

Si bien es cierto las Instituciones trabajan en conjunto pero en su gran mayoría, los procesos de comunicar, transmitir y receptor información a nivel interinstitucional y sectorial, que se realizan fuera y hacia la Gobernación de Imbabura, son procesos manuales, lentos y en ciertos casos excesivamente burocráticos. La población que necesita ser informada es numerosa, y la Gobernación no cuenta con un sistema informático que sea capaz de realizar el intercambio virtual de información y a la vez sirva como fuente de comunicación interinstitucional y contribuya a la toma de decisiones de las Autoridades como el Sr. Gobernador o el Equipo Político.

Al no existir un sistema automatizado ocasiona problemas que afectan las gestiones de la organización, entre los cuales tenemos: falta de decisiones rápidas y acertadas, falta de precisión en la información, retraso en plazos de entrega de información solicitada. Todos

estos problemas llevan en sí al retraso de otras actividades, uso mínimo de los recursos disponibles, incluidos las personas, el tiempo, el flujo de papeles.

La comunicación política se realiza de una manera manual y burocrática, siendo la Gobernación de Imbabura una institución política que representa al Gobierno, entonces necesariamente debe estar a la vanguardia de la tecnología y tener su propio espacio para la comunicación en línea.

1.3 EL PROBLEMA:

En la actualidad la en la Gobernación de Imbabura (Ministerio del Interior) se evidencia insuficiente Articulación Institucional y Sectorial, debido a la falta comunicación y trabajo conjunto entre Instituciones gubernamentales, se deben definir procesos que ayuden a mejorar la transmisión de información, agilizar su trabajo y atención a la ciudadanía, es por ello que se ven en la necesidad de automatizar estos procesos para trabajar conjuntamente entre Instituciones y contribuir a la toma de decisiones en beneficio de la Provincia.

Con la tecnología que se encuentra en constante cambio la Gobernación de Imbabura ve la necesidad de automatizar los procesos y sobre todo mejorar la comunicación y envío de información entre Instituciones que hasta ahora se los ha realizado escasamente por falta de un recurso tecnológico que permita realizarlo.

Todas las instituciones tanto públicas como privadas tratan de expandir y mejorar sus actividades automatizando procesos para brindar un servicio eficaz y eficiente, más aun cuando se trata de una Institución Gubernamental que trabaja para el bienestar de la ciudadanía, de esta manera contribuir con el crecimiento, desarrollo y así cumplir en su gran mayoría con el Plan Nacional del Buen Vivir.

1.4 OBJETIVOS DEL PROYECTO

1.4.1 Objetivo General

- Implementar un Sistema Informático de Acopio de Información y Reporte de Medios de Comunicación mediante el uso de herramientas libres y estándares abiertos, que mejoren la Articulación de procesos entre Instituciones y contribuyan al trabajo conjunto y a la toma de decisiones de la Gobernación Imbabura (Ministerio del Interior).

1.4.2 Objetivos Específicos

- ✚ Definir procesos que ayuden a mejorar la transmisión de la información entre Instituciones (Reingeniería).
- ✚ Establecer los casos de uso y requisitos del sistema indispensables para cubrir con las verdaderas necesidades de la Institución.
- ✚ Desarrollar una aplicación Web amigable para el usuario y así pueda ingresar la información de una forma sencilla, cubriendo con todas las especificaciones previamente establecidas.
- ✚ Utilizar herramientas de desarrollo probadas, de alto rendimiento y tiempos de respuesta mínimos para el correcto desempeño de la aplicación.
- ✚ Estudiar las tecnologías estándar, definidas en este documento, aplicadas al desarrollo de Sistemas Web, analizando la que brinda mayores y mejores beneficios.
- ✚ Estudiar y analizar el patrón de diseño MVC (Modelo Vista Controlador).
- ✚ Estudiar y aplicar técnicas de Desarrollo de Sistemas Web.
- ✚ Estudiar y aplicar todas las fases de la metodología de desarrollo de software RUP.
- ✚ Capacitar a los usuarios de las diferentes Instituciones que harán uso del Sistema de Acopio de Información y Reporte de Medios de Comunicación.

1.5 ALCANCE

El desarrollo a realizar pretende cubrir los siguientes aspectos:

- ✚ Realizar el diseño, desarrollo e implementación de una herramienta virtual que permita a los usuarios en este caso los representantes de las Instituciones, enviar información generada para la articulación y toma de decisiones de la Gobernación.
- ✚ Crear un Módulo de auditoria que permita visualizar los procesos y monitorear el correcto desempeño y los cambios realizados al sistema.
- ✚ Registrar las localizaciones geográficas de las instituciones, delegados institucionales y lugar de suceso de la noticia.
- ✚ Contar con la información de las Instituciones y actores institucionales necesarios para el intercambio de información.
- ✚ Generar reportes de los diferentes medios de comunicación y así visualizar sus publicaciones en las que se involucren a las instituciones gubernamentales.
- ✚ Generar un manual de usuario que permita a los mismos conocer el funcionamiento de la herramienta desarrollada.

Módulos: Después de definir la BDD previo al análisis y definición de casos de uso, están el módulo de Instituciones y Actores Institucionales, módulo de Información y Medios de Comunicación, módulo de Localización Geográfica, módulo de Recursos e Infraestructura de la Institución, módulo de Reportes e Informes, módulo de Administración del sistema y el módulo de Auditoria.

Usuarios: Los usuarios serán de 4 tipos:

El **Administrador** que tendrá acceso a todo el sistema.

El **Cliente** que realizará procesos básicos como actualización y envío de la información y actualización de recursos etc.

El **Periodista** que realizará el ingreso de Medios de Comunicación y Periodistas de los mismos medios, para luego generar reportes de prensa y reportes de programas.

El **Auditor** en el cual quedarán registradas todas las actualizaciones inserciones y borrados del sistema para una eventual auditoria.

Seguridades: Se tiene muy en cuenta la vulnerabilidad de los sistemas en la actualidad así que se implementara el sistema bajo la plataforma GNU Linux y con todas las seguridades necesarias.

1.6 JUSTIFICACIÓN

El presente estudio de factibilidad se realiza para determinar la viabilidad del proyecto, el cual tiene como finalidad automatizar los procesos de articulación de información de la gobernación de Imbabura con diversas Instituciones, esto con el objetivo de agilizar los trámites, y así contribuir al trabajo conjunto por el bien de la ciudadanía.

Mediante el análisis de requerimientos realizado en la Gobernación se observó la falta de un sistema que les permita y facilite el trabajo en conjunto entre instituciones Gubernamentales, realizar envío, recepción de todo tipo de recursos, documentos e información que detalla las actividades a realizar así como también que instituciones y actores institucionales intervienen en las mismas y verificar la ubicación geográfica de generación de la información.

Desde el punto de vista tecnológico la solución propuesta es realizable debido a la gran cantidad de recursos y medios tecnológicos con los que contamos en la actualidad se deberá contar con un medio servidor que para la aplicación será bajo la plataforma GNU Linux para la cual se pretende buscar financiamiento o a su vez se implementará con un servidor

básico, para la mayoría de recursos adicionales se cuenta con el apoyo de medios tecnológicos y capacitación necesaria para realizarlos.

Para la implementación del sistema, la institución adquirirá las herramientas necesarias tanto en hardware como en software, primeramente se adquirirá un hosting de gran rendimiento debido al manejo de archivos como documentos, imágenes y videos y a futuro se planea comprar un servidor para el mejor desempeño no solo del sistema sino que además de la página web y diversos procesos de comunicación entre Instituciones para los que se planea emplear el servidor.

Para el desarrollo e implementación del sistema se utilizará herramientas con licencias GPL ya que la institución no cuenta con algún software en especial, estas herramientas son las siguientes:

Herramientas

Las que se utilizarán en el desarrollo del aplicativo son las siguientes:

Lenguaje de Programación:	PHP 5.2.6
Gestor de Base de Datos:	Postgresql 9.0.4
Servidor Web:	Apache 2.5.10
Framework:	Symfony 1.4
IDE de Desarrollo	NetBeans 5.9.1
Cliente web	Extjs 4.1

Tabla 1: Herramientas de Desarrollo

Fuente: Propia

A medida que se vaya desarrollando la tesis se podrán incorporar herramientas que se crean necesarias para la implementación del sistema.

Arquitectura del Sistema

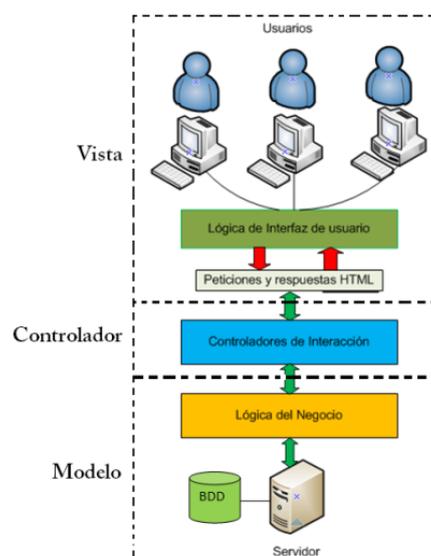


Ilustración 1: Modelo Vista Controlador

Fuente: Propia

Esta es la arquitectura básica de desarrollo y separa sus funcionalidades en tres capas diferenciando claramente la lógica de desarrollo, estas capas son lógica de interfaz de usuario, controlador de interacción, lógica de negocio. Las capas hacen referencia a la forma como la aplicación es segmentada desde el punto de vista lógico. Los niveles corresponden a la forma en que las capas lógicas se encuentran distribuidas de forma física. Esta lógica se basa en el patrón de diseño MVC (Modelo Vista Controlador).

Metodología de Desarrollo

Se usará la metodología de desarrollo RUP en sus cuatro etapas:



Ilustración 2: Fases de la Metodología RUP
Fuente: Propia

Se usará la metodología de desarrollo RUP (Proceso Unificado Racional) además del **Lenguaje Unificado de Modelado UML** para el análisis, implementación y documentación de sistema a desarrollar. Tiene cuatro etapas:

- **Incepción o Fase inicial:** Comprensión del problema y la tecnología. Delimitación del ámbito o alcance del sistema. Establecimiento de una línea base de la arquitectura.
- **Elaboración:** Construir una versión ejecutable de la arquitectura de la aplicación. Trabajo en requerimientos, modelo de negocios, análisis y diseño.
- **Construcción:** Completar el esqueleto de la aplicación con la funcionalidad. Construir una versión beta.
- **Transición:** Hacer disponible a la aplicación para los usuarios finales. Construir la versión Final.

Capítulo 2: Marco Teórico



RESUMEN DEL CAPÍTULO

- *Software Libre y Estándares Utilizados.*
- *Servidor de Aplicaciones Apache.*
- *Base de Datos PostgreSQL.*
- *Framework Symfony.*
- *ExtJS.*

2 MARCO TEÓRICO

2.1 SOFTWARE LIBRE Y ESTÁNDARES UTILIZADOS

2.1.1 SOFTWARE LIBRE

Al referirse o mencionar software se establece o se define como la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

Un software es "libre" cuando garantiza las siguientes libertades:

Libertad	Descripción
0	La libertad de usar el programa, con cualquier propósito.
1	La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.
2	La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.
3	La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.
Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.	

Tabla 2: Libertades del Software Libre
Fuente: http://es.wikipedia.org/wiki/Software_libre

2.1.2 HTML(*HyperTextMarkupLanguage*)

“Lenguaje de marcado de hipertexto, hace referencia al lenguaje de marcado predominante para la elaboración de páginas web que se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. El HTML se escribe en forma de «etiquetas», rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo, JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML”. (Luján Mora, 2013)

2.1.3 CSS

“El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). Como una de las tecnologías preferidas para el diseño Web, las hojas de estilo en cascada facilita a diseñadores web y a programadores seleccionar estilos coherentes para webs”. (López Guimera, 2013)

2.1.4 SERVIDOR WEB

“Un servidor web es un programa que corre sobre un servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de petición, el servidor web busca una página web o bien ejecuta un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.” (Domingo Muñoz, 2013)

2.1.5 PHP

“PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante” (<http://es.wikipedia.org>, 2013)

2.2 SERVIDOR DE APLICACIONES APACHE

2.2.1 Introducción a Apache

“**Apache** es el **Servidor Web** más utilizado, **líder** con el mayor número de instalaciones a nivel mundial muy por delante de otras soluciones como el IIS (Internet Information Server) de Microsoft. Apache es un proyecto de código abierto y uso gratuito, multiplataforma (hay versiones para todos los sistemas operativos más importantes), muy robusto y que destaca por su seguridad y rendimiento.” (<http://www.digitalllearning.es>, 2013)

2.2.2 Estructura y funcionamiento del servidor Apache

El servidor Apache está configurado por módulos, algunos de ellos vienen instalados por defecto y otros los deberemos instalar si queremos usarlos. Los diversos módulos aportan mucha de la funcionalidad que podría considerarse básica para un servidor web. Algunos de estos módulos son:

- **mod_ssl** - Comunicaciones Seguras vía TLS.
- **mod_rewrite** - reescritura de direcciones (generalmente utilizado para transformar páginas dinámicas como php en páginas estáticas html para así engañar a los

navegantes o a los motores de búsqueda en cuanto a cómo fueron desarrolladas estas páginas).

- **mod_dav** - Soporte del protocolo WebDAV (RFC 2518).
- **mod_deflate** - Compresión transparente con el algoritmo deflate del contenido enviado al cliente.
- **mod_auth_ldap** - Permite autenticar usuarios contra un servidor LDAP.
- **mod_proxy_ajp** - Conector para enlazar con el servidor JakartaTomcat de páginas dinámicas en Java (servlets y JSP).

El servidor de base puede ser extendido con la inclusión de módulos externos entre los cuales se encuentran:

- **mod_cband** - Control de tráfico y limitador de ancho de banda.
- **mod_perl** - Páginas dinámicas en Perl.
- **mod_php** - Páginas dinámicas en PHP.
- **mod_python** - Páginas dinámicas en Python.
- **mod_rexx** - Páginas dinámicas en REXX y Object REXX.
- **mod_ruby** - Páginas dinámicas en Ruby.
- **mod_mono** - Páginas dinámicas en Mono
- **mod_security** - Filtrado a nivel de aplicación, para seguridad.

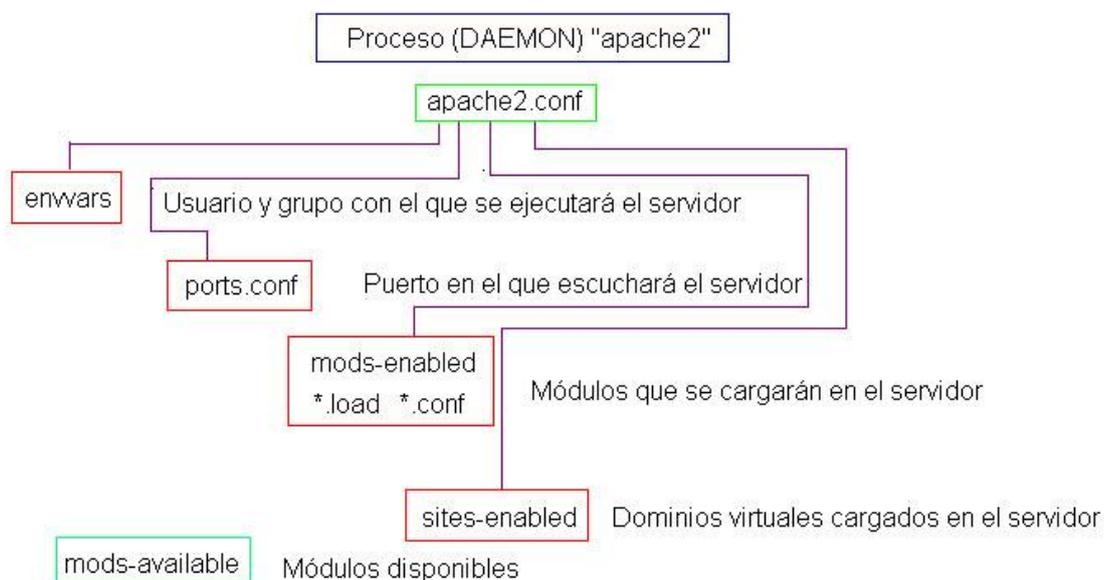


Ilustración 3: Estructura del servidor Apache2.

Fuente: www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/

El archivo de configuración principal de Apache 2 se llama “apache2.conf” y se encuentra en el directorio “/etc/apache2”. El usuario y grupo con el que se ejecuta el servidor Apache se llama **www-data**, Además se cargarán todos los módulos que se encuentren almacenados en los directorios de configuración “mods-enabled” y “sites-enabled”.

2.2.3 Publicación de aplicaciones Symfony PHP

Para publicar una aplicación realizada en Symnfony 1.4 accedemos al archivo *httpd.conf* ubicado para el caso del sistema Operativo Centos 5.4 en el que se desarrolló la aplicación, en la ubicación `/etc/httpd/conf/httpd.conf` y procedemos a crear un host virtual con las instrucciones siguientes:

```
1 # Asegúrate de tener sólo una vez esta línea en su configuración
2 NameVirtualHost 127.0.0.1:8080
3
4 # Esta es la configuración de la Aplicacion Acopio
5
6 # Acopio con symfony 1.4
7 <VirtualHost *:80>
8 DocumentRoot "/home/sfprojects/testrepo/web"
9 DirectoryIndex index.php
10 ServerName symfony.local
11 Alias /sf /home/sfprojects/testrepo/lib/vendor/symfony/data/web/sf
12 <Directory "/home/sfprojects/testrepo/web">
13     AllowOverride All
14     Allow from All
15 </Directory>
16 </VirtualHost>
```

Ilustración 4: Configuración de la aplicación Symfony 1.4 en Apache.
Fuente: Propia

En la Línea:

```
Alias /sf /home/sfprojects/testrepo/lib/vendor/symfony/data/web/sf
```

El alias El alias `/sf` da acceso a las imágenes y los archivos JavaScript necesarios para adecuadamente mostrar las páginas symfony por defecto y la barra de herramientas de depuración web.

Para la línea:

```
ServerName symfony.local
```

El nombre de dominio `symfony.local` tiene que ser declarado localmente. Si se ejecuta un sistema Linux, que tiene que hacerse en el archivo `/etc/hosts`. Si ejecuta Windows XP, VISTA o WIN7, este archivo se encuentra en el directorio `C:\WINDOWS\system32\drivers\etc\`.

```
127.0.0.1 symfony.local
```

2.3 BASE DE DATOS POSTGRESQL



Ilustración 5: Logo de la BDD PostgreSQL
Fuente: www.postgresql.org

2.3.1 Introducción a PostgreSQL

“PostgreSQL es una base de datos relacional, distribuida bajo licencia BSD y con su código fuente disponible libremente. Es el motor de bases de datos de código abierto más potente del momento y en sus últimas versiones empieza a no tener que envidiarle nada a otras bases de datos comerciales.” (Martinez, 2012)

Sus características más importantes son:

- Instalación Ilimitada
- Soporte
- Ahorros en costo de operación
- Estabilidad y Confiabilidad
- Extensible
- Multiplataforma
- Diseñador para ambientes de alto volumen
- Herramientas Gráficas de diseño y administración
- Soporte SQL ANSI – Estándares ACID
- Transacciones
- Integridad Referencial
- Múltiples Tipos de datos definidos
- Soporte de tipos y funciones de usuario
- Conectividad TCP/IP, ODBC, JDBC
- Interfaz con diversos lenguajes de programación.
- Documentación, Extensa

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema, un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

2.3.2 Arquitectura de Postgres

Para entender de mejor manera la arquitectura 4 muestra el grafico de su arquitectura, estableciendo sus principales componentes y funcionalidades:

- **Aplicación cliente:** Aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP ó sockets locales.
- **Demonio postmaster:** Este es el proceso principal de PostgreSQL. Encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autentificar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes
- **Ficheros de configuración:** Los 3 ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf
- **Procesos hijos postgres:** Procesos hijos que se encargan de autentificar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- **PostgreSQL share buffer cache:** Memoria compartida usada por PostgreSQL para almacenar datos en caché.
- **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO).
- **Kernel disk buffer cache:** Caché de disco del sistema operativo.
- **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

Algunas de las limitaciones PostgreSQL son las siguientes:

Máximo tamaño base de datos	Ilimitado (Depende del sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

Tabla 3: Limitaciones de PostgreSQL

Fuente: <http://es.scribd.com/doc/24242911/Informe-de-Limites-Postgres>

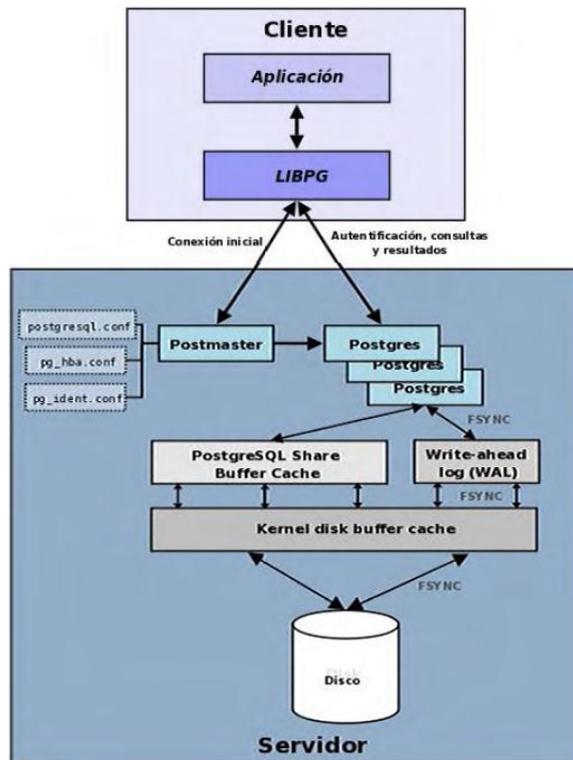


Ilustración 6: Arquitectura de la bdd Postgres

Fuente: <http://www.slideshare.net/eddiemalca/clase-2-arquitectura-de-postgres>

2.4 FRAMEWORK SYMFONY



Ilustración 7: Logo del Framework Symfony

Fuente: www.postgresql.org

2.4.1 Introducción a Symfony

Symfony es un framework PHP de código abierto que gracias a sus excelentes características y grandes fuentes de documentación se ha convertido en uno de los más populares durante los más de cuatro años que lleva generando código y aplicaciones de entorno web.

Este framework puede ser usado para desarrollar aplicaciones con estilo y poco esfuerzo, una vez que el desarrollador tenga los suficientes conocimientos, aunque la curva de aprendizaje sea un poco extensa, ciertamente vale la pena recorrerla por sus grandes beneficios, ahorro de código y extensos pluggins que nos facilitaran aún más nuestro trabajo.

Symfony añade una nueva capa por encima de PHP y proporciona herramientas que simplifican el desarrollo de las aplicaciones web complejas, para entenderlo, solamente es necesario estar familiarizado con los conceptos básicos de la programación moderna, sobre todo la programación orientada a objetos (OOP), el mapeo de objetos a bases de datos (ORM) y el desarrollo rápido de aplicaciones (RAD). El único requisito técnico previo es el conocimiento de PHP.

2.4.2 El entorno de desarrollo de Symfony

“Las aplicaciones de Symfony se pueden ejecutar en diferentes entornos. Todos los entornos comparten el mismo código PHP (salvo el código del controlador frontal) pero pueden tener configuraciones completamente diferentes. Cuando se crea una aplicación, Symfony crea por defecto 3 entornos: producción (prod), pruebas (test) y desarrollo (dev). También es posible añadir cualquier nuevo entorno que se considere necesario”. (<http://librosweb.es>, 2013)

- **El entorno de desarrollo:** Este es el ambiente utilizado por desarrolladores web para añadir nuevas funciones, corregir los errores,
- **El entorno de prueba:** Este entorno se utiliza para probar automáticamente la aplicación.
- **El entorno staging:** Este entorno es utilizado por el cliente para poner a prueba la aplicación e informar errores o características faltantes.
- **El entorno de producción:** Este es el entorno donde un usuario final interactúa.



Ilustración 8: Entorno de Producción Symfony1.4
Fuente: Propia

Para la aplicación realizada se ha utilizado en su gran mayoría el entorno de desarrollo, y posteriormente el entorno de producción con una previa validación en el entorno de pruebas.

2.4.3 Arquitectura de Symfony

Symfony está basado en el patrón clásico del diseño web conocido como arquitectura MVC, que está formado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

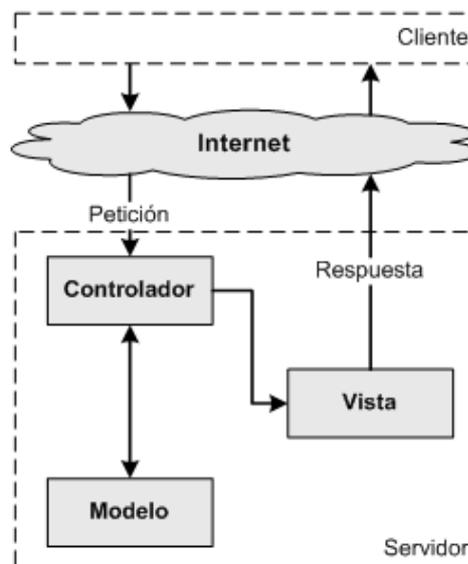


Ilustración 9: El patrón MVC.

Fuente: http://librosweb.es/symfony_1_4/capitulo_2/el_patron_mvc.html

“Symfony divide los proyectos web en aplicaciones y módulos. Dentro de un proyecto, se pueden definir diferentes aplicaciones, compartiendo el mismo modelo de datos (acceso a la BDD), todos los proyectos normalmente están formados por una parte pública a la que acceden los usuarios y una parte de administración con la que los responsables del sitio (Administrador o Administradores) manejan su funcionamiento, gestionan sus contenidos o agregan alguna nueva funcionalidad al sistema. La parte pública es llamada frontend, en esta a su vez se puede acceder a varios entornos, siendo los más utilizados el entorno de desarrollo y luego de concluida la aplicación o en la fase de test el entorno de producción, la parte de administración llamada backend es otra aplicación distinta dentro del mismo

proyecto. El Sistema incluye las dos aplicaciones llamadas frontend y backend.” (<http://librosweb.es>, 2013)

Además, cada aplicación se puede dividir en varios módulos. Cada módulo suele corresponderse con una tabla de la base de datos, aunque no es obligatorio que así sea. Las aplicaciones de nuestro proyecto están formadas por dos módulos cada una, llamados artículo y comentario.

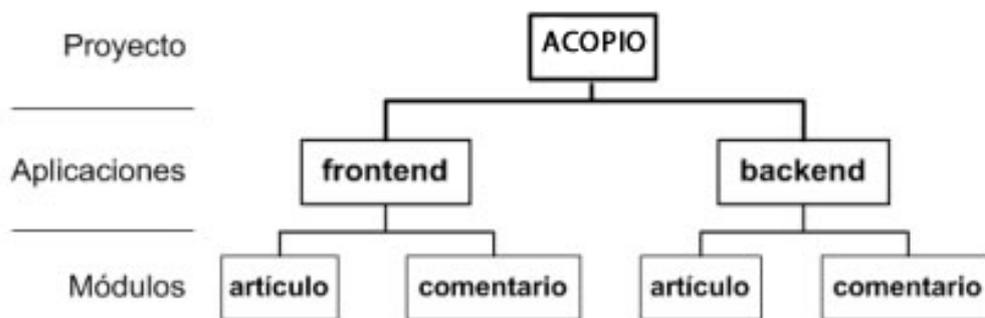


Ilustración 10: Estructura del proyecto Symfony 1.4
Fuente: Propia

Ejemplo proyecto Demo Joobeet

Jobeet es un ejemplo de aprendizaje muy completo, de gran funcionalidad y en el cual además se utiliza y aplica la gran mayoría de bondades y beneficios que ofrece Symfony 1.4, sin duda cualquier persona que desee aprender esta herramienta lo primero que debe realizar es desarrollar el ejemplo en su totalidad, en el ejemplo se definen los siguientes casos de uso:

- **Caso de Uso F1:** En la página principal, los usuarios ven los últimos puestos de trabajo activos.
- **Caso de Uso F2:** Un usuario puede solicitar todos los puestos de trabajo de una categoría determinada.
- **Caso de Uso F3:** Un usuario refina la lista con algunas palabras clave.
- **Caso de Uso F4:** Un usuario hace clic en un puesto de trabajo para ver información más detallada.
- **Caso de Uso F5:** Un usuario envía un puesto de trabajo.
- **Caso de Uso F6:** Un usuario se registra para ser un afiliado.
- **Caso de Uso F7:** Un afiliado recupera la lista activa de puestos de trabajo.
- **Caso de Uso B1:** Un administrador configura el sitio web.
- **Caso de Uso B2:** Un administrador gestiona los puestos de trabajo.
- **Caso de Uso B3:** Un administrador gestiona los afiliados.

Y durante el transcurso de 23 días se cubrirán todos estos requerimientos de una forma muy didáctica y sobretodo resolviendo en su gran mayoría el problema de la curva de aprendizaje Symfony. (<http://symfony.com>, 2013)

- **Día 1:** Comenzando el Proyecto
- **Día 2:** El Proyecto
- **Día 3:** El Modelo De Datos
- **Día 4:** El Controlador y la Vista
- **Día 5:** El Enrutamiento
- **Día 6:** Más acerca del Modelo
- **Día 7:** Jugando con la Página de Categoría
- **Día 8:** Pruebas Unitarias
- **Día 9:** Las Pruebas Funcionales
- **Día 10:** Los Formularios
- **Día 11:** Probando tus Formularios
- **Día 12:** El Generador de Admin.
- **Día 13:** El usuario
- **Día 14:** Feeds o Canales
- **Día 15:** Servicios Web
- **Día 16:** El Correo Electrónico
- **Día 17:** El Motor de Búsqueda
- **Día 18:** AJAX
- **Día 19:** Internacionalización y Localización
- **Día 20:** Los Plugins
- **Día 21:** La Memoria Cache
- **Día 22:** Puesta en Producción
- **Día 23:** Otra Mirada a Symfony

2.4.4 Plugins Symfony 1.4

Un plugin es una extensión desarrollada por la comunidad Symfony, que realiza algún tipo de función específica y que se integra fácilmente al resto de la aplicación, además es muy fácil crear, también es bastante potente, ya que un plugin puede contener cualquier cosa, desde la configuración de los módulos hasta los recursos web. Un Plugin muy popular gracias a su gran funcionalidad y propiedades es **sfDoctrineGuardPlugin**, para garantizar el acceso al backend, nos provee de un formulario para ingreso de usuario y sobre todo las funciones para control de usuarios y permisos de acceso a la aplicación, bastara ingresar la instrucción siguiente para instalar el plugin.

```
$ phpsymfonyplugin:installsfDoctrineGuardPlugin
```

Bastara con reemplazar las siguientes instrucciones:

```
// apps/backend/lib/myUser.class.php
classmyUserextendssfGuardSecurityUser
{
}
```

Y en el archivo settings.yml se debe cambiar la acción por defecto empleada por la página de login:

```
# apps/backend/config/settings.yml
all:
  .settings:
    enabled_modules: [default, sfGuardAuth]

    # ...
    .actions:
login_module:    sfGuardAuth
login_action:    signin

# ...
```

Como los plugins son compartidos entre todas las aplicaciones de un proyecto, tiene que permitir explícitamente los módulos que desea utilizar agregando el ítem `enabled_modules`. (<http://symfony.com>, 2013)

Un formulario de login con tres campos de entrada: 'Username', 'Password' y 'Remember' con un checkbox. Debajo de 'Remember' hay un botón 'sign in'.

Ilustración 11: Formulario de ingreso de sfDoctrineGuardPlugin.
Fuente: http://symfony.com/legacy/doc/jobee/1_4/es/13?orm=Doctrine

2.5 EXTJS 4.1



Ilustración 12: Logo Extjs.
Fuente: <http://www.sencha.com/products/extjs/>

2.5.1 Introducción a Extjs

Antes de poder entrar a examinar ExtJS primero tenemos que hablar sobre RIA, acrónimo de Rich Internet Applications (Aplicaciones Ricas en Internet). Lo que RIA intenta proveer es aquello de lo que siempre ha adolecido la web, una experiencia de usuario muy parecida o igual a la que se tiene en las aplicaciones de escritorio.

Las aplicaciones web tradicionales tienen problemas como la recarga continua de las páginas cada vez que el usuario pide nuevo contenido, o la poca capacidad multimedia, para lo cual se han hecho necesarios plug-ins externos. (Sánchez Rosas, 2008)

“**Ext JS**(pronunciado como "ekst") es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM. Fue desarrollada por Sencha. Originalmente construida como una extensión de la biblioteca YUI por Jack Slocum, en la actualidad puede usarse como extensión para la biblioteca jQuery y Prototype. Desde la versión 1.1 puede ejecutarse como una aplicación independiente”. (<http://es.wikipedia.org>, 2013)

Algunos conceptos importantes que utilizan en Extjs:

2.5.1.1 RIA

Son aplicaciones web que tienen la mayoría de las características de las aplicaciones de escritorio tradicionales, estas aplicaciones utilizan un “navegador web” estandarizado para ejecutarse y por medio de un “plugin” o independientemente con una “virtual machine”, se agregan las características adicionales.

Esta surge como una combinación de las ventajas que ofrecen las aplicaciones Web y las aplicaciones tradicionales. Buscan mejorar la experiencia del usuario.

Normalmente en las aplicaciones Web, hay una recarga continua de páginas cada vez que el usuario pulsa sobre un enlace. De esta forma se produce un tráfico muy alto entre el cliente y el servidor, llegando muchas veces, a recargar la misma página con un mínimo cambio.

En los entornos RIA, en cambio, no se producen recargas de página, ya que desde el principio se carga toda la aplicación, y sólo se produce comunicación con el servidor cuando se necesitan datos externos como datos de una Base de Datos o de otros ficheros externos.

Ejemplo: Un ejemplo muy bueno de RIA es el desktop creado por el ExtJSteam que utiliza un menú muy parecido al del sistema operativo Windows, tiene íconos en el escritorio y despliega ventanas que se pueden mover, cerrar y cambiar de tamaño. (<http://es.wikipedia.org>, 2013)

2.5.1.2 JAVA SCRIPT

JavaScript es un lenguaje de programación interpretado (scripting), es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java.

No es un lenguaje de Programación Orientada a Objetos propiamente dicho como Java, pero gracias a que es basado en prototipos, se puede aplicar conceptos de programación orientada objetos.

Ejemplo:

```
<script type="text/javascript">
functionEnviaAlerta()
{
var texto= document.forms[0].caja.value;
alert(texto);
}
</script>
```

2.5.1.3 AJAX

“Ajax, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano.

De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones. Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y DocumentObjectModel (DOM).” (<http://es.wikipedia.org>, 2013)

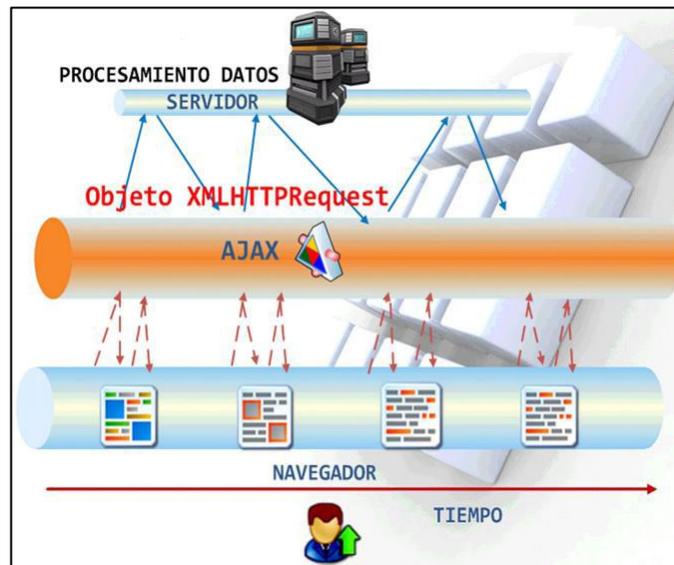


Ilustración 13: Funcionamiento Ajax.
Fuente: <http://www.desarrolloweb.com>

2.5.1.4 El objeto XMLHttpRequest

Su objetivo es hacer peticiones asíncronas al servidor, es la columna vertebral de todas las aplicaciones AJAX. Está admitido por todos los navegadores. Microsoft lo introdujo en IE 5 como un objeto ActiveX. (Horna García & Lara Vasquez, 2008)

2.5.1.5 JSON

JSON, acrónimo de JavaScript ObjectNotation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador semántico de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando el procedimiento eval(), lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web. (Horna García & Lara Vasquez, 2008)

2.5.2 ExtJs es Asíncrono

La programación tradicional para Internet, o también llamada Web 1.0 ejecuta el código en sucesión, es decir espera a que una línea de código se complete antes de ejecutar la siguiente. Al igual que la construcción de una casa, los cimientos deben ser completados antes de que las paredes se puedan construir, a continuación, las paredes deben estar completas antes de que el techo sea construido.

Con ExtJS, que es una herramienta tipo Web 2.0, podemos fácilmente empezar por el techo de la casa antes que los cimientos estén construidos. Es algo así como imaginarse que el techo de la casa está siendo construido en una fábrica, mientras al mismo tiempo, se están construyendo los cimientos y las paredes y cuando todo esté listo, simplemente se arma todo más rápido.

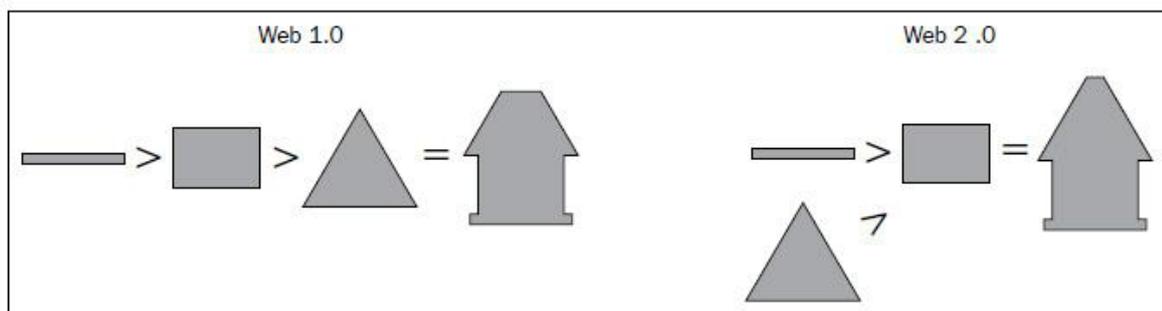


Ilustración 14: Web2.0

Fuente: <http://www.buenastareas.com/ensayos/Curso-Extjs/3156528.html>

Esto nos presenta otras cosas que antes no solíamos tener, ahora ya no estamos obligados a tomar el enfoque línea por línea del desarrollo web.

ExtJS nos ayuda a salir por medio de eventos y manejadores que podemos pegar a nuestra funcionalidad. Esta manera de hacer las cosas se denomina a sincronía. (Jora, 2011)

2.5.3 Funcionalidades

ExtJs dispone de un conjunto de componentes (widgets) para incluir dentro de una aplicación web, como:

- Cuadros y áreas de texto.
- Campos para fechas.
- Campos numéricos.
- Combos.
- Radiobuttons y checkboxes.
- Editor HTML.

- Elementos de datos (con modos de sólo lectura, datos ordenables, columnas que se pueden bloquear y arrastrar, etc.).
- Árbol de datos.
- Pestañas.
- Barra de herramientas.
- Menús al estilo de Windows.
- Paneles divisibles en secciones.
- Sliders.
- Gráficos

The image shows a window titled "Form Fields" with a light blue background. It displays several form controls:

- Text field:** A single-line text input containing "Text field value".
- Password field:** A single-line text input with a white background and a grey border.
- File upload:** A single-line text input followed by a "Browse..." button.
- TextArea:** A multi-line text area containing "Textarea value".
- Display field:** A single-line text input containing "Display field value" in green text.
- Number field:** A single-line text input containing the number "5", with up and down arrow buttons on the right.
- Checkbox:** An unchecked checkbox followed by the text "box label".
- Radio buttons:** Two radio buttons, the first labeled "radio 1" and the second labeled "radio 2".
- Date Field:** A single-line text input with a calendar icon on the right.
- Time Field:** A single-line text input with a dropdown arrow on the right.

Ilustración 15: Entorno (Formulario Extjs).

Fuente: <http://cdn.sencha.com/ext-4.1.1a-gpl/examples/form/field-types.html>

2.5.4 Ventajas

Una de las grandes ventajas de utilizar ExtJS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de layouts similar al que provee Java Swing, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari, etc.).

Entre las principales ventajas tenemos:

- Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.

- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.
- Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico.

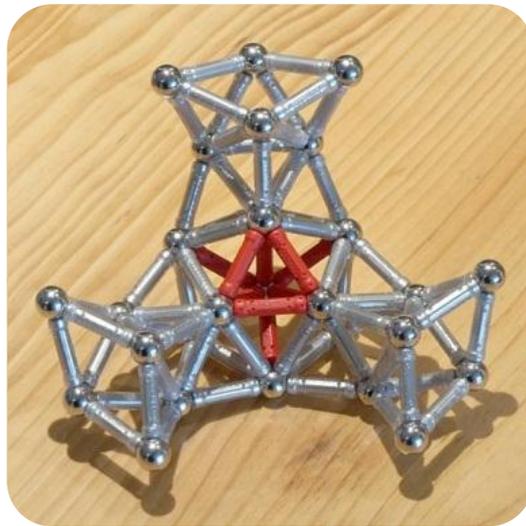
2.5.5 Desventajas

Esta es una lista de algunos puntos negativos en las aplicaciones RIA:

- No se pueden usar fuera de línea. Por su naturaleza web estas Necesita una plataforma. En este caso dependemos de ExtJS para mostrar los componentes y hacer el render de la aplicación.
- El javascript no es tan rápido como quisiéramos, sin embargo con la entrada de Google Chrome y el nuevo motor de Mozilla las cosas van mejorando.
- Descargas lentas. Al ser aplicaciones grandes, especialmente porque cargan todo al inicio, hace que el tiempo de descarga sea mayor al de una aplicación web tradicional.
- Problemas con los motores de búsqueda. Los motores de búsqueda indexan el contenido web estático por lo que los textos cargados de manera dinámica no serán encontrados.
- Accesibilidad. Estas aplicaciones tienen problemas con los programas de accesibilidad pues, al igual que los motores de búsqueda, no trabajan bien con texto cargado dinámicamente.
- aplicaciones no pueden ser usadas en el cliente como cualquier otra aplicación.

En el caso específico de ExtJS existe un factor que hay que tener en cuenta y que nace del hecho de ser una librería Javascript independiente de cualquier tecnología del lado servidor. No existe una forma fácil de realizar binding entre los component visuales con el respectivo modelo, lo cual genera que el programador tenga que escribir más código para validar y enlazar los formularios.

Capítulo 3: Descripción y Funcionamiento de los Módulos



RESUMEN DEL CAPÍTULO

- *Funcionamiento del sistema.*
- *Descripción de los Módulos.*
- *Funcionamiento de los Módulos.*

3 DESCRIPCIÓN Y FUNCIONAMIENTO DE LOS MÓDULOS

3.1 Funcionamiento del sistema

3.1.1 Vista general.

En este capítulo se detallan los Modelos Funcionales elaborados como base para el desarrollo informático del Sistema de Acopio de Información y Reporte de medios de comunicación (AIRCOM). Este documento define al sistema que integra los módulos de Administración, Reportes e Informes, Instituciones y Actores Institucionales, Localización Geográfica, Información y Medios de comunicación, Recursos e Infraestructura, y Auditoria los cuales se encuentran interrelacionados entre sí para la realización y cumplimiento de los diferentes requerimientos.

Al describir las funcionalidades y establecer los procesos para cada uno de los módulos del sistema se ha diseñado y estableciendo como módulos los siguientes:

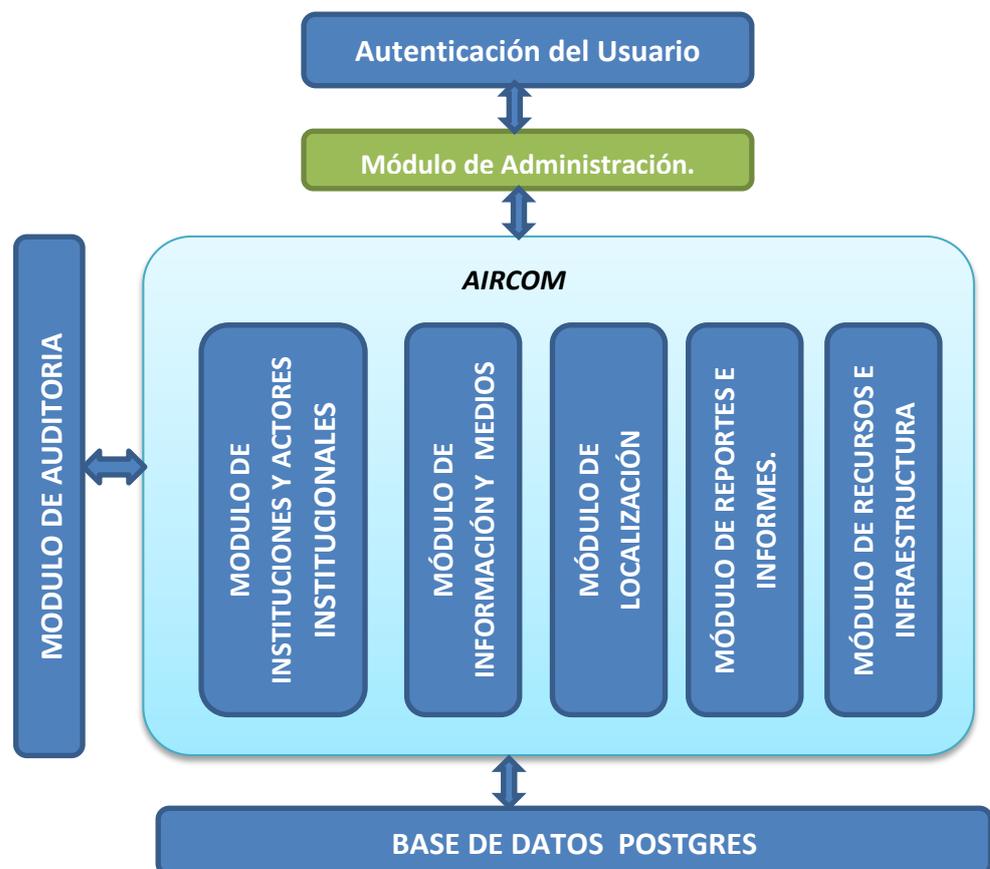


Ilustración 16: Prospectiva del Sistema.
Fuente: Propia

3.1.2 Características del sistema

El sistema AIRCOM tiene las siguientes características que se describen a continuación:

Característica	Descripción
Administración Centralizada.	Se instala y actualiza únicamente del lado del servidor ya que es una aplicación Web y no requiere que se apliquen dichos cambios a los clientes ya que los clientes consumen del servidor.
Fácil Registro de la Información.	La interfaz es amigable y de fácil manejo lo que agiliza el proceso de capacitaciones.
La Disponibilidad es alta	Los usuarios podrán acceder inmediatamente al sistema desde cualquier punto de la Intranet o desde internet a través de cualquier navegador.
Prontitud y eficiencia en la presentación de Información.	Al ser un sistema integrado se puede acceder rápidamente a la información que se encuentren en los diferentes módulos del sistema y tener toda la información necesaria para una tarea en una misma pantalla.
Fácil de Usar	El diseño y la funcionalidad son muy simples y de fácil navegación entre opciones del sistema.
Facilidad para analizar la información.	A través de los diferentes reportes y opciones de búsquedas y consultas con los que consta el sistema.
Sistema Multiplataforma.	El acceso al sistema se podrá realizar desde cualquier navegador web siendo independiente el Sistema Operativo que se maneje.

Tabla 4: Características del Sistema
Fuente: Propia

3.1.3 Dependencias para la Implementación

Para la correcta funcionalidad se debe implementar en un ambiente de pruebas similar al de producción para evitar posibles errores y conflictos de información.

3.1.3.1 Costos

Para la implementación del sistema está planificada la adquisición de un servidor con características adecuadas para el funcionamiento del sistema. Cabe mencionar que el

sistema no aplica costos de licenciamiento ya que se desarrolló a través de una plataforma de código abierto.

3.1.3.2 Licenciamiento

Las tecnologías utilizadas para el desarrollo e implementación del sistema no necesitan de algún pago para la adquisición de licencias ya que el licenciamiento de estas herramientas está regido por las cláusulas de GPL establecido en el software de libre. Para el desarrollo del Sistema de Acopio de Información y Reportes de Medio de Comunicación se utilizará las siguientes herramientas:

- a) PostgreSQL como servidor de Base de Datos.
- b) Framework Symfony(PHP) para desarrollo del lado del servidor.
- c) Doctrine para el mapeo de Datos.
- d) EXTJS (RichClient) para la interfaz.
- e) HTML, CSS, Ajax, JavaScript como estándares de código abierto.
- f) Como metodología de Desarrollo RUP (RationalUnifiedProcess).

3.1.3.3 Instalación

La Instalación es un proceso simple ya que la información estará ubicada en un solo punto llamado servidor, al cual los clientes accederán a través de cualquier navegador web desde internet o intranet dependiendo el caso.

3.1.4 Funcionamiento

El Sistema de Acopio de Información Institucional y Reporte de Medios de Comunicación fue creado utilizando el framework Symfony en la versión 1.4 para la cual se utiliza MVC como patrón base para diseñar aplicaciones interactivas.

La interface gráfica del Sistema utiliza ExtJs para diseñar la parte que de visualización por parte del usuario, teniendo como página inicial el proceso de autenticación de usuario. El flujo de la aplicación inicia por la página de Login o Autenticación, si un usuario intenta acceder directamente a otra página, la aplicación verifica que no se ha autenticado y le re direcciona a la página de autenticación.

Luego de autenticarse el usuario podrá navegar por las opciones que presenta la aplicación de acuerdo al rol previamente establecido de la aplicación dependiendo del rol y el tipo de usuario.

A continuación se muestra el funcionamiento en General de Sistema:

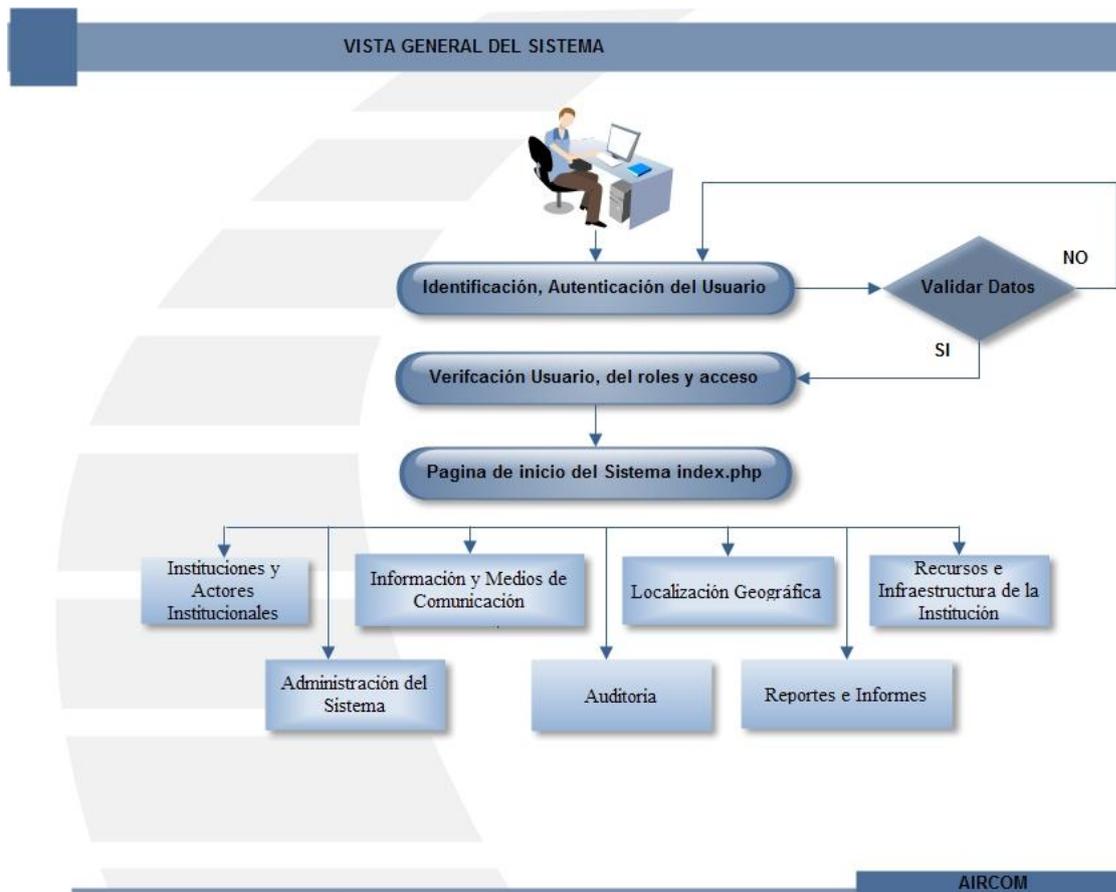


Ilustración 17: Vista General del Sistema.
Fuente: Propia

3.2 Descripción y Funcionamiento de los Módulos

A continuación se indica la descripción y funcionamiento del módulo de Instituciones y Actores Institucionales, módulo de Información y Medios de Comunicación, módulo de Localización Geográfica, módulo de Recursos e Infraestructura de la Institución, módulo de Reportes e Informes, módulo de Administración del sistema.

3.2.1 Descripción del módulo de Administración

El módulo de Administración se encarga de realizar todas las funcionalidades de carácter administrativo con el objetivo de perfeccionar el funcionamiento del sistema. Este proceso

se realiza al iniciar la aplicación y luego comprueba la autorización a las distintas funcionalidades o módulos del sistema.

La seguridad es uno de los requerimientos más importantes que deben ser cumplidos. Está basada en las restricciones más comúnmente usadas en cualquier sistema, con el uso de los conceptos de autenticación y autorización, basados en requerir un usuario y contraseñas válidos, además de un conjunto de roles que le permitirán al usuario realizar solo aquellas operaciones que se le han asignado en el módulo de administración.

3.2.2 Funcionamiento del Módulo de Administración del Sistema

Solo un usuario administrador podrá acceder al módulo de administración donde podrá realizar las acciones que se describen a continuación:

Paso	Acción
1	Autenticarse como usuario administrador
2	Ingresar al menú vista Administración
3	Seleccionar una de las siguientes Opciones: <ol style="list-style-type: none"> 1. Reportes 2. Administrar Tipos de Usuarios 3. Administración de Roles 4. Administración de Usuarios
3.1	Administrar Instituciones <ol style="list-style-type: none"> 1) Visualizar lista de Instituciones existentes. 2) Registro de Nuevas Instituciones. 3) Modificar Instituciones. 4) Eliminar Instituciones.
3.2	Administrar Tipos de Usuarios <ol style="list-style-type: none"> 1) Visualizar lista de Tipos de Usuarios 2) Registro de Nuevo Tipo de Usuario 3) Modificar Tipos de Usuarios 4) Eliminar Tipos de Usuarios
3.3	Administración de Usuarios <ol style="list-style-type: none"> 1) Visualizar lista de Usuarios registrados 2) Registro de nuevos usuarios, seleccionar el tipo de usuario ya asignado los roles previamente. 3) Modificar usuarios, para cualquier cambio debe ingresar la contraseña del usuario a modificar. 4) Eliminar usuarios

Tabla 5: Funcionamiento del Módulo de Administración.
Fuente: Propia

3.2.3 Descripción del Módulo de Auditoria

El módulo de auditoria es aquel donde se registran todas las transacciones importantes de realizadas por un usuario en la aplicación, los accesos y cambios en la base de datos, el usuario que la registro, la hora, el valor y del campo que modificado.

3.2.4 Funcionamiento del Módulo de Auditoria.

Solo un usuario de tipo auditor tiene permisos acceso al módulo de auditoria, podrá realizar las siguientes consultas como se muestra a continuación.

Paso	Acción
1	Autenticarse como usuario Auditor
2	Ingresar al módulo de Auditoria
3	Listar Registros
4	Seleccionar una consulta mediante las siguientes Opciones: 1. Buscar por Usuario 2. Buscar pro Fecha de Modificación. 3. Buscar por tipo de Modificación. 4. Buscar por nombre de la Tabla de datos. 5. Buscar por nombre de la columna.

Tabla 6: Funcionamiento del Módulo de Auditoria.
Fuente: Propia

3.2.5 Descripción del Módulo de Recursos e Infraestructura de la Institución.

El módulo de infraestructura es aquel donde se registran todos los recursos con los que cuenta la Gobernación, no es un módulo extenso ya que únicamente está dedicado solventar la necesidad de almacenar y registrar dichos recursos.

3.2.6 Funcionamiento del Módulo de Recursos e Infraestructura de la Institución.

Paso	Acción
1	Autenticarse como usuario Administrador
2	Ingresar al módulo de Inventarios
3	Seleccionar una de las siguientes Opciones: 1. Lista de Productos 2. Lista de Proveedores 3. Lista de Categorías 4. Lista de Departamentos 5. Listado de Componentes 6. Listado de Componentes Listado de Estados

3.1	Lista de Productos a) Crear b) Editar c) Borrar d) Cargar Listado de Productos
3.2	Lista de Proveedores a) Crear b) Editar c) Borrar d) Cargar Listado de Proveedores
3.3	Lista de Categorías a) Crear b) Editar c) Borrar d) Cargar Listado de Categorías
3.4	Lista de Departamentos a) Crear b) Editar c) Borrar d) Cargar Listado de Departamentos
3.5	Lista de Componentes a) Crear b) Editar c) Borrar d) Cargar Listado de Componentes
3.6	Lista de Elementos a) Crear b) Editar c) Borrar d) Cargar Listado de Elementos
3.7	Lista de Estados a) Crear b) Editar c) Borrar d) Cargar Listado de Estados

Tabla 7: Funcionamiento del Módulo de Recursos e Infraestructura de la Institución.
Fuente: Propia.

3.2.7 Descripción del Módulo de Localizaciones Geográficas

El módulo de Localizaciones Geográficas hace uso del API de Google Maps, para ubicar los puntos de referencia con latitud y longitud de las: Instituciones, Medios de Comunicación y lugares donde se generan los proyectos que se realizan en beneficio de la Ciudadanía Imbabureña.

3.2.8 Funcionamiento del módulo de Localización Geográfica

Paso	Acción
1	Autenticarse como usuario Administrador o Cliente
2	Ingresar al módulo de Instituciones, Medios de Comunicación o Proyectos
3	Seleccionar una de las siguientes Opciones: <ol style="list-style-type: none"> 1. Lista de Instituciones 2. Lista de Medios 3. Lista de Proyectos
3.1	Lista de Instituciones <ol style="list-style-type: none"> 1) Crear o editar <ol style="list-style-type: none"> a) Ingresar en la pantalla de Ubicaciones Geográficas. 2) Ingresar en el recuadro la dirección buscada. 3) Seleccionar y arrastrar el marcador hasta encontrar la ubicación.
3.2	Lista de Medios de Comunicación <ol style="list-style-type: none"> 4) Crear o editar <ol style="list-style-type: none"> a) Ingresar en la pantalla de Ubicaciones Geográficas. 5) Ingresar en el recuadro la dirección buscada. a) Seleccionar y arrastrar el marcador hasta encontrar la ubicación.
3.3	Lista de Proyectos <ol style="list-style-type: none"> 6) Crear o editar <ol style="list-style-type: none"> a) Ingresar en la pantalla de Ubicaciones Geográficas. 7) Ingresar en el recuadro la dirección buscada. a) Seleccionar y arrastrar el marcador hasta encontrar la ubicación.

Tabla 8: Funcionamiento del módulo de Localización Geográfica
Fuente: Propia

3.2.9 Descripción del Módulo de Instituciones y Actores Institucionales.

En el módulo de Instituciones y Actores Institucionales es en donde se registran las Instituciones que tiene algún tipo de vínculo con la Gobernación de Imbabura y a su vez a los responsables de estas instituciones ya sean Directores Provinciales de Ministerios, Jefes Políticos, tenientes políticos, presidentes de juntas parroquiales y diversas autoridades.

3.2.10 Funcionamiento del Módulo de Instituciones y Actores Institucionales.

Paso	Acción
1	Autenticarse como usuario Administrador
2	Ingresar a la opción de proyectos
3	Seleccionar una de las siguientes Opciones: <ol style="list-style-type: none"> 1. Lista de Instituciones 2. Lista de Responsables 3. Lista de Proyectos
3.1	Lista de Instituciones

	<ul style="list-style-type: none"> a) Crear b) Editar c) Borrar d) Cargar Listado de Instituciones
3.2	Lista de Responsables <ul style="list-style-type: none"> a) Crear b) Editar c) Borrar d) Cargar Listado de Responsables
3.3	Lista de Proyectos <ol style="list-style-type: none"> 1. Crear <ul style="list-style-type: none"> a) Nuevo Proyecto b) Ingresar los valores a los campos requeridos, la Ubicación, Objetivos, Instituciones y Responsables involucrados. c) Registrar el proyecto. d) Imprimir el proyecto. 2. Editar <ul style="list-style-type: none"> a) Editar Proyecto b) Ingresar los valores a los campos requeridos, la Ubicación, Objetivos, Instituciones y Responsables involucrados. c) Actualizar el proyecto. d) Imprimir el proyecto. 3. Borrar 4. Cargar Listado de Proyectos

Tabla 9: Funcionamiento del Módulo de Instituciones y Actores Institucionales.
Fuente: Propia

3.2.11 Descripción del Módulo de Información y Medios de Comunicación.

En este módulo, se realiza el Almacenamiento categorización e indexación de los medios de comunicación, así como también el ingreso de información.

En las opciones de Reporte de Prensa y Reporte de Programa se Registra de forma detallada lo que se dijo de las Instituciones y de los Representantes Instituciones, además se puede adjuntar archivos de evidencia (recortes de periódicos, videos, audio) de esta información.

3.2.12 Funcionamiento del Módulo de Información y Medios de Comunicación.

Paso	Acción
1	Autenticarse como usuario Administrador o Periodista
2	Ingresar al módulo de Medios de Comunicación
3	Seleccionar una de las siguientes Opciones: <ol style="list-style-type: none"> 1. Lista de Medios 2. Lista de Programas

	<ul style="list-style-type: none"> 3. Espacio Contratado 4. Espacio de Prensa Contratado 5. Espacio de Radio y Tv Contratado 6. Horario 7. Lista de Periodistas 8. Lista de Partidos Políticos
3.1	Lista de Medios <ul style="list-style-type: none"> a) Crear b) Editar c) Borrar d) Cargar Listado de Medios
3.2	Lista de Programas <ul style="list-style-type: none"> a) Crear b) Editar c) Borrar d) Cargar Listado de Programas
3.3	Espacio Contratado <ul style="list-style-type: none"> a) Crear b) Editar c) Borrar d) Cargar Listado de Espacios Políticos Contratados
3.4	Espacio de Prensa Contratado <ul style="list-style-type: none"> a) Crear b) Editar c) Borrar d) Cargar Listado de Espacios de Prensa Contratados
3.5	Espacio de Radio y Tv Contratado <ul style="list-style-type: none"> a) Crear b) Editar c) Borrar d) Cargar Listado de Espacios de Radio y Tv Contratados.
3.6	Horario <ul style="list-style-type: none"> a) Crear b) Editar c) Borrar d) Cargar Listado de Horarios
3.7	Lista de Periodistas <ul style="list-style-type: none"> a) Crear b) Editar c) Borrar d) Cargar Listado de Periodistas
3.8	Lista de Partidos Políticos <ul style="list-style-type: none"> a) Crear b) Editar

	<ul style="list-style-type: none"> c) Borrar d) Cargar Listado de Partidos Políticos
--	--

Tabla 10: Funcionamiento del Módulo de Información y Medios de Comunicación.

Fuente: Propia

3.2.13 Descripción del Módulo de Reportes e Informes.

Modulo en el cual se presenta la Visualización, impresión y exportación de reportes según especificaciones realizadas de los usuarios (Cliente, Administrador, Periodista o Auditor) a cualquiera de los módulos anteriores, verificando el acceso y restricciones adecuadas para cada uno de los roles asignados.

3.2.14 Funcionamiento del Módulo de Reportes e Informes.

Paso	Acción
1	Autenticarse como usuario Administrador, Cliente, Periodista o Auditor.
2	Ingresar al cualquier módulo.
3	Seleccionar una de las Opciones de los módulos.
4	Imprimir

Tabla 11: Funcionamiento del Módulo de Reportes e Informes.

Fuente: Propia

Capítulo 4: Diseño y Desarrollo del Aplicativo



RESUMEN DEL CAPÍTULO

- *Introducción.*
- *Fase de Incepción.*
- *Fase de Elaboración.*
- *Fase de Construcción.*

4 DISEÑO Y DESARROLLO DEL APLICATIVO

4.1 Introducción

Para realizar el diseño y en general el desarrollo del sistema de Acopio de Información y Reporte de Medios de comunicación con los módulos: Instituciones y Actores Institucionales, módulo de Información y Medios de Comunicación, módulo de Localización Geográfica, módulo de Recursos e Infraestructura de la Institución, módulo de Reportes e Informes, módulo de Administración del sistema y el módulo de Auditoria, se ha utilizado la metodología de desarrollo RUP (Rational Unified Process).

Antes, una corta introducción a las fases de la metodología RUP:

- 1. Inicio (Inception).**-El objetivo general de esta fase es establecer un acuerdo entre todos los interesados acerca de los objetivos del proyecto. Esta fase es significativamente primaria para el desarrollo de nuevo software, ya que se asegura de identificar los riesgos relacionados con el negocio y requerimientos. Para proyectos de mejora de software existente esta fase es más breve y se centra en asegurar que vale la pena y es posible, desarrollar el proyecto.
- 2. Elaboración.**- El objetivo en esta fase es establecer la arquitectura base del sistema para proveer bases estables para el esfuerzo de diseño e implementación en la siguiente fase. La arquitectura debe abarcar todas las consideraciones de mayor importancia de los requerimientos y una evaluación del riesgo.
- 3. Construcción.**- El objetivo de la fase de construcción es clarificar los requerimientos faltantes y completar el desarrollo del sistema basados en la arquitectura base. Vista de cierta forma esta fase es un proceso de manufactura, en el cual el énfasis se torna hacia la administración de recursos y control de las operaciones para optimizar costos, tiempo y calidad.
- 4. Transición.**- Esta fase se enfoca en asegurar que el software esté disponible para sus usuarios. Esta fase se puede subdividir en varias iteraciones, además incluye pruebas del producto para poder hacer el entregable del mismo, así como realizar ajuste menores de acuerdo a ajuste menores propuestos por el usuario. En este punto, la retroalimentación de los usuarios se centra en depurar el producto, configuraciones, instalación y aspectos sobre utilización.

4.2 FASE DE INCEPCIÓN

En esta fase se describe principalmente el alcance del proyecto a través del documento de Visión, se determinó los requerimientos a través de actas de trabajo y se desarrolló un plan de desarrollo de software.

4.2.1 Documento de Visión

4.2.1.1 Introducción

4.2.1.1.1 Propósito

El propósito de este documento es definir los requerimientos del Sistema de Acopio de Información y Reporte de Medios de Comunicación (AIRCOM).

El sistema AIRCOM se encargará de realizar todo el proceso de recolección de la información generada por las Instituciones dependientes o con algún grado de cooperación con la Gobernación de Imbabura de manera detallada y automática. Así mismo se encargara mantener un registro de los medios de comunicación y sus publicaciones referidas a las instituciones gubernamentales.

El detalle de cómo el sistema cubrirá las necesidades de los usuarios se especifica en los Casos de Uso.

4.2.1.1.2 Alcance

El documento de Visión presentara una perspectiva del Sistema de Acopio de Información y Reporte de Medios de Comunicación para la Gobernación de Imbabura (Ministerio del Interior).

El Software permitirá enviar información por parte de los encargados o responsables de las instituciones gubernamentales o que estén relacionados con la gobernación de algún modo, así mismo permitirá al equipo de comunicación de la Gobernación organizar la información y hacer un seguimiento a los principales medios de comunicación de la provincia.

4.2.1.1.3 Definiciones, Siglas y Abreviaturas

Ver Glosario.

4.2.1.1.4 Referencias

Glosario

Resumen del Modelo de Casos de Uso

4.2.1.2 Posicionamiento

4.2.1.2.1 Oportunidad de negocio

Este sistema nace como una necesidad y carencia de medios y recursos informáticos automatizados por parte de la Gobernación, y se acentúa a partir del surgimiento del Área de comunicación social, con lo cual se hace sumamente imprescindible la utilización de recursos tecnológicos. El proceso de creación del sistema acogerá las sugerencias de Todo el Equipo Político, el Área de comunión Social de la Gobernación y los sujetos políticos que estarán involucrados en el proceso de envío e intercambio de la Información.

La Articulación Interinstitucional es un proceso indispensable que tiene la Gobernación con las diversas instituciones, explotando las capacidades de cada uno de las partes, actualmente no se cuenta con un proceso automatizado que facilite la realización de estos procesos.

Luego de un análisis realizado por los desarrolladores, se pudo establecer la necesidad del sistema para cubrir las necesidades de la Gobernación, se presenta algunas falencias o dificultades:

1. No existe un gestor de base de datos

No existe sistema Informático Alguno disponible en la Gobernación, debiendo definir las herramientas adecuadas para el correcto funcionamiento como base de datos y tecnología de lado del servidor y del lado del cliente.

2. Tecnología desactualizada

Al no contar con un sistema Informático, no se está aprovechando de los recursos tecnológicos actuales, que pueden facilitar todo tipo de procesos de la Institución.

3. Falta de control y auditoría

Se debe mantener un registro de las principales actividades y procesos que se realice, si bien es cierto esto se lleva a cabo de una manera manual, no es lo adecuado por motivos de veracidad de datos y prontitud en la entrega y visualización de la información.

4. Seguridad deficiente

La información está en riesgo debido a que esta se almacena de forma manual, pudiendo perderse datos importantes o simplemente no registrarlos.

Con referencia a lo expuesto anteriormente, el Sr. Gobernador junto al equipo político y de Comunicación de la Gobernación de Imbabura ha planteado crear un sistema informático. El sistema "AIRCOM" entonces brindará las facilidades necesarias para el manejo de la información generada por la institución que trabajen en conjunto con la gobernación. Por otro lado al diseñar e implementar un nuevo sistema, obliga a definir los procesos más relevantes para la aplicación, estos módulos son establecidos:

- a) Módulo de Instituciones y Actores Institucionales,
- b) Módulo de Información y Medios de Comunicación,
- c) Módulo de Localización Geográfica,
- d) Módulo de Recursos e Infraestructura de la Institución,
- e) Módulo de Reportes e Informes,
- f) Módulo de Administración del sistema
- g) Módulo de Auditoria.

4.2.1.2.2 Definición del problema

El problema de	No existe ningún tipo de aplicación actual para la automatización de procesos de la Gobernación. Dificultad para el mantenimiento de los procesos ya que se llevan a cabo de forma manual. Falta de control y prontitud en la entrega de información.
Que afecta a	Gobernación Imbabura en General, Gobernador, Equipo Político, Área de Comunicación Social y empleados que llevan el registro de las actividades realizadas (Control de Agenda y Secretaria) Incluso la ciudadanía Imbabureña al no ser informada de las actividades que realiza la Gobernación en conjunto con las diversas instituciones para nuestro beneficio.
El impacto de ello es	Si no se crea un Sistema para la automatización de procesos, afectara principalmente a la toma de decisiones oportunas por parte de la gobernación ya que no se tiene

	<p>un control adecuado de cómo se está llevando a cabo el trabajo conjunto con diversas instituciones, tampoco se podrá visualizar la información con prontitud y de una forma estructurada.</p> <p>Existen muchas actividades y procesos manuales que no permiten una gestión eficiente.</p> <p>Inconsistencia en la información generada.</p>
Una solución exitosa debería	<p>Automatizar los procesos y requerimientos para la articulación interinstitucional de los involucrados en el proceso de mejoramiento de la calidad de vida de la población Imbabureña.</p> <p>Llevar un control adecuado de los medios de Comunicación de la provincia, sus posiciones políticas, calidad de información generada y principales publicaciones referentes a las instituciones gubernamentales y políticas.</p> <p>Implementar una solución informática de calidad soportada por una metodología eficiente de desarrollo de software.</p>

Tabla 12: Definición del Problema
Fuente: Propia (Metodología RUP.)

4.2.1.3 Descripción de los interesados y usuarios

Las siguientes tablas indican la responsabilidad de interesados y usuarios.

4.2.1.3.1 Resumen de los interesados

Interesados son todas aquellas personas involucradas directamente en la definición y alcance del proyecto. A continuación se presenta la lista de los interesados:

Nombre	Descripción	Responsabilidad
Coordinador del proyecto	Responsable del proyecto a nivel directivo (Analista de	Establecer los procesos y lineamientos generales para el

	Conflictos) de la Gobernación Imbabura (Ministerio del Interior)	desarrollo del proyecto. Coordinar los diferentes requerimientos que surjan en el desarrollo del sistema. Proveer de los recursos necesarios para el correcto desarrollo del proyecto.
Responsable del proyecto	Responsable del proyecto (Desarrollador de la tesis).	Responsable del análisis y diseño del proyecto. Gestiona el correcto desarrollo del proyecto en lo referente a la construcción e implantación.
Responsable funcional	Responsables del proyecto por parte de las dos Áreas las que beneficiara el sistema, como son la de Jefatura Política y el Área de Comunicación.	Responsables de coordinar con los usuarios la correcta definición de los requerimientos y los lineamientos y funciones específicas del sistema.
Jefe Departamental	Responsable de proporcionar la información para el registro de datos, reportes y medios de comunicación.	Definir la estructura de la información que se utilizará para el levantamiento y registro de la misma, en las Áreas asignadas de la Gobernación.

Tabla 13: Resumen de Interesados.
Fuente: Propia (Metodología RUP)

4.2.1.3.2 Resumen de los usuarios

Los usuarios son todas aquellas personas que actúan directamente en el uso del sistema AIRCOM. A continuación se presenta el listado de los mismos:

Nombre	Descripción	Responsabilidad
Administrador del sistema	Persona responsable del manejo y control del sistema "AIRCOM" y de su administración.	Administrar eficazmente el sistema (creación de nuevos usuarios y gestionar acceso a usuarios, facilitar mantenimiento al sistema frente a nuevos requerimientos).
Administrador funcional del sistema	Personas designadas de las dos Áreas involucradas directamente con el sistema, como son: Jefatura Política y área de Comunicación que administraran el sistema AIRCOM según su naturaleza.	Administrar funcionalmente el sistema: verificar la información generada, definir los tipos de transacciones, nuevos formatos de envío, interpretar la información generada, etc.
Usuario del sistema	Responsables de cada una de las Instituciones Gubernamentales o GAD's que deberán enviar información a la Gobernación mediante el sistema AIRCOM.	Ingresar la información referente a cada a la labor realizada (reporte de proyectos, reporte de medios o información relevante) generada en cada una de las Instituciones en las que es Responsable.
Usuario de gestión del sistema	Personal de la Gobernación Imbabura.	Validar la información proveniente de las diferentes Instituciones gubernamentales o GAD's de la Provincia. Consolidar la información.

Tabla 14: Resumen de los Usuarios.
Fuente: Propia (Metodología RUP.)

4.2.1.3.3 Entorno actual de los usuarios

- a) El Área de comunicación y el Equipo político dela Gobernación Imbabura, serán usuarios del sistema AIRCOM, y beneficiará así a todos quienes hagan uso del

mismo ya que permitirá registrar y llevar un control adecuado de todos los procesos y sobretodo del trabajo en conjunto que se realice con la Gobernación y las diferentes Instituciones que están bajo su responsabilidad.

- b) Debido a las políticas establecidas para el uso de software libre en el Ecuador y siendo La Gobernación de Imbabura una institución del estado, el sistema será implementado en su totalidad con software libre.
- c) Actualmente en la Gobernación no se maneja ningún tipo de software para la automatización de procesos de articulación Institucional y Reportes de Medios de comunicación, lo que determina de forma urgente a la institución a diseñar su propio sistema que se ajuste a sus necesidades.
- d) Los Usuarios podrán acceder al sistema desde un navegador de Internet previo a su asignación de Roles, usuarios y contraseñas, beneficiando así el flujo de trabajo interinstitucional.
- e) Los Módulos de Recursos e Infraestructura de la Institución, Módulo de Información y Medios de Comunicación, Módulo de localización Geográfica, Módulo de Instituciones y actores Institucionales deberán interactuar entre sí según sea necesario.

4.2.1.3.3.1 Coordinador de Proyecto

Representante	Ing. Víctor Hugo Portocarrero
Descripción	Responsable del proyecto por parte de la Gobernación de Imbabura.
Tipo	Analista de Conflictos.
Responsabilidades	Establecer los lineamientos generales para el desarrollo del proyecto. Definir los principales procesos, coordinar y solventar los diferentes requerimientos que surjan en el desarrollo del sistema.
Criterios de éxito	Mantener activo el sistema luego de su implementación. Mantener la funcionalidad del sistema.
Implicación	Revisor de la Administración

Entregables	N/A
Comentarios	Mantener una relación constante con el desarrollo del proyecto. Brindar apoyo a nivel gerencial cuando sea necesario.

Tabla 15: Perfil de Coordinador del Proyecto.
Fuente: Propia (Metodología RUP)

4.2.1.3.3.2 Responsable del Proyecto

Representante	Marlon Arichávala.
Descripción	Responsables del proyecto por parte de la Gobernación de Imbabura.
Tipo	Analistas de sistemas
Responsabilidades	Responsable del análisis, diseño e implementación del proyecto. Gestiona el correcto desarrollo del proyecto en lo referente a la construcción e implantación.
Criterios de éxito	Cumplir con el cronograma determinado. Cumplir con los requerimientos establecidos para el sistema. Obtener un sistema de calidad que cumpla con los requerimientos funcionales establecidos.
Implicación	Jefe de proyecto (Project Manager)
Entregables	Documento de visión Resumen del modelo de casos de uso Especificaciones del modelo de casos de uso Diseño ER de la base de datos y el diccionario. Especificaciones complementarias

Tabla 16: Perfil Responsable del Proyecto.
Fuente: Propia (Metodología RUP.)

4.2.1.3.3.3 Responsable Funcional

Representante	Analista de Conflictos, David Sandoval
Descripción	Responsables del proyecto por Áreas de parte de la Gobernación Imbabura.
Tipo	Jefe departamental y Experto en el tema.
Responsabilidades	Responsable de coordinar con los diferentes usuarios (Responsables de las Instituciones) la correcta determinación de

	los requerimientos y la correcta concepción del sistema. Coordinar las pruebas de validación del nuevo sistema. Coordinar y asegurar la capacitación de los usuarios. Receptar y clasificar e Interpretarla Información Generada.
Criterios de éxito	Obtener un sistema de calidad que cumpla con los requerimientos funcionales establecidos.
Implicación	Aprueba las especificaciones funcionales y las pruebas realizadas.
Entregables	Documento de revisión de las especificaciones funcionales. Documento de revisión de las pruebas funcionales
Comentarios	

Tabla 17: Perfil responsable Funcional.
Fuente: Propia (Metodología RUP)

4.2.1.3.4 Perfiles de usuario.

Los perfiles de cada uno de los usuarios que tendrán acceso al sistema AIRCOM se especifican a continuación en las siguientes tablas.

4.2.1.3.4.1 Administrador del Sistema.

Representante	N/A
Descripción	Persona designada por la Gobernación para administrar el sistema AIRCOM.
Tipo	Operador, Analista de Sistemas
Responsabilidades	Administrar funcionalmente el sistema: gestionar acceso a usuarios, dar mantenimiento al sistema frente a nuevos requerimientos, interpretar y presentar.
Criterios de éxito	Dar una oportuna respuesta, coordinar con los responsables de cada área y solventar las necesidades y problemas que se presenten oportunamente.
Implicación	Administración y mantenimiento de la aplicación.
Entregables	Bitácora de control de nuevos requerimientos. Bitácora de control de incidencias del nuevo sistema.
Comentarios	N/A

Tabla 18: Perfil Administrador del Sistema.
Fuente: Propia (Metodología RUP.)

4.2.1.3.4.2 Administrador Funcional del Sistema

Representante	Analista de conflictos, David Sandoval
Descripción	Persona delegadas por áreas; Equipo Político y equipo de Comunicación de la Gobernación Imbabura que administran el sistema AIRCOM
Tipo	Expertos en el Tema
Responsabilidades	Creación de nuevas cuentas de usuarios, definición requerimientos, ingreso de información, presentación de resultados etc.
Criterios de éxito	Dar una respuesta oportuna ante los requerimientos y proveer una buena administración de sus respectivas áreas asignadas.
Implicación	Administración y mantenimiento de la aplicación por su respectiva Área.
Entregables	N/A
Comentarios	N/A

Tabla 19: Perfil Administrador Funcional del Sistema
Fuente: Propia (Metodología RUP)

4.2.1.3.4.3 Usuario del Sistema

Representante	N/A
Descripción	Responsable seleccionado de cada una las diferentes instituciones que trabajen conjuntamente con la Gobernación y que harán uso de sistema AIRCOM.
Tipo	Delegado Institucional de las Instituciones involucradas.
Responsabilidades	Ingresar la información concierne y generada por parte de cada institución en el sistema AIRCOM. Realizar consultas de información de reportes, etc.
Criterios de éxito	N/A
Implicación	N/A
Entregables	N/A
Comentarios	N/A

Tabla 20: Perfil del Usuario del Sistema.
Fuente: Propia (Metodología RUP)

4.2.1.3.5 Necesidades de los interesados y Usuarios

Necesidades	Prioridad	Inquietudes	Solución Actual	Solución propuesta
Diseñar un sistema que facilite la consolidación e integración de información interinstitucional mediante la automatización de procesos por parte de la Gobernación de Imbabura.	Alta	El sistema debe consolidar la información para facilitar registro y administración de información.	No existe un sistema que se encargue de estos procesos. Los procesos de articulación se manejan manualmente.	Implementar el sistema de acopio de Información Institucional y Reporte de medios de comunicación con herramientas de software actuales que nos permita integrar de forma eficiente estos procesos en un solo sistema.
Elaborar el sistema utilizando herramientas y software libre que facilite y agilice su desarrollo.	Alta	Se debe utilizar software libre para el desarrollo del sistema.	N/A	Desarrollar el sistema utilizando las herramientas de desarrollo como, framework Symfony, interfaces de usuario ricas EXTJS, servidor de aplicaciones apache, estándares abiertos como CSS, etc.

Necesidades	Prioridad	Inquietudes	Solución Actual	Solución propuesta
La interfaz del sistema debe ser fácil de manejar, cumpliendo con todos los requerimientos establecidos.	Alta	Cumplir con todos los requerimientos de los usuarios.	N/A	Desarrollo con la ayuda de los expertos en cada uno de los departamentos y Áreas involucradas de la Gobernación de Imbabura.
Lograr la implementación del sistema en el menor tiempo posible para su utilización desde el próximo año.	Alta	Cumplir con las especificaciones definidas previo al debido análisis y levantamiento de la Información.	Actualmente no se cuenta con un sistema que maneje los procesos de articulación entre instituciones.	Trabajar con el sistema AIRCOM desde el año 2013 ya con el sistema implementado y en funcionamiento.

Tabla 21: Necesidades de los interesados y usuarios
Fuente: propia (Metodología RUP.)

4.2.1.3.6 Alternativas y Competencias

Adquirir un sistema desarrollado por empresas de Desarrollo de Software.

Se ha mostrado interés en buscar alternativas externas para solucionar los diversos requerimientos, pero no existen herramientas que establezcan las verdaderas necesidades de las Gobernación.

4.2.1.4 Vista General del Producto

Esta sección provee información detallada de las funciones del sistema a implantar y de las interfaces con otras aplicaciones existentes.

4.2.1.4.1 Perspectiva del Producto

La perspectiva del producto indica los módulos de los cuales estará compuesto el sistema y el orden de funcionamiento.

El sistema de Acopio de Información y Reporte de Medios de comunicación para la Gobernación de Imbabura (Ministerio del Interior) es diseñado a partir de las múltiples necesidades y con el fin de automatizar y mejorar los procesos de trabajo en conjunto entre instituciones de la Provincia o GAD's con los que se establezcan relaciones para el beneficio de la ciudadanía Imbabureña



Ilustración 18: Prospectiva del Producto.
Fuente: Propia

4.2.1.4.2 Resumen de capacidades

Beneficios para el usuario	Características que lo Soportan
Administración Centralizada	Al ser una aplicación web, solo necesita ser actualizada del lado del servidor. Los clientes no requieren de Instaladores o recursos adicionales.
El registro y envío de la información se realiza con rapidez.	Debido a su facilidad de manejo y entendimiento se podrá ingresar la información con mayor rapidez.
Fácil uso	Interfaz fácil y amigable

El acceso a la información y acceso a consultas se realizan con mayor facilidad.	Al contar con un sistema computacional, el tiempo de respuesta a una consulta se mejora Brindará diferentes reportes y funciones de consulta.
Acceso desde varios Navegadores Web.	Soportada por las versiones más comunes de navegadores como son Firefox, Internet Explorer, Google Chrome y Safari.
Acceso desde varios sistemas operativos.	El diseño del sistema está basado en tecnologías como, HTML, CSS, JAVA SCRIPT, ETC lo que hacen de la aplicación funcional, independientemente del sistema operativo instalado.
Mejor Productividad	Al automatizar los procesos de articulación se podrá trabajar como mayor prontitud y de una manera más organizada y eficiente.
Datos Actualizados.	Gracias a la Aplicación se tendrá a mano una fuente de consulta instantánea de los proyectos realizados por las instituciones, así como también una fuente de consulta de los principales medios de Comunicación de la Provincia.

Tabla 22: Resumen de capacidades.
Fuente: Propia

4.2.1.4.3 Suposiciones y Dependencias

Para el correcto uso e implementación del sistema AIRCOM, la Gobernación de Imbabura deberá contar con los recursos necesarios para que todas las instituciones tengan acceso a la aplicación y se pueda establecer el normal flujo de trabajo en conjunto.

4.2.1.4.4 Costos y Precios

Costos de Hardware

Descripción	Costo Institución\$	Costo Real \$
Servidor HPDL380G7 E5640	4500	0
Router Inalámbrico.	60	0
10 Puntos de red	100	0
Impresora HP	100	0
Total de Hardware	4760	0

Costos de Software

Descripción	Costo Institución\$	Costo Real \$
Internet	200	100
Motor de Base de Datos	0	0
Servidor de Aplicaciones	0	0
IDE de Desarrollo	0	0
Hosting	200	0
Total de Software	400	100

Materiales de Oficina

Descripción	Costo Institución \$	Costo Real \$
Resmas hojas de papel bond	20	20
Copias (Documentos y Libros)	30	30
DVD	10	10
Esferos, clips, carpetas, etc	20	20
Flash Memory	20	20
Total de Materiales de Oficina	100	100

Varios

Descripción	Costo Institución \$	Costo Real \$
Movilización	50	50
Capacitación	80	40
Imprevistos	50	50
Empastado y Anillado	100	100
Infraestructura Física	0	0
Imprevistos	260	68
Total de Varios	540	308

Total Costos	5800 \$	508 \$
---------------------	----------------	---------------

Tabla 23: Costos del Proyecto.
Fuente: Propia

4.2.1.4.5 Licenciamiento e Instalación

LICENCIAMIENTO: Dado que se usará software libre las herramientas utilizadas para desarrollar el sistema no requiere ningún tipo de licencia. La Herramienta EXTJS utilizada en la parte del cliente del sistema tiene un licenciamiento de tipo **GPL license v3**, esta licencia garantiza una serie de libertades al usuario, como el libre uso y adaptación del código fuente del programa, obligando a que las modificaciones de este código estén disponibles para el resto de usuarios si se redistribuye el programa.

INSTALACIÓN: Por ser una aplicación web para la parte del servidor la instalación será centralizada ya sea en un Hosting o en el Servidor HP que la Gobernación planea comprar para sus diversas aplicaciones. La instalación del Sistema será realizada por el desarrollador del sistema.

4.2.1.5 Características del producto

4.2.1.5.1 Facilidad de acceso y Uso

AIRCOM será desarrollado utilizando tecnología Web, desarrollando la parte visual con EXTJS optimizando de esta forma el rendimiento y facilidad de uso por parte de los usuarios, creando pantallas con una presentación amigable gracias a íconos intuitivos, tipografía legible y un entorno de fácil interpretación.

4.2.1.5.2 Unificación de la Información

Uno de los principales objetivos del sistema AIRCOM es automatizar y registrar los procesos de generación de información de las Diferentes Instituciones hacia la Gobernación y proporcionar herramientas con criterios de búsqueda parametrizables, ahorrando mucho tiempo a la hora de obtener la información.

4.2.1.5.3 Conexión Fácil y Dinámica

El usuario solo introduce el nombre y su contraseña y se desplegará su cuenta de acuerdo al rol establecido para el tipo de usuario previamente definido.

4.2.1.5.4 Base de datos Centralizada

Ya que se trata de un sistema integrado para la Gobernación de Imbabura en el cual intervendrán varias Instituciones, se hará uso de una sola base de datos la cual se debe instalar, ya que la institución no cuenta con esta tecnología al momento, permitiendo así guardar la información en un solo punto central.

4.2.1.6 Rangos de Calidad

La implementación del sistema se realizará bajo los parámetros de calidad de la Metodología de Desarrollo de Software RUP, contemplando los parámetros que la metodología define para asegurar la producción de software de alta calidad y que se ajuste a las necesidades de los usuarios finales a tiempo.

4.2.1.7 Precedencia y Prioridad

Las prioridades del sistema son:

- a) Realizar de forma correcta los procesos de envío de Información.
- b) Mantener la información unificada y congruente.
- c) Interactuar entre los diferentes módulos del sistema.
- d) Proporcionar una herramienta que tenga una interfaz amigable con el usuario.
- e) Desarrollar un sistema multiusuario con el respectivo control de acceso y privilegios a cada usuario de acuerdo al rol que éste tenga.
- f) Desarrollar métodos fáciles de búsqueda de información.

4.2.1.8 Otros requerimientos del Producto

4.2.1.8.1 Sistema Multiplataforma

Se utiliza en su totalidad la tecnología web, por lo que se puede acceder al sistema mediante cualquier navegador web que soporte tecnologías de este tipo como son HTML, CSS, Java Script, Ajax, etc.

4.2.1.8.2 Acceso a Internet

Cada institución en la que se haya definido aún responsable Institucional para el trabajo conjunto con la Gobernación deberá contar únicamente con acceso al internet.

4.2.1.8.3 Servidor Web

El servidor de aplicaciones definido para la aplicación será Apache que es un servidor de código abierto basado en plataformas Unix.

4.2.1.8.4 Sistemas Operativos

El Sistema Operativo definido para el Servidor es Centos en la versión 5.4, el resto de usuarios podrá acceder a la aplicación desde cualquier sistema operativo que cuente con un navegador web soportado para la aplicación.

4.2.1.8.5 Protocolo de Comunicaciones

El protocolo de comunicación entre el servidor y sus clientes es HTTP.

4.2.1.8.6 Plataforma de Redes

Se establece para la aplicación el protocolo TCP/IP

4.2.2 Plan de Desarrollo de Software

4.2.2.1 Introducción

Este Plan de Desarrollo del Software es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al proyecto AIRCOM para la Gobernación de Imbabura. Este documento presenta una visión general del enfoque de desarrollo propuesto. El desarrollo del sistema AIRCOM utilizará la metodología de unificación de procesos, la misma que permitirá utilizar las normas para definir el proyecto y de esta manera organizar el desarrollo la documentación generada. El enfoque del desarrollo constituye una configuración del proceso RUP de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables). Este documento es a su vez uno de los artefactos de RUP.

4.2.2.1.1 Propósito

El propósito del Plan de Desarrollo de Software es proporcionar la información necesaria para definir, planificar y controlar el desarrollo de un sistema admisible. En él se describe el enfoque de desarrollo del software.

Los usuarios del plan de desarrollo de software son:

- a) La coordinación del proyecto, quien utiliza el presente plan, para organizar la agenda, necesidades de recursos y para realizar seguimiento.
- b) Los miembros del equipo de desarrollo que ejecutan y desarrollan el contenido de la planificación del cronograma, lo usan para entender ¿qué deben hacer?, ¿cuándo deben hacerlo? Y ¿Qué otras actividades dependen de ello?

4.2.2.1.2 Alcance

El Plan de Desarrollo del Software describe el plan global usado para el desarrollo del sistema AIRCOM.

El detalle de las iteraciones individuales se describe en los planes de cada iteración, documentos que se aportan en forma separada.

Durante el proceso de desarrollo en el artefacto “Visión” se definen las características del producto a desarrollar, lo cual constituye la base principal de cada etapa.

Para la versión 1.0 del Plan de Desarrollo del Software, se ha tomado como base en la captura de requisitos por medio del stakeholder en este caso al analista de conflictos de la Gobernación de Imbabura para hacer una estimación aproximada, una vez comenzado el proyecto y durante la fase de Inicio se generará la primera versión del artefacto “Visión”, el cual se utilizará para refinar este documento. Posteriormente, el avance del proyecto y el

seguimiento en cada una de las iteraciones ocasionará el ajuste de este documento produciendo nuevas versiones actualizadas.

4.2.2.2 Vista General del Proyecto

4.2.2.2.1 Propósito, Alcance y Objetivos

La información que a continuación se incluye ha sido extraída de las diferentes reuniones que se han realizado con el stakeholder de la Gobernación de Imbabura desde el inicio del proyecto, Ing. Víctor Hugo Portocarrero.

La Gobernación de Imbabura como entidad Gubernamental perteneciente al Ministerio del Interior, busca mediante el trabajo en conjunto con las instituciones bajo su control, otras Instituciones del Gobierno y GAD's satisfacer las necesidades de la población Imbabureña, asciendo uso de todo tipo de recursos con los que cuenta, así también se hace necesario realizar el proceso de seguimientos de Medios de comunicación de la provincia, para su análisis e interpretación de publicaciones e información generada, por parte de las mismas.

En una institución de estas características es indispensable el uso de las nuevas tecnologías de la información, por ello, la gobernación considera necesario el desarrollo e implementación de un de un sistema que cumpla y satisfaga dichas necesidades.

El proyecto detallará la propuesta para el desarrollo de los módulos, cómo interactúan entre sí, como se conforman para un buen funcionamiento y con ello cumplir los objetivos planteados en el documento "Visión".

Módulo de Administración:

- Administración de usuarios.
- Asignación de roles.
- Asignación de permisos.
- Verificación de integridad de las bases de datos.
- Ingreso y Reparación de datos (Ministerios, sujetos políticos, medios de comunicación y ubicaciones).
- Actualización de los recursos del Sistema.
- Recuperación de contraseñas.

Módulo de Auditoria:

- Visualización de usuarios, fechas, recursos en general en los que se ha creado, actualizado o modificado un registro de cualquier tabla de la Base de Datos.
- Control de integridad de registros.
- Verificación y Seguimiento de Actividades de todos los módulos.

Módulo de Recursos e Infraestructura de la Institución:

- Ingreso y actualización de la Infraestructura y herramientas con que cuenta el departamento o Área.
- Asignación de recursos a Áreas y Departamentos.
- Movimientos de recursos a las diferentes áreas o departamentos.

Módulo de Información y Medios de Comunicación:

- Almacenamiento categorización e indexación de la información enviada por las Instituciones (Tema, Síntesis, Beneficiarios, Responsable, Ubicación, Presupuestos, categorización e Indexación).
- Ingreso y categorización de medios de comunicación (Ubicación, Responsables, tendencia política, espacios contratados, presupuestos asignados a contratación del medio, cobertura del medio).
- Almacenamiento y categorización de recursos enviados por las instituciones (recortes de periódicos, videos, audio).
- Boletines de prensa.
- Almacenamiento y categorización de los medios de comunicación (recortes de periódicos, videos, audio).

Módulo de Instituciones y actores Institucionales:

- Ingreso, Actualización y Modificación de Instituciones (Ministerios, Contactos, direcciones, representantes, ubicaciones y dependencias).
- Ingreso, Actualización y Modificación de actores Institucionales tales como Representantes de instituciones y Directores Provinciales de Ministerios, Jefes Políticos, tenientes políticos y Presidentes de juntas Parroquiales y diversas autoridades.

Módulo de Localización Geográfica:

- Uso de localizaciones geográficas mediante el uso de Google Maps.
- Ubicaciones para el módulo de instituciones y actores o delegados Institucionales.
- Ubicaciones para el módulo de información (donde se generó la información) y para los reportes de medios (ubicación geográfica del medio de comunicación).

Módulo de Reportes e Informes:

- Visualización, impresión y exportación de reportes según especificaciones realizadas de los usuarios (Cliente, Administrador o Auditor) a cualquiera de los módulos anteriores, verificando el acceso y restricciones adecuadas para cada uno de los roles asignados a los usuarios: **Módulo de Recursos e Infraestructura de la Institución, Módulo de Información y Medios de Comunicación, Módulo de localización Geográfica, Módulo de Instituciones y actores Institucionales.**

4.2.2.2.2 Suposiciones y restricciones

Las suposiciones y restricciones respecto del sistema, y que se derivan directamente de las entrevistas con el stakeholder de la “GI” son:

Debe contemplarse las implicaciones de los siguientes puntos críticos:

- a) Seguridad y protección la Información.
- b) Seguridad del envío e intercambio de información
- c) Adaptación a la normativa de Protección de Datos
- d) Adaptabilidad y la facilidad de uso.

La lista de suposiciones y restricciones se incrementará durante el desarrollo del proyecto, particularmente una vez establecido el artefacto “Visión”.

4.2.2.2.3 Entregables del Proyecto

Utilizando la metodología RUP, a lo largo de todo el desarrollo del software se van generando documentos entregables, que son documentos explicativos de cada actividad. A continuación se describen cada uno de los artefactos que serán generados y utilizados por el proyecto.

En la metodología de RUP, todos los documentos entregables, se manejan versiones de cada uno porque pueden ir variando a lo largo de todo el proceso de elaboración del proyecto.

1) Plan de Desarrollo del Software

Es el presente documento.

2) Glosario

Es un entregable que contiene una recopilación de términos poco conocidos o de difícil interpretación con su respectiva explicación o significado.

3) Modelo de Casos de Uso

El modelo de Casos de Uso presenta una descripción de los pasos para llevar a cabo algún proceso o funcionalidad del sistema. Se representa mediante Diagramas de Casos de Uso.

4) Visión

Este documento presenta el enfoque del producto desde la perspectiva del cliente, detallando las necesidades y características del producto.

5) Especificaciones de Casos de Uso

Para aquellos casos de uso que su funcionalidad no sea evidente, se realiza una explicación detallada utilizando un documento plantilla; dicha plantilla incluye según la especificación del caso de uso: flujo básico de eventos, flujo alternativo de eventos, precondiciones y pos condiciones.

6) Prototipos de Interfaces de Usuario

Se trata de prototipos que permiten al usuario hacerse una idea de las interfaces que proveerá el sistema.

7) Modelo de Datos

Este documento describe la representación lógica de la estructura de los datos almacenada en una base de datos, describiendo los tipos de datos y la forma de relacionarse con la demás tablas, así como también las restricciones de integridad.

8) Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba.

9) Lista de Riesgos

Este documento contiene la lista de los riesgos conocidos en el proyecto, organizados descendentemente de acuerdo a la importancia que tenga el riesgo; También contiene acciones específicas de contingencia o mitigación del riesgo.

10) Manual de Instalación

Este documento proporciona una guía detallada de los pasos necesarios para la correcta instalación del sistema.

11) Producto

El sistema AIRCOM se entregará almacenado en un CD con los mecanismos apropiados para su instalación.

12) Actas de Trabajo

Las actas de trabajo contienen las actividades, responsabilidades, observaciones y revisiones que se realicen en el desarrollo del proyecto.

4.2.2.4 Evolución del Plan de Desarrollo de Software

El Plan de Desarrollo del Software se revisará semanalmente y se refinará antes del comienzo de cada iteración.

4.2.2.3 Organización del Proyecto

Se muestra como está estructurado el proyecto indicando participantes, interfaces y roles.

4.2.2.3.1 Participantes del Proyecto

Las personas involucradas en el desarrollo del sistema son las siguientes:

Analistas.- Con experiencia en el entorno de desarrollo del proyecto, con el fin de que los prototipos puedan ser lo más cercanos posibles al producto final.

Programadores.- Los conocimientos que deben tener son: amplio conocimientos en el lenguaje de programación PHP, administración de la base de datos PostgreSQL, diseño UML, metodología RUP.

4.2.2.3.2 Interfaces Externas

La GI proporcionará los requisitos del sistema y los encargados de evaluar los artefactos de acuerdo a cada subsistema.

El equipo de desarrollo interactuará activamente con ellos para la especificación y validación de los artefactos generados.

4.2.2.3.3 Roles y Responsabilidades

A continuación se describen las principales responsabilidades de cada uno de los puestos en el equipo de desarrollo durante las fases de Inicio y Elaboración, de acuerdo con los roles que desempeñan en RUP.

Puesto	Responsabilidad
Analista	Captura, especificación y validación de requisitos, interactuando con el cliente y los usuarios mediante entrevistas. Colaboración en la elaboración de las pruebas funcionales y el modelo de datos.
Programador	Son los responsables del desarrollo del proyecto, desarrollando los módulos del proyecto, creando versiones de los módulos hasta llegar a la versión final.

Tabla 24: Roles y Responsabilidades.
Fuente: Propia (Metodología RUP.)

4.2.2.4 Gestión del Proceso

Se muestra que actividades se realizaran en cada fase de la metodología RUP.

4.2.2.4.1 Plan del Proyecto

En esta sección se presenta la organización en fases e iteraciones y el calendario del proyecto.

4.2.2.4.2 Plan de las Fases

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación muy preliminar)

Fase	Nro. Iteraciones	Duración
Fase de Inicio	1	4 semanas
Fase de Elaboración	1	12 semanas
Fase de Construcción	2	12 semanas
Fase de Transición	1	4 semanas

Tabla 25: Planificación de fases.
Fuente: Propia (Metodología RUP.)

Los hitos que marcan el final de cada fase se describen en la siguiente tabla.

Descripción	Hito
Fase de Inicio	En esta fase desarrollará los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión, previamente establecidos los requerimientos en las actas de trabajo. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente / usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera reléase de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como hito. La primera iteración tendrá como objetivo la identificación

	y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos. Ambas iteraciones tendrán una duración de una semana.
Fase de Construcción	Durante la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el Modelo de Análisis / Diseño. El producto se construye en base a 2 iteraciones, cada una produciendo una release a la cual se le aplican las pruebas y se valida con el cliente / usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la release 3.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para pruebas beta.
Fase de Transición	En esta fase se prepararán dos releases para distribución, asegurando una implantación y cambio del sistema previo de manera adecuada, incluyendo el entrenamiento de los usuarios. El hito que marca el fin de esta fase incluye, la entrega de toda la documentación del proyecto con los manuales de instalación y todo el material de apoyo al usuario, la finalización del entrenamiento de los usuarios y el empaquetamiento del producto.

Tabla 26: Hitos Finales de cada Fase.
Fuente: Propia (Metodología RUP).

4.2.2.4.2.1 Objetivos de las Iteraciones

Fase	Iteración	Descripción	Hitos Asociados	Objetivos
Incepción	Iteración Inicial	Definición del modelo de negocio y plan de proyecto	Actas de Trabajo Documento de Visión.	Clasificar de forma general los requerimientos del Usuario. Determinar el alcance e impacto del proyecto. Determinar la

				factibilidad del proyecto desde el punto de vista del negocio.
Elaboración	Desarrollar Prototipo	Desarrollar el prototipo de la Arquitectura. Definir y diseñar todos los casos de uso.	Documento de Arquitectura.	Definir arquitectura y herramientas. Realizar la lista de riesgos técnicos. Prototipo temprano para revisión de usuarios.
Construcción	Primera Iteración: Desarrollar versión beta del producto.	Implementar y probar los casos de uso para la versión beta.	Versión beta del producto	Implementación de todas las características claves obtenidas de la arquitectura realizada previamente y desde la perspectiva del usuario.
	Segunda Iteración: Desarrollar el reléase inicial.	Implementación y desarrollo de los casos de uso restantes, corregir errores y observaciones de la versión beta.	Producto (Software)	Software revisado por los usuarios involucrados en cada uno de los módulos del sistema.
	Tercera Iteración: Desarrollo del sistema completo.	Corregir y mejorar los defectos del reléase inicial. Desarrollo del sistema completo.	Producto (Software)	Toda la funcionalidad del sistema completa para su liberación.
Transición	Liberar el Software	Empaqueta e instalar el reléase.	Software liberado	Sistema en producción.

Tabla 27: Objetivos de las Iteraciones.
Fuente: Propia (Metodología RUP.)

4.2.2.4.2.2 Calendario del Proyecto

A continuación se presenta un calendario de las principales tareas del proyecto incluyendo sólo las fases de Inicio y Elaboración. Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto.

La siguiente figura ilustra este enfoque, en ella lo ensombrecido marca el énfasis de cada disciplina (workflow) en un momento determinado del desarrollo.

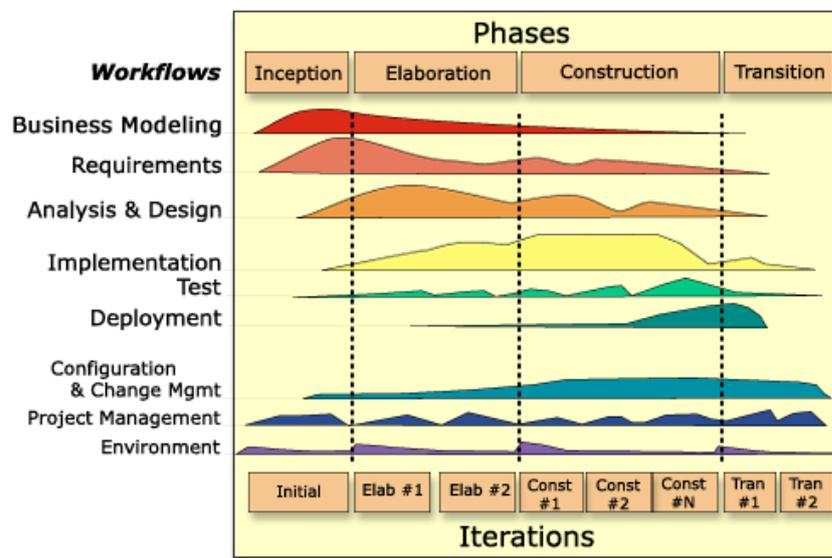


Ilustración 19: Calendario de las principales tareas del proyecto.
Fuente: Metodología RUP.

Para este proyecto se ha establecido el siguiente calendario. La fecha de aprobación indica cuándo el artefacto en cuestión tiene un estado de completitud suficiente para someterse a revisión y aprobación, pero esto no quita la posibilidad de su posterior refinamiento y cambios.

Disciplinas / Artefactos generados o modificados durante la Fase de Inicio	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 1 08/08/2011	Siguiente fase
Requisitos	Semana 1 08/08/2011	Semana 2 15/08/2011
Glosario	Semana 1 08/08/2011	Semana 3 22/08/2011
Visión	Semana 2 15/08/2011	Semana 3 22/08/2011
Modelo de Casos de Uso	Semana 4 29/08/2011	siguiente fase
Especificación de Casos de Uso	Semana 3 22/08/2011	siguiente fase
Análisis / Diseño		
Modelo de Análisis / Diseño	Semana 2 15/08/2011	siguiente fase
Modelo de Datos	Semana 2 15/08/2011	siguiente fase
Implementación		
Modelo de Implementación	Semana 3 22/08/2011	siguiente fase
Pruebas	Transcurso del desarrollo	
Casos de Pruebas Funcionales	Semana 4 29/08/2011	siguiente fase
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 1.0 y planes de las Iteraciones	Semana 1 08/08/2011	Semana 3 22/08/2011
Ambiente	Durante todo el proyecto	

Tabla 28: Calendario Fase de Inicio.
Fuente: Propia (Metodología RUP.)

Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración	Comienzo	Aprobación
Modelado del Negocio		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	Semana 1 08/08/2011	aprobado
Requisitos		
Glosario	Semana 1 08/08/2011	aprobado
Visión	Semana 2 15/08/2011	aprobado
Modelo de Casos de Uso	Semana 3 22/08/2011	Semana 5 05/09/2011
Especificación de Casos de Uso	Semana 3 22/08/2011	Semana 5 05/09/2011
Análisis / Diseño		
Modelo de Datos	Semana 2 15/08/2011	Revisar en cada iteración
Implementación		
Modelo de Implementación	Semana 3 22/08/2011	Revisar en cada iteración
Pruebas		
Casos de Pruebas Funcionales	Semana 3 22/08/2011	Revisar en cada iteración
Gestión de Cambios y Configuración	Durante todo el proyecto	
Gestión del proyecto		
Plan de Desarrollo del Software en su versión 2.0 y planes de las Iteraciones	Semana 4 29/08/2011	Revisar en cada iteración
Ambiente	Durante todo el proyecto	

Tabla 29: Calendario Fase de Elaboración.
Fuente: Propia (Metodología RUP.)

4.2.2.4.3 Seguimiento y control del Proyecto

4.2.2.4.3.1 Gestión de Requisitos

Los requisitos del sistema son especificados en el artefacto Visión. Cada requisito tendrá una serie de atributos tales como importancia, estado, iteración donde se implementa, etc. Estos atributos permitirán realizar un efectivo seguimiento de cada requisito. Los cambios en los requisitos serán gestionados mediante una Solicitud de Cambio, las cuales serán evaluadas y distribuidas para asegurar la integridad del sistema y el correcto proceso de gestión de configuración y cambios.

4.2.2.4.3.2 Control de Plazos

El calendario del proyecto tendrá un seguimiento y evaluación semanal por el jefe de proyecto y por la persona encargada del seguimiento y control asignada en la gobernación.

4.2.2.4.3.3 Control de Calidad

Los defectos detectados en las revisiones y formalizados también en una Solicitud de Cambio tendrán un seguimiento para asegurar la conformidad respecto de la solución de dichas deficiencias. Para la revisión de cada artefacto y su correspondiente garantía de calidad se utilizarán las guías de revisión y listas de verificación incluidas en RUP.

4.2.2.4.3.4 Gestión de Riesgos

A partir de la fase de Inicio se mantendrá una lista de riesgos asociados al proyecto y de las acciones establecidas como estrategia para mitigarlos o acciones de contingencia. Esta lista será evaluada al menos una vez en cada iteración.

4.2.2.4.3.5 Gestión de Configuración

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. También se incluirá la gestión de las Solicitudes de Cambio y de las modificaciones que éstas produzcan, informando y publicando dichos cambios para que sean accesibles a todo los participantes en el proyecto. Al final de cada iteración se establecerá una baseline (un registro del estado de cada artefacto, estableciendo una versión), la cual podrá ser modificada por Solicitud aprobada.

4.2.2.5 Referencias

- a) Diseño, desarrollo e implementación de un sistema contable para la empresa cronos consultora especializada S.C.C, MERLO ROSAS MARIO ROBERTO; RECALDE SALAZAR, FERNANDO DANIEL
- b) Diseño e implementación de un sistema de facturación de agua potable, para las parroquias rurales del cantón Otavalo, SUAREZ BENAVIDEZ, MARÍA ELENA
- c) Sistema web para la gestión del recurso humano de la Dirección de Educación de Imbabura SG-RRHH, mediante la utilización del Framework Symfony, ACOSTA TARAMUEL, ELSI GABRIELA.
- d) Documentación de Rational Unified Process, manuales de ayuda, tutoriales, etc.

4.2.3 Actas de Trabajo

En las actas de trabajo se registraron los requerimientos especificados en cada una de las visitas realizadas a la gobernación para el levantamiento de procesos e información.

4.3 FASE DE ELABORACIÓN

En esta fase se realiza el diseño básico de la arquitectura a implementar y se realiza los casos de uso con su respectiva descripción de cada uno de ellos.

4.3.1 Arquitectura

A continuación se presenta el documento de arquitectura generado por la metodología RUP, el mismo que se describe la forma general del sistema.

4.3.1.1 Introducción

4.3.1.1.1 Propósito

El presente documento presenta una vista de alto nivel de la arquitectura del sistema de Acopio de Información Interinstitucional y Reporte de Medios de Comunicación (AIRCOM), objetivos, restricciones, los casos de uso más relevantes, los patrones de arquitectura aplicados y las principales decisiones sobre el diseño del sistema.

4.3.1.1.2 Alcance

El alcance de la arquitectura es definir los componentes arquitectónicos del sistema para tener una visión clara del sistema que describa los aspectos funcionales del mismo.

4.3.1.2 Representación de la Arquitectura

El presente documento presenta la arquitectura como una serie de vistas; vista de casos de uso, vista de procesos, vista de despliegue y vista de implementación. Los modelos han sido desarrollados usando el Edraw Max, Excel, JDeveloper.

4.3.1.3 Objetivos y Restricciones de la Arquitectura

Existen requerimientos y restricciones de relevancia para la definición de la arquitectura:

- ✚ Diseño basado en capas de propósito claro y con alto grado de cohesión.
- ✚ Desacoplamiento entre capas que permita el fácil reemplazo de los mismos.
- ✚ Capas altamente reutilizables.
- ✚ Todos los requerimientos descritos en el documento de Visión deben ser tomados en consideración para el desarrollo de la arquitectura definida.

4.3.1.4 Vista de Casos de Uso

Los casos de uso y los procesos de cada módulo ayudan a ver claramente cuál será el funcionamiento del sistema.

4.3.1.4.1 Procesos

4.3.1.4.1.1 Proceso Modulo de Administración y Seguridad

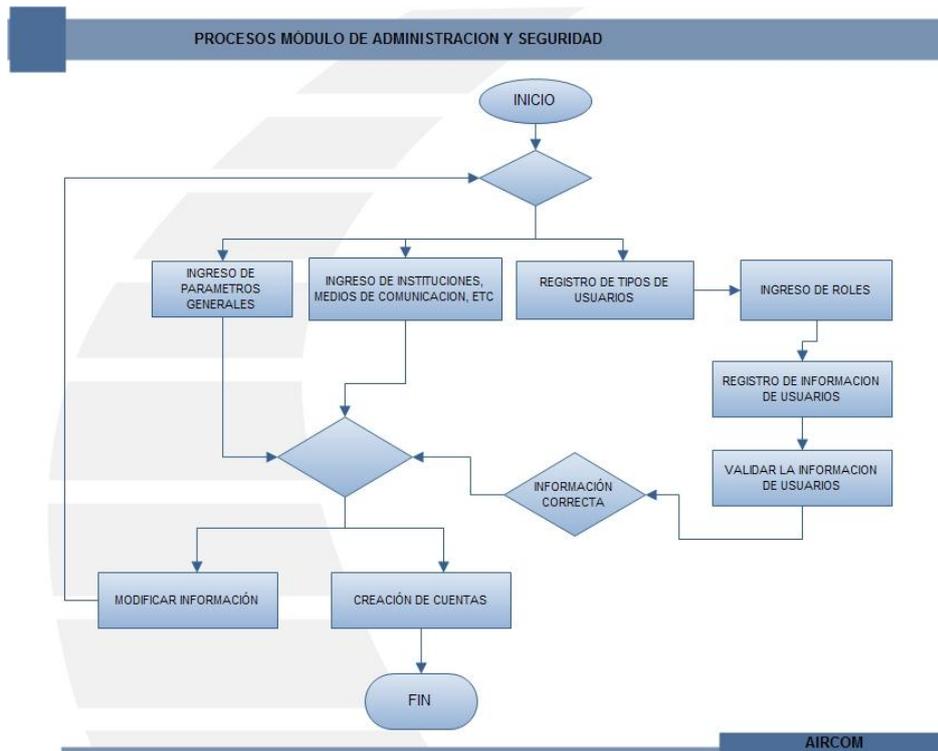


Ilustración 20: Procesos del Módulo Administración y Seguridad.

Fuente: Propia

4.3.1.4.1.2 Proceso Modulo de Auditoria

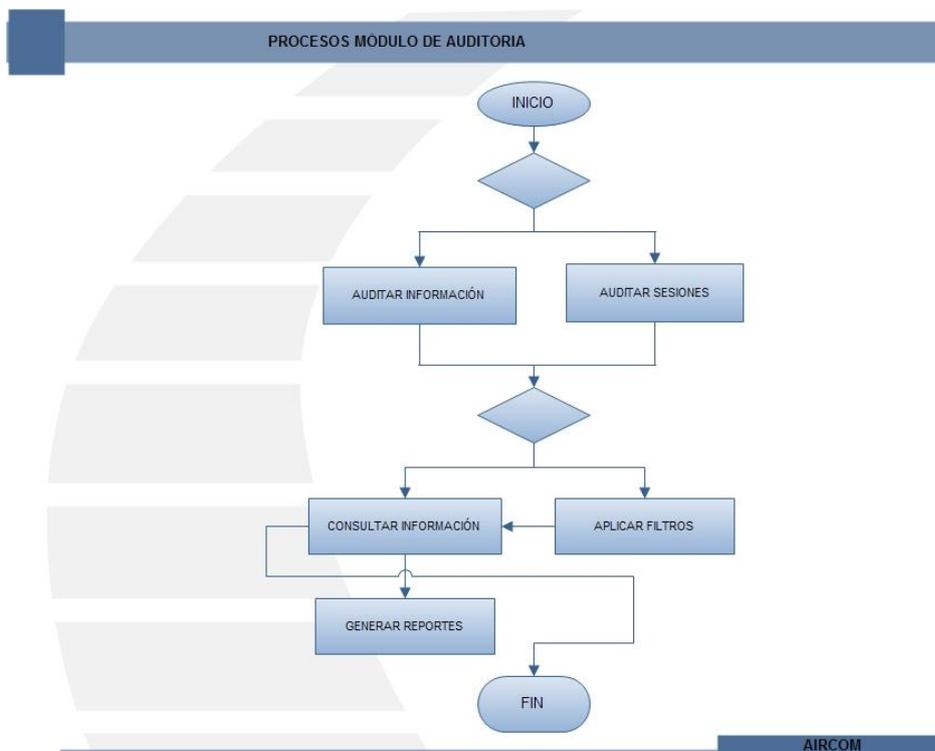


Ilustración 21: Proceso Modulo de Auditoria.

Fuente: Propia.

4.3.1.4.1.3 Módulo de Recursos e Infraestructura de la Institución

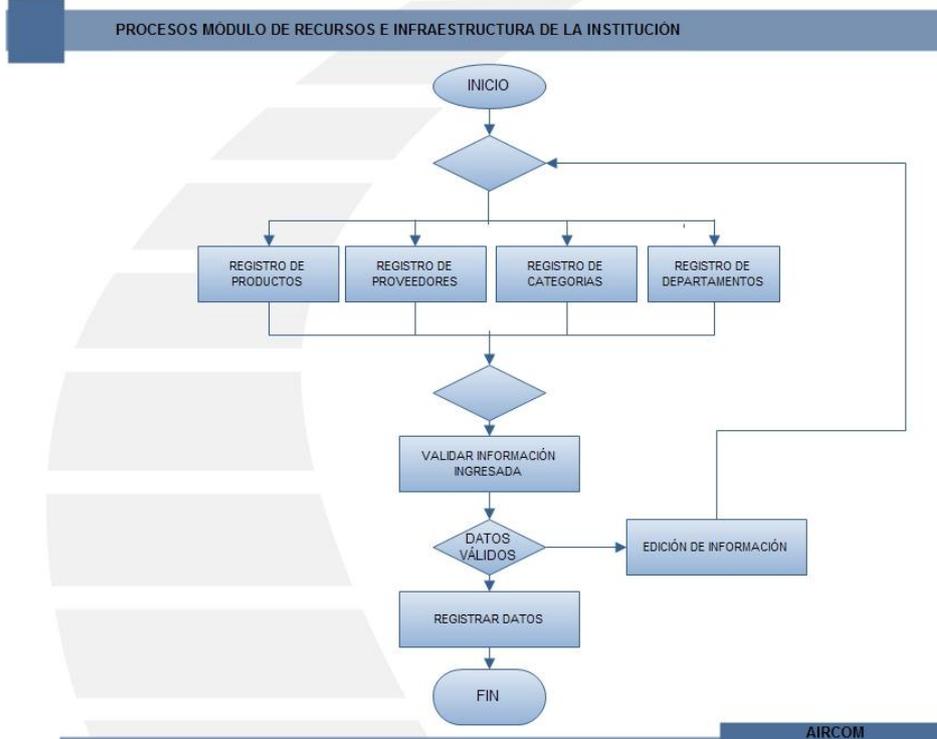


Ilustración 22: Módulo de Recursos e Infraestructura de la Institución.
Fuente: Propia.

4.3.1.4.1.4 Módulo de Información y Medios de Comunicación:

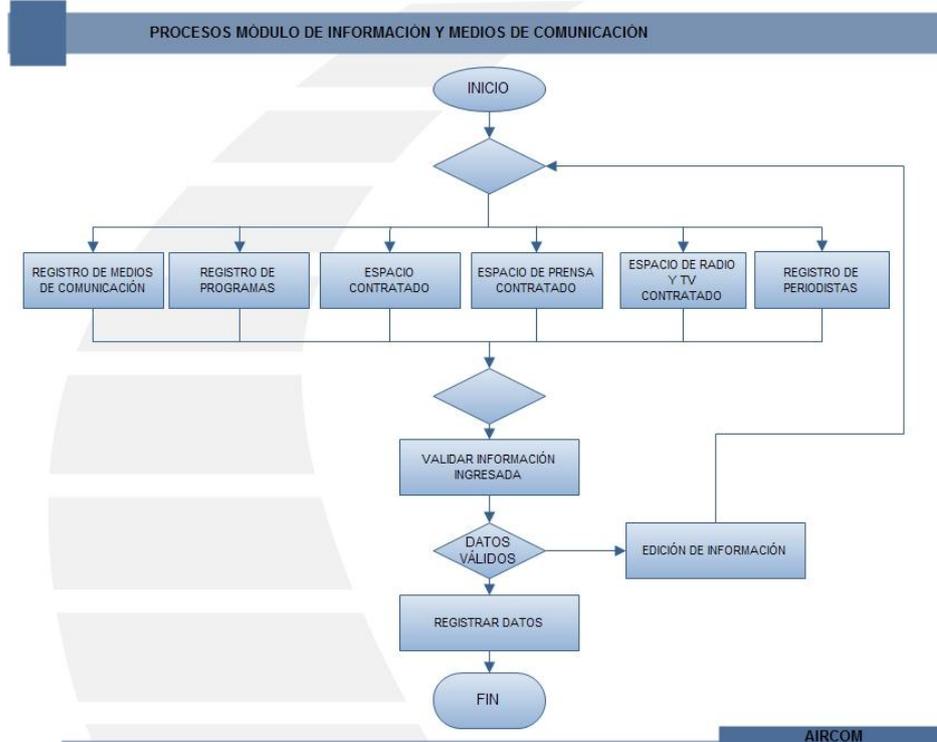


Ilustración 23: Módulo de Información y Medios de Comunicación.
Fuente: Propia.

4.3.1.4.1.5 Módulo de Instituciones y Actores Institucionales

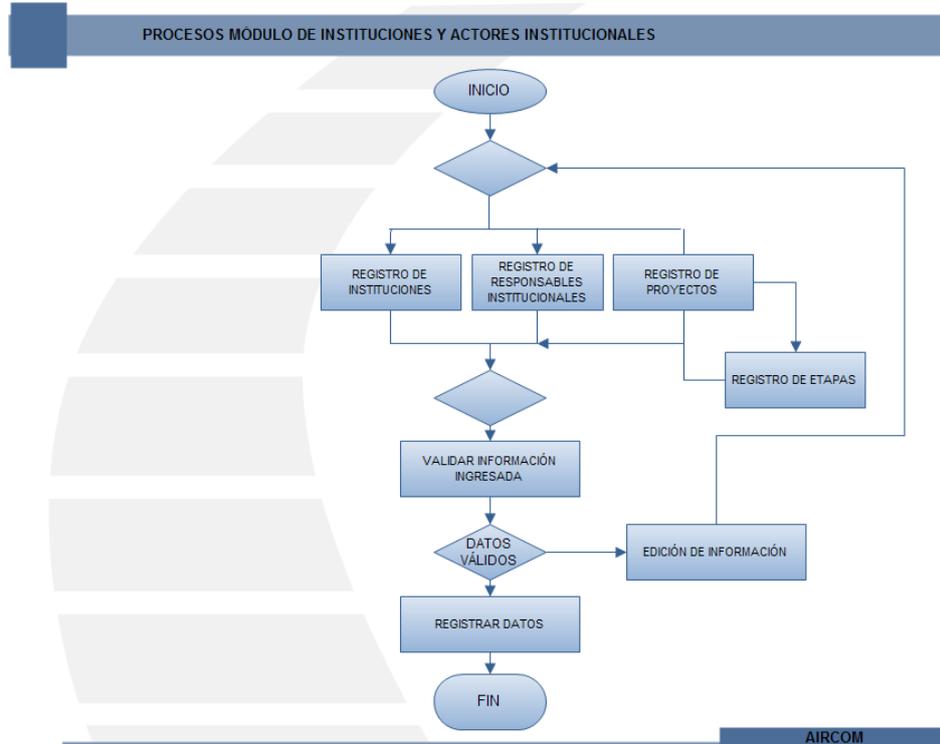


Ilustración 24: Módulo de Instituciones y Actores Institucionales
Fuente: Propia.

4.3.1.4.1.6 Módulo de Localización Geográfica

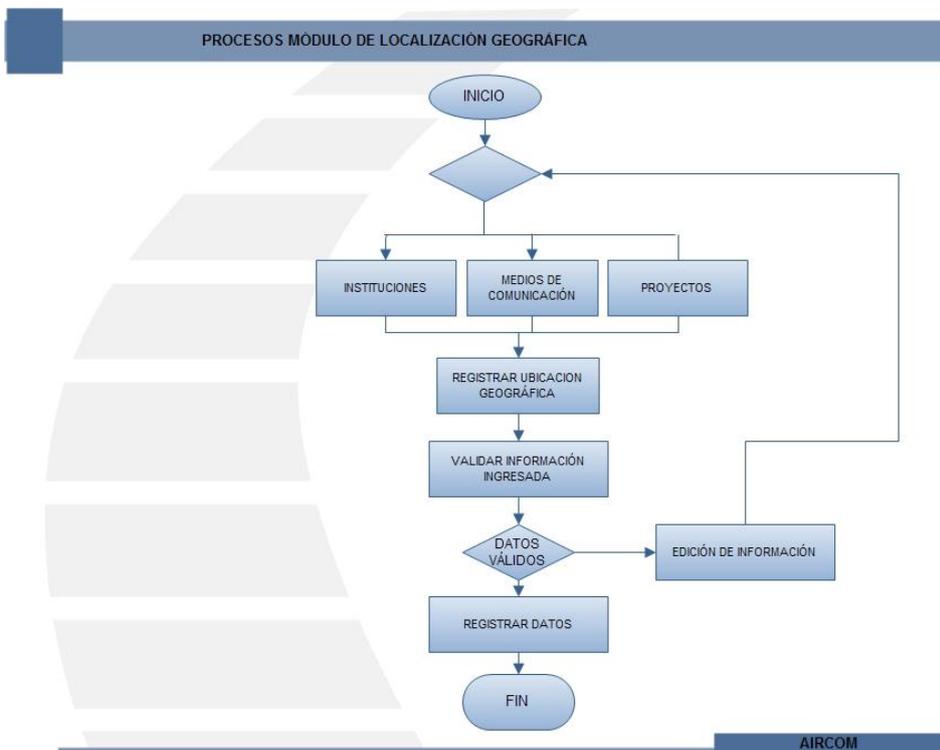


Ilustración 25: Módulo de Localización Geográfica.
Fuente: Propia.

4.3.1.4.1.7 Módulo de Reportes e Informes.

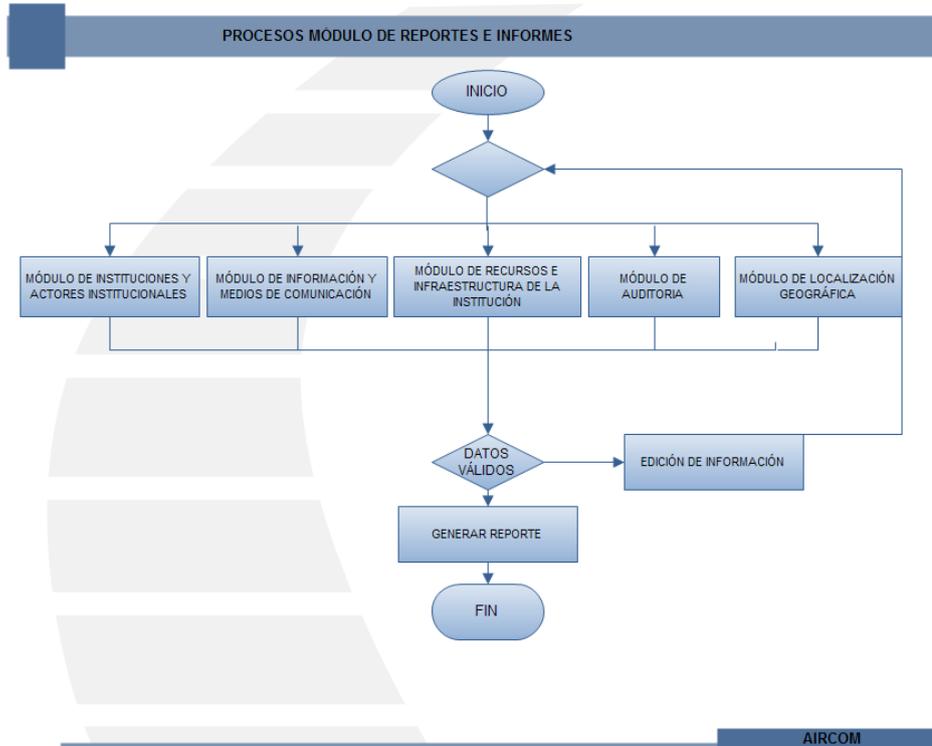


Ilustración 26: Módulo de Reportes e Informes.
Fuente: Propia.

4.3.1.4.2 Actores

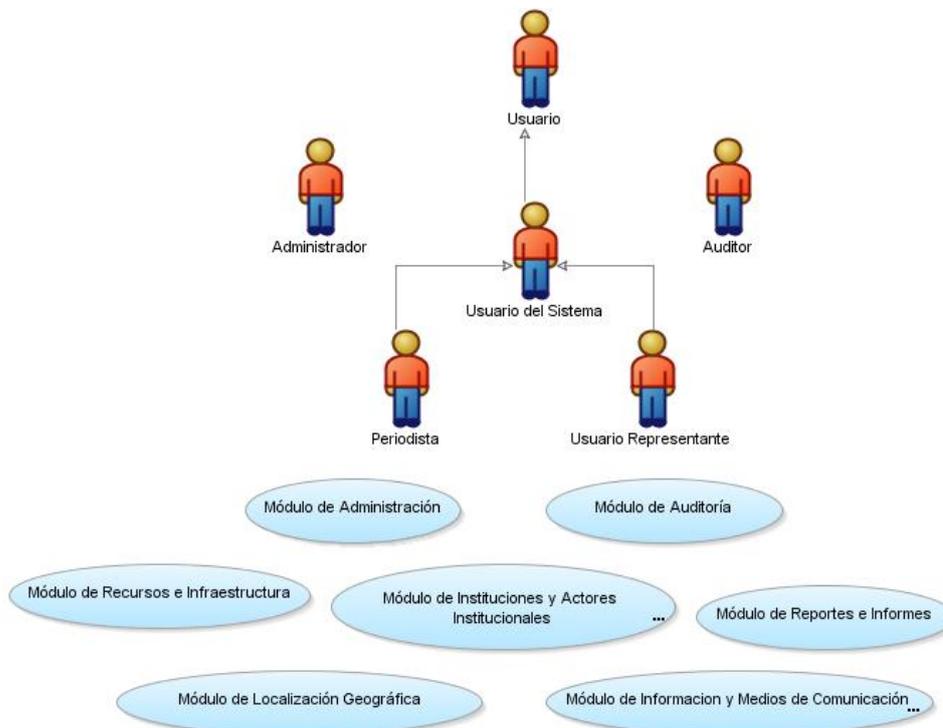


Ilustración 27: Actores.
Fuente: Propia

Nro.	Stakeholder	Descripción
1	Usuario	Usuario general del sistema que puede ingresar requerimientos y consultar la base de datos
2	Administrador	Administrador del sistema quién gestiona los objetos y recursos del sistema
4	Usuario Auditor	Realiza consultas de las operaciones realizadas por los usuarios, (ingreso, modificación, eliminación) de la información del sistema.
5	Usuario Periodista	Procesa y registra toda la información generada en los principales medios de comunicación.
6	Usuario Representante.	Registra la información generada en las diversas Instituciones Gubernamentales.

Tabla 30: Descripción de Clientes.
Fuente: Propia

4.3.1.4.3 Modelos de Casos de Negocio

EL modelo de casos de negocio presenta una visión global del sistema desde el punto de vista del negocio.

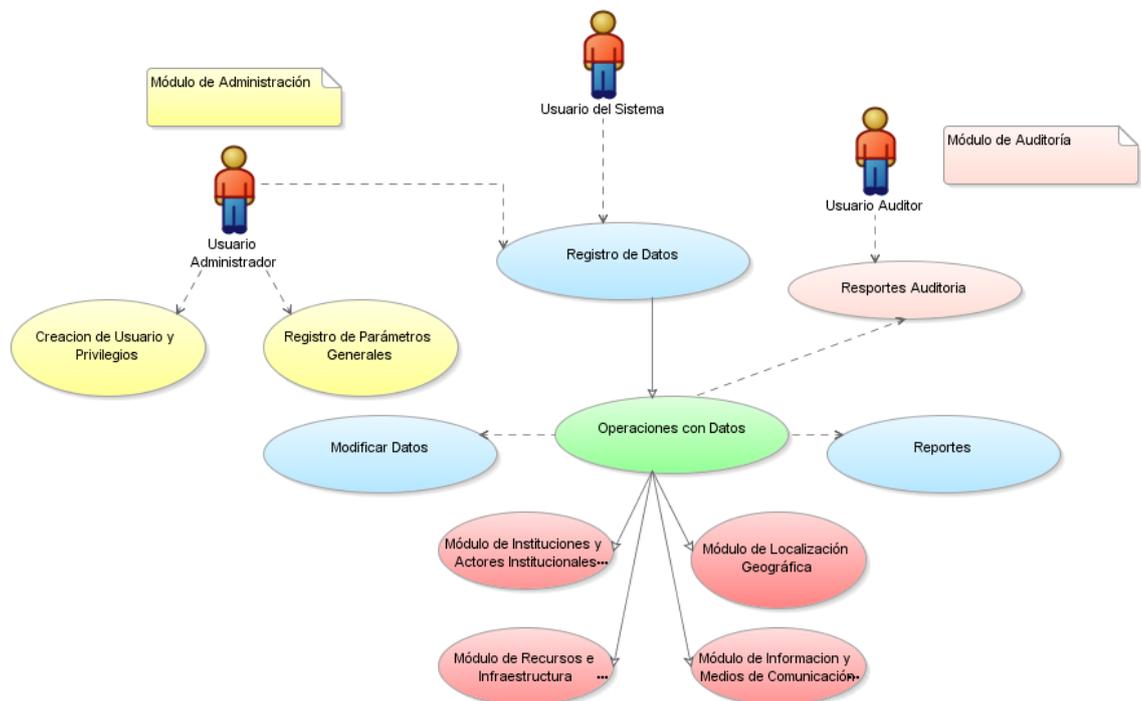


Ilustración 28: Modelos de Casos de Negocio.
Fuente: Propia

4.3.1.4.4 Modelos de Casos de Uso

En este apartado se muestran los diagramas de caso de uso del sistema, de acuerdo a la especificación de los requerimientos, los cuales permiten mostrar la funcionalidad central del sistema en forma detallada.

4.3.1.4.4.1 Caso de Uso Administración del Sistema

A continuación se presentan los casos de uso para el actor Auditor relacionados con el sistema y que influyen en la arquitectura.

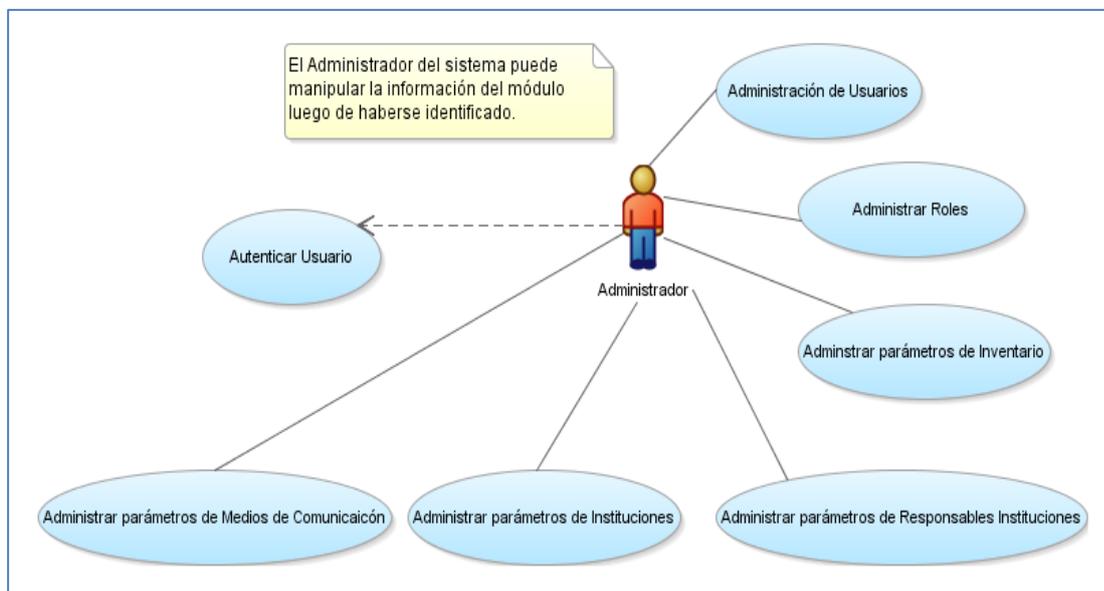


Ilustración 29: Casos de Uso del Módulo de Administración y Seguridad.
Fuente: Propia

CASO DE USO	DESCRIPCIÓN
CREAR Y ADMINISTRAR USUARIOS	Permite al Administrador crear cuentas para nuevos usuarios, cuentas con las cuales pueden acceder al sistema, ofrece la funcionalidad de eliminar, modificar y crear cuentas de usuario además de ver el listado de cuentas creadas.
CREAR Y ADMINISTRAR ROLES	Permite asignar un conjunto de permisos a un usuario, los cuales especifican a su poseedor si dispone de acceso a una determinada función del sistema o a una parte de la interfaz de usuario del sistema (modulo).
CREAR Y ADMINISTRAR MEDIOS DE COMUNICACIÓN	Permite al usuario administrador crear, consultar y verificar los medios de comunicación y a los periodistas que pertenecen a ellas, para

	posteriormente integrarlos a los reportes de prensa y reportes de programa.
CREAR Y ADMINISTRAR INSTITUCIONES	Permite al usuario Administrador crear, consultar y verificar las Instituciones que están sujetas al envío de la información para el trabajo conjunto a realizar con la Gobernación.
CREAR Y ADMINISTRAR RESPONSABLES INSTITUCIONALES	Permite registrar la información detallada de los representantes Institucionales, datos personales, fotografía e información relevante acerca de su cargo.
CREAR Y ADMINISTRAR INVENTARIO	Permite al usuario Administrador llevar un registro de los recursos con los que cuenta la Gobernación de Imbabura.

Tabla 31: Caso de Uso Administración del Sistema.
Fuente: Propia

4.3.1.4.2 Caso de Uso Auditoria de Sistema

A continuación se presentan los casos de uso para el actor Administrador relacionados con el sistema y que influyen en la arquitectura.

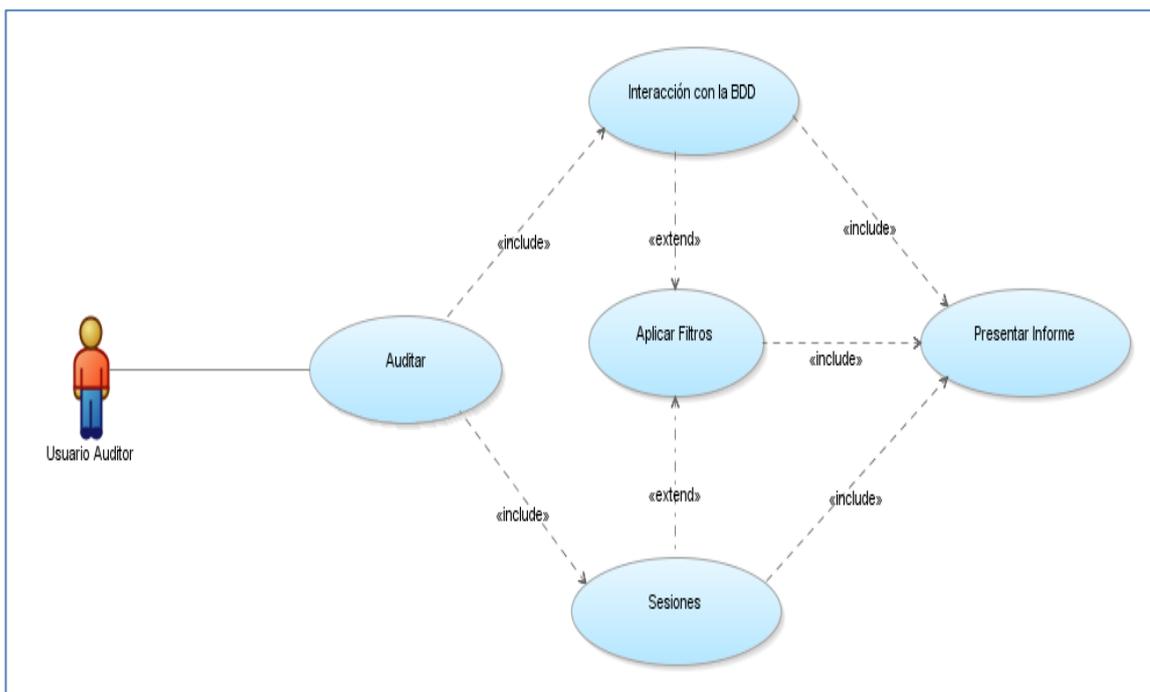


Ilustración 30: Casos de uso del módulo Auditoría Informática.
Fuente: Propia.

CASO DE USO	DESCRIPCIÓN
CONSULTAS POR USUARIO, FECHA DE MODIFICACIÓN, TIPO DE MODIFICACIÓN, NOMBRE DE LA TABLA, Y NOMBRE DE LA COLUMNA.	Permite al usuario Auditor consultar y verificar las modificaciones y operaciones realizadas por el resto de usuarios en los principales módulos del sistema.
EXPORTAR CONSULTA	Permite al usuario Auditor exportar los datos consultados según previas especificaciones.

Tabla 32: Caso de Uso Auditoria de Sistema.
Fuente: Propia.

4.3.1.4.3 Caso de Uso Periodista

A continuación se presentan los casos de uso para el actor Periodista relacionados con el sistema y que influyen en la arquitectura para el proceso de reporte de Prensa.

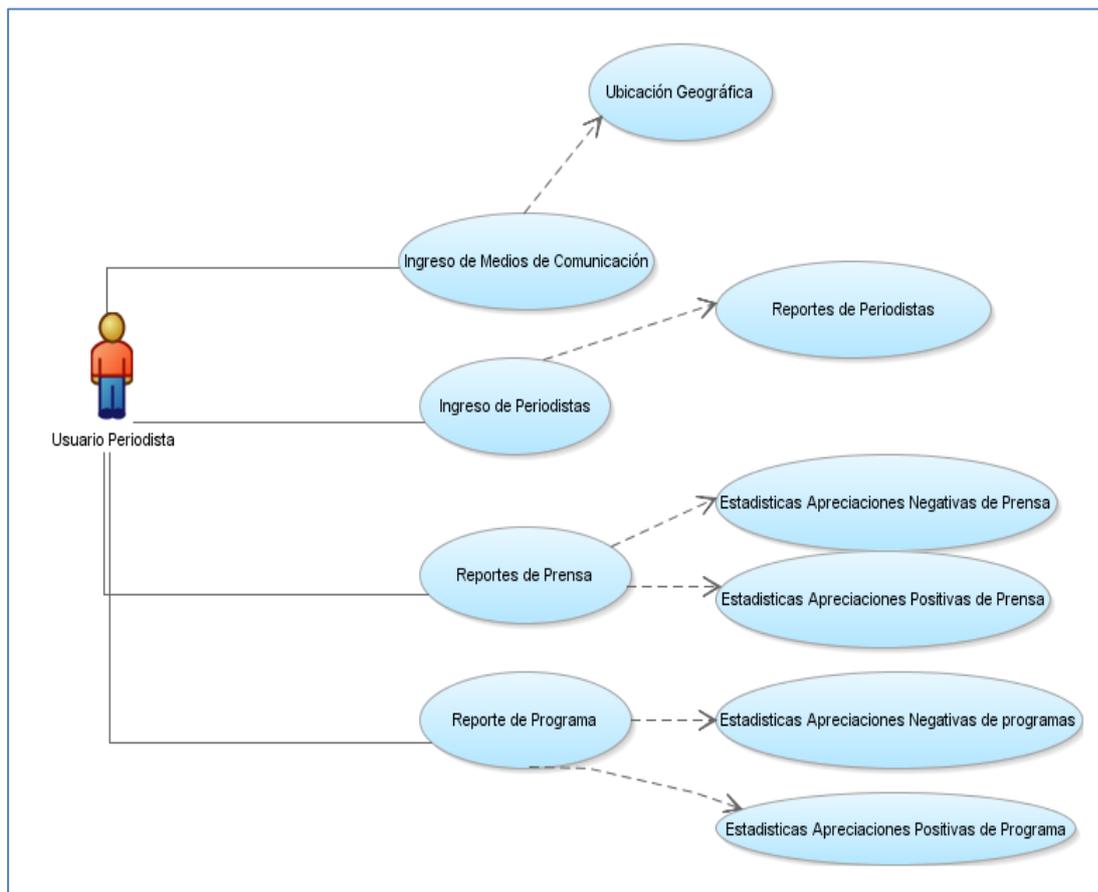


Ilustración 31: Caso de Uso Periodista.
Fuente: Propia

CASO DE USO	DESCRIPCIÓN
CREAR Y ADMINISTRAR MEDIOS DE COMUNICACIÓN Y PERIODISTAS	Permite al usuario Periodista crear, consultar y verificar los medios de comunicación y a los periodistas que pertenecen a ellas, para posteriormente integrarlos a los reportes de prensa y reportes de programa.
ENVIAR REPORTES DE PRENSA	Permite registrar la información detallada de lo expuesto por los principales medios de comunicación de prensa referente a las Instituciones y a los representantes de dichas Instituciones.
CONSULTAR ESTADÍSTICAS DE APRECIACIONES NEGATIVAS	Presenta un gráfico estadístico dinámico en el cual se muestra el número de apreciaciones negativas y el detalle de los mismos que se han registrado en el sistema.
CONSULTAR ESTADÍSTICAS DE APRECIACIONES POSITIVAS	Presenta un gráfico estadístico dinámico en el cual se muestra el número de apreciaciones positivas y el detalle de los mismos que se han registrado en el sistema.
ENVIAR REPORTES DE PROGRAMA	Permite registrar la información detallada de lo expuesto por los principales medios de comunicaron de radio y tv referente a las Instituciones y a los representantes de dichas Instituciones.
CONSULTAR ESTADÍSTICAS DE APRECIACIONES NEGATIVAS	Presenta un gráfico estadístico dinámico en el cual se muestra el número de apreciaciones negativas y el detalle de los mismos que se han registrado en el sistema.
CONSULTAR ESTADÍSTICAS DE APRECIACIONES POSITIVAS	Presenta un gráfico estadístico dinámico en el cual se muestra el número de apreciaciones positivas y el detalle de los mismos que se han registrado en el sistema.

Tabla 33: Caso de Uso Periodista.

Fuente: Propia.

4.3.1.4.4 Caso de Uso Usuario Representante Institucional (Cliente)

A continuación se presentan los casos de uso para el actor Cliente relacionados con el sistema y que influyen en la arquitectura.

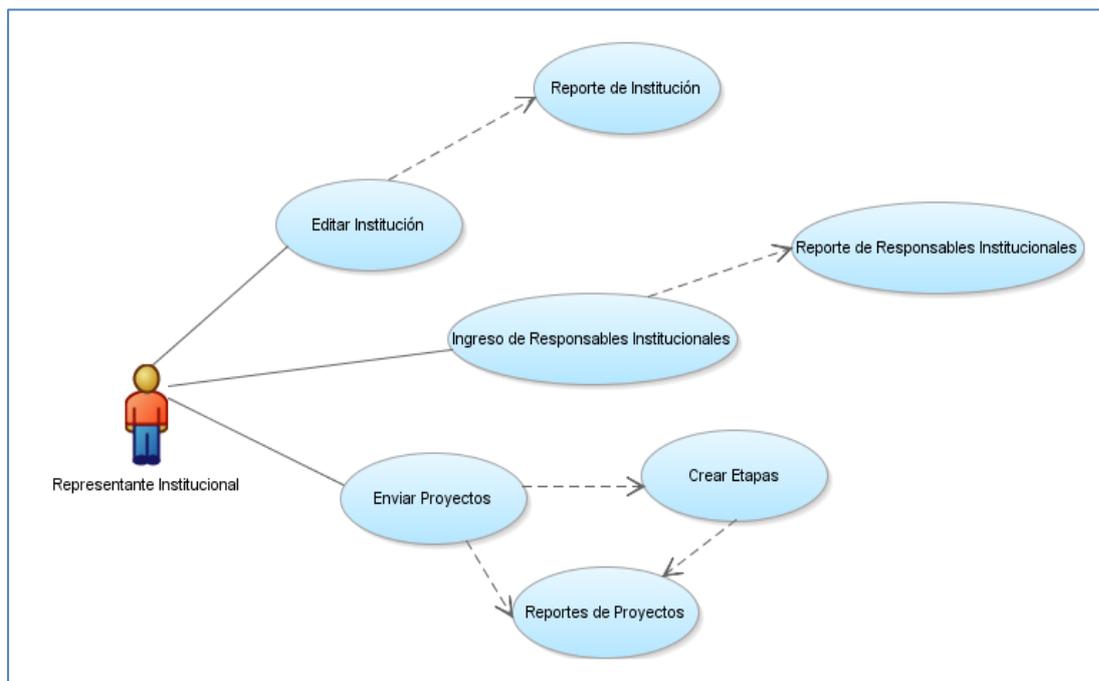


Tabla 34: Casos de uso Usuario Cliente (Responsable Institucional).
Fuente: propia.

CASO DE USO	DESCRIPCIÓN
CREAR Y ADMINISTRAR INSTITUCIONES	Permite al usuario Cliente crear, consultar y verificar las Instituciones que están sujetas al envío de la información para el trabajo conjunto a realizar con la Gobernación.
CREAR Y ADMINISTRAR RESPONSABLES INSTITUCIONALES	Permite registrar la información detallada de los representantes Institucionales, datos personales, fotografía e información relevante acerca de su cargo.
ENVIAR PROYECTOS	Permite registrar o editar un proyecto con ubicaciones geográficas, objetivos del buen vivir con los que se cumple, Instituciones con las que se trabaja y Responsables Institucionales que integran el proyecto.
AGREGAR ETAPAS	Permite agregar etapas de un proyecto, integrando datos como beneficiarios y archivos de evidencia de lo realizado.
EXPORTAR PROYECTOS Y ETAPAS.	Genera un reporte del estado del proyecto y etapas ingresadas en el sistema.

Tabla 35: Casos de uso Usuario Cliente (Responsable Institucional).
Fuente: Propia.

4.3.1.4.5 Especificación de casos de Uso

En esta sección se detallan las principales especificaciones de los casos de uso del sistema:

4.3.1.4.5.1 Asignación de Usuarios al Sistema

Las especificaciones del caso de uso “Asignar Usuarios” son descritas en la siguiente Tabla.

Caso de Uso: Asignación de Usuarios al Sistema	
Actores	Administrador
Descripción	Tiene como función asignar usuarios al sistema para que estos puedan ingresar a los módulos designados.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser Administrador. • El usuario autenticado con éxito en el Sistema. • Ingresar en el menú Usuarios • Seleccionar la opción Lista de Usuarios.
Post condiciones	Caso de Uso: Asignación de Usuarios al Sistema.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema escoge la opción Lista de Usuarios. 2. El sistema muestra la ventana correspondiente <ul style="list-style-type: none"> • Nuevo <ol style="list-style-type: none"> 3. Ingresar el nombre de usuario, email, seleccionar la opción activo si desea que lo este, seleccionar la opción administrador, si desea que sea súper administrador, escoger el tipo de usuario (rol), entre las opciones administrador, auditor, periodista o cliente, ingresar la contraseña y confírmela en los campos correspondientes, seleccione la Institución y el responsable Institucional al que se le asignara esta cuenta. 4. El actor hace clic en el botón Guardar 5. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente. 	
Flujo Alternativo	
<ul style="list-style-type: none"> • Editar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un usuario y hace clic en la opción editar 2. El sistema carga los datos del cliente en una nueva ventana 	

<p>3. El actor modifica los campos y presiona el botón Guardar</p> <ul style="list-style-type: none"> • Eliminar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un usuario. 2. El actor presiona la opción Eliminar 3. El sistema pregunta al actor si realmente desea borrar el registro. 4. El usuario se elimina si el actor presiona el botón aceptar. 5. Se muestra un mensaje de confirmación.
<p>Excepciones</p> <ul style="list-style-type: none"> • Error al editar al Usuario <ol style="list-style-type: none"> 1. Después de ejecutar el paso 3 del Flujo alternativo. 2. El sistema encuentra datos mal ingresados 3. El sistema informa al actor que verifique la información ingresada.

Tabla 36: Perfil del Usuario del Sistema.
Fuente: Propia (Metodología RUP)

4.3.1.4.5.2 Registrar Responsable Institucional

Caso de Uso: Registrar Responsable Institucional	
Actores	Administrador, Cliente
Descripción	Tiene como función asignar nuevos responsables al sistema para que estos puedan ser tomados y formar parte de algún proyecto.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser Administrador o Cliente. • El usuario autenticado con éxito en el Sistema. • Seleccionar la opción Lista de Responsables.
Post condiciones	Caso de Uso: Registrar responsable Institucional.
Flujo Normal de Eventos	

<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema escoge la opción Lista de Responsables. 2. El sistema muestra la ventana correspondiente <ul style="list-style-type: none"> • Nuevo <ol style="list-style-type: none"> 3. Ingresar los Datos Principales, cargar la Fotografía, Ingresar los datos Institucionales, ingresar los Datos Personales y la Síntesis de las Funciones 4. El actor hace clic en el botón Guardar 5. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
<p>Flujo Alternativo</p>
<ul style="list-style-type: none"> • Editar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un responsable y hace clic en la opción editar 2. El sistema carga los datos del responsable Institucional en una nueva ventana 3. El actor modifica los campos y presiona el botón Guardar <ul style="list-style-type: none"> • Eliminar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un responsable. 2. El actor presiona la opción Eliminar 3. El sistema pregunta al actor si realmente desea borrar el registro. 4. El Responsable Institucional se elimina si el actor presiona el botón aceptar. 5. Se muestra un mensaje de confirmación.
<p>Excepciones</p>
<ul style="list-style-type: none"> • Error al editar al Responsable <ol style="list-style-type: none"> 1. Después de ejecutar el paso 3 del Flujo alternativo. 2. El sistema encuentra datos mal ingresados 3. El sistema informa al actor que verifique la información ingresada. <ul style="list-style-type: none"> • Error al eliminar al Responsable <ol style="list-style-type: none"> 1. Después de ejecutar el paso 2 del Flujo alternativo. 2. El sistema informa al usuario que no puede eliminar al Responsable seleccionado. 3. El sistema informa la razón de porque no se puede eliminar al Responsable seleccionado.

Tabla 37: Registrar Responsable Institucional.
Fuente: Propia.

4.3.1.4.5.3 Registrar Institución

Caso de Uso: Registrar Institución	
Actores	Administrador
Descripción	Tiene como función ingresar una Institución al sistema para que estas puedan ser tomadas y formar parte de algún proyecto, reporte o informe.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser Administrador. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Proyectos y opción Listado de Instituciones.
Post condiciones	Caso de Uso: Registrar Institución.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Proyectos escoge la opción Lista de Instituciones. 2. El sistema muestra la ventana correspondiente <ul style="list-style-type: none"> • Nuevo <ol style="list-style-type: none"> 3. Ingresar los Datos Generales de la Institución como Nombre, Tipo, Cobertura, Nivel, Dirección, Teléfono, fax email, Institución de la que depende y el sector al que pertenece. 4. El actor hace clic en el botón Siguiente. 5. El sistema muestra un mapa, y el actor busca y asigna la ubicación de la Institución en el mapa. 6. El actor hace clic en el botón Guardar 7. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente. 	
Flujo Alternativo	
<ul style="list-style-type: none"> • Editar <ol style="list-style-type: none"> 1. El actor selecciona un registro de una Institución y hace clic en la opción editar. 	

<ol style="list-style-type: none"> 2. El sistema carga los datos de la Institución en una nueva ventana 3. El actor modifica los campos y presiona el botón Guardar <ul style="list-style-type: none"> • Eliminar <ol style="list-style-type: none"> 1. El actor selecciona un registro de una Institución. 2. El actor presiona la opción Eliminar 3. El sistema pregunta al actor si realmente desea borrar el registro. 4. La Institución se elimina si el actor presiona el botón aceptar. 5. Se muestra un mensaje de confirmación.
Excepciones
<ul style="list-style-type: none"> • Error al editar la Institución <ol style="list-style-type: none"> 1. Después de ejecutar el paso 3 del Flujo alternativo. 2. El sistema encuentra datos mal ingresados 3. El sistema informa al actor que verifique la información ingresada. <ul style="list-style-type: none"> • Error al eliminar la Institución <ol style="list-style-type: none"> 1. Después de ejecutar el paso 2 del Flujo alternativo. 2. El sistema informa al usuario que no puede eliminar la Institución seleccionada. 3. El sistema informa la razón de porque no se puede eliminar la Institución.

Tabla 38: Registrar Institución
Fuente: Propia.

4.3.1.4.5.4 Enviar Proyecto

Las Especificaciones del caso de uso “enviar Proyecto” son descritas en la siguiente Tabla:

Caso de Uso: Enviar Proyectos	
Actores	Cliente
Descripción	Tiene como función ingresar un nuevo proyecto al sistema.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser cliente. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Proyectos y opción Lista de Proyectos.
Post condiciones	Caso de Uso: Enviar Proyecto.
Flujo Normal de Eventos	

1. El actor ingresa al menú del Sistema en Proyectos escoge la opción Lista de Proyectos.
2. El sistema muestra la ventana correspondiente
- **Nuevo**
3. Ingresar estado del proyecto, objetivo, nombre, síntesis, adjuntar documento, presupuesto, fecha inicial y fecha final y ámbito del proyecto.
4. El actor hace clic en el botón Siguiente.
5. El sistema muestra un mapa, y el actor busca y asigna la ubicación de donde se realizara el proyecto en el mapa.
6. El actor hace clic en el botón Siguiente.
7. El actor agrega los objetivos de buen vivir asciendo clic en la opción + o en la opción – para quitar el objetivo ya agregado.
8. El actor hace clic en el botón Siguiente.
9. El actor agrega las Instituciones que van a intervenir en este proyecto asciendo clic en la opción + o en la opción – para quitar el instituciones ya agregadas.
10. El actor hace clic en el botón Siguiente.
11. El actor agrega a los Responsables Institucionales que van a intervenir en este proyecto asciendo clic en la opción + o en la opción – para quitar el Responsable ya agregado.
12. El actor hace clic en el botón Guardar
13. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.

Flujo Alternativo

- **Editar**
- 1. El actor selecciona un registro de un proyecto y hace clic en la opción editar.
- 2. El sistema carga los datos del proyecto en una nueva ventana
- 3. El actor modifica los campos y presiona el botón Guardar
- **Eliminar**
- 1. El actor selecciona un registro de un proyecto.
- 2. El actor presiona la opción Eliminar.
- 3. El sistema pregunta al actor si realmente desea borrar el registro.
- 4. El proyecto se elimina si el actor presiona el botón aceptar.
- 5. Se muestra un mensaje de confirmación.

Excepciones

- **Error al editar el Proyecto**

1. Después de ejecutar el paso 3 del Flujo alternativo.
2. El sistema encuentra datos mal ingresados
3. El sistema informa al actor que verifique la información ingresada.

- **Error al eliminar el Proyecto**

1. Después de ejecutar el paso 2 del Flujo alternativo.
2. El sistema informa al usuario que no puede eliminar el proyecto seleccionado.
3. El sistema informa la razón de porque no se puede eliminar el proyecto.

Tabla 39: Enviar Proyecto
Fuente: Propia.

4.3.1.4.5.5 Crear una Etapa de Proyecto

Las Especificaciones del caso de uso “Crear una Etapa de Proyecto” son descritas en la siguiente Tabla:

Caso de Uso: Crear una Etapa de Proyecto	
Actores	Cliente
Descripción	Tiene como función agregar una nueva etapa a un proyecto existente en el sistema.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser cliente. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Proyectos y opción Lista de Proyectos.
Post condiciones	Caso de Uso: Crear una Etapa de Proyecto.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Proyectos escoge la opción Lista de Proyectos. 2. El sistema muestra la ventana correspondiente <ul style="list-style-type: none"> • (+) Crear una Etapa • Añadir 3. Ingresar estado del nombre, objetivo, si depende de otra etapa, síntesis, presupuesto, 	

<p>fecha inicial y fecha final</p> <ol style="list-style-type: none"> 4. El actor hace clic en el botón Agregar Beneficiario. 5. Ingresar la descripción y cantidad de beneficiarios 6. El actor hace clic en el botón Agregar /Subir Archivos 7. El actor hace clic en el botón Agregar Archivo de Evidencia 8. Ingresar la descripción y adjuntar el archivo(texto, audio, video) e ingresar la fecha de generación del mismo 9. El actor hace clic en el botón Guardar 10. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
<p>Flujo Alternativo</p> <ul style="list-style-type: none"> • Editar <ol style="list-style-type: none"> 1. El actor selecciona un registro de una Etapa en el lado derecho de la pantalla y hace clic en la opción editar. 2. El sistema carga los datos de la Etapa en una nueva ventana 3. El actor modifica los campos y presiona el botón Guardar • Eliminar <ol style="list-style-type: none"> 1. El actor selecciona un registro de una Etapa. 2. El actor presiona la opción Eliminar 3. El sistema pregunta al actor si realmente desea borrar el registro. 4. La Etapa se elimina si el actor presiona el botón aceptar. 5. Se muestra un mensaje de confirmación.
<p>Excepciones</p> <ul style="list-style-type: none"> • Error al editar la Etapa <ol style="list-style-type: none"> 1. Después de ejecutar el paso 3 del Flujo alternativo. 2. El sistema encuentra datos mal ingresados 3. El sistema informa al actor que verifique la información ingresada. • Error al eliminar la Etapa <ol style="list-style-type: none"> 1. Después de ejecutar el paso 2 del Flujo alternativo. 2. El sistema informa al usuario que no puede eliminar la etapa seleccionada. 3. El sistema informa la razón de porque no se puede eliminar el proyecto.

Tabla 40: Crear una Etapa de Proyecto
Fuente: Propia.

4.3.1.4.5.6 Registrar un Medio de Comunicación

Las Especificaciones del caso de uso “Registrar un Medio de Comunicación” son descritas en la siguiente Tabla:

Caso de Uso: Registrar Medio de Comunicación	
Actores	Administrador, Periodista.
Descripción	Tiene como función ingresar un Medio de Comunicación al sistema para que estas puedan ser tomadas y formar parte de algún reporte o informe.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser Administrador o Periodista. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Medios de Comunicación y opción Listado de Medios.
Post condiciones	Caso de Uso: Registrar Medio de Comunicación.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Medios de Comunicación escoge la opción Lista de Medios. 2. El sistema muestra la ventana correspondiente <ul style="list-style-type: none"> • Nuevo 3. Ingresar los Datos Generales del Medio de Comunicación como Nombre, Tipo, de Medio, Sector al que pertenece, frecuencia de transmisión, política, publico objetivo, cobertura, teléfono, fax email, el nombre de la persona representante del medio y el contacto.. 4. El actor hace clic en el botón Siguiente. 5. El sistema muestra un mapa, y el actor busca y asigna la ubicación en el mapa. 6. El actor hace clic en el botón Guardar 7. El sistema guarda y muestra un mensaje indicando que la operación finaliza 	

correctamente.
Flujo Alternativo
<ul style="list-style-type: none"> • Editar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un Medio de Comunicación y hace clic en la opción editar. 2. El sistema carga los datos del Medio en una nueva ventana 3. El actor modifica los campos y presiona el botón Guardar • Eliminar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un Medio de Comunicación. 2. El actor presiona la opción Eliminar 3. El sistema pregunta al actor si realmente desea borrar el registro. 4. El Medio de Comunicación se elimina si el actor presiona el botón aceptar. 5. Se muestra un mensaje de confirmación.
Excepciones
<ul style="list-style-type: none"> • Error al editar al Medio de Comunicación <ol style="list-style-type: none"> 1. Después de ejecutar el paso 3 del Flujo alternativo. 2. El sistema encuentra datos mal ingresados 3. El sistema informa al actor que verifique la información ingresada. • Error al eliminar al Medio de Comunicación <ol style="list-style-type: none"> 1. Después de ejecutar el paso 2 del Flujo alternativo. 2. El sistema informa al usuario que no puede eliminar el Medio seleccionado. 3. El sistema informa la razón de porque no se puede eliminar al Medio de Comunicación.

Tabla 41: Registrar un Medio de Comunicación
Fuente: Propia.

4.3.1.4.5.7 Registrar un Programa

Las Especificaciones del caso de uso “Registrar un Programa” son descritas en la siguiente Tabla:

Caso de Uso: Registrar un Programa	
Actores	Administrador, Periodista
Descripción	Tiene como función ingresar un programa específico de algún Medio de Comunicación al sistema para que este

	pueda ser tomado en cuenta y formar parte de algún reporte o informe.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser Administrador o Periodista. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Medios de Comunicación y opción Listado de Programas.
Post condiciones	Caso de Uso: Registrar un Programa
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Medios de Comunicación escoge la opción Lista de Programas. 2. El sistema muestra la ventana correspondiente • Nuevo 3. Ingresar los Datos Generales del Medio de Comunicación como Nombre, y selecciona el Medio de Comunicación al que pertenece. 4. El actor hace clic en el botón Guardar 5. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente. 	
Flujo Alternativo	
<ul style="list-style-type: none"> • Editar 1. El actor selecciona un registro de un Programa ya ingresado y hace clic en la opción editar. 2. El sistema carga los datos del Programa en una nueva ventana. 3. El actor modifica los campos y presiona el botón Guardar • Eliminar 1. El actor selecciona un registro de un Programa. 2. El actor presiona la opción Eliminar 3. El sistema pregunta al actor si realmente desea borrar el registro. 4. El Programa se elimina si el actor presiona el botón aceptar. 5. Se muestra un mensaje de confirmación. 	

Excepciones
<ul style="list-style-type: none"> • Error al editar un Programa <ol style="list-style-type: none"> 1. Después de ejecutar el paso 3 del Flujo alternativo. 2. El sistema encuentra datos mal ingresados 3. El sistema informa al actor que verifique la información ingresada. • Error al eliminar un Programa <ol style="list-style-type: none"> 1. Después de ejecutar el paso 2 del Flujo alternativo. 2. El sistema informa al usuario que no puede eliminar el Programa seleccionado. 3. El sistema informa la razón de porque no se puede eliminar el Programa.

Tabla 42: Registrar un Programa.
Fuente: Propia.

4.3.1.4.5.8 Registrar un Periodista

Las Especificaciones del caso de uso “Registrar un Periodista” son descritas en la siguiente Tabla:

Caso de Uso: Registrar un Periodista	
Actores	Administrador, Periodista
Descripción	Tiene como función ingresar un Periodista de algún Medio de Comunicación al sistema para que este pueda ser tomado en cuenta y formar parte de un reporte o informe.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser Administrador o Periodista. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Medios de Comunicación y opción Listado de Periodistas.
Post condiciones	Caso de Uso: Registrar un Periodista.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Medios de Comunicación escoge la opción 	

<p>Lista de Periodistas.</p> <p>2. El sistema muestra la ventana correspondiente</p> <ul style="list-style-type: none"> • Nuevo <p>3. Ingresar los Datos Generales del Periodistas como son: Si está activo, su número de cedula, nombres, apellidos, cargar su foto, y el medio de comunicación al que pertenece, Cargo, Política, Instrucción, Titulo, Genero, Teléfono, email y una síntesis de sus principales funciones.</p> <p>4. El actor hace clic en el botón Guardar</p> <p>5. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.</p>
<p>Flujo Alternativo</p>
<ul style="list-style-type: none"> • Editar <p>1. El actor selecciona un registro de un Periodista ya ingresado y hace clic en la opción editar.</p> <p>2. El sistema carga los datos del Periodista en una nueva ventana.</p> <p>3. El actor modifica los campos y presiona el botón Guardar</p> <ul style="list-style-type: none"> • Eliminar <p>1. El actor selecciona un registro de un Periodista.</p> <p>2. El actor presiona la opción Eliminar</p> <p>3. El sistema pregunta al actor si realmente desea borrar el registro.</p> <p>4. El Periodista se elimina si el actor presiona el botón aceptar.</p> <p>5. Se muestra un mensaje de confirmación.</p>
<p>Excepciones</p>
<ul style="list-style-type: none"> • Error al editar un Periodista <p>1. Después de ejecutar el paso 3 del Flujo alternativo.</p> <p>2. El sistema encuentra datos mal ingresados</p> <p>3. El sistema informa al actor que verifique la información ingresada.</p> <ul style="list-style-type: none"> • Error al eliminar un Periodista <p>1. Después de ejecutar el paso 2 del Flujo alternativo.</p> <p>2. El sistema informa al usuario que no puede eliminar al Periodista seleccionado.</p> <p>3. El sistema informa la razón de porque no se puede eliminar al Periodista.</p>

Tabla 43: Registrar un Periodista.

Fuente: Propia.

4.3.1.4.5.9 Crear un Reporte de Prensa

Las Especificaciones del caso de uso “Crear un Reporte de Prensa” son descritas en la siguiente Tabla:

Caso de Uso: Crear un Reporte de Prensa	
Actores	Periodista, Administrador
Descripción	Tiene como función ingresar un nuevo Reporte de Prensa al sistema.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser periodista o administrador. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Reporte de Prensa y opción Lista de Reportes de Prensa.
Post condiciones	Caso de Uso: Crear un Reporte de Prensa
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Reporte de Prensa escoge la opción Lista de Reportes de Prensa. 2. El sistema muestra la ventana correspondiente <ul style="list-style-type: none"> • Nuevo <ol style="list-style-type: none"> 3. Seleccionar el Medio de Comunicación, ingresar la sección, página, titular, síntesis, la Fecha de Envío, la Fecha en que se generó la Noticia y el ámbito del Reporte. 4. El actor hace clic en el botón Siguiente. 5. El actor agrega las Instituciones que van a intervenir en este proyecto haciendo clic en la opción + o en la opción – para quitar el instituciones ya agregadas. 6. El actor hace clic en el botón Siguiente. 7. El actor agrega a los Responsables Institucionales que van a intervenir en este proyecto haciendo clic en la opción + o en la opción – para quitar el Responsable ya agregado. 8. El actor hace clic en el botón Siguiente. 9. El actor hace clic en el botón añadir archivo de evidencia cuantas veces desee agregarlos. 	

<ol style="list-style-type: none"> 10. El actor ingresa la descripción, adjunta un archivo (texto, audio, video) e ingresa la fecha de generación del archivo. 11. El actor hace clic en el botón Guardar 12. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.
<p>Flujo Alternativo</p> <ul style="list-style-type: none"> • Editar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un Reporte de Prensa y hace clic en la opción editar. 2. El sistema carga los datos del Reporte de Prensa seleccionado en una nueva ventana 3. El actor modifica los campos y presiona el botón Guardar • Eliminar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un Reporte de Prensa. 2. El actor presiona la opción Eliminar 3. El sistema pregunta al actor si realmente desea borrar el registro. 4. El Reporte de Prensa se elimina si el actor presiona el botón aceptar. 5. Se muestra un mensaje de confirmación.
<p>Excepciones</p> <ul style="list-style-type: none"> • Error al editar el Reporte de Prensa <ol style="list-style-type: none"> 1. Después de ejecutar el paso 3 del Flujo alternativo. 2. El sistema encuentra datos mal ingresados 3. El sistema informa al actor que verifique la información ingresada. • Error al eliminar el Reporte de Prensa <ol style="list-style-type: none"> 1. Después de ejecutar el paso 2 del Flujo alternativo. 2. El sistema informa al usuario que no puede eliminar el reporte de Prensa seleccionado. 3. El sistema informa la razón de porque no se puede eliminar el Reporte de Prensa.

Tabla 44: Crear un Reporte de Prensa.
Fuente: Propia.

4.3.1.4.5.10 Reporte de Apreciaciones Negativas de Prensa

Las Especificaciones del caso de uso “Reporte de Apreciaciones Negativas de Prensa” son descritas en la siguiente Tabla:

Caso de Uso: Reporte de Apreciaciones Negativas de Prensa	
Actores	Periodista, Administrador
Descripción	Muestra un cuadro estadístico y un listado de apreciaciones negativas de las cuales han sido sujetas las instituciones por parte de los principales medios de comunicación y periodistas.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser periodista o administrador. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Reporte de Prensa y opción Apreciaciones Negativas.
Post condiciones	Caso de Uso: Reporte de Apreciaciones Negativas de Prensa.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Reporte de Prensa escoge la opción Apreciaciones Negativas. 2. El sistema muestra la ventana correspondiente con un gráfico estadístico y un listado de todos los registros con apreciaciones negativas. <ul style="list-style-type: none"> • Exportar listado. <ol style="list-style-type: none"> 3. El actor hace clic en el botón Exportar. 4. El sistema genera y muestra una pantalla con el reporte preparado y listo para exportar. 	
Flujo Alternativo	
<ul style="list-style-type: none"> • Exportar Listado de una Institución Específica. <ol style="list-style-type: none"> 1. El actor hace clic en una barra del gráfico estadístico. 2. El sistema carga los datos correspondientes a la Institución seleccionada. 5. El actor hace clic en el botón Exportar. 6. El sistema genera y muestra una pantalla con el reporte preparado y listo para exportar. 	
Excepciones	
<ul style="list-style-type: none"> • N/A 	

Tabla 45: Reporte de Apreciaciones Negativas de Prensa.
Fuente Propia.

4.3.1.4.5.11 Reporte de Apreciaciones Positivas de Prensa.

Las Especificaciones del caso de uso “Reporte de Apreciaciones Positivas de Prensa” son descritas en la siguiente Tabla:

Caso de Uso: Reporte de Apreciaciones Positivas de Prensa	
Actores	Periodista, Administrador
Descripción	Muestra un cuadro estadístico y un listado de apreciaciones positivas de las cuales han sido sujetas las instituciones por parte de los principales medios de comunicación y periodistas.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser periodista o administrador. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Reporte de Prensa y opción Apreciaciones Positivas.
Post condiciones	Caso de Uso: Reporte de Apreciaciones Positivas de Prensa.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Reporte de Prensa escoge la opción Apreciaciones Positivas. 2. El sistema muestra la ventana correspondiente con un gráfico estadístico y un listado de todos los registros con apreciaciones positivas. • Exportar listado. 3. El actor hace clic en el botón Exportar. 4. El sistema genera y muestra una pantalla con el reporte preparado y listo para exportar. 	
Flujo Alternativo	
<ul style="list-style-type: none"> • Exportar Listado de una Institución Específica. 3. El actor hace clic en una barra del gráfico estadístico. 	

<ol style="list-style-type: none"> 4. El sistema carga los datos correspondientes a la Institución seleccionada. 5. El actor hace clic en el botón Exportar. 6. El sistema genera y muestra una pantalla con el reporte preparado y listo para exportar.
Excepciones
<ul style="list-style-type: none"> • N/A

Tabla 46: Reporte de Apreciaciones Positivas de Prensa.

Fuente: Propia.

4.3.1.4.5.12 Crear un reporte de Programa

Las Especificaciones del caso de uso “Crear un Reporte de Programa” son descritas en la siguiente Tabla:

Caso de Uso: Crear un Reporte de Programa	
Actores	Periodista, Administrador
Descripción	Tiene como función ingresar un nuevo Reporte de Programa al sistema.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser periodista o administrador. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Reporte de Programa y opción Lista de Reportes de Programa.
Post condiciones	Caso de Uso: Crear un Reporte de Programa.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Reporte de Programa escoge la opción Lista de Reportes de Programa. 2. El sistema muestra la ventana correspondiente <ul style="list-style-type: none"> • Nuevo <ol style="list-style-type: none"> 3. Seleccionar el Medio de Comunicación, selecciona el programa, titular, síntesis, la Fecha de Envío, la Fecha en que se generó la Noticia y el ámbito del Reporte. 	

4. El actor hace clic en el botón Siguiente.
5. El actor agrega las Instituciones que van a intervenir en este Reporte de Programa haciendo clic en la opción + o en la opción – para quitar el instituciones ya agregadas.
6. El actor hace clic en el botón Siguiente.
7. El actor agrega a los Responsables Institucionales que van a intervenir en este proyecto haciendo clic en la opción + o en la opción – para quitar el Responsable ya agregado.
8. El actor hace clic en el botón Siguiente.
9. El actor hace clic en el botón añadir archivo de evidencia cuantas veces desee agregarlos.
10. El actor ingresa la descripción, adjunta un archivo (texto, audio, video) e ingresa la fecha de generación del archivo.
11. El actor hace clic en el botón Guardar
12. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente.

Flujo Alternativo

- **Editar**

1. El actor selecciona un registro de un Reporte de Programa y hace clic en la opción editar.
2. El sistema carga los datos del Reporte de programa seleccionado en una nueva ventana
3. El actor modifica los campos y presiona el botón Guardar

- **Eliminar**

1. El actor selecciona un registro de un Reporte de programa.
2. El actor presiona la opción Eliminar
3. El sistema pregunta al actor si realmente desea borrar el registro.
4. El Reporte de Programa se elimina si el actor presiona el botón aceptar.
5. Se muestra un mensaje de confirmación.

Excepciones

- **Error al editar el Reporte de Programa.**
 1. Después de ejecutar el paso 3 del Flujo alternativo.
 2. El sistema encuentra datos mal ingresados
 3. El sistema informa al actor que verifique la información ingresada.
- **Error al eliminar el Reporte de Programa.**
 1. Después de ejecutar el paso 2 del Flujo alternativo.
 2. El sistema informa al usuario que no puede eliminar el Reporte de Programa seleccionado.
 3. El sistema informa la razón de porque no se puede eliminar el Reporte de Programa.

Tabla 47: Crear un Reporte de Programa.

Fuente: Propia.

4.3.1.4.5.13 Reporte de Apreciaciones Negativas de Programa

Las Especificaciones del caso de uso “Reporte de Apreciaciones Negativas de Programa” son descritas en la siguiente Tabla:

Caso de Uso: Reporte de Apreciaciones Negativas de Programa	
Actores	Periodista, Administrador
Descripción	Muestra un cuadro estadístico y un listado de apreciaciones negativas de las cuales han sido sujetas las instituciones por parte de los principales medios de comunicación y periodistas.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser periodista o administrador. • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Reporte de Programa y opción Apreciaciones Negativas.
Post condiciones	Caso de Uso: Reporte de Apreciaciones Negativas de Programa.
Flujo Normal de Eventos	

<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Reporte de Programa escoge la opción Apreciaciones Negativas. 2. El sistema muestra la ventana correspondiente con un gráfico estadístico y un listado de todos los registros con apreciaciones negativas. <ul style="list-style-type: none"> • Exportar listado. <ol style="list-style-type: none"> 3. El actor hace clic en el botón Exportar. 4. El sistema genera y muestra una pantalla con el reporte preparado y listo para exportar.
Flujo Alternativo
<ul style="list-style-type: none"> • Exportar Listado de una Institución Específica. <ol style="list-style-type: none"> 1. El actor hace clic en una barra del gráfico estadístico. 2. El sistema carga los datos correspondientes a la Institución seleccionada. 5. El actor hace clic en el botón Exportar. 6. El sistema genera y muestra una pantalla con el reporte preparado y listo para exportar.
Excepciones
<ul style="list-style-type: none"> • N/A

Tabla 48: Reporte de Apreciaciones Negativas de Programa.
Fuente: Propia.

4.3.1.4.5.14 Reporte de Apreciaciones Positivas de Programa.

Las Especificaciones del caso de uso “Reporte de Apreciaciones Positivas de Programa” son descritas en la siguiente Tabla:

Caso de Uso: Reporte de Apreciaciones Positivas de Programa	
Actores	Periodista, Administrador
Descripción	Muestra un cuadro estadístico y un listado de apreciaciones positivas de las cuales han sido sujetas las instituciones por parte de los principales medios de comunicación y periodistas.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser periodista

	<p>o administrador.</p> <ul style="list-style-type: none"> • El usuario autenticado con éxito en el Sistema. • Seleccionar el menú Reporte de Programa y opción Apreciaciones Positivas.
Post condiciones	Caso de Uso: Reporte de Apreciaciones Positivas de Programa.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema en Reporte de Programa escoge la opción Apreciaciones Positivas. 2. El sistema muestra la ventana correspondiente con un gráfico estadístico y un listado de todos los registros con apreciaciones positivas. <ul style="list-style-type: none"> • Exportar listado. <ol style="list-style-type: none"> 3. El actor hace clic en el botón Exportar. 4. El sistema genera y muestra una pantalla con el reporte preparado y listo para exportar. 	
Flujo Alternativo	
<ul style="list-style-type: none"> • Exportar Listado de una Institución Específica. <ol style="list-style-type: none"> 1. El actor hace clic en una barra del gráfico estadístico. 2. El sistema carga los datos correspondientes a la Institución seleccionada. 5. El actor hace clic en el botón Exportar. 6. El sistema genera y muestra una pantalla con el reporte preparado y listo para exportar. 	
Excepciones	
<ul style="list-style-type: none"> • N/A 	

Tabla 49: Reporte de Apreciaciones Positivas de Programa.
Fuente: Propia.

4.3.1.4.5.15 Registro de productos para Inventario.

Las Especificaciones del caso de uso “Registro de productos para Inventario” son descritas en la siguiente Tabla:

Caso de Uso: Registro de productos para Inventario	
Actores	Administrador
Descripción	Tiene como función asignar nuevos productos al sistema para que estos puedan ser tomados y formar parte del Inventario.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser Administrador. • El usuario autenticado con éxito en el Sistema. • Seleccionar la opción Lista de Productos.
Post condiciones	Caso de Uso: Registro de productos para Inventario.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema Inventarios, escoge la opción Lista de Productos. 2. El sistema muestra la ventana correspondiente <ul style="list-style-type: none"> • Nuevo <ol style="list-style-type: none"> 3. Ingresar la Descripción del producto, marca, modelo, cantidad, fecha de Compra, seleccionar el proveedor y la categoría a la que pertenece. 4. El actor hace clic en el botón Guardar 5. El sistema guarda y muestra un mensaje indicando que la operación finaliza correctamente. 	
Flujo Alternativo	
<ul style="list-style-type: none"> • Editar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un producto y hace clic en la opción editar 2. El sistema carga los datos del producto correspondiente en una nueva ventana 3. El actor modifica los campos y presiona el botón Guardar <ul style="list-style-type: none"> • Eliminar <ol style="list-style-type: none"> 1. El actor selecciona un registro de un producto. 2. El actor presiona la opción Eliminar 3. El sistema pregunta al actor si realmente desea borrar el registro. 4. El Producto se elimina si el actor presiona el botón aceptar. 5. Se muestra un mensaje de confirmación. 	

Excepciones
<ul style="list-style-type: none"> • Error al editar el Producto <ol style="list-style-type: none"> 1. Después de ejecutar el paso 3 del Flujo alternativo. 2. El sistema encuentra datos mal ingresados 3. El sistema informa al actor que verifique la información ingresada. • Error al eliminar al Responsable <ol style="list-style-type: none"> 1. Después de ejecutar el paso 2 del Flujo alternativo. 2. El sistema informa al usuario que no puede eliminar el producto seleccionado. 3. El sistema informa la razón de porque no se puede eliminar el producto seleccionado.

Tabla 50: Registro de productos para Inventario.
Fuente: Propia.

4.3.1.4.5.16 Reporte de Auditoría

Las Especificaciones del caso de uso “Reporte de Auditoría” son descritas en la siguiente Tabla:

Caso de Uso: Reporte de Auditoría	
Actores	Administrador, Auditor
Descripción	Tiene como función consultar y presentar un reporte con las modificaciones en la base de datos de los procesos más relevantes en el sistema.
Precondición	<ul style="list-style-type: none"> • Usuario con rol y permiso de Autenticación. • El rol del Usuario debe ser Administrador, Auditor. • El usuario autenticado con éxito en el Sistema. • Seleccionar en el Menú Auditoria la opción Lista de Registros.
Post condiciones	Caso de Uso: Reporte de Auditoria.
Flujo Normal de Eventos	
<ol style="list-style-type: none"> 1. El actor ingresa al menú del Sistema Auditoría, escoge la opción Lista de Registros. 2. El sistema muestra la ventana correspondiente <ul style="list-style-type: none"> • Filtros <ol style="list-style-type: none"> 3. Ingresar los datos según el tipo de filtrado que desee (fecha de modificación, Usuario, Tabla, columna, Tipo de Modificación, Valor anterior, o Nuevo Valor). 	

4. El actor hace clic en el botón Consultar.
5. El sistema muestra el listado de los datos según el filtro Realizado.
Flujo Alternativo
N /A
Excepciones
<ul style="list-style-type: none"> • Ningún Registro Mostrado. <ol style="list-style-type: none"> 1. Después de ejecutar el paso 3 del Flujo alternativo. 2. El sistema no encuentra datos a mostrar. 3. El sistema informa al actor que verifique la información ingresada.

Tabla 51: Reporte de Auditoria.
Fuente: Propia.

4.3.2 Vista Dinámica.

4.3.2.1 Introducción

La vista dinámica tiende a analizarse como pequeñas piezas del sistema, es como escenarios individuales de operaciones. Representa las interacciones de los objetos de un sistema. Puede representar como el sistema responderá a las acciones de los usuarios, como los datos son movidos del almacenamiento a la vista del usuario y como los objetos son creados y manipulados; todo lo cual los convierte en los diagramas más usados en proyectos, ya que son los que revelan más directamente características específicas requeridas en el código final.

4.3.2.2 Diagramas de Secuencia

Los diagramas de secuencia ilustran la interacción entre objetos y el orden secuencial en el que ocurren dichas interacciones, es decir cómo se comunican los objetos entre sí.

A continuación se detalla la secuencia de interacción del mecanismo implementación MVC.

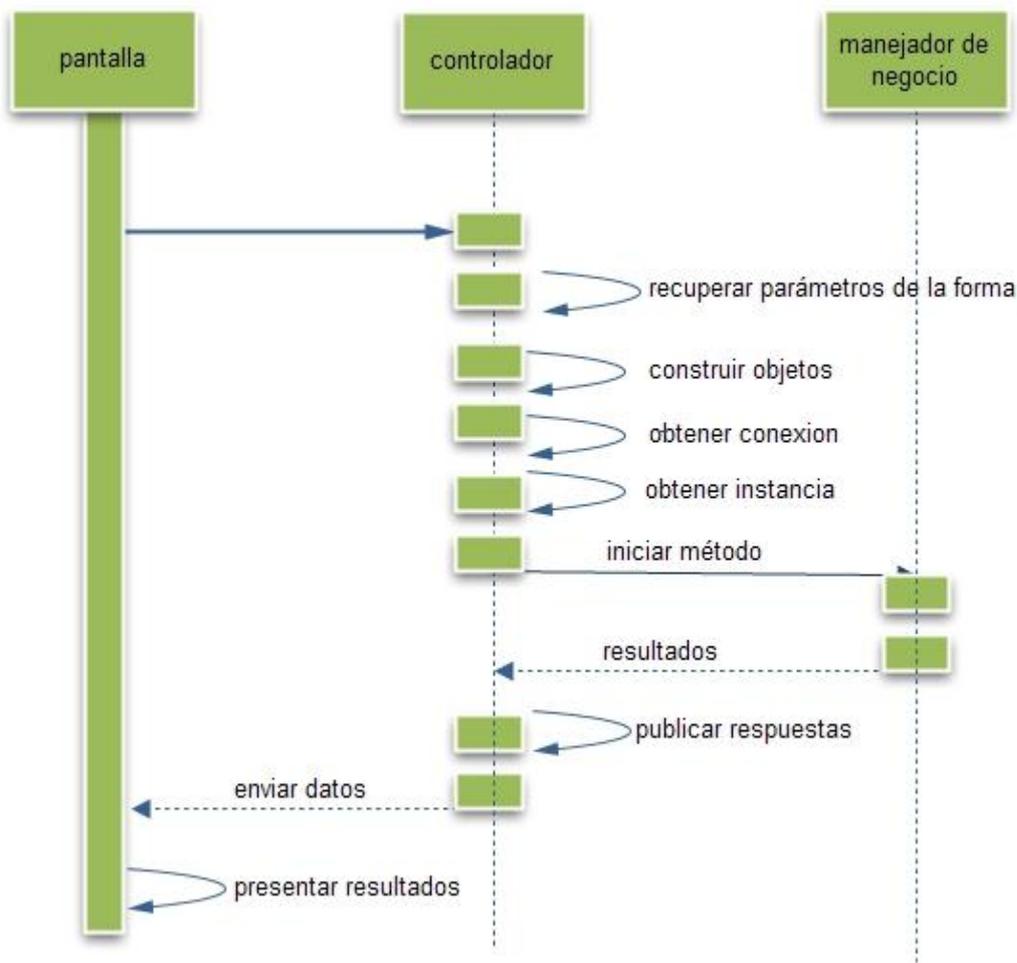


Ilustración 32: Diagramas de Secuencia.
Fuente: Propia

4.3.3 Vista Lógica

4.3.3.1 Introducción

En esta vista se presentan las unidades lógicas que componen la arquitectura del proyecto, ya que el sistema manejará un patrón de arquitectura de software llamado MVC (Modelo - Vista - Controlador). Es un patrón que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El primer refinamiento realizado consiste en la descomposición en subsistemas. Los subsistemas representan cortes verticales al diseño del sistema. Cada subsistema consiste en el agrupamiento de diferentes funcionalidades relacionadas entre sí y posee la capacidad de funcionar como un sistema en sí mismo.

4.3.3.2 Descomposición de Subsistemas

La explicación va relacionada con la información del patrón MVC.

- ✚ La capa del modelo define la lógica de negocio (la base de datos pertenece a esta capa), los datos se encuentran en la base de datos PostgreSQL los mismos que serán recuperados con el controlador.
- ✚ La vista es lo que utilizan los usuarios para interactuar con la aplicación (los gestores de plantillas pertenecen a esta capa). Estas serán creadas en php desarrollado por el framework symfony 1.4 y se presenta haciendo uso de la tecnología EXT JS que es una tecnología de clientes ricos para hacer más vistosa y atractiva a la aplicación, añadiendo los componentes que se requiera para hacer más agradable al usuario. Estas páginas se guardan en varios directorios repartidos por todo el proyecto.
- ✚ El controlador es un bloque de código que realiza llamadas al modelo para obtener los datos y se los pasa a la vista para que los muestre al usuario. La comunicación entre la base de datos y el sistema la realiza el framework de desarrollo symfony. En Symfony, la capa del controlador, que contiene el código que liga la lógica de negocio con la presentación, está dividida en varios componentes que se utilizan para diversos propósitos:
 - El controlador frontal es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse.
 - Las acciones contienen la lógica de la aplicación. Verifican la integridad de las peticiones y preparan los datos requeridos por la capa de presentación.
 - Los objetos request, response y session dan acceso a los parámetros de la petición, las cabeceras de las respuestas y a los datos persistentes del usuario. Se utilizan muy a menudo en la capa del controlador.

4.3.4 Vista de Despliegue

El diagrama de despliegue muestra de manera gráfica los nodos que conforman el sistema, además se los describe indicando la localización de las tareas en los nodos físicos. Cada nodo representa un recurso de ejecución como una computadora o servidor y se encuentra conectado con otros mediante un enlace de comunicación.

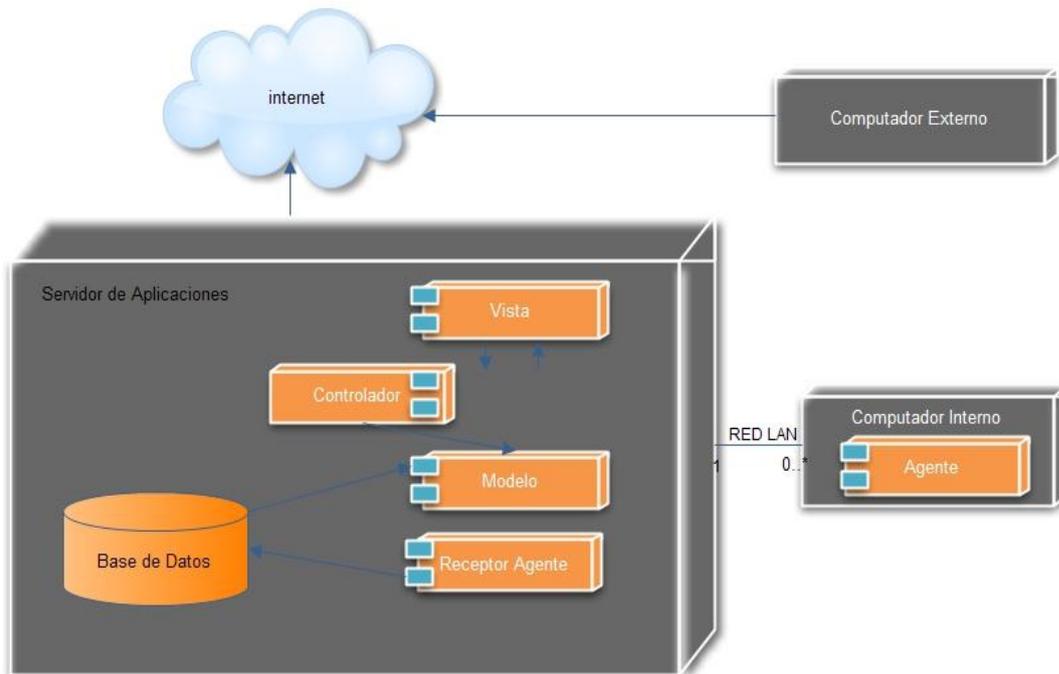


Ilustración 33: Vista de Despliegue.
Fuente: Propia

4.3.4.1 Servidor de Aplicaciones

En principio será una maquina ubicada en la gobernación de Imbabura, hasta que este sea remplazado por un servidor de características óptimas para el correcto desempeño de esta y muchas más aplicaciones que se tiene planeado implementar. Los usuarios tendrán acceso al servidor vía LAN e internet.

4.3.4.2 Computadores Internos

Son las computadoras que son utilizadas por cada uno de los usuarios que accedan al sistema principalmente para los módulos de reportes de medio de comunicación, recursos e infraestructura y el usuario administrador que será un miembro activo de la Gobernación Imbabura, como están conectadas directamente al servidor principal vía LAN.

4.3.4.3 Computadores Externos

Son los computadores que accederán a la aplicación mediante una conexión a internet.

4.3.5 Diseño en Capas

En la siguiente figura podemos observar la distribución de los paquetes en cada una de las capas del sistema.

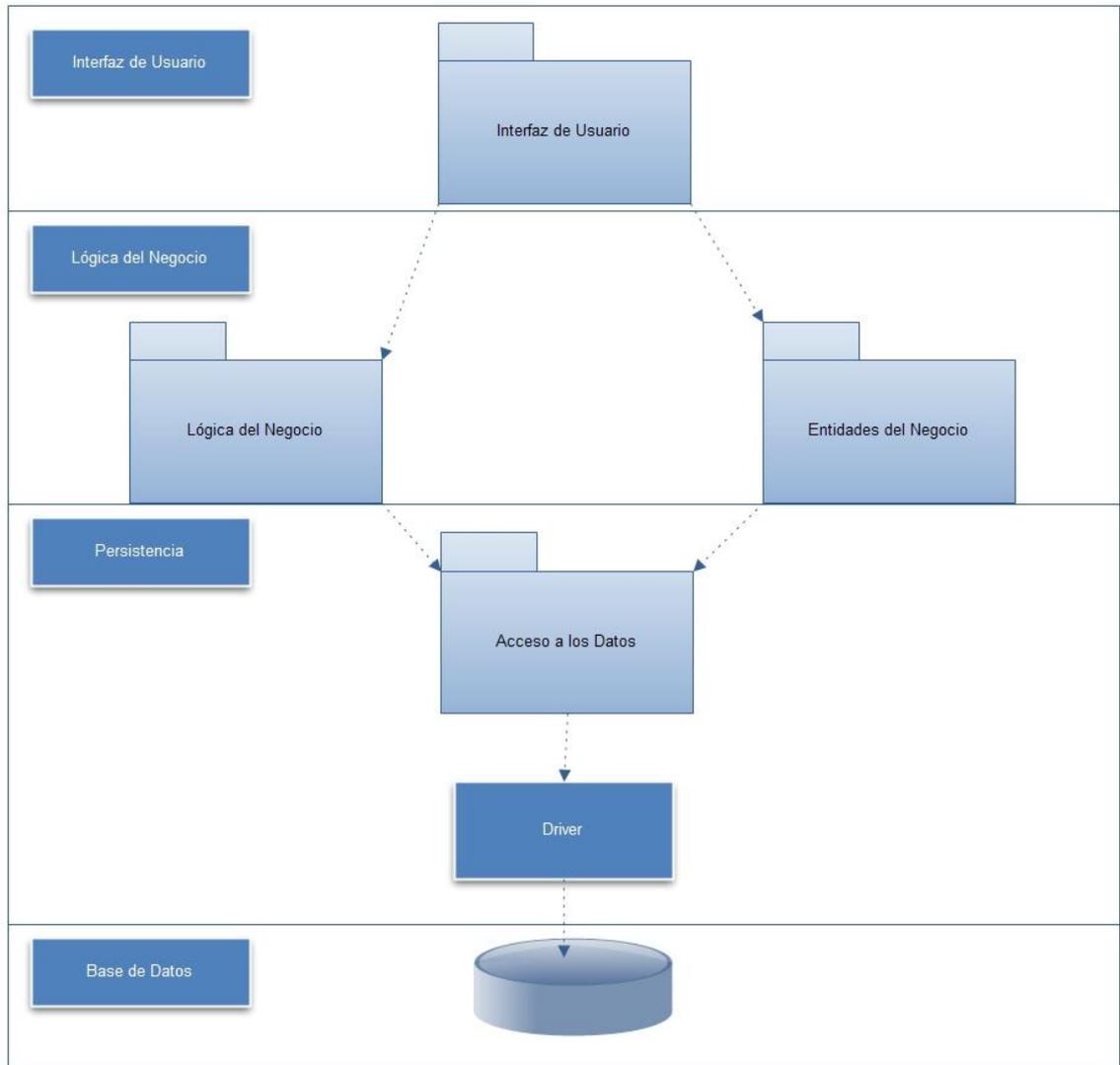


Ilustración 34: Diseño en Capas.
Fuente: Propia.

4.3.5.1 Capa Interfaz de Usuario

Esta capa contiene el paquete Interfaz de Usuario, el cual almacena todas las clases y vistas con las el usuario puede interactuar con la aplicación.

4.3.5.2 Capa Lógica del Negocio

Esta capa contiene los paquetes de Servicios de Negocio y Entidades de Negocio, aquí se maneja la lógica para el manejo de las principales operaciones establecidas.

4.3.6 Vista de Datos

Se realizará la estructura de los datos desarrollando los siguientes modelos de datos:

4.3.6.1 Modelo Relacional

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que estos se almacenen no tiene relevancia. Este tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto.

De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar registro y a cada columna también se le puede llamar campo o atributo.

4.3.6.2 Modelo Físico

Modelo físico de los datos es una representación de un diseño de los datos que considere las instalaciones y los apremios de un sistema de gerencia dado la base de datos. En el ciclo de vida de un proyecto se deriva típicamente de un modelo lógico de los datos, aunque puede ser reverso dirigido de una puesta en práctica dada la base de datos. Un modelo físico completo de los datos incluirá todos los artefactos de la base de datos requeridos para crear relaciones entre las tablas o para alcanzar metas del funcionamiento.

4.3.7 Tamaño y Desempeño

4.3.7.1 Tiempo de respuesta en el acceso a la base de datos

El sistema proveerá accesos a la base de datos con un tiempo de respuesta no mayor a los 4 segundos.

4.3.7.2 Tiempo de respuesta de transacciones

El programa no demorará más de 5 segundos en generar una distribución óptima para los cortes haciendo uso del mejor algoritmo elegido y guardándolo en base de datos mediante el uso de Doctrine que es un mapeador de objetos-relacional (ORM).

4.3.7.3 Espacio en disco para Servidor de Base de Datos

El espacio en disco necesario para la parte del Servidor de Base de datos deberá tener como mínimo 3 GB libres en disco para su correcto funcionamiento. La arquitectura seleccionada apoya los requisitos de latencia y capacidad en disco en puesta en práctica de una arquitectura cliente servidor y web. La porción del cliente solo se pone en ejecución en las PC locales de los distintos ambientes dentro de la gobernación. Los componentes se han diseñado para asegurarse de obtener requisitos mínimos de disco y memoria en el lado de las PC del cliente.

4.3.8 Calidad

Para un mejor aprovechamiento de la arquitectura de software se dan los siguientes requerimientos:

4.3.8.1 Usabilidad

El sistema permitirá un manejo intuitivo por parte de los usuarios.

4.3.8.2 Eficiencia

El programa no demorará más demasiado, en generar una distribución óptima para los cortes haciendo uso del algoritmo elegido.

4.3.8.3 Seguridad

El sistema permitirá el acceso a funcionalidades dependiendo del perfil del usuario que ingresa al sistema, validando su ingreso a través de una clave.

4.3.8.4 Confiabilidad

El sistema tendrá en cuenta que la información ingresada sea válida, para lo cual mostrará mensajes que expliquen al usuario acerca de los errores que éste pudiera cometer y de aquellos que pueda cometer el mismo sistema.

4.3.8.5 Mantenimiento

El sistema será flexible, facilitando su mantenimiento a futuro, además permita continuar con el crecimiento e inserción de nuevas opciones y procedimientos.

4.3.9 Arquitectura del Proyecto

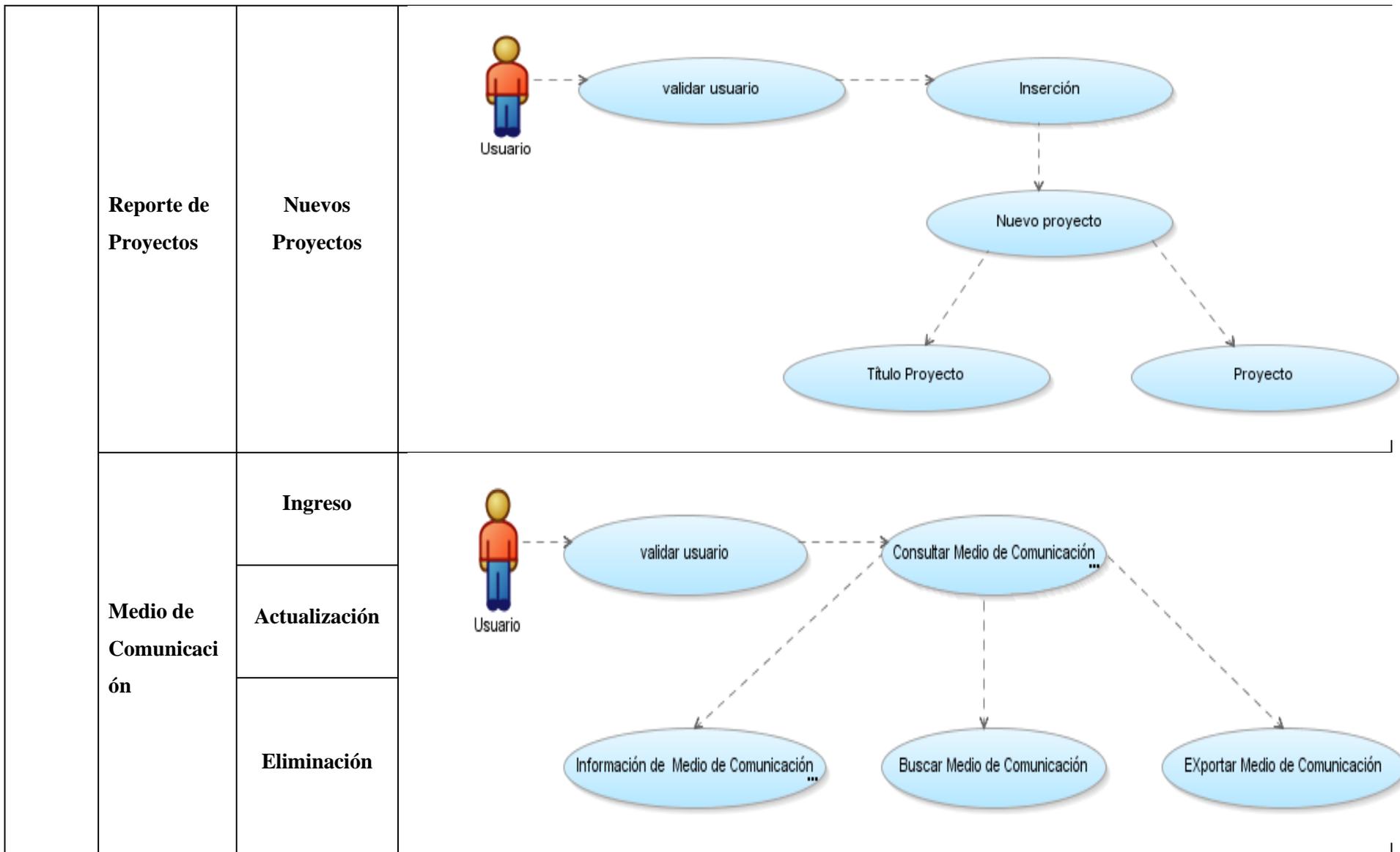
4.3.9.1 Introducción

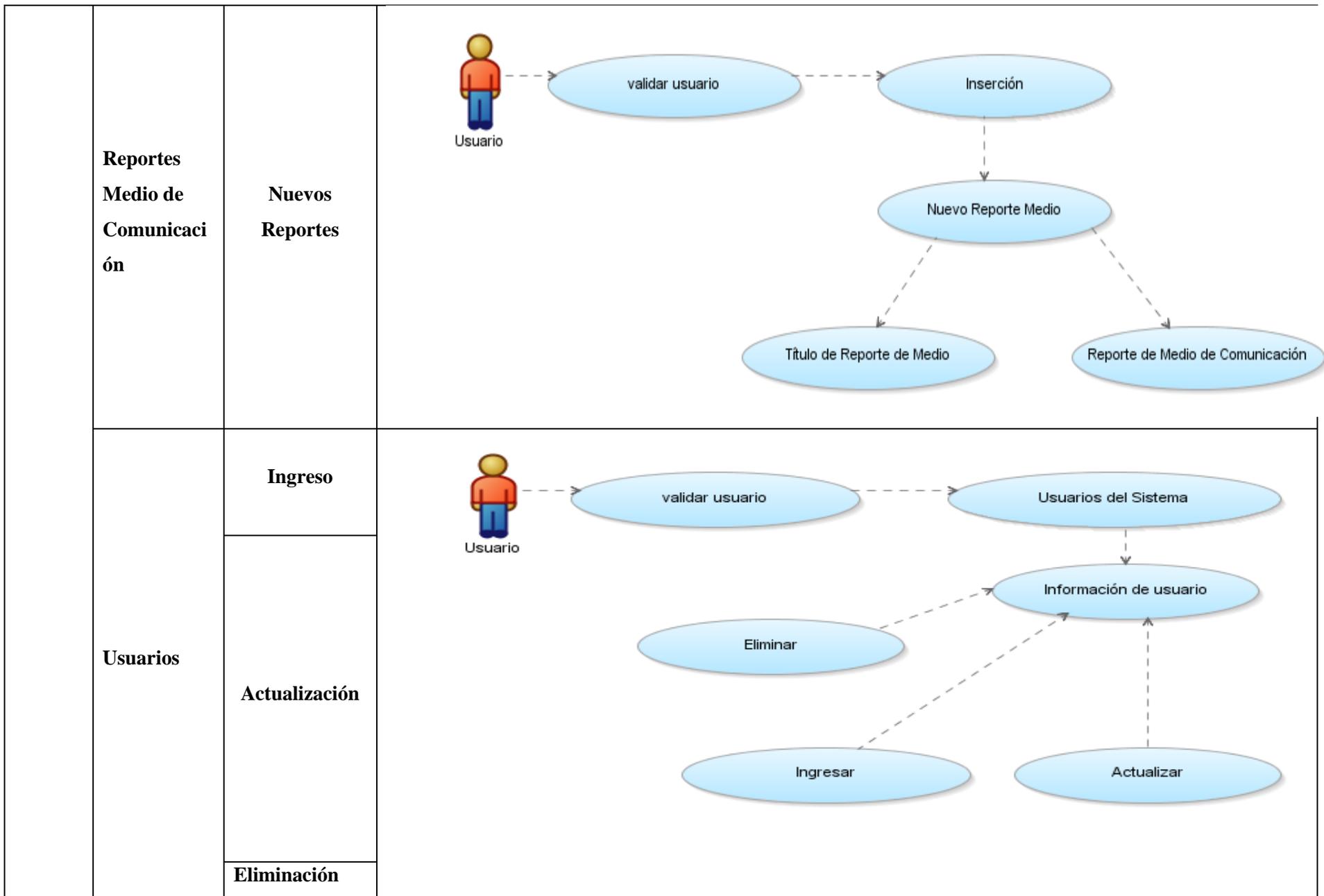
La base arquitectónica presentada, describe a la aplicación del sistema de acopio de información y reporte de medios de comunicación. El desarrollo de un sistema con gran cantidad de software requiere que este sea visto desde diferentes perspectivas, diferentes usuarios tanto del equipo de desarrollo como del usuario final (experto de dominio). También tenemos que considerar que una arquitectura no debe centrarse específicamente en la estructura y en el comportamiento, sino que abarque temas como el uso, funcionalidad, rendimiento, capacidad de adaptación, reutilización, capacidad para ser comprendida, restricciones, compromisos entre alternativas, así como aspectos estéticos y de presentación.

4.3.9.2 Paquetes de Análisis

Los paquetes de análisis permiten organizar del modelo de análisis en grupos manejables. Se identifican los siguientes paquetes de análisis:

Modulo	Paquetes de Análisis	Casos de uso	Gráfico Casos de Uso
Sistema Web	Instituciones	Ingreso	<pre> graph LR Usuario[Usuario] -.-> validar_usuario[validar usuario] validar_usuario -.-> consultar_instituciones[Consultar Instituciones] consultar_instituciones -.-> informacion_institucion[información de institución] consultar_instituciones -.-> buscar_institucion[buscar Institución] consultar_instituciones -.-> exportar_institucion[exportar Institución] </pre>
		Actualización	
		Eliminación	
	Responsables	Ingreso	<pre> graph LR Usuario[Usuario] -.-> validar_usuario[validar usuario] validar_usuario -.-> consultar_responsable[Consultar Responsable] consultar_responsable -.-> informacion_responsable[información de Responsable] consultar_responsable -.-> buscar_responsable[buscar Responsable] consultar_responsable -.-> exportar_responsable[exportar Responsable] </pre>
		Actualización	
		Eliminación	





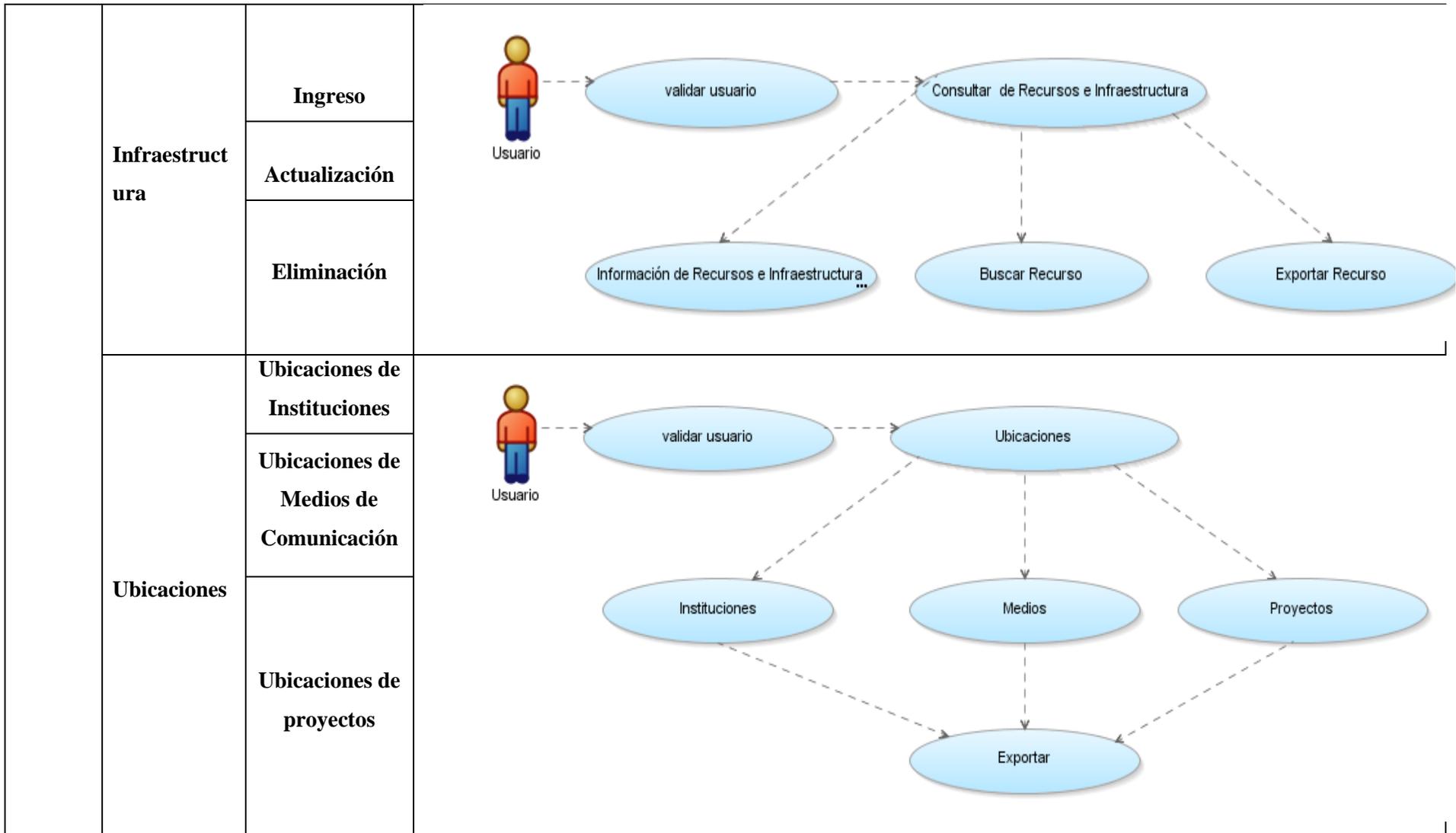


Ilustración 35: Paquetes de Análisis.
Fuente: Propia.

4.3.10 Lista de Riesgos

4.3.10.1 Introducción

La lista de riesgos permite analizar cada uno de los posibles problemas que se tendrá a lo largo del desarrollo del proyecto, magnitud, descripción, impactos que pueden causar, el indicador que nos permitirá verificar si el riesgo está por producirse o convertirse en un problema, una estrategia de mitigación para evitar que estos riesgos se cumplan, y el plan de contingencia que se dará si el riesgo se llegara a cumplir.

4.3.10.1.1 Propósito

El propósito de este documento es describir el proceso de administración de riesgos presentes a lo largo de la elaboración del proyecto, esta lista nos permitirá revisar periódicamente para evitar que uno de ellos se convierta en un problema. Este proceso de administración comprende la identificación, monitoreo y posterior mitigación de los riesgos.

4.3.10.1.2 Alcance

La lista de riesgos tiene como alcance definir varios tipos de riesgos en nuestro caso analizamos diferentes tipos:

- ✚ Riesgos de recursos (Organización y persona).
- ✚ Riesgos tecnológicos.
- ✚ Riesgos de dependencia externa.

4.3.10.1.3 Revisión General

En el plan de riesgos que a elaborar, se analizara su magnitud que será la que permita clasificar que tan perjudicial puede ser para el proyecto el impacto que puede tener el proyecto, como podemos monitorear determinando riesgos, que está haciendo para reducir impactos de riesgos y en caso de que afecten cual será el plan de contingencia para que el riesgo no se convierta en problema y afecte al proyecto.

4.3.10.2 Riesgos

4.3.10.2.1 El cliente EXT JS y framework symfony1.4 son herramientas poco conocidas por el desarrollador

Magnitud de riesgo o ranking.

Alto.

Descripción.

Este riesgo se puede presentar por la falta de conocimientos acerca de la utilización de estas herramientas ya que en las aulas no han sido estudiadas aun nivel requerido para el desarrollo de una aplicación de gran magnitud y requerimientos específicos.

Impactos.

- ✚ Demora al momento de la programación en nivel de presentación de formularios.
- ✚ Demora al momento de realizar la conexión de la interfaz gráfica con el modelo de datos.
- ✚ Falta de soporte para el tipo de forma o vista que se requiera.

Indicadores

% información de Interfaz de Usuario= información sobre las maneras de presentación de formas y su conexión a php mediante Symfony 1.4/ información en cada tabla e integración con el lado del servidor * 100

Umbral: 100%

Estrategia de mitigación.

Investigar y obtener conocimientos adecuados relacionados al software que se va a utilizar con el fin de asegurar una correcta utilización del mismo.

Plan de contingencia.

Pertenecer e inscribirse en foros con la comunidad de desarrollo que manejen estas tecnologías para el momento que se necesite solicitar ayuda o hacerles conocer las inquietudes sobre un determinado problema ya que existe gran cantidad de información sobre las herramientas utilizadas.

4.3.10.2.2 La Gobernación Imbabura no cuenta con Servidores y equipos tecnológicos donde instalar la Aplicación.

Magnitud de riesgo o ranking.

Medio.

Descripción.

Este riesgo se presenta por la falta de recursos e infraestructura los que son requeridos para el correcto funcionamiento y sobretodo el control total de la aplicación.

Impactos.

- ✚ Demora al momento de realizar consultas debido a que no es igual la velocidad de una aplicación con servidor propio a una alojada en un hosting.
- ✚ Falta de Privacidad, Autonomía, control total y acceso a los datos.

Indicadores

% servidor externo= / seguridad y autonomía de la información * 100

Umbral: 70%

Estrategia de mitigación.

Investigar y obtener conocimientos adecuados relacionados al software que se va a utilizar con el fin de asegurar una correcta utilización del mismo.

Plan de contingencia.

Pertenecer e inscribirse en foros con la comunidad de desarrollo que manejen estas tecnologías para el momento que se necesite solicitar ayuda o hacerles conocer las inquietudes sobre un determinado problema ya que existe gran cantidad de información sobre las herramientas utilizadas.

4.3.10.2.3 Computador de acceso es atacado por virus

Magnitud de riesgo o ranking.

Medio.

Descripción.

Este riesgo se puede presentar por el simple hecho de conexión internet, ataque a los discos duros memoria del computador, etc. ya que al pasar conectado a internet pueden estar expuestos al contagio de algún tipo de virus.

Impactos.

- ✚ Publicidad exagerada en navegadores.
- ✚ Falta de control en el sistema.
- ✚ Reinicio imprevisto de los equipos.
- ✚ Incorrecto funcionamiento de navegadores.
- ✚ Fallos en el sistema de archivos y discos duros.

Indicadores

$\% \text{ archivos de sistema infectados} = \frac{\text{información necesaria para el correcto funcionamiento}}{\text{archivos infectados}} * 100$

Umbral: 50%

Estrategia de mitigación.

Realizar la instalación de un buen antivirus y que sea fácil de actualizar periódicamente para contrarrestar el brote de nuevas amenazas, realizar la desinfección del ordenador cada cierto periodo. Desinfectar flash memory o pen drives cada vez que se introduzca en el equipo.

Plan de contingencia.

Utilizar sistemas operativos Gnu/Linux ya que su exposición a los virus es significativamente menor que un sistema operativo Windows.

4.3.10.2.4 Modelo Vista controlador utilizado por Symfony no se acople a la metodología RUP.

Magnitud de riesgo o ranking.

Bajo.

Descripción.

Investigar a fondo el modelo MVC de Symfony y en caso de ser necesario acoplar los objetos de una manera adecuada, para construir la aplicación usando la metodología RUP.

Impactos.

- ✚ Diagramas de clase complejos.
- ✚ Cambio de arquitectura.

Indicadores

% Acoplamiento de metodología RUP y patrón MVC.

Umbral: 100%

Estrategia de mitigación.

Utilizar un diseñador de diagramas libre que facilite la construcción de la aplicación y que no sea tan complicado su manera de usarlo.

Plan de contingencia.

Realizar los diagramas según el patrón MVC.

4.3.10.2.5 Incompatibilidad con navegadores de internet y configuración en los clientes de acceso a la aplicación.

Magnitud de riesgo o ranking.

Bajo.

Descripción.

Debido a los diversos navegadores de internet no exista una compatibilidad, e igual funcionamiento, presentación de resultados y tiempo de respuestas debido a sus diversas programaciones.

Impactos.

-  Mostrar información de manera incorrecta.
-  Mostrar páginas web incompletas.
-  Reaccionar de una mala manera en las terminales clientes.

Indicadores

% Compatibilidad= Navegadores existentes/ Navegadores compatibles con el sistema * 100.

Umbral: 70%

Estrategia de mitigación.

Mediante casos de prueba, buscar componentes que puedan ser compatibles con todos los navegadores y activar todos los complementos necesarios.

Plan de contingencia.

Estandarizar el uso y establecer un navegador que mejor se adapte al funcionamiento de la aplicación, e realizar pruebas a los navegadores que manejen o gestionen la información multimedia de una manera adecuada.

4.3.10.2.6 Desconocer los procesos manejados por la Gobernación de Imbabura.**Magnitud de riesgo o ranking.**

Bajo.

Descripción.

Debido a la falta de conocimientos sobre los procesos realizados por la Gobernación de Imbabura, se podría estar confundido en la manera de ir estructurando la aplicación.

Impactos.

- ✚ No establecer los requerimientos verdaderos y más necesarios para la Institución.
- ✚ Realizar

Indicadores

% Conocimiento de los procesos que realiza la Gobernación de Imbabura.

Umbral: 30%

Estrategia de mitigación.

Gracias a los seis meses de pasantías realizados en la institución se pudo, conocer a nivel general del funcionamiento de los procesos más relevantes y necesarios para el desarrollo de la aplicación.

Plan de contingencia.

Antes de realizar el sistema realizar pruebas con el equipo político de la Gobernación y sus principales autoridades.

4.3.10.2.7 Resumen de Riesgos

Ranking	Descripción del Riesgo e Impacto	Estrategia de reducción del riesgo
7	La liberación del sistema AIRCOM podría no estar lista la finalización del programa mi primer empleo de duración de 6 meses.	Aumentar el esfuerzo en el desarrollo del sistema.
7	Que la Gobernación de Imbabura no haga la adquisición del servidor a tiempo.	Recordar al equipo político y Gobernador de la adquisición del servidor.
5	Que las características del servidor donde será alojado el sistema, no cumpla con los requerimientos necesarios para un correcto funcionamiento del sistema.	Antes de su uso especificar los debidos requerimientos de hardware y software.
3	Abandono del proyecto por parte del desarrollador	Aumentar el compromiso de terminar todo el sistema.
3	Incompatibilidad del navegador de internet de los usuarios.	Instalar el navegador compatible con el sistema.
3	Modificación de los requerimientos puestos en marcha.	Realizar actas de reunión de trabajo, definir bien los requerimientos y que el interesado apruebe dicha acta con la respectiva firma.
3	Saturación de la red informática por parte de los usuarios en actividades ajenas al sistema.	Sugerir políticas de uso de internet.
2	Presencia de virus en las computadores de los usuarios finales.	Instalar un antivirus y actualizar constantemente la base de datos del anti virus, además proponer el uso de sistemas Operativos GNU/LINUX

Tabla 52: Resume de Riesgos.
Fuente: Propia (Metodología RUP)

4.3.11 Estrategia de Pruebas

4.3.11.1 Pruebas de Integridad de datos

Objetivo:	Asegurar la integridad de datos
Técnica:	<ol style="list-style-type: none">1. Registrar datos con tipos válidos.2. Registrar datos en entidad que tengan relación con otras.3. Revisar el esquema de base de datos para asegurarse que los datos se han guardado satisfactoriamente y de acuerdo a los estándares definidos.
Criterio de completitud	Todos los métodos de acceso y procesos de la Base de datos funcionan como fueron diseñados.
Consideraciones especiales	<ol style="list-style-type: none">1. Se debe utilizar un conjunto pequeño de datos para incrementar la visibilidad de cualquier evento anormal o inesperado.2. Los datos de pruebas deberían ser reales y de uso común.

Tabla 53: Pruebas de Integridad de Datos.

Fuente: Metodología RUP

4.3.11.2 Pruebas del Sistema

Objetivo:	Asegurar la apropiada navegación dentro del sistema, ingreso de datos, procesamiento y recuperación.
Técnica:	Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que: <ol style="list-style-type: none">1. Los resultados esperados ocurren cuando se utiliza un dato válido.2. Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido.3. Cada regla de negocios es aplicada adecuadamente.
Criterio de completitud	<ol style="list-style-type: none">1. Todas las pruebas planeadas han sido ejecutadas.2. Todos los defectos que se identificaron han sido tenidos en cuenta.
Consideraciones especiales	Considerar aspectos que impactan la implementación y ejecución de las pruebas del Sistema

Tabla 54: Pruebas del Sistema.

Fuente: Metodología RUP

4.3.11.3 Pruebas del ciclo de Negocio.

Objetivo:	Asegurar que el sistema funciona de acuerdo con el modelo de negocios emulando todos los eventos en el tiempo y en función del tiempo.
Descripción de la prueba:	Las pruebas del ciclo de negocio deberían emular las actividades ejecutadas en él a través del tiempo. Debería identificarse un periodo, como por ejemplo un año, y las transacciones y actividades que podrían ocurrir durante un periodo de un año deberían ejecutarse. Incluyendo todos los ciclos y eventos diarios, semanales y mensuales.
Técnicas:	<p>Ejecute cada caso de uso, flujo básico o función utilizando datos válidos e inválidos, para verificar que:</p> <ol style="list-style-type: none"> 1. Incremente el número de veces en que una función es ejecutada para simular diferentes usuarios sobre un periodo especificado 2. Todas las fechas o funciones que involucren tiempos serán probadas con datos válidos e inválidos de fechas o periodos de tiempo. 3. Todas las funciones ocurren en un periodo de tiempo serán ejecutadas en el tiempo apropiado. 4. Los resultados esperados ocurren cuando los datos válidos son usados. 5. Los mensajes de error o de advertencia aparecen en el momento adecuado, cuando se utiliza un dato inválido. 6. Cada regla de negocios es aplicada adecuadamente.
Criterio de completitud	<ol style="list-style-type: none"> 7. Todas las pruebas planeadas han sido ejecutadas. 8. Todos los defectos que se identificaron han sido tenidos en cuenta.
Consideraciones especiales	<ol style="list-style-type: none"> 1. Las fechas y eventos del sistema pueden requerir actividades especiales de soporte. 2. Se requiere un modelo de negocios para identificar requisitos y procedimientos de prueba apropiados.

Tabla 55: Pruebas del Ciclo de Negocio.
Fuente: Metodología RUP

4.3.11.4 Pruebas de Interfaz de Usuario

Objetivo:	<p>Verificar lo siguiente:</p> <p>La navegación a través de los objetos de la prueba refleja la funcionalidad del proyecto; Se realiza una navegación de todos los menús y los formularios de cada submenú.</p> <p>Los elementos de los formularios tales como botones, íconos, alertas, etc. Deben mantener un solo formato.</p>
Descripción de la prueba:	<p>Las pruebas de interfaz de usuario verifican la adecuada interacción del usuario con el software. El objetivo es verificar que cada interfaz corresponda con la acción que realiza y que la interfaz tenga una adecuada navegación.</p>
Técnicas:	<p>Con la ayuda de los usuarios que usarán el sistema, se les pide que usen el sistema realizando las actividades y procesos cotidianos en el sistema.</p> <p>Los usuarios del sistema son reales y trabajan en su área de trabajo normal.</p> <p>Los desarrolladores no están presentes.</p> <p>Los usuarios son advertidos que el sistema puede fallar.</p>
Criterio de completitud	<p>Se establece un periodo de pruebas, en el que los errores detectados no sean clasificados como críticos para el sistema.</p>
Consideraciones especiales	<p>Se debe establecer el mecanismo de comunicación entre los usuarios y los desarrolladores para que los errores que se detecten puedan ser solucionados.</p>

Tabla 56: Pruebas de Interfaz de Usuario

Fuente: propia (Metodología RUP)

4.3.11.5 Pruebas de Desempeño.

Objetivo:	<p>Validar y verificar los requisitos de desempeño especificados para el sistema.</p> <p>Validar el tiempo de respuesta para las transacciones o procesos bajo las siguientes condiciones:</p> <p>Volumen normal anticipado.</p> <p>Volumen máximo anticipado.</p>
------------------	--

Descripción de la prueba:	<p>Las pruebas de desempeño miden los tiempos de respuesta que tiene el sistema y otros aspectos sensibles al tiempo.</p> <p>Las pruebas se realizan varias veces, cambiando entre una y otra cargas diferentes. La prueba inicial debe ejecutarse con una carga similar a la esperada, mientras que la prueba final se ejecuta utilizando una carga máxima esperada.</p> <p>Las pruebas se pueden utilizar para calibrar el desempeño del sistema en función de condiciones como el hardware, usuarios, etc.</p> <p>Algunas características que pueden afectar el desempeño son:</p> <ol style="list-style-type: none"> 1. Cuellos de botella en el disco. 2. Cuellos de botella en el CPU. 3. Capacidad de almacenamiento. 4. Capacidades físicas del hardware donde está alojado el sistema. 5. Virus informáticos. 6. Congestión en la red de datos.
Técnicas:	<ol style="list-style-type: none"> 1. Con la ayuda de los usuarios que usarán el sistema, se les pide que usen el sistema realizando las actividades y procesos cotidianos en el sistema. 2. Los usuarios del sistema son reales y trabajan en su área de trabajo normal. 3. Se pide a los usuarios que todos accedan a una determinada acción o proceso al mismo tiempo. 4. Los desarrolladores no están presentes. 5. Los usuarios son advertidos que el sistema puede fallar.
Criterio de completitud	Se establece un periodo de pruebas, en el que los errores detectados no sean clasificados como críticos.
Consideraciones especiales	Se debe establecer el mecanismo de comunicación entre los usuarios y los desarrolladores para que los errores que se detecten puedan ser solucionados.

Tabla 57: Pruebas de Desempeño.
Fuente: Propia (Metodología RUP)

4.3.11.6 Pruebas de Seguridad y Control de Acceso.

Objetivo:	<p>Seguridad en el Funcionamiento y Datos, verificar que los usuarios puedan acceder solo aquellas funciones y datos para los cuales se le ha otorgado permisos al momento de crear su perfil de usuario.</p> <p>Seguridad en Administración del Sistema, comprobar que solo aquellos usuarios con permisos privilegiados puedan acceder a las funciones parametrizables del sistema y opciones del sistema.</p>
------------------	--

Descripción de la prueba:	Las pruebas se realizará enfocándonos en dos aspectos principales: 1. Seguridad en la aplicación controlando el acceso a determinada información y funciones del negocio, y 2. Seguridad del sistema realizando registro de accesos de usuarios al sistema.
Técnicas:	1. Seguridad de Datos y Funciones, identificar los tipos de usuarios y asignar funciones a las que tiene acceso el tipo de usuarios para asignarle al usuario. 2. Realizar pruebas para cada tipo de usuario y verificar los permisos creado transacciones para cada tipo de usuario. 3. Re direccionamiento a la página de autenticación si el usuario aún no se ha logueado o registrado para el acceso al sistema. 4. Modificar los tipos de usuario y verificar si los permisos han cambiado para el usuario.
Criterio de completitud	Para cada tipo de dato se puede asignar las funciones y datos apropiados para su desempeño.
Consideraciones especiales	El acceso al sistema debe ser revisado con el administrador de la red y de la base de datos. Con esto podremos visualizar cualquier anomalía.

Tabla 58: Pruebas de Seguridad y Control de Acceso

Fuente: Propia (Metodología RUP).

4.3.11.7 Herramientas

Tarea	Herramientas
Registro de Defectos	Microsoft Word
Gestión de Proyecto	JDEVELOPER, Edraw Max, Microsoft Word
Herramientas DBMS	PGADMIN III

Tabla 59: Herramientas.

Fuente: Propia (Metodología RUP).

4.3.11.8 Recursos

Rol	Recurso Requerido	Responsabilidad Específica
Administrador de Pruebas	Marlon Arichávala	Proveer las directrices de las pruebas. Adquirir los recursos Necesarios.
Diseñador de Pruebas	Marlon Arichávala	Identificar y priorizar las pruebas. Generar Plan de Pruebas
Gestión de Proyecto	Marlon Arichávala	Responsables de ejecutar las pruebas, registro de resultados.
Administrador de BDD	Marlon Arichávala	Administrar y asegurar los datos de pruebas.

Tabla 60: Recursos.

Fuente: Propia (Metodología RUP).

4.3.12 Prototipo de Interfaz de Usuario

El siguiente artefacto detalla una parte importante del proyecto, que es la que interfaz gráfica con la que el usuario interactúa de forma directa con el sistema, esta se debe presentar con elementos adecuados, de vista atractiva y además que sea de fácil manejo. A continuación se presentan los modelos definidos en RUP como prototipos de interfaces gráficas de usuario diseñados para la aplicación final.

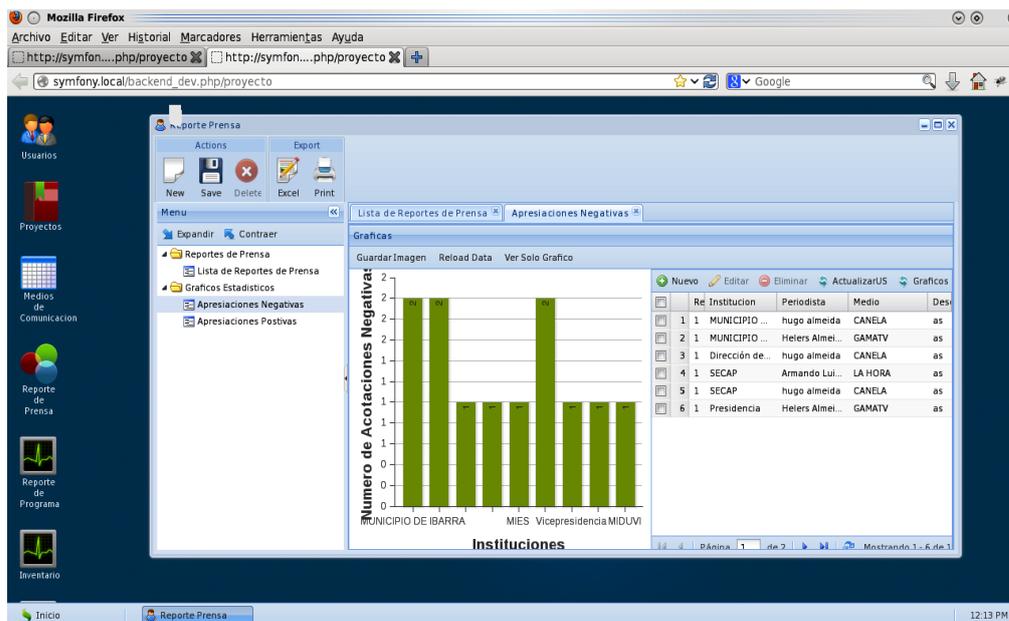
4.3.13 Interfaces de Usuario

4.3.13.1 Inicio



Ilustración 36: Formulario de Inicio.
Fuente: Propia.

4.3.13.2 Menú Principal Backend (Administrador, Periodista, Auditor).



Re	Institucion	Periodista	Medio	Des
1	MUNICIPIO ...	hugo almeida	CANELA	as
2	MUNICIPIO ...	Helers Almei...	GAMATV	as
3	Dirección de...	hugo almeida	CANELA	as
4	SECAP	Armando Lui...	LA HORA	as
5	SECAP	hugo almeida	CANELA	as
6	Presidencia	Helers Almei...	GAMATV	as

Ilustración 37: Menú Principal Backend.
Fuente: Propia.

4.3.13.3 Menú principal Frontend (Cliente)

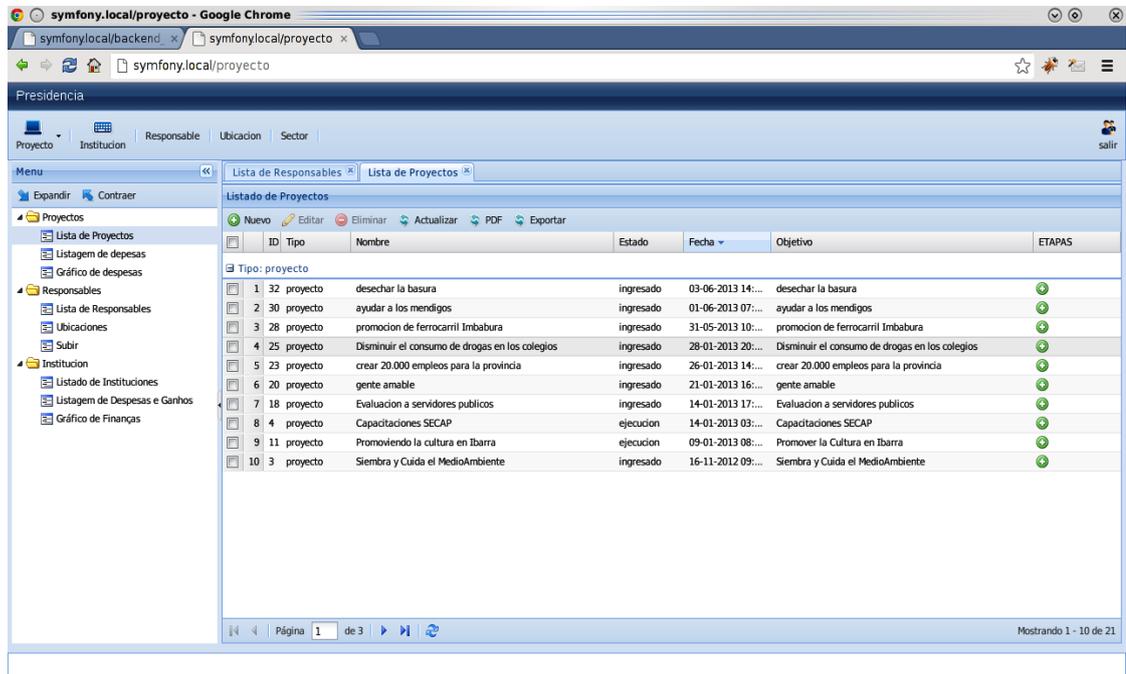


Ilustración 38: Menú principal Frontend.
Fuente: Propia.

4.3.13.4 Ejemplo de Edición, Formulario de Ingreso de Proyectos

4.3.13.4.1 Sección de datos proyecto.

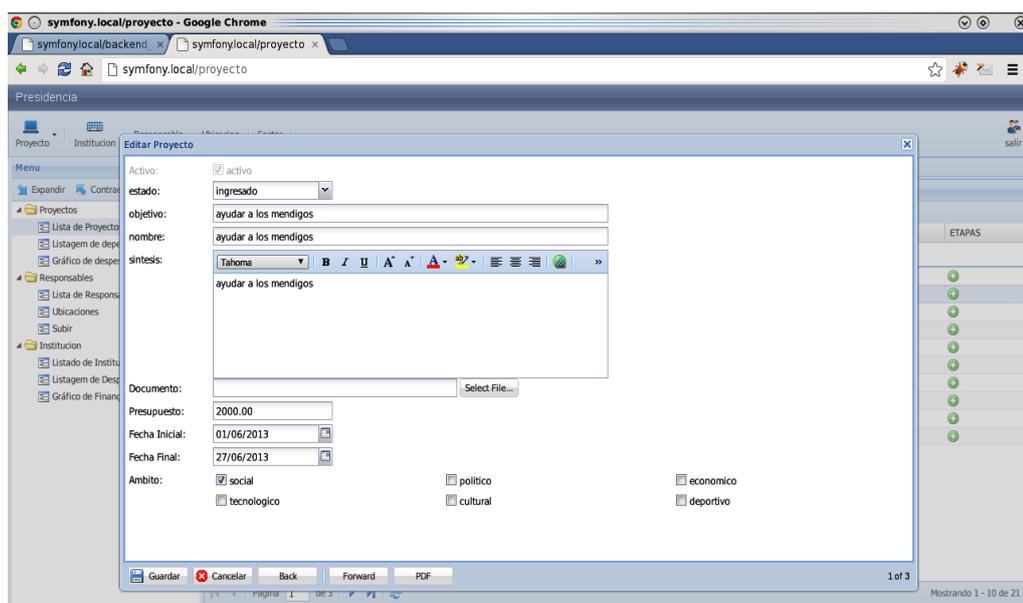


Ilustración 39: Sección de Ingreso de proyecto.
Fuente: Propia.

4.3.13.4.2 Sección de ubicación del proyecto.

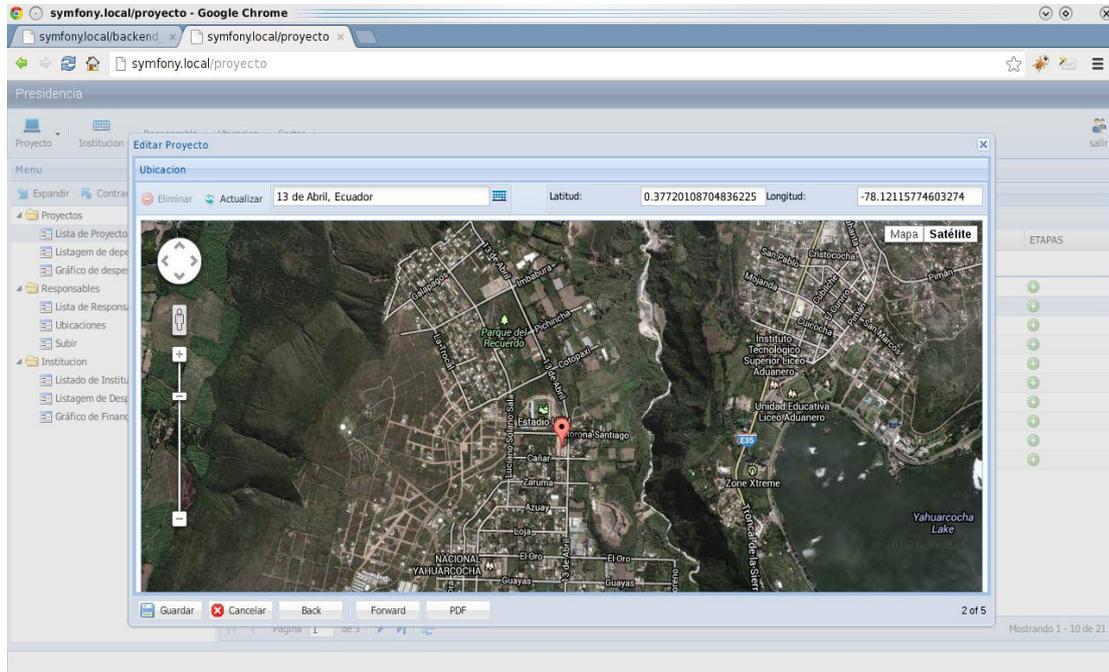


Ilustración 40: Sección de Ubicación del Proyecto.
Fuente: Propia.

4.3.13.4.3 Sección de asignación de objetivos.

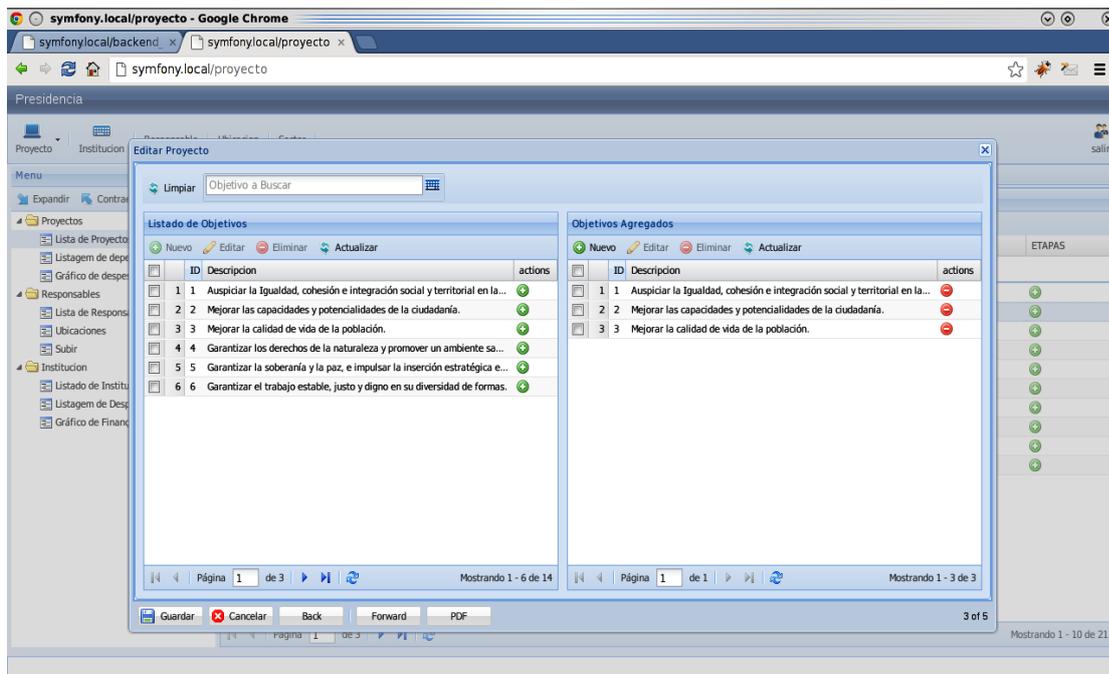


Ilustración 41: Sección de asignación de objetivos.
Fuente: Propia

4.3.14 Estándares de Programación

4.3.14.1 Introducción

Uno de los aspectos más importantes al iniciar un proyecto informático es el desarrollo de normas y políticas que permiten la mejor comprensión de los documentos, código de programación de las bases de datos entre otros recursos.

Si se trata de interpretar los bloques de código de uno o varios proyectos al momento de dar mantenimiento así directamente se perderá tiempo y al momento de realizar cambios los realizara de una manera errónea por lo cual el documento dará gran ayuda para lograr entender la codificación del sistema.

4.3.14.1.1 Propósito

Este artefacto tiene como objetivo dar a conocer a los interesados los estándares de programación que registran el desarrollo y mantenimiento a la aplicación que se va a realizar.

4.3.14.1.2 Descripción

El documento presenta las reglas que permiten estandarizar el desarrollo del sistema “AIRCOM”, utilizando PostgresSql como base de datos y el lenguaje de programación php, mediante el framework de desarrollo Symfony 1.4 y la tecnología de clientes ricos EXTJS 4.1. Para lo cual se debe tener en cuenta lo siguiente:

- ✚ Base de datos.
- ✚ Lenguaje de programación.
- ✚ Resumen de Tecnologías.

4.3.14.2 Estandarización de la Base de Datos

4.3.14.2.1 Normas a Aplicarse.

- ✚ Los nombres de bases de datos, tablas deberán ir en minúsculas.
- ✚ Los nombres de las variables van en singular de forma descriptiva y de manera corta para que sean fáciles de recordar y el esquema se haría más complicado.
- ✚ Es recomendable usar abreviaturas si el nombre es demasiado largo.
- ✚ Los nombres de clave primaria deberán ir precedidos de “id”.
- ✚ Para la creación de índices fki_ seguido de las siglas del nombre de la tabla.
- ✚ Los nombres para las constrains de claves foráneas deberán ir precedidos de “pk_” seguido del nombre de la tabla.

4.3.14.3 Estandarización del Lenguaje de Programación

Symfony al ser un framework de desarrollo de alto rendimiento y de grandes características, utiliza los estándares de programación principales aunque detallaremos algunos para conocimiento general.

Estándar 1: Las funciones deben ser llamadas sin espacios entre el nombre de la función, el signo de paréntesis y el primer parámetro; espacios entre cada coma por parámetros y espacios entre el último paréntesis, el signo de paréntesis cerrado y el signo de punto y coma (;).

Estándar 2: El estilo de los comentarios debe ser como el estilo de los comentarios para C (`/* */` ó `//`).

Estándar 3: Cuando se incluya un archivo de dependencia incondicionalmente utilice “`requiere_once`” y cuando sea incondicionalmente “`include_once`”.

Estándar 4: Siempre utilice las etiquetas `<?php ?>` para abrir un bloque de código. No utilice el método de etiquetas cortas porque esto depende de las directivas de configuración del archivo `PHP.ini` y hace que el script no sea tan portable.

Estándar 5: Los nombres de las clases deben empezar con letra mayúscula. Los nombres de las variables y de las funciones pueden empezar con letra minúscula, por si estas tienen más de una palabra, cada nueva palabra debe empezar con letra mayúscula. Si una función en una clase es privada deberá comenzar con el signo de guion mayor para más fácil identificación.

Estándar 6: Los archivos con código PHP, deben ser guardados en formato ASCII utilizando la codificación ISO-8859-1, ya que este formato es el mismo que se utiliza para guardar archivos de texto plano (.txt). La razón por la que se utiliza este estándar es por la que determinados editores HTML (dreamweaver), agregan códigos de carácter extraño de salto de línea (como si se tratara de un archivo binario) y esto puede ocasionar que el intérprete de PHP, encuentre problemas al momento de leer el script.

4.3.14.4 Resumen de Tecnologías.

El desarrollo completo del Sistema debe realizarse bajo los siguientes parámetros tecnológicos:

Elemento	Tecnología (Frameworks)	Observaciones
Base de datos	PostgreSQL	Gestor de bases de datos, de gran escalabilidad, multiplataforma, estable y confiable, diseñado para ambientes de alto volumen, libre bajo licencia BSD.
Framework	Symfony	Tecnología y Framework para aplicaciones php de entornos web, con un modelo de programación de capas, potente por su fácil desarrollo, desempeño, adopción.
	EXTJS	Tecnología de clientes ricos del lado del cliente basado en java script, que permite crear aplicaciones web con Ajax
	MVC	Patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos
IDE	NetBeans	Entorno de Desarrollo integrado libre, orientado principalmente para el lenguaje PHP y Java.
Servidor web	Apache	Servidor Web libre, compatible con PHP, muy fiable, multiplataforma.
Tipo MIME	CSS	Lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XHTML, separando la estructura y presentación de un documento.
	HTML	Lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto
	JavaScript	Lenguaje de programación interpretado, usado principalmente para mejorar la interfaz de usuario y páginas web dinámicas.
Control de versiones	Subversión	Mantiene el registro de todo el trabajo y los cambios en los ficheros que forman parte de un proyecto.
Sistema operativo	De desarrollo: Centos	Distribución Linux popular en el segmento de servidores web, derivada y con total compatibilidad de Red Hat.

	De producción: Centos	Distribución Linux popular en el segmento de servidores web, derivada y con total compatibilidad de Red Hat.
Herramienta de reportes	Extjs y Fpdf	Herramientas para la creación de reportes con contenido enriquecido.
Metodología	RUP	Metodología estándar utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Tabla 61: Tecnologías empleadas.
Fuente: propia.

4.4 FASE DE CONSTRUCCIÓN

En esta fase de construcción se realiza principalmente diseño e implementación, aquí se evoluciona del prototipo inicial a un producto operacional con requerimientos especificados en la fase 2. Las características de la aplicación son desarrolladas e integradas al producto, además de ser extensamente probadas para su primera versión.

Los documentos generados en este documento son el modelo de datos, plan de pruebas y la lista de riesgos como se muestran a continuación.

4.4.1 Modelo de Datos

Indica cómo están estructurados los diagramas de bases de datos del sistema especificados en el desarrollo del presente proyecto.

4.4.1.1 Introducción

4.4.1.1.1 Propósito

El propósito de este documento es definir y presentar el modelo de datos del sistema de acopio de Información y Medios de Comunicación “AIRCOM”.

4.4.1.1.2 Alcance

Se aplica a los modelos del sistema.

4.4.1.1.3 Definición de Siglas y Abreviaturas

Ver glosario.

4.4.1.1.4 Referencias

Glosario

Resumen de los Requerimientos de los Interesados.

Modelo de Casos de Uso

4.4.1.2 Modelo de Datos

4.5 FASE DE TRANSICIÓN

El Objetivo de esta etapa de RUP es lograr que el usuario quede satisfecho con el trabajo realizado y lograr un producto final tan rápidamente y costo efectivo como sea posible.

En esta fase se realiza la entrega del Sistema, se realiza capacitaciones y se entrega el manual de instalación.

4.5.1 Manual de Instalación

A continuación se detalla como instalar cada una de las herramientas que son necesarias para el presente proyecto.

4.5.2 Introducción

4.5.2.1 Copiar los archivos necesarios para la instalación

En el CD del proyecto se encuentra una carpeta con todo el software necesario, copiar todos esos archivos y pegarlos en el la ubicación de su preferencia.

Se abre una consola, debe de autenticarse como súper usuario (root), acceder desde consola al directorio de en donde ubico previamente los archivos y dar todos los permisos a todos los archivos (chmod 777 *).

Luego de realizar este paso, estamos listos para proceder a instalar el software.

4.5.2.2 Instalación y Configuración de herramientas

La Instalación de herramientas de desarrollo para el Sistema de Acopio de Información Interinstitucional y Reporte de Medios de Comunicación se detalla a continuación, el Sistema Operativo sobre el cual estarán instaladas es Centos en la versión 5.4.

4.5.2.2.1 Apache

Por lo general el sistema operativo Centos al ser un sistema de características ideales para definirlo como un servidor, trae por defecto ya instalado apache, caso contrario, procedemos a instalarlo con el siguiente comando:

```
yum install httpd
```

Una vez instalado lo configuramos para que inicie automáticamente:

```
chkconfig --levels 235 httpdon
```

y lo iniciamos:

```
/etc/init.d/httpdstart
```

Accedemos la navegador web de su preferencia y verificamos si apache se ha instalado correctamente tecleando en la barra de dirección `127.0.0.1` o simplemente `localhost`.

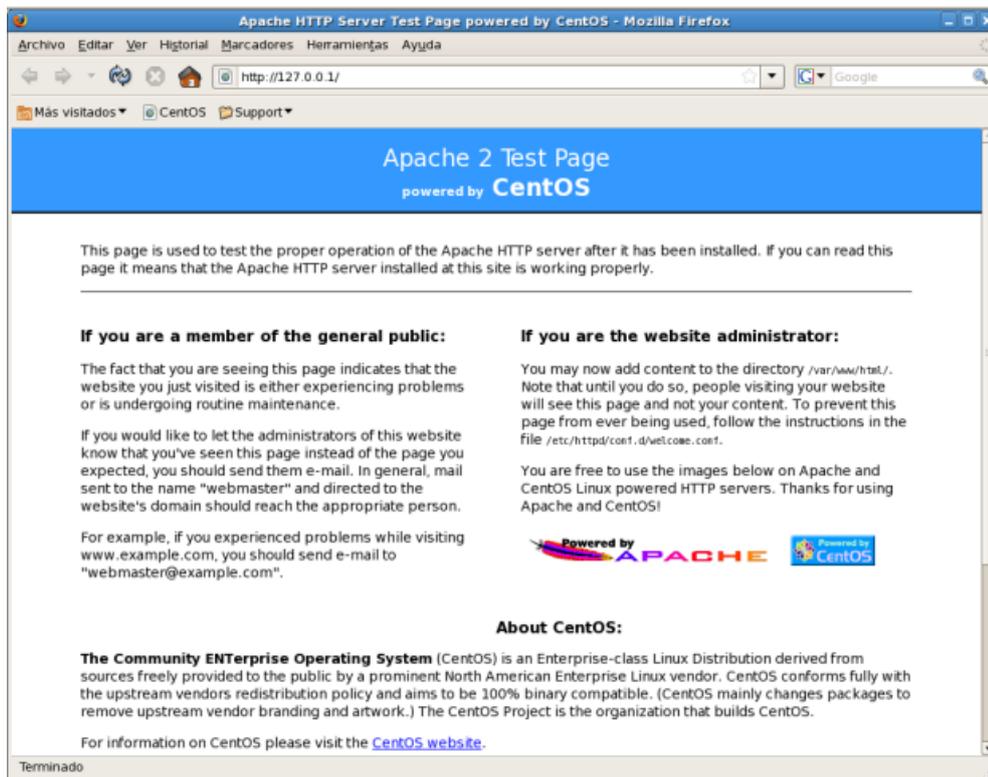


Ilustración 42: Instalación de Apache.
Fuente: propia

4.5.2.2 PHP

Ahora instalamos PHP en la versión 5.3 que es la utilizada en el desarrollo del sistema y lo vinculamos a Apache, para actualizar a PHP en CentOS realizamos los siguientes pasos:

Actualizar CentOS

```
# yum update
```

Reiniciar

```
# reboot
```

Desinstalar php-5.1.x

```
# yum removephp*
```

Instalar php-5.3, se han seleccionado varias extensiones:

```
# yum install php53 php53-mbstring php53-gd \
php53-mysql php53-xml php53-soap php53-pdo
```

Para integrar PHP, con PostgreSQL que es la base de datos que utilizaremos debemos localizar el archivo `php.ini` que se encuentra en la Ubicación para el caso de Centos 5.4 en `/etc/php.ini`

```
nano/etc/php.ini
```

y procedemos a des comentar (quitar los puntos y comas) de las líneas siguientes:

```
;extension=php_pgsq1.dll
;extension=php_pdo_pgsq1.dll
```

De tal manera que nos quedara:

```
extension=php_pgsq1.dll
extension=php_pdo_pgsq1.dll
```

De esta sencilla forma se instalara PHP 5.3. Para comprobar la versión:

```
# php -v
PHP 5.3.9 (cli) (built: Jan 10 2012 22:05:12)
Copyright (c) 1997-2012 The PHP Group
Zend Engine v2.3.0, Copyright (c) 1998-2012 Zend Technologies
Generamos un archivo para probar que PHP funciona correctamente usando
```

```
nano /var/www/html/info.php
echo '<?phpphpinfo(); ?>' > /var/www/html/info.php
```

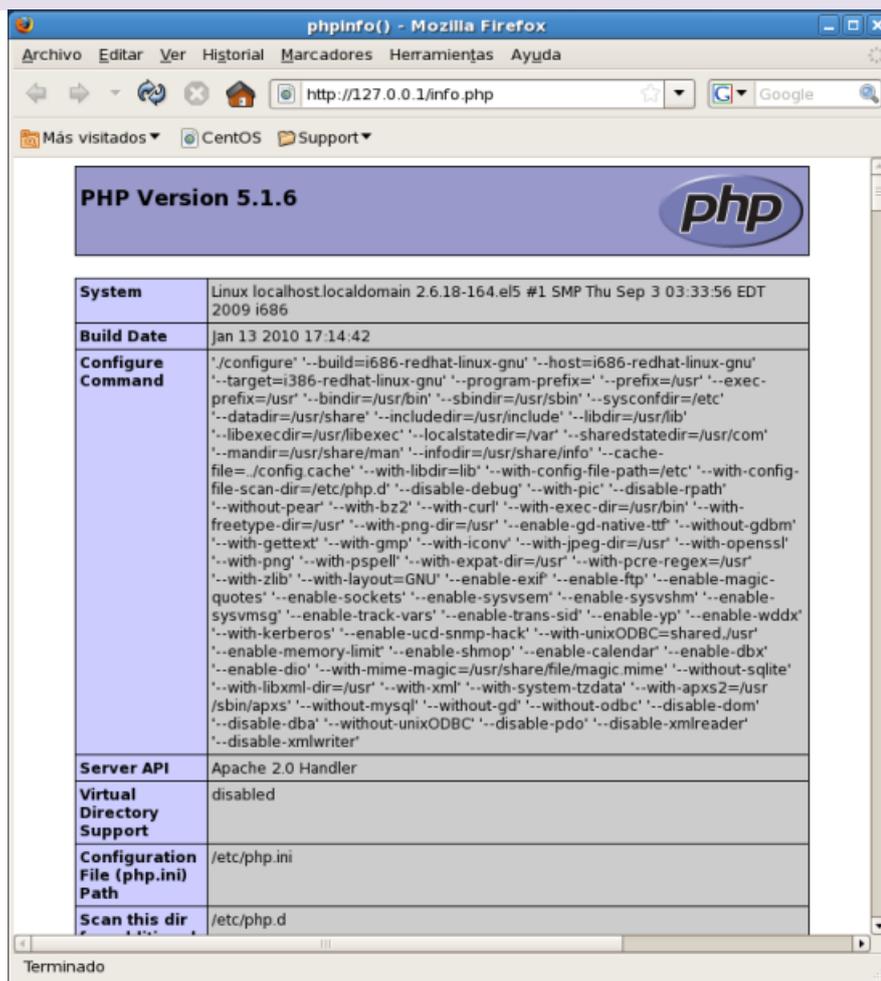


Ilustración 43: Instalación de PHP.
Fuente: propia

4.5.2.2.3 PostgresSql

Una vez activados los repositorios tenemos que instalar el servidor de base de datos:

```
$>yum install postgresql postgresql-server
```

Es posible que en este paso nos dé un problema de librerías por culpa de un paquete llamado **apr-util**. En este caso deberemos instalar primero este paquete (por separado) y luego volver a lanzar la instalación del servidor:

```
$>yum install apr-util
$>yum install postgresql postgresql-server
```

La solución al problema de dependencias no debería reproducirse si instalamos el servidor de bases de datos en estos dos pasos.

Arrancar el servidor y habilitar conexiones remotas

Para arrancar el servidor seguiremos los pasos siguientes:

```
$>service postgresql initdb
$>service postgresql start
```

Con esto ya tendremos el servidor de base de datos arrancado y aceptando conexiones en local, pero no podremos acceder al servidor desde conexiones remotas. Para habilitar las conexiones remotas debemos asegurarnos que no hay ningún tipo de bloqueo que no sea el del servidor Postgre, modificar dos ficheros y reiniciar el servicio. El primer fichero que vamos a modificar es **/var/lib/pgsql/data/pg_hba.conf**.

En **pg_hba.conf** se tiene que dar acceso a la red desde la que vamos a acceder (dirección IP), Para dar acceso a una red añadiremos lo siguiente al fichero:

```
Host all all 192.168.1.5/0 trust
```

El segundo fichero que debemos modificar es **/var/lib/pgsql/data/postgresql.conf**. En este debemos buscar la línea que pone:

```
listen_addresses='localhost'
```

Y poner:

```
listen_addresses='*'
```

Una vez realizados estos cambios se tiene que reiniciar el servicio:

```
$> /etc/init.d/postgresqlrestart
```

Cuando el servicio arranque de nuevo debería ser posible establecer conexiones remotas al servidor con el usuario postgres y el password que le corresponda. En caso de que no se haya definido password para dicho usuario, éste se puede setear de forma manual. Como usuario ROOT ejecutamos el siguiente comando:

```
Passwd postgres
```

El sistema nos pedirá que introduzcamos dos veces el password.

4.5.2.2.4 NetBeans

NetBeans que es un IDE de desarrollo de aplicación que facilita al programador las tareas de desarrollo, para Instalar la versión 7.2 que es en la que se desarrolló la aplicación:

Primero vamos a descargar de la página oficial de Oracle la versión de Netbeans en la que viene integrado conjuntamente con Sun's Java debido a que NetBeans utiliza OpenJDK para su funcionamiento, podemos descargar esta versión del siguiente enlace:

```
http://www.oracle.com/technetwork/java/javase/downloads/jdk-7-netbeans-download-432126.html
```

Una vez descargado el archivo procedemos a dar permisos y ejecutarlo con los siguientes comandos:

```
# chmod +x jdk-7u10-nb-7_2_1-linux-i586-m1.sh  
# chmod +x jdk-7u10-nb-7_2_1-linux-x64-m1.sh
```

Se desplegara el siguiente asistente para la instalacion de NetBeans

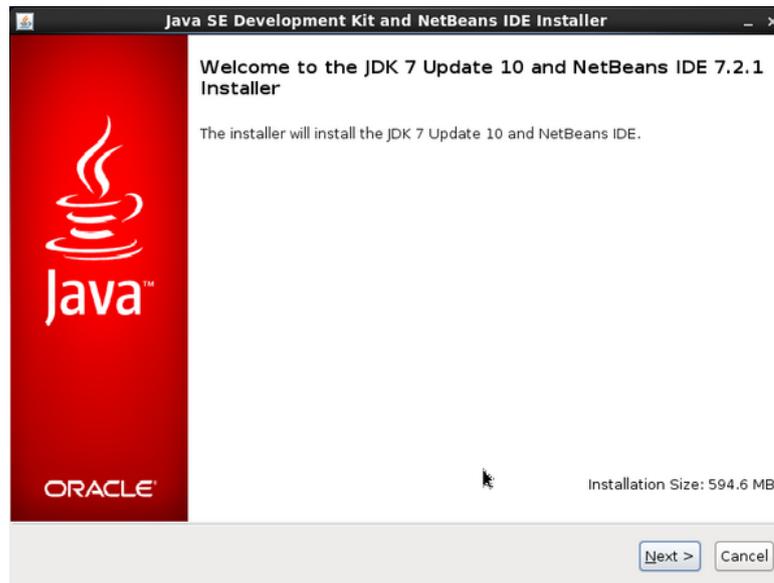


Ilustración 44: Instalación NetBeans.
Fuente: propia

Aceptamos la licencia **jUnitLicenseAgreement** y damos click en el botón Next.

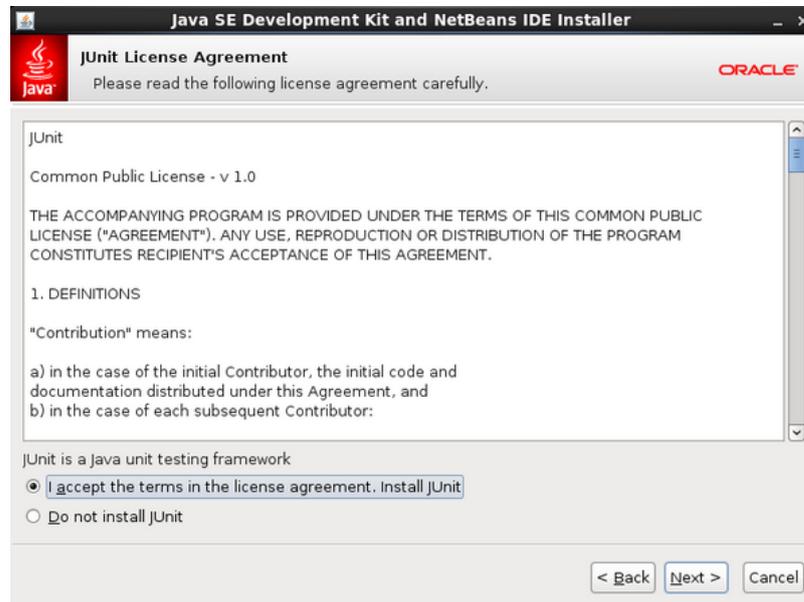


Ilustración 45: Licencia NetBeans.
Fuente: propia

Establecemos la ubicación para la carpeta donde se ubicará **JDK**.

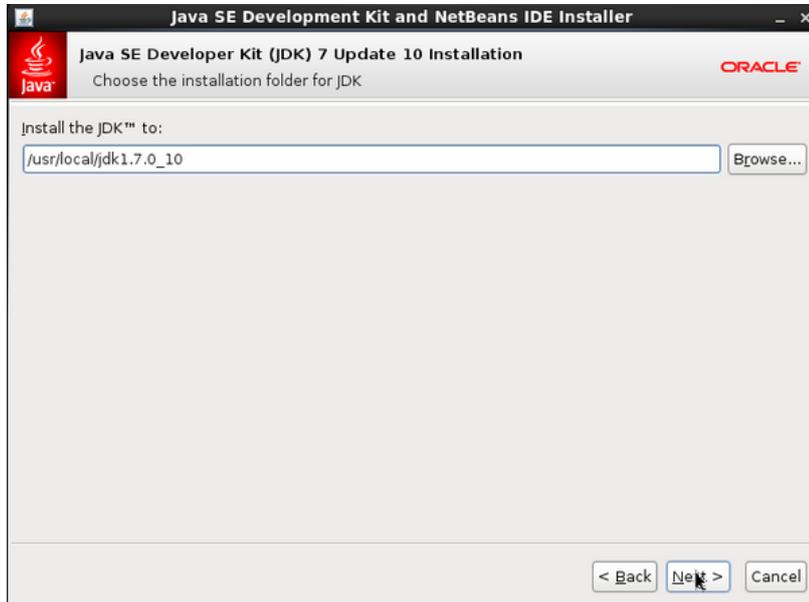


Ilustración 46: Ubicación JDK.
Fuente: propia

Establecemos la ubicación para la instalación del **IDE** NetBeans.

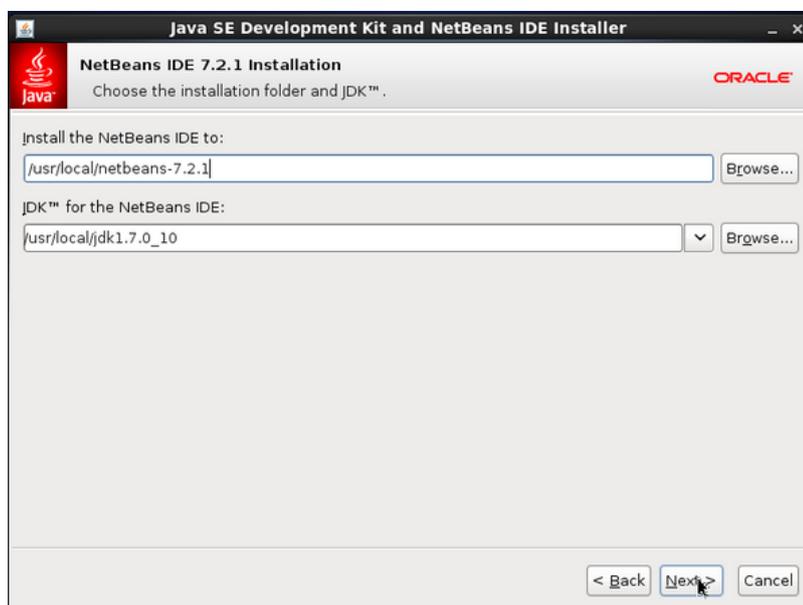


Ilustración 47: Ubicación IDE NetBeans.
Fuente: propia

A continuación, hacemos clic en "Install" para iniciar el proceso de instalación. En la primera puesta en marcha NetBeans IDE, el instalador se descargará e instalará la librería JUnit.



Ilustración 48: Iniciar el proceso de instalación NetBeans
Fuente: propia

Se completara el proceso de Instalación

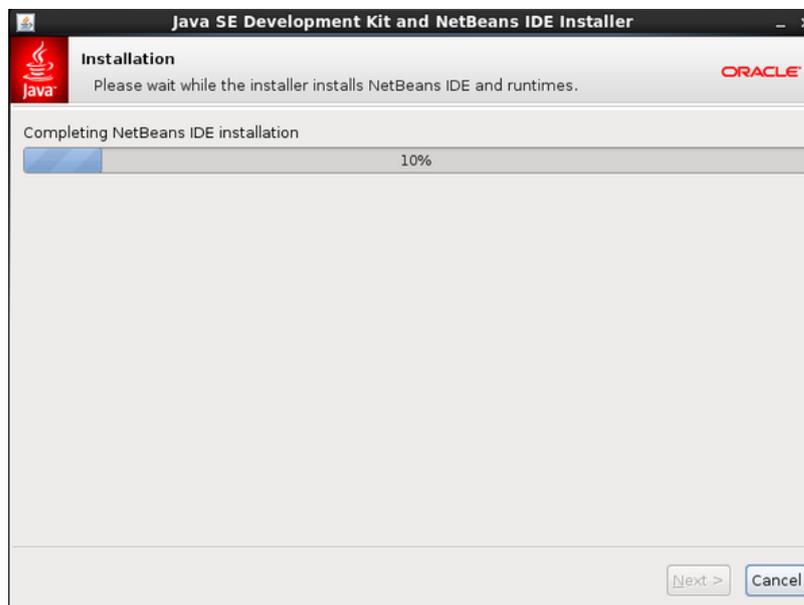


Ilustración 49: Progreso de Instalación NetBeans
Fuente: propia

Una vez completado el proceso de Instalacion Hacemos click en el boton “**Finish**”.

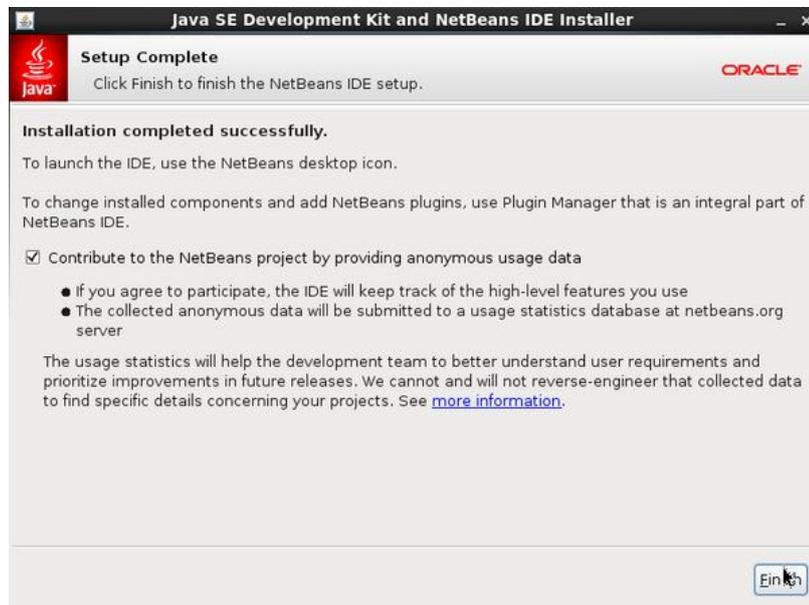


Ilustración 50: Finalización de Instalación NetBeans.
Fuente: propia

Para iniciar el IDE NetBeans, utilice el siguiente comando.

```
# /usr/local/netbeans-7.2.1/bin/netbeans
```

Y listo se desplegara el IDE de desarrollo NetBeans en su versión 5.2

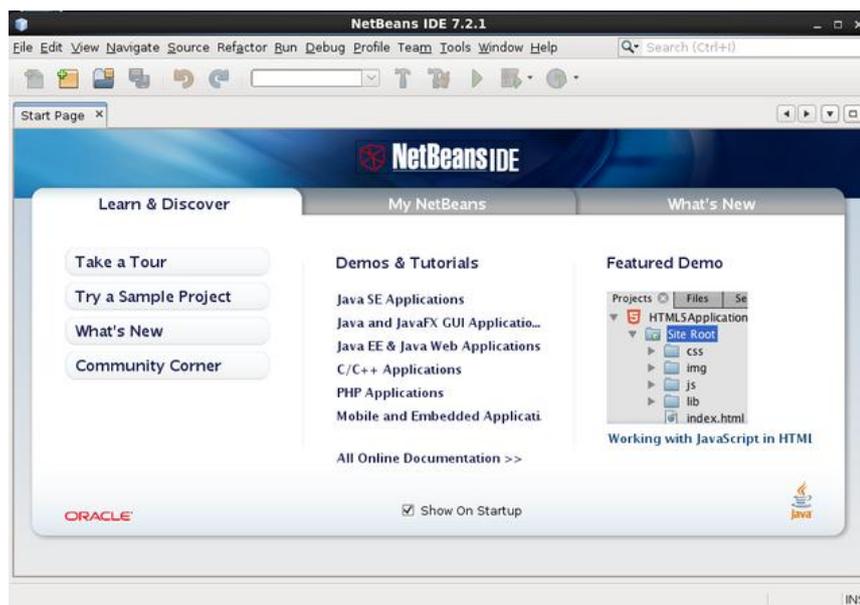


Ilustración 51: Entorno NetBeans
Fuente: propia

4.5.2.2.5 Framework Symfony 1.4

En primer lugar, creamos un directorio donde vamos a guardar todos los archivos relacionados con el proyecto, como indicamos anteriormente el sistema operativo donde escogido es Centos:

```
$ mkdir -p /home/sfprojects/testrepo
$ cd /home/sfprojects/testrepo
```

Descargamos Symfony en la versión 1.4 que es la escogida para el desarrollo de la aplicación del siguiente enlace:

```
http://symfony.com/download#symfony1
```

Ubicamos el archivo en el directorio lib/vendor que acabamos de crear y lo descomprimos:

```
$ cd lib/vendor
$ tarzxpf symfony-1.4.0.tgz
$ mv symfony-1.4.0 symfony
$ rm symfony-1.4.0.tgz
```

Como las configuraciones PHP varían mucho de una distribución a otra, tenemos que comprobar que nuestra configuración PHP cumple los requisitos mínimos de Symfony. Iniciamos el script de comprobación de la configuración que viene con Symfony desde la línea de comandos:

```
$ cd ../../
$ php lib/vendor/symfony/data/bin/check_configuration.php
```

Nos deberá aparecer la siguiente vista:

```
*****
*
*  symfony requirements check  *
*
*****

php.ini used by PHP: /apache2/php/etc/php.ini
```

```

** Mandatory requirements **

OK      requires PHP >= 5.2.4
OK      php.ini: requires zend.zel_compatibility_mode set to off

** Optional checks **

OK      PDO is installed
OK      PDO has some drivers installed: sqlite2, sqlite, mysql
OK      PHP-XML module installed
[[WARNING]] XSL module installed
*** Install the XSL module (recommended for Propel) ***
OK      can use token_get_all()
OK      can use mb_strlen()
OK      can use iconv()
OK      can use utf8_decode()
OK      has a PHP accelerator
OK      php.ini: short_open_tag set to off
OK      php.ini: magic_quotes_gpc set to off
OK      php.ini: register_globals set to off
OK      php.ini: session.auto_start set to off

```

Ilustración 52: Instalación Symfony 1.4

Fuente: http://symfony.com/legacy/doc/jobee/1_4/es/01?orm=Doctrine

La configuración del Servidor Web

Ahora es el momento de cambiar la configuración de Apache para que el nuevo proyecto sea accesible.

Buscamos y abrimos el archivo de configuración httpd.conf y añadimos la siguiente configuración al final:

```

# Listen for virtual host requests on all IP addresses
NameVirtualHost *:80

# Acopio con symfony 1.4
<VirtualHost *:80>
DocumentRoot "/home/sfprojects/testrepo/web"
DirectoryIndex index.php
ServerNamesymfony.local
Alias /sf /home/sfprojects/testrepo/lib/vendor/symfony/data/web/sf
<Directory "/home/sfprojects/testrepo/web">
AllowOverride All
    Allow from All
</Directory>
</VirtualHost>

```

Ahora procedemos a integrar el Framework Symfony 1.4 en el IDE de desarrollo NetBeans 7.2. En primer lugar abrimos NetBeans y vamos *Tools > Options*, hacemos clic en PHP, vamos a *symfony* y en la opción *Symfony script* ponemos:

`/home/sfprojects/testrepo/lib/vendor/Symfony/data/bin/symfony`, tal y como se muestra en la siguiente imagen.

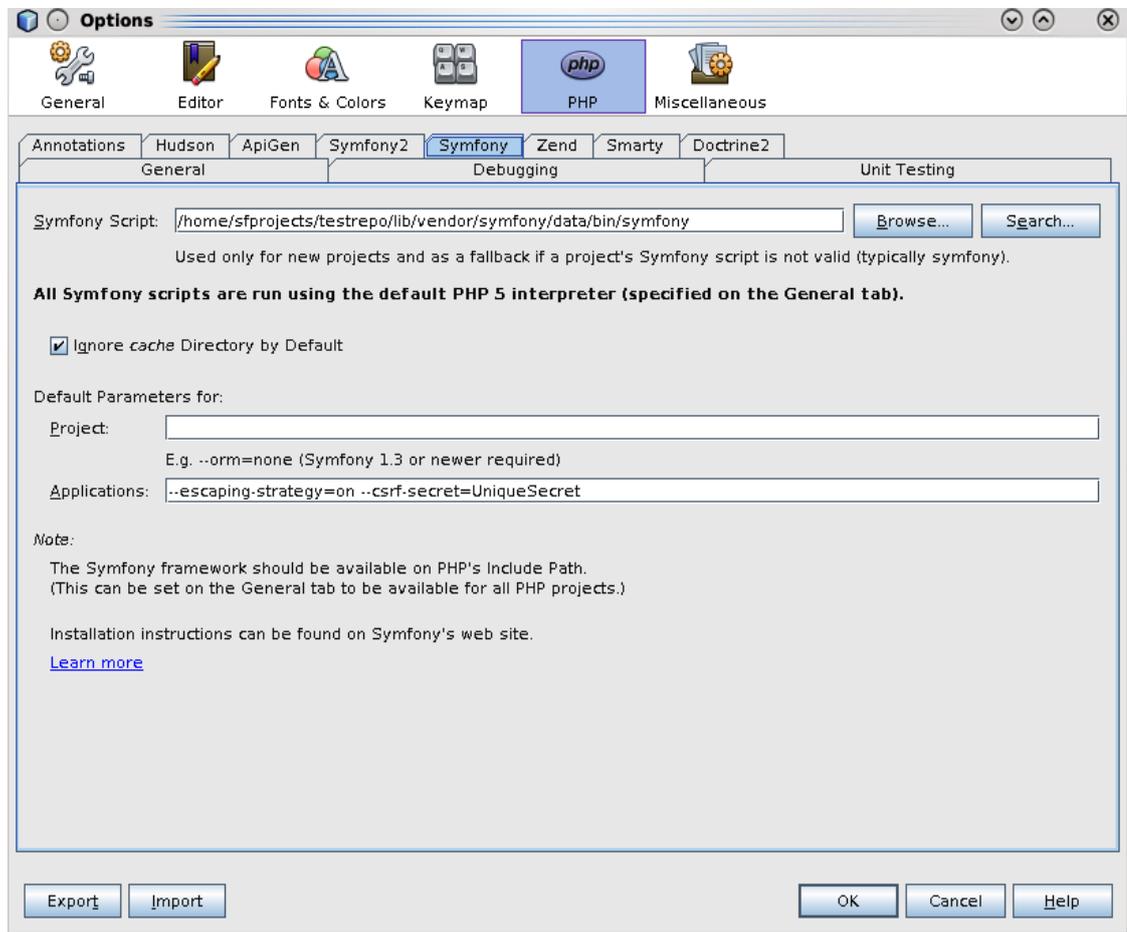


Ilustración 53: Configuración de Symfony con NetBeans
Fuente: Propia.

Ahora probamos que la integración se ha realizado de forma correcta, dentro de NetBeans nos vamos a la carpeta de la aplicación creada anteriormente *testrepo* y hacemos clic derecho sobre la misma, escogemos la opción *RunSymfonyCommand* nos deberá aparecer lo siguiente:

Y listo tenemos integrados el framework de desarrollo Symfony 1.4 y el IDE de desarrollo NetBeans en la versión 7.2.1, ya podemos generar código desde la ventana *RunSymfonyCommand*(Acopio).

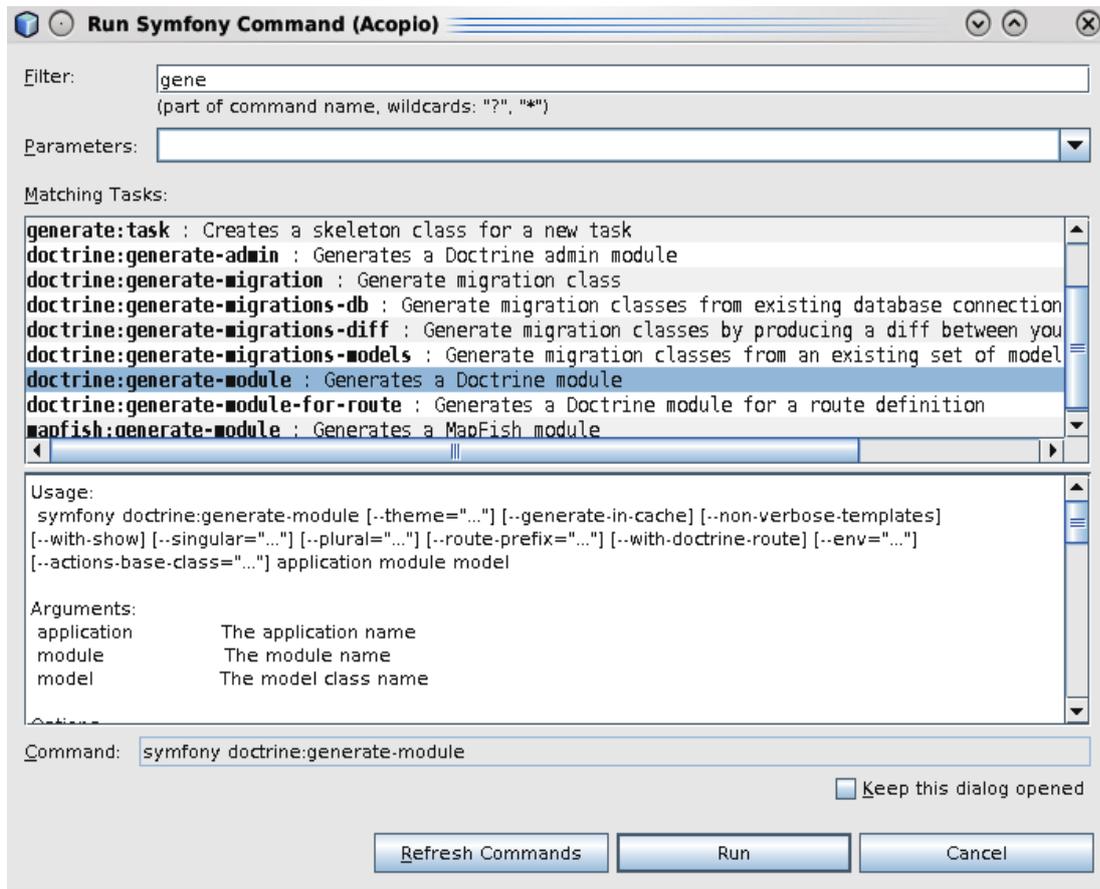


Ilustración 54: Integración Symfony con NetBeans.
Fuente: Propia.

4.5.2.2.6 ExtJs 4.1

ExtJs (pronunciado como "ekst") es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM.

Para integrar ExtJs al desarrollo de la aplicación primeramente lo descargamos del siguiente enlace:

<http://www.sencha.com/blog/ext-js-4-1-final-released>

Descomprimos la carpeta copiamos y ubicamos únicamente los archivos siguientes:

```
ext-all.js
ext-all-debug.js
ext-all-dev.js
```

Bajo el directo de la aplicación siguiente `/home/sfprojects/testrepo/web/js/Ext`, copiamos además el archivo `ext-all.css`, en la ruta `/home/sfprojects/testrepo/web/js/css`,

```
ext-all.css
```

Lo último por hacer, para completar la integración entre, la aplicación Symfony y el cliente Extjs es ir al archivo `/home/sfprojects/testrepo/apps/backend/config/view.yml` editarlo como muestra la figura:

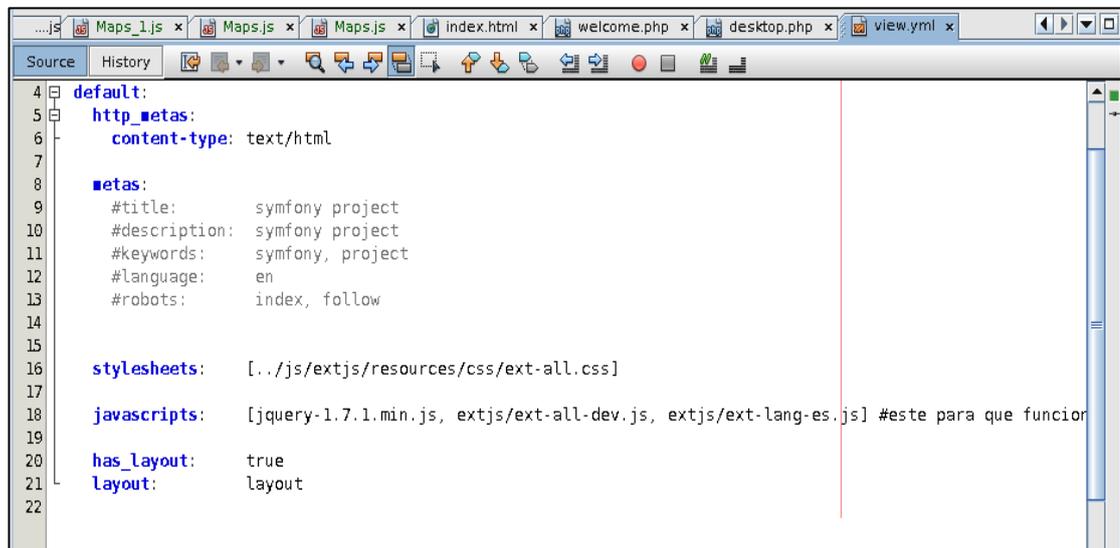


Ilustración 55: Integración ExtJs con Symfony
Fuente: Propia.

4.5.2.2.7 PgAdmin III

PgAdmin 3 es una herramienta de código abierto para la administración de bases de datos PostgreSQL.

En el sistema operativo Centos, podemos instalar *PgAdmin3* de muchas maneras a continuación se describe una de ellas:

1. Instalamos epel

```
rpm -Uvh http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-4.noarch.rpm
```

2. Instalamos rpmfusionrepel

```
rpm -Uvh
http://download1.rpmfusion.org/free/el/updates/testing/5/i386/rpmfusion-free-release-5-0.1.noarch.rpm
```

3. Por ultimo *pgAdmin III*

```
Yum install pgadmin3
```

Luego de completada la instalación, ya podremos abrir el programa e ingresamos las configuraciones necesarias para establecer la conexión con la base de datos que haya sido creada, en este caso el servidor será localhost, la base de datos y usuario serán postgres.

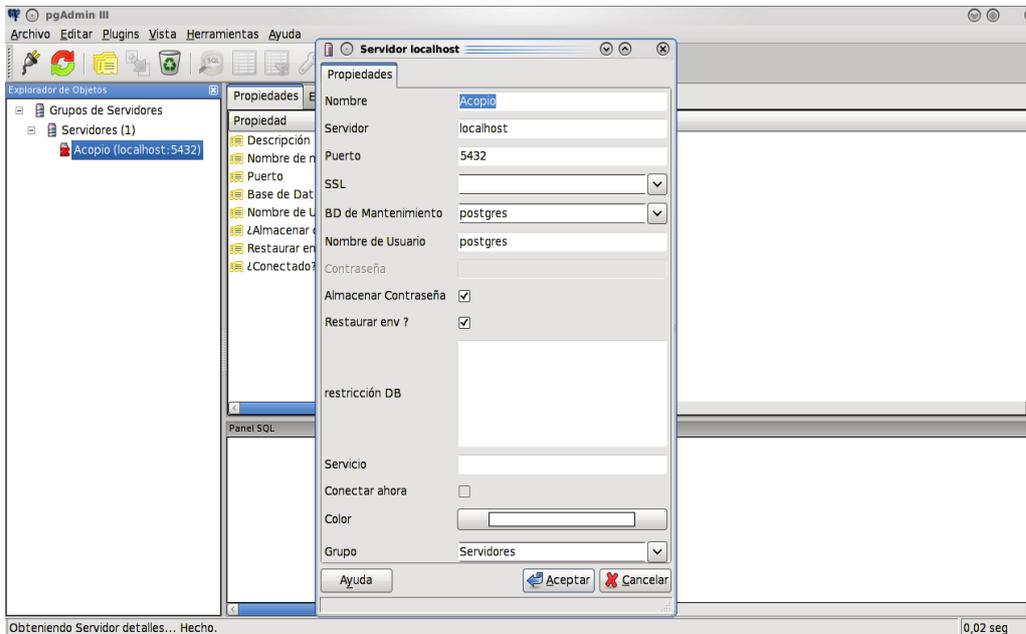


Ilustración 56: Entorno PgAdminIII.
Fuente: Propia.

Listo, ya podemos crear funciones, procedimientos y manipular las tablas de la base de datos en general de una manera más visual y práctica.

4.5.2.2.8 PowerArchitect

SQL Power Architect es una aplicación de modelado de datos fácil de usar, diseñada para permitir a los usuarios realizar ingeniería inversa de las bases de datos existentes, realizar creación de perfiles de datos en las bases de datos de fuente y conectarse a bases de datos específicas. Para instalar Power Architect simplemente descargamos el archivo de la página oficial:

http://www.sqlpower.ca/page/architect_download_os

Luego lo descomprimos en la ubicación de nuestra preferencia y lo ejecutamos de la siguiente manera usando el uso de java.

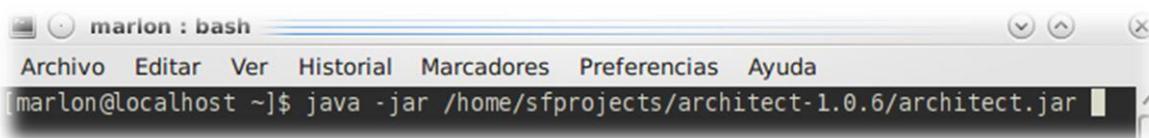


Ilustración 57: Iniciar PowerArchitect
Fuente: Propia.

Nos aparecerá una ventana como la siguiente, donde podremos crear tablas, realizar ingeniería inversa, generar sql de acuerdo al modelo generado, etc.

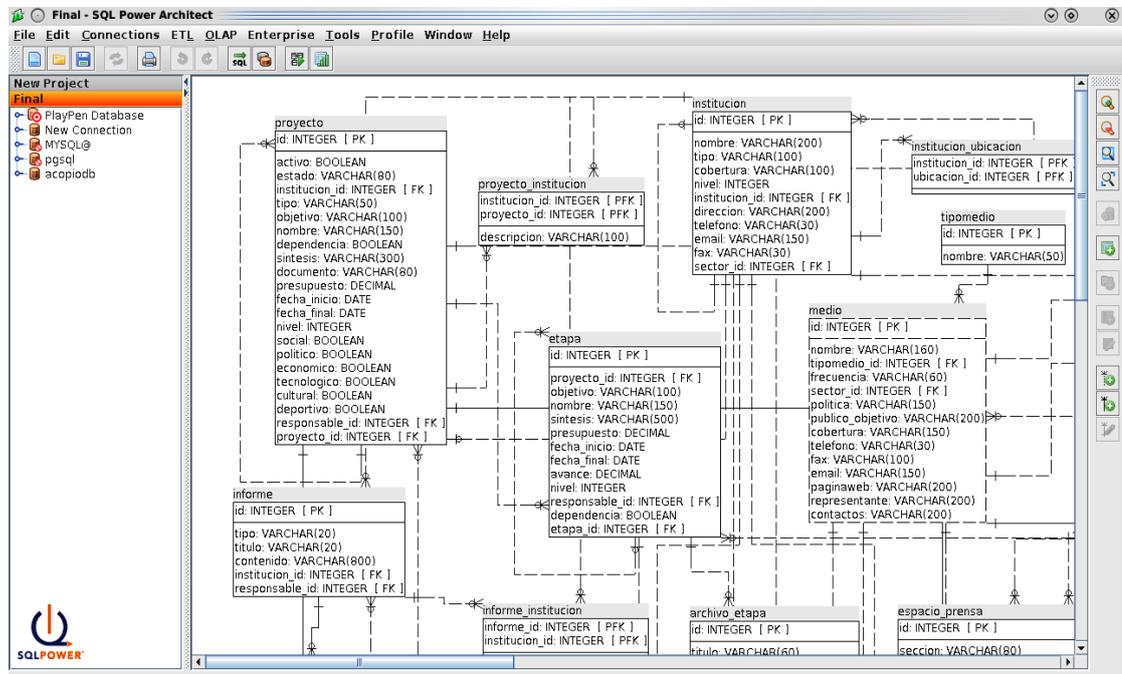


Ilustración 58: Entorno Power Architect.
Fuente: Propia.

Para poder conectarse con una base de datos existente primeramente, debemos establecer la conexión con la base de datos, ingresamos de la siguiente manera:

The screenshot shows the 'Database Connection: PlayPen Database' dialog box. It has two tabs: 'General' and 'Kettle'. The 'General' tab is active and contains the following fields:

- Connection Name: PlayPen Database
- Database Type: PostgreSQL
- Connect Options: Hostname localhost, Port 5432, Database acopiodb
- JDBC URL: jdbc:postgresql://localhost:5432/acopiodb
- Username: dmarlon
- Password:

Below the fields is a 'Test Connection' button. Below that, the text reads: 'Connection test successful', 'Database Product Name: PostgreSQL', 'Database Product Version: 8.4.13', 'Database Driver Name: PostgreSQL Native Driver', and 'Database Driver Version: PostgreSQL 8.2 JDBC3 with SSL (bui'. At the bottom are 'OK' and 'Cancel' buttons.

Ilustración 59: Configuración Power Arquitect.
Fuente: Propia.

Capítulo 5: Conclusiones y Recomendaciones.



RESUMEN DEL CAPÍTULO

- *Conclusiones.*
- *Recomendaciones*

5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- 1) El desarrollo de estos módulos permitió alcanzar los objetivos planteados.
- 2) Se pudo cubrir las necesidades de la Gobernación de Imbabura, para la mejor articulación e intercambio de información entre Instituciones y el control de los principales Medios de Comunicación de la Provincia.
- 3) El uso de herramientas libres no solo es de gran ayuda al momento de producir software, además se cumple con todas las normas de calidad, y políticas de utilización de software libre establecidas por parte del Estado.
- 4) Se creó e implementó los módulos descritos en este proyecto, con los cuales se pudo contar con un sistema integrado de calidad para la administración y control de los procesos de la Gobernación de Imbabura.
- 5) Todas las instituciones que tiene relación de algún tipo con la Gobernación pueden enviar su información de una forma automática, rápida y eficiente.
- 6) La metodología de desarrollo RUP proporciona un acercamiento disciplinado a la asignación de tareas y responsabilidades en el transcurso de desarrollo de un proyecto de software.
- 7) La utilización del modelo MVC permite separar muchos aspectos y brinda al programador reutilización de código y flexibilidad que son importantes en la programación orientada a objetos.
- 8) El Framework de desarrollo Symfony facilita gran parte de los accesos a la base de datos pero siempre con un coste de tiempo de respuesta.

5.2 Recomendaciones

- 1) Es recomendable la implementación de un módulo de administración el cual nos permita mantener una correcta gestión de usuarios, y así mantener un nivel de privacidad y control de accesos a la información.
- 2) Se recomienda utilizar la metodología RUP en proyectos de Desarrollo de Software para mantener la información ordenada y documentada durante todo el proceso de desarrollo e implementación, utilizando las plantillas que proporciona RUP, ya que facilitan y garantizan el producto software.
- 3) Al momento de estudiar nuevas herramientas, primero se debe verificar que exista la suficiente documentación para tener soluciones rápidas y precisas a nuestros problemas.
- 4) Al ver el funcionamiento y resultado obtenido se recomienda utilizar software libre para el desarrollo y la implementación de aplicaciones o sistemas web, ya que el software libre mantiene las características del software comercial.
- 5) Se recomienda utilizar tecnologías RIA (Aplicaciones Ricas en Internet), ya que brinda al usuario experiencia de muy parecida o igual a la que se tiene en las aplicaciones de escritorio, mejorando significativamente la presentación y diseño de la aplicación.
- 6) Ampliar la funcionalidad de obtención de reportes con información generada en los módulos para mayor interpretación de datos y resultados.
- 7) Seguir contribuyendo con herramientas y aplicaciones tecnológicas que ayuden al trabajo conjunto de las instituciones gubernamentales para el bienestar de la ciudadanía.

5.3 GLOSARIO

5.3.1 Introducción

El presente glosario forma parte del sistema “AIRCOM” y es una recopilación de términos empleados en todo el sistema.

Este glosario no es un diccionario informático; representa una guía general de las definiciones y conceptos que se usan con frecuencia y tienen una relevante importancia para la adecuada comprensión dentro del sistema.

5.3.2 Propósito

La función principal del presente glosario es proporcionar un estándar de términos y conceptos asociadas a los términos técnicos del sistema.

5.3.3 Referencias

Éste documento hace referencia a términos empleados en toda la documentación del sistema.

5.3.4 Glosario

AIRCOM.- Nombre del sistema financiero desarrollado.

ACTIONS.- Métodos de capa de vista, para el uso de componentes en páginas jsp.

ACTORES O ROLES.- Son los personajes encargados de la realización de las actividades definidas dentro de los flujos de trabajo de cada una de las disciplinas del RUP.

AJAX (Asynchronous JavaScript and XML). - Es una técnica de desarrollo web para crear aplicaciones interactivas

API.- Interfaz de Programación de Aplicaciones (**ApplicationProgrammingInterface**) es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

ARTEFACTO.- Un artefacto es un producto tangible resultante del proceso de desarrollo de software. En ocasiones un artefacto puede referirse a un producto

terminado, pero más habitualmente se refiere a la documentación generada a lo largo del desarrollo del producto.

CASOS DE USO.- Es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.

CSS.- El nombre hojas de estilo en cascada viene del inglés **CascadingStyleSheets**.

CRUD.- Es el acrónimo de crear, obtener, actualizar y eliminar, es decir las funciones básicas en bases de datos o en la capa de persistencia de un sistema de software.

DIAGRAMAS DE CLASE.-Muestra las clases (descripciones de objetos que comparten características comunes) que componen el sistema y cómo se relacionan entre sí.

DIAGRAMAS DE SECUENCIA.-Muestra una interacción a lo largo del tiempo, además muestra las instancias participantes en una interacción y su “línea de vida”,

FRAMEWORK.- Es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación.

GNU/Linux.- GNU/Linux es uno de los términos empleados para referirse a la combinación del núcleo o kernel libre similar a Unix denominado Linux, que es usado con herramientas de sistema GNU.

HTML.- (Siglas de **HyperTextMarkupLanguage**) Lenguaje de marcado de hipertexto.

JDEVELOPER.- Es un entorno de desarrollo integrado desarrollado por Oracle.

MAPPING.-Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional.

NETBEANS.- Es un IDE, Entorno Integrado de Desarrollo.

ORM.- Es el Mapeo Objeto-Relacional (**Object-RelationalMapping**) es una técnica de programación para convertir datos entre el lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, utilizando un motor de persistencia.

POO.-Programación Orientada a Objetos.

RUP.- El Proceso Unificado de Rational (**RationalUnifiedProcess**) es un proceso de desarrollo de software y junto con el UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

UML.-UnifiedModelingLanguage. Lenguaje Unificado de Modelado, notación estándar para el modelado de sistemas software.

YUM.- Es una colección de herramientas que sirven para automatizar el proceso de instalación, actualización, configuración y eliminación de paquetes de software.

6 REFERENCIAS BIBLIOGRÁFICAS.

- López Guimera, E. (2013). <http://www.h2e.es>. Retrieved from <http://www.h2e.es>:
<http://www.h2e.es/diccionario/css-hojas-estilo>
- Sánchez Rosas, J. (2008). <http://blogs.antartec.com>. Retrieved from
<http://blogs.antartec.com>:
<http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>
- <http://es.wikipedia.org>. (2013, Mayo 28). Retrieved from <http://es.wikipedia.org>:
<http://es.wikipedia.org/wiki/PHP>
- <http://es.wikipedia.org>. (2013, Marzo 14). Retrieved from <http://es.wikipedia.org>:
http://es.wikipedia.org/wiki/Ext_JS
- <http://es.wikipedia.org>. (2013, Mayo 14). Retrieved from <http://es.wikipedia.org>:
<http://es.wikipedia.org/wiki/AJAX>
- <http://librosweb.es>. (2013). Retrieved from <http://librosweb.es>:
http://librosweb.es/symfony_1_2/capitulo_5/entornos.html
- <http://symfony.com>. (2013, Junio 13). Retrieved from <http://symfony.com>:
http://symfony.com/legacy/doc/jobee/1_4/es?orm=Doctrine
- <http://symfony.com>. (2013, Junio 09). Retrieved from <http://symfony.com>:
http://symfony.com/legacy/doc/jobee/1_4/es/13?orm=Doctrine
- <http://www.digitallearning.es>. (2013). Retrieved from <http://www.digitallearning.es>:
<http://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/>
- Domingo Muñoz, J. (2013). www.josedomingo.org. Retrieved from www.josedomingo.org:
www.josedomingo.org/web/mod/resource/view.php?id=2049
- Horna García, M., & Lara Vasquez, J. (2008, Octubre 15). <http://www.slideshare.net>. Retrieved from <http://www.slideshare.net>: <http://www.slideshare.net/njelov/ajax-huancayo>
- Jora, M. (2011, Noviembre). <http://www.buenastareas.com>. Retrieved from
<http://www.buenastareas.com>: <http://www.buenastareas.com/ensayos/Curso-Extjs/3156528.html>
- Luján Mora, S. (2013, Junio 08). <http://es.wikipedia.org>. Retrieved from
<http://es.wikipedia.org>: <http://es.wikipedia.org/wiki/HTML>
- Martínez, R. (2012). <http://www.linux-es.org>. Retrieved from <http://www.linux-es.org>:
<http://www.linux-es.org/node/536>
- The Rational Unified Process, An Introduction.” de Philipp Kruchten.

7 ANEXOS

LOS SIGUIENTES ARCHIVOS SE LOS ENCONTRARÁ EN LA CARPETA DE ANEXOS, UBICADA EN CD DE ESTE PROYECTO.

Anexo A: Manual de usuario

Anexo B: Manual Técnico

Anexo C: Entregables de RUP

C1: Documento de Visión.

C2: Descripción de las tablas y campos de Base de Datos.