



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

**TRABAJO DE GRADO, PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS COMPUTACIONALES**

TEMA:

**“SISTEMA DE GESTIÓN DE INFORMACIÓN INSTITUCIONAL PARA EL
GOBIERNO PROVINCIAL DE IMBABURA.”**

APLICATIVO:

**IMPLEMENTACIÓN E IMPLANTACIÓN DEL MÓDULO WEB DE
GESTIÓN DE CONTRATOS (GPI-CONTRATOS).**

AUTOR: ANÍBAL ALEXANDER BENALCÁZAR CHULDE

DIRECTOR: ING. MARCO PUSDÁ

IBARRA – ECUADOR

2013



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 de Repositorio Digital Institucional, determina la necesidad de disponer de textos completos en formato 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 pongo a disposición la siguiente información:

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1002822854
APELLIDOS Y NOMBRES:	Benalcázar Chulde Aníbal Alexander
DIRECCIÓN:	Otavalo Cda. Los Lagos
EMAIL:	anibal_benalcazar@yahoo.es
TELÉFONO MÓVIL:	0986113485

DATOS DE LA OBRA	
TÍTULO:	SISTEMA DE GESTIÓN DE INFORMACIÓN INSTITUCIONAL PARA EL GOBIERNO PROVINCIAL DE IMBABURA.
AUTOR:	ANÍBAL ALEXANDER BENALCÁZAR CHULDE
FECHA:AAAAMMDD	2013-07-25
SÓLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSTGRADO
TÍTULO POR EL QUE OPTA:	INGENIERÍA EN SISTEMAS COMPUTACIONALES
ASESOR/DIRECTOR:	ING. MARCO PUSDA

2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Aníbal Alexander Benalcázar Chulde, con cédula de identidad Nro. 1002822854, en calidad de autor y titular de los derechos patrimoniales del trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación del trabajo en el Repositorio Digital Institucional y 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 144.



(Firma):.....

Nombre: Aníbal Alexander Benalcázar Chulde

Cédula: 1002822854

Ibarra, Diciembre de 2013



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

**CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA
UNIVERSIDAD TÉCNICA DEL NORTE**

Yo, Aníbal Alexander Benalcázar Chulde con cédula de identidad Nro. 1002822854, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5, 6, en calidad de autor del trabajo de grado denominado **“SISTEMA DE GESTIÓN DE INFORMACIÓN INSTITUCIONAL PARA EL GOBIERNO PROVINCIAL DE IMBABURA”**, que ha sido desarrollado para optar por el título de Ingeniero en Sistemas Computacionales, en la Universidad Técnica del Norte, 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 citada. 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.

(Firma):.....

Nombre: Aníbal Alexander Benalcázar Chulde

Cédula: 1002822854

Ibarra, Diciembre de 2013



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

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 tanto la obra es original y que es el titular de los derechos 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, Diciembre de 2013

EL AUTOR:

(Firma):.....

Nombre: Aníbal Alexander Benalcázar Chulde

Cédula: 1002822854



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CERTIFICACIÓN DEL ASESOR

El señor egresado Aníbal Alexander Benalcázar Chulde ha trabajado en el desarrollo del proyecto de tesis “**Sistema de Gestión de Contratos para el Gobierno Provincial de Imbabura**”, previo a la obtención del Título de Ingeniero en Sistemas Computacionales, realizándola con interés profesional y responsabilidad, lo cual certifico en honor a la verdad.

A handwritten signature in blue ink, appearing to be "Marco Pusdá", is written over a horizontal line.

Ing. Marco Pusdá
DIRECTOR DE TESIS

Ibarra, 13 de Diciembre del 2013

Ing. Javier Chiza, Director de Tecnologías de la información del Gobierno Provincial de Imbabura.

CERTIFICA:

Que, siendo auspiciantes del proyecto de tesis del Egresado **ANIBAL ALEXANDER BENALCÁZAR CHULDE** con CI: 100282285-4 quien desarrolló su trabajo con el tema "**SISTEMA DE GESTIÓN DE INFORMACIÓN INSTITUCIONAL PARA EL GOBIERNO PROVINCIAL DE IMBABURA (GPI-GESTIÓN)**" con el aplicativo "**IMPLEMENTACIÓN E IMPLANTACIÓN DEL MODULO WEB DE GESTIÓN DE CONTRATOS (GPI-CONTRATOS)**", me es grato informar que se ha superado a satisfacción las pruebas técnicas y la revisión del cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado por parte del egresado: **ANIBAL ALEXANDER BENALCAZAR CHULDE**.

El egresado **ANIBAL ALEXANDER BENALCÁZAR CHULDE** puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.

Atentamente,



Ing. Javier Chiza

DIRECTOR DE TECNOLOGÍA DE LA INFORMACIÓN





UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

DEDICATORIA

A mis padres Marco Aníbal Benalcázar y Nancy Chulde por haberme brindado oportunidades valiosas de superación en mi vida, por haber hecho de cada buen momento, un logro compartido.

A mi hija Arihana, por ser ella mi inspiración y a la vez mis ganas de seguir adelante superándome profesionalmente.

A mis esposa Jhoana, por haber estado ahí siempre con su apoyo incondicional e impulsándome a alcanzar esta meta, que se ha sacrificado junto mí y ha sido mi soporte para no darme por vencido.

A mi Hermano Edwin Benalcázar, por apoyarme en momentos difíciles y estar conmigo en las buenas y malas.



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

AGRADECIMIENTO

A Dios, por ser él quien me brinda la posibilidad de vivir y estar aquí presente ante la culminación de este proyecto.

A mis padres Marco Aníbal y Nancy, por haber apoyado en los buenos y malos momentos de mi vida siempre me han apoyado en mis estudios siendo un eje principal para poder lograr mis metas.

Mi agradecimiento en especial al Ing. Marcelo Jingo, Coordinador del proyecto en el Gobierno Provincial de Imbabura, pues su apoyo resultó fundamental en el desarrollo del proyecto, además de demostrar ser un amigo y un compañero de trabajo.

Un agradecimiento al Ing. Roberto López, Coordinador DBA del proyecto por su apoyo vital en cuanto a la parte técnica.

A mis maestros, que me impartieron sus conocimientos a lo largo de mi vida estudiantil.

ÍNDICE DE CONTENIDO

CERTIFICACIÓN DEL ASESOR	vi
ÍNDICE DE CONTENIDO	x
ÍNDICE DE ILUSTRACIONES	xiv
ÍNDICE DE TABLAS	xvi
RESUMEN.....	xvii
SUMMARY	xviii
CAPÍTULO I.....	1
1.- INTRODUCCIÓN	1
1.1.- ANTECEDENTES.....	1
1.2.- DEFINICIÓN DEL PROBLEMA.....	1
1.3.- OBJETIVOS DEL PROYECTO	2
1.3.1.- OBJETIVO GENERAL	2
1.3.2.- OBJETIVOS ESPECÍFICOS.....	2
1.4.- JUSTIFICACIÓN.....	3
1.5.- ALCANCE	3
.....	4
CAPÍTULO II.....	5
2.- MARCO TEÓRICO	5
2.1.- RESEÑA HISTÓRICA DEL GOBIERNO PROVINCIAL DE IMBABURA	5
2.1.1.- LA VISIÓN	5
2.1.2.- LA MISIÓN.....	5
2.1.3.- LOS OBJETIVOS ESTRATÉGICOS	5
2.1.4.- ORGANIGRAMA ESTRUCTURAL DEL GPI	6
2.1.5.- ELEMENTOS CONCEPTUALES DEL GPI PARA EL MODELADO DEL SISTEMA	7
2.2.- SCRUM METODOLOGÍA ÁGIL	8
2.2.1.- LA ESENCIA DE SCRUM	9
2.2.2.- FASES O PROCESOS DE SCRUM.....	9
2.2.3.- ELEMENTOS DEL SCRUM.....	11
2.3.- METODOLOGÍAS ÁGILES VS METODOLOGÍAS TRADICIONALES	19
2.4.- LENGUAJE DE MODELADO UNIFICADO (UML)	20
2.4.1.- DIAGRAMAS DE CASOS DE USO	21

2.5.-	ARQUITECTURA DEL SISTEMA.....	22
2.5.1.-	OBJETIVOS ARQUITECTÓNICOS.....	22
2.5.2.-	RESTRICCIONES.....	23
2.5.3.-	ARQUITECTURA FLEX ZEND.....	23
2.5.4.-	DESCRIPCIONES DEL SISTEMA.....	25
2.6.-	HERRAMIENTAS DE DESARROLLO.....	31
2.6.1.-	PHP.....	31
2.6.2.-	FLEX.....	32
2.6.3.-	ACTIONSCRIPT.....	33
2.6.4.-	AS3.....	33
2.6.5.-	MXML.....	35
2.6.6.-	BASE DE DATOS POSTGRESQL.....	35
2.6.7.-	Servidor Web APACHE.....	40
2.6.8.-	IDE FLASH BUILDER 4.5 FOR PHP.....	41
CAPÍTULO III	42
3. PLANIFICACIÓN DE LA ITERACIÓN (FASE DE INICIO)	42
3.1.-	ANÁLISIS Y DEFINICIÓN DE REQUERIMIENTOS.....	42
3.1.1.-	RELATO DEL USUARIO.....	42
3.1.2.-	FORMATO DE LA HISTORIA DE USUARIO.....	45
3.1.3.-	HISTORIAS DE USUARIOS.....	46
3.1.4.-	ACTIVIDADES DE INGENIERÍA.....	52
3.2.-	VISIÓN DEL PRODUCTO.....	52
3.2.1.-	INTRODUCCIÓN.....	52
3.2.2.-	POSICIONAMIENTO.....	52
3.2.3.-	DESCRIPCIÓN DE LOS INTERESADOS Y USUARIOS.....	54
3.2.4.-	VISTA GENERAL DEL PRODUCTO.....	57
3.2.5.-	DESCRIPCIÓN GLOBAL DEL PRODUCTO.....	61
3.2.6.-	RANGOS DE CALIDAD.....	61
3.2.7.-	OTROS REQUERIMIENTOS DEL SISTEMA.....	61
3.3.-	METODOLOGÍA DE TRABAJO INTERNO.....	62
3.3.1.-	INTRODUCCIÓN.....	62
3.3.2.-	VISTA GENERAL DEL PROYECTO.....	63
3.3.3.-	ORGANIZACIÓN DEL PROYECTO.....	67
3.3.4.-	PROCESO DE TRABAJO.....	67

3.4.-	PRODUCT BACKLOG.....	68
CAPÍTULO IV.....		69
4.	EJECUCIÓN DE LA ITERACIÓN	69
4.1.-	PLANIFICACIÓN DE LOS SPRINTS	69
4.2.-	MODELO DE CASOS DE USO.....	79
4.2.1.-	ACTORES.....	79
4.2.2.-	CASO DE USO ADMINISTRACIÓN DEL SISTEMA	80
4.2.3.-	CASO DE USO INGRESO DE CONTRATOS.....	81
4.2.4.-	CASO DE USO GARANTÍAS y PAGOS DE CONTRATOS.....	81
4.2.5.-	CASO DE USO FISCALIZACIÓN DE CONTRATO.....	82
4.2.6.-	CASO DE USO REPORTES DEL SISTEMA.....	82
4.3.-	ESPECIFICACIÓN DE LOS CASOS DE USO	83
4.3.1.-	ESPECIFICACIÓN DEL CASO DE USO: ACCEDER AL SISTEMA	83
4.3.2.-	ESPECIFICACIÓN DEL CASO DE USO: REGISTRAR OBJETOS DE CONTRATACIÓN.....	88
4.3.3.-	ESPECIFICACIÓN DEL CASO DE USO: REGISTRAR TIPOS DE CONTRATOS.....	91
4.3.4.-	ESPECIFICACIÓN DEL CASO DE USO: REGISTRAR PLANTILLAS DE CONTRATOS.....	95
4.3.5.-	ESPECIFICACIÓN DEL CASO DE USO: CONSULTA OBJETO DE CONTRATACIÓN.....	100
4.3.6.-	ESPECIFICACIÓN DEL CASO DE USO: CONSULTA TIPOS DE CONTRATOS.....	102
4.3.7.-	ESPECIFICACIÓN DEL CASO DE USO: INGRESO DATOS DE CONTRATOS.....	103
4.3.8.-	ESPECIFICACIÓN DEL CASO DE USO: INGRESO DE GARANTÍAS	108
4.3.9.-	ESPECIFICACIÓN DEL CASO DE USO: PAGOS DE CONTRATOS	110
4.3.10.-	ESPECIFICACIÓN DEL CASO DE USO: FISCALIZAR CONTRATOS .	111
4.3.11.-	ESPECIFICACIÓN DEL CASO DE USO: GENERAR REPORTES.....	113
4.4.-	MODELO ENTIDAD-RELACIÓN	115
4.5.-	MODELO FÍSICO	116
CAPÍTULO V.....		117
5.	INSPECCIÓN Y ADAPTACIÓN (IMPLEMENTACIÓN Y PRUEBAS).....	117
5.1.-	IMPLEMENTACIÓN DEL SISTEMA.....	117
5.1.1.-	PRUEBAS.....	117
5.1.2.-	INSTALACIÓN DEL SISTEMA.....	118

5.1.3.- CAPACITACIÓN AL PERSONAL DEL GPI	118
CAPÍTULO VI.....	120
6. CONCLUSIONES Y RECOMENDACIONES.....	120
6.1.- CONCLUSIONES.....	120
6.2.- RECOMENDACIONES.....	121
CAPÍTULO VII.....	122
7. GLOSARIO DE TÉRMINOS.....	122
7.1.- ABREVIATURA.....	122
7.2.- DEFINICIONES	122
CAPÍTULO VIII.....	124
8. BIBLIOGRAFÍA	124
CAPÍTULO IX.....	127
9. ANEXOS	127
ANEXO 1: ANTEPROYECTO DE TESIS.....	127
ANEXO 2: MANUAL DE INSTALACIÓN	127
ANEXO 3: MANUAL DE USUARIO.....	127
ANEXO 4: DICCIONARIO DE DATOS	127

ÍNDICE DE ILUSTRACIONES

Ilustración 1 : Módulo GPI-CONTRATOS.....	4
Ilustración 2: Organigrama Estructural GPI	6
Ilustración 3: Proceso Scrum.....	9
Ilustración 4: Elementos SCRUM	12
Ilustración 5: Sprint.....	15
Ilustración 6: Builds Continuos	18
Ilustración 7: Arquitectura de la Aplicación	22
Ilustración 8: Arquitectura Flex Zend	24
Ilustración 9: Flujo de Datos ZENDAMF	25
Ilustración 10: Estructura del patrón MVC	26
Ilustración 11: Diseño de Capas.....	28
Ilustración 12: Capa Interfaz de Usuario.....	29
Ilustración 13: Flex Enterprise Services.....	30
Ilustración 14: Arquitectura PostgreSQL.....	37
Ilustración 15: Adobe Flash Builder for PHP.....	41
Ilustración 16: MODELO GPI-GESTIÓN	58
Ilustración 17: Formato del Product Backlog	65
Ilustración 18: Caso de uso Administración del Sistema.....	80
Ilustración 19: Caso de Uso Ingreso de Contratos.....	81
Ilustración 20: Caso de Uso Ingreso de Garantías y Pagos de Contratos	81
Ilustración 21: Caso de Uso Fiscalizar Contrato	82
Ilustración 22: Caso de Uso Generar Reportes	82
Ilustración 23: Dirección de acceso al sistema	83
Ilustración 24: Pantalla de ingreso al sistema GPI-GESTIÓN.....	84
Ilustración 25: Pantalla Principal GPI-CONTRATOS	84
Ilustración 26: Panel Cabecera GPI-CONTRATOS	85
Ilustración 27: Menú Principal GPI-CONTRATOS	86
Ilustración 28: Mensaje de error inicio del sistema	87
Ilustración 29: Pantalla ingreso objeto de contratación	88
Ilustración 30: Mensaje Seleccionar padre objeto de contratación	89
Ilustración 31: Pantalla ingresar nuevo objeto de contratación	89
Ilustración 32: Pantalla cancelar ingreso objeto contratación	90
Ilustración 33: Mensaje Objeto Contratación es requerido.....	90
Ilustración 34: Pantalla Ingreso Tipos de Contratos.....	91
Ilustración 35: Pantalla seleccionar objeto de contratación.....	92
Ilustración 36: Pantalla tipos de contratos	92
Ilustración 37: Pantalla ingresar tipo de contrato	93
Ilustración 38: Pantalla cancelar ingreso tipo de contrato.....	94
Ilustración 39: Error ingreso de información tipos de contratos.....	94
Ilustración 40: Pantalla ingreso plantilla de contrato	95
Ilustración 41: Pantalla seleccionar objeto de contratación.....	96
Ilustración 42: Pantalla seleccionar tipo de contrato	96
Ilustración 43: Barra de herramientas ingreso plantilla de contrato.....	97
Ilustración 44: Pantalla agregar registro plantilla de contrato.....	97

Ilustración 45: Pantalla agregar clausula a plantilla de contrato	98
Ilustración 46: Pantalla actualizar clausula plantilla contrato	98
Ilustración 47: Pantalla cancelar ingreso clausula plantilla contrato	99
Ilustración 48: Error e el ingreso de información de cláusulas.....	99
Ilustración 49: Pantalla consultar objeto de contratación.....	101
Ilustración 50: Pantalla consultar tipo de contrato	102
Ilustración 51: Pantalla Ingreso datos de contratos	103
Ilustración 52: Pantalla ingresar etiquetas de información de contratos	104
Ilustración 53: Seleccionar guardar datos de contratos.....	105
Ilustración 54: Seleccionar visualizar documento de contrato	105
Ilustración 55: Guardar documento de contrato	106
Ilustración 56: Visualizar documento de contrato	106
Ilustración 57: Mensaje Campo Requerido	107
Ilustración 58: Mensaje Insertar Todas las etiquetas.....	107
Ilustración 59:Pantalla de Ingreso de Contratos	108
Ilustración 60: Pantalla Pagos de Contratos	110
Ilustración 61: Pantalla Fiscalización de Contratos	111
Ilustración 62: Pantalla Reportes	113
Ilustración 63: Modelo Entidad Relación	115
Ilustración 64: Modelo Físico	116

ÍNDICE DE TABLAS

Tabla 1: Roles de Scrum	13
Tabla 2: Límites de PostgreSQL.....	40
Tabla 3: Formato Historias de Usuarios	46
Tabla 4: Historia de Usuario 1	46
Tabla 5: Historia de Usuario 2	47
Tabla 6: Historia de Usuario 3	47
Tabla 7: Historia de Usuario 4	48
Tabla 8: Historia de Usuario 5	49
Tabla 9: Historia de Usuario 6	49
Tabla 10: Historia de Usuario 7	50
Tabla 11: Historia de Usuario 8	51
Tabla 12: Historia de Usuario 9	51
Tabla 13: Definición del Problema.....	53
Tabla 14: Posicionamiento del proyecto	54
Tabla 15: Resumen de los interesados	56
Tabla 16: Resumen de los usuarios	57
Tabla 17: Necesidades y características	60
Tabla 18: Formato Sprint Backlog	67
Tabla 19: Product Backlog GPI-CONTRATOS	68
Tabla 20: SPRINT 0	69
Tabla 21: SPRINT 1	70
Tabla 22: SPRINT 2	71
Tabla 23: SPRINT 3	72
Tabla 24: SPRINT 4	73
Tabla 25: SPRINT 5	74
Tabla 26: SPRINT 6	75
Tabla 27: SPRINT 7	76
Tabla 28: SPRINT 8	77
Tabla 29: SPRINT 9	78
Tabla 30: Actores	80
Tabla 31: Módulos de Carga en Menú Principal	86
Tabla 32: Especificación de Caso de Uso Acceder al Sistema	87
Tabla 33: Especificación de Caso Registrar Objetos de contratación	90
Tabla 34 Especificación de casos de uso Registrar tipos de contratos.....	95
Tabla 35: Especificación de caso de uso Registrar Plantilla	100
Tabla 36: Especificación de caso de uso Consulta objeto de contratación	101
Tabla 37: Especificación caso de uso Consulta Tipos de contratos.....	103
Tabla 38: Especificación de caso de uso Ingreso datos de contratos	108
Tabla 39: Especificación de casos de uso Ingreso de Garantías	109
Tabla 40: Especificación de caso e uso Pago de contratos	111
Tabla 41: Especificación caso de uso Fiscalizar contratos	112
Tabla 42: Especificación de caso de uso Generar reportes.....	114
Tabla 43: Abreviaturas	122
Tabla 44: Definiciones.....	123

RESUMEN

Actualmente en el medio de desarrollo de sistemas informáticos ha constituido un gran avance para la optimización de procesos. El uso de nuevas tecnologías y de la web ha generado un cambio importante y significativo para el sector público y privado en el Ecuador. En el Gobierno Provincial de Imbabura se coordina, planifica y ejecuta obras y convenios, los cuales son realizados en base a contratos que deben elaborarse de acuerdo a las leyes y normas vigentes. Del mismo modo, dentro de su planta administrativa y de operación, el GPI mantiene contratos de talento humano y de servicios que permiten cumplir con las funciones inherentes a la gestión pública, debido a esto el GPI adopto la necesidad de generar el proyecto de gestión de contratos; dicho proyecto ayudara a los diferentes departamentos implicados en la gestión de contratos. El propósito principal de este trabajo de grado fue que el departamento jurídico y talento humano, disponga de una herramienta informática que genere los contratos a partir de una plantilla ya ingresada en el sistema. Para ello se desarrolló el sistema GPI-CONTRATOS que es parte de la suite de aplicaciones GPI-GESTION del Gobierno Provincial de Imbabura. El sistema optimiza la gestión de contratos que se generan dentro del GPI, permite ingresar los tipos de contratos, plantillas de contratos, ingresar datos de los contratos para q se generen en un pdf, también podrá controlar las garantías, los pagos y fiscalizar los contratos de Obras que se generan el GPI.

SUMMARY

Currently in the middle of development of computer systems has been a breakthrough for process optimization. The use of new web technologies and has produced a significant change important and public and private sector in Ecuador. Gobierno Provincial de Imbabura is coordinated, planned and executed works and agreements, which are made on the basis of contracts, should be tailored according to the laws and regulations. Similarly, in its administrative and operating plant, the GPI has contracts for human talent and services that enable performing the functions inherent in public administration, because of this the GPI saw the need to build this project management contracts, the project will help the different departments involved in contract management. The main purpose of this paper grade was that the legal department and human talent, provides a tool that generates contracts from a template and entered in the system. This system was developed GPI-CONTRATOS is part of the suite of applications GPI-GESTION Imbabura Provincial Government. The system optimizes the management of contracts that are generated within the GPI, can enter the types of contracts, contract templates, contract data entry to q are generated in a PDF, you can also control the guarantees, payments and monitor contracts Works that are generated.

CAPÍTULO I

1.- INTRODUCCIÓN

Hoy en día el uso de sistemas de información para la administración, procesamiento y distribución de la información en una organización, se hace cada vez más indispensable.

Estos sistemas permiten lograr ahorros significativos en tiempo y en mano de obra, debido a que automatizan tareas operativas de la organización y ofrecen un gran apoyo en el proceso de toma de decisiones que permiten, entre otras cosas, lograr ventajas competitivas en el momento de la implantación y el uso de la información.

Este proyecto se presenta el diseño, implementación e implantación de un Sistema de Gestión de Contratos (GPI-CONTRATOS¹) para el Gobierno provincial de Imbabura (GPI²); que comprenderá en el control de los contratos generados en esta Dependencia.

1.1.- ANTECEDENTES

El Gobierno provincial de Imbabura es una Institución de derecho público, que goza de autonomía política, administrativa y financiera, representa a la provincia y tiene capacidad de realizar actos que fueren necesarios para el cumplimiento de sus fines, en la forma y condiciones que determinan la Constitución y las Leyes.

1.2.- DEFINICIÓN DEL PROBLEMA

Dentro de sus competencias, el GPI coordina, planifica y ejecuta obras y convenios, los cuales son realizados en base a contratos que deben elaborarse de acuerdo a las leyes y normas vigentes. Del mismo modo, dentro de su planta administrativa y de operación, el GPI mantiene contratos de talento humano y de servicios que permiten cumplir con las funciones inherentes a la gestión pública.

Sobre lo expuesto, el GPI requiere manejar información para el estudio, evaluación y control de contratos, los diferentes departamentos involucrados en el proceso de contratación, enfrentan serias dificultades en el acceso ágil y oportuno a la información de los diferentes contratos.

¹GPI-CONTRATOS.- Sistema de Gestión de Contratos

² GPI.- Gobierno Provincial de Imbabura

Las causas que dificultan el manejo adecuado de los contratos son:

- Información manejada en hojas Excel.
- Duplicidad de datos
- Falta de integridad
- Carencia de acceso rápido a Información

Por lo expuesto,

- Se incrementan la posibilidad de generar información errónea debido a dispersión y duplicidad de datos en los diferentes departamentos.
- Implica demasiadas actividades manuales que pueden provocar errores.
- El nivel operativo no dispone de información sobre contratos
- Se dificulta la entrega de información centralizada al nivel directivo lo que dificulta la entrega de datos veraces y únicos.
- Los diferentes departamentos implicados en la gestión de contratos no tienen acceso automatizado a la información de forma rápida y eficaz.

1.3.- OBJETIVOS DEL PROYECTO

1.3.1.- OBJETIVO GENERAL

Desarrollar e implantar un sistema Informático web de Gestión de Contratos para el “**Gobierno Provincial de Imbabura**” basado en el uso de herramientas de software libre para automatizar el proceso contractual y a la vez evitar pérdida de datos, redundancia de registros y generar información veraz y oportuna.

1.3.2.- OBJETIVOS ESPECÍFICOS

- Analizar y diagnosticar los procesos contractuales que maneja el GPI.
- Realizar el diseño minucioso de la arquitectura y funcionamiento de la solución planteada.
- Desarrollar, validar e implantar el sistema web utilizando herramientas de software libre como Apache, PostgreSQL y PHP.
- Capacitar a los usuarios finales del sistema de Gestión de Contratos.
- Elaborar la documentación pertinente relacionada con el uso y administración del sistema.

1.4.- JUSTIFICACIÓN

En la actualidad la decisión de adoptar sistemas de gestión de procesos en las empresas se ha tornado un tema prioritario ya que constituye un gran aporte en obtener mejor tiempo de respuesta en el servicio al cliente que luego se traduce en beneficios para la propia empresa.

El sistema de gestión de contratos es parte de una suite de aplicaciones denominada GPI-GESTIÓN³ que consta de los siguientes módulos:

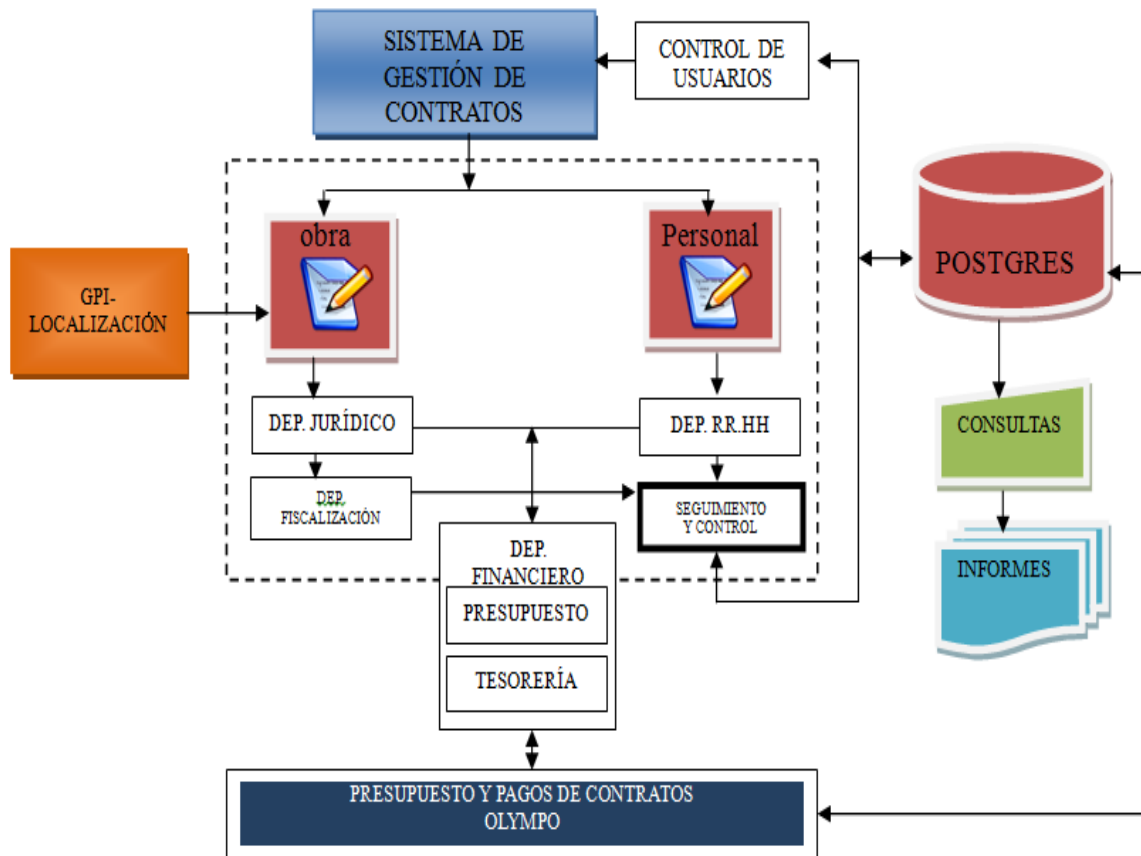
1. GPI-SISTEMA
2. GPI-ORGANIZACIÓN
3. GPI-PLANIFICACIÓN
4. GPI-PROYECTOS
5. GPI-PRESUPUESTO
6. GPI-COMPRAS PÚBLICAS
7. **GPI-CONTRATOS**
8. GPI-LOCALIZACIÓN
9. GPI-TALENTO HUMANO
10. GPI-AUDITORÍA

1.5.- ALCANCE

El alcance de este proyecto es diseñar, construir e implantar en una intranet un sistema de información web que ofrezca apoyo en la gestión de los diferentes contratos generado en el GPI.

La finalidad del proyecto es permitir acceso rápido a la información y mantener el control de los procesos contractuales en forma rápida y eficaz, consta de los siguientes módulos.

³ GPI-GESTION.- Suite de Aplicaciones del GPI



Fuente: Propia
 Ilustración 1 : Módulo GPI-CONTRATOS

CAPÍTULO II

2.- MARCO TEÓRICO

2.1.- RESEÑA HISTÓRICA DEL GOBIERNO PROVINCIAL DE IMBABURA

El Gobierno Provincial de Imbabura se consolida como una institución de derecho público, autónoma, descentralizada, transparente, eficiente, equitativa, incluyente y solidaria; líder del desarrollo económico, social y ambiental provincial.

2.1.1.- LA VISIÓN

El Gobierno Provincial de Imbabura se consolida como una institución de derecho público, autónoma, descentralizada, transparente, eficiente, equitativa, incluyente y solidaria; líder del desarrollo económico, social y ambiental provincial.

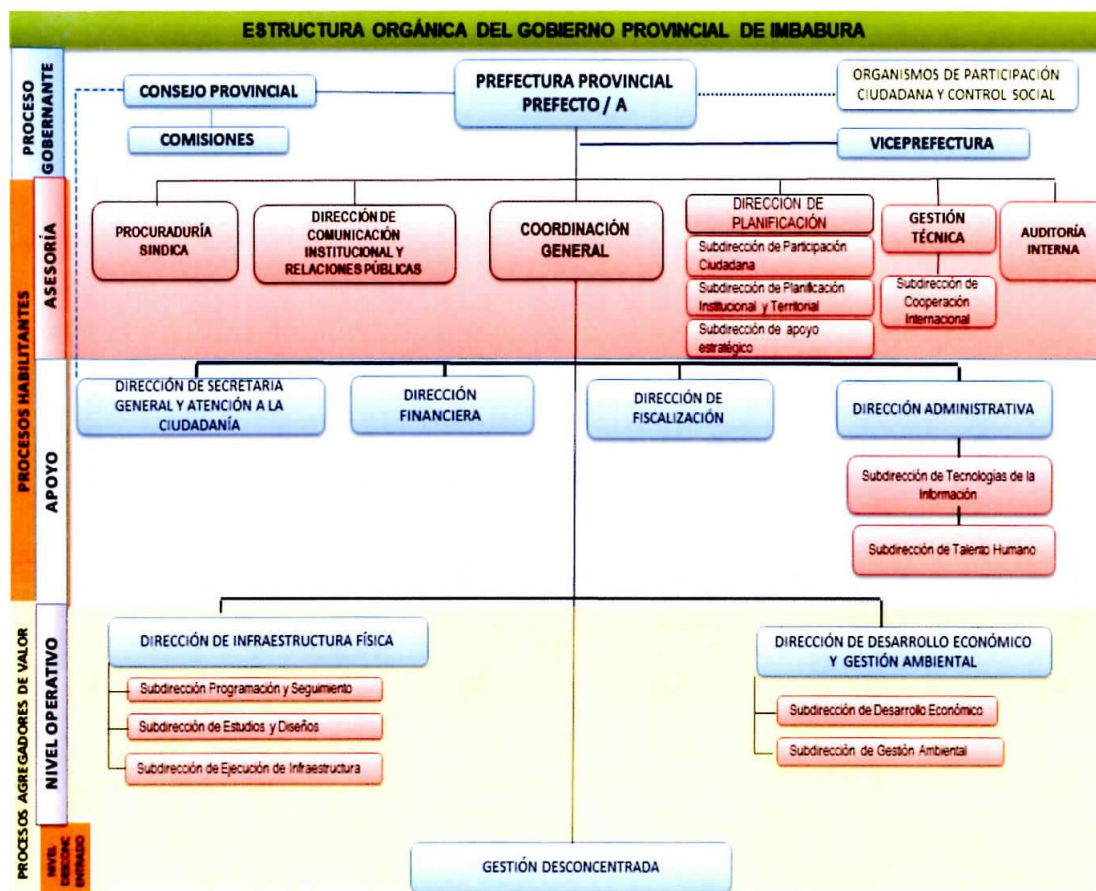
2.1.2.- LA MISIÓN

El Gobierno Provincial de Imbabura es la institución encargada de coordinar, planificar, ejecutar y evaluar el Plan de Desarrollo Provincial Participativo; fortaleciendo la productividad, la viabilidad, el manejo adecuado de sus recursos naturales y promoviendo la participación ciudadana; a fin de mejorar la calidad de vida de sus habitantes.

2.1.3.- LOS OBJETIVOS ESTRATÉGICOS

- Fomentar el desarrollo económico provincial
- Consolidar el sistema de transporte y movilidad provincial
- Implementar el sistema de gestión ambiental provincial con enfoque intercultural y visión de cuenca hidrográfica
- Diseñar políticas, planes y programas, tendientes a fortalecer la inclusión social, el desarrollo cultural que permitan hacer de Imbabura una provincia equitativa, solidaria e intercultural.
- Generar mecanismos de articulación y lineamientos para la coordinación endógena institucional e interinstitucional.
- Tecnificar los procesos de administración y gestión institucional.

2.1.4.- ORGANIGRAMA ESTRUCTURAL DEL GPI



Fuente: www.imbabura.gob.ec
 Ilustración 2: Organigrama Estructural GPI

2.1.5.- ELEMENTOS CONCEPTUALES DEL GPI PARA EL MODELADO DEL SISTEMA

2.1.5.1. PERSONA NATURAL

Persona Natural es un concepto netamente jurídico, se le denomina al hombre o mujer mayor de 18 años, sujeta de derechos y obligaciones.⁴

2.1.5.2. PERSONA JURÍDICA

Se entiende por persona jurídica a las entidades que, para la realización de determinados fines colectivos, las normas jurídicas les reconocen capacidad para obligarse y disfrutar de derechos. Actúan como sujetos de derecho, esto es, capacidad para adquirir y poseer bienes de toda clase, para contraer obligaciones y ejercitar acciones civiles o criminales. Las personas jurídicas nacen como consecuencia de un acto jurídico.⁵

2.1.5.3. CONTRATISTA

Toda persona natural o jurídica que ejecuta una obra, suministra bienes.

2.1.5.4. CONTRATO

Instrumento Jurídico que regula, la ejecución de una obra.

2.1.5.5. ADMINISTRADOR DE CONTRATO

El administrador de contrato debe ser parte de todo el proceso contractual, desde la planificación y organización del cierre del contrato.

2.1.5.6. FISCALIZADOR DE CONTRATO

Persona designada por la máxima autoridad para realizar la inspección y vigilancia del cumplimiento de las obras contratadas.

2.1.5.7. LICITACIÓN DE CONTRATO

Es el procedimiento competitivo de selección del contratista, en el que pueden participar personas naturales y jurídicas, previo cumplimiento de los requisitos establecidos en la Ley.

⁴ ESPOL, W. (s.f.). *Persona Natural*. Recuperado, de http://www.wiki.espol.edu.ec/index.php/Personas_naturales

⁵ Wikipedia. (s.f.). *Persona Jurídica*. Recuperado, de http://es.wikipedia.org/wiki/Persona_jurídica

2.1.5.8. SOLICITUD U ORDEN DE SERVICIO

Es la petición por escrito que realizan las autoridades de las dependencias dentro del GPI.

2.1.5.9. OBRA

Es la actividad que se realiza en una dependencia la cual es originada por una solicitud de servicio.

2.2.- SCRUM METODOLOGÍA ÁGIL

SCRUM es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.⁶

En SCRUM se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, SCRUM está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

SCRUM también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

SCRUM es un proceso para la gestión y control del producto que trata de eliminar la complejidad en estas áreas para centrarse en la construcción de software que satisfaga las necesidades del negocio. Es simple y escalable, ya que no establece prácticas de ingeniería del software sino que se aplica o combina, fácilmente, con otras prácticas ingenieriles, metodologías de desarrollo o estándares ya existentes en la organización.

SCRUM se concentra, principalmente, a nivel de las personas y equipo de desarrollo que construye el producto. Su objetivo es que los miembros del equipo trabajen juntos y de forma eficiente obteniendo productos complejos y sofisticados. Pretende que no haya problemas ocultos, asuntos u obstáculos que puedan poner en peligro el proyecto. Los

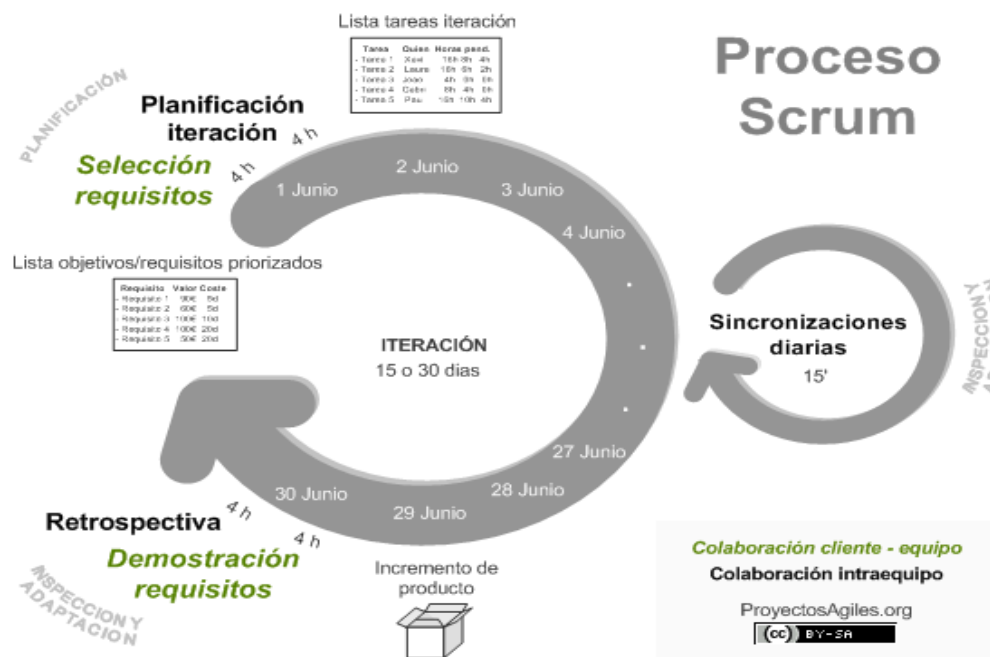
⁶ Xavier Albaladejo, Certified Scrum Master. (s.f.). *Proyectos Agiles*. Obtenido de *Proyectos Agiles*: <http://www.proyectosagiles.org>

equipos se guían por su conocimiento y experiencia más que por plan es de proyecto formalmente definido. La planificación detallada se realiza sobre cortos espacios de tiempo lo que permite una constante retroalimentación que proporciona inspecciones simples y un ciclo de vida adaptable. Así, el desarrollo de productos se produce de forma incremental y con un control empírico del proceso que permite la mejora continua.

2.2.1.- LA ESENCIA DE SCRUM⁷

- Más que una metodología de desarrollo, SCRUM es una herramienta para gestionar proyectos.
- Está basado en el hecho de que el trabajo es realizado por equipos auto-organizados y auto-dirigidos, logrando motivación, responsabilidad y compromiso
- Es un proceso constructivo, iterativo e incremental donde las iteraciones tienen una duración fija.
- Contiene definiciones de roles, prácticas y productos de trabajo escritos de forma simple y soportada en un conjunto de valores y principios (métodos ágiles).

2.2.2.- FASES O PROCESOS DE SCRUM⁸



Fuente: <http://www.proyectosagiles.org>
Ilustración 3: Proceso Scrum

⁷ Citón, M. L. (2006). *Metodología Ágil SCRUM*.

Obtenido de <http://www.slideshare.net/testlucero/metodo-agil-scrum>

⁸ Xavier Albaladejo, Certified Scrum Master. (s.f.). *Proyectos Ágiles*. Obtenido de Proyectos Ágiles: <http://www.proyectosagiles.org>

En SCRUM un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

2.2.2.1. PLANIFICACIÓN DE LA ITERACIÓN (SPRINT⁹)

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

2.2.2.1.1. SELECCIÓN DE REQUISITOS

(4 horas máximo). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

2.2.2.1.2. PLANIFICACIÓN DE LA ITERACIÓN (SPRINT)

(4 horas máximo). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se auto asignan las tareas.

2.2.2.2. EJECUCIÓN DE LA ITERACIÓN (SPRINT)

Cada día el equipo realiza una reunión de sincronización (15 minutos máximos). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización?
- ¿Qué voy a hacer a partir de este momento?
- ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el Facilitador se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.

⁹ Wikipedia. (s.f.). *SPRINT*. Recuperado el 24 de Junio de 2013, de <http://es.wikipedia.org/wiki/Scrum>

- Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

2.2.2.3. INSPECCIÓN Y ADAPTACIÓN

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

2.2.2.3.1. DEMOSTRACIÓN

(4 horas máximo). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.

2.2.2.3.2. RETROSPECTIVA

(4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

2.2.3.- ELEMENTOS DEL SCRUM

- Roles
 - Product Owner¹⁰ (propietario del producto)
 - SCRUM Master¹¹
 - Team (Equipo)¹²
- Poda de requerimientos
- Product Backlog¹³
- Sprint
 - Planificación
 - Sprint Backlog
 - SCRUM diario
 - Estimaciones

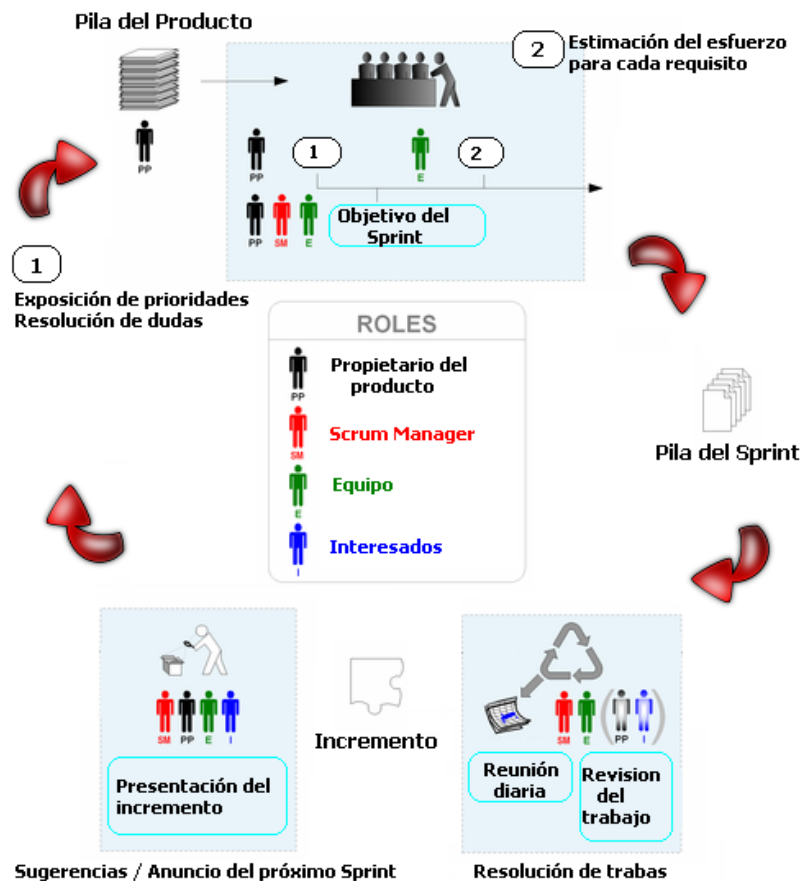
¹⁰ Product Owner.- Dueño del Producto. Recuperado de <http://es.wikipedia.org/wiki/Scrum>

¹¹ SCRUM Master.- Responsable del Proyecto. Recuperado de <http://es.wikipedia.org/wiki/Scrum>

¹² Team.- Equipo de programadores

¹³ Product Backlog.- Pila de producto. Recuperado de <http://es.wikipedia.org/wiki/Scrum>

- Builds Continuos
- Revisión del Sprint
- Reunión retrospectiva
- Valores
 - Foco, comunicación, respeto y coraje



Fuente: <http://santimacnet.wordpress.com/SCRUM>
 Ilustración 4: Elementos SCRUM

2.2.3.1. ROLES

En SCRUM se definen varios roles, estos están divididos en dos grupos: cerdos y gallinas. El nombre de los grupos está inspirado en el chiste sobre un cerdo y una gallina que se relata a continuación.

Un cerdo y una gallina se encuentran en la calle. La gallina mira al cerdo y dice: "Hey, ¿por qué no abrimos un restaurante?" El cerdo mira a la gallina y le dice: "Buena idea, ¿cómo se llamaría el restaurante?" La gallina piensa un poco y contesta: "¿Por qué no lo

llamamos "Huevos con jamón?" "Lo siento pero no", dice el cerdo, "Yo estaría comprometido pero tú solamente estarías involucrada".¹⁴

De esta forma, los cerdos están comprometidos a construir software de manera regular y frecuente, mientras que el resto son gallinas: interesados en el proyecto pero realmente irrelevantes porque, si éste falla, no son un cerdo, es decir, no son los que se habían comprometido a sacarlo adelante. Las necesidades, deseos, ideas e influencias de los roles gallina se tienen en cuenta, pero no de forma que pueda afectar, distorsionar o entorpecer el proyecto SCRUM.

ROLES CERDO Comprometidos con el proyecto	ROLES GALLINA Implicados en el proyecto
Dueño del producto	Jurídico
Equipo	Financiero
SCRUM Master	Fiscalización, usuarios, etc.

Fuente: Propia
Tabla 1: Roles de Scrum

2.2.3.1.1. ROLES CERDO

Los Cerdos son los que están comprometidos con el proyecto y el proceso SCRUM; ellos son los que "ponen el jamón en el plato".

PRODUCT OWNER (DUEÑO DEL PRODUCTO)

Representa a todos los interesados en el producto final. Es el responsable oficial del proyecto, gestión, control y visibilidad de la lista de acumulación o lista de retraso del producto (Product Backlog). Toma las decisiones finales de las tareas asignadas al registro y convierte sus elementos en rasgos a desarrollar.

Sus áreas de responsabilidad son:

- Financiación del proyecto
- Requisitos del sistema
- Retorno de la inversión del proyecto
- Lanzamiento del proyecto

¹⁴ Wikiscies. (pas.f.). *ROLES DE SRUM*. Recuperado el 26 de Junio de 2013, de <http://bi-trabajodegrado.wikispaces.com/Scrum>

SCRUM MASTER (FACILITADOR LÍDER DEL PROYECTO)

Responsable del proceso SCRUM, de cumplir la meta y resolver los problemas. Así como también, de asegurarse que el proyecto se lleve a cabo de acuerdo con las prácticas, valores y reglas de SCRUM y que progrese según lo previsto.

Interactúa con el cliente y el equipo. Coordina los encuentros diarios, y se encarga de eliminar eventuales obstáculos. Debe ser miembro del equipo y trabajar a la par.

TEAM (EQUIPO)

Responsable de transformar el Backlog de la iteración en un incremento de la funcionalidad del software. Tiene autoridad para reorganizarse y definir las acciones necesarias o sugerir remoción de impedimentos.

2.2.3.1.2. ROLES GALLINA

Los roles gallina en realidad no son parte del proceso SCRUM, pero deben tenerse en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados (stakeholders). Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

USUARIOS

Se refiere a la gente que va a utilizar el sistema.

STAKEHOLDERS¹⁵ (CLIENTES, PROVEEDORES)

Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirán el beneficio acordado que lo justifica. Sólo participan directamente durante las revisiones del sprint.

MANAGERS

Es la gente que establece el ambiente para el desarrollo del producto.

2.2.3.2. PODA DE REQUERIMIENTOS

La poda de requerimientos es una buena práctica implícita en modelos ágiles, y consiste en hacer lo que el cliente realmente desea, no más.

La primera actividad en un proyecto SCRUM es armar una lista exhaustiva de los requerimientos originales del sistema. Posteriormente se realiza una revisión para evaluar que requerimientos son realmente necesarios, cuales pueden posponerse y cuáles

¹⁵ Wikispaces. (s.f.). STAKEHOLDERS.- Beneficiados del producto. Recuperado de <http://bi-trabajodegrado.wikispaces.com/Scrum>

eliminarse. Para ello debe identificarse un representante con capacidad de decisión, priorizar los requerimientos en base a su importancia y acordar cuales son los prioritarios para la fecha de entrega.

2.2.3.3. PRODUCT BACKLOG

Con los requerimientos priorizados y podados se arma el Backlog del Producto. Este es una forma de registrar y organizar el trabajo pendiente para el producto (actividades y requerimientos).

Es un documento dinámico que incorpora constantemente las necesidades del sistema. Por lo tanto, nunca llega a ser una lista completa y definitiva. Se mantiene durante todo el ciclo de vida y es responsabilidad del Product Owner.

2.2.3.4. SPRINT

SCRUM está basado en el control empírico de procesos. Se utiliza cuando la capacidad de predicción es vaga, la incertidumbre alta o el proceso es demasiado complejo para ser modelado y definido.

En el enfoque empírico de control de procesos se establecen reglas simples y se crea una disciplina de inspección frecuente para adaptarse rápidamente a situaciones no previstas o problemas.

Un Sprint es el periodo de tiempo durante el que se desarrolla un incremento de funcionalidad. Constituye el núcleo de SCRUM, que divide de esta forma el desarrollo de un proyecto en un conjunto de pequeñas carreras.



Fuente: www.scrum.org
Ilustración 5: Sprint

Aspectos a tener en cuenta en un Sprint:

- Duración máxima del Sprint: 30 días, recomendada 15 días
- Durante el Sprint no se puede modificar el trabajo que se ha acordado en el Backlog.
- Solo es posible cambiar el curso de un Sprint, abortándolo, y solo lo puede hacer el SCRUM Master si decide que no es viable por alguna de las razones siguientes:
 - La tecnología acordada no funciona.
 - Las circunstancias del negocio han cambiado.
 - El equipo ha tenido interferencias.

2.2.3.4.1. PLANIFICACIÓN

Se planifica en detalle el trabajo al inicio de cada Sprint asumiendo que los objetivos no van a cambiar durante el mismo. De esta manera se atenúa el riesgo.

Aspectos a tener en cuenta sobre la planificación de un Sprint:

- Una determinación general de alcance, frecuentemente basada en una EDT¹⁶ (Estructura de División del Trabajo).
- Estimaciones de esfuerzo de alto nivel realizadas durante la etapa de concepción del proyecto.
- Esfuerzo dedicado a labores de soporte o de preparación de los ambientes requeridos por el proyecto.
- Esfuerzo asociado a las reuniones diarias de planificación y de revisión.
- Requerimientos de recursos de infraestructura o logísticos (pc's, redes, licencias, papel, pizarras, etc.).
- Habilidades presentes y necesarias en el equipo.
- Restricciones asociadas al conocimiento del negocio, la tecnología o externas (legales, reglamentarias, estándares).

El rol de SCRUM Master durante la planificación es:

- Dirigir la planificación.
- Facilitar acuerdos entre el equipo y el Product Owner del proyecto.
- Registrar problemas y riesgos detectados durante la planificación.
- Registrar las tareas, asignaciones y estimaciones.
- Iniciar el Backlog del Sprint.

¹⁶ EDT.- Estructura de división del trabajo

2.2.3.4.2. SPRINT BACKLOG

Trabajo o tareas determinadas por el equipo para realizar en un Sprint.

- Tareas a convertir en producto funcional.
- Las tareas se estiman en una duración entre 1 y 20 horas de trabajo. La de mayor duración deben intentar descomponerse en sub-tareas en el rango de tiempo indicado.
- La estimación como el avance de las tareas se actualiza día a día.

2.2.3.4.3. SCRUM DIARIO

SCRUM asume que el proceso es complejo y que es necesario inspeccionarlo frecuentemente, por eso se realiza una reunión diaria de seguimiento.

El foco de la reunión es determinar el avance en las tareas y detectar problemas que estén haciendo lento el progreso del equipo o que eventualmente impidan a un equipo cumplir con la meta del sprint. La idea es que ningún problema quede sin resolver o por lo menos sin iniciar alguna acción de respuesta dentro de las 24 horas después de su detección.

La reunión es además un espacio definido para que cada miembro del equipo comunique a los demás el estado de su trabajo.

El rol de SCRUM Master durante el SCRUM es:

- Dirigir la reunión y mantiene el foco
- Hacer preguntas para aclarar dudas.
- Registrar documentar los problemas para su resolución después de la reunión.
- Asegura que los miembros cuenten con el ambiente adecuado para la reunión.

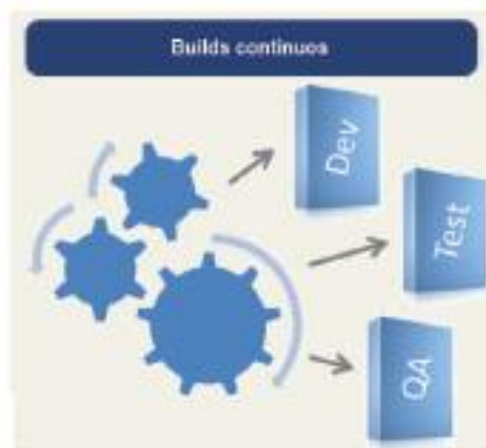
2.2.3.4.4. ESTIMACIONES

Las estimaciones se realizan por primera vez en la reunión de planificación al inicio del Sprint. Luego las tareas se re-estiman todos los días y se registran sus cambios en el Backlog del Sprint; esta actividad puede ser realizada inmediatamente antes o después del SCRUM diario

2.2.3.4.5. BUILDS CONTINUOS¹⁷

La estrategia que generalmente se utiliza es la de Builds continuos y prueba básica para la funcionalidad del sistema consiste en:

- Los programadores desarrollan el Backlog del Sprint, y al finalizar, notifican al integrador.
- El integrador toma el código y lo integra con el resto del producto.
- Se compila el software y se prueba superficialmente el producto, para verificar la estabilidad de la versión.
- Si se encuentran problemas se devuelve al desarrollador.
- Si no se han encontrado errores se notifica al equipo que hay una nueva versión “estable” del código a usar como base.



Fuente: <http://santimacnet.wordpress.com/Builds>
Ilustración 6: Builds Continuos

2.2.3.4.6. REVISIÓN DEL SPRINT

El objetivo de la reunión de revisión es presentar el producto o porción del producto desarrollada por el equipo a los usuarios. La reunión se utiliza para detectar inconformidades mayores que se vuelven elementos del Backlog de Producto y que eventualmente se resuelven en el siguiente SPRINT.

2.2.3.4.7. REUNIÓN RETROSPECTIVE

SCRUM involucra el concepto de mejora continua a través de las reuniones de retrospectión. Las reuniones buscan detectar los puntos positivos y negativos del Sprint para generar propuestas de mejora para futuros Sprints. Las reuniones de retrospectión

¹⁷ Builds continuos. Recuperado de <http://www.slideshare.net/testlucero/metodo-agil-scrum>

son el concentrador del aprendizaje organizacional sobre el SCRUM. Los puntos positivos y negativos se registran y se definen ítems para cada uno. Los ítems de acción se toman en cuenta en los siguientes Sprints. A esta reunión acude el SCRUM Master y opcionalmente el Product Owner.

2.2.3.5. VALORES

2.2.3.5.1. FOCO

Los individuos y sus interacciones son más importantes que el proceso o las herramientas. El recurso humano es el principal factor de éxito de un proyecto de software.

2.2.3.5.2. COMUNICACIÓN

SCRUM pone en comunicación directa y continua a clientes y desarrolladores. El cliente se integra en el equipo para establecer prioridades y resolver dudas. De esta forma ve el avance día a día y es posible ajustar a la agenda y las funcionalidades de forma consecuente.

2.2.3.5.3. RESPETO

SCRUM diferencia claramente entre dos grupos para garantizar que quienes tienen la responsabilidad tienen también la autoridad necesaria para poder lograr el éxito (cerdos), y que quienes no tienen la responsabilidad, los observadores externos (gallinas), no produzca interferencias innecesarias.

2.2.3.5.4. CORAJE

El coraje implica saber tomar decisiones difíciles. Reparar un error cuando se detecta. Mejorar el código siempre que el feedback y las sucesivas iteraciones se manifiesten susceptibles de mejora.

2.3.- METODOLOGÍAS ÁGILES VS METODOLOGÍAS TRADICIONALES

Simplificándolo mucho, se puede decir que las metodologías tradicionales (SSAD¹⁸, RUP, etc.) ponen por encima de lo demás los objetivos de la definición y del control del trabajo, mientras que las metodologías ágiles priman la libertad del equipo, la comunicación entre el cliente y el equipo, realizar sólo las tareas que aportan valor al cliente, y finalmente

¹⁸ SSAD.- Metodología de Desarrollo de software. Recuperado de <http://85517amdsi.blogspot.com/2010/08/metodologia-ssadm-y-metrica.html>

definir el trabajo tal y como éste se va realizando para el conocimiento adquirido durante su desarrollo evite realizar tareas innecesarias.

Una consideración importante es el tamaño de los equipos. La inmensa mayoría de los proyectos se realizan por empresas o equipos muy pequeños (de 1 a 10 personas) donde las metodologías tradicionales son difíciles de aplicar, e incluso imposibles si se quiere hacer de manera exhaustiva.

Otro aspecto influyente es el contexto y misión de los equipos, que pueden ser estables en su actividad o cambiar más o menos frecuentemente de miembros o proyecto.

Existen diversas creencias falsas que pueden perjudicar la adopción de una nueva metodología, sea del tipo que sea.

1. Las metodologías tradicionales son pesadas y que suponen obligatoriamente un “todo o nada”.
2. Las metodologías ágiles son más modernas y mejores que cualquiera de las tradicionales.
3. Las actividades “de calidad” son inútiles y sólo funcionan en equipos grandes, no se adaptan a nuestros proyectos. Cualquier cosa que nos quite tiempo de tareas técnicas (programar, etc.) es una pérdida de tiempo.
4. Trabajar con una metodología ágil no supone aplicar disciplina alguna, sino que es una manera fácil de dar estructura al equipo.

2.4.- LENGUAJE DE MODELADO UNIFICADO (UML)¹⁹

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booc, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co, para crear una notación unificada en la base de la construcción de sus herramientas CASE. En el proceso de creación UML han participado, no obstante, otras empresas de gran peso como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

UML es un lenguaje estándar que sirve para escribir los planos del software, puede utilizarse para visualizar, especificar, construir y documentar todos los artefactos que componen un sistema con gran cantidad de software. UML puede usarse para modelar

¹⁹ UML.- Lenguaje para modelar un software. Recuperado de https://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado

desde sistemas de información hasta aplicaciones distribuidas basadas en Web, basados en sistemas empotrados de tiempo real. UML es solo un lenguaje por lo que es solo una parte de un método de desarrollo software, es independiente del proceso para que sea óptimo debe usarse en un proceso dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental.

Este lenguaje describe nueve tipos de diagramas como son los casos de uso, clases, objetos, colaboración, secuencia, componentes, implementación, despliegue y estados de transición. A continuación se describen detalladamente los distintos diagramas.

2.4.1.- DIAGRAMAS DE CASOS DE USO

Los casos de uso representan los escenarios para los requisitos funcionales que indican que hará el sistema. UML proporciona notación para los diagramas de caso de uso con el fin de ilustrar los nombres de los casos de uso, los actores y las relaciones entre ellos.

Elementos:

Los elementos que pueden aparecer en un diagrama de Casos de Uso son: actores, casos de uso y relaciones de casos de uso:

- **Actores**

Un actor representa como una persona, o una aplicación programada u organización y que realiza algún tipo de interacción con el sistema. Se representa mediante una figura humana dibujada. Esta representación sirve tanto para actores q son personas como para otro tipo de actores.

- **Casos de Uso**

Un caso de uso es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, el actor usa el sistema para llevar a cabo una tarea específica. Expresa una unidad coherente de funcionalidad, y representa en el Diagrama de Casos de Uso mediante una elipse con el nombre de casos de uso en su interior. El nombre de caso de uso debe reflejar la tarea específica que el actor desea llevar acabo usando el sistema.

Relaciones entre los Casos de Uso:

UML define los siguientes tipos de relación en los diagramas de casos de uso:

- **Comunicación**

Vinculación que existe entre dos o más casos de uso.

- **Inclusión**

Una instancia del Caso de Uso origen incluye también el comportamiento descrito por el Caso de Uso destino. <<include>> reemplazó al denominado <<uses>>

- **Extensión**

El caso de uso origen extiende el comportamiento del Caso de Uso destino <<extend>>. Es decir, una relación de un caso de uso A hacia un caso de uso B indica que el caso de uso B implementa la funcionalidad del caso de uso A

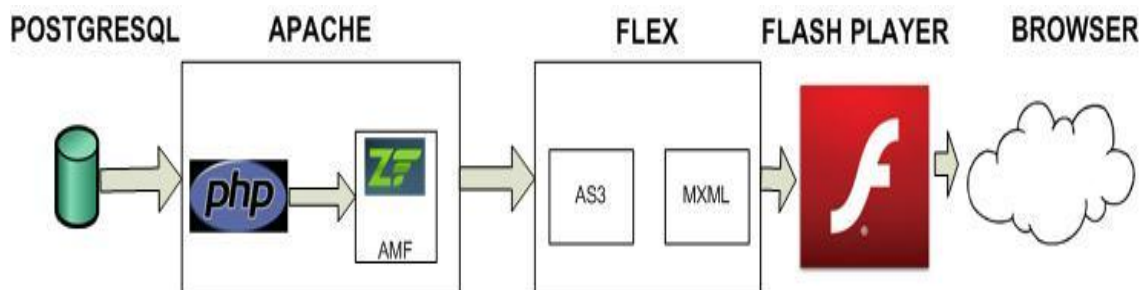
2.5.- ARQUITECTURA DEL SISTEMA

2.5.1.- OBJETIVOS ARQUITECTÓNICOS

El objetivo principal del Sistema GPI-CONTRATOS es diseñar e implementar el sistema de Gestión de Contratos en el Gobierno Provincial de Imbabura integrándolo a los módulos de GPI-GESTION usando tecnología RIA, todo esto a efecto de optimizar el uso de los recursos en el Gobierno Provincial de Imbabura.

El Sistema GPI-CONTRATOS será implementado en las siguientes plataformas:

- Servidor web: Apache
- Lenguaje de Programación: PHP
- Base de datos: PostgreSQL
- Framework MVC: ZEND
- Framework de Presentación: FLEX



Fuente: propia.

Ilustración 7: Arquitectura de la Aplicación

2.5.2.- RESTRICCIONES

El Sistema usará como motor de Base de Datos a PostgreSQL.

Patrón de arquitectura de software: *Modelo Vista Controlador (MVC)*

Se utilizará las tecnologías que brindan los framework definidos para cada una de las capas de la aplicación:

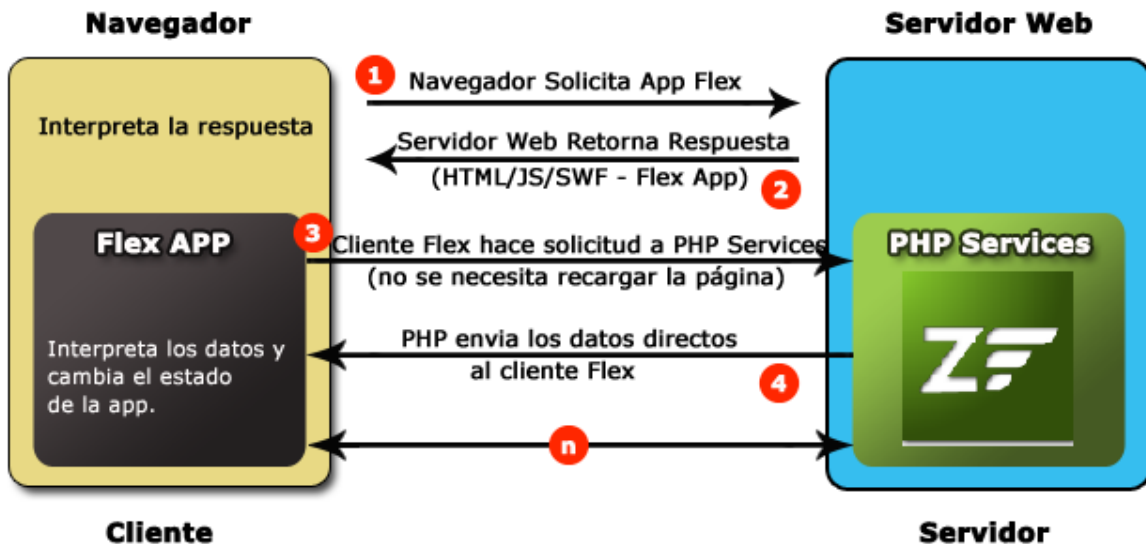
- Para la capa de presentación: Flex, aplicación Web.
- Para la capa de Acceso a Datos y Modelo de Negocio: framework Zend

2.5.3.- ARQUITECTURA FLEX ZEND²⁰

La arquitectura Flex Zend, funciona como se observa en la Ilustración 8, y que se explica a continuación:

1. Desde el navegador el cliente, solicita al servidor Web, la app Flex.
2. El servidor retorna al navegador la respuesta, que puede ser HTML, JS, SWF – Flex App, PHP.
3. Una vez cargada la aplicación Flex en el navegador, el cliente hace solicitud de servicios php mediante zendamf, que se encuentran del lado del servidor, esto sin necesidad de recargar la página.
4. PHP retorna los datos directamente a cliente Flex.
5. Se repite la iteración entre Flex y PHP, sin necesidad de recargar la página en el navegador n veces.

²⁰ Flores, H. (2002). *Arquitectura Flex Zend*. Obtenido de http://haroldflores.com/blog_introduccion-aplicaciones-php-con-flex.php



Fuente: http://haroldflores.com/blog_introducion-aplicaciones-php-con-flex.php.
 Ilustración 8: Arquitectura Flex Zend

En una aplicación Flex, se maneja un concepto de capas, donde la capa de lógica del negocio y acceso a base de datos se encuentra del lado del servidor y es manejada por la aplicación PHP, mientras la capa de presentación o vista es manejada por Flex que ofrece el mecanismo para llamar a las páginas del servidor, y obtener la respuesta, y para esto existen dos formas: sobre http o utilizando sockets.

Sobre http, hay cuatro formas distintas de conectarse al servidor, REST SERVICES, WEB SERVICES, REMOTING, RPC (CALLRESPONDER).²¹

Se puede utilizar el HTTP SERVICE CLASS, para utilizar REST SERVICES, mandando las variables POST en la solicitud y recibiendo la respuesta en XML.

WEBSERVICE Class, la podemos utilizar en el caso que tengamos un WEB SERVICES en el servidor (SOAP/WSDL).

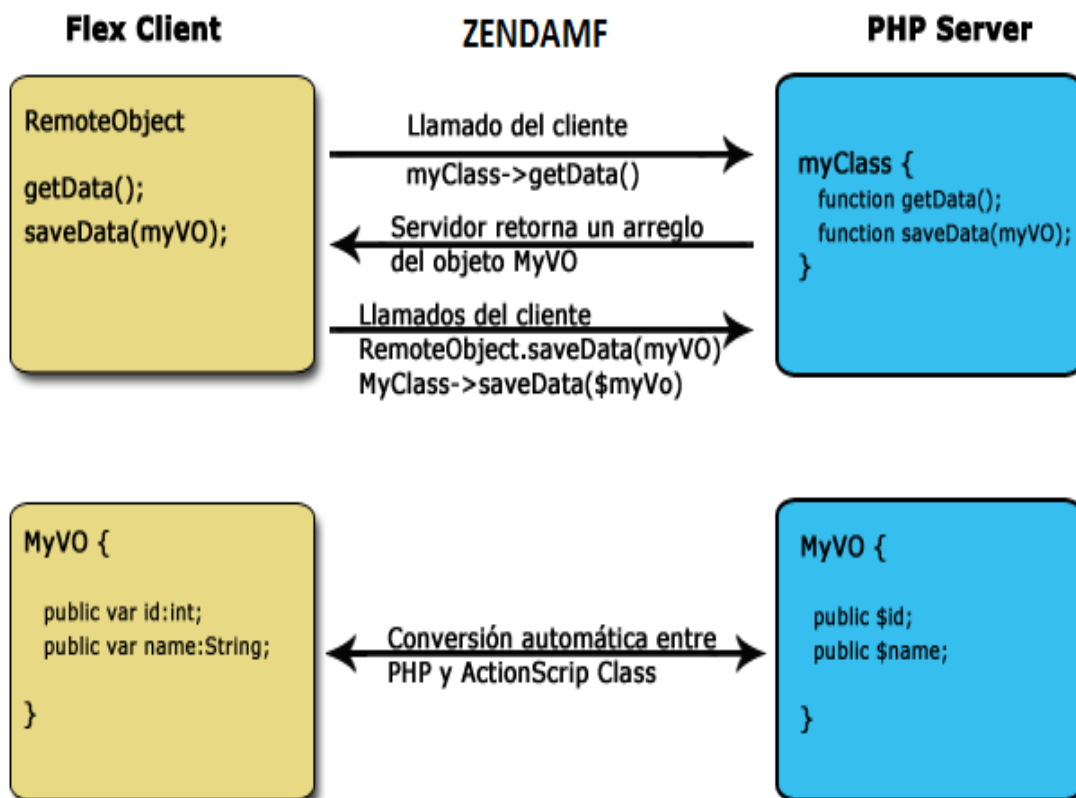
Y la forma que con la que se desarrollará la aplicación, es al acceso Remoting. Para esta se utiliza la class RPC (CALLRESPONDER).

Ventajas de Utilizar CALLRESPONDER

1. Se puede acceder a cualquier clase PHP que este en el servidor, por medio de sus métodos públicos utilizando los servicios creados en PHP.

²¹ Wikipedia. (s.f.). RPC. Obtenido de http://es.wikipedia.org/wiki/Remote_Procedure_Call

- Se puede mapear el modelo de datos de PHP, al modelo de datos en Action Script, que es muy importante porque se tiene el beneficio de chequeo de error en tiempo de compilación y ayuda para completar código, en el lado de Flex.
- El formato de mensajes para este método es ZENDAMF (Zend Action Message Format), que es binario que lo hace más rápido y pequeño comparado con SOAP/XML/JSON.



Fuente: http://haroldflores.com/blog_introduccion-aplicaciones-php-con-flex.php
Ilustración 9: Flujo de Datos ZENDAMF

2.5.4.- DESCRIPCIONES DEL SISTEMA

Esta vista describe la estructura total del modelo de desarrollo, la descomposición del software en capas y subsistemas. Seguirá el patrón Model View Controller (MVC)²², el cual plantea la siguiente división:

²² MVC.- Patrón de Diseño Modelo Vista Controlador

➤ **Capa Vista (View)**

Esta capa maneja las clases que permiten la interacción del usuario con el sistema.

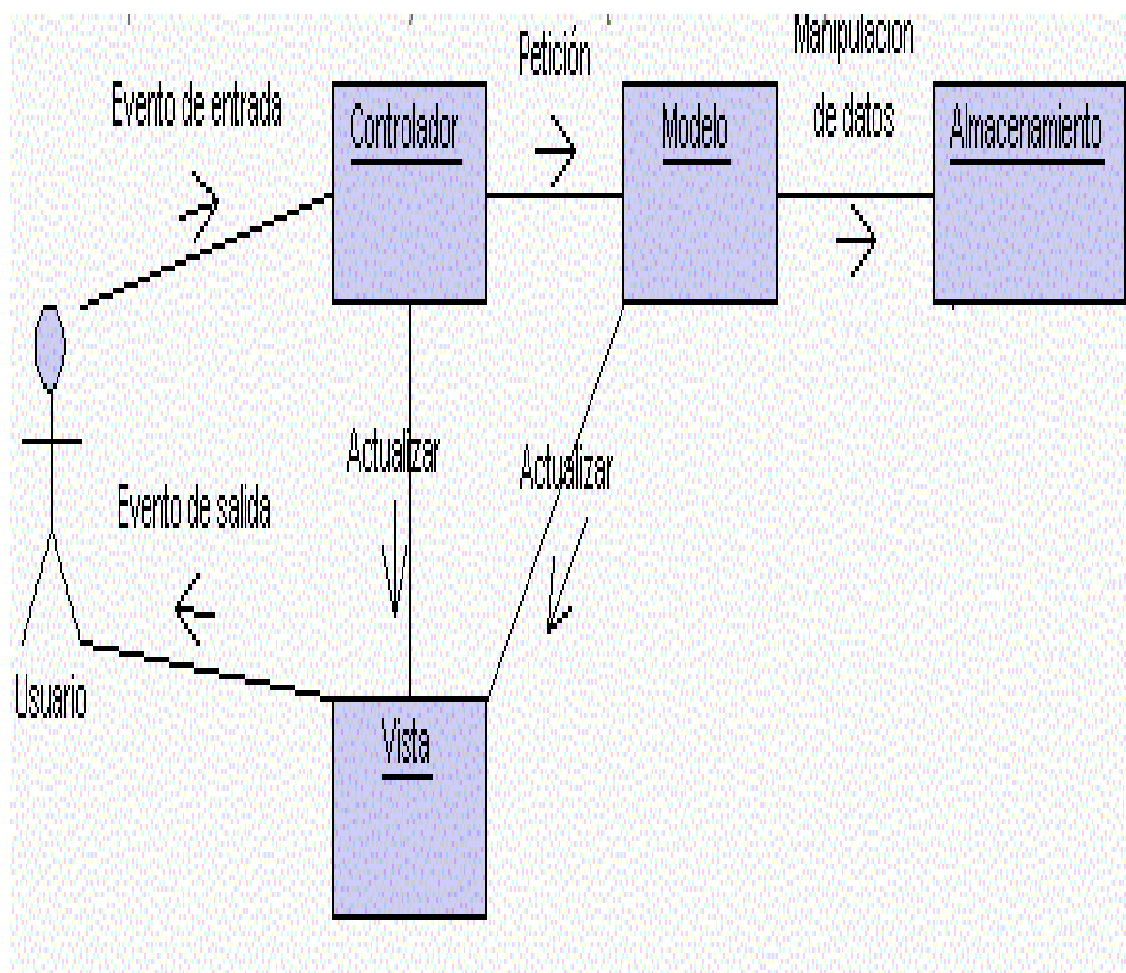
Datos y reglas de negocio.

➤ **Capa Modelo (Model)**

Esta capa representa la realidad. Representa a las entidades de negocio y a las clases encargadas de la persistencia de la información. Muestra la información del modelo al usuario.

➤ **Capa Controlador (Controller)**

Esta capa permite manejar la lógica del negocio de los módulos involucrados.



Fuente: propia.

Ilustración 10: Estructura del patrón MVC

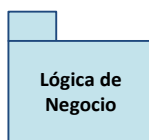
La vista lógica del sistema GPI-CONTRATOS está compuesta de cuatro paquetes principales:

1. Interfaz de usuario (GUI),
2. Servicios del negocio (Business Logic),
3. Objetos del negocio (Business Entity)
4. Objetos de persistencia (Data Access Objects).

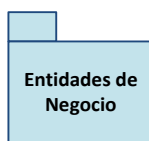
El primer paquete, Interfaz de usuario, contiene las clases visuales de la aplicación, las cuales crean las ventanas que utiliza el usuario para comunicarse con el sistema para una adecuada y fácil utilización de los paquetes que se encuentran en las siguientes capas.



El paquete Lógica del Negocio contiene las clases que permiten llevar a cabo los servicios del negocio, como son: Controlar el mantenimiento de usuarios, gestionar fines, métodos etc.



El paquete Entidades del Negocio contiene las clases que utilizará el sistema, entre las principales clases se tiene: Fines, Métodos, Indicadores, etc.

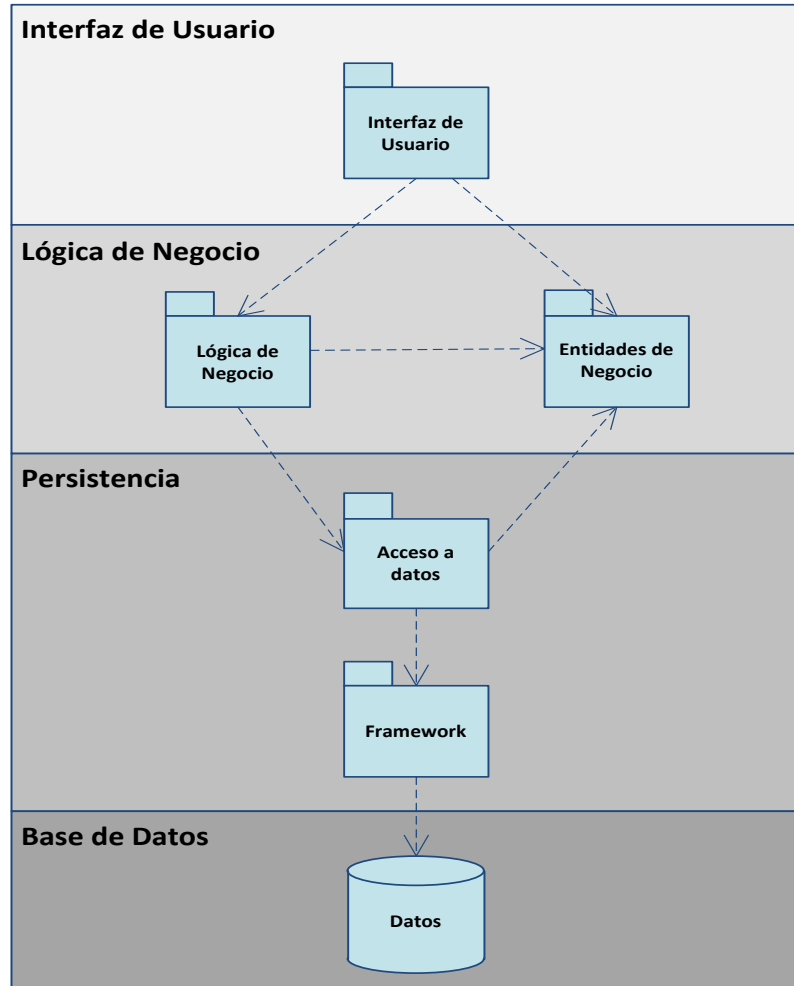


El paquete de Objetos de Acceso de Datos contiene las clases que el sistema utiliza para manejar la persistencia de los objetos con la Base de Datos.



2.5.4.1. Diseño en Capas

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

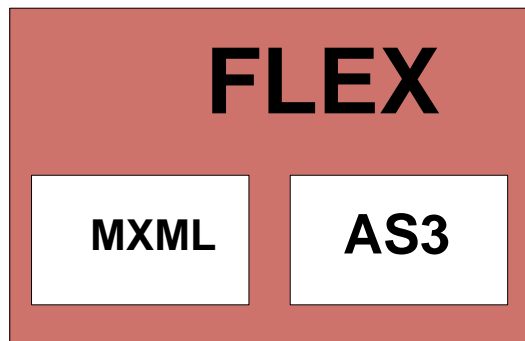


Fuente: propia.
Ilustración 11: Diseño de Capas

2.5.4.2. Capa Interfaz de Usuario

Esta capa contiene el paquete Interfaz de Usuario, el cual almacena todas las clases con las cuales el usuario puede interactuar como lo son las ventanas.

Las interfaces del usuario están diseñadas en Flex 4, que maneja componentes MXML y el lenguaje ActionScript 3, AS3



Fuente: propia.
Ilustración 12: Capa Interfaz de Usuario

Los componentes serán diseñados en componentes MX y Spark. Flex se definen dos tipos de componentes: Spark y MX. Los componentes Spark son nuevos para Flex 4 y se definen spark. Paquetes *. Los componentes de dominio enviado en versiones anteriores de Flex y se definen en el mx. * Paquetes.

Las principales diferencias entre spark y los componentes de MX son cómo utilizar los estilos CSS con los componentes y la forma de la máscara (skins). Para los componentes de contenedores, hay diferencias adicionales acerca de cómo realizar el diseño de los contenedores.

Spark y MX definir algunos de los mismos componentes. Por ejemplo, el Spark define un control de botón en el paquete de spark.components, y MX define un control de botón en el paquete mx.controls. Cuando un componente está disponible en la Spark y MX, Adobe recomienda que utilice el componente Spark.

Spark y MX también definen componentes que son únicos. Por ejemplo, el Spark define los componentes para realizar efectos tridimensionales. MX define los componentes de visualización de datos, tales como la cuadrícula de datos y los controles AdvancedDataGrid, no incluidos en el Spark. Las aplicaciones a menudo contiene una mezcla de componentes Spark y MX.

2.5.4.3. Capa Lógica del Negocio

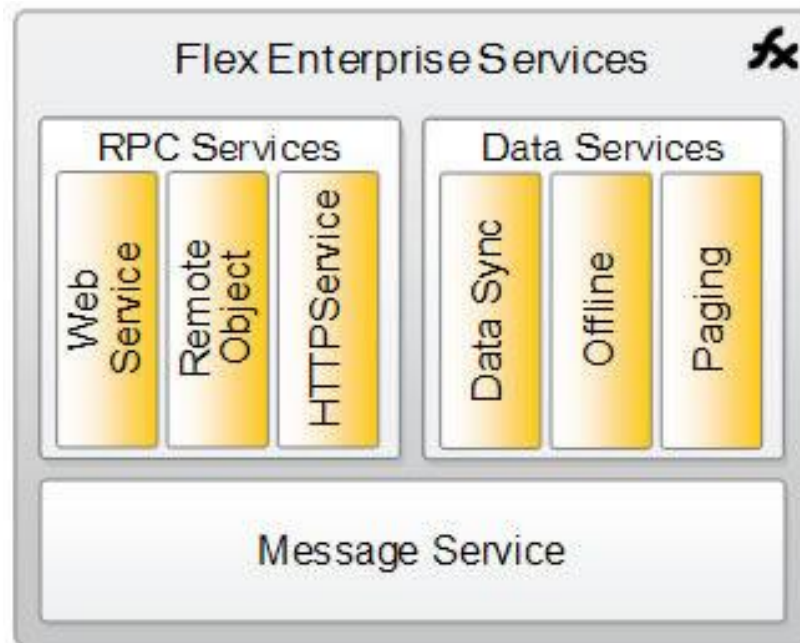
Esta capa contiene los paquetes de Servicios de Negocio y Entidades de Negocio. Contiene la lógica para el manejo de las operaciones del negocio.

La lógica del negocio está dada por los servicios php y ValueObjects VO, servicios Flex DataServices DS

Los servicios php son clases php que contienen la referencia al modelo zend y además contienen los métodos públicos que dan la funcionalidad CRUD y otras funcionalidades y los métodos privados que son complemento de los métodos públicos

Los DS o también conocidos como servicios de Flex son funcionalidades de Flex que contienen la referencia a los servicios php.

Existen diferentes tipos de servicios los cuales visualizamos en la siguiente ilustración.



Fuente: <http://www.avokasystems.com>
Ilustración 13: Flex Enterprise Services

HTTPService.- Mediante el cual podemos consumir XML sobre HTTP principalmente. Este servicio es usado para hacer conexiones HTTP (o HTTPS) más tradicionales incluyendo los datos dentro del cuerpo de un mensaje get o post.

RemoteObject.- Acceso directo a métodos en objetos java en el servidor mediante AMF (un protocolo binario y ligero basado en SOAP). También podemos acceder a servicios en otros lenguajes puesto que ya existen diversas implementaciones la mayoría open source.

Web Services – Acceso a Servicios Web de la manera tradicional mediante SOAP. Usado principalmente cuando necesitamos un estándar para comunicarnos por encima del rendimiento en las comunicaciones.

2.5.4.4. Capa Persistencia

Esta capa contiene el paquete de Objetos de Acceso de Datos, que brinda una interfaz transparente para la interacción con el Framework el cual enviará al Driver el conjunto de sentencias para interactuar con la Base de Datos.

Framework: ZEND²³

Zend Framework es un framework open source para PHP desarrollado por Zend, empresa encargada de la mayor parte de las mejoras hechas a PHP, por lo que se podría decir que es el framework “oficial”. Zend Framework se referirá por sus iniciales ZF²⁴.

Características principales

- ZF implementa el patrón MVC, es 100% orientado a objetos y sus componentes tienen un bajo acoplamiento por lo que los puedes usar en forma independiente. Un punto importante es que nos brinda un estándar de codificación que deberíamos seguir en nuestros proyectos.
- A su vez, cuenta con soporte para internalización y localización de aplicaciones (construir sitios multi idioma, convertir formatos de fechas, monedas, etc. según la región. Algo importantísimo para crear aplicaciones con un enfoque global y llegar de la mejor manera a la mayor cantidad de gente posible).
- Facilita el setup de nuestro proyecto brindándonos herramientas para crear la estructura de directorios, clases, etc. por línea de comandos, integración con PHPUnit por medio de Zend_Test para facilitar el testing de nuestra aplicación.
- Posee adapters para gran cantidad de tipos de bases de datos diferentes; brinda componentes para la autenticación y autorización de usuarios, envío de mails, cache en varios formatos, creación de web services, etc.

2.6.- HERRAMIENTAS DE DESARROLLO

2.6.1.- PHP²⁵

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

²³ maestrosdelweb. (s.f.). *Zend Framework*. Obtenido de <http://www.maestrosdelweb.com/editorial/guia-zend/>

²⁴ ZF.- Zend Framework

²⁵ Wikipedia. (s.f.). *PHP*. Obtenido de <http://es.wikipedia.org/wiki/PHP>

los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

2.6.2.- FLEX²⁶

Adobe Flex (hasta 2005 Macromedia Flex) es un término que agrupa una serie de tecnologías publicadas desde Marzo de 2004 por Macromedia para dar soporte al despliegue y desarrollo de Aplicaciones Enriquecidas de Internet, basadas en su plataforma propietaria Flash.

Flex fue inicialmente liberado como una aplicación de la J2EE o biblioteca de etiquetas JSP que compilaba el lenguaje de marcas Flex (MXML) y ejecutaba mediante ActionScript aplicaciones Flash (archivos SWF binarios). Versiones posteriores de Flex soportan la creación de archivos estáticos que son compilados, y que pueden ser distribuidos en línea sin la necesidad de tener una licencia de servidor.

El objetivo de Flex es permitir a los desarrolladores de aplicaciones web construir rápida y fácilmente Aplicaciones de Internet Ricas, también llamadas RIAs. En un modelo multi-capa, las aplicaciones Flex son el nivel de presentación.

Flex pone en relieve el desarrollo de Interfaces gráficas de usuario usando un lenguaje XML llamado MXML. Flex tiene varios componentes y características que aportan funcionalidades tales como Servicios Web, objetos remotos, arrastrar y soltar, columnas ordenables, gráficas, efectos de animación y otras interacciones simples.

El cliente solo carga la aplicación una vez, mejorando así el flujo de datos frente a aplicaciones basadas en HTML (PHP, ASP, JSP, CFMX), las cuales requieren de ejecutar plantillas en el servidor para cada acción. El lenguaje y la estructura de archivos de Flex buscan el desacoplamiento de la lógica y el diseño.

El servidor Flex también actúa como un gateway permitiendo al cliente comunicarse con servicios web XML y objetos remotos (tales como Coldfusion CFCs, clases Java, y cualquiera que soporte el formato de mensajes de acciones).

²⁶ Wikipedia. (s.f.). *FLEX*. Obtenido de http://es.wikipedia.org/wiki/Adobe_Flex

2.6.3.- ACTIONSCRIPT²⁷

Adobe ActionScript es el lenguaje de programación de la Plataforma Adobe Flash. Originalmente desarrollado como una forma para que los desarrolladores programen de forma más interactiva. La programación con ActionScript permite mucha más eficiencia en las aplicaciones de la plataforma Flash para construir animaciones de todo tipo, desde simples a complejas, ricas en datos e interfaces interactivas.

2.6.4.- AS3²⁸

ActionScript 3.0 ofrece un modelo de programación robusto que resultará familiar a los desarrolladores con conocimientos básicos sobre programación orientada a objetos. Algunas de las principales funciones de ActionScript 3.0 son:

- Una nueva máquina virtual ActionScript, denominada AVM2²⁹, que utiliza un nuevo conjunto de instrucciones de código de bytes y proporciona importantes mejoras de rendimiento.
- Una base de código de compilador más moderna, que se ajusta mejor al estándar ECMAScript³⁰ (ECMA 262) y que realiza mejores optimizaciones que las versiones anteriores del compilador.
- Una interfaz de programación de aplicaciones (API) ampliada y mejorada, con un control de bajo nivel de los objetos y un auténtico modelo orientado a objetos.
- Un núcleo del lenguaje basado en el próximo borrador de especificación del lenguaje ECMAScript (ECMA-262) edición 4.
- Una API XML basada en la especificación de ECMAScript para XML (E4X) (ECMA-357 edición 2). E4X es una extensión del lenguaje ECMAScript que añade XML como un tipo de datos nativo del lenguaje.
- Un modelo de eventos basado en la especificación de eventos DOM (modelo de objetos de documento) de nivel 3.

2.6.4.1. VENTAJAS DE AS3

ActionScript 3.0 aumenta las posibilidades de creación de scripts de las versiones anteriores de ActionScript. Se ha diseñado para facilitar la creación de aplicaciones muy complejas con conjuntos de datos voluminosos y bases de código reutilizables y

²⁷ Wikipedia. (s.f.). *ActionScript*. Obtenido de <http://es.wikipedia.org/wiki/ActionScript>

²⁸ Wikipedia. (s.f.). *AS3*. Obtenido de <http://es.wikipedia.org/wiki/ActionScript>

²⁹ Adobe. (s.f.). *AVM2.- Máquina Virtual de ActionScript*. Recuperado de <http://www.adobe.com/content/dam/Adobe/en/devnet/actionscript/articles/avm2overview.pdf>

³⁰ Wikipedia. (s.f.). *ECMAS.- especificación de un lenguaje de programación*. Recuperado de <http://es.wikipedia.org/wiki/ECMAScript>

orientadas a objetos. Aunque no se requiere para el contenido que se ejecuta en Adobe Flash Player 9, ActionScript 3.0 permite introducir unas mejoras de rendimiento que sólo están disponibles con AVM2, la nueva máquina virtual. El código ActionScript 3.0 puede ejecutarse con una velocidad diez veces mayor que el código ActionScript heredado.

La versión anterior de la máquina virtual ActionScript (AVM1) ejecuta código ActionScript 1.0 y ActionScript 2.0. Flash Player 9 admite AVM1 por compatibilidad con contenido existente y heredado de versiones anteriores.

2.6.4.2. ESTRUCTURA

Flash está compuesto por objetos, con su respectiva ruta dentro del swf. Cada uno de estos en ActionScript pertenece a una clase (MovieClip, botones, Vectores (Arrays), etc.), que contiene Propiedades y Métodos o funciones.

- **Propiedades:** Dentro del archivo raíz de la clase, están declaradas como variables (alpha, useHandCursor, length).
- **Métodos o funciones:** Dentro del archivo raíz de la clase, están declaradas como funciones (stop(), gotoAndPlay(), getURL()).

2.6.4.3. CLASES

- Algunas clases de ActionScript son:
- Accessibility (nivel superior)
- Array (instancias)
- Boolean (instancias)
- Button (instancias)
- Capabilities (nivel superior)
- Color (instancias)
- ContextMenu (instancias)
- ContextMenuItems (instancias)
- Date (instancias)
- Error (instancias)
- Key (nivel superior)
- LoadVars (instancias)
- Math (nivel superior)
- Mouse (nivel superior)
- MovieClip (instancias)
- MovieClipLoader (instancias)

- NetConnection (instancias)
- NetStream (instancias)
- Number (nivel superior)
- Object (instancias)
- PrintJob (instancias)
- Selection (nivel superior)
- Sound (instancias)
- Stage (nivel superior)
- String (instancias)
- StyleSheet (instancias)
- System (nivel superior)
- TextField (instancias)
- TextFormat (instancias)
- XML (instancia)
- XMLSocket

2.6.5.- MXML³¹

MXML es un lenguaje descriptivo desarrollado inicialmente por Macromedia hasta el 2005 para la plataforma FLEX de Adobe. MXML se basa en XML y su acrónimo "Macromedia eXtensible Markup Language" Lenguaje que describe interfaces de usuario, crea modelos de datos y tiene acceso a los recursos del servidor, del tipo RIA³² (Rich Internet Application).

MXML tiene una mayor estructura en base a etiquetas, similar a HTML, pero con una sintaxis menos ambigua, proporciona una gran variedad e inclusive permite extender etiquetas y crear sus propios componentes.

Una vez compilado genera ficheros .swf³³

2.6.6.- BASE DE DATOS POSTGRESQL³⁴

Los sistemas de mantenimiento de Bases de Datos relacionales tradicionales (DBMS³⁵) soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. En los sistemas comerciales actuales, los

³¹ Wikipedia. (s.f.). *MXML.- Macromedia eXtensible Markup Language*. Obtenido de <http://es.wikipedia.org/wiki/MXML>

³² Wikipedia. (s.f.). *RIA.- Aplicaciones de internet enriquecidas*. Obtenido de <http://es.wikipedia.org/wiki/RIA>

³³ Wikipedia. (s.f.). *Swf.- Archivo compilado de flash*. Obtenido de <http://www.wikipedia.com/>

³⁴ postgresql. (Febrero de 2013) *PostgreSQL*. Obtenido de <http://www.postgresql.org/>

³⁵ Wikipedia. (s.f.). *DBMS.- Sistema manejador de base de datos*. Obtenido de <http://www.wikipedia.com/>

tipos posibles incluyen numéricos de punto flotante, enteros, cadenas de caracteres, cantidades monetarias y fechas. Está generalmente reconocido que este modelo será inadecuado para las aplicaciones futuras de procesamiento de datos. El modelo relacional sustituyó modelos previos en parte por su "simplicidad espartana". Sin embargo, como se ha mencionado, esta simplicidad también hace muy difícil la implementación de ciertas aplicaciones.

PostgreSQL ofrece una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos en una vía en la que los usuarios pueden extender fácilmente el sistema:

- clases
- herencia
- tipos
- funciones

Otras características aportan potencia y flexibilidad adicional:

- Restricciones (Constraints)
- Disparadores (triggers)
- Reglas (rules)
- Integridad transaccional

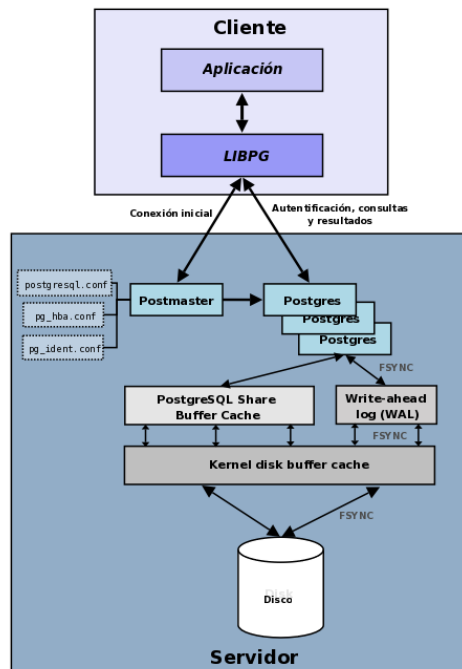
Estas características colocan a PostgreSQL³⁶ en la categoría de las Bases de Datos identificadas como objeto-relacionales. Nótese que éstas son diferentes de las referidas como orientadas a objetos, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. PostgreSQL tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos. De hecho, algunas Bases de Datos comerciales han incorporado recientemente características en las que PostgreSQL fue pionera.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD³⁷ y con su código fuente disponible libremente. 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. A continuación tenemos un gráfico que ilustra de manera general los componentes más importantes en un sistema PostgreSQL.

³⁶ PostgreSQL.- Motor de base de datos libre.

³⁷ Wikipedia. (s.f.). *BSD.- Licencia de software*. Recuperado de https://es.wikipedia.org/wiki/Licencia_BSD

2.6.6.1. ARQUITECTURA POSTGRESQL



Fuente: http://www.postgresql.org.es/sobre_postgresql
Ilustración 14: Arquitectura PostgreSQL

- **Aplicación cliente:** Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP o sockets locales.
- **Demonio postmaster:** Este es el proceso principal de PostgreSQL. Es el 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 autenticar 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 PostgreSQL:** Procesos hijos que se encargan de autenticar 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 (WAL³⁸):** Componente del sistema encargado de asegurar la integridad de los datos (recuperación de tipo REDO)

³⁸ Wikipedia. (s.f.). WAL.- Asegura la integridad de datos. Recuperado de <http://www.slideshare.net/eddiemalca/clase-2-arquitectura-de-postgres>

- **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.

2.6.6.2. CARACTERÍSTICAS DE POSTGRESQL

Sus características técnicas la hacen una de las bases de datos más potentes y robustos del mercado. Su desarrollo comenzó hace más de 16 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

A continuación tienen algunas de las características más importantes y soportadas por PostgreSQL:

Características Generales

- Es una base de datos 100% ACID³⁹
- Integridad referencial
- Tablespaces
- Nested transactions (savepoints)
- Replicación asincrónica/sincrónica / Streaming replication - Hot Standby
- Two-phase commit
- PITR - point in time recovery
- Copias de seguridad en caliente (Online/hot backups)
- Unicode
- Juegos de caracteres internacionales
- Regionalización por columna
- Multi-Version Concurrency Control (MVCC)
- Múltiples métodos de autenticación
- Acceso encriptado vía SSL
- Actualización in-situ integrada (pg_upgrade)
- SE-PostgreSQL
- Completa documentación
- Licencia BSD

³⁹ Wikipedia. (s.f.). ACID.- *Atomicidad, consistencia, aislamiento, durabilidad*. Recuperado de <http://es.wikipedia.org/wiki/ACID>

- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.

Programación / Desarrollo

- Funciones/procedimientos almacenados (stored procedures⁴⁰) en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de oracle), PL/Perl, PL/Python y PL/Tcl
- Bloques anónimos de código de procedimientos (sentencias DO)
- Numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido)
- APIs⁴¹ para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.

SQL

- SQL92,SQL99,SQL2003,SQL2008
- Llaves primarias (primary keys) y foráneas (foreign keys)
- Check, Unique y Not null constraints
- Restricciones de unicidad postergables (deferrable constraints)
- Columnas auto-incrementales
- Índices compuestos, únicos, parciales y funcionales en cualquiera de los métodos de almacenamiento disponibles, B-tree, R-tree, hash ó GiST
- Sub-selects
- Consultas recursivas
- Funciones 'Windows'
- Joins
- Vistas (views)
- Disparadores (triggers) comunes, por columna, condicionales.
- Reglas (Rules)
- Herencia de tablas (Inheritance)
- Eventos LISTEN/NOTIFY

⁴⁰ Store Procedure.- procedimientos almacenados

⁴¹ APIS.- Interfaz de programación de aplicaciones

Límites.- Algunos de los límites de PostgreSQL son:

Límite	Valor
Máximo tamaño base de dato	Ilimitado (Depende de tu sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo numero de filas por tabla	Ilimitado
Máximo numero de columnas por tabla	250 - 1600 (dependiendo del tipo)
Máximo numero de indices por tabla	Ilimitado

Fuente: http://www.postgresql.org.es/sobre_postgresql
Tabla 2: Límites de PostgreSQL

2.6.7.- Servidor Web APACHE⁴²

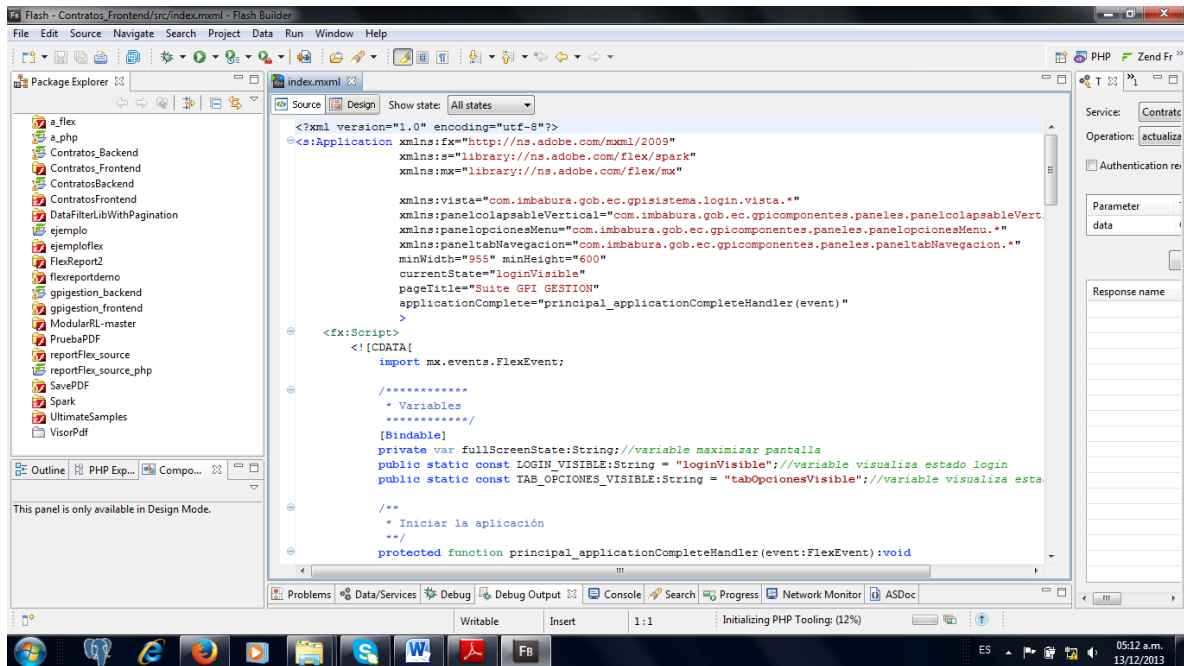
El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPD 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor "parcheado") suena igual que Apache Server.

en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache.

⁴² Wikipedia. (s.f.). APACHE. Recuperado el 22 de Junio de 2013, de <http://es.wikipedia.org/wiki/Apache>

2.6.8.- IDE FLASH BUILDER 4.5 FOR PHP

Flash Builder 4.5 for PHP es un entorno de desarrollo integrado basado en eclipse y desarrollado por Adobe que interactúa con dos grandes herramientas como son Zend y Flex



Fuente: Propia
Ilustración 15: Adobe Flash Builder for PHP

CAPÍTULO III

3. PLANIFICACIÓN DE LA ITERACIÓN (FASE DE INICIO)

3.1.- ANÁLISIS Y DEFINICIÓN DE REQUERIMIENTOS

3.1.1.- RELATO DEL USUARIO

Punto 1. Acercamiento con el cliente

El área de sistemas del “Gobierno Provincial de Imbabura” y su afán de establecer estándares de arquitectura que debe seguir el área de TIC⁴³ para la implementación de sistemas de información acuerdan realizar el módulo de Gestión de Contratos para el manejo correcto de la información generada en los mismos previo al análisis de especificaciones y requerimientos recolectados en las áreas afectadas dentro de la Institución.

Punto 2. Análisis y definición de requerimientos del sistema GPI-CONTRATOS

Las diferentes áreas implicadas en la gestión de contratos y las personas a cargo de cada departamento especifican los siguientes requerimientos para el sistema GPI-CONTRATOS.

Punto 2.1. Módulo Administración:

El Administrador tiene la capacidad de crear nuevos usuarios o entidades del sistema, determinar el perfil de cada uno, asignar módulos según roles de usuario. Esto accede mediante el módulo GPI-SISTEMA⁴⁴ que es parte de la suite de aplicaciones del Gobierno provincial de Imbabura GPI-GESTIÓN.

Punto 2.2. Módulo Plantilla de Contrato:

Ingresar un modelo o plantilla que será única para cada contrato generado dentro del GPI, las plantillas se podrán crear, modificar y eliminar.

Punto 2.3. Módulo de Contrato:

Crear un contrato a partir de una plantilla ingresada ya existente para optimizar recursos y tiempo, contratos tales como:

- Ínfima cuantía Bienes y Servicios

⁴³ TIC.- Tecnologías de la Información y Comunicación

⁴⁴ GPI-SISTEMA.- Módulo de GPI-GESTIÓN que controla el acceso a la aplicación.

- Menor cuantía Bienes y Servicios
- Cotización Bienes y Servicios
- Menor Cuantía Obras
- Cotización de Obras
- Licitación de Obras
- Entre otros.
- Plazo Fijo
- Servicios Ocasionales

Donde cada contrato tiene una diferente plantilla ya que las cláusulas cambian en cada tipo de contrato.

Ingresar datos en cada tipo de contrato establecido anteriormente, donde cada contrato tiene:

- Ingreso del número de contrato
- Ingreso del contratista
- Ingreso del tipo de contrato
- Ingreso de la descripción del contrato
- Ingresar la fecha de suscripción del contrato.
- Ingresar el monto del contrato.
- Ingresar la ubicación en donde se ejecutara el contrato.
- Ingreso del Administrador o Fiscalizador del contrato.
- Entre otros datos.

Las plantillas de cada contrato ingresadas se pueden crear, modificar; y eliminar esto depende si la plantilla ha cambiado según datos de la INCOP⁴⁵.

Punto 2.4. Módulo Garantías del Contrato:

Ingresar los datos de las garantías establecidas para cada contrato existente en el "GPI", tales como: Garantías Bancarias, Pólizas de Seguro, Letras de cambio (Montos pequeños).

Se procederá a controlar las fechas de vencimientos y renovaciones de garantías y si en cada contrato se cumple estas garantías antes mencionadas.

⁴⁵ INCOP.- Instituto Nacional de Contratación Pública.

Punto 2.5. Módulo Pagos del Contrato:

Los pagos de cada contrato se realizaran mediante planillas, cada planilla que se presenta se va ingresando al sistema, datos como:

Pagos:

- Anticipo
 - Fecha de entrega del anticipo
 - % del anticipo de acuerdo con lo establecido en el contrato.
- Pagos contra presentación de planillas periódicas debidamente aprobadas por Fiscalización.
 - Fecha de presentación de planillas aprobadas.
 - Valor del pago de la planilla.

Punto 2.6. Módulo Fiscalización del Contrato:

Ingresar al sistema datos para fiscalización de la obra tales como:

- Fecha de inicio de obra
- Fecha de entrega del anticipo de dinero
- Fiscalizador de obra
- Administrador de obra
- Datos de avance de obra
- Fecha de recepción provisional de la obra
- Fecha de recepción definitiva de la obra

Punto 3. Módulo Reportes del sistema GPI-CONTRATOS

- Lista de todos los Contratos generados en el GPI por:
 - Número de Contrato
 - Contratista
 - Fecha
 - Ubicación
 - Tipo
- Reporte de Garantías
 - Número de contrato
 - Contratista
 - Aseguradora

- Tipo
- Fecha de Vencimiento
- Fecha de Emisión
- Reporte de Fiscalizadores y Administradores
 - Lista de Contratos a Fiscalizar
 - Contratos Fiscalizados
 - Lista de Fiscalizadores para cada Contrato

Responsabilidades asumidas:

Se termina la reunión logrando así los requerimientos que estarán registrados en documentos llamados historias de usuario para proceder a establecer el Product Backlog del sistema y así empezar a desarrollar el sistema GPI-CONTRATOS y esperando una próxima y pronta reunión para continuar con las especificaciones y funcionalidades del sistema.

3.1.2.- FORMATO DE LA HISTORIA DE USUARIO

SCRUM puede ser mejorado a través de las historias de usuario y menciona que en lugar que los ítems del Product Backlog constituya nuevas características, problemas a investigar, errores a ser corregidos, y demás, estos sean exclusivamente historias de usuario.

Una historia de usuario es una descripción corta y simple acerca de una funcionalidad concebida desde la perspectiva de la persona que la usara, usualmente un usuario del sistema.

Las historias de usuario constan de tres partes importantes:

- **ID.-** identificador único de la historia de usuario
- **Usuario.-** nombre de la persona q lista los requerimientos.
- **Nombre de la Historia.-** descripción corta de la historia
- **Prioridad en negocio.-** definido por el propietario del producto, es el grado de prioridad para el desarrollo de cada historia de usuario y se mide en tres puntos:
 - ALTA MEDIA BAJA
- **Iteración Asignada.-** número a q iteración o sprint pertenece la historia de usuario.
- **Programador responsable.-** persona responsable de desarrollar la historia de usuario propuesta.

- **Descripción.**- descripción de la historia de usuario a fondo.

HISTORIA DE USUARIO	
Número:	Usuario:
Nombre de Historia:	
Prioridad en negocio:	
Iteración asignada:	
Programador responsable:	
Descripción:	

Fuente: propia.
Tabla 3: Formato Historias de Usuarios

3.1.3.- HISTORIAS DE USUARIOS

Las historias de usuario se representan a continuación de manera tabular, son exactamente los requerimientos podados por el cliente en sus propias palabras.

HISTORIA DE USUARIO	
Número: 1	Usuario: Administrador del Sistema
Nombre de Historia: Autenticación de Usuarios en el Sistema	
Prioridad en negocio: BAJA	
Iteración asignada: 1	
Programador responsable: Aníbal Benalcázar	
Descripción: El sistema va a tener varios roles para las diferentes funciones, se requiere que el sistema permita ingresar el sistema bajo una contraseña, un usuario y con un rol para que pueda asignar los módulos autorizados según su rol.	

Fuente: propia.
Tabla 4: Historia de Usuario 1

HISTORIA DE USUARIO	
Número: 2	Usuario: Administrador del Sistema
Nombre de Historia: Creación y Modificación de Objetos de Contratación	
Prioridad en negocio: ALTA	
Iteración asignada: 2	
Programador responsable: Aníbal Benalcázar	

Descripción:

Se requiere ingresar los objetos de contratación para tener un listado de los contratos generados en el gpi según los objetos de contratación como:

BIENES Y SERVICIOS

NORMALIZADOS

NO NORMALIZADOS

OBRAS

CONSULTORÍA

TALENTO HUMANO

Fuente: propia.
Tabla 5: Historia de Usuario 2

HISTORIA DE USUARIO	
Número: 3	Usuario: Administrador del Sistema
Nombre de Historia: Creación y Modificación de Tipos de Contratos	
Prioridad en negocio: ALTA	
Iteración asignada: 2	
Programador responsable: Aníbal Benalcázar	
Descripción: Se requiere ingresar los tipos de contratos para tener un listado de los contratos generados en el gpi según los tipos de contratos como: Ínfima cuantía Bienes y Servicios Menor cuantía Bienes y Servicios Cotización Bienes y Servicios Menor Cuantía Obras Cotización de Obras Licitación de Obras Entre otros. Plazo fijo Servicios Ocasionales	

Fuente: propia.
Tabla 6: Historia de Usuario 3

HISTORIA DE USUARIO	
Número: 4	Usuario: Administrador del Sistema, Abogado
Nombre de Historia: Crear Plantilla de contrato	
Prioridad en negocio: ALTA	
Iteración asignada: 3	
Programador responsable: Aníbal Benalcázar	
<p>Descripción:</p> <p>Crear un contrato a partir de una plantilla ingresada ya existente para optimizar recursos y tiempo, contratos tales como:</p> <p>Ínfima cuantía Bienes y Servicios Menor cuantía Bienes y Servicios Cotización Bienes y Servicios Menor Cuantía Obras Cotización de Obras Licitación de Obras Entre otros.</p>	

Fuente: propia.
Tabla 7: Historia de Usuario 4

HISTORIA DE USUARIO	
Número: 5	Usuario: Abogado, Secretaria
Nombre de Historia: Ingreso y Modificación Contratos	
Prioridad en negocio: ALTA	
Iteración asignada: 4	
Programador responsable: Aníbal Benalcázar	
<p>Descripción:</p> <p>Previo a la selección de un contrato dentro del GPI se debe tener un sistema que permita ingresar todos los datos de dicho contrato para llevar un control de todos los contratos que se generan en el GPI.</p> <p>Los datos generados en un contrato son:</p>	

Ingreso del número de contrato
Ingreso del contratista
Ingreso de la descripción del contrato
Ingresar la fecha de suscripción del contrato.
Ingresar el monto del contrato.
Ingresar la ubicación en donde se ejecutara el contrato.
Ingreso del Administrador o Fiscalizador del contrato.

Fuente: propia.
Tabla 8: Historia de Usuario 5

HISTORIA DE USUARIO	
Número: 6	Usuario: Administrador de Contrato
Nombre de Historia: Control de Garantías	
Prioridad en negocio: ALTA	
Iteración asignada: 5	
Programador responsable: Aníbal Benalcázar	
<p>Descripción:</p> <p>Ingresados los datos de las garantías establecidas para cada contrato existente en el "GPI".</p> <p>Se procederá a controlar las fechas de vencimientos y renovaciones de garantías y si en cada contrato se cumple estas garantías antes mencionadas.</p> <p>Uno de los requisitos para la firma de un contrato es la presentación de Garantías que abalicen dos cosas:</p> <p>El buen uso del Anticipo entregado</p> <p>El Cumplimiento a largo plazo sobre la ejecución del contrato y del uso de materiales de buena calidad.</p> <p>De acuerdo al monto del contrato el contratista presentará uno de los siguientes tipos de Garantías:</p> <p>Garantías Bancarias</p> <p>Pólizas de Seguro</p> <p>Letras de cambio (Montos pequeños)</p>	

Fuente: propia.
Tabla 9: Historia de Usuario 6

HISTORIA DE USUARIO	
Número: 7	Usuario: Tesorero
Nombre de Historia: Pagos de contratos	
Prioridad en negocio: ALTA	
Iteración asignada: 6	
Programador responsable: Aníbal Benalcázar	
<p>Descripción:</p> <p>Los pagos de cada contrato se realizarán mediante planillas, cada planilla tiene datos que son necesarios para saber las fechas de pagos.</p> <p>El número de planillas depende del tipo contrato, también se realiza los pagos de reajustes hechos a las planillas.</p> <p>Pagos:</p> <p>Anticipo</p> <p>Fecha de entrega del anticipo</p> <p>% del anticipo de acuerdo con lo establecido en el contrato.</p> <p>Pagos contra presentación de planillas periódicas debidamente aprobadas por Fiscalización.</p> <p>Fecha de presentación de planillas aprobadas.</p> <p>Valor del pago de la planilla.</p>	

Fuente: propia.
Tabla 10: Historia de Usuario 7

HISTORIA DE USUARIO	
Número: 8	Usuario: Fiscalizador
Nombre de Historia: Fiscalización de contratos	
Prioridad en negocio: ALTA	
Iteración asignada: 7	
Programador responsable: Aníbal Benalcázar	
<p>Descripción:</p> <p>Ingresar al sistema datos para fiscalización de la obra tales como:</p>	

<ul style="list-style-type: none"> • Fecha de inicio de obra • Fecha de entrega del anticipo de dinero • Fiscalizador de obra • Fecha de recepción provisional de la obra • Fecha de recepción definitiva de la obra

Fuente: propia.
Tabla 11: Historia de Usuario 8

HISTORIA DE USUARIO	
Número: 9	Usuario: Administrador, Abogado, Fiscalizador
Nombre de Historia: Reportes de contratos	
Prioridad en negocio: ALTA	
Iteración asignada: 8	
Programador responsable: Aníbal Benalcázar	
<p>Descripción:</p> <p>Lista de todos los Contratos generados en el GPI por:</p> <ul style="list-style-type: none"> • Número de Contrato • Contratista • Fecha • Ubicación • Tipo <p>Reporte de Garantías</p> <ul style="list-style-type: none"> • Número de contrato • Contratista • Aseguradora • Tipo • Fecha de Vencimiento • Fecha de Emisión <p>Reporte de Fiscalizadores y Administradores</p> <ul style="list-style-type: none"> • Lista de Contratos a Fiscalizar • Contratos con Actas de Recepción Provisional pendientes para legalización • Actas de recepción Provisional legalizadas • Contratos próximos a realizar las actas de Recepción Definitiva • Lista de Fiscalizadores 	

Fuente: propia.
Tabla 12: Historia de Usuario 9

3.1.4.- ACTIVIDADES DE INGENIERÍA

Las actividades de ingeniería no son más de requerimientos no funcionales, que necesariamente aportan para el desarrollo del proyecto y que van de la mano con las historias de usuario. Estas actividades de ingeniería generan historias técnicas no descritas en este documento ya q estarán en Anexos.

3.2.- VISIÓN DEL PRODUCTO.

3.2.1.- INTRODUCCIÓN

3.2.1.1. PROPÓSITO

El propósito de este documento es definir a alto nivel los requerimientos del Sistema de gestión de Contratos (GPI-CONTRATOS) El sistema se encargará de gestionar todos los contratos que se generan en el “Gobierno Provincial de Imbabura”. Cada contrato se registrará en el sistema. El detalle de cómo el sistema GPI-CONTRATOS cubre las necesidades de los usuarios se especifica en los casos de uso generados por las historias usuario recolectadas en cada departamento afectado, que son información adicional no especificada en este documento.

3.2.1.2. ALCANCE

Este documento de visión se aplica al sistema de gestión de contratos que será desarrollado por el tesista en coordinación con el personal de las áreas involucradas y a cargo del departamento de Sistemas del “Gobierno provincial del Imbabura”.

3.2.2.- POSICIONAMIENTO

3.2.2.1. DEFINICIÓN DEL PROBLEMA

El problema de	No poseer un sistema informático que gestione y registre los contratos. Actualmente, todos los procesos relacionados directa o indirectamente con la gestión de contratos se realizan de forma manual.
Que afecta	Al personal de los diferentes departamentos inmersos en la gestión de contratos. Los diferentes departamentos involucrados en el proceso de contratación, enfrentan serias dificultades en el acceso ágil y oportuno a la información de los diferentes

	<p>contratos.</p> <p>Las causas que dificultan el manejo adecuado de los contratos son:</p> <p>Información manejada en hojas Excel.</p> <p>Duplicidad de datos</p> <p>Falta de integridad</p> <p>Carencia de acceso rápido a Información</p>
<p>El impacto de ellos</p>	<p>Se incrementan la posibilidad de generar información errónea debido a dispersión y duplicidad de datos en los diferentes departamentos.</p> <p>Implica demasiadas actividades manuales que pueden provocar errores.</p> <p>El nivel operativo no dispone de información sobre contratos</p> <p>Se dificulta la entrega de información centralizada al nivel directivo lo que dificulta la entrega de datos veraces y únicos.</p> <p>Los diferentes departamentos implicados en La Gestión de Contratos no tienen acceso automatizado a la información de forma rápida y eficaz lo que hace la gestión de contratos se la realice de forma lenta.</p>
<p>Una solución exitosa debería</p>	<p>Implementar una solución informática de calidad que siga una metodología eficiente de desarrollo de software y solucione totalmente los requerimientos internos de los involucrados en el proceso de gestión y control de contratos.</p>

Fuente: propia.

Tabla 13: Definición del Problema

3.2.2.2. DECLARACIÓN DEL POSICIONAMIENTO DEL PROYECTO

Para	El beneficio propio de la institución y usuarios de los departamentos implicados en la gestión de contratos.
Quien(es)	Los usuarios y autoridades exigen mayor eficiencia en el proceso de gestión de contratos. Además la institución requiere automatizar los procesos de generar contratos.
El (nombre del producto)	Es una solución informática, denominada: Sistema de gestión de contratos – (GPI-CONTRATOS), q es un módulo del sistema de gestión GPI-GESTION q realiza el departamento de sistemas del Gobierno Provincial de Imbabura.
Que	Agilizará los procesos de gestión de contratos, además garantizará la validez de la información correspondientes a los diferentes contratos generados en el GPI, para que los departamentos q manejan la gestión de contratos de la institución obtenga mayor transparencia y credibilidad de las autoridades y de los clientes (ciudadanía).
Debido a que	Actualmente los procesos se realizan manualmente y no cubre con las expectativas del usuario, y se requiere estar al margen de la competencia, automatizando una de las actividades como es la gestión de contratos.
Nuestro producto	Garantiza un eficiente rendimiento, ya que será desarrollado con la nueva tecnología y cumplirá con la normatividad de la institución, aportando a su productividad y crecimiento.

Fuente: propia.

Tabla 14: Posicionamiento del proyecto

3.2.3.- DESCRIPCIÓN DE LOS INTERESADOS Y USUARIOS

Para proveer de forma efectiva los productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurar que el conjunto de participantes en el proyecto los representan adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los

problemas más importantes que éstos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que éstos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por qué estos requisitos son necesarios.

3.2.3.1. RESUMEN DE LOS INTERESADOS

Las partes interesadas son todas aquellas personas directamente involucradas en la definición y alcance de este proyecto. A continuación se presenta la lista de los interesados:

Nombre	Descripción	Responsabilidades	Rol
GPI	Gobierno provincial de Imbabura (Áreas afectadas en la gestión de Contratos)	<ul style="list-style-type: none"> Representa a todas las áreas interesadas en el desarrollo del sistema GPI-CONTRATOS. 	PRODUCT OWNER
Ing. Fernando Mino	Jefe de Sistemas GPI	<ul style="list-style-type: none"> Evaluación y aprobación del proyecto. 	SCRUM MASTER
Ing. Marcelo Jingo	Jefe de proyectos GPI	<ul style="list-style-type: none"> Responsable de controlar el desarrollo y seguimiento del Proyecto. 	SCRUM MASTER
Ing. Marco PUSDÁ	Director de Tesis	<ul style="list-style-type: none"> Responsable de controlar el desarrollo, seguimiento del Proyecto y funcionamiento de la aplicación. 	SCRUM MASTER
Responsable del proyecto	Responsable del proyecto por parte del tesista - Sr. Aníbal Benalcázar	<ul style="list-style-type: none"> Responsable del diseño, configuración, documentación y requisitos para la construcción del sistema. 	TEAM (EQUIPO)

		- Encargado del desarrollo del proyecto, de acuerdo a los prototipos que se presentaran antes del producto final.	
--	--	---	--

Fuente: propia.
Tabla 15: Resumen de los interesados

3.2.3.2. RESUMEN DE LOS USUARIOS

Los usuarios del sistema son todas aquellas personas que actúan directamente en el uso del Sistema de Gestión de Contratos, a continuación se detalla una lista de los usuarios que intervienen:

Nombre	Descripción	Responsabilidad	Rol
Administrador del sistema	Persona que administrar el sistema por parte de la institución.	<ul style="list-style-type: none"> - Gestionar las cuentas de usuarios (crear, modificar y actualizar) - Ingresar Objetos de Contratación - Ingresar Tipos de Contratos - Ingresar las plantillas de los contratos - Ingresar las garantías del contrato - Generar reportes. 	STAKEHOLDER
Secretaria (Jurídico) Secretaria (TT.HH)	Digita datos de contratos.	<ul style="list-style-type: none"> • Ingresar datos de contratistas • Ingresar datos de Fiscalizadores y Administradores de contratos. • Ingresar datos de entidades • Digita contratos. • Ingresar Garantías 	STAKEHOLDER

Abogado (Jurídico)	Digita datos de contratos.	<ul style="list-style-type: none"> • Ingresar datos de contratistas • Ingresar datos de Fiscalizadores y Administradores de contratos. • Ingresar datos de entidades • Digitar contratos. • Ingresar Garantías 	STAKEHOLDER
Tesorera (Financiero)	Realiza Pagos de los contratos.	<ul style="list-style-type: none"> • Gestiona Pagos de los contratos • Gestiona Garantías 	STAKEHOLDER
Fiscalizador(Planificación)	Fiscaliza los Contratos.	<ul style="list-style-type: none"> • Ingresar datos de la fiscalización de un contrato. 	STAKEHOLDER

Fuente: propia.

Tabla 16: Resumen de los usuarios

3.2.3.3. ENTORNO DE USUARIO

Los usuarios podrán acceder a los servicios que prestará el sistema GPI-CONTRATOS con su nombre, contraseña de usuario y su respectivo rol asignado por el sistema GPI-SISTEMA otro modulo del sistema de gestión del “GOBIERNO PROVINCIAL DE IMABURA” GPI-GESTIÓN.

El Sistema será diseñado en la herramienta FLEX y con la base de datos PostgreSQL

El usuario final podrá acceder al sistema desde un navegador de Internet (browser), en cualquier terminal que se encuentre conectado a la intranet el uso de esta herramienta informática beneficiará a los departamentos inmersos en la gestión de contratos.

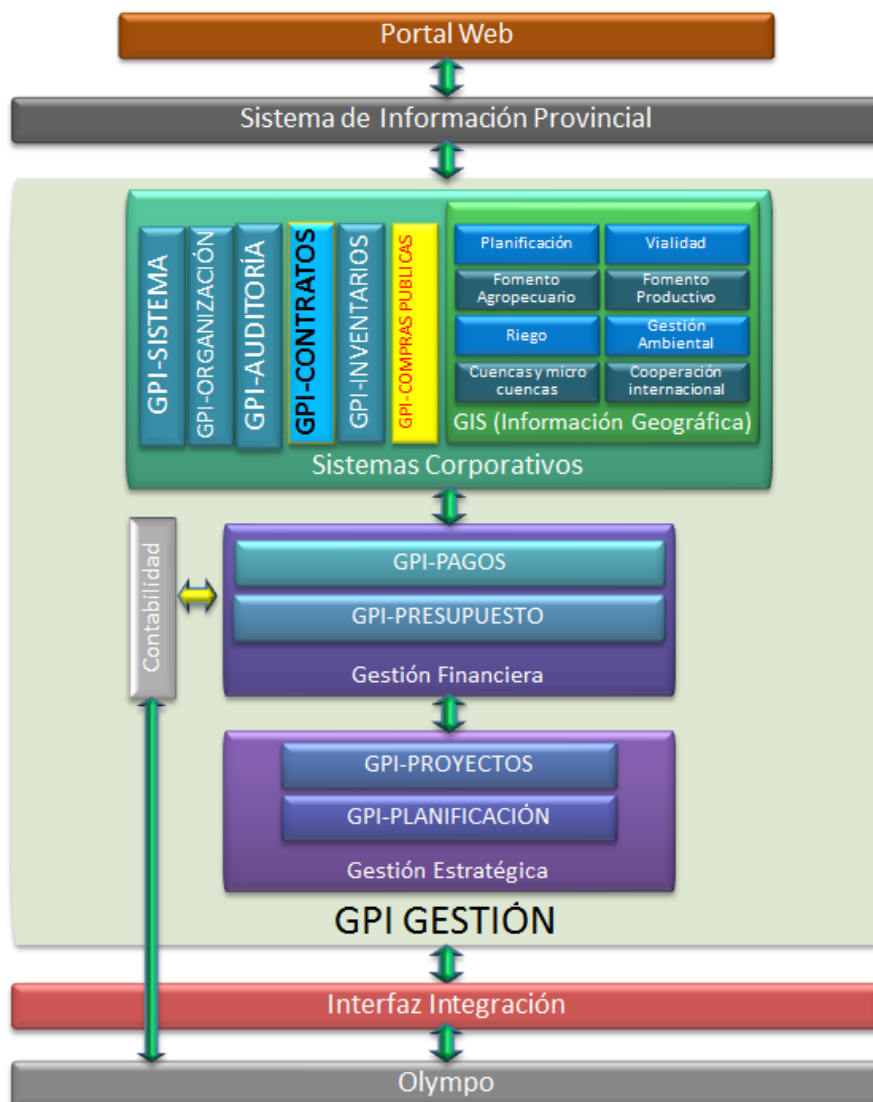
3.2.4.- VISTA GENERAL DEL PRODUCTO

Esta sección provee información a alto nivel de las funciones del sistema a implantar.

3.2.4.1. PERSPECTIVA DEL PRODUCTO

El sistema GPI-CONTRATOS se ejecuta a través del servidor web Apache y los usuarios pueden acceder, por medio de un navegador web, instalado en su máquina, la cual puede tener los sistemas operativos Linux o Windows. Además, y tiene una base de datos centralizada en PostgreSQL. El sistema GPI-CONTRATOS es un módulo del sistema GPI-GESTIÓN, lo cual depende de otros módulos del sistema.

Este sistema se acoplará a los diferentes módulos del Sistema Integrado de gestión del Gobierno Provincial de Imbabura “GPI-GESTIÓN” como se muestra en el siguiente gráfico.



Fuente: GPI-GESTIÓN
Ilustración 16: MODELO GPI-GESTIÓN

3.2.4.2. NECESIDADES Y CARACTERÍSTICAS

Necesidad del cliente	Prioridad	Características del sistema que cubrirán esta necesidad	Entrega planificada
<p>Previo a la selección de un contrato dentro del GPI se debe tener un sistema que permita ingresar todos los datos de dicho contrato para llevar un control de todos los contratos que se generan en el GPI.</p>	Alta	<p>El sistema GPI-CONTRATOS contempla:</p> <ul style="list-style-type: none"> • módulo de ingreso de plantillas de contratos • módulo de ingreso de contratos. 	<p>De acuerdo a las especificaciones por parte del cliente y siguiendo con el programa de trabajo establecido en el documento plan de desarrollo de software (detallado más adelante).</p>
<p>Automatizar la gestión de garantías de los contratos</p>		<p>El sistema GPI-CONTRATOS contempla:</p> <ul style="list-style-type: none"> • módulo gestión garantías. 	<p>De acuerdo a las especificaciones por parte del cliente y siguiendo con el programa de trabajo establecido en el documento plan de desarrollo de software (detallado más adelante).</p>
<p>Automatización de planillas de recepción del avance de la obra.</p>	Alta	<p>El sistema GPI-CONTRATOS contempla:</p> <ul style="list-style-type: none"> • módulo fiscalización de obras. 	<p>De acuerdo a las especificaciones por parte del cliente y siguiendo con el programa de trabajo establecido en el documento plan de desarrollo de</p>

			software (detallado más adelante).
Automatizar el pago de planillas y ver las distintas fechas que se presentan en la gestión de un contrato.	Alta	El sistema GPI-CONTRATOS contempla: <ul style="list-style-type: none"> • módulo gestión de pagos de contratos. 	De acuerdo a las especificaciones por parte del cliente y siguiendo con el programa de trabajo establecido en el documento plan de desarrollo de software (detallado más adelante).
Automatizar el registro y control de contratos de personal	Alta	El sistema GPI-CONTRATOS contempla: <ul style="list-style-type: none"> • módulo de ingreso de plantillas de contratos • módulo de ingreso de contratos. 	De acuerdo a las especificaciones por parte del cliente y siguiendo con el programa de trabajo establecido en el documento plan de desarrollo de software (detallado más adelante).

Fuente: propia.

Tabla 17: Necesidades y características

3.2.4.3. ALTERNATIVAS Y SISTEMAS DE LA COMPETENCIA

3.2.4.3.1. ADQUIRIR SISTEMA EXTERNO

Se podría buscar alternativas externas para solucionar los diversos requerimientos, pero en la actualidad no existen herramientas en el mercado que cumpla totalmente con las expectativas, puesto que al tratarse de una institución dedicada a la educación y producción se necesita de un sistema personalizado.

3.2.5.- DESCRIPCIÓN GLOBAL DEL PRODUCTO

3.2.5.1. CONEXIÓN FÁCIL Y DINÁMICA.

El usuario sólo introduce el nombre y contraseña de su cuenta el rol que desempeña este usuario se mostrará de una forma dinámica.

3.2.5.2. BASE DE DATOS CENTRALIZADA.

Ya que el Sistema forma parte del Sistema de Gestión GPI-GESTIÓN y hacen uso de la misma base de datos esto permite guardar la información en un solo punto central.

3.2.6.- RANGOS DE CALIDAD

La implementación del sistema se realizará bajo los parámetros de calidad de la Metodología de Desarrollo ágil como lo es SCRUM.

3.2.7.- OTROS REQUERIMIENTOS DEL SISTEMA

3.2.7.1. SISTEMA MULTIPLATAFORMA.

Se utiliza una tecnología web, por la cual se puede utilizar cualquier navegador web que soporte DHTML, CSS, JavaScript, y tener instalado flash player q es compatible con todos los navegadores del mercado.

3.2.7.2. ACCESO A INTERNET.

Los usuarios que deseen accede al sistema fuera de la institución deberán tener acceso al internet.

3.2.7.3. SERVIDOR WEB.

El servidor web a utilizarse es Apache que estará instalado el soporte para la base de datos PostgreSQL y el framework zend php.

3.2.7.4. PROTOCOLOS DE COMUNICACIONES.

El protocolo para la comunicación entre el servidor y el Cliente es el HTTP.

3.2.7.5. PLATAFORMA DE REDES.

Basado en el protocolo TCP/IP.

3.2.7.6. CONFIGURACIONES DE EQUIPOS SERVIDORES.

El GPI proveerá de la infraestructura necesaria para el desarrollo y producción que requiere el Sistema.

3.3.- METODOLOGÍA DE TRABAJO INTERNO.

3.3.1.- INTRODUCCIÓN

La metodología de trabajo interno, es una versión preliminar preparada para ser incluida en la propuesta elaborada como respuesta al Sistema de Gestión de Contratos para el Gobierno provincial de Imbabura. Este documento provee una visión global del enfoque de desarrollo propuesto.

Para el proyecto utilizaremos metodología SCRUM. Se incluirá el detalle para las fases de Planificación de la iteración, Ejecución de la iteración y adicionalmente se esbozarán las fases posteriores de Inspección y Adaptación para dar una visión global de todo el proceso.

El enfoque de desarrollo propuesto constituye una configuración del proceso SCRUM de acuerdo a las características del proyecto, seleccionando los roles de los participantes, las actividades a realizar y los artefactos (entregables) que serán generados.

3.3.1.1. PROPÓSITO

El propósito del documento es proporcionar la información necesaria para controlar el proyecto por parte de los jefes de cada equipo. En él se describe el enfoque de desarrollo del software.

3.3.1.2. ALCANCE

El documento de metodología interna de trabajo describe el método global usado para el desarrollo del Sistema de Gestión "GPI-GESTIÓN", que abarca el módulo de Gestión de Contratos "GPI-CONTRATOS". 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 para la planificación de las iteraciones.

Este plan está basado en la captura de requisitos por medio de entrevistas con el stakeholders de las áreas implicadas en la Gestión de Contratos del Gobierno Provincial de Imbabura, para hacer una estimación aproximada, una vez comenzado el proyecto y

durante la fase de “Planificación de la Iteración” se generará la primera versión del artefacto “Visión”, el cual se utilizará para afinar 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.

3.3.2.- VISTA GENERAL DEL PROYECTO

3.3.2.1. FUNDAMENTACIÓN DE LA METODOLOGÍA

El presente proyecto se realizará bajo una metodología de desarrollo SCRUM, por los siguientes motivos:

- En lugar de hacer todo de una cosa a la vez, los equipos SCRUM hacen un poco de todo... todo el tiempo.
- Es un modo de desarrollo adaptable, antes que predictivo. Es decir que se puede tomar decisiones para resolver problemas sobre la marcha y adaptar la manera de trabajo según demande el trabajo a realizar.
- Emplea el modelo de construcción incremental basado en iteraciones y revisiones.
- Entrega de un producto funcional al finalizar cada Sprint.
- Posibilidad de ajustar la funcionalidad en base a la necesidad de negocio del cliente.
- Alcance acotado y viable.
- Equipos integrados y comprometidos con el proyecto, se auto-administran.

3.3.2.2. SUPOSICIONES Y RESTRICCIONES

Las suposiciones y restricciones respecto del Sistema de Gestión de Contratos del Gobierno Provincial de Imbabura que se derivan directamente de las entrevistas con el stakeholders de las diferentes áreas son:

El proyecto está completamente financiado por el Gobierno Provincial de Imbabura y no habrá inconvenientes relacionados al costo total del proyecto ni a la agilidad con la que se deben atender los desembolsos parciales del mismo.

El sistema será diseñado sobre plataforma WEB y cumplirá con los estándares de calidad vigentes para desarrollo de software. Esto se conseguirá cumpliendo con el estándar PMI para dirección de proyectos, metodología SCRUM para el proceso de ingeniería de software y herramientas de ADOBE para la construcción de las aplicaciones.

3.3.2.3. VALORES DEL TRABAJO

Los valores que deben ser practicados por todos los miembros involucrados en el desarrollo y que hacen posible que la metodología SCRUM tenga éxito son:

- Autonomía del equipo
- Respeto en el equipo
- Responsabilidad y auto-disciplina
- Foco en la tarea
- Información transparencia y visibilidad

3.3.2.4. ENTREGABLES DEL TRABAJO

Es preciso destacar que de acuerdo a la filosofía de SCRUM, se tienen documentos de trabajo interno a lo largo del proceso de desarrollo. Los cuales son los siguientes:

Documentos

- Product Backlog
- Sprint Backlog

3.3.2.4.1. PRODUCT BACKLOG

La pila del producto es el inventario de funcionalidades, mejoras, tecnología y corrección de errores que deben incorporarse al producto a través de las sucesivas iteraciones de desarrollo.

En el caso del presente proyecto es la lista de documentos y actividades que se realizarán durante el desarrollo del proyecto.

Representa todo aquello que esperan los clientes, usuarios, y en general los interesados. Todo lo que suponga un trabajo que debe realizar el equipo tiene que estar reflejado en esta pila.

A diferencia de un documento de requisitos del sistema, la pila del producto nunca se da por completada; está en continuo crecimiento y evolución.

Habitualmente se comienza a elaborar con el resultado de una reunión de "fertilización cruzada" o brainstorming; donde colabora todo el equipo a partir de la visión del propietario del producto y basado en las primeras entrevistas con el cliente.

3.3.2.4.2. FORMATO DEL PRODUCT BACKLOG (PILA DEL PRODUCTO)

La pila del producto no es un documento de requisitos, sino una herramienta de referencia para el equipo.

Si se emplea formato de lista, es recomendable que al menos incluya la siguiente información en cada línea:

- Identificador único de la funcionalidad o trabajo.
- Descripción de la funcionalidad.
- Campo o sistema de priorización.
- Estimación (tiempo y costo)

Dependiendo del tipo de proyecto, funcionamiento del equipo y la organización, pueden resultar aconsejables otros campos:

- Observaciones
- Criterio de validación
- Equipo asignado

Es preferible no adoptar ningún protocolo de trabajo de forma rígida. El formato del Product Backlog no es cerrado. Aquí tenemos un ejemplo:

ID#	MODULO	SPRINT	TAREA	ESTIMACION (días)	RESPONSABLE

Fuente: Propia
Ilustración 17: Formato del Product Backlog

3.3.2.4.3. SPRINT BACKLOG

La pila del sprint, (Sprint Backlog en inglés) es la lista que descompone las funcionalidades de la pila del producto en las tareas necesarias para construir un incremento: una parte completa del producto.

La realiza cada equipo durante la reunión de planificación del sprint, asignando cada tarea a una persona e indicando en la misma lista cuánto tiempo falta aún para que la termine.

Es útil, porque descompone el proyecto en unidades de tamaño adecuado para determinar el avance a diario e identificar riesgos y problemas sin necesidad de procesos complejos de gestión.

Es también una herramienta de soporte para la comunicación directa del equipo y entre equipos.

Condiciones

- Realizada de forma conjunta por todos los miembros del equipo.
- Cubre todas las tareas identificadas por el equipo para conseguir el objetivo del sprint.
- Sólo el equipo lo puede modificar durante el sprint.
- Es visible para todo el equipo. Idealmente en un documento online.

Formato y soporte

Tres son las opciones: Hoja de cálculo, Pizarra física o pared, Herramienta colaborativa o de gestión de proyectos.

Y sobre la que mejor se adecúa a las características del proyecto, oficina y equipo, lo apropiado es diseñar el formato más cómodo para todos, teniendo en cuenta los siguientes criterios:

- Incluye la información: lista de tareas, persona responsable de cada una, estado en el que se encuentra y tiempo de trabajo que queda para completarla.
- Sólo incluye la información estrictamente necesaria.
- El medio y modelo elegido es la opción posible que más facilita la consulta y comunicación diaria y directa del equipo.
- Sirve de soporte para registrar en cada reunión diaria del sprint, el tiempo que le queda a cada tarea.

con su carga de trabajo cada vez que se tenga un documento de la fase previa terminado.

Se continuara el ciclo hasta que todas las actividades constadas en el Product Backlog estén completas.

Durante el proceso el jefe de proyecto aplicara el diagrama burn up para medir el nivel de avance que tenga el equipo.

3.4.- PRODUCT BACKLOG

PRODUCT BACKLOG					
ID	MÓDULO	SPRINT	TAREA	ESTIMACIÓN	RESPONSABLE
0	NINGUNO	0	Instalación de Software y Levantamiento de requerimie	10	Anibal Benalcázar
1	AS	1	Autenticación de Usuarios en el Sistema	10	Anibal Benalcázar
2	AS	2	Creación y Modificación de Objetos de Contratación	10	Anibal Benalcázar
3	AS	3	Creación y Modificación de Tipos de Contratos	10	Anibal Benalcázar
4	PC	4	Creación y Modificación Plantilla de Contrato	10	Anibal Benalcázar
5	CC	5	Ingreso y Modificación Contratos	10	Anibal Benalcázar
6	GC	6	Control de Garantías	10	Anibal Benalcázar
7	PC	7	Ingreso Pagos de Contratos	10	Anibal Benalcázar
8	FC	8	Fiscalización de Contratos	10	Anibal Benalcázar
9	RS	9	Reportes del Sistema	10	Anibal Benalcázar

Fuente: propia.

Tabla 19: Product Backlog GPI-CONTRATOS







Los módulos son los siguientes:

- Administración del Sistema
- Contratos
- Garantías de Contratos
- Pagos de Contratos
- Fiscalización de Contratos
- Reportes del Sistema

4.2.- MODELO DE CASOS DE USO

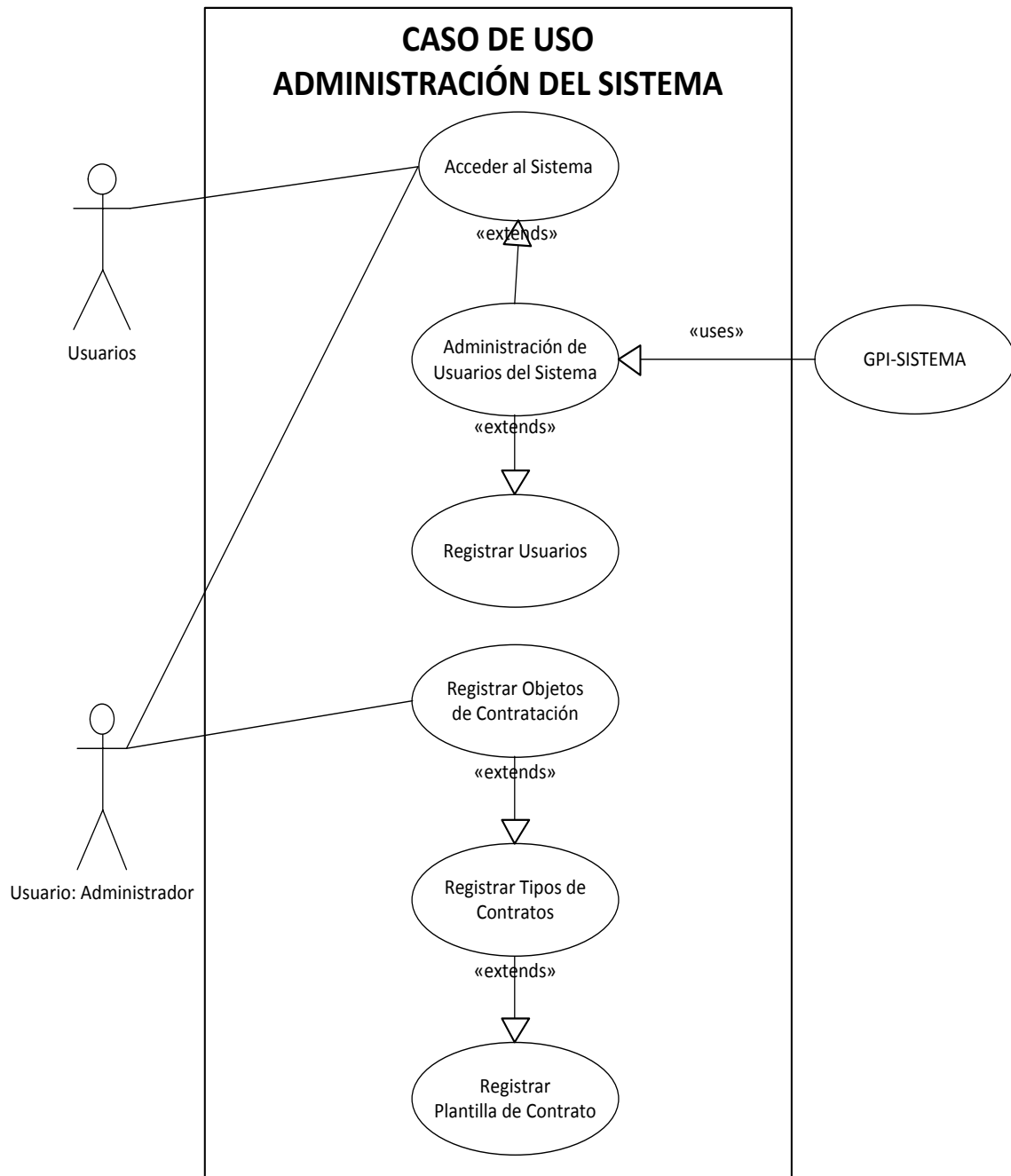
El modelo de casos de uso presenta las funciones del sistema y los actores que se hacen uso de ellas. Se presentan mediante casos de uso. Los casos de uso se generan con la información del Product Backlog.

4.2.1.- ACTORES

ACTORES	STAKEHOLDER	DESCRIPCION
 Usuario: Administrador	Administración del Sistema	Rol encargado de Gestionar el ingreso de usuarios, objetos de contratación, tipos de contratos, planillas de los contratos.
 Usuario: Abogado (Juridico)	Abogado Departamento Jurídico	Rol encargado de ingresar los datos de un contrato, según plantilla de contrato ingresada por el sistema: Contratos de Bienes y Servicios, Obra, Consultoría.
 Usuario: Secretaria (Juridico)	Secretaria Departamento Jurídico	Rol encargado de ingresar los datos de un contrato, según plantilla de contrato ingresada por el sistema: Contratos de Bienes y Servicios, Obra, Consultoría.
 Usuario: Secreataria (Talento Humano)	Secretaria Departamento Talento Humano	Rol encargado de ingresar los datos de un contrato, según plantilla de contrato ingresada por el sistema: Contratos de TT.HH
 Usuario: Tesorero	Tesorero Departamento Financiero	Rol encargado de registrar el pago de los contratos y controlar el pago de los contratos.
 Usuario: Fiscalizador	Fiscalizador Departamento Fiscalización	Rol encargado de fiscalizar un contrato, verificación de las fechas de avances de obra.

Fuente: propia.

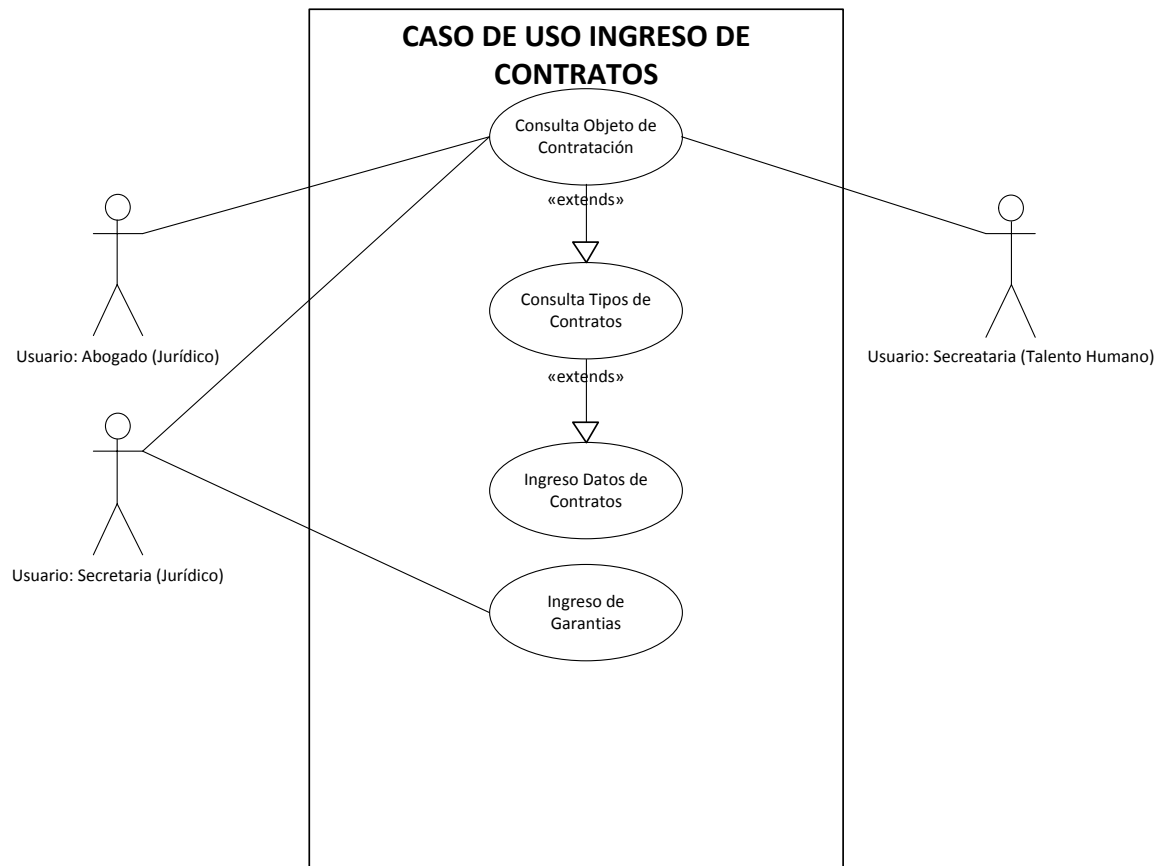
4.2.2.- CASO DE USO ADMINISTRACIÓN DEL SISTEMA



Fuente: propia.

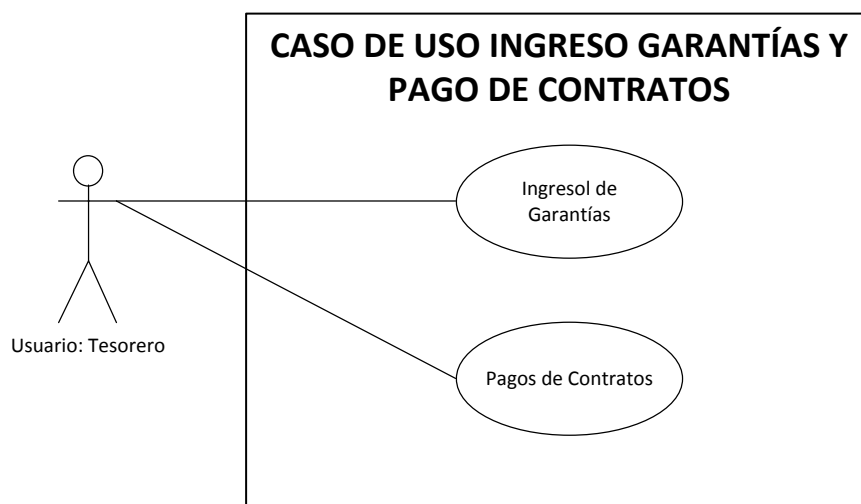
Ilustración 18: Caso de uso Administración del Sistema

4.2.3.- CASO DE USO INGRESO DE CONTRATOS



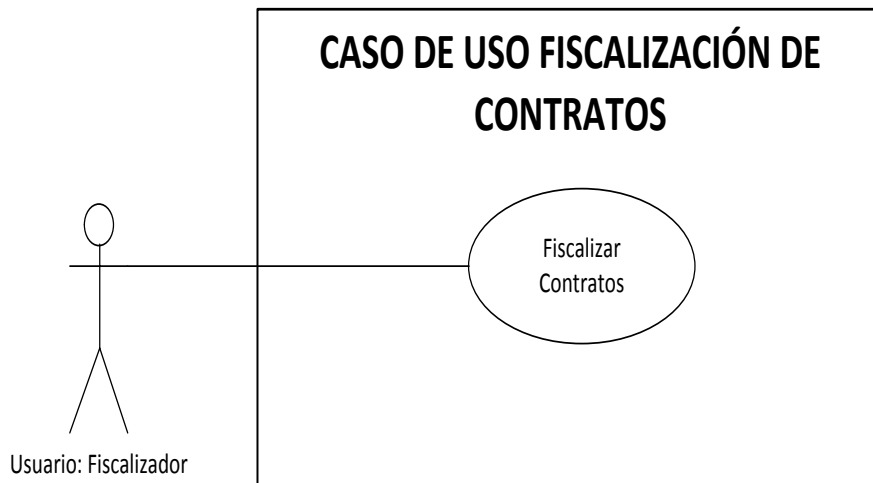
Fuente: propia.
Ilustración 19: Caso de Uso Ingreso de Contratos

4.2.4.- CASO DE USO GARANTÍAS y PAGOS DE CONTRATOS



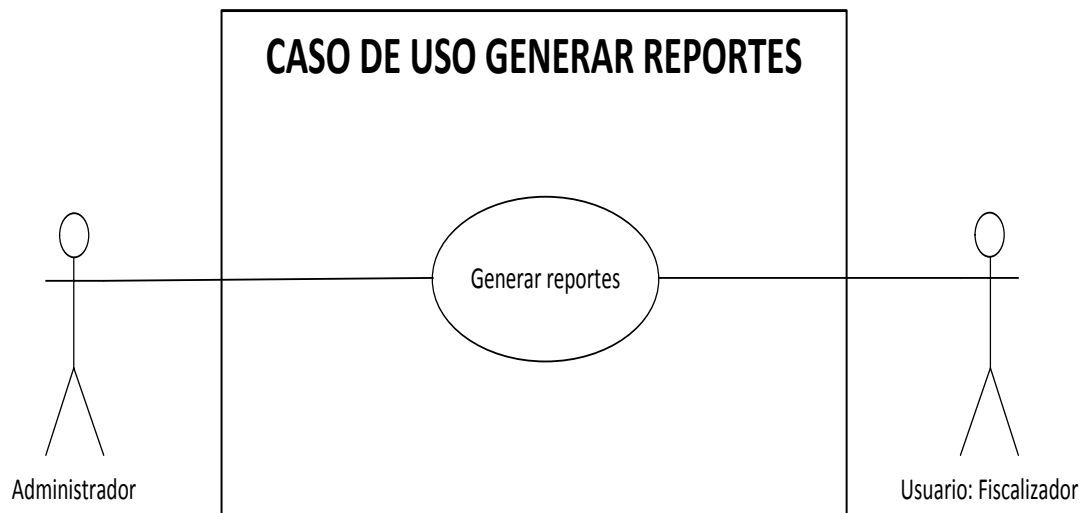
Fuente: propia.
Ilustración 20: Caso de Uso Ingreso de Garantías y Pagos de Contratos

4.2.5.- CASO DE USO FISCALIZACIÓN DE CONTRATO



Fuente: propia.
Ilustración 21: Caso de Uso Fiscalizar Contrato


4.2.6.- CASO DE USO REPORTES DEL SISTEMA



Fuente: propia.
Ilustración 22: Caso de Uso Generar Reportes

4.3.- ESPECIFICACIÓN DE LOS CASOS DE USO

4.3.1.- ESPECIFICACIÓN DEL CASO DE USO: ACCEDER AL SISTEMA

Referencia	Caso de Uso Acceder al Sistema
GPICON-CU-001	<p>1. Descripción Breve</p> <p>El caso de uso describe el ingreso de un usuario al sistema según los permisos asignados por el Modulo de GPI-SISTEMA.</p> <p>Los usuarios corresponden a usuarios propios del sistema.</p> <p>Cada usuario tiene sus privilegios específicos por lo que es necesario que este activo en el sistema. En las contraseñas de usuarios se usan el algoritmo de encriptación MD5 para su seguridad.</p> <p>2. Flujo Básico de Eventos</p> <p>A. El usuario accede a través de un browser (cualquier browser que use flash player) a la dirección de acceso al sistema (ilustración #). La URL de acceso es:</p>  <p>Fuente: propia.</p> <p>Ilustración 23: Dirección de acceso al sistema</p> <p>B. Se presenta al usuario la siguiente pantalla de ingreso al sistema en la cual se solicita la cuenta de usuario y su contraseña.</p>



Fuente: propia.

Ilustración 24: Pantalla de ingreso al sistema GPI-GESTIÓN

- C. Una vez ingresado su cuenta de usuario y contraseña se debe seleccionar la opción “Ingresar” y si los datos están correctos se accede al sistema donde se muestra la pantalla del sistema.



Fuente: propia.

Ilustración 25: Pantalla Principal GPI-CONTRATOS

- D. La parte superior de la pantalla se encuentra el panel horizontal del sistema donde se encuentran las opciones de usuario, ayuda, cerrar la sesión del usuario y modo pantalla completa.

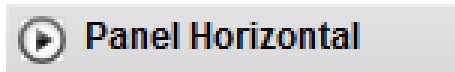


Fuente: propia.

Ilustración 26: Panel Cabecera GPI-CONTRATOS

Colapsar panel

Este panel tiene la opción de colapsar para que el ambiente de trabajo del usuario tenga más espacio de visibilidad.



Usuario

Muestra la información del usuario q entro al sistema.



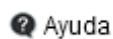
Cerrar

Cierra la sesión del usuario



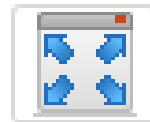
Ayuda

Muestra información del sistema

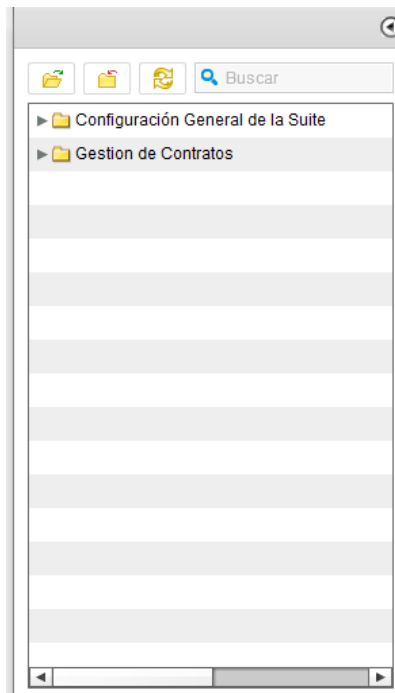


Pantalla completa

Muestra la pantalla completa del sistema.



- E. En la parte izquierda de la pantalla se muestra el menú del sistema GPI-CONTRATOS donde se cargaran los módulos del sistema de Gestión de Contratos.



Fuente: propia.
Ilustración 27: Menú Principal GPI-CONTRATOS

Administración del sistema	Usuarios	
	Ingresar Usuarios	
	Actualizar Usuarios	
Gestión de Contratos	Objetos de Contratación	
	Tipos de Contratos	
	Ingresar Plantilla de Contrato	
	Ingreso de Contratos	
	Buscar Contratos	
	Reportes:	Listar Contratos
Garantías	Ingreso de Garantías	
	Reportes:	Listar Garantías
Pagos de Contratos	Ingreso de Pagos	
	Reportes:	Listar Pagos

Fuente: propia.
Tabla 31: Módulos de Carga en Menú Principal

3. Flujos Alternativos

3.1. Error de contraseña o usuario

En el punto C, dice q si el usuario ingresa de manera errónea su cuenta de usuario o contraseña el sistema se encarga de mostrar un mensaje de error de validación de credenciales.



Fuente: propia.

Ilustración 28: Mensaje de error inicio del sistema

3.2. Salida del Sistema

Se puede salir del sistema con la opción de cerrar sesión, cerrando la sesión actual del usuario.

4. Precondiciones

El usuario para acceder al sistema debe tener una cuenta de usuario ingresada en la base de datos del sistema, en caso de no tenerla debe solicitar al administrador del sistema. A demás se requiere que el sistema y la base de datos se encuentren instalados y configurados en un servidor de aplicaciones.

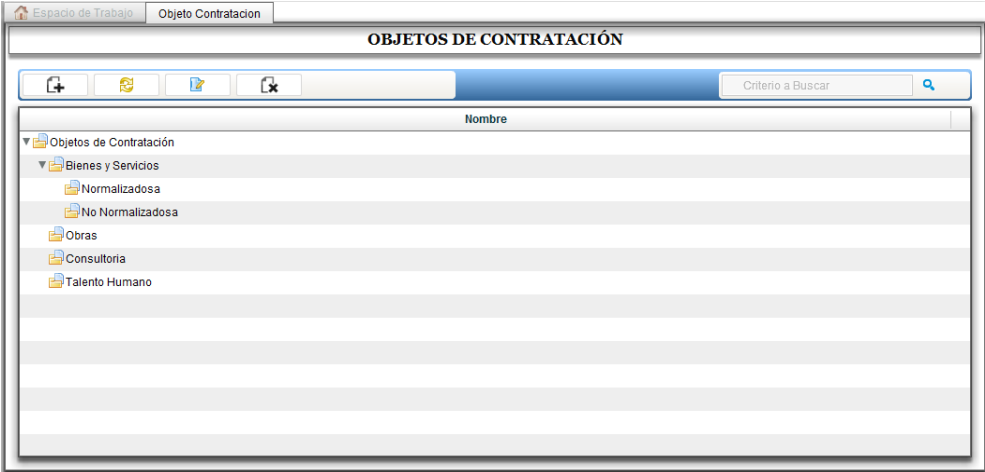
5. Post-Condiciones

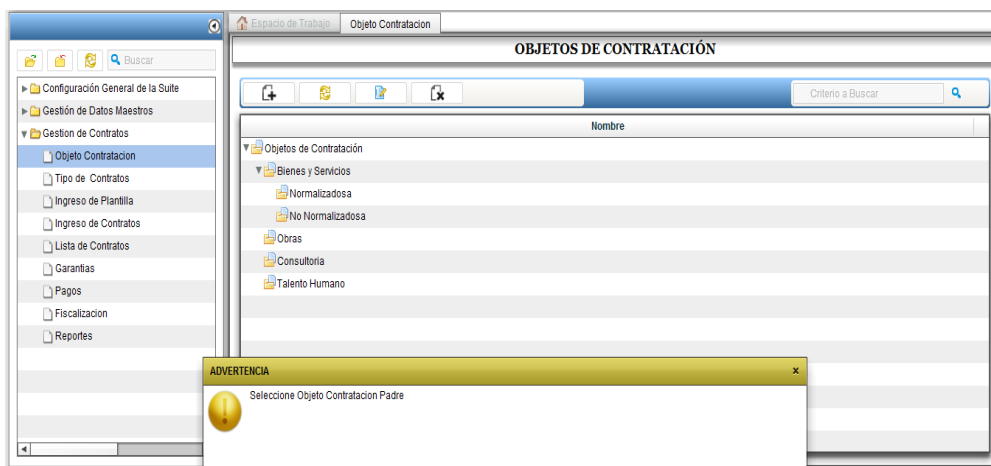
El sistema se conecta mediante la sesión del usuario y se asigna los módulos y privilegios asignados por el Modulo GPI-SISTEMA.

Fuente: propia.

Tabla 32: Especificación de Caso de Uso Acceder al Sistema

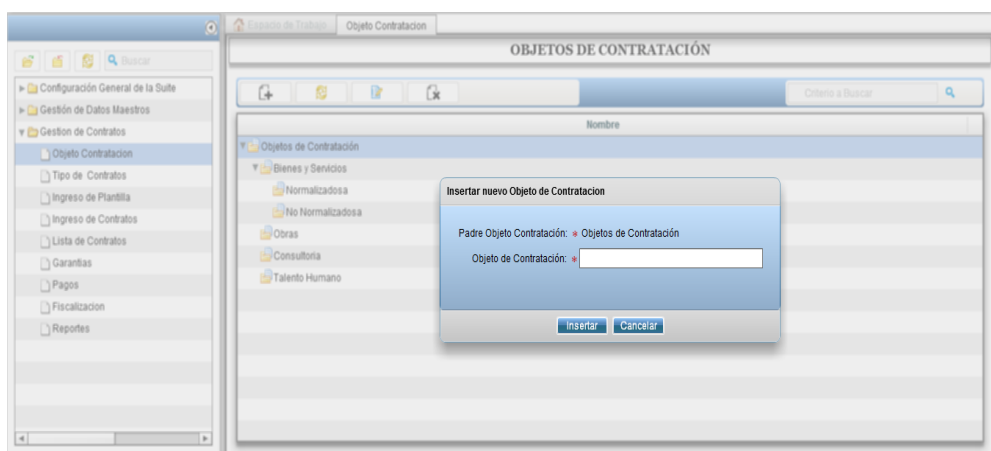
4.3.2.- ESPECIFICACIÓN DEL CASO DE USO: REGISTRAR OBJETOS DE CONTRATACIÓN

Referencia	Caso de Uso Registrar Objetos de Contratación
GPICON-CU-002	<p>1. Descripción Breve</p> <p>El caso de uso describe el registro o modificación de los datos de los objetos de contratación, estos datos son muy importantes para el ingreso de contratos y creación de las plantillas de contratos.</p> <p>2. Flujo Básico de Eventos</p> <p>A. El usuario ingresa al sistema</p> <p>B. El usuario va al menú en el panel Vertical donde estará el módulo de Objetos de Contratación dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Objetos de Contratación aquí se presenta un menú donde puede hacer un crud de la tabla de objetos de Contratación.</p> <div data-bbox="375 1142 1364 1612" data-label="Image"></div> <p>Fuente: propia.</p> <p>Ilustración 29: Pantalla ingreso objeto de contratación</p> <p>C. Para ingresar un nuevo Objeto de Contratación seleccionar Nuevo, le aparece la opción se debe seleccionar el padre del Objeto de Contratación si no se selecciona el padre no se podrá ingresar la información.</p>



Fuente: propia.
Ilustración 30: Mensaje Seleccionar padre objeto de contratación

D. Seleccionado el padre del objeto de contratación a ingresar sale una pantalla de ingreso de nuevo Objeto de Contratación.



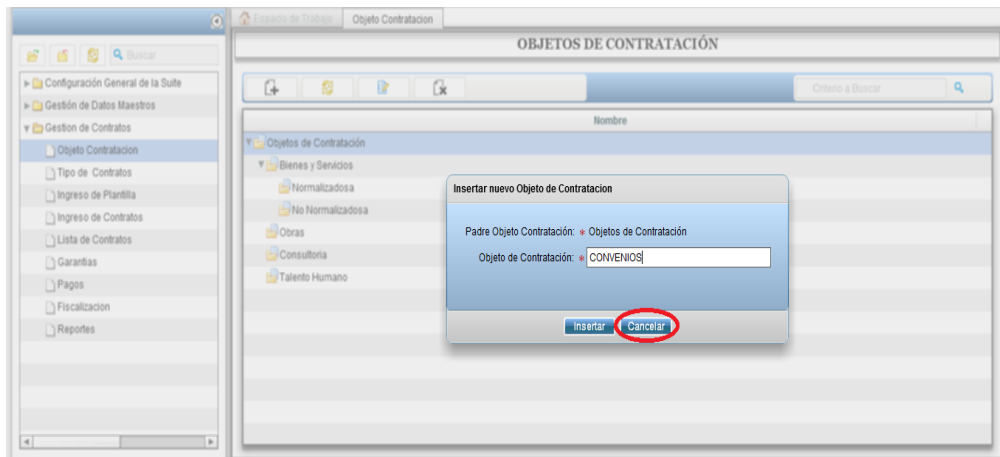
Fuente: propia.
Ilustración 31: Pantalla ingresar nuevo objeto de contratación

E. El usuario digita los datos indicados y presiona el botón “Ingresar”

3. Flujos Alternativos

3.1. Cancelar Ingreso de Objeto de contratación

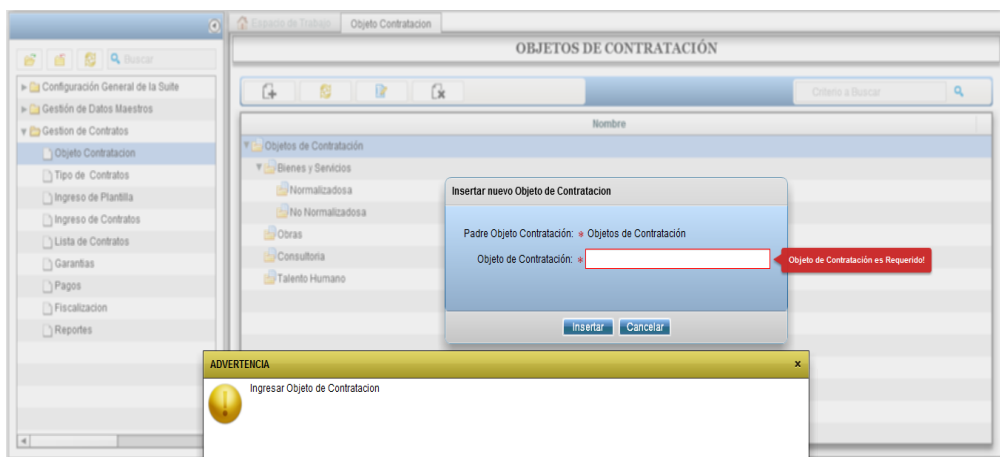
Si el usuario decide cancelar el ingreso del Objeto de Contratación debe presionar el botón cancelar y saldrá de la pantalla de ingreso.



Fuente: propia.

Ilustración 32: Pantalla cancelar ingreso objeto contratación

3.2. Error en el Ingreso de Objeto de Contratación si no se ingresa un nombre saldrá el error de validación



Fuente: propia.

Ilustración 33: Mensaje Objeto Contratación es requerido

4. Precondiciones

Se requiere previamente ingresar al sistema.

5. Post-condiciones

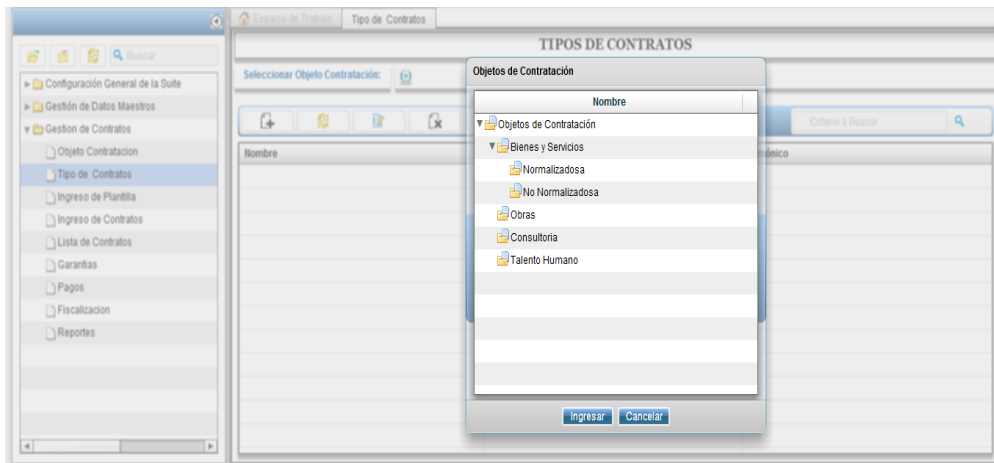
El ingreso de objetos de contratación es muy importante sin objetos de contratación no se puede ingresar los tipos de contratos.

Fuente: propia.

Tabla 33: Especificación de Caso Registrar Objetos de contratación

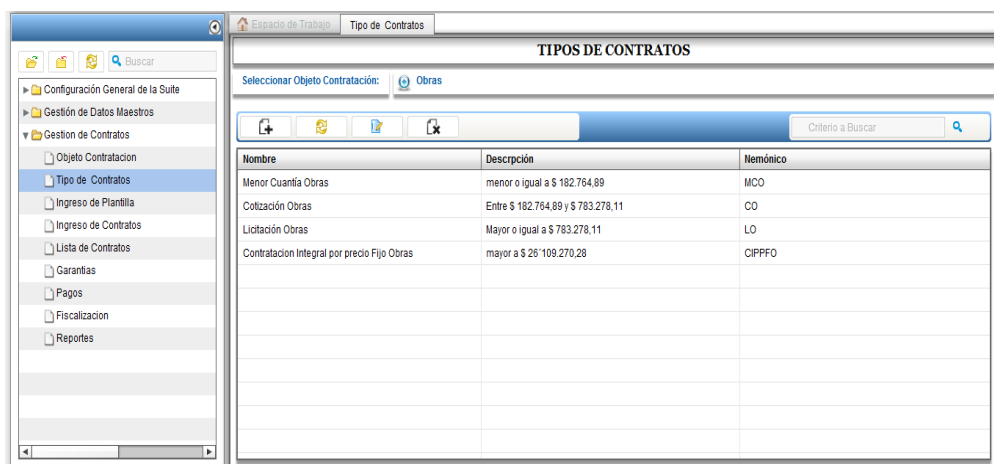
4.3.3.- ESPECIFICACIÓN DEL CASO DE USO: REGISTRAR TIPOS DE CONTRATOS

Referencia	Caso de Uso Registrar Tipos de Contratos
GPICON-CU-003	<p>1. Descripción Breve</p> <p>El caso de uso describe el registro o modificación de los datos de los Tipos de Contratos, estos datos son muy importantes para el ingreso de contratos y creación de las plantillas de contratos.</p> <p>2. Flujo Básico de Eventos</p> <p>A. El usuario ingresa al sistema</p> <p>B. El usuario va al menú en el panel Vertical donde estará el módulo de Tipos de Contratos dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Tipos de Contratos, aquí se presenta un menú donde puede hacer un crud de la tabla de Tipos de Contratos.</p> <div data-bbox="405 1249 1401 1704" data-label="Image"> </div> <p>Fuente: propia.</p> <p>Ilustración 34: Pantalla Ingreso Tipos de Contratos</p> <p>C. Para ver los Tipos de Contratos ya ingresados o ingresar uno nuevo debemos de seleccionar el objeto de contratación del contrato.</p>



Fuente: propia.
 Ilustración 35: Pantalla seleccionar objeto de contratación

- D. Seleccionado el objeto de Contratación se despliega una lista de los Objetos de Contratación.
- E. Para ingresar se debe escoger un objeto de contratación, lo que hace esto es filtrar los tipos de contratos según el objeto de contratación perteneciente a ese tipo de contrato.



Fuente: propia.
 Ilustración 36: Pantalla tipos de contratos

F. Para ingresar un nuevo Tipo de Contrato, ya filtrado los tipos de contratos por objeto de contratación se procede a dar click en el botón Nuevo donde aparecerá una pantalla para ingresar el nuevo tipo de contrato.



Fuente: propia.

Ilustración 37: Pantalla ingresar tipo de contrato

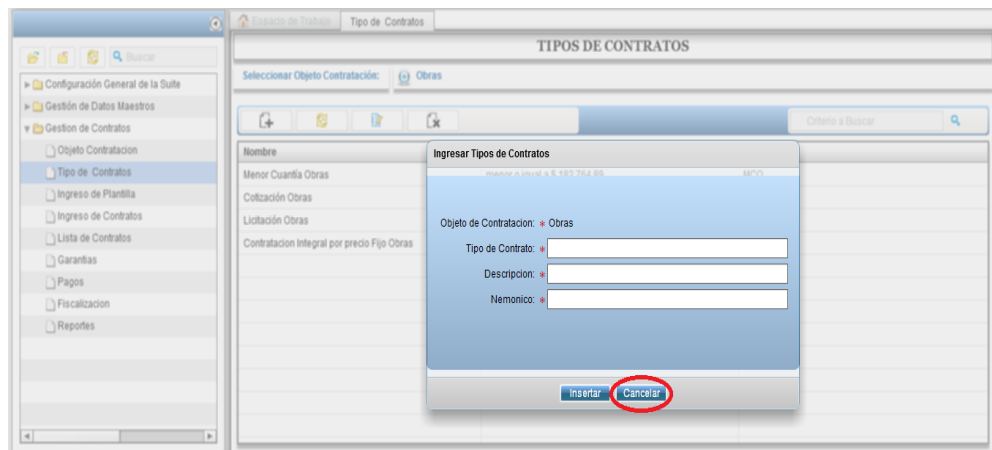
G. Se ingresa la información solicitada en el formulario de Agregar nuevo Contrato Tipo.

H. Para ingresar los datos en la base de datos se debe dar click en el botón Ingresar.

3. Flujos Alternativos

3.1. Cancelar Ingreso de Tipos de Contratos

Si el usuario decide cancelar el ingreso del Tipo de Contrato debe presionar el botón cancelar y saldrá de la pantalla de ingreso.

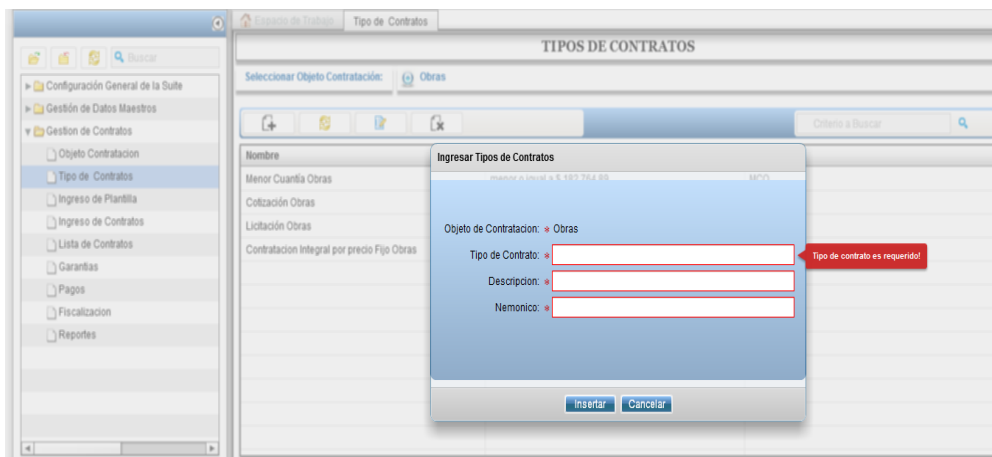


Fuente: propia.

Ilustración 38: Pantalla cancelar ingreso tipo de contrato

3.2. Error en el ingreso de información de Tipos de Contratos

Si la información de tipos de contratos no es correcta saldrá un error de validación de los campos incompletos, si un campo está incompleto saldrá el error y no dejara ingresar el Tipo de Contrato.



Fuente: propia.

Ilustración 39: Error ingreso de información tipos de contratos


4. Precondiciones

Se requiere ingresar previamente al sistema

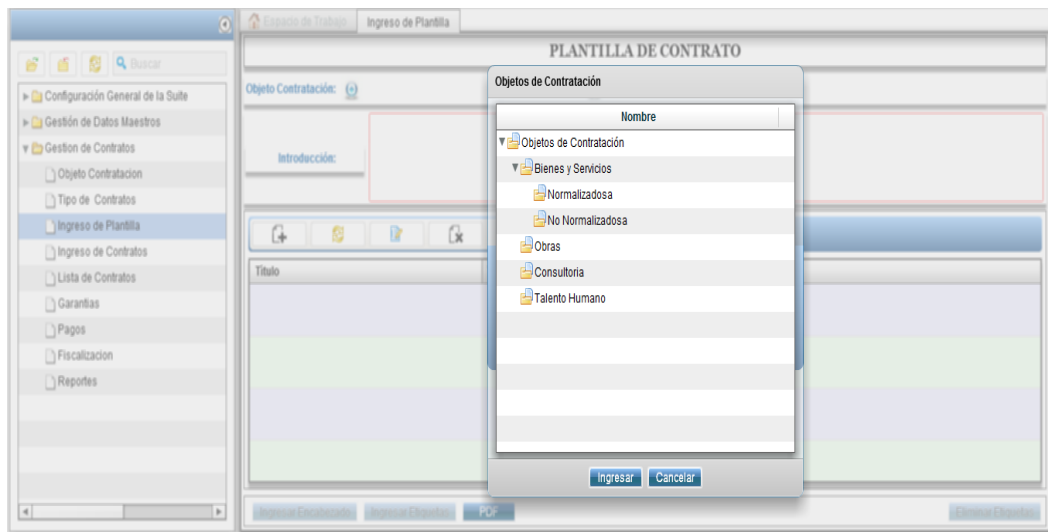
	<p>5. Post-Condiciones</p> <p>El ingreso de los tipos de contratos está ligado al objeto de contratación, si no se tiene estos datos no se puede ingresar la plantilla de contrato.</p>
--	---

Fuente: propia.
 Tabla 34 Especificación de casos de uso Registrar tipos de contratos

4.3.4.- ESPECIFICACIÓN DEL CASO DE USO: REGISTRAR PLANTILLAS DE CONTRATOS

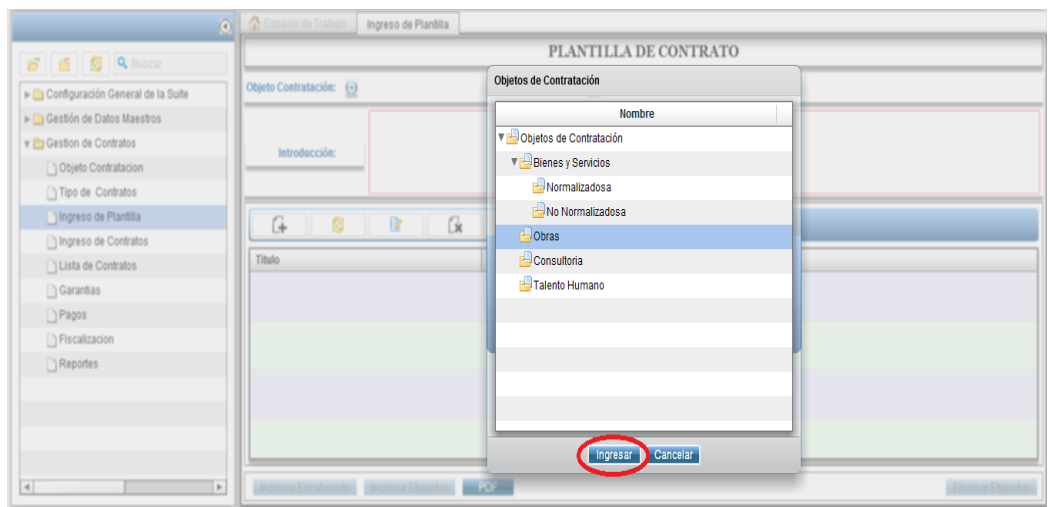
Referencia	Caso de Uso Registrar Plantillas de Contratos
GPICON-CU-004	<p>1. Descripción Breve</p> <p>El caso de uso describe la creación de las plantillas de contratos.</p> <p>2. Flujo Básico de Eventos</p> <p>A. El usuario ingresa al sistema</p> <p>B. El usuario va al menú en el panel Vertical donde estará el módulo de Ingreso Plantillas dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Ingreso Plantillas.</p>
	
	<p>Fuente: propia. Ilustración 40: Pantalla ingreso plantilla de contrato</p>

- C. Para el ingreso de plantillas de contratos, es importante consultar los Objetos de Contratación y Tipos de contratos para el ingreso de plantillas.



Fuente: propia.

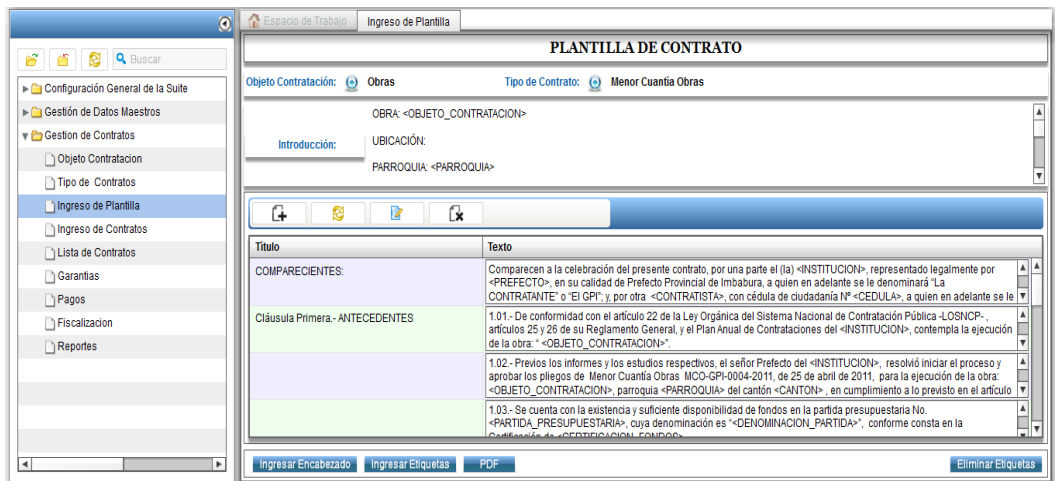
Ilustración 41: Pantalla seleccionar objeto de contratación



Fuente: propia.

Ilustración 42: Pantalla seleccionar tipo de contrato

- D. Obtenido el objeto de contratación y tipo de contrato aparecerá una tabla en donde estarán las cláusulas de los contratos. En donde el usuario podrá, actualizar, crear, eliminar una cláusula, con la barra de menú que se encuentra en la parte superior de la tabla.



Fuente: propia.

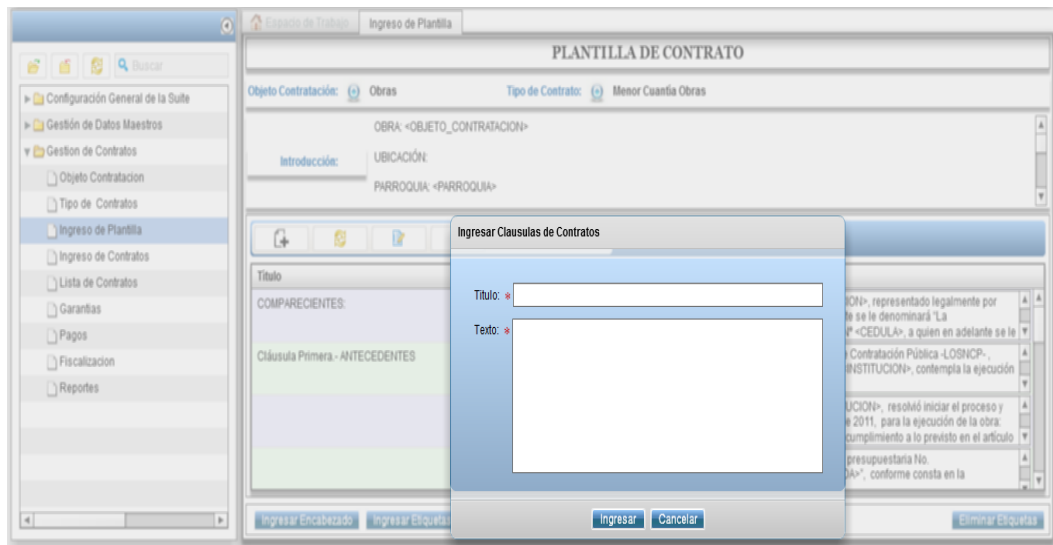
Ilustración 43: Barra de herramientas ingreso plantilla de contrato

- E. En introducción se va a ingresar el objeto de la contratación de la plantilla del contrato.
- F. Luego se ingresa las cláusulas de la plantilla del contrato que se tiene que ingresar en orden de acuerdo a la plantilla.
- G. Para ingresar una nueva cláusula se va a la barra de herramientas y seleccionamos Nuevo Registro y nos parecer una ventana en donde podemos ingresar una nueva cláusula del contrato.



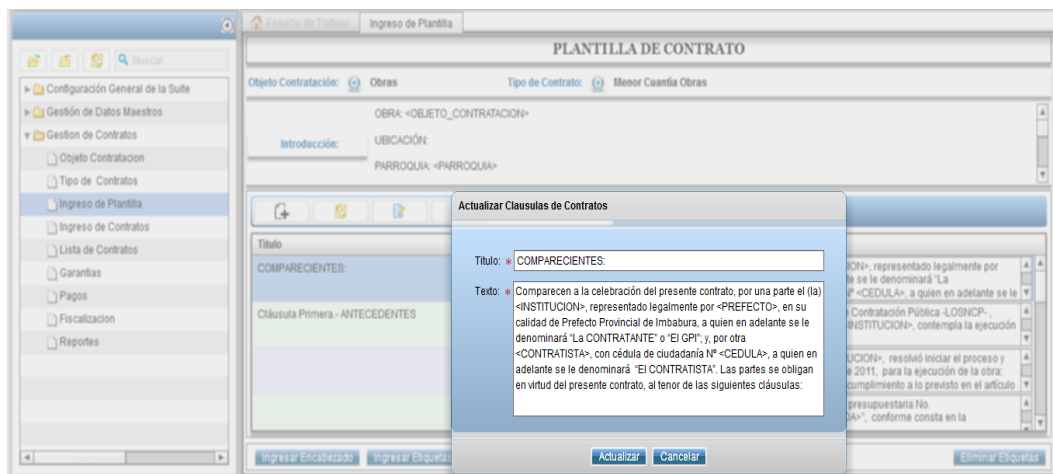
Fuente: propia.

Ilustración 44: Pantalla agregar registro plantilla de contrato



Fuente: propia.
 Ilustración 45: Pantalla agregar clausula a plantilla de contrato

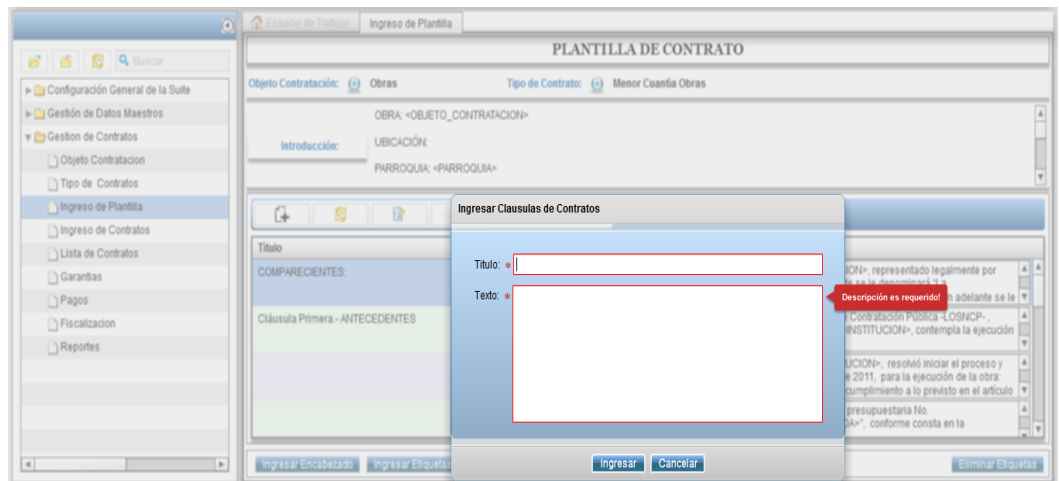
H. Para actualizar una clausula seleccionar la cláusula a actualizar y dar click en Actualizar Registro y saldrá una ventana de Editar Registro.



Fuente: propia.
 Ilustración 46: Pantalla actualizar clausula plantilla contrato

3. Flujos Alternativos

3.3. Cancelar Ingreso de cláusula de la plantilla del contrato.

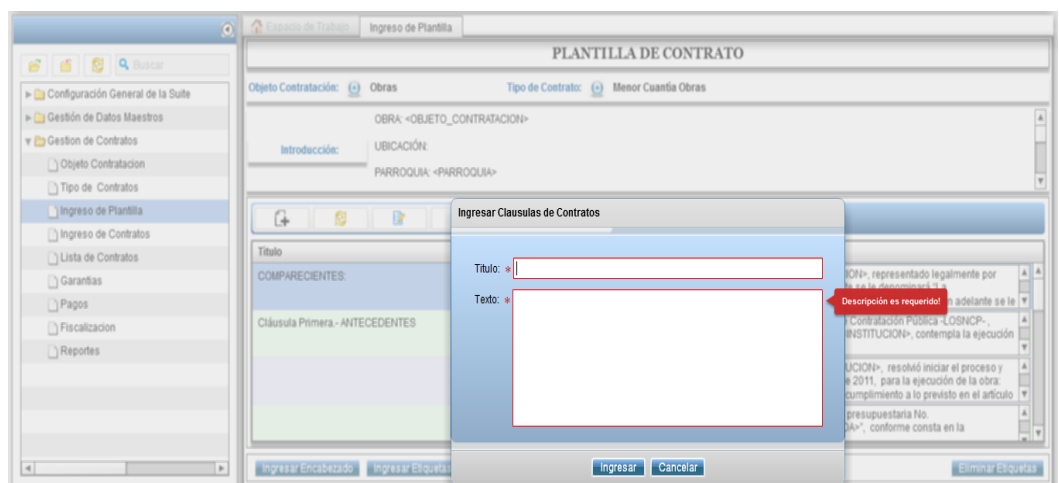


Fuente: propia.

Ilustración 47: Pantalla cancelar ingreso clausula plantilla contrato

3.4. Error en el ingreso de información de las cláusulas de la plantilla del contrato.

Si la información de las cláusulas no es correcta saldrá un error de validación de los campos incompletos, si un campo está incompleto saldrá el error y no dejara ingresar la cláusula.



Fuente: propia.

Ilustración 48: Error e el ingreso de información de cláusulas

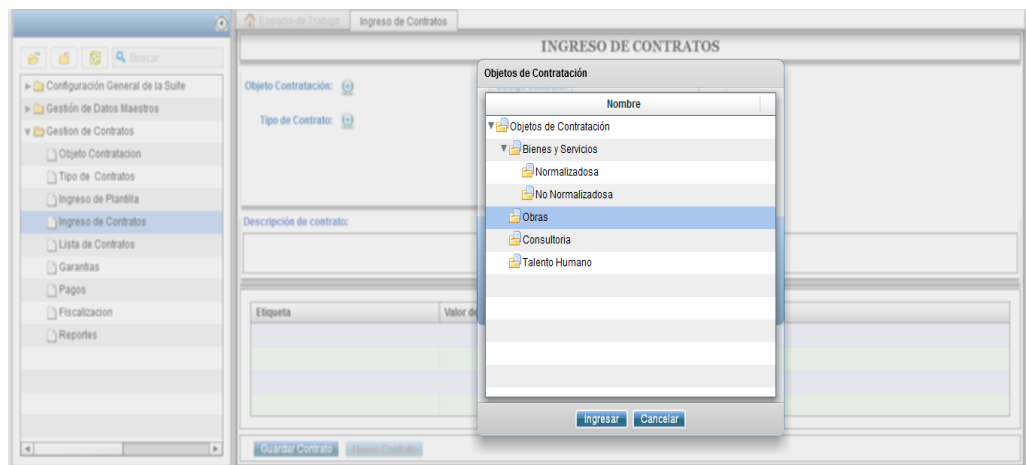
	<p>4. Precondiciones</p> <p>Se requiere previamente ingresar al sistema.</p> <p>Se requiere tener plantillas debidamente estructuradas para el ingreso al sistema.</p> <p>Se requiere tener la plantilla con la información a reemplazar con la cadena < >, ya que el sistema reemplazara las etiquetas que estén encerradas dentro de esta cadena. Ejemplo:</p> <p><INSTITUCIÓN></p> <p><DIRECCIÓN></p> <p>5. Post-Condiciones</p> <p>Luego de ingresar la plantilla se debe revisar si está bien estructurada con el botón Visualizar PDF para ver si las etiquetas de la plantilla están correctas.</p>
--	--

Fuente: propia.
 Tabla 35: Especificación de caso de uso Registrar Plantilla

4.3.5.- ESPECIFICACIÓN DEL CASO DE USO: CONSULTA OBJETO DE CONTRATACIÓN

Referencia	Caso de Uso Consulta Objeto de Contratación
GPICON-CU-005	<p>1. Descripción Breve</p> <p>El caso de uso describe la selección de un objeto de contratación para el ingreso de datos del contrato.</p> <p>2. Flujo Básico de Eventos</p> <p>A. El usuario ingresa al sistema</p> <p>B. El usuario va al menú en el panel Vertical donde estará el módulo de Ingreso Contratos dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Ingreso Contratos.</p>

C. Consultar objeto de contratación con la opción Objeto Contratación.



Fuente: propia.

Ilustración 49: Pantalla consultar objeto de contratación

3. Flujos Alternativos

Ninguno

4. Precondiciones

Se requiere ingresar previamente al sistema

Se requiere ingresar datos de objetos de contratación antes de consultar

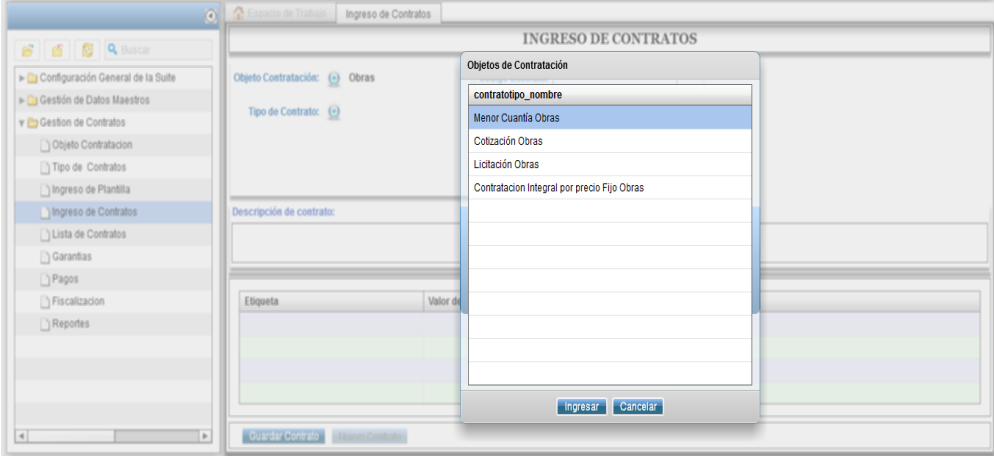
5. Post-Condiciones

Si no se consulta un objeto de contratación no se podrá acceder a los tipos de contratos.

Fuente: propia.

Tabla 36: Especificación de caso de uso Consulta objeto de contratación

4.3.6.- ESPECIFICACIÓN DEL CASO DE USO: CONSULTA TIPOS DE CONTRATOS


Referencia	Caso de Uso Consulta Tipos de Contratos
GPICON- CU-006	<p>1. Descripción Breve</p> <p>El caso de uso describe la selección de un tipo de contrato para el ingreso de datos del contrato.</p> <p>2. Flujo Básico de Eventos</p> <p>A. El usuario ingresa al sistema.</p> <p>B. El usuario va al menú en el panel Vertical donde estará el módulo de Ingreso Contratos dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Ingreso Contratos.</p> <p>C. Consultar tipo de contrato con la opción Tipo de Contrato</p>  <p>Fuente: propia.</p> <p>Ilustración 50: Pantalla consultar tipo de contrato</p> <p>3. Flujos Alternativos</p> <p>Ninguno</p>

	<p>4. Precondiciones</p> <p>Se requiere ingresar previamente al sistema</p> <p>Se requiere ingresar datos de los tipos de contratos antes de consultar</p> <p>5. Post-Condiciones</p> <p>Si no se consulta un tipo de contrato no se podrá acceder al registro de un contrato y no se podrá ver los valores de las etiquetas a reemplazar.</p>
--	--

Fuente: propia.

Tabla 37: Especificación caso de uso Consulta Tipos de contratos

4.3.7.- ESPECIFICACIÓN DEL CASO DE USO: INGRESO DATOS DE CONTRATOS

Referencia	Caso de Uso Ingreso Datos de Contratos
GPICON-CU-007	<p>1. Descripción Breve</p> <p>El caso de uso describe el ingreso de datos de un contrato.</p> <div style="text-align: center;">  </div> <p>Fuente: propia.</p> <p>Ilustración 51: Pantalla Ingreso datos de contratos</p> <p>2. Flujo Básico de Eventos</p>

- A. El usuario ingresa al sistema.
- B. El usuario va al menú en el panel Vertical donde estará el módulo de Ingreso Contratos dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Ingreso Contratos.
- C. Consultar objeto de contratación con la opción Objeto Contratación.
- D. Consultar tipo de contrato con la opción Tipo de Contrato
- E. Ingresar Datos específicos del formulario de Contrato como son:
 - i. Monto
 - ii. Ubicación
 - iii. Partida presupuestaria
 - iv. Contratista
 - v. Objeto de Contratación
- F. Luego ingresar la información del contrato a cambiar en la plantilla según el tipo de contrato y las etiquetas creadas en la plantilla y guardadas en la base de datos.

Etiqueta	Valor del campo
INSTITUCION	GOBIERNO PROVINCIAL DE IMBABURA
PREFECTO	
CONTRATISTA	
CEDULA	

Fuente: propia.

Ilustración 52: Pantalla ingresar etiquetas de información de contratos

G. Luego seleccionar Guardar Contrato para q la información del contrato se guarde en la base de datos.

Etiqueta	Valor del campo
INSTITUCION	GOBIERNO PROVINCIAL DE IMBABURA
PREFECTO	
CONTRATISTA	
CEDULA	

Fuente: propia.

Ilustración 53: Seleccionar guardar datos de contratos

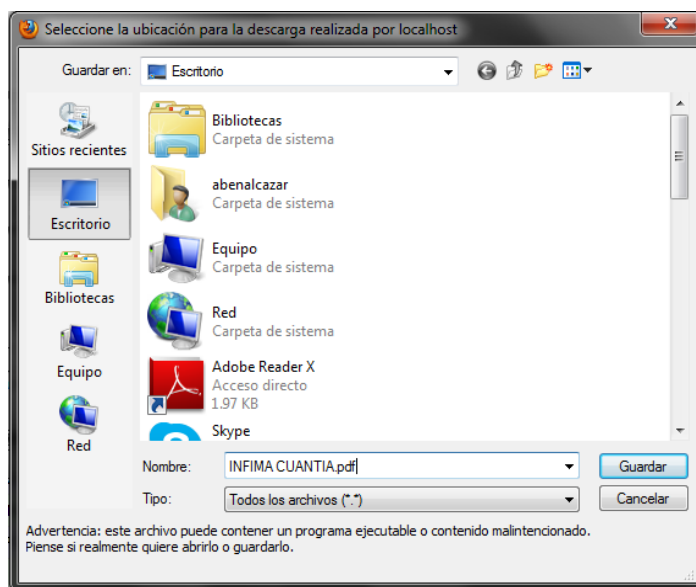
H. Para generar un documento PDF del contrato seleccionamos Visualizar PDF.

Etiqueta	Valor
DIRECTOR GESTION FINANCIERA	1
SERVICIO CONTRATADO POR EL GPI	1
PLAZO DEL CONTRATO	11
VALOR EN LETRAS DEL PAGO DEL CONTRATO	q
VALOR EN NUMEROS DEL PAGO DEL CONTRATO	1

Fuente: propia.

Ilustración 54: Seleccionar visualizar documento de contrato

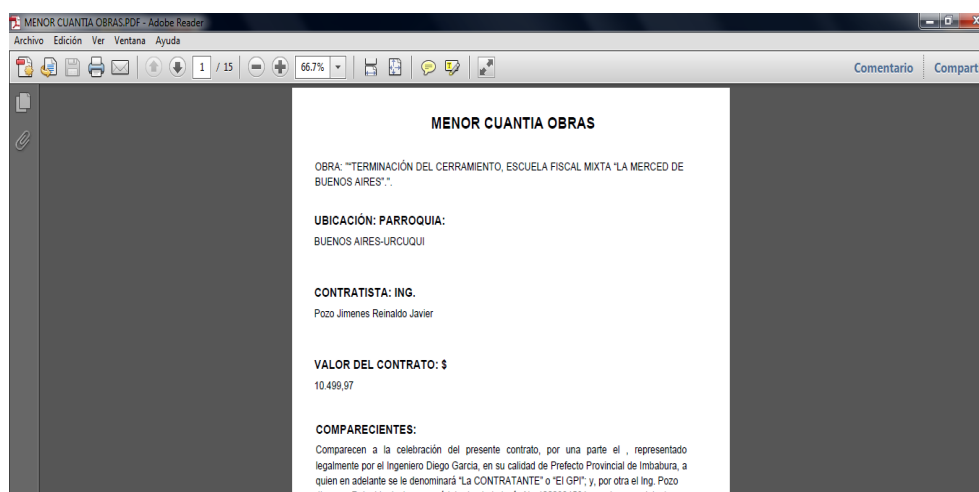
- I. Se abrirá la ventana donde puede guardar el contrato generado seleccionar una ruta donde se va a guardar.



Fuente: propia.

Ilustración 55: Guardar documento de contrato

- J. Luego nos presentará el contrato con la información ingresada en formato pdf.



Fuente: propia.

Ilustración 56: Visualizar documento de contrato

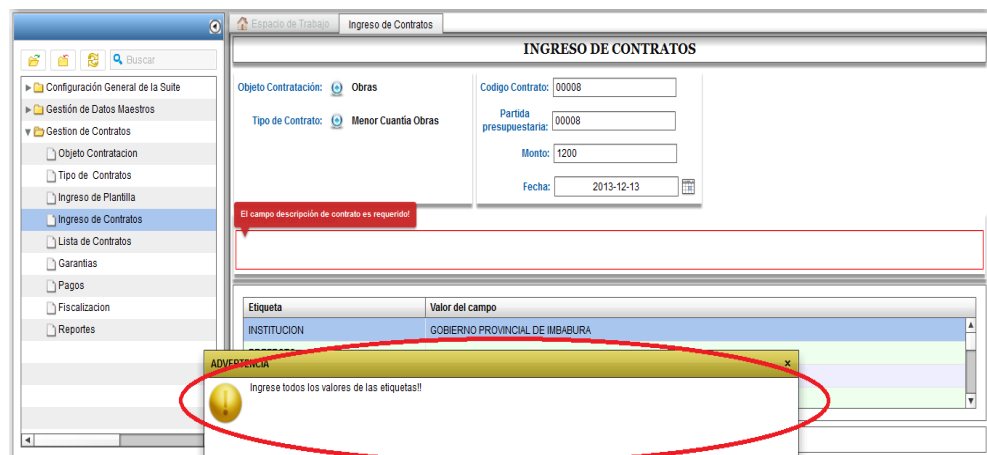
3. Flujos Alternativos

3.1. Mensaje de campo requerido, los valores del formulario de información del contrato son requeridos, si no se ingresa la información le saldrá un error de que el campo es requerido.



Fuente: propia.
Ilustración 57: Mensaje Campo Requerido

3.2. Mensaje Ingrese todos los valores de las etiquetas
Debemos ingresar la información de todas las etiquetas, si no se ingresa todas las etiquetas en el momento que damos click en Guardar contrato no nos dejara y nos saldrá el mensaje.



Fuente: propia.
Ilustración 58: Mensaje Insertar Todas las etiquetas

4. Precondiciones

Se requiere ingresar previamente al sistema

Se requiere ingresar datos de los objetos de contratación antes de ingresar el contrato.

Se requiere ingresar datos de los tipos de contratos antes ingresar la información del contrato, si no se escoge una opción no aparecerán las etiquetas a ser reemplazadas e ingresadas en el sistema

Se debe tener la plantilla ingresada en el sistema, para obtener automáticamente las etiquetas a ser reemplazadas en el contrato a generar.

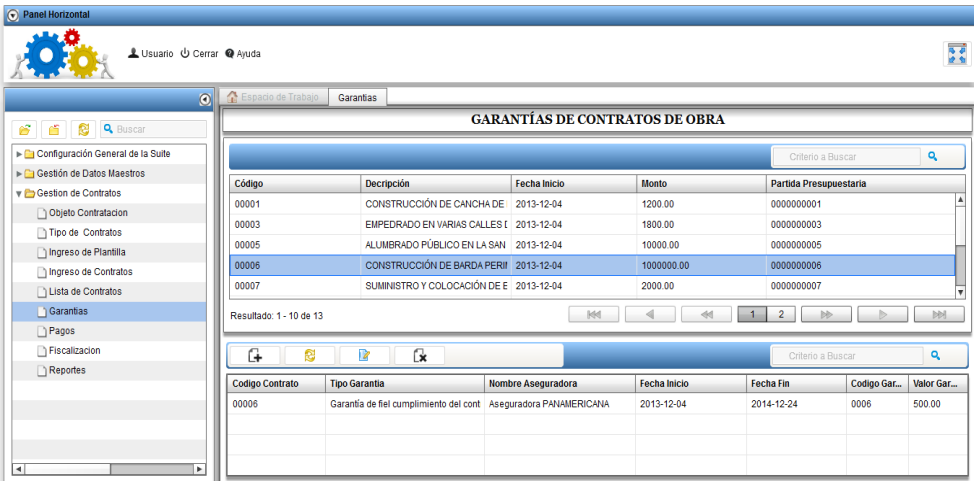
5. Post-Condiciones

Si no se ingresa la información de los contratos no se puede acceder a la información de cada uno para generar los reportes.

Fuente: propia.

Tabla 38: Especificación de caso de uso Ingreso datos de contratos


4.3.8.- ESPECIFICACIÓN DEL CASO DE USO: INGRESO DE GARANTÍAS

Referencia	Caso de Uso Ingreso de Garantías
GPICON- CU-009	<p>1. Descripción Breve</p> <p>El caso de uso describe el ingreso de Garantías de los contratos de obra.</p>  <p>Fuente: propia.</p> <p>Ilustración 59: Pantalla de Ingreso de Contratos</p>

	<p>2. Flujo Básico de Eventos</p> <p>A. El usuario ingresa al sistema.</p> <p>B. El usuario va al menú en el panel Vertical donde estará el módulo de Ingreso Contratos dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Contratos Obra.</p> <p>C. Seleccionar un contrato de Obra y dar click en la parte derecha en el botón Garantías, sobre este contrato y se desplegará una pantalla donde podrá ingresar la información de las garantías designadas para este contrato.</p> <p>3. Flujos Alternativos</p> <p>3.1. Mensaje de campo requerido, los valores del formulario de garantías del contrato son requeridos, si no se ingresa la información le saldrá un error de que el campo es requerido.</p> <p>4. Precondiciones</p> <p>Se requiere ingresar previamente al sistema Se requiere ingresar datos de las garantías de los contratos.</p> <p>5. Post-Condiciones</p> <p>Si no se ingresa la información de las garantías de los contratos no se puede acceder a la información de cada uno para generar los reportes.</p>
--	--

Fuente: propia.
Tabla 39: Especificación de casos de uso Ingreso de Garantías


4.3.9.- ESPECIFICACIÓN DEL CASO DE USO: PAGOS DE CONTRATOS

Referencia	Caso de Uso Pagos de Contratos
GPICON- CU-011	<p>1. Descripción Breve</p> <p>El caso de uso describe el ingreso de Pagos de los contratos de obra.</p>  <p>Fuente: propia.</p> <p>Ilustración 60: Pantalla Pagos de Contratos</p> <p>2. Flujo Básico de Eventos</p> <ul style="list-style-type: none">A. El usuario ingresa al sistema.B. El usuario va al menú en el panel Vertical donde estará el módulo de Ingreso Contratos dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Contratos Obra.C. Seleccionar un contrato de Obra y dar click en la parte derecha en el botón Pagos, sobre este contrato y se desplegará una pantalla donde podrá ingresar la información de los pagos para este contrato. <p>3. Flujos Alternativos</p> <ul style="list-style-type: none">3.1. Mensaje de campo requerido, los valores del formulario de pagos del contrato son requeridos, si no se ingresa la información le saldrá un error de que el campo es requerido.

	<p>4. Precondiciones</p> <p>Se requiere ingresar previamente al sistema Se requiere ingresar datos de los pagos de los contratos.</p> <p>5. Post-Condiciones</p> <p>La información de los pagos de los contratos es importante para generar los reportes de contratos</p>
--	---

Fuente: propia.
Tabla 40: Especificación de caso e uso Pago de contratos

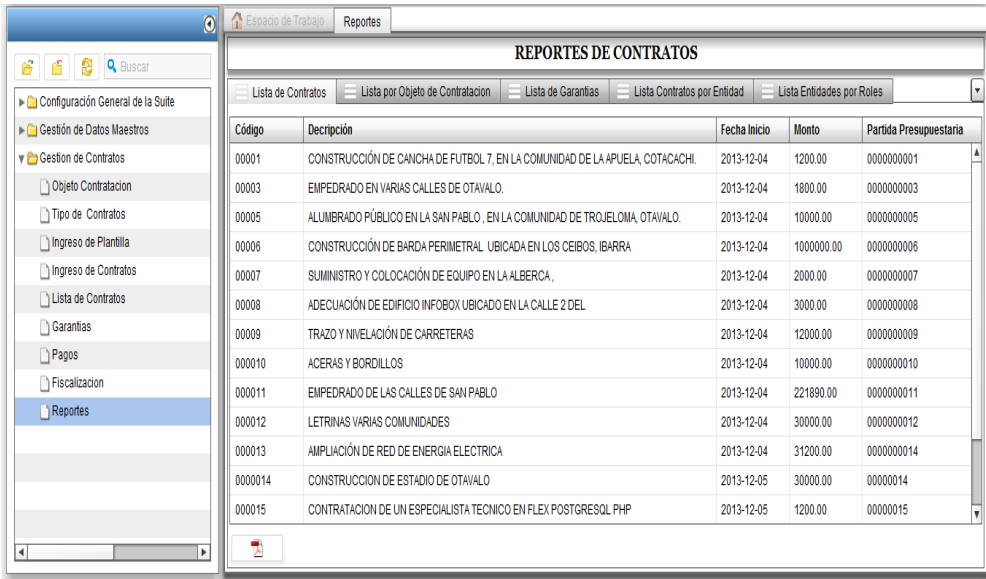
4.3.10.- ESPECIFICACIÓN DEL CASO DE USO: FISCALIZAR CONTRATOS

Referencia	Caso de Uso Fiscalizar Contratos
<p>GPICON- CU-012</p>	<p>1. Descripción Breve</p> <p>El caso de uso describe la fiscalización de los contratos de obra.</p> <div style="text-align: center;">  </div>
	<p style="text-align: center;">Fuente: propia. Ilustración 61: Pantalla Fiscalización de Contratos</p> <p>2. Flujo Básico de Eventos</p>

	<p>A. El usuario ingresa al sistema.</p> <p>B. El usuario va al menú en el panel Vertical donde estará el módulo de Ingreso Contratos dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Contratos Obra.</p> <p>C. Seleccionar un contrato de Obra y dar click en la parte derecha en el botón Fiscalizar, sobre este contrato y se desplegará una pantalla donde podrá ingresar la información de contrato.</p> <p>D. Ingresar la fecha de Inicio de Obra</p> <p>E. Ingresar las fechas de avance de Obra</p> <p>F. Ingresar la fecha de fin de Obra</p> <p>3. Flujos Alternativos</p> <p>3.1. Mensaje de campo requerido, los valores del formulario de fiscalización son requeridos, si no se ingresa la información le saldrá un error de que el campo es requerido.</p> <p>4. Precondiciones</p> <p>Se requiere ingresar previamente al sistema Se requiere ingresar las fechas de fiscalización de la obra de los contratos.</p> <p>5. Post-Condiciones</p> <p>La información de los pagos de los contratos es importante para generar los reportes de contratos</p>
--	--

Fuente: propia.
Tabla 41: Especificación caso de uso Fiscalizar contratos

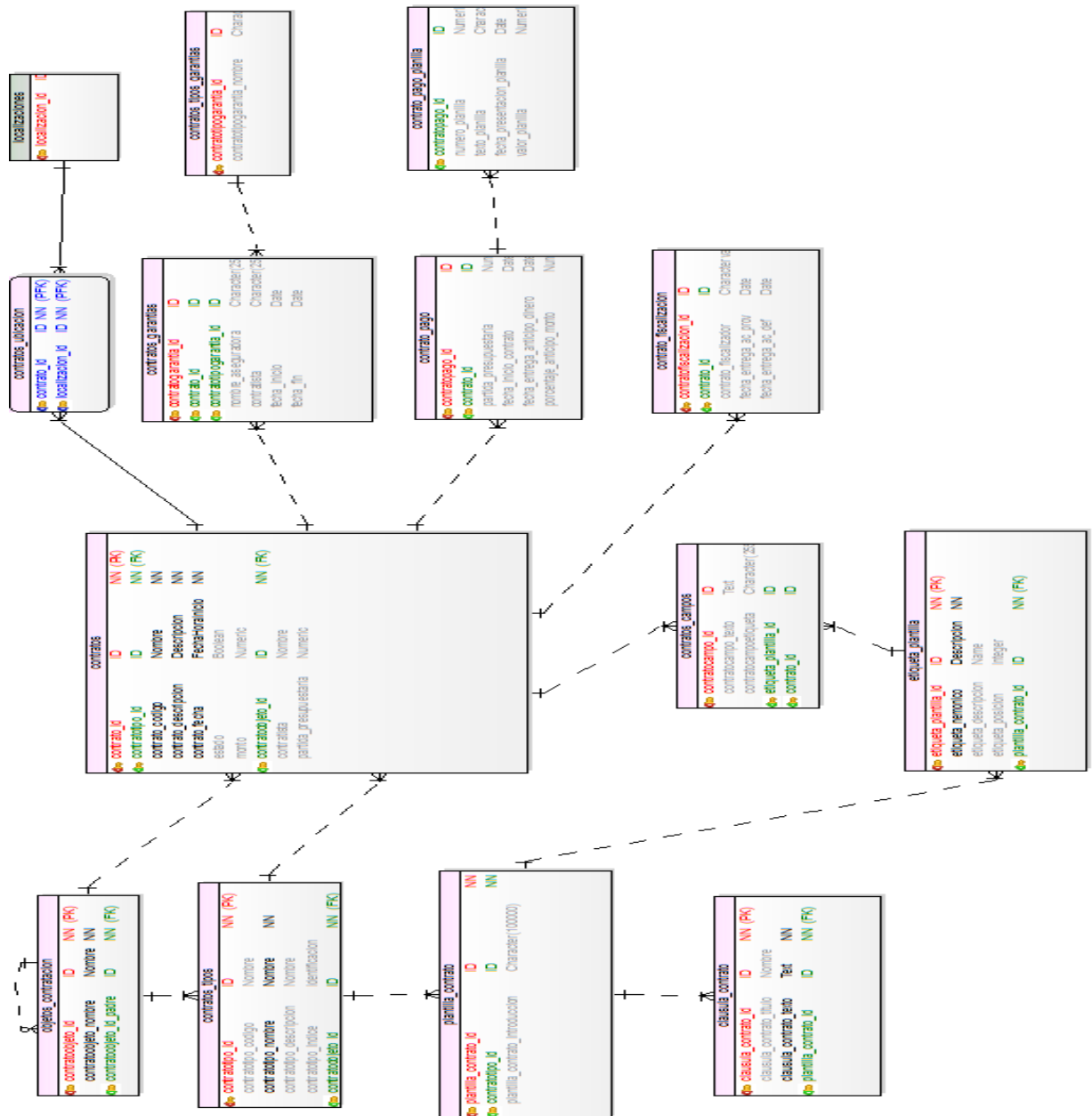
4.3.11.- ESPECIFICACIÓN DEL CASO DE USO: GENERAR REPORTES

Referencia	Caso de Uso Generar Reportes
GPICON- CU-013	<p>1. Descripción Breve</p> <p>Este caso de uso explica cómo generar los reportes disponibles del sistema GPI-CONTRATOS</p> <div style="text-align: center;">  </div> <p style="text-align: center;">Fuente: propia. Ilustración 62: Pantalla Reportes</p> <p>2. Flujo Básico de Eventos</p> <ol style="list-style-type: none"> A. El usuario ingresa al sistema. B. El usuario va al menú en el panel Vertical donde estará el módulo de Reportes dar doble click y en panel de navegación o espacio de trabajo se presentara el módulo de Reporte de Contratos. C. Tendremos tres opciones donde podremos obtener los siguientes reportes: <ol style="list-style-type: none"> i. Listado de todos los contratos por objeto de contratación. ii. Listado de las Garantías por Numero de Contrato iii. Listado de Contratos Fiscalizados iv. Listado de Fiscalizadores.

	<p>3. Flujos Alternativos N/A</p> <p>4. Precondiciones Se requiere ingresar previamente al sistema</p> <p>5. Post-Condiciones</p> <p>Todos los reportes pueden ser impresos directamente o guardados en el sistema.</p>
--	---

Fuente: propia.
Tabla 42: Especificación de caso de uso Generar reportes

4.4.- MODELO ENTIDAD-RELACIÓN



Fuente: propia.
Ilustración 63: Modelo Entidad Relación

CAPÍTULO V

5. INSPECCIÓN Y ADAPTACIÓN (IMPLEMENTACIÓN Y PRUEBAS)

5.1.- IMPLEMENTACIÓN DEL SISTEMA

5.1.1.- PRUEBAS

En el momento de las pruebas del sistema se utilizó datos de prueba y luego datos reales que permitieron la depuración de cada módulo, hasta cumplir con la totalidad de los requerimientos planteados.

El sistema se instalara en los servidores del Gobierno provincial de Imbabura, donde se va a configurar previo a la adquisición de dichos servidores, por lo tanto se implantó en un servidor local donde se realizó las diferentes pruebas del sistema.

Prueba de Integridad de Datos

Se comprobó que todos los métodos de acceso a la base de datos funcionan correctamente, para garantizar la integridad de los datos en el almacenamiento y recuperación de los mismos.

Pruebas funcionales del sistema

Se realizó las siguientes evaluaciones dependiendo de los casos de uso identificados:

- Verificar el acceso de usuarios del sistema.
- Verificar el registro, actualización, eliminación de objetos de contratación
- Verificar el registro, actualización, eliminación de los tipos de contratos
- Verificar el registro, actualización, eliminación de plantillas de los contratos donde se van a ingresar etiquetas (tokens) para cambio de información automática dependiendo del tipo de contrato q sea se va a tener una plantilla única para ese tipo de contrato.
- Verificar el registro, de datos de contratos según etiquetas de la plantilla del contrato q van a ser remplazadas con los datos ingresados en la plantilla para luego generar el contrato en un pdf.
- Verificar la búsqueda de contratos y actualización de los mismos si existieran errores en los datos ingresados del contrato.
- Verificar el control de fechas de las garantías del contrato.

- Verificar el ingreso de pagos de los contratos de obra.
- Verificar el ingreso de avances de obra y fechas de entrega recepción de la obra.
- Verificar la generación de reportes del sistema.

En todas las evaluaciones antes mencionadas se comprobó el correcto funcionamiento de cada una de las plantillas, paralelamente se realizó las comprobaciones de validaciones, reglas y condiciones establecidas para el sistema, asegurando los resultados y mensajes de error y advertencia esperados en cada caso.

Pruebas de Interfaz de usuario

Se verificó la correcta navegación por cada una de las pantallas y el estado adecuado de los objetos en las mismas.

Pruebas de seguridad y control de acceso

Se estableció que únicamente los usuarios puedan acceder al sistema, los usuarios q se encuentre registrados y activos en el sistema.

Las pruebas realizadas en el sistema GPI-CONTRATOS sirvieron para comprobar la funcionalidad, interfaz de usuario, integridad de datos, seguridad y control de acceso al sistema, lo cual permitió la aprobación y aceptación de las partes interesadas.

5.1.2.- INSTALACIÓN DEL SISTEMA

El proceso que se realizó para la instalación del sistema se puede ver en el manual de instalación en el ANEXO MANUAL DE INSTALACIÓN

5.1.3.- CAPACITACIÓN AL PERSONAL DEL GPI

Para el buen funcionamiento del sistema se capacito al personal implicado, explicando detalladamente cada módulo

Se está capacitando a la persona encargada de administrar el sistema, explicando el módulo de Administración del sistema, como son crear usuarios, crear objetos de contratación, crear tipos de contratos, crear las plantillas de los contratos.

Se está capacitando a la persona encargada de Ingresar los datos de los contratos, se explicó cómo debe de escoger los objetos de contratación y el tipo de contrato para que acceda a la plantilla de contrato para ingresar los datos en las etiquetas generadas en la plantilla.

Se está capacitando a la persona encargada de ingresar las garantías y pagos de contratos, como ingresar las fechas y controlar las mismas.

Se está capacitando a la persona encargada de ver las fechas de inicio y avance de contratos de obras para fiscalizar.

También se está capacitando al personal para que acceda a los reportes instalados en el sistema de contratos.

CAPÍTULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1.- CONCLUSIONES

- Con la implementación del sistema es posible un adecuado ingreso de información de los contratos debido a que se encuentran divididos por objeto de contratación y por tipos de contratos teniendo un orden de los contratos que existen en el Gobierno Provincial de Imbabura.
- Al ser una aplicación web permitirá que el usuario acceda al sistema sin necesidad de tener un instalador de escritorio, ingresara desde el navegador web con la url de la aplicación instalada en el servidor de aplicaciones.
- El uso de software libre en el proyecto se dio por los beneficios que conlleva, en relación al de sus alternativas privadas, por ejemplo porque no se invierte en la compra de licencias, y además porque el Gobierno Provincial de Imbabura es una entidad pública que debe cumplir con el mandato 1014 de Software libre.
- Al ingresar información a una base de datos que antes se realizaban manualmente y se guardaba en papel se obtuvo el ahorro de tiempo y de costos al momento de realizar las transacciones, teniendo integridad de información segura.
- Los módulos que componen el sistema de contratos están completamente integrados, son funcionales y de fácil uso, diseñados y desarrollados exclusivamente para cubrir todas las necesidades de los interesados.
- La reutilización de código agiliza el proceso de desarrollo. En el proyecto se desarrollaron componentes reutilizable como por ejemplo la barra de herramientas que es un estándar para cada componente de GPI-CONTRATOS.
- Se diseñó una aplicación de fácil manejo para el usuario para evitar cualquier tipo de error por parte de los usuarios que puedan causar pérdida de información.
- Al utilizar la metodología SCRUM se obtuvo las historias de usuario donde se describió las necesidades de los usuarios implicados y así desarrollarlas por medio del Product Backlog establecido.
- La metodología de desarrollo SCRUM utilizada para la implementación del presente proyecto, permite llevar a cabo un proceso de desarrollo de forma ordenada, flexible a los cambios y ágil en comparación a las metodologías tradicionales como RUP.

6.2.- RECOMENDACIONES

- Se debe adquirir los servidores de aplicaciones para la instalación del sistema con una dirección ip específica y que se encuentre en la intranet del GPI, teniendo los servidores listos se procederá a cargar el sistema GPI-CONTRATOS a ambiente de producción.
- Por la seguridad de la información se debe sacar respaldos de la información.
- Se recomienda hacer respaldos mensuales de la base de datos en un lugar externo de la institución, en caso de pérdida de la información se puede subir el ultimo respaldo.
- Para el desarrollo de aplicaciones web es recomendable usar herramientas de software libre, para evitar los altos costos de licenciamiento y la poca flexibilidad que presenta el software propietario. Se sugiere el uso de software libre, pues la tendencia actual es la utilización de estas herramientas que economizan la construcción de un sistema, permiten reutilizar código, dando un buen mantenimiento al sistema que se haya construido.
- Se recomienda a la carrera CISIC que se aplique el presente proyecto como un modelo para quienes se interesen más por conocer lo que es las aplicaciones RIA con Flex y PHP además sobre la utilización de SCRUM como metodología ágil de desarrollo.

CAPÍTULO VII

7. GLOSARIO DE TÉRMINOS

7.1.- ABREVIATURA

ERS	Especificación de Requisitos de Software
RUP	Rational Unified Process
UML	Unified Modeling Language
GPI	Gobierno Provincial de Imbabura
RIA	Rich Internet Applications
AS3	Action Script 3
DS	DataServices
VO	Value Objects

Fuente: propia.
Tabla 43: Abreviaturas

7.2.- DEFINICIONES

Paquetes	Agrupaciones de casos de uso y actores por funcionalidad que proveen.
Actor	Alguien o algo externo al sistema que interactúa con él.
Historias de Usuario	Requerimientos de los usuarios de forma específica.
Caso de Uso	Secuencia de acciones que el sistema realiza, la cual proporciona un resultado de valor observable.
MS Visio	Se refiere a la herramienta que permite realizar el modelado de los diagramas presentados en este documento.
Data Modeler	Herramienta para el modelado de bases de datos

Flash Builder 4.5 for PHP	Herramienta para el desarrollo de la aplicación
FLEX	Framework de Desarrollo del frontal de la aplicación e utiliza mxml y as3.

Fuente: propia.
Tabla 44: Definiciones

CAPÍTULO VIII

8. BIBLIOGRAFÍA

LIBROS

Joshua Noble, Todd Anderson. (2010). *Flex 3 Cookbook*. Estados Unidos: Reilly Media S.A. Libro.

Anderson, T. (2010). *Flex 4 in Action*. Estados Unidos: Adobe Systems Incorporated.

Stallons, J. (2010). *Flex 4 Getting*. Estados Unidos: Adobe Systems Incorporated.

PUBLICACIONES EN LÍNEA

Wikipedia. (n.d.). *Product Owner*. Retrieved from <http://es.wikipedia.org/wiki/Scrum>

Xavier Albaladejo, Certified Scrum Master. (n.d.). *Proyectos Ágiles*. From Proyectos Ágiles: <http://www.proyectosagiles.org/about>

Wikipedia. (n.d.). *SPRINT*. Retrieved Junio 24, 2013, from <http://es.wikipedia.org/wiki/Scrum>

Citón, M. L. (2006). *Metodología Ágil SCRUM*. Retrieved from <http://www.slideshare.net/testlucero/metodo-agil-scrum>

ESPOL, W. (n.d.). *Persona Natural*. Retrieved Junio 22, 2013, from http://www.wiki.espol.edu.ec/index.php/Personas_naturales

Wikipedia. (n.d.). *Persona Jurídica*. Retrieved Junio 22, 2013, from http://es.wikipedia.org/wiki/Persona_jur%C3%ADdica

Wikipedia. (n.d.). *Scrum Master*. From <http://es.wikipedia.org/wiki/Scrum>

Wikipedia. (n.d.). *TEAM*. Retrieved from <http://es.wikipedia.org/wiki/Scrum>

Wikipedia. (n.d.). *Product Backlog*. Retrieved from <http://es.wikipedia.org/wiki/Scrum>

Wikispaces. (n.d.). *ROLES DE SRUM*. Retrieved Junio 26, 2013, from <http://bi-trabajodegrado.wikispaces.com/Scrum>

Wordpress. (n.d.). *Builds Contínuos*. Retrieved from <http://santimacnet.wordpress.com/Builds>

Wikispaces. (n.d.). *Metodologías VS Tradicionales*. Retrieved from <http://metodologiasagiles.wikispaces.com/metodos+agiles+vs+metodos+tradicionales>

Flores, H. (2002). *Arquitectura Flex Zend*. Retrieved from http://haroldflores.com/blog_introduccion-aplicaciones-php-con-flex.php

Wikipedia. (n.d.). *RPC*. Retrieved from http://es.wikipedia.org/wiki/Remote_Procedure_Call

maestrosdelweb. (n.d.). *Zend Framework*. Retrieved from <http://www.maestrosdelweb.com/editorial/guia-zend/>

Wikipedia. (n.d.). *PHP*. Retrieved from <http://es.wikipedia.org/wiki/PHP>

Wikipedia. (n.d.). *FLEX*. Retrieved from http://es.wikipedia.org/wiki/Adobe_Flex

Wikipedia. (n.d.). *ActionScript*. Retrieved from <http://es.wikipedia.org/wiki/ActionScript>

Wikipedia. (n.d.). *AS3*. Retrieved from <http://es.wikipedia.org/wiki/ActionScript>

arpug. (n.d.). *postgres*. Retrieved from <http://www.arpug.com.ar/trac/wiki/TutorialPostgreSql?version=14&format=pdf>

Joshua Noble, Todd Anderson. (2010). *Flex 3 Cookbook*. Estados Unidos: Reilly Media S.A. Libro.

Stallons, J. (2010). *Flex 4 Getting*. Estados Unidos: Adobe Systems Incorporated.

Anderson, T. (2010). *Flex 4 in Action*. Estados Unidos: Adobe Systems Incorporated.

GPI. (n.d.). *Organigrama estructural GPI*. Retrieved from <http://www.imbabura.gob.ec/>

PostgreSQL. (2013, Febrero). Retrieved from <http://www.postgresql.org/>

ANEXOS

CAPÍTULO IX

9. ANEXOS

ANEXO 1: ANTEPROYECTO DE TESIS

Este documento está elaborado en formato digital y se encuentra en el CD

ANEXO 2: MANUAL DE INSTALACIÓN

Este documento está elaborado en formato digital y se encuentra en el CD

ANEXO 3: MANUAL DE USUARIO

Este documento está elaborado en formato digital y se encuentra en el CD

ANEXO 4: DICCIONARIO DE DATOS

Este documento está elaborado en formato digital y se encuentra en el CD