

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**“DESARROLLO DE UN SOFTWARE PARA EL SEGUIMIENTO DE
LAS OBRAS POR PARTE DE LA DIRECCIÓN ESTADAL DEL
MINISTERIO DEL PODER POPULAR PARA LAS OBRAS PÚBLICAS Y
VIVIENDA - ESTADO ANZOÁTEGUI”**

REALIZADO POR:

Rodríguez Chirinos, Dima José

**Trabajo de Grado presentado como requisito parcial para optar al Título
de: Ingeniero en Computación**

Barcelona, Agosto de 2010

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**“DESARROLLO DE UN SOFTWARE PARA EL SEGUIMIENTO DE
LAS OBRAS POR PARTE DE LA DIRECCIÓN ESTADAL DEL
MINISTERIO DEL PODER POPULAR PARA LAS OBRAS PÚBLICAS Y
VIVIENDA - ESTADO ANZOÁTEGUI”**

JURADO CALIFICADOR:

Ing. Gabriela Veracierta

Tutor Académico

Ing. Pedro Dorta

Jurado Principal

Ing. Rhonald Rodríguez

Jurado Principal

Barcelona, Agosto de 2010

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



**“DESARROLLO DE UN SOFTWARE PARA EL SEGUIMIENTO DE
LAS OBRAS POR PARTE DE LA DIRECCIÓN ESTADAL DEL
MINISTERIO DEL PODER POPULAR PARA LAS OBRAS PÚBLICAS Y
VIVIENDA - ESTADO ANZOÁTEGUI”**

Ing. Gabriela Veracierta

Tutor Académico

Ing. Damelis Vásquez

Tutor Externo

**Trabajo de Grado presentado como requisito parcial para optar al
Título de: Ingeniero en Computación**

Barcelona, Agosto de 2010

RESOLUCIÓN

De acuerdo al artículo 41 del Reglamento de Trabajos de Grado de la Universidad de Oriente.

“Los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario, para su autorización.”

Dedicatorias

DEDICATORIAS

Dedicado a mi tío Francisco Gómez, una de las pocas personas que creyó en mi, pero que lamentablemente ya no esta con nosotros para ver este sueño realizado.

Dedicado especialmente a mi padre, ya que sin su apoyo no hubiera podido lograr uno de mis más apreciados sueño.

Dedicado a mi madre adorada, la cual, le agradezco la paciencia que ha tenido para verme graduado.

Dedicado a mi tía carmen Guerra, la cual considero como mi segunda madre y es la persona que estuvo a mi lado estos años de estudio. De verdad muchas gracias, Te quiero Tía.

Dedicado a mi esposa novia Yanire, la cual adoro y amo con todo mi corazón, parte de este proyecto es gracias a ti mi amor te amo.

Dedicado a mi familia, quienes me han brindado todo su apoyo desde el principio de mi carrera.

Y finalmente a todas aquellas personas que de alguna u otra manera creyeron en mí y estuvieron apoyándome hasta el final. De verdad Muchas gracias.

Agradecimientos

AGRADECIMIENTOS

Antes que todo gracias a dios por fortalecerme y darme la sabiduría para poder realizar este proyecto.

A mis padres Dimas y Carmen, por el apoyo brindado, por darme ese empuje para seguir adelante y cumplir esta meta, no tan fácil de conseguir pero valió el esfuerzo.

A mi tía Carme y a mi tío Francisco, quienes me brindaron un hogar durante mis años de estudios en la UDO de todo corazón les doy las gracias y con el favor de dios espero retribuirles su esfuerzo.

Un agradecimiento especial a mi esposa novia Yanire, la cual, ha visto desde un principio y vivido al final este arduo camino todo los sacrificios hecho para llegar hasta aquí brindándome todo sus respaldo y comprensión, a de más de todo su amor, Gracias mi amor te amo.

Este agradecimiento va para mi tutora Gabriela Veracierta, quien con su paciencia y consejos logro encaminarme y así poder ver este proyecto hecho realidad. Mi más sincero agradecimiento hacia usted.

A mis hermanos Dionardis, Donny, Deivis y diorvin, a mis primos Fran, Claudina, Eiglys, Yesenia y primitos, quienes me apoyaron incondicionalmente y a quienes quiero mucho.

A Jean Carlos Días, Norelkis Tirado, Álvaro, Luis Cova, José Miguel, Endri, Víctor Guerra, Maickel Lezama, José Idrogo y demás amigos y compañeros, quienes

Agradecimientos

me brindaron una mano amiga cuando más lo necesitaba. También agradezco a mis amigos Marines Villegas, Félix Saavedra y esposa, quienes a pesar de la distancia me han brindado su gran e incondicional apoyo.

Y finalmente gracias a todas aquellas personas que por alguna u otras razón estuvieron involucrados en la realización de este proyecto. Humildemente gracias

RESUMEN

El siguiente proyecto de investigación se basa en el desarrollo de un sistema para el seguimiento de las obras viales y de edificaciones (SISOVE) para el centro regional del ministerio del poder popular para las obras públicas y vivienda del estado Anzoátegui. El software se encarga de proporcionar una interfaz amigable y de fácil manejo al personal que laboran en los distintos departamentos de la institución y muy especialmente al director del centro, que tiene la tarea de llevar el seguimiento de las obras ejecutadas en el estado, este trabajo se realizará por medio de consultas al sistema, el cual, le mostrará la imagen de un mapa referencial del estado y puntos que representaran a cada obra ingresada en el sistema. Esta información relacionada con las obras es guardada directamente en la base de datos ubicada en el departamento de informática de la institución, lo que mejorará la comunicación entre los departamentos. Este proyecto se construyó siguiendo el Proceso Unificado de Desarrollo de Software, y el Lenguaje de Modelado UML. Y para la construcción del sistema se utilizó el lenguaje de programación java, contenido en el entorno de programación Easyeclipse y finalmente como sistema manejador de base de datos se utilizó PostgreSQL.

ÍNDICE DE CONTENIDO

Contenido	Pág.
RESOLUCIÓN	IV
DEDICATORIAS	V
AGRADECIMIENTOS.....	VI
RESUMEN	VIII
ÍNDICE DE CONTENIDO	IX
ÍNDICE DE FIGURAS.....	XIV
ÍNDICE DE TABLAS	XVII
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	20
1.1. INTRODUCCIÓN	20
1.2. OBJETIVO GENERAL.....	24
1.3. OBJETIVOS ESPECÍFICO.....	24
CAPÍTULO II: MARCO TEÓRICO	25
2.1. ANTECEDENTES.....	25
2.2. RESEÑA HISTÓRICA.....	26
2.3. MISIÓN DE LA INSTITUCIÓN	27
2.4. VISIÓN DE LA INSTITUCIÓN	27
2.5. ORGANIGRAMA DE LA INSTITUCIÓN	28
2.6. UBICACIÓN DE LA INSTITUCIÓN	30
2.7. OBRA.....	30
2.8. CONTRATO DE OBRA.....	32
2.8.1. Tipos de Contratos que Maneja el Ministerio	32
2.9. AVANCE DEL CONTRATO.....	32
2.10. RELACIÓN QUE EXISTE ENTRE LAS FASES DE EJECUCIÓN DE UNA OBRA.....	32

Índice de Contenido

2.11. INSPECCIÓN DE OBRAS	33
2.11.1. Ingeniero Inspector	34
2.11.2. Ingeniero Residente.....	34
2.12. ENTE CONTRATANTE.....	34
2.13. ENTE EJECUTOR.....	35
2.14. CONTRATISTA	35
2.15. DESARROLLO DE SOFTWARE	35
2.15.1. Ingeniería de Software	36
2.15.2. Proceso Unificado de Desarrollo de Software	36
2.15.3. Iterativo e Incremental	36
2.15.4. Dirigido por caso de uso	37
2.15.5. Centrado en la Arquitectura	38
2.15.6. Fases del Proceso Unificado de Desarrollo de Software	38
2.15.6.1. Fase de Inicio	38
2.15.6.2. Fase de Elaboración	38
2.15.6.3. Fase de construcción	38
2.15.6.4. Fase de Transición	38
2.15.7. Flujo de Trabajo del Proceso Unificado.....	39
2.15.7.1. Captura de Requisitos	39
2.15.7.2. Análisis.....	39
2.15.7.3. Diseño	40
2.15.7.4. Implementación.....	40
2.15.7.5. Prueba.....	40
2.16. El Lenguaje Unificado de Modelado (UML).....	41
2.16.1. Diagramas UML.....	42
2.16.2. Diagrama de Casos de Uso	42
2.16.2.1. Diagrama de Clase	42
2.16.2.2. Diagrama de Secuencia	43
2.16.2.3. Diagrama de Objeto	44

Índice de Contenido

2.16.2.4. Diagrama de Colaboración.....	45
2.16.2.5. Diagrama de Estado	45
2.16.2.6. Diagrama de Actividades	46
2.16.2.7. Diagrama de Componente.....	46
2.16.2.8. Diagrama de Despliegue	47
2.17. PROGRAMACIÓN ORIENTADA A OBJETO (POO)	48
2.17.1. Características de la programación orientada a objeto.....	48
2.18. LENGUAJE DE PROGRAMACIÓN JAVA	50
2.18.1. Características	51
2.19. Base de Datos.....	53
2.19.1. Ventajas de las Bases de Datos	53
2.19.2. Sistema de Manejador de Base de Datos	54
2.19.3. Objetivos del SDBD.....	54
2.19.4. Diseño de Una Base de Datos	55
2.20. MANEJADOR DE BASE DE DATOS POSTGRES.....	56
2.21. JAVA 2D.....	57
2.22. DECRETO SOFTWARE LIBRE N° 3390.....	57
CAPÍTULO III: FASE DE INICIO	59
3.1. INTRODUCCIÓN	59
3.2. BREVE DESCRIPCIÓN DEL SISTEMA	59
3.3. REQUERIMIENTOS DEL SOFTWARE	60
3.4. MODELO DE DOMINIO.....	61
3.4.1. Terminología Empleada.....	63
3.5. DIAGRAMA DE CASO DE USO	65
3.5.1. Identificación de Actores	66
3.5.2. Descripción de Caso de Uso de Contexto.....	67
3.5.3. Casos de Usos detallados	70
3.5.3.1. Descripción de Caso de Uso Visualizar Mapas	70
3.5.3.2. Descripción de Caso de Uso Visualizar Punto.....	72

Índice de Contenido

3.5.3.3. Descripción de Caso de Uso Consultar Obra.....	73
3.5.3.4. Descripción de Caso de Uso Cargar Mapa	75
3.5.3.5. Descripción de Caso de Uso Gestionar Obra.....	77
3.5.3.6. Descripción de Caso de Uso Crear Punto de Referencia a Obra	78
3.5.3.7. Descripción de Caso de Uso Configurar Sistema	79
3.6. ANÁLISIS	80
3.6.1. Diagramas De Clases De Análisis.....	81
3.6.1.1. Diagrama de Clase de Análisis del Caso de Uso Visualizar Mapas	87
3.6.1.2. Diagrama de Clase de Análisis del Caso de Uso Visualizar Puntos	88
3.6.1.3. Diagrama de Clase de Análisis del Caso de Uso Consultar Obra.....	88
3.6.1.4. Diagrama de Clase de Análisis del Caso de Uso Cargar Mapa	90
3.6.1.5. Diagrama de Clase de Análisis del Caso de Uso Gestionar Obra.....	90
3.6.1.6. Diagrama de Clase de Análisis del Caso de Uso Crear Punto de Referencia a Obra	90
3.6.1.7. Diagrama de Clase de Análisis del Caso de Uso Configurar Sistema	92
3.6.2. Diagrama De Colaboración.....	93
3.6.3. Diagrama de Paquete de Análisis.....	100
3.7. RESUMEN DE LA FASE DE INICIO	102
CAPÍTULO IV: FASE DE ELABORACIÓN.....	103
4.1. INTRODUCCIÓN	103
4.2. DIAGRAMA DE CLASE DE DISEÑO.....	103
4.2.1. Diseño de las Clases.....	105
4.3. DIAGRAMA DE CAPA.....	111
4.4. DIAGRAMA DE SECUENCIA	112
4.5. DISEÑO DE LA BASE DE DATOS	120
4.5.1. Descripción de la base de datos	121
4.6. DISEÑO DE LA INTERFAZ	137
4.6.1. Diseño de la ventana de Validación	137
4.6.2. Diseño de la Ventana Principal.....	138

Índice de Contenido

4.6.3. Diseño de las Ventanas Internas o Hijas	138
4.7. DIAGRAMA DE COMPONENTES	139
4.8. DIAGRAMA DE DESPLIEGUE	139
4.9. RESUMEN DE LA FASE DE ELABORACIÓN	141
CAPÍTULO V: FASE DE CONSTRUCCIÓN.....	142
5.1. INTRODUCCIÓN	142
5.2. HERRAMIENTAS DE DESARROLLO UTILIZADAS	142
5.2.1. Lenguaje de programación Java.....	143
5.2.2. Manejador de base de datos PostgreSQL.....	143
5.3. Componentes del sistema.....	144
5.3.1. Codificación de los componentes.....	144
5.3.1.1. Codificación de los componentes relacionados a los mapas y puntos de referencia a las obras.....	144
5.4. PRUEBAS.....	196
5.4.1. Pruebas por Unidad.....	197
5.4.2. Pruebas de Integración	200
5.5. Resumen de la Fase de Construcción.....	207
CONCLUSIONES	208
RECOMENDACIONES	210
BIBLIOGRAFÍA	211
ANEXO A	214
ANEXO B	220
METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:.....	276

ÍNDICE DE FIGURAS

Figura		Pág.
Figura 2.1	Organigrama Estructural de la Dirección Estatal del Ministerio del Poder Popular para las Obras Públicas y Vivienda Anzoátegui	31
Figura 2.2	Relación entre Obra, Contrato y Avance	33
Figura 2.3	Proceso Unificado de Desarrollo de Software.	37
Figura 2.4	Diagrama de Casos de Uso.	42
Figura 2.5	Diagrama de Clase.	43
Figura 2.6	Diagrama de Secuencia.	44
Figura 2.7	Diagrama de Objeto.	44
Figura 2.8	Diagrama de Colaboración.....	45
Figura 2.9	Diagrama de Estado	46
Figura 2.10	Diagrama de Actividades.	47
Figura 2.11	Diagrama de Componente.....	47
Figura 2.12	Diagrama de Despliegue.	48
Figura 3.1	Modelo de Dominio	62
Figura 3.2	Descripción de Caso de Uso de Contexto SISOVE	68
Figura 3.3	Caso de Uso Detallado de Visualizar Mapas.	71
Figura 3.4	Caso de Uso Detallado de Visualizar Puntos.....	73
Figura 3.5	Caso de Uso Detallado de Consultar Obra.....	75
Figura 3.6	Caso de Uso Detallado de Cargar Mapa.	76
Figura 3.7	Caso de Uso Detallado de Gestionar Obra.....	78
Figura 3.8	Caso de Uso Detallado de Crear Punto de Referencia a Obra.	79
Figura 3.9	Caso de Uso Detallado de Configurar Sistema.	80
Figura 3.10	Esteriotipos Case de interfaz, control y de entidad.	81
Figura 3.11	Diagrama de Clase de Análisis del Caso de Uso Visualizar Mapas. ..	88

Índice de Figuras

Figura	Pág.
Figura 3.12 Diagrama de Clase de Análisis del Caso de Uso Visualizar Puntos... 89	89
Figura 3.13 Diagrama de Clase de Análisis del Caso de Uso Consultar obra..... 89	89
Figura 3.14 Diagrama de Clase de Análisis del Caso de Uso Cargar Mapa. 91	91
Figura 3.15 Diagrama de Clase de Análisis del Caso de Uso Gestionar Obra..... 91	91
Figura 3.16 Diagrama de Clase de Análisis del Caso de Uso Crear Punto de Referencia a Obra..... 92	92
Figura 3.17 Diagrama de Clase de Análisis del Caso de Uso Configurar Sistema 92	92
Figura 3.18Diagrama de colaboración detallado del caso de uso Visualizar Mapa 94	94
Figura 3.19Diagrama de colaboración detallado del caso de uso Visualizar Puntos 95	95
Figura 3.20Diagrama de colaboración detallado del caso de uso Consultar Obra 96	96
Figura 3.21 Diagrama de colaboración detallado del caso de uso Cargar Mapa.... 97	97
Figura 3.22Diagrama de colaboración detallado del caso de uso Gestionar Obra 98	98
Figura 3.23 Diagrama de colaboración detallado del caso de uso Crear Punto de Referencia a Obra..... 99	99
Figura 3.24 Diagrama de colaboración detallado del caso de uso Configurar Sistema 100	100
Figura 3.25 Identificación de Paquetes de análisis a partir de los Casos de Uso . 101	101
Figura 3.26 Diagrama de Paquete de Análisis..... 101	101
Figura 4.1 Diagrama de Clase de Diseño del Sistema SISOVE..... 104	104
Figura 4.2 Diagrama de capas del sistema 113	113
Figura 4.3 Diagrama de Secuencia del Caso de Uso Configurar Sistema..... 113	113
Figura 4.4 Diagrama de Secuencia del Caso de Uso Visualizar Mapas..... 114	114
Figura 4.5 Diagrama de Secuencia del Caso de Uso Visualizar Puntos..... 115	115

Índice de Figuras

Figura		Pág.
Figura 4.6	Diagrama de Secuencia del Caso de Uso Cargar Mapa.....	116
Figura 4.7	Diagrama de Secuencia del Caso de Uso Consultar Obra	117
Figura 4.8	Diagrama de Secuencia del Caso de Uso Gestionar Obra	118
Figura 4.9	Diagrama de Secuencia del Caso de Uso Crear Punto de Referencia a Obra	119
Figura 4.10	Modelo de la Base de Datos BD_SISOVE.	122
Figura 4.11	Ventana de Validación	138
Figura 4.12	Ventana principal del sistema	138
Figura 4.13	Ventanas hijas	139
Figura 4.14	Diagrama de Componentes	140
Figura 4.15	Diagrama de despliegue del sistema	141
Figura 5.1	Diagrama de Componente del sistema SISOVE.....	145
Figura 5.2	Ventana Interna “Consultar Obra”.....	201
Figura 5.3	Crear Punto de Referencia a Obra.....	201
Figura 5.4	Ventana interna “Agregar Obra”.....	202
Figura 5.5	Ventana de dialogo con mensaje de Carga Exitosa de datos	203
Figura 5.6	Ventana Interna “Consultar Obra”.....	203
Figura 5.7	Tabla Punto	204
Figura 5.8	Tabla Obra.....	204
Figura 5.9	Ventana interna “Consultar Contrato”	205
Figura 5.10	Ventana interna “Agregar Contrato”.....	206
Figura 5.11	Ventana interna “Consultar Contrato”	206
Figura 5.12	Tabla Contrato.....	207

ÍNDICE DE TABLAS

Tabla	Pág.
Tabla 3.1. Requisitos Funcionales	60
Tabla 3.2. Requisitos No Funcionales.....	61
Tabla 3.3. Descripción de las clases del dominio (1/3).....	63
Tabla 3.4. Descripción de las clases del dominio (2/3).....	64
Tabla 3.5. Descripción de las clases del dominio (3/3).....	65
Tabla 3.6. Clases de Interfaz del sistema (1/3)	82
Tabla 3.7. Clases de Interfaz del sistema (2/3)	83
Tabla 3.8. Clases de Interfaz del sistema (3/3)	84
Tabla 3.9. Clases de Control del sistema (1/3).....	84
Tabla 3.10. Clases de Control del sistema (2/3).....	85
Tabla 3.11. Clases de Control del sistema (3/3).....	86
Tabla 3.12. Clases de Entidad del sistema (1/2)	86
Tabla 3.13. Clases de Entidad del sistema (2/2)	87
Tabla 4.1. Clases que componen al sistema (1/7)	105
Tabla 4.2. Clases que componen al sistema (2/7)	106
Tabla 4.3. Clases que componen al sistema (3/7)	107
Tabla 4.4. Clases que componen al sistema (4/7)	108
Tabla 4.5. Clases que componen al sistema (5/7)	109
Tabla 4.6. Clases que componen al sistema (6/7).	110
Tabla 4.7. Clases que componen al sistema (7/7)	111
Tabla 4.8. Tabla Punto	121
Tabla 4.9. Tabla Usuario	123
Tabla 4.10. Tabla Mapa	124
Tabla 4.11. Tabla Obra.....	124

Índice de Tablas

Tabla	Pág.
Tabla 4.12. MontoEstimado.....	125
Tabla 4.13. Tabla ObraContrato.....	125
Tabla 4.14. Tabla Contrato (1/3).....	126
Tabla 4.15. Tabla Contrato (2/3).....	127
Tabla 4.16. Tabla Contrato (3/3).....	128
Tabla 4.17. Tabla MetasCuantitativas.....	128
Tabla 4.18. Tabla CaracteristicaContrato(1/2).....	128
Tabla 4.19. Tabla CaracteristicaContrato(2/2).....	129
Tabla 4.20. Tabla Empresa.....	129
Tabla 4.21. Tabla ContratoEmpresa.....	130
Tabla 4.22. Tabla IngEmpresa	130
Tabla 4.23. Tabla IngResidente(1/2).....	130
Tabla 4.24. Tabla IngResidente(2/2).....	131
Tabla 4.25. Tabla ContratoAvance	131
Tabla 4.26. Tabla Avance (1/3).....	131
Tabla 4.27. Tabla Avance (2/3).....	132
Tabla 4.28. Tabla Avance(3/3).....	133
Tabla 4.29. Tabla IngContrato	134
Tabla 4.30. Tabla IngInspector (1/2).....	134
Tabla 4.31. Tabla IngInspector (2/2).....	135
Tabla 4.32. Tabla AvanceMetasCuantitativa	135
Tabla 4.33. Tabla Foto (1/2)	135
Tabla 4.34. Tabla Foto (2/2)	136
Tabla 4.35. Tabla Actividades	136
Tabla 4.36. Tabla Valuaciones.....	136
Tabla 5.1. Clases de equivalencia para las pruebas de caja negra de “Administrar obra”. (1/2)	197

Índice de Tablas

Tabla	Pág.
Tabla 5.2. Clases de equivalencia para las pruebas de caja negra de “Administrar obra”. (2/2)	198
Tabla 5.3. Caso de prueba de caja negra de “Incluir obra” (1/2)	198
Tabla 5.4. Caso de prueba de caja negra de “Incluir obra” (2/2)	199

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. INTRODUCCIÓN

Inicialmente es creado para el año de 1977 el Ministerio de Transporte y Comunicaciones, el cual, nace por la necesidad de unificar en un solo organismo las actividades relacionadas con Transporte y Comunicaciones. El Decreto Presidencial N° 1.990 dispuso la iniciación de sus actividades. En el mismo año mediante el Decreto N° 2,088 es creado el Ministerio de Desarrollo Urbano, siendo este el encargado de las tareas relacionadas con el Desarrollo Urbano.

Luego estos dos Ministerios son unificados para crear el Ministerio de Infraestructura, la misma se realizó cumpliendo los lineamientos que establecen en la promulgación de la reforma de la Ley Orgánica de la Administración Central, publicada en Gaceta Oficial de la República Bolivariana de Venezuela, N° 36,775 de fecha 30/08/99, posteriormente modificada a través de la Gaceta Oficial de la República Bolivariana de Venezuela, N° 36,850 de fecha 14/12/99.

En Diciembre del año 2006 existe una reorganización estructural del Estado por la cual se crea el Ministerio del Poder Popular para las Telecomunicaciones e Informática y se cambia de nombre para el Ministerio del Poder Popular para la Infraestructura, anteriormente MINFRA, teniendo como nuevas competencias las publicadas en Gaceta Oficial N° 38.599 de fecha 08 de Enero de 2007 según Decreto N° 5.103, mediante el cual se dicta el Decreto Sobre Organización y Funcionamiento de la Administración Pública Nacional.

CAPITULO I: Planteamiento del Problema

Recientemente fue creado el Ministerio para el Poder Popular para las Obras Públicas y Vivienda (MOPVI), a través de la Gaceta Oficial número 39.130 de fecha martes 03 de marzo de 2009, que pasa a sustituir a los Ministerios del Poder Popular para la Infraestructura y para la Vivienda y Hábitat.

La creación de este Ministerio atiende a la necesidad del Estado venezolano de planificar y unificar la construcción de obras públicas y vivienda, de desarrollar bajo un mismo plan las políticas rectoras de ambas materias, para ajustarlas al crecimiento y necesidades de la población.

Entre las funciones asignadas a la recién designada Cartera se incluye la formulación y el seguimiento de políticas en materia de vialidad, circulación y tránsito. Además se le confiere la rectoría del Sistema Nacional de Vivienda y Hábitat, que comprende la formulación, ejecución y seguimiento de la política integral del Estado en materia de Vivienda y Hábitat.

Actualmente el Ministerio para el Poder Popular para las Obras públicas y Vivienda cuenta con un total de 24 Direcciones Estadales ubicadas cada una por estado, los cuales tienen en común la *Misión* de Ejercer la rectoría mediante la formulación, adopción, seguimiento, evaluación de las políticas, estrategias, programas y proyectos en los Sectores de Ordenación del Territorio Urbanístico y Transporte, apoyados en nuestro talento humano, garantizando servicios y el desarrollo equilibrado de la infraestructura física, para mejorar la calidad de vida del pueblo soberano, bajo un modelo socialista de participación e inclusión. También comparten la *Visión* de Ser modelo en la administración Pública Nacional, generando productos y servicios de excelencia en materia de Planificación Urbanística, Equipamiento Territorial, Modos de Transporte y Vialidad; en cooperación y consenso con los Poderes Públicos Nacionales y el Poder Popular Comunal a objeto de consolidar la infraestructura física.

CAPITULO I: Planteamiento del Problema

La Dirección Estatal MOPVI-Anzoátegui lleva el seguimiento y la supervisión de las obras públicas y viales (principal y rural), realizada en la entidad mediante el levantamiento de informes por parte de los ingenieros inspectores asignados a tres departamentos:

Vialidad, este se encarga de supervisar, coordinar e inspeccionar las obras de las principales vías del estado, entre otras asignaciones.

Equipamiento Territorial, tiene dentro de sus funciones fiscalizar las obras de equipamiento que ejecuta el ministerio en el estado, garantizando que se realice de acuerdo a las especificaciones de la obra.

Servicio autónomo de vialidad agrícola (S.A.V.A) su meta es mejorar las condiciones de transitabilidad de la red vial agrícola y de velar por su mantenimiento, abarcando otras funciones.

Estas dependencias generan un conjunto de reportes que sirven para llevar el seguimiento de los avances de las obras, estos son realizados por los departamentos de Vialidad y Equipamiento territorial, luego son remitidos a la división de Gestión que tiene como objetivo dirigir, coordinar, controlar y desarrollar las actividades orientadas en la ejecución e inspección de las obras publicas de equipamiento y vialidad en la región.

La dirección regional MOPVI-Anzoátegui y la Oficina Estratégica de Seguimiento y Evaluación de Políticas Públicas del Sector Infraestructura (O.E.S.E.P.P.S.I.) evaluarán la información generada por los informes, para determinar en que grado se encuentran las obras.

CAPITULO I: Planteamiento del Problema

El S.A.V.A. remite la información generada de la ejecución de una obra de vialidad agrícola a un ente superior directamente, como lo es la sede central del S.A.V.A. en caracas e indirectamente a la dirección del MOPVI-Anzoátegui.

En la actualidad los pasos descrito anteriormente son realizado de manera manual, siendo improductiva la labor a la hora de evaluar la información generada de la ejecución de una obra, es por esto, que se necesita la automatización de este proceso mediante una aplicación que contara con la imagen del mapa del estado Anzoátegui y puntos de referencia dinámicos a las obras, siendo los últimos encargados de tener un enlace con la información de las obras y de contratos por parte de los departamentos de Vialidad, de Equipamiento Territorial y el S.A.V.A.

Dicho control automático se realizara mediante la creación de un software hecho en lenguaje Java, el cual por sus numerosas ventajas fue elegido, siendo una de estas la de poder ejecutarse en múltiples plataforma. A de más este lenguaje de programación se adapta al decreto presidencial **38.095** en el cual indica que “La Administración Pública Nacional empleará prioritariamente Software Libre desarrollado con Estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. A tales fines, todos los órganos y entes de la Administración Pública Nacional iniciarán los procesos de migración gradual y progresiva de éstos hacia el Software Libre desarrollado con Estándares Abiertos”. También utilizaremos postgresSQL como manejador de base de datos, siendo este uno de los mejores manejadores de base de datos y de libre utilización.

Este programa será de gran utilidad para que la Dirección regional MOPVI-Anzoátegui, debido a que, llevara de manera rápida y sencilla el seguimiento de las obras, ya que, podrá localizar las obras dependiendo de su ubicación en el mapa, el cual ser referenciado por un punto que le indicara dependiendo su color el estatus de

CAPITULO I: Planteamiento del Problema

la obra de manera dinámica, de esta forma se podrán atender las obras con mayor prioridad y tener conocimiento general del estado de todas las obras.

1.2. OBJETIVO GENERAL

Desarrollar un Software para el Seguimiento de las Obras por parte de La Dirección Estatal del Ministerio del Poder Popular para las Obras Públicas y Vivienda - Estado Anzoátegui.

1.3. OBJETIVOS ESPECÍFICO

1. Identificar los requerimientos para la elaboración del software.
2. Diseñar la base de datos del sistema que facilite la integración de la información de una forma consistente y segura.
3. Diseñar interfaces amigables.
4. Programar los módulos del software.
5. Realizar las pruebas del software desarrollado.

CAPÍTULO II: MARCO TEÓRICO

2.1. ANTECEDENTES

Fuad, N. (2008) realizó un trabajo titulado: “**Desarrollo de un software para el control de inventario del material importado de una ensambladora de vehículos**”. Con este Software se buscó el control de inventario, automatizando los procesos realizados para dicho control.

Ana, O. y Stefano N. (2007) realizó un trabajo titulado “**Desarrollo de un software para la automatización y seguimiento de las actividades que se realiza en el almacén de un taller de mantenimiento aeronáutico**”. La idea del proyecto es controlar el proceso aeronáutico, además de tener conexión directa con otros departamentos con los cuales se puede obtener información necesaria para llevar el control de las actividades del almacén.

Carlos, M (2005) realizó un trabajo titulado “**Desarrollo de un software para el seguimiento de los bienes ubicados en el área de control de almacenamiento de una institución aduanera, ubicada en Guanta, estado Anzoátegui**”. El software es capaz de establecer de manera eficaz y eficiente todas las instancias de una mercancía desde la perspectiva del tiempo y la legalidad lo cual brinda información cierta y confiable a los usuarios del área de mantenimiento.

Deomary, V y Jorge R. (2005) realizó un trabajo titulado “**Desarrollo de un Software para el Avance, Control y Valuaciones de Obras Civiles y Mecánicas, Aplicado al departamento de planificación de una empresa de construcción**”. El proyecto se refiere al desarrollo de un software que automatice los procesos manuales, aumentando la comunicación entre departamentos.

CAPÍTULO II: Marco Teórico

Frank, Álvarez. (2004) realizó un trabajo titulado **Desarrollo de un Software para la Automatización de la Gestión y Control de Proyectos de una Empresa que Presta Servicios Petroleros**. El proyecto se refiere al desarrollo de un software que automatice de manera más eficiente el proceso utilizado, debido al aumento de proyectos requeridos.

Dimas, R (2003) realizó un trabajo titulado “**Desarrollo de un sistema para la administración y control de suministros y equipos perteneciente a un almacén de una empresa petrolera**”. Este software permite la administración y el control de suministros y equipos del almacén AUX-1 de la gerencia de la planta sur, PDVSA, en el distrito San Tomé.

2.2. RESEÑA HISTÓRICA

Inicialmente es creado para el año de 1977 el Ministerio de Transporte y Comunicaciones, el cual, nace por la necesidad de unificar en un solo organismo las actividades relacionadas con Transporte y Comunicaciones. El Decreto Presidencial N° 1.990 dispuso la iniciación de sus actividades. En el mismo año mediante el Decreto N° 2,088 es creado el Ministerio de Desarrollo Urbano, siendo este el encargado de las tareas relacionadas con el Desarrollo Urbano.

Luego estos dos Ministerios son unificados para crear el Ministerio de Infraestructura, la misma se realizó cumpliendo los lineamientos que establecen en la promulgación de la reforma de la Ley Orgánica de la Administración Central, publicada en Gaceta Oficial de la República Bolivariana de Venezuela, N° 36,775 de fecha 30/08/99, posteriormente modificada a través de la Gaceta Oficial de la República Bolivariana de Venezuela, N° 36,850 de fecha 14/12/99.

CAPÍTULO II: Marco Teórico

En Diciembre del año 2006 existe una reorganización estructural del Estado por la cual se crea el Ministerio del Poder Popular para las Telecomunicaciones e Informática y se cambia de nombre para el Ministerio del Poder Popular para la Infraestructura, anteriormente MINFRA, teniendo como nuevas competencias las publicadas en Gaceta Oficial N° 38.599 de fecha 08 de Enero de 2007 según Decreto N° 5.103, mediante el cual se dicta el Decreto Sobre Organización y Funcionamiento de la Administración Pública Nacional.

Recientemente fue creado el Ministerio para el Poder Popular para las Obras Públicas y Vivienda (MOPVI), a través de la Gaceta Oficial número 39.130 de fecha martes 03 de marzo de 2009, que pasa a sustituir a los Ministerios del Poder Popular para la Infraestructura y para la Vivienda y Hábitat.

2.3. MISIÓN DE LA INSTITUCIÓN

“Es de Ejercer la rectoría mediante la formulación, adopción, seguimiento, evaluación de las políticas, estrategias, programas y proyectos en los Sectores de Ordenación del Territorio Urbanístico y Transporte, apoyados en nuestro talento humano, garantizando servicios y el desarrollo equilibrado de la infraestructura física, para mejorar la calidad de vida del pueblo soberano, bajo un modelo socialista de participación e inclusión” (www.mopvi.gob.ve/. 2009).

2.4. VISIÓN DE LA INSTITUCIÓN

“Es Ser modelo en la administración Pública Nacional, generando productos y servicios de excelencia en materia de Planificación Urbanística, Equipamiento Territorial, Modos de Transporte y Vialidad; en cooperación y consenso con los Poderes Públicos Nacionales y el Poder Popular Comunal a objeto de consolidar la infraestructura física” (www.mopvi.gob.ve/. 2009).

2.5. ORGANIGRAMA DE LA INSTITUCIÓN

La institución tiene la siguiente organización (ver figura 2.1) en el nivel superior vemos a los organismos adscrito y tutelados que cumplen funciones que le señalan sus respectivas leyes de creación, reglamentos y demás normativa, a demás también esta en el nivel superior la Dirección Estatal Anzoátegui, encargada de ejercer la rectoría en la formulación, adaptación, seguimiento y evaluación de las políticas públicas en materia de desarrollo urbanístico y transporte, mediante los estudios, programas, proyectos, obras y servicios de infraestructura.

Luego vemos a la comisión de auditoria que tiene como función la verificación exhaustiva efectuada a través de un examen posterior, objetivo, sistemático y profesional de las actividades administrativas y financieras que realiza la dirección Estatal.

También vemos en el organigrama a la Consultoría jurídica, la cual, asesora y asiste jurídicamente al director de la dirección, a las divisiones, unidades y departamentos, organismos adscrito y a demás particulares sobre los asuntos sometidos a consideración relacionados con el área de competencia de este ministerio.

Más abajo vemos la Oficina Estratégica de Seguimiento y Evaluación de Políticas Públicas del Sector Infraestructura, siendo esta la que realiza el seguimiento y la evaluación de las políticas públicas del ministerio y entes adscritos a través de la implementación y/o aplicación de mecanismos y herramientas que permitan realimentar el proceso de toma de decisiones en el nivel estratégico.

A nivel medio tenemos la Oficina de Administración y Servicio y la oficina de recursos humanos. La función de la primera es coordinar y administrar las actividades

CAPÍTULO II: Marco Teórico

financieras, fiscales, contables y de ejecución presupuestaria de la dirección estatal, a través de la aplicación de normas y procedimientos que rigen la competencia de la misma. La segunda oficina tiene como objetivo asistir técnicamente al director y al resto de las dependencias, en el reclutamiento, selección, clasificación, remuneración, registro y control, a demás de aplicar el sistema de evaluación de desempeño y eficiencia de personal, entre otras funciones.

Por último tendremos las divisiones, como son la de gestión y la de planificación de infraestructura a demás de la oficina de servicio autónomo de vialidad agrícola. La División de gestión se encarga dirigir, coordinar, controlar y desarrollar las actividades orientadas en la ejecución e inspección de obras públicas de equipamiento y vialidad en la región. La otra división, la planificación, tiene como función coordinar, elaborar, revisar y actualizar estudios urbanos a través de convenios interinstitucionales referidos a planes de ordenación, desarrollo urbano, planes especiales o cualquier otra investigación. Y por ultimo tenemos al servicio autónomo de vialidad agrícola creado con la finalidad de mejorar las condiciones de transitabilidad de la red vial agrícola existente en el país.

Por último tenemos la Oficina MOPVI Zona sur y es quien representa al ministerio, antes las autoridades locales y demás instituciones públicas y privadas ubicadas en la zona sur del estado Anzoátegui. Su objetivo es la de dirigir y coordinar la planificación y control de las obras de las ciudades del sur del estado Anzoátegui, diseñando y desarrollando planes, programas y estudio de carácter urbanístico, manteniendo la asistencia técnica permanentemente a las municipalidades, promoviendo la participación comunal hacia el logro de un desarrollo armónico y equilibrado de la ciudad.

2.6. UBICACIÓN DE LA INSTITUCIÓN

El Ministerio del Poder Popular para las Obras Públicas y Vivienda dirección estatal Anzoátegui esta ubicado en la Av. Intercomunal Jorge Rodríguez, Sector Las Garzas, Sede MOPVI Lechería Edo. Anzoátegui. (www.mopvi.gob.ve/. 2009).

2.7. OBRA

Es un conjunto de actividades de ingeniería civil (construcción) desarrolladas en un espacio físico definido y que generan un producto final tangible para el uso colectivo. A los efectos del sistema, una “Obra” es una entidad distinta a un “Contrato”. (Mata, L. 2007)

Ejemplos de obra:

- La construcción de una autopista
- La construcción de un gran puente
- La construcción de un edificio
- La construcción de un aeropuerto, sus ampliaciones o sus mejoras

Ejemplos de objetos que NO son obras:

- La adquisición de un helicóptero
- La compra de una flota de autobuses
- El mantenimiento de un tren

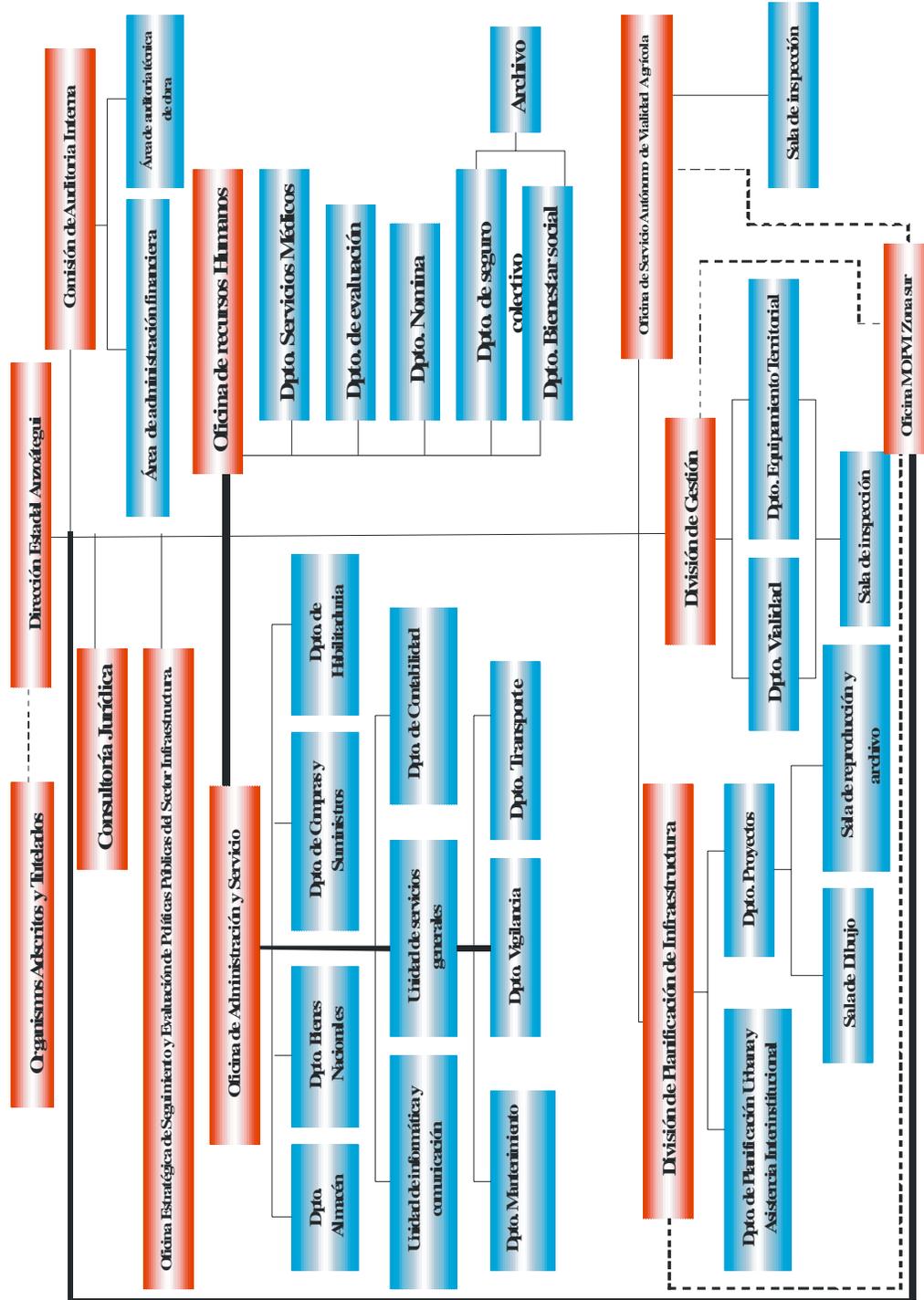


Figura 2.1 Organigrama Estructural de la Dirección Estatal del Ministerio del Poder Popular para las Obras Públicas y Vivienda Anzoátegui. (Fuente: Dima Rodríguez)

2.8. CONTRATO DE OBRA

Se trata de un documento físico por medio del cual una empresa se compromete a construir una obra o parte de la misma, durante un lapso de tiempo y con recursos financieros delimitados. (Mata, L. 2003)

2.8.1. Tipos de Contratos que Maneja el Ministerio

Debido ha como esta conformado el ministerio de de obras públicas y vivienda, este maneja hasta los momentos dos (2) tipos de contratos:

- *Contrato Asociado a una Obra Existente*: el objetivo de este contrato es la de ejecutar una parte de una obra, teniendo como relación una obra.
- *Contrato No asociado a Obra*: es un contrato que por su tamaño, relevancia o naturaleza no forma parte de una gran obra, por esto este contrato es considerado como una obra.

2.9. AVANCE DEL CONTRATO

Se trata de la actualización sistemática de la información sobre la evolución del contrato en el tiempo, a fin de conocer en un momento determinado su estatus físico, administrativo y financiero. (Mata, L. 2003)

2.10. RELACIÓN QUE EXISTE ENTRE LAS FASES DE EJECUCIÓN DE UNA OBRA

La siguiente imagen (ver figura. 2.2) muestra el esquema que permite explicar las relaciones existentes entre Obra, Contrato y Avance. Observe que en todos los casos

CAPÍTULO II: Marco Teórico

se trata de relaciones “de 1 a N”, es decir, por cada obra pueden registrarse N contratos y por cada contrato pueden registrarse N Avances. (Mata, L. 2003)

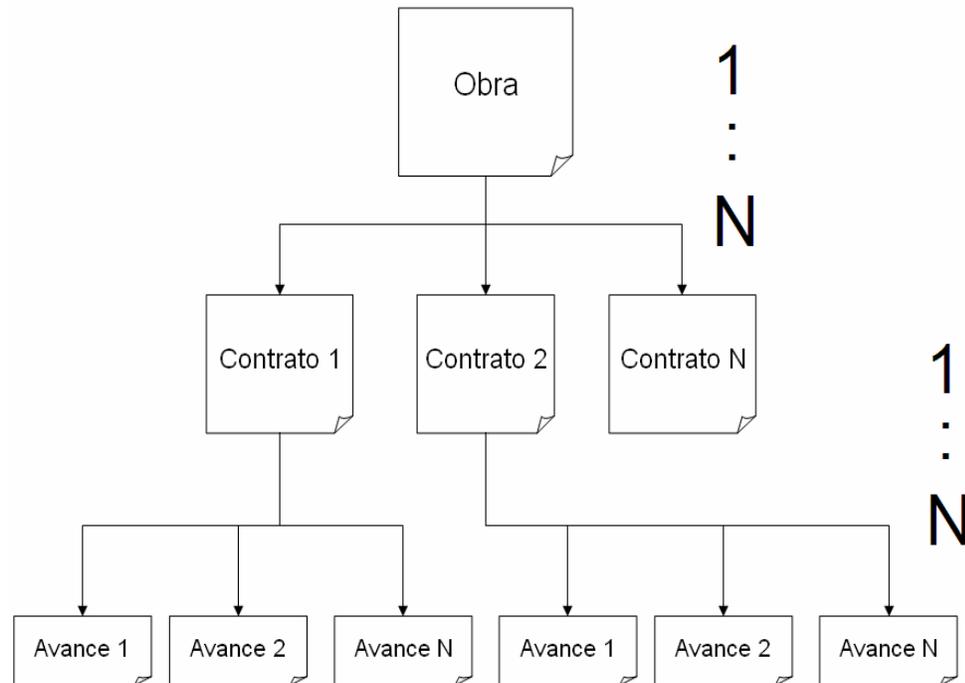


Figura 2.2 Relación entre Obra, Contrato y Avance. (Fuente: Manual del usuario SASEOMINFRA, 2007)

2.11. INSPECCIÓN DE OBRAS

Se refiere a la actividad o acción por lo general permanente, requerida para garantizar que la ejecución de una cierta obra se desarrolle de conformidad con las mejores normas técnicas de trabajo según los planos, especificaciones y demás documentos que constituyen el proyecto correspondiente. Cubre además, el control que es necesario efectuar para que la adquisición de equipos, materiales y servicios, se realice con la suficiente racionalidad y cumplimiento de las especificaciones técnicas pertinentes. (Mata, L. 2003)

2.11.1. Ingeniero Inspector

Profesional de la Ingeniería (Colegiado) especializado en el campo de la naturaleza del proyecto a ejecutar, con experiencia suficiente para ser el responsable de la Inspección de la obra y de controlar que la construcción se realiza cumpliendo cada una de sus requerimientos y etapas, de acuerdo con los planos, las especificaciones y demás documentos del proyecto. El Ingeniero Inspector es el representante del Ente Contratante y generalmente es el único autorizado para comunicar al contratista, las posibles modificaciones (Autorizadas por el contratante) o nuevas propuestas técnicas que modifiquen el proyecto o el presupuesto. (Mata, L. 2003)

2.11.2. Ingeniero Residente

Profesional de la Ingeniería especializado en el campo de la naturaleza de la obra, encargado de dirigir por parte del Contratista, la ejecución, conforme a los planos y especificaciones técnicas establecidas en el proyecto, velando por el mejor aprovechamiento de los equipos, herramientas, recursos humanos adecuados y necesarios; cumpliendo las Normas de Seguridad e Higiene Industrial y de acuerdo a las condiciones establecidas en el contrato suscrito por el Contratista. El Ingeniero Residente es el representante técnico del Contratista en la obra y es el encargado de la planificación, ejecución de la obra y de las actividades de control, tales como calidad, organización del personal, actas, mediciones, valuaciones y demás actos administrativos similares.

2.12. ENTE CONTRATANTE

Organismo Público, Persona Natural o Jurídica, de carácter Público o Privado, encargado de ejecutar una obra de ingeniería, Contratar un servicio o adquirir bienes, bien sea para uso colectivo o particular, mediante el aporte de los recursos financieros

CAPÍTULO II: Marco Teórico

y la contratación de una o varias empresas contratistas, según la magnitud o complejidad del proyecto. Ente que origina la contratación y la avala un Contrato mediante la firma de su representante legal, efectuando los pagos correspondientes. (Mata, L. 2003)

2.13. ENTE EJECUTOR

Entidad encargada de llevar adelante una determinada obra con la aplicación de las medidas de control e inspección. Ejemplo: el ente contratante puede ser una Gobernación de Estado y el ejecutor podría ser una Dirección de un Ministerio, o también un Instituto Estatal y una dependencia regional adscrita a un Ministerio, respectivamente. Generalmente el Contratante y el Ejecutor son la misma persona jurídica. (Mata, L. 2003)

2.14. CONTRATISTA

Empresa Calificada que asume legalmente ante el Ente Contratante, la obligación y responsabilidad de realizar una obra a través de una relación contractual en un lapso establecido y por un monto acordado. Según el tipo de contrato, aporta elementos de trabajo, materiales, equipos, personal y otros recursos auxiliares necesarios y previstos en el presupuesto contratado. (Mata, L. 2003)

2.15. DESARROLLO DE SOFTWARE

La realización de cualquier sistema computarizado, que busca no solo solucionar una necesidad sino perdurar y evolucionar, requiere del uso de ciertos conocimientos teóricos, así como de técnicas de diseño, metodologías y herramientas, que garanticen la calidad del software. (Fairley, R. 1987)

2.15.1. Ingeniería de Software

Es la disciplina tecnológica y administrativa dedicada a la producción sistemática de productos de programación, que son desarrollados y modificados a tiempo y dentro de un presupuesto definido.

2.15.2. Proceso Unificado de Desarrollo de Software

El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado es un marco de desarrollo de software iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado Racional o simplemente RUP.

El PU se define como un proceso dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental. Está basado en componentes (el sistema software en construcción está formado por componentes software) y utiliza el nuevo estándar de modelado visual UML para describir los distintos modelos generados durante el desarrollo del proceso, es decir, para preparar todos los esquemas de un sistema software. En este proceso, se distinguen cuatro fases fundamentales, dentro de cada una de las cuales se plantean una serie de flujos de trabajo iterativos que permiten la evolución del proyecto de manera incremental (ver figura 2.3). (Ivar, J.; Grady, B. y James, R. 2000)

2.15.3. Iterativo e Incremental

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado

CAPÍTULO II: Marco Teórico

un *incremento* del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación y Prueba.

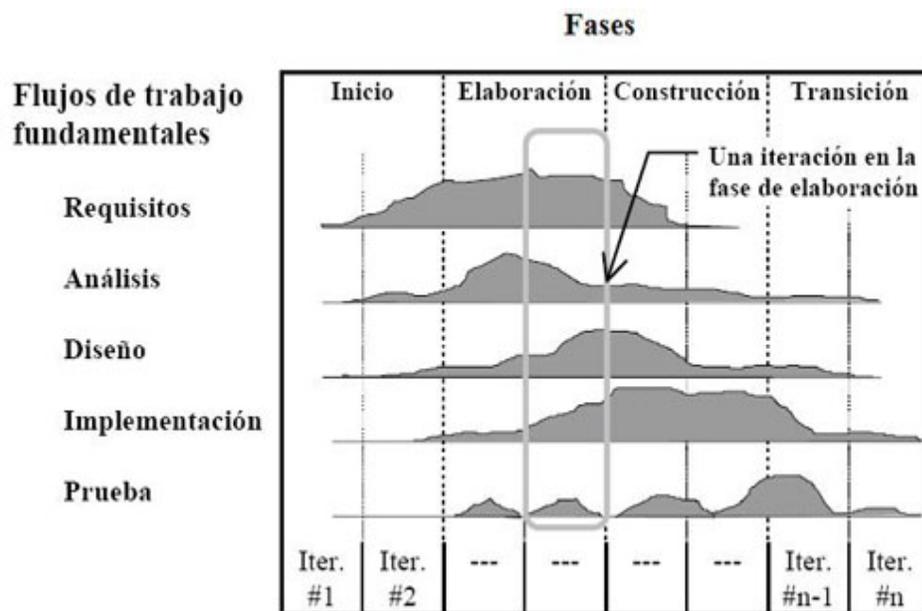


Figura 2.3 *Proceso Unificado de Desarrollo de Software*. (Fuente: http://www.espilebarbier.com/imagenes/Cuerpo/Graficas/fases_es.jpg)

2.15.4. Dirigido por caso de uso

En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc.

2.15.5. Centrado en la Arquitectura

El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema.

2.15.6. Fases del Proceso Unificado de Desarrollo de Software

2.15.6.1. Fase de Inicio

En la fase de inicio se define el negocio y sus objetivos, así como la factibilidad del proyecto, se delimita el ámbito, se describe la arquitectura candidata posible y se identifican los riesgos críticos. El énfasis recae en los requerimientos.

2.15.6.2. Fase de Elaboración

En la fase de elaboración se capturan los requerimientos restantes (aproximadamente el 80% del mismo), se establece la arquitectura base para la construcción y se monitorean los riesgos. Se enfocan en los requerimientos, el análisis y el diseño.

2.15.6.3. Fase de construcción

En la fase de construcción se realiza el sistema según una arquitectura estable, se genera una versión beta del mismo (operativa) y se crea material de usuario. El énfasis está en la implementación y la prueba.

2.15.6.4. Fase de Transición

En la fase de transición se actualiza el entorno en el que va a funcionar el sistema (sistema operativo, hardware, etc.), se crean los manuales del sistema, se

ajusta el software al entorno de trabajo, se corrigen los errores y deficiencias y se genera una versión formal del sistema.

2.15.7. Flujo de Trabajo del Proceso Unificado

El PU consta de varios flujos de trabajo. En cada incremento o fase incremental se ejecuta parte de estos flujos de trabajo, variando el peso de cada uno de ellos en cada incremento.

2.15.7.1. Captura de Requisitos

El propósito fundamental del flujo de trabajo de los requisitos es guiar el desarrollo hacia el sistema correcto.

Hay diferentes puntos de partida para la captura de requisitos. En algunos casos comenzamos haciendo un *modelo del negocio* o partimos de uno ya desarrollado. En otros casos si es un sistema acotado que no da soporte al negocio podemos partir de un modelo de objetos sencillo como un modelo del dominio.

2.15.7.2. Análisis

Durante el análisis, analizamos los requisitos que se describieron en la captura de requisitos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que nos ayude a estructurar el sistema entero, incluyendo su arquitectura.

2.15.7.3. Diseño

Durante el diseño modelamos el sistema y su arquitectura para que soporte los requisitos funcionales y no funcionales. Una entrada esencial al diseño es el modelo de análisis.

El diseño es el centro de atención al final de la fase de elaboración y comienzo de las iteraciones de construcción.

2.15.7.4. Implementación

En la implementación empezamos con el resultado del diseño e implementamos el sistema en término de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables, y similares.

La implementación es el centro durante las iteraciones de construcción, aunque también se lleva a cabo trabajo de implementación durante la fase de elaboración, para crear la línea base ejecutable de la arquitectura, y durante la fase de transición para tratar defectos tardíos.

2.15.7.5. Prueba

El objetivo de esta fase es asegurar que los requisitos se han cumplido y que el software está disponible para los usuarios finales. Por eso esta fase está dirigida por la retroalimentación de los usuarios, a partir de la información que se deduzca de la versión beta del sistema en funcionamiento.

2.16. EL LENGUAJE UNIFICADO DE MODELADO (UML)

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

UML también intenta solucionar el problema de propiedad de código que se da con los desarrolladores, al implementar un lenguaje de modelado común para todos los desarrollos se crea una documentación también común, que cualquier desarrollador con conocimientos de UML será capaz de entender, independientemente del lenguaje utilizado para el desarrollo.

Este lenguaje nos ayuda a interpretar grandes sistemas mediante gráficos o mediante texto obteniendo modelos explícitos que ayudan a la comunicación durante el desarrollo ya que al ser estándar, los modelos podrán ser interpretados por personas que no participaron en su diseño (e incluso por herramientas) sin ninguna ambigüedad. En este contexto, UML sirve para especificar modelos concretos, no ambiguos y completos.

Debido a su estandarización y su definición completa no ambigua, y aunque no sea un lenguaje de programación, UML se puede conectar de manera directa a lenguajes de programación como Java, C++ o Visual Basic, esta correspondencia permite lo que se denomina como ingeniería directa (obtener el código fuente partiendo de los modelos) pero además es posible reconstruir un modelo en UML partiendo de la implementación, o sea, la ingeniería inversa. (Booch, B y Rumbaugh, J. 1996)

2.16.1. Diagramas UML

Un diagrama es la representación gráfica de un conjunto de elementos con sus relaciones. En concreto, un diagrama ofrece una vista del sistema a modelar. Para poder representar correctamente un sistema, UML ofrece una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas.

2.16.2. Diagrama de Casos de Uso

Un diagrama de casos de uso explica gráficamente un conjunto de casos de un sistema, los actores y la relación entre éstos y los casos de uso (ver figura 2.4). Estos últimos se muestran en óvalos y los actores son figuras estilizadas. Hay líneas de comunicaciones entre los casos y los actores; las flechas indican el flujo de la información o el estímulo.

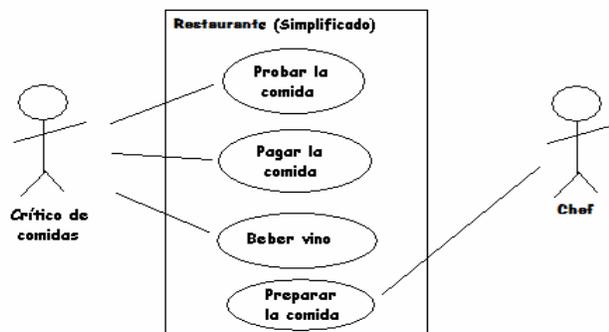


Figura 2.4 Diagrama de Casos de Uso. (Fuente:

http://upload.wikimedia.org/wikipedia/commons/3/3c/UML_diagrama_caso_de_uso.png)

2.16.2.1. Diagrama de Clase

Muestran un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Estos diagramas son los más comunes en el modelado de sistemas orientados a objetos y cubren la vista de diseño estática o la vista de procesos estática.

CAPÍTULO II: Marco Teórico

Estos diagramas sirven para visualizar las relaciones existentes entre las distintas clases y la forma en que colaboran unas con otras. Las relaciones entre las distintas clases son las relaciones comunes existentes en UML aunque con matices, una asociación se traduce como que desde los objetos de una clase se puede acceder a los objetos de otra. Una dependencia se puede visualizar como que la clase utilizada es un parámetro de un método de la clase que la utiliza. Una generalización se traduce como una herencia entre clases (ver figura 2.5).

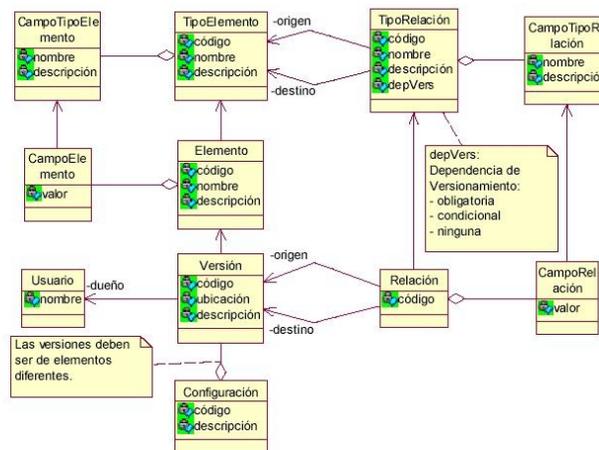


Figura 2. Diagrama de Clases UML para un Modelo Genérico para la Gestión de Configuración del Software

Figura 2.5 Diagrama de Clase. (Fuente:

<http://www.histaintl.com/soluciones/configuracion/imagenes/diagrama.jpg>)

2.16.2.2. Diagrama de Secuencia

El diagrama de secuencia de un sistema es una representación que muestra, en determinado escenario de un caso de uso, los eventos generados por actores externos, su orden y los eventos del sistema (ver figura 2.6). A todos los sistemas se les trata como una caja negra; los diagramas se centran en evento que trascienden las fronteras del sistema y que fluyen de los actores a los sistemas.

2.16.2.3. Diagrama de Objeto

Muestran un conjunto de objetos y sus relaciones, son como fotos instantáneas de los diagramas de clases y cubren la vista de diseño estática o la vista de procesos estática desde la perspectiva de casos reales o prototípicos (ver figura 2.7).

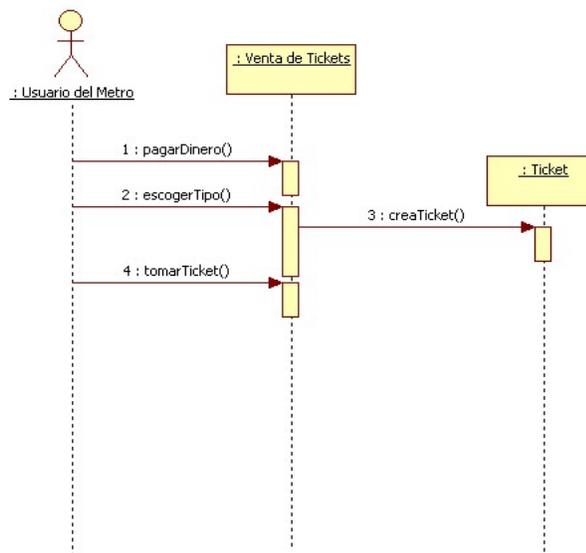


Figura 2.6 Diagrama de Secuencia.(Fuente:

<http://synergix.files.wordpress.com/2008/07/comprarticketmetro.jpg>)

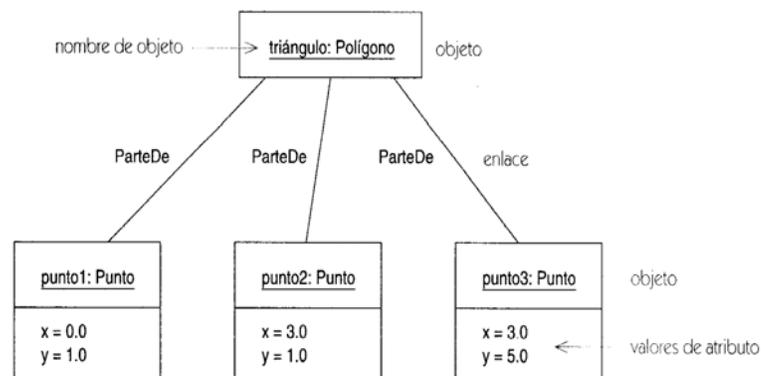


Figura 2.7 Diagrama de Objeto.(Fuente: <http://objectwiz.org/object-diagram>)

2.16.2.4. Diagrama de Colaboración

Los diagramas de colaboración son un tipo de diagramas de interacción. Constan de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar unos objetos a otros. (Ver figura 2.8). Cubren la vista dinámica del sistema. Los diagramas de colaboración muestran la organización estructural de los objetos que envían y reciben mensajes.

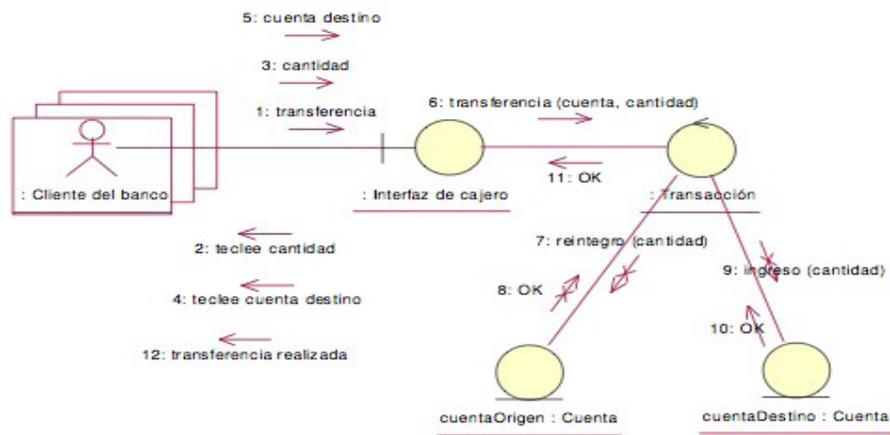


Figura 2.8 *Diagrama de Colaboración* (Fuente:

http://solincastillo.net63.net/sistemas/images/analisis/diagrama_de_colaboracion.jpg)

2.16.2.5. Diagrama de Estado

Muestran una máquina de estados compuesta por estados, transiciones, eventos y actividades. Estos diagramas cubren la vista dinámica de un sistema y son muy importantes a la hora de modelar el comportamiento de una interfaz, clase o colaboración (ver figura 2.9).

CAPÍTULO II: Marco Teórico

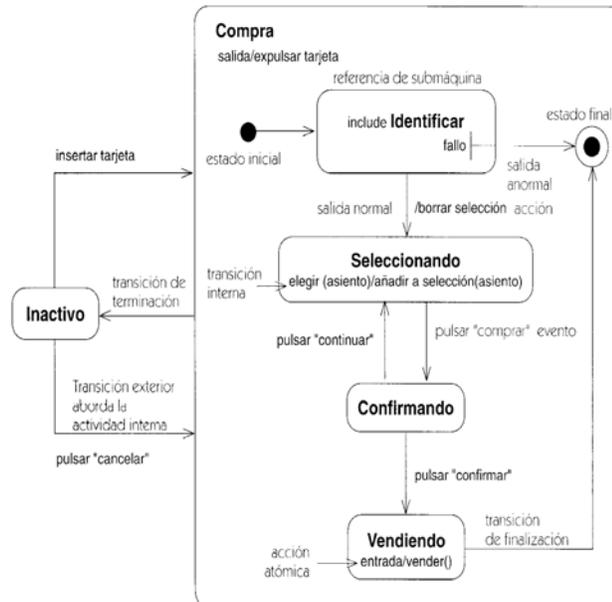


Figura 2.9 Diagrama de Estado. (Fuente: <http://org23.zorpia.com/0/3010/19267377.931bc7.jpg>)

2.16.2.6. Diagrama de Actividades

Son un tipo especial de diagramas de estados que se centra en mostrar el flujo de actividades dentro de un sistema. Los diagramas de actividades cubren la parte dinámica de un sistema y se utilizan para modelar el funcionamiento de un sistema resaltando el flujo de control entre objetos (ver figura 2.10).

2.16.2.7. Diagrama de Componente

Muestra la organización y las dependencias entre un conjunto de componentes. Cubren la vista de la implementación estática y se relacionan con los diagramas de clases ya que en un componente suele tener una o más clases, interfaces o colaboraciones (ver figura 2.11).

CAPÍTULO II: Marco Teórico

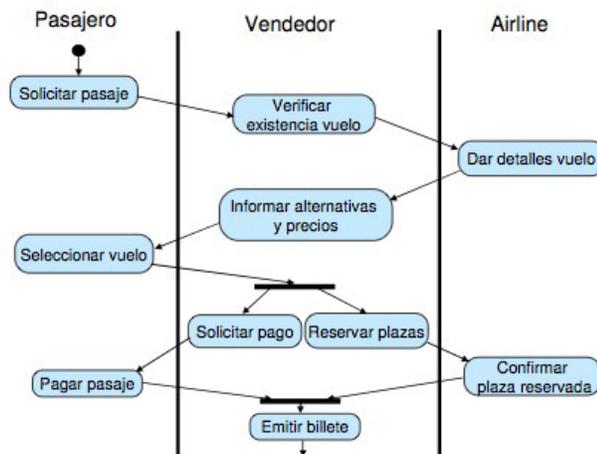


Figura 2.10 Diagrama de Actividades. (Fuente:

<http://www.luiscorbete.com/site/tuto/uml/uml/diagramaActividad/act7.png>)

2.16.2.8. Diagrama de Despliegue

Representan la configuración de los nodos de procesamiento en tiempo de ejecución y los componentes que residen en ellos. Muestran la vista de despliegue estática de una arquitectura y se relacionan con los componentes ya que, por lo común, los nodos contienen uno o más componentes (ver figura 2.12).

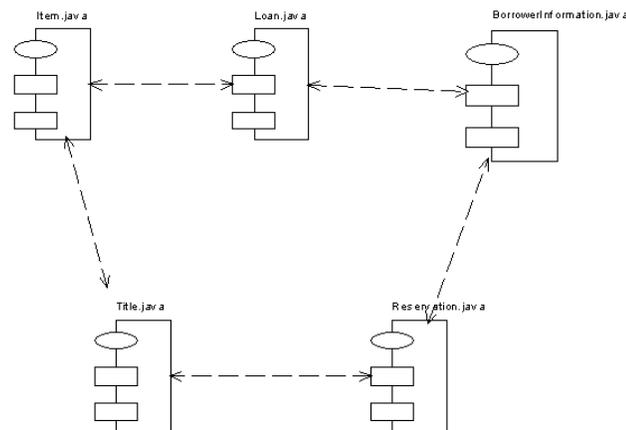


Figura 2.11 Diagrama de Componente. (Fuente:

<http://profesores.fib.unam.mx/carlos/aydoo/compos.html>)

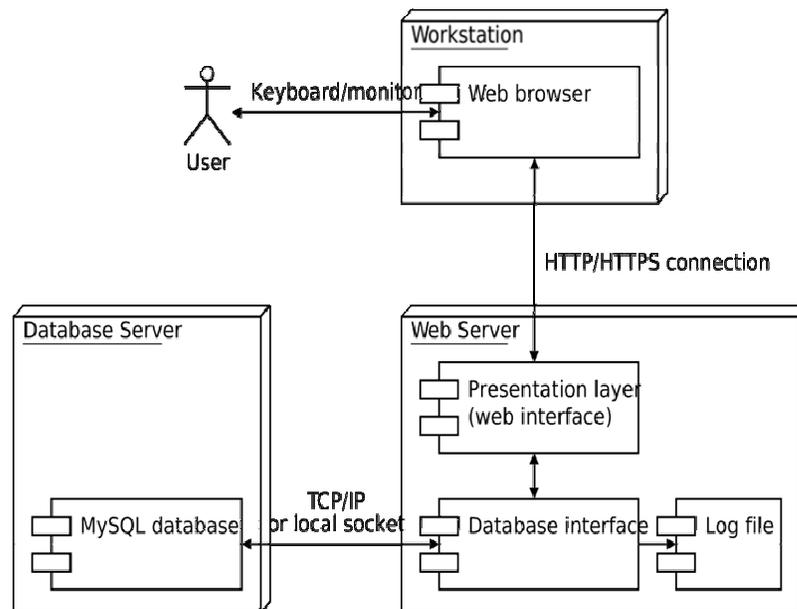


Figura 2.12 Diagrama de Despliegue. (Fuente: http://es.wikipedia.org/wiki/Archivo:UML_Deployment_Diagram.svg)

2.17. PROGRAMACIÓN ORIENTADA A OBJETO (POO)

La **programación orientada a objetos** o **POO** (**OOP** según sus siglas en inglés) es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos. (http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos. 2010)

2.17.1. Características de la programación orientada a objeto

Las características siguientes son las más importantes:

CAPÍTULO II: Marco Teórico

- **Abstracción:** Cada objeto en el sistema sirve como modelo de un agente abstracto que puede realizar trabajo, informar y cambiar su estado, y comunicarse con otros objetos en el sistema sin revelar cómo se implementan estas características. Los procesos, las funciones o los métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción.
- **Encapsulamiento:** Significa reunir a todos los elementos que pueden considerarse pertenecientes a una misma entidad, al mismo nivel de abstracción. Esto permite aumentar la cohesión de los componentes del sistema. Algunos autores confunden este concepto con el principio de ocultación, principalmente porque se suelen emplear conjuntamente.
- **Principio de ocultación:** Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una interfaz a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.
- **Polimorfismo:** comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia

producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama asignación tardía o asignación dinámica. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.

- **Herencia:** las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que re-implementar su comportamiento. Esto suele hacerse habitualmente agrupando los objetos en clases y estas en árboles o enrejados que reflejan un comportamiento común. Cuando un objeto hereda de más de una clase se dice que hay herencia múltiple.

2.18. LENGUAJE DE PROGRAMACIÓN JAVA

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

(http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java, 2010)

2.18.1. Características

- **Simple:** Java ofrece toda la funcionalidad de un lenguaje potente, pero sin las características menos usadas y más confusas de éstos. C++ es un lenguaje que adolece de falta de seguridad, pero C y C++ son lenguajes más difundidos, por ello Java se diseñó para ser parecido a C++ y así facilitar un rápido y fácil aprendizaje.
- **Orientado A Objetos:** Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo.
- **Distribuido:** Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como *http* y *ftp*. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales.
- **Robusto:** Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo
- **Arquitectura Neutral:** Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará. Cualquier máquina que tenga el sistema de ejecución (*run-time*) puede ejecutar ese código objeto, sin importar en modo alguno la máquina en que ha sido generado. Actualmente existen sistemas run-time para Solaris 2.x, SunOs 4.1.x, Windows 95, Windows

CAPÍTULO II: Marco Teórico

NT, Linux, Irix, Aix, Mac, Apple y probablemente haya grupos de desarrollo trabajando en el porting a otras plataformas.

- **Seguro:** La seguridad en Java tiene dos facetas. En el lenguaje, características como los punteros o el *casting* implícito que hacen los compiladores de C y C++ se eliminan para prevenir el acceso ilegal a la memoria. Cuando se usa Java para crear un navegador, se combinan las características del lenguaje con protecciones de sentido común aplicadas al propio navegador.
- **Portable:** Más allá de la portabilidad básica por ser de arquitectura independiente, Java implementa otros estándares de portabilidad para facilitar el desarrollo. Los enteros son siempre *enteros* y además, enteros de 32 bits en complemento a 2. Además, Java construye sus interfaces de usuario a través de un sistema abstracto de ventanas de forma que las ventanas puedan ser implantadas en entornos Unix, Pc o Mac.
- **Interpretado:** El intérprete Java (sistema run-time) puede ejecutar directamente el código objeto. Enlazar (linkar) un programa, normalmente, consume menos recursos que compilarlo, por lo que los desarrolladores con Java pasarán más tiempo desarrollando y menos esperando por el ordenador.
- **Multithreaded (MultiHilos):** Los threads (a veces llamados, procesos ligeros), son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar los threads contruidos en el lenguaje, son más fáciles de usar y más robustos que sus homólogos en C o C++.
- **Dinámico:** Java se beneficia todo lo posible de la tecnología orientada a objetos. Java no intenta conectar todos los módulos que comprenden una aplicación hasta el tiempo de ejecución. Las librerías nuevas o actualizadas no paralizarán las aplicaciones actuales

2.19. BASE DE DATOS

Una Base de Datos es un conjunto de datos relacionados entre sí. Siendo lo datos hechos conocidos que pueden registrarse y que tienen un significado implícito. Ya que, representa algún aspecto del mundo real, en ocasiones llamado mini mundo o universo de discurso. Las modificaciones del mini mundo se reflejan en la base de datos.

Los Datos son almacenados de forma integrada y compartida pues son un conjunto de archivos independientes, donde se puedan acceder a la misma fracción de la Base de Dato, incluso al mismo tiempo y utilizarla con fines diferentes.(Charlis, R. 2001)

2.19.1. Ventajas de las Bases de Datos

La utilización de bases de datos como plataforma para el desarrollo de sistemas de aplicación en las organizaciones se ha incrementado notablemente en los últimos años, se debe a las ventajas que ofrece su utilización, algunas de las cuales se comentarán a continuación:

- **Globalización de la información:** permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- **Eliminación de información inconsistente:** debe haber coherencia, si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deberán hacerse a todas las copias del archivo de facturas.
- **Permite compartir información:** la información almacenada puede ser compartida por un gran número de usuarios.

CAPÍTULO II: Marco Teórico

- Permite mantener la integridad en la información: la integridad de la información es una de sus cualidades altamente deseable y tiene por objetivo que sólo se almacena la información correcta.
- Independencia de datos: el concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo de Sistemas de Bases de Datos. La independencia de datos implica un divorcio entre programas y datos.

2.19.2. Sistema de Manejador de Base de Datos

Es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

2.19.3. Objetivos del SMBD

Dentro de los principales objetivos de un sistema manejador de base de datos se podrá encontrar.

- Independencia de los datos.
- Seguridad.
- Reserva.
- Integridad.
- Respaldo.
- Recuperación.
- Redundancia mínima.
- Consistencia.
- Capacidad de auditoria.
- Control de concurrencia.

- Desempeño.

2.19.4. Diseño de Una Base de Datos

Existen distintos modos de organizar la información y representar las relaciones entre los datos en una base de datos. Los Sistemas administradores de bases de datos convencionales usan uno de los tres modelos lógicos de bases de datos para hacer seguimiento de las entidades, atributos y relaciones. Los tres modelos lógicos principalmente de bases de datos son el jerárquico, de redes y el relacional. Cada modelo lógico tiene ciertas ventajas de procesamiento y también ciertas ventajas de negocios.

Modelo Jerárquico de Datos: es una clase de modelo lógico de bases de datos que tiene una estructura arborescente. Un registro subdivide en segmentos que se interconectan en relaciones padre e hijo y muchos más. Los primeros sistemas administradores de bases de datos eran jerárquicos. Puede representar dos tipos de relaciones entre los datos: relaciones de uno a uno y relaciones de uno a muchos.

Modelo de Datos en Red: es una variación del modelo de datos jerárquico. De hecho las bases de datos pueden traducirse de jerárquicas a en redes y viceversa con el objeto de optimizar la velocidad y la conveniencia del procesamiento. Mientras que las estructuras jerárquicas describen relaciones de muchos a muchos.

Modelo Relacional de Datos: es el más reciente de estos modelos, supera algunas de las limitaciones de los otros dos anteriores. El modelo relacional de datos representa todos los datos en la base de datos como sencillas tablas de dos dimensiones llamadas relaciones. Las tablas son semejantes a los archivos planos, pero la información en más de un archivo puede ser fácilmente extraída y combinada.

2.20. MANEJADOR DE BASE DE DATOS POSTGRES

El manejador de bases de datos PostgreSQL es considerado junto con MySQL como los más robustos en el mundo del software libre (fue liberado bajo la licencia BSD), su logo es representado por un elefante en clara referencia a la capacidad de memoria de estos animales. Es el resultado de una larga evolución comenzada con el proyecto Ingres en la Universidad de Berkeley. El líder del proyecto, Michael Stonebraker abandonó Berkeley para comercializar Ingres en 1982, pero finalmente regresó a la academia. Tras su retorno a Berkeley en 1985, Stonebraker comenzó un proyecto post-Ingres para resolver los problemas con el modelo de base de datos relacional que habían sido aclarados a comienzos de los años 80. El principal de estos problemas era la incapacidad del modelo relacional de comprender "tipos", es decir, combinaciones de datos simples que conforman una única unidad. Actualmente estos son llamados objetos.

El proyecto resultante, llamado Postgres, era orientado a introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos. Estas funcionalidades incluían la habilidad de definir tipos, pero también la habilidad de describir relaciones - las cuales hasta ese momento eran ampliamente utilizadas pero mantenidas completamente por el usuario. En Postgres la base de datos "comprendía" las relaciones y podía obtener información de tablas relacionadas utilizando reglas.

Comenzando en 1986, el equipo liberó una serie de documentos describiendo la base del sistema y en 1988 poseían un prototipo funcional. La primera versión fue liberada a un pequeño grupo de usuarios en junio de 1989, seguido por la segunda posteriormente con un sistema de reglas reescrito en junio de 1990. Para la tercera versión, liberada en 1991, el sistema de reglas fue reescrito nuevamente, pero también agregó soporte para múltiples administradores de almacenamiento y un sistema de consultas mejorado. Hacia 1993, el manejador de bases de datos Postgres había

CAPÍTULO II: Marco Teórico

crecido inmensamente en popularidad y poseía una demanda asfixiante de nuevas funcionalidades. Tras liberar la cuarta versión, la cual era una simple versión de limpieza, el proyecto fue abandonado.

A pesar de que el proyecto Postgres hubiese finalizado oficialmente, la licencia BSD bajo la cual Postgres había sido liberado permitió a desarrolladores de código abierto el obtener una copia del código para continuar su desarrollo. (Lockhart, T. 1996)

2.21. JAVA 2D

La API de Java 2D es un conjunto de clases avanzadas para gráficos 2D y de imágenes, que abarca la línea de arte, texto y las imágenes en un único modelo. La API proporciona un amplio apoyo para la imagen de composición y canal alfa imágenes, un conjunto de clases para proporcionar espacio de color precisa definición ya la conversión, y un rico conjunto de visualización de imágenes orientado a los operadores. (<http://java.sun.com/products/java-media/2D/index.jsp>)

2.22. DECRETO SOFTWARE LIBRE N° 3390

El Decreto de software libre (N° 3390) dictado por el ejecutivo Venezolano el 28 de Diciembre, estableció el uso prioritario del software libre en toda la administración pública, incluyendo ministerios y oficinas gubernamentales, además de fomentar la investigación y el desarrollo de software libre.

Artículo 1. La Administración Pública Nacional empleará prioritariamente Software Libre desarrollado con Estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. A tales fines, todos los órganos y entes de la Administración Pública Nacional iniciarán los procesos de migración gradual y progresiva de éstos

CAPÍTULO II: Marco Teórico

hacia el Software Libre desarrollado con Estándares Abiertos. (Gaceta oficial N° 38.095.2004)

CAPÍTULO III: FASE DE INICIO

3.1. INTRODUCCIÓN

A continuación se iniciará la fase de inicio del desarrollo del sistema, esto comprenderá la primera etapa del Proceso Unificado de Desarrollo del Software. Entre los objetivos de esta fase se encuentran el estudio de la funcionalidad del sistema SISOVE (Sistema para el Seguimiento de Obras viales y de Edificaciones), la definición de su arquitectura básica y la planificación del proyecto, considerando los recursos disponibles y partiendo de diagramas que bosquejan los requisitos, los cuales son la base para la futura definición de la arquitectura del software.

3.2. BREVE DESCRIPCIÓN DEL SISTEMA

El objetivo principal de la aplicación SISOVE (Sistema para el Seguimiento de Obras viales y de Edificaciones), es permitir el manejo automatizado de toda la información relativa al avance de las obras llevadas por el ministerio de obras públicas y vivienda en el estado Anzoátegui. Se denomina Sistema para el seguimiento de obras porque dicho sistema se encarga de mostrar información relevante de las obras llevadas a cabo pero de forma gráfica y dinámica, la información es suministrada por los distintos departamentos relacionados con el área de vialidad y edificaciones, por medio del sistema SISOVE y consultada por el mismo por las personas que requieran la información.

3.3. REQUERIMIENTOS DEL SOFTWARE

El punto de arranque en un proyecto de desarrollo de software es la descripción de los requerimientos del Sistema, en el cual se plasman las funcionalidades que demandan los futuros usuarios de la aplicación, en el mismo se describen los requerimientos funcionales y no funcionales del sistema que se desea. Los requerimientos funcionales (ver tabla 3.1) describen al sistema en términos de entrada-salida, mientras que los no-funcionales (ver tabla 3.2), en términos de cualidades deseables del sistema.

Los requerimientos funcionales se estructuran mediante casos de uso, ya que ellos proporcionan un medio intuitivo y sistemático para capturarlos.

Tabla 3.1. Requisitos Funcionales

N° de Requisito	Funcionalidad
1	Desarrollar un módulo en donde se pueda visualizar la imagen de un mapa con puntos que hagan referencia a una obra y así poder tener información de dicha obra de manera dinámica y rápida.
2	Crear un módulo encargado de capturar los datos de las obras, contratos y avances de la obras. Necesarios para llevar a cabo el control y seguimiento de las obras.
3	Desarrollar un módulo en donde se pueda consultar datos de las obras, contratos y avances necesarios para determinar su estado.
4	Restringir el acceso al sistema a usuarios no autorizados, y dar ciertos privilegios a los administradores del sistema.
5	Permitir cargar las imágenes de los mapas en la base de datos, no perdiendo así su referencia dando una mayor seguridad a la hora acceder a ellas
6	Desarrollar un módulo donde se pueda crear los puntos y hacer la referencia del mismo a las obras creadas.

Fuente: Dima Rodríguez

Tabla 3.2. Requisitos No Funcionales

REQUISITOS	DESCRIPCIÓN
Apariencia	El diseño de la interfaz del usuario debe poseer una estructura amigable para que resulte de fácil entendimiento para los usuarios.
Utilizable	El aprendizaje del uso debe ser suficientemente sencillo por parte de personas no especializadas.
Confidencial	La información no debe estar asequible, excepto por usuarios autorizados.
Plataforma Software	Puede ser utilizado en: cualquier plataforma Windows o Linux.
Integro	Los datos que contiene el sistema deben de ser los mismos que se han de transmitir.

Fuente: Dima Rodríguez

3.4. MODELO DE DOMINIO

El modelo del dominio captura los aspectos más importantes en el contexto del sistema y los representa como objetos o clases del negocio. En el diagrama de dominio (Figura 3.1), se representan el esquema estructural de la estación de flujo y las actividades realizadas por parte de su personal sobre las mismas.

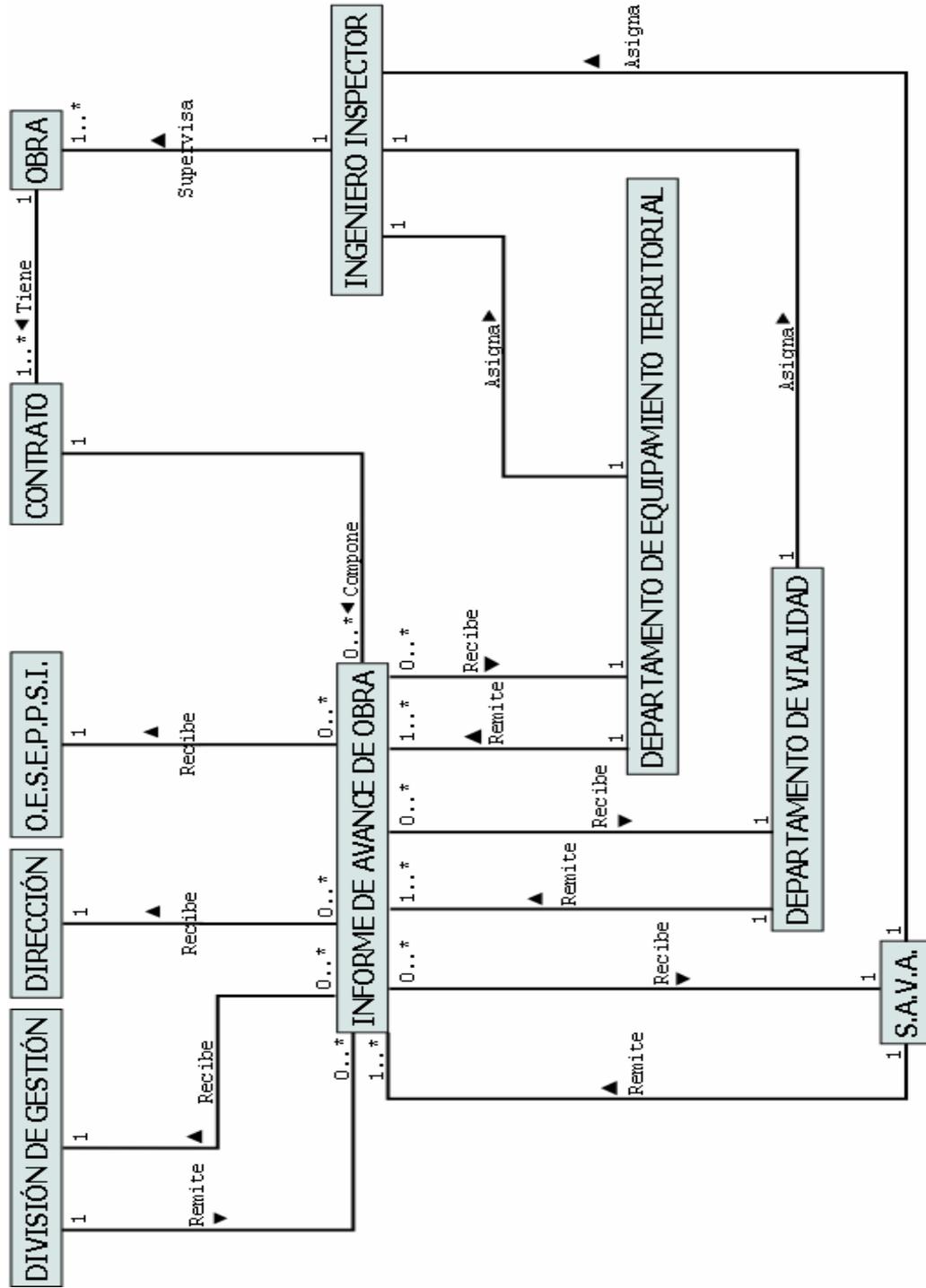


Figura 3.1 Modelo de Dominio (Fuente: Dima Rodríguez)

3.4.1. Terminología Empleada

La terminología empleada en el diagrama de modelo de dominio será descrita en la siguiente tabla 3.3

Tabla 3.3. Descripción de las clases del dominio (1/3)

TÉRMINO	DEFINICIÓN
<p style="text-align: center;">OBRA</p>	<p>Es un conjunto de actividades de ingeniería civil (construcción) desarrolladas en un espacio físico definido y que generan un producto final tangible para el uso colectivo.</p>
<p style="text-align: center;">CONTRATO</p>	<p>Documento físico por medio del cual una empresa se compromete a construir una obra o parte de la misma. Varios contratos podrían estar asociados a una misma obra.</p>
<p style="text-align: center;">OFICINA ESTRATÉGICA DE SEGUIMIENTO Y EVALUACIÓN DE POLÍTICAS PÚBLICAS DEL SECTOR INFRAESTRUCTURA. (O.E.S.E.P.P.S.I.)</p>	<p>Realiza seguimiento y evaluación de políticas públicas del Ministerio y entes adscritos, a través de la implementación y/o aplicación de mecanismos y herramientas que permitan realizar el proceso de toma de decisiones en el nivel estratégico.</p>

Fuente: Dima Rodríguez

Tabla 3.4. Descripción de las clases del dominio (2/3)

TÉRMINO	DEFINICIÓN
DIRECCIÓN	Ejercer la rectoría mediante la formulación, adopción, seguimiento, evaluación de las políticas, estrategias, programas y proyectos en los Sectores de Ordenación del Territorio Urbanístico y Transporte.
DIVISIÓN GESTION	Dirige, coordina, controla y desarrolla actividades orientadas en la ejecución e inspección de obras públicas de Equipamiento y vialidad, remitiendo toda esa información a la dirección por medio de informes.
INFORME DE AVANCE DE OBRA	Se trata de la actualización sistemática de la información sobre la evolución del contrato en el tiempo, a fin de conocer en un momento determinado su estatus físico, administrativo y financiero.
INGENIERO INSPECTOR	Profesional de la ingeniería (Colegiado) responsable de la inspección de las obra y de controlar que la construcción se realiza cumpliendo cada una de sus requerimientos y etapas, de acuerdo con los planos, las especificaciones y demás documentos del proyecto. Cada departamento asigna su ingeniero, ya que, son de distintas naturaleza

Fuente: Dima Rodríguez

Tabla 3.5. Descripción de las clases del dominio (3/3)

TÉRMINO	DEFINICIÓN
DEPARTAMENTO DE EQUIPAMIENTO TERRITORIAL	Tramita la contratación de obras de equipamiento e inspecciona la ejecución de las mismas por medio de ingenieros inspectores que se encargaran de velar que todos los parámetros sean cumplidos de manera estricta.
DEPARTAMENTO DE VIALIDAD	Supervisa y coordina las obras de vialidad principal del estado. Así como también de inspeccionar estas obras por medio de los ingenieros inspectores.
SERVICIO AUTONOMO VIALIDAD AGRICOLA (S.A.V.A)	Coordina, supervisa y fiscaliza mediante los ingenieros inspectores, las obras a realizar en las vías agrícolas, aprobadas y contratadas a nivel central.

Fuente: Dima Rodríguez

3.5. DIAGRAMA DE CASO DE USO

Los casos de uso han sido adoptados casi universalmente para la captura de requisitos de software en general, durante el desarrollo de este proyecto tomaran una gran relevancia ya que esta herramienta se utilizará para dirigir el proceso de desarrollo en su totalidad.

Normalmente, un sistema posee muchos tipos de usuario. Cada usuario es representado por un actor. Los actores utilizan el sistema interactuando con los casos de usos.

3.5.1. Identificación de Actores

Los actores no son las abstracciones del sistema. Ellos representan los usuarios externos que pueden usar e interactuar con el sistema que se está diseñando.

- **Usuario:** Es la persona interesada en “visualizar las imágenes de los mapas”, ya que, en estos se encuentran los puntos que hacen referencia a las obras, por ende, podrá “consultar” los datos relacionadas a esta, y para realizar dichas operaciones el usuario debe de estar registrado para tener “Acceso al Sistema”. Este actor necesita el sistema para mantenerse informado del avance de las obras de forma rápida y sencilla.
- **Administrador:** Un Administrador de Avance de Obra es la persona encargada de “Cargar Imagen de Mapa”, bien sea de vialidad agrícola, de las principales vías o por municipio, para luego “Pintar punto de Referencia a Obra”, la cual, serán el enlace de manera visual a la obra. Luego se ha de “Gestionar Obra”, en el cual están inmerso los contratos y avances de dicha obra. El administrado puede “consultar obra” y los datos relacionados con la misma para verificar que se haya ingresado bien la data o simplemente obtener información. También podrá “Visualizar Mapa” de manera de chequear tanto la carga de la imagen como la información contenida en los puntos. Este actor maneja información muy importante, por ende, debe estar registrado como administrador para poder acceder al sistema de forma casi total. Este actor necesita el sistema para mantener la data actualizada de forma más eficiente minimizando los tiempos de espera y errores.
- **Súper Usuario:** A parte de poder realizar todo lo que hace el administrador también es el encargado de configurar el sistema para que éste se conecte a la base de datos por medio de la red local.

CAPÍTULO III: Fase de Inicio

- **Manejador de Base de Datos:** Se considera un actor debido a que interactúa de forma directa con el sistema, y toda la información que el sistema a de manipular está contenida en la base de datos distribuida de manera relacionada en tablas.
- **Archivo de Configuración:** Considerado un actor debido a que interactúa de forma directa con el sistema, este se encarga de contener la información necesaria para la conexión entre en sistema y el manejador de base de datos.

3.5.2. Descripción de Caso de Uso de Contexto

Para describir las acciones del sistema es importante el utilizar los casos de uso, ya que, muestra el punto de vista de los usuarios que utilizaran el sistema, la idea es involucrarlo en las etapas iniciales del análisis y diseño del sistema. En la figura 3.2 se observan siete (7) casos de uso: Visualizar puntos, Visualizar Mapa, Consultar Obra, Cargar Mapa, Gestionar Obra, Crear Punto de Referencia a Obra. Y Configurar Sistema.

Caso de Uso: **Visualizar Mapas**

Actores: Usuario, Administrador, Súper Usuario, Manejador de Base de Datos.

Descripción: Permite al usuario, al administrador y al Súper Usuario visualizar las imágenes de los mapas previamente cargado a la base de datos del sistema por el manejador de base de datos y que sirva de referencia para la ubicación geográfica de la obra. Dependiendo del tipo de consulta a realizarse, se mostrarán el mapa de vialidad agrícola, de las principales vías o el mapa en donde se hará referencia a los municipios del estado.

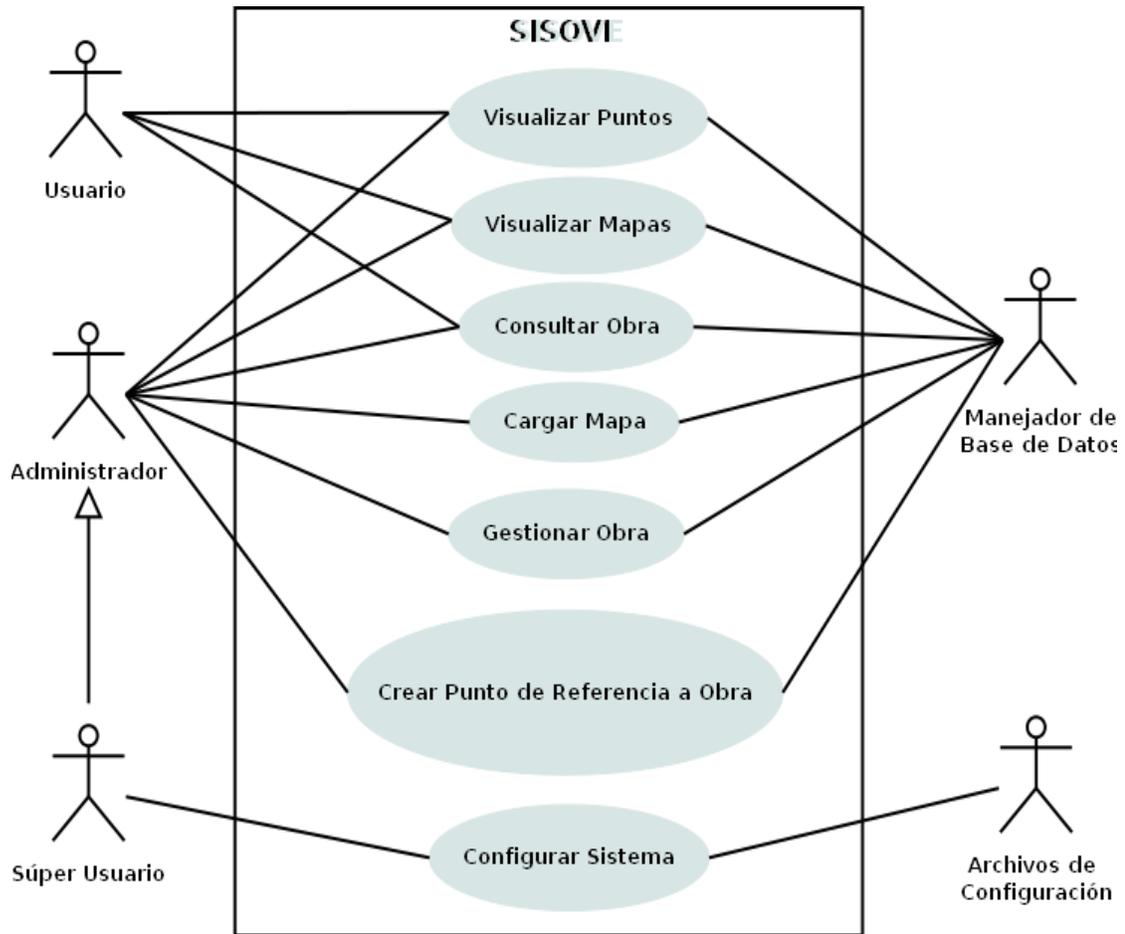


Figura 3.2 Descripción de Caso de Uso de Contexto SISOVE (Fuente: Dima Rodríguez)

Caso de Uso: **Consultar Obra**

Actores: Usuario, Administrador, Súper Usuario, Manejador de Base de Datos.

Descripción: El usuario, el administrador y Súper Usuario pueden realizar diversos tipos de consultas. La información se adquirirá de la base de datos por medio del sistema manejador de base de datos. El sistema contendrá un conjunto de parámetros de búsqueda que permitirán obtener consultas más específicas.

CAPÍTULO III: Fase de Inicio

Caso de Uso: Cargar Mapa

Actores: Administrador, Súper Usuario, Manejador de Base de Datos.

Descripción: Permite al Administrador y al Súper Usuario cargar la imagen del mapa de un archivo en la base de datos del sistema por el manejador de base de datos y que servirán de referencia para la ubicación de la obra. Dependiendo del tipo de consulta, éste cargara a la base de datos un mapa con la vialidad agrícola, otro con las principales vías y un último mapa en donde se hará referencia a los municipios del estado.

Caso de Uso: Gestionar Obra

Actores: Administrador, Súper Usuario, Manejador de Base de Datos.

Descripción: Permite al Administrador y al Súper Usuario manipular datos de una obra, contrato o avance en la base de datos del sistema por medio del manejador de base de datos.

Caso de Uso: Crear Punto de Referencia a Obra

Actores: Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Permite al Administrador y al Súper Usuario ubicar un punto de referencia sobre la imagen del mapa, este es enlazado a una obra, para luego tener acceso por medio del punto. La referencia y datos del punto son guardadas en la base de datos del sistema por el manejador de base de datos.

Caso de Uso: Configurar Sistema

Actores: Súper Usuario y Archivo de configuración.

Descripción: Permite al Súper Usuario configurar el sistema para que este pueda ser utilizado en una red local y así obtener la información de los

distintos departamentos que generan la información.

3.5.3. Casos de Usos detallados

3.5.3.1. Descripción de Caso de Uso Visualizar Mapas

Seguidamente se muestra el diagrama de casos de uso detallado del caso de uso *Visualizar Mapas* (ver figura 3.3). Los actores pueden visualizar diferentes imágenes de mapas, que contendrán los puntos de referencia a las obras creadas así el usuario tendrá una visión general de donde se encuentran las obras ubicadas en la región. Estas imágenes de los mapas serán almacenados en la base de datos para evitar posibles pérdidas del mismo, encargándose de esto el manejador de base de datos.

Caso de Uso: **Mostrar mapa principales vías**

Actores: Usuario, Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por los Usuario, el Administrador y el Súper Usuario para ver la imagen del mapa de las principales vías. Esto le permitida tanto a los usuarios como los administradores tener una referencia de donde se encuentran ubicadas la obras viales. Esta imagen será guarda en la base de datos para tener acceso a él.

Caso de Uso: **Mostrar mapa vías Agrícolas**

Actores: Usuario, Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por los Usuario, el Administrador y el Súper Usuario para ver la imagen del mapa con las vías agrícolas. Esto le permitida tanto a los usuarios como los administradores tener una referencia de

donde se encuentran la obras de vías agrícolas. Esta imagen será guarda en la base de datos para tener acceso a él.

Caso de Uso: **Mostrar mapa por Municipios**

Actores: Usuario, Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por los Usuario, el Administrador y el Súper Usuario para ver la imagen del mapa por municipios, así poder ubicar las obras por municipio. Esto le permitida tanto a los usuarios como los administradores tener una referencia de donde se encuentran la obras. Esta imagen será guarda en la base de datos para tener acceso a él.

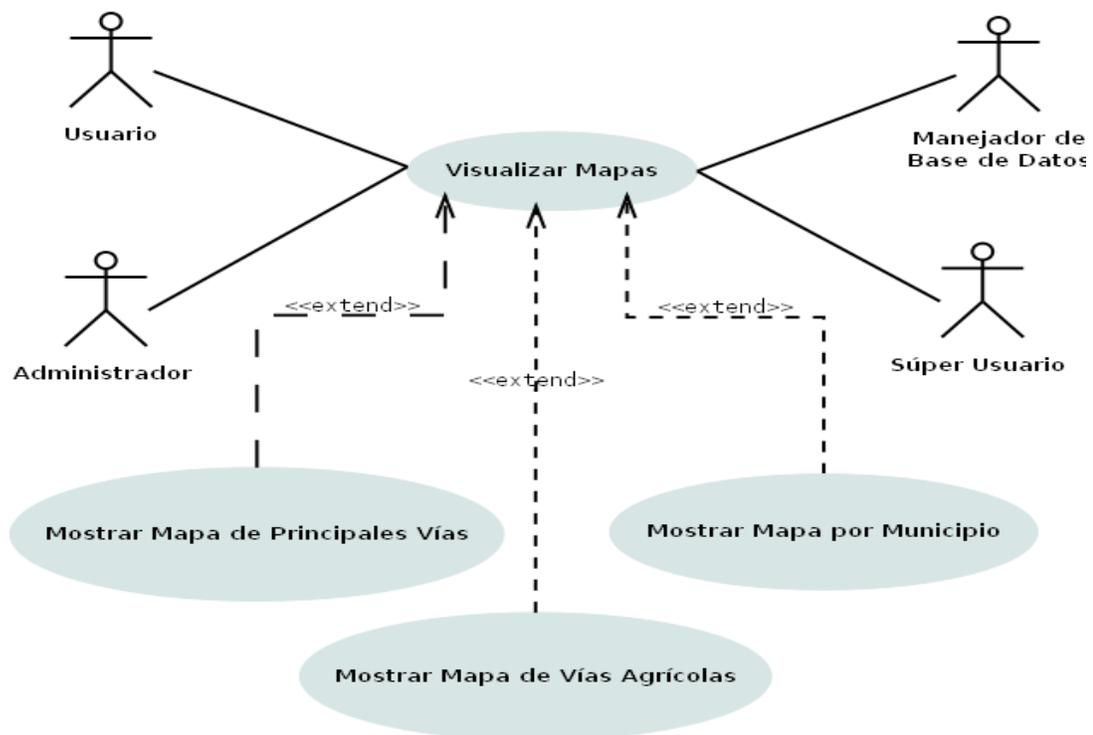


Figura 3.3 Caso de Uso Detallado de Visualizar Mapas.(Fuente: Dima Rodríguez)

3.5.3.2. Descripción de Caso de Uso Visualizar Punto

Tenemos ahora en la figura 3.4 el caso de uso detallado de *Visualizar Puntos*. Aquí el usuario, el administrador y el súper usuario podrán visualizar las obras y contratos no asociados a una obra.

Caso de Uso: **Mostrar Punto de Obra de Vías.**

Actores: Usuario, Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por los Usuario, el Administrador y el Súper Usuario para ver los puntos de las obras en las Principales vías. Esto les permitirá tanto a los usuarios como los administradores y súper usuario tener una referencia de la obra. Los datos de esta imagen son traídas de la base de datos para ser pintada.

Caso de Uso: **Mostrar Punto de Obra de Vía Agrícola.**

Actores: Usuario, Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por los Usuario, el Administrador y el Súper Usuario para ver los puntos de las obras en las vías Agrícolas. Esto le permitida tanto a los usuarios como los administradores tener una referencia de la obra. Los datos de esta imagen son traídas de la base de datos para ser pintada.

Caso de Uso: **Mostrar Punto de Obra de Edificaciones**

Actores: Usuario, Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por los Usuario, el Administrador y el Súper Usuario

para ver los puntos de las obras de Edificaciones. Esto le permitida tanto a los usuarios como los administradores tener una referencia de la obra. Los datos de esta imagen son traídas de la base de datos para ser pintada.

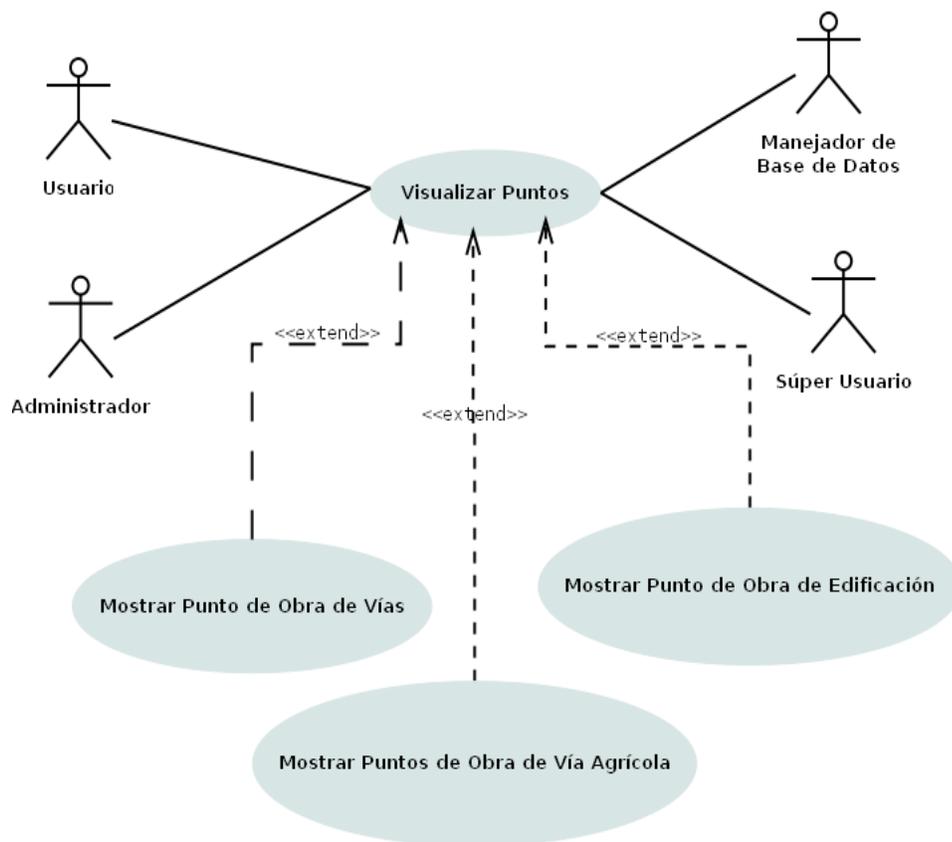


Figura 3.4 Caso de Uso Detallado de Visualizar Puntos. (Fuente: Dima Rodríguez)

3.5.3.3. Descripción de Caso de Uso Consultar Obra

Tenemos ahora en la figura 3.5 el caso de uso detallado de *Consultar Obra*. Aquí el usuario y en administrador podrán buscar las obras, contratos o avances de un contrato cargados al sistema.

CAPÍTULO III: Fase de Inicio

Caso de Uso: Buscar Obras

Actores: Usuario, Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Se muestra una lista de las obras ingresadas al sistema a los Usuarios, Administradores y Súper Usuario, a demás se podrá imprimir esta lista si se desea. Esta lista es traída de la base de datos por el manejador de base de datos.

Caso de Uso: Buscar Contratos

Actores: Usuario, Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Se muestra una lista de los contratos ingresados al sistema a los Usuarios, Administradores y Súper Usuario, a demás se podrá imprimir esta lista si se desea. Esta lista es traída de la base de datos por el manejador de base de datos.

Caso de Uso: Buscar Avances

Actores: Usuario, Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Se muestra una lista de los avances de un contrato a los Usuarios, Administradores y Súper Usuario, a demás se podrá imprimir esta lista si se desea. Esta lista es traída de la base de datos por el manejador de base de datos.

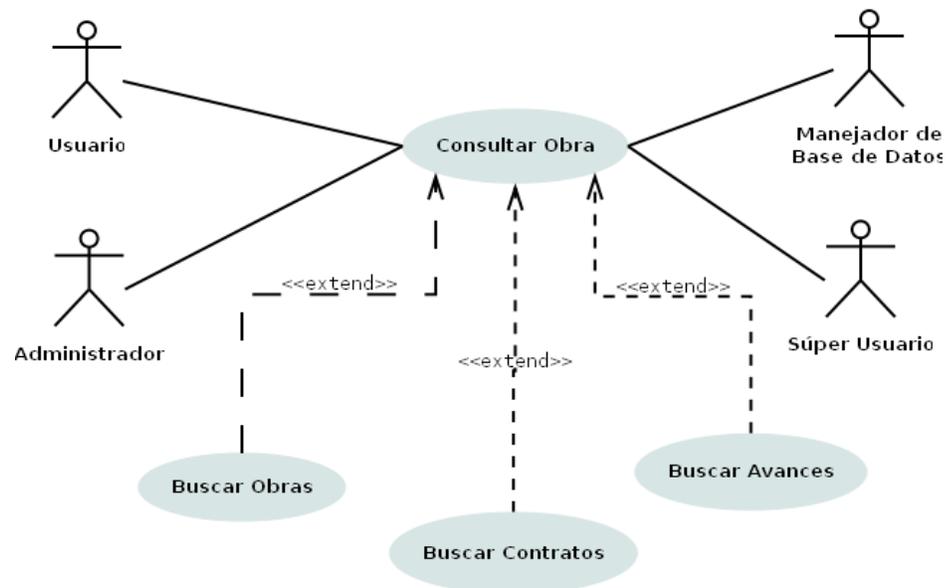


Figura 3.5 Caso de Uso Detallado de Consultar Obra. (Fuente: Dima Rodríguez)

3.5.3.4. Descripción de Caso de Uso Cargar Mapa

En la presente figura 3.6 se observa el caso de uso detallado de *Cargar Mapa*. Por medio de éste el administrador se encargara de escoger la imagen del mapa a ingresar al sistema, estas imágenes pueden ser tres (3), de vialidad principal, agrícola o la imagen del mapa por municipio. Estas imágenes serán guardadas en la base de datos por el Manejador de Base de Datos para ser resguardada con mayor seguridad.

Caso de Uso: **Cargar Mapa de Principales Vías**

Actores: Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: El administrador o el súper usuario tendrán la opción de seleccionar el mapa que será utilizado de referencia para mostrar las obras de vialidad principal. Esta imagen será guardada en la base de datos por el Manejador de Base de Datos.

Caso de Uso: Cargar Mapa de Vías Agrícolas

Actores: Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: El administrador o el súper usuario tendrán la opción de seleccionar el mapa que será utilizado de referencia para mostrar las obras de vialidad Agrícola. Esta imagen será guardada en la base de datos por el Manejador de Base de Datos.

Caso de Uso: Cargar Mapa por Municipio

Actores: Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: El administrador o el súper usuario tendrán la opción de escoger el mapa que será utilizado de referencia para mostrar las obras de Equipamiento territorial. Esta imagen será guardada en la base de datos por el Manejador de Base de Datos.

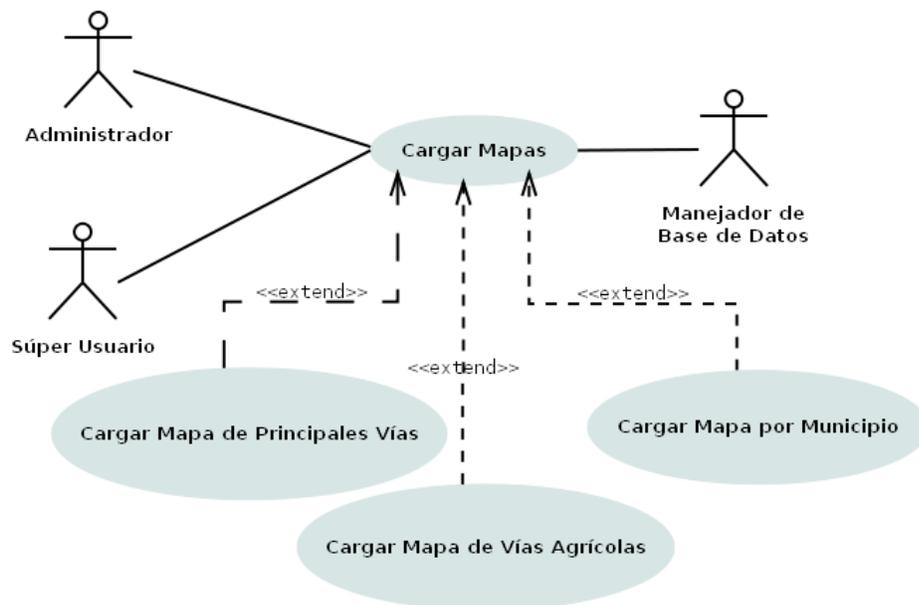


Figura 3.6 Caso de Uso Detallado de Cargar Mapa. (Fuente Dima Rodríguez)

3.5.3.5. Descripción de Caso de Uso Gestionar Obra

La siguiente figura 3.7 muestra el caso de uso detallado de *Gestionar Obra*. Aquí el administrador podrá agregar al sistema los datos de las obras, los contratos y los avances del mismo. Esta información será almacenada para su posterior utilización en la base de datos.

Caso de Uso: **Administrar Obra**

Actores: Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por el administrador o el súper usuario para incluir, modificar o bien sea eliminar los datos de una obra, estos datos han de ser guardado, modificados o eliminados en la base de datos por medio del Manejador de Base de Datos.

Caso de Uso: **Administrar Contrato**

Actores: Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por el administrador o el súper usuario para incluir, modificar o eliminar los datos de un contrato, estos datos han de ser guardado, modificados o eliminados en la base de datos por medio del Manejador de Base de Datos.

Caso de Uso: **Administrar Avance**

Actores: Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por el administrador o el súper usuario para incluir, modificar o eliminar los datos de un avance, estos datos han de ser guardado, modificados o eliminados en la base de datos por medio del Manejador de Base de Datos.

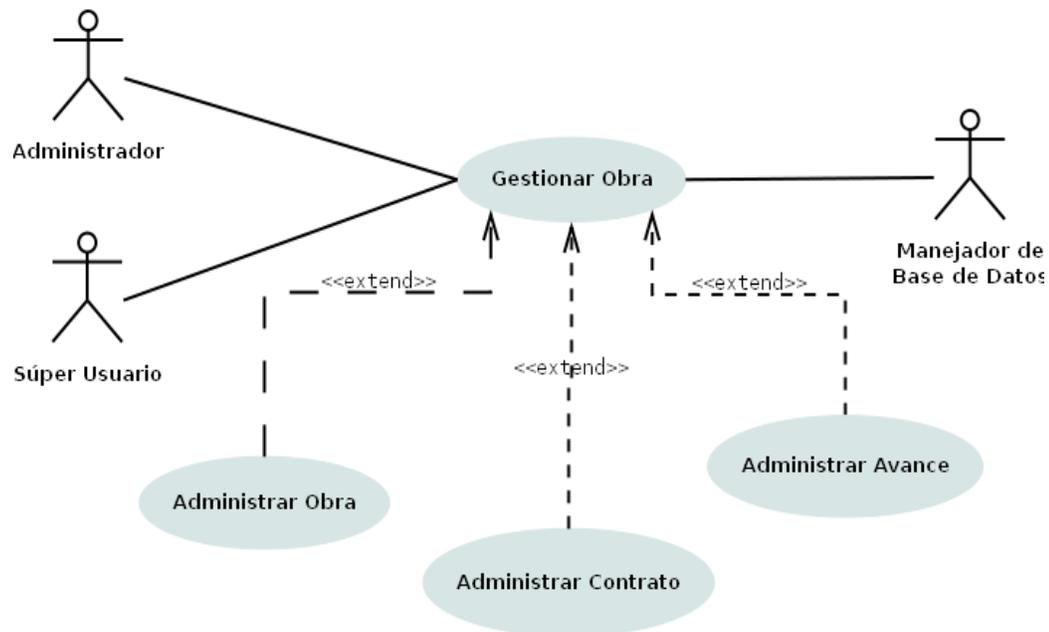


Figura 3.7 Caso de Uso Detallado de Gestionar Obra. (Fuente: Dima Rodríguez)

3.5.3.6. Descripción de Caso de Uso Crear Punto de Referencia a Obra

A continuación se muestra en la figura 3.9 el caso de uso detallado de *Crear Punto de Referencia a Obra*. Para tener una visión más amplia de la ubicación y estatus de las obras, se crearán puntos que simularan las obras y que cambiaran de color dependiendo del estatus de la misma.

Caso de Uso: **Gestionar Punto**

Actores: Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: Es utilizado por el administrador de Avance de Obra para Crear, Mover o Borrar los puntos de acceso a las obras, estos datos han de ser guardado, modificados o eliminados en la base de datos por medio del Manejador de Base de Datos.

Caso de Uso: **Enlazar a Obra**

Actores: Administrador, Súper Usuario y Manejador de Base de Datos.

Descripción: El punto se creará al mismo instante que la obra, ya que, se mantendrá un enlace entre esos dos objetos, para así mantener una referencia del punto con respecto a al obra.

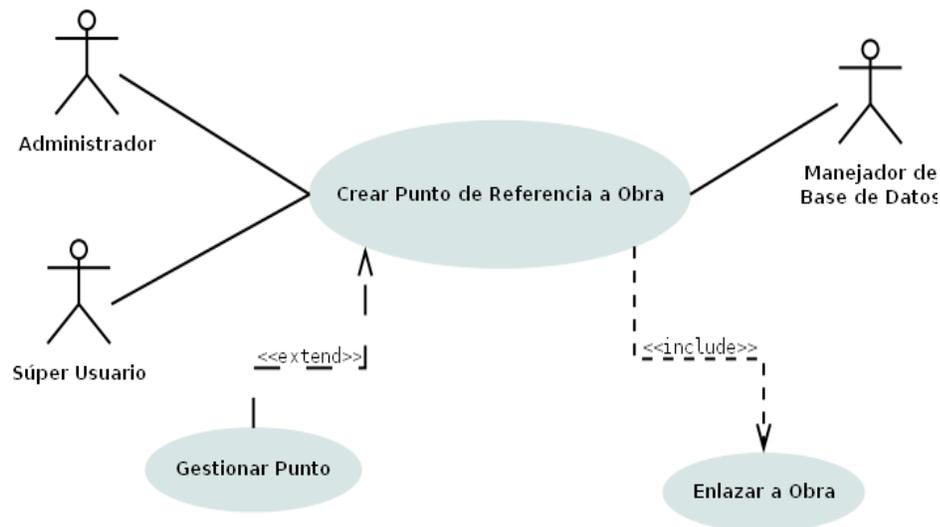


Figura 3.8 Caso de Uso Detallado de Crear Punto de Referencia a Obra. (Fuente: Dima Rodríguez)

3.5.3.7. Descripción de Caso de Uso Configurar Sistema

Y por último mostraremos el caso se uso Configurar Sistema figura 3.9. Y no es más que configurar al sistema, para que este mantenga la conexión con la base de datos.

Caso de Uso: **Configurar Conexión a base de datos**

Actores: Súper Usuario y Archivo de Configuración.

Descripción: El sistema se debe configurar para que pueda tener comunicación con la base de datos, ya que este último se encontrará en una

máquina servidora y se enlaza por medio de una red local.

Esta configuración la realiza el súper usuario ya que su acceso no requiere comunicación con base de datos, éste debe configurar un archivo con los datos necesario para la conexión.

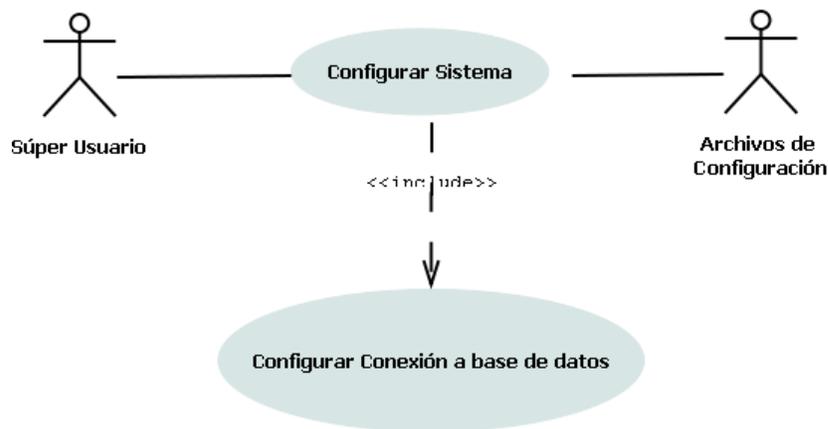


Figura 3.9 Caso de Uso Detallado de Configurar Sistema. (Fuente: Dima Rodríguez)

3.6. ANÁLISIS

El análisis es la etapa en donde el usuario participa para detallar el funcionamiento de los casos de uso, abarcando todas las variantes de cada caso de uso. En algunas ocasiones es necesario refinar de manera gradual el análisis de cada caso de uso. Es muy común refinar el modelo repitiendo pasos ya concluidos, eso no implica un mal análisis o diseño, simplemente es un proceso gradual para encontrar el mejor modelo, preferentemente siempre deben ser ampliaciones y no correcciones a lo realizado.

El lenguaje que se utiliza en el análisis se basa en un modelo de objetos conceptual, que se llama modelo de análisis. El modelo de análisis ayuda a refinar los requisitos.

3.6.1. Diagramas De Clases De Análisis

Clases del Análisis se utilizan para realizar una descripción abstracta de la realización de los casos de uso del modelo de casos de uso.

Esta clase se centra en el tratamiento de los requisitos funcionales y pospone los no funcionales, hasta llegar a las actividades de diseño e implementación. Esto hace que una clase de análisis se más evidente en el contexto del dominio del problema, menos específica que sus contrapartidas en diseño e implementación.

Las clases de análisis están estereotipadas en tres tipos que son: la clase de interfaz, control y de entidad en la figura 3.10 se muestran los estereotipos.

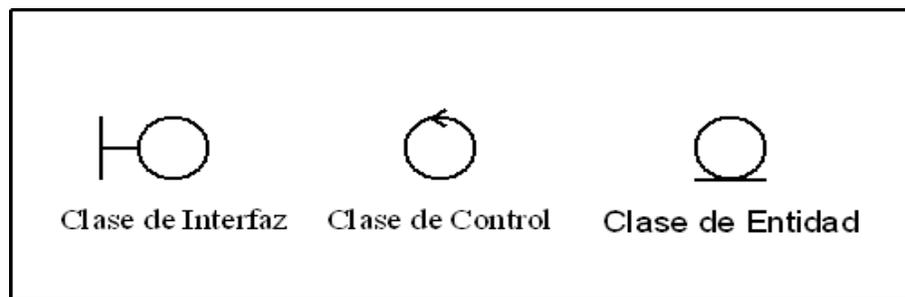


Figura 3.10 Estereotipos Clase de interfaz, control y de entidad. (Fuente: Dima Rodríguez)

Las clases interfaz, se utilizan para modelar la interacción entre el sistema y sus actores y a menudo representan abstracciones de ventanas, formularios, paneles, interfaces de comunicación, etc. La clase control representa coordinación, secuencia, transacciones y control de otros objetos y se usa con frecuencia para encapsular el control de un caso de uso en concreto. Y por ultimo tenemos la clase entidad, la cual, se utiliza para modelar información que posee una vida larga y que a menudo es persistente, este también puede modelar la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto del mundo real, o un

CAPÍTULO III: Fase de Inicio

suceso del mundo real. En la tabla 3.4 se muestran las clases de interfaz del sistema, en la tabla 3.5 se muestra la clase control del sistema y en la tabla 3.6 se observa la clase de entidad del sistema.

Tabla 3.6. Clases de Interfaz del sistema (1/3)

Clases	Definición	Tipo
:UIVisualizarMapas	Esta clase permite al usuario, al administrador y al súper usuario interactuar con el sistema, seleccionando los mapas que serán traídos de la base de datos para ser mostrados y que han de ser utilizado bien sea para inserta, consultar o mover los puntos asociados una obra o contrato no asociado.	Interfaz
:IUmapa	Esta clase permite al usuario, al administrador y al súper usuario interactuar con el sistema, y ver las imágenes de los mapas y los puntos que hacen referencia a una obra o contrato no asociado.	Interfaz
:UIBuscarObra	Esta clase permite al usuario, al administrador y al súper usuario interactuar con el sistema, consultando la información relacionada con las obras guardadas en la base de datos.	Interfaz

Fuente: Dima Rodríguez

CAPÍTULO III: Fase de Inicio

Tabla 3.7. Clases de Interfaz del sistema (2/3)

Clases	Definición	Tipo
:UIBuscarContrato	Esta clase permite al usuario, al administrador y al súper usuario interactuar con el sistema, consultando la información relacionada con los contratos guardados en la base de datos.	Interfaz
:UIBuscarAvance	Esta clase permite al usuario, al administrador y al súper usuario interactuar con el sistema, consultando la información relacionada con los avances guardados en la base de datos.	Interfaz
:UICargarMapaViaAgricola	Esta clase permite al administrador y al súper usuario interactuar con el sistema, para poder cargar la imagen del mapa de vía agrícola al sistema. Esta imagen será guardada en la base de datos.	Interfaz
:UICargarPrincipalesVias	Esta clase permite al administrador y al súper usuario interactuar con el sistema, para poder cargar la imagen del mapa de las principales vías al sistema. Esta imagen será guardada en la base de datos.	Interfaz
:UIObra	Esta clase permite al administrador y al súper usuario interactuar con el sistema, cargando, modificando o eliminando la información relacionando con una obra.	Interfaz

Fuente: Dima Rodríguez

CAPÍTULO III: Fase de Inicio

Tabla 3.8. Clases de Interfaz del sistema (3/3)

Clases	Definición	Tipo
:UICargarporMunicipio	Esta clase permite al administrador y al súper usuario interactuar con el sistema, para poder cargar la imagen del mapa por municipio al sistema. Esta imagen será guardada en la base de datos.	Interfaz
:IUContrato	Esta clase permite al administrador y al súper usuario interactuar con el sistema, cargando, modificando o eliminando la información relacionando con un contrato.	Interfaz
:IUAvance	Esta clase permite al administrador y al súper usuario interactuar con el sistema, cargando, modificando o eliminando la información relacionando con un avance.	Interfaz
:IUConfigurarBD	Esta clase permite al súper usuario interactuar con el sistema, configurando un archivo que pueda hacer conexión entre el sistema y el manejador de base de dato.	Interfaz

Fuente: Dima Rodríguez

Tabla 3.9. Clases de Control del sistema (1/3)

Clases	Definición	Tipo
:Mostrar mapa de vías Agrícola	Esta clase permite solicitar y devolver la imagen del mapa de vías agrícolas guardada en la base de datos.	Control

Fuente: Dima Rodríguez

CAPÍTULO III: Fase de Inicio

Tabla 3.10. Clases de Control del sistema (2/3)

Clases	Definición	Tipo
Mostrar mapa de principales vías	Esta clase permite solicitar y devolver la imagen del mapa de las principales vías guardada en la base de datos.	Control
:Mostrar mapa por Municipios	Esta clase permite solicitar y devolver la imagen del mapa por municipio guardada en la base de datos.	Control
:GestorMapa	Esta clase permite manipular los datos de los mapas en la base de datos.	Control
:GestorPunto	Esta clase permite manipular los datos de los puntos en la base de datos.	Control
:Mostrar puntos de obras de vías Agrícola	Esta clase permite solicitar y devolver los puntos que hacen referencia a las obras de vías agrícolas guardada en la base de datos.	Control
:Mostrar puntos de obras de principales vías	Esta clase permite solicitar y devolver los puntos que hacen referencia a las obras de las principales vías guardada en la base de datos.	Control
:Mostrar puntos de obras por Municipios	Esta clase permite solicitar y devolver los puntos que hacen referencia a las obras de equipamiento guardada en la base de datos.	Control
:GestorObra	Esta clase permite manipular los datos de las obras en la base de datos.	Control
:GestorContrato	Esta clase permite manipular los datos de los contrato en la base de datos.	Control

Fuente: Dima Rodríguez

CAPÍTULO III: Fase de Inicio

Tabla 3.11. Clases de Control del sistema (3/3)

Clases	Definición	Tipo
:GestorAvance	Esta clase permite manipular los datos de los avances en la base de datos.	Control
:ManipularObra	Esta clase permite manipular los datos de la obra en el sistema, bien sea crear, modificar o eliminar una obra.	Control
:ManipularContrato	Esta clase permite manipular los datos de la obra en el sistema, bien sea crear, modificar o eliminar un contrato.	Control
:ManipularAvance	Esta clase permite manipular los datos de la obra en el sistema, bien sea crear, modificar o eliminar un avance.	Control
:ManipularPunto	Esta clase permite manipular los datos de la obra en el sistema, bien sea crear, modificar o eliminar un punto.	Control
EnlazarObra	Esta clase permite enlazar el punto creado, con la obra que hará referencia al punto.	Control

Fuente: Dima Rodríguez

Tabla 3.12. Clases de Entidad del sistema (1/2)

Clases	Definición	Tipo
:Mapa	Esta clase permite contener los datos del mapa para ser manipulado luego en el sistema.	Entidad

Fuente: Dima Rodríguez

Tabla 3.13. Clases de Entidad del sistema (2/2)

Clases	Definición	Tipo
:Punto	Esta clase permite contener los datos del punto para ser manipulado luego en el sistema.	Entidad
:Obra	Esta clase permite contener los datos de la obra para ser manipulado luego en el sistema.	Entidad
:Contrato	Esta clase permite contener los datos del contrato para ser manipulado luego en el sistema.	Entidad
:Avance	Esta clase permite contener los datos del avance para ser manipulado luego en el sistema.	Entidad

Fuente: Dima Rodríguez

3.6.1.1. Diagrama de Clase de Análisis del Caso de Uso Visualizar Mapas

Este diagrama de clase de análisis del caso de uso Visualizar Mapas, comprende al usuario, al administrador y al súper usuario que utilizarán la clase interfaz UIVisualizarMapas, para que les sea mostrado los mapas, que serán traídos de la base de datos por el gestor mapa y que tendrá como clase de entidad mapa. Las imágenes han de ser mostrada a los usuarios por medio de la interfaz IUmapa se puede Observar la figura 3.11

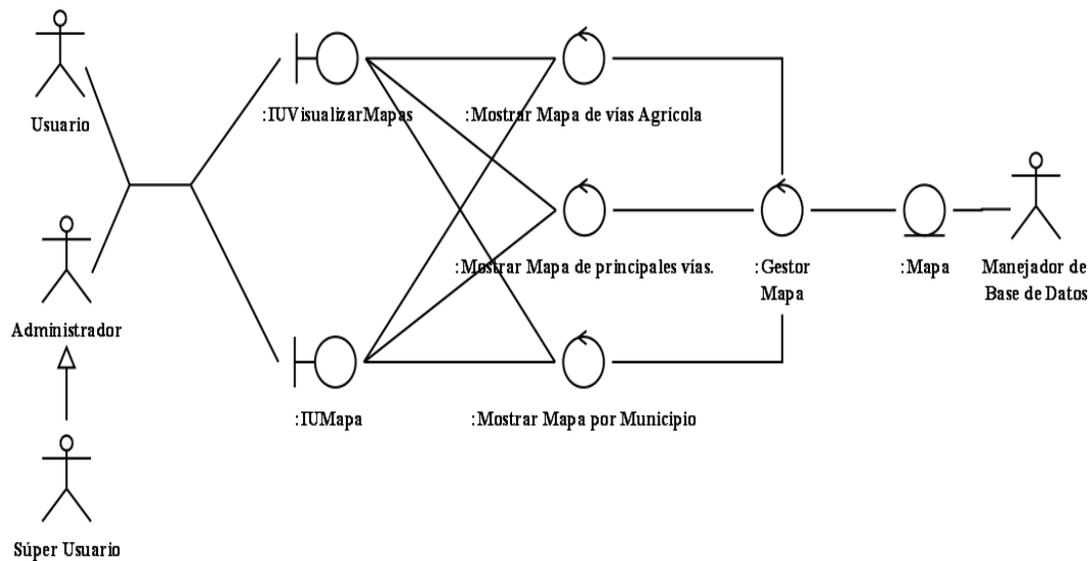


Figura 3.11 Diagrama de Clase de Análisis del Caso de Uso Visualizar Mapas. (Fuente: Dima Rodríguez)

3.6.1.2. Diagrama de Clase de Análisis del Caso de Uso Visualizar Puntos

Este diagrama de clase de análisis del caso de uso Visualizar puntos, comprende al usuario, al administrador y al súper usuario que utilizarán la clase interfaz `UIVisualizarMapa`, para que les sea mostrado los puntos que hacen referencia a las obras, éstos que serán traídos de la base de datos por el gestor punto y que tendrá como clase de entidad punto. Los puntos han de ser mostrados a los usuarios por medio de la interfaz `IUmapa` como se puede observar en la figura 3.12

3.6.1.3. Diagrama de Clase de Análisis del Caso de Uso Consultar Obra

Este diagrama de clase de análisis del caso de uso Consultar Obra, comprende al usuario, al administrador y al súper usuario que utilizarán las clases interfaz `UIBuscarObra`, `UIBuscarContrato` y `UIBuscarAvance`, para que se puedan consultar las obras, contratos o avances, que serán traídos de la base de datos por los gestores

CAPÍTULO III: Fase de Inicio

obra, contrato o avance y que tendrán como clase de entidad obra, contrato o avance. Observe la figura 3.13

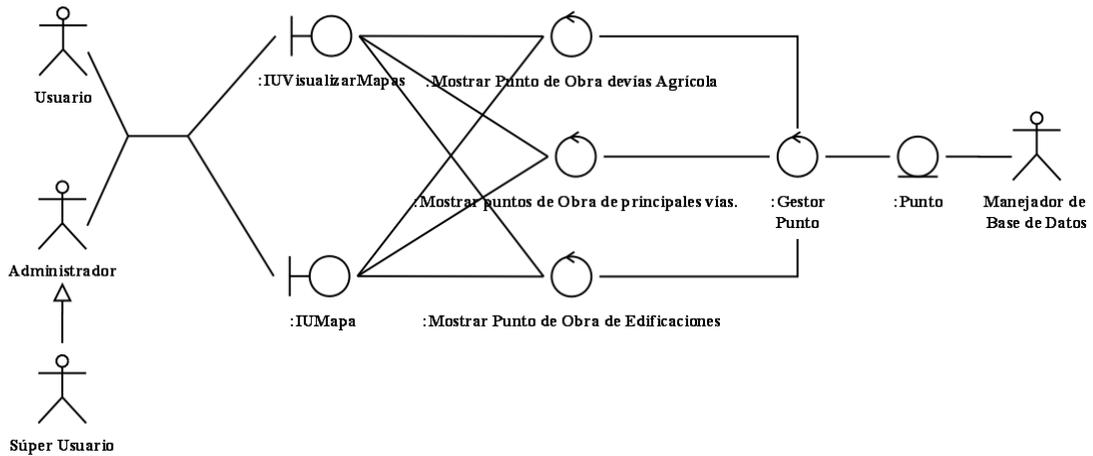


Figura 3.12 Diagrama de Clase de Análisis del Caso de Uso Visualizar Puntos. (Fuente: Dima Rodríguez)

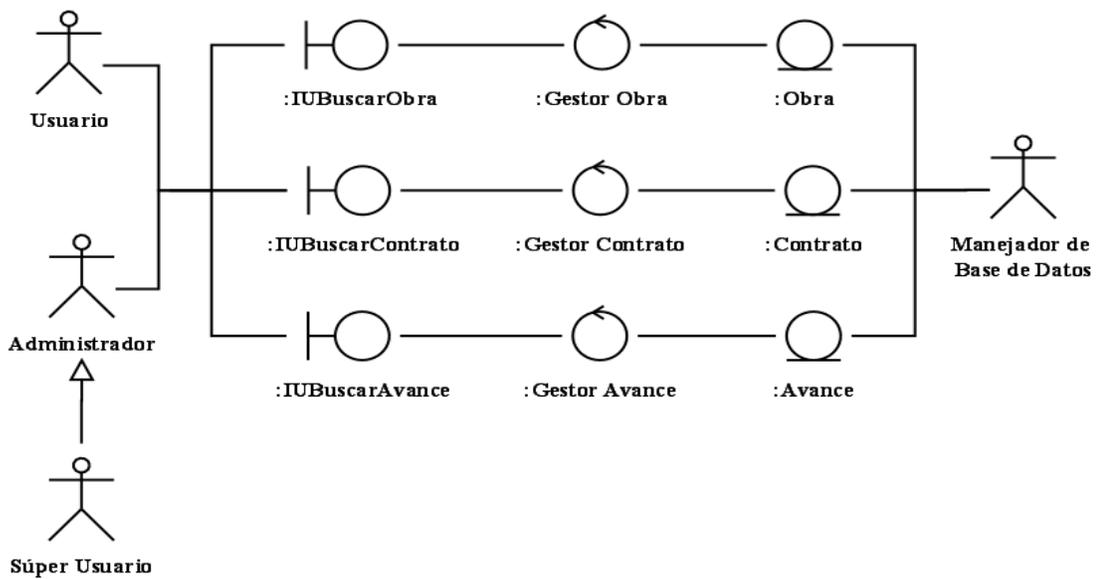


Figura 3.13 Diagrama de Clase de Análisis del Caso de Uso Consultar obra. (Fuente: Dima Rodríguez)

3.6.1.4. Diagrama de Clase de Análisis del Caso de Uso Cargar Mapa

Este diagrama de clase de análisis del caso de uso Cargar Mapa, esta comprendido por el administrador y el súper usuario que utiliza la clase de interfaz UICargarMapaViaAgricola, la interfaz UICargarPrincipalesVias, y la interfaz UICargarporMunicipio para poder cargar las distintas imágenes de mapas al sistema. Estas imágenes serán guardadas en la base de datos por el Gestor Mapa y que tendrán como clase de entidad Mapa. Mire la figura 3.14

3.6.1.5. Diagrama de Clase de Análisis del Caso de Uso Gestionar Obra

Este diagrama de clase de análisis del caso de uso Gestionar Obra, comprende al administrador y el súper usuario que utilizarán las clases de interfaz UIObra, IUContrato y IUAvance, para que se puedan Manipular los datos de las obras, contratos o avances, que serán manejados en la base de datos por el gestor obra, gestor contrato o gestor avance y que tendrán como clase de entidad obra, contrato y avance. Vea la figura 3.15

3.6.1.6. Diagrama de Clase de Análisis del Caso de Uso Crear Punto de Referencia a Obra

El diagrama de clase de análisis del caso de uso Crear Punto de Referencia a Obra, comprende el administrador y el súper usuario que utilizarán la clase interfaz UIObra y UI Visualizar Mapa, para Manipular el punto y así poder enlazarlo a la obra, que serán manejados en la base de datos por el gestor punto y que tendrán como clase de entidad punto. Se observa en la figura 3.16

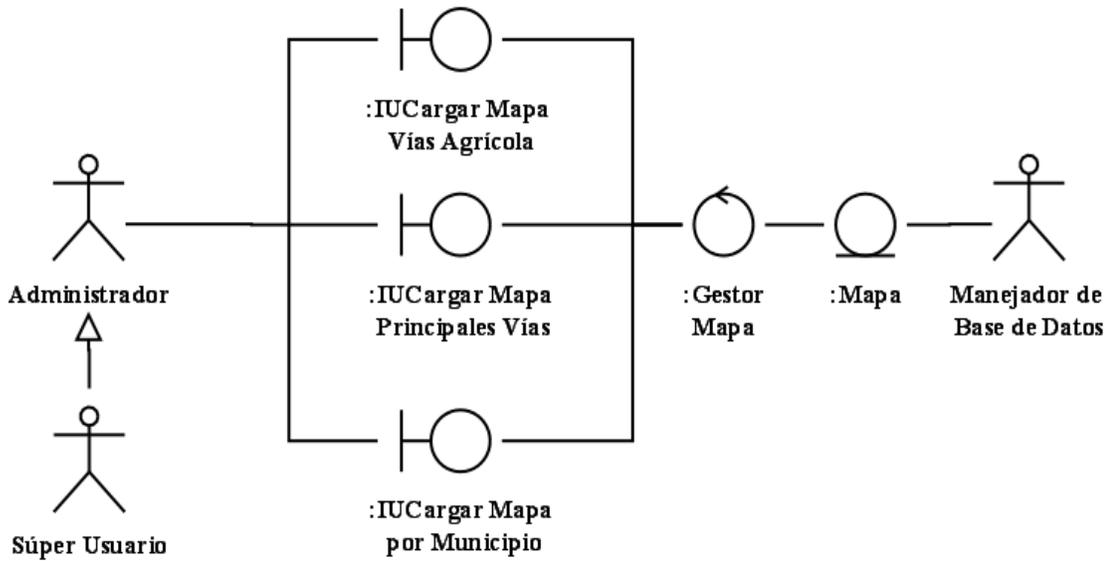


Figura 3.14 Diagrama de Clase de Análisis del Caso de Uso Cargar Mapa. (Fuente: Dima Rodríguez)

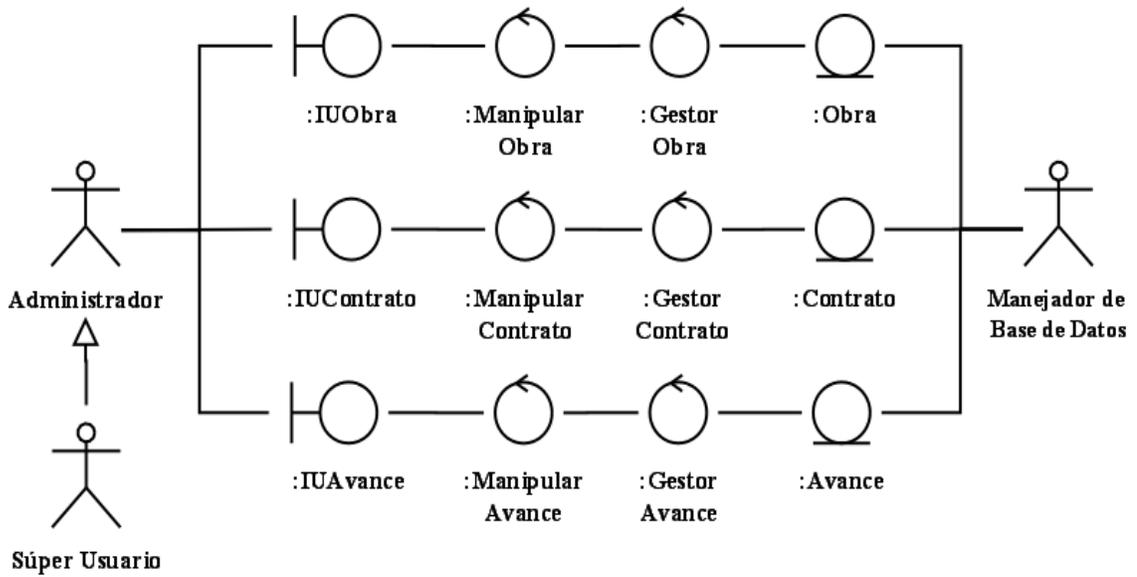


Figura 3.15 Diagrama de Clase de Análisis del Caso de Uso Gestionar Obra. (Fuente: Dima Rodríguez)

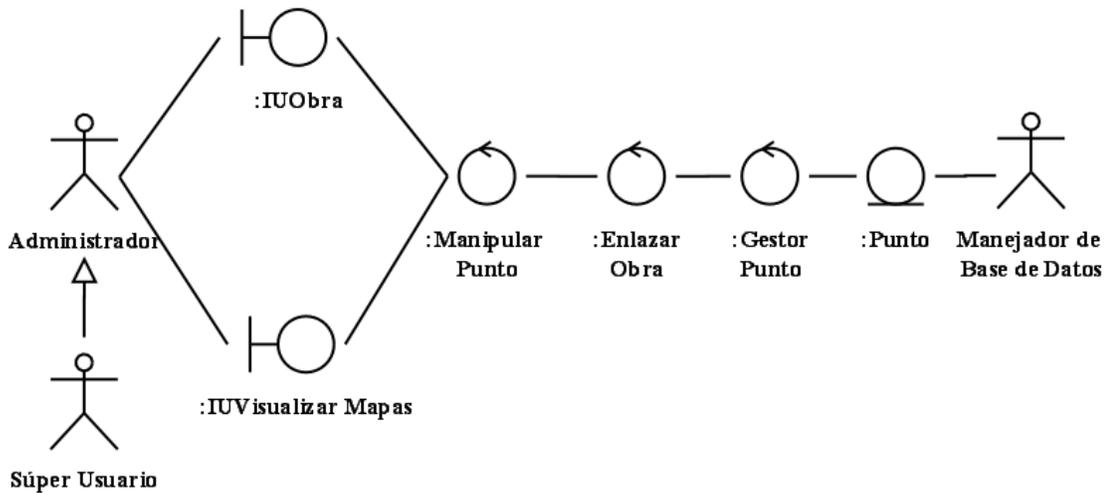


Figura 3.16 Diagrama de Clase de Análisis del Caso de Uso Crear Punto de Referencia a Obra
(Fuente: Dima Rodríguez)

3.6.1.7. Diagrama de Clase de Análisis del Caso de Uso Configurar Sistema

El diagrama de clase de análisis del caso de uso Configurar Sistema comprende al súper usuario que utilizara la clase IUConfigurarBD, para configurar un archivo que pueda hacer conexión entre el sistema y el manejador de base de datos. Observe la figura 3.17

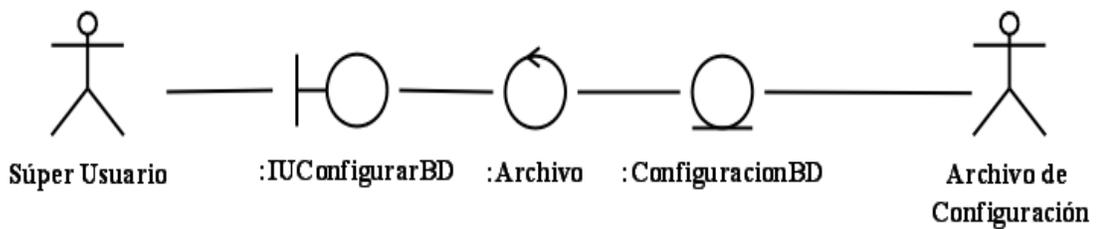


Figura 3.17 Diagrama de Clase de Análisis del Caso de Uso Configurar Sistema (Fuente: Dima Rodríguez)

3.6.2. Diagrama De Colaboración

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Fijando interés en las relaciones entre los objetos y su topología. También destaca la organización de los objetos que participan en una interacción.

En los diagramas de colaboración los objetos son representados por un nodo de un grafo y los enlaces que conectan esos objetos como arcos del grafo. Los mensajes enviados de un objeto, a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.

En la figura 3.18 se describe el diagrama de colaboración detallado del caso de uso *Visualizar Mapas*. Aquí se mostrará la interacción entre los objetos que actuarán en la visualización de los mapas.

En la imagen anterior se puede apreciar como el usuario, administrador o súper usuario (1, 11, 21) selecciona el mapa de vía agrícola o el mapa de principales vías o el mapa por municipio a mostrar por medio del IUVisualizar Mapas. Luego se procesa la (2, 12, 22) solicitud para mostrar el mapa de vía agrícola o el mapa de principales vías o el mapa por municipio. Posterior a esto, se podrá (3, 13, 23) buscar cualquiera de los mapas nombrado anteriormente. Seguidamente se ha de (4, 14, 24) recuperar bien sea el mapa de vía agrícola o el mapa de principales vías o el mapa por municipio, (5, 15, 25) se buscan en la base de datos, procesando esta búsqueda en la misma por medio del GestorMapa. Una vez obtenida la imagen de la base de datos se procederá a (6, 16, 26) devolverla, el GestorMapa (7, 17, 27) retorna el objeto de

CAPÍTULO III: Fase de Inicio

tipo mapa, para ser (8, 18, 28) devuelta y así (9, 19, 29) ser cargado el mapa en la interfaz IUMapas. Y luego de haber realizado todas las operaciones tanto el usuario como el administrador y súper usuario podrán (10, 20, 30) ver las imagen de los diferentes mapas.

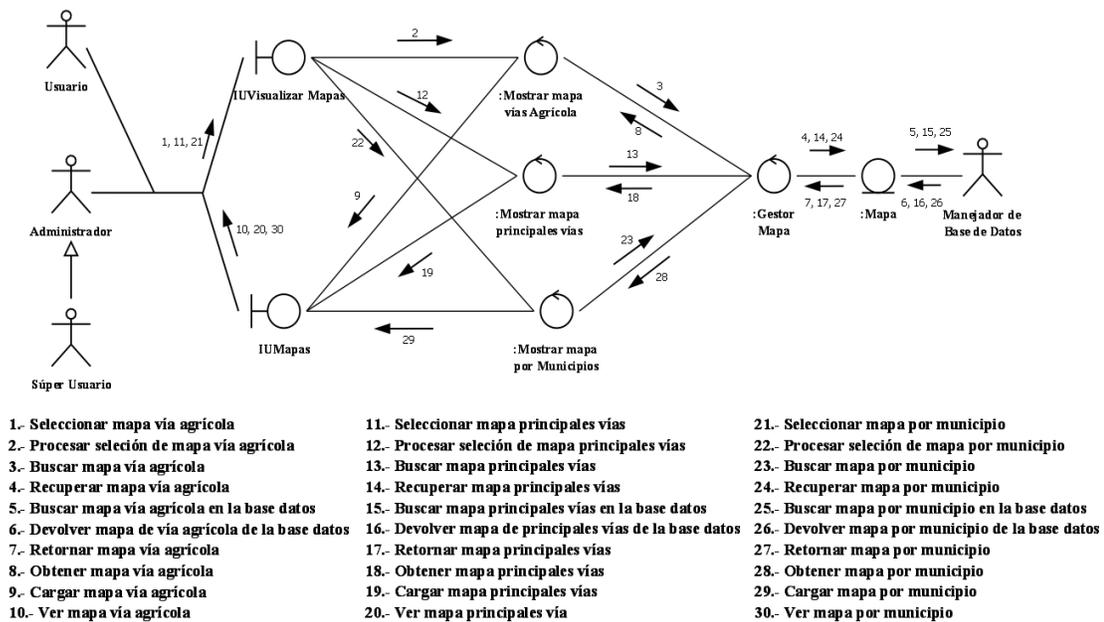


Figura 3.18 Diagrama de colaboración detallado del caso de uso Visualizar Mapas (Fuente: Dima Rodríguez)

En la figura 3.19 se describe el diagrama de colaboración detallado del caso de uso Visualizar Puntos. Aquí se mostrara la interacción entre los objetos que actuaran en la visualización de los mapas.

En el diagrama mostrado se podrá ver como los usuarios del sistema podrán seleccionar los mapas en donde se encuentran los puntos de las obras, por medio de la clase interfaz IUVisualizar Mapas (1, 11, 21). Luego se procesa la (2, 12, 22) solicitud de mostrar, el mapa de vía agrícola o el mapa de principales vías o el mapa por municipio con sus respectivos puntos. Posterior a esto, se podrá (3, 13, 23)

CAPÍTULO III: Fase de Inicio

buscara los puntos asignados a los mapas. Seguidamente se ha de (4, 14, 24) recuperar los puntos de cualquiera de los mapas seleccionados (5, 15, 25) buscada en la base de datos, procesando esta búsqueda en la base de datos por medio del GestorPunto, una vez obtenida la información de la base de datos se procederá a (6, 16, 26) devolver los Puntos, el Gestor punto las (7, 17, 27) retornara los objeto de tipo punto, para ser (8, 18, 28) devuelta y así (9, 19, 29) ser cargado los puntos en la interfaz IUMapa. Y con esto tanto el usuario como el administrador y súper usuario podrán (10, 20, 30) ver los puntos en el mapa.

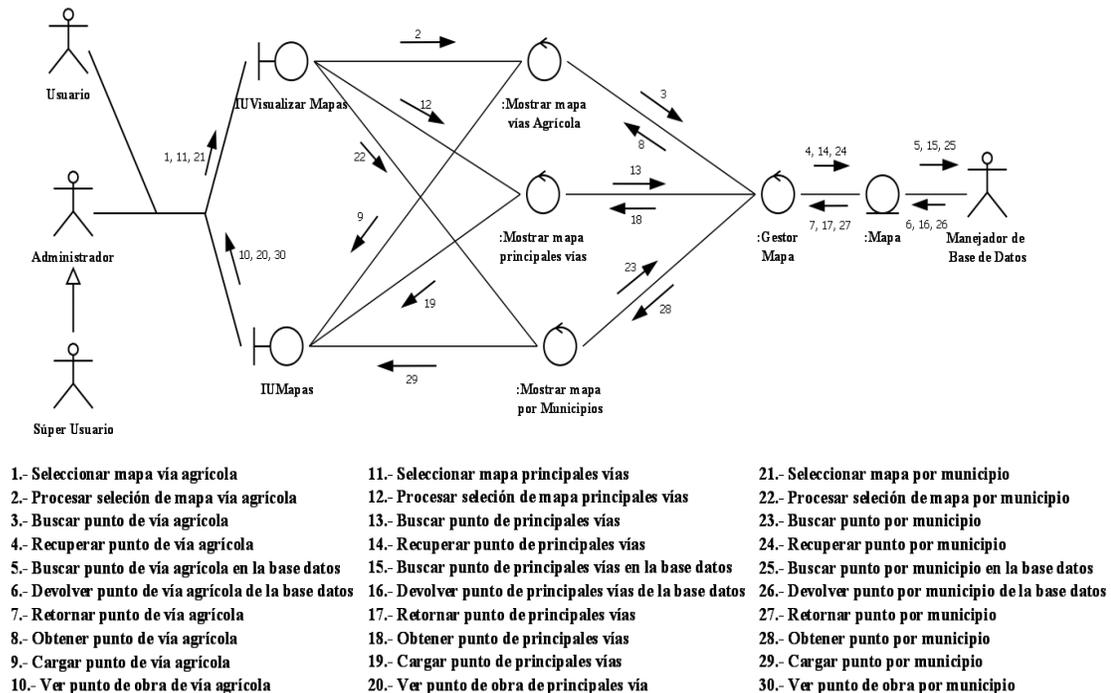
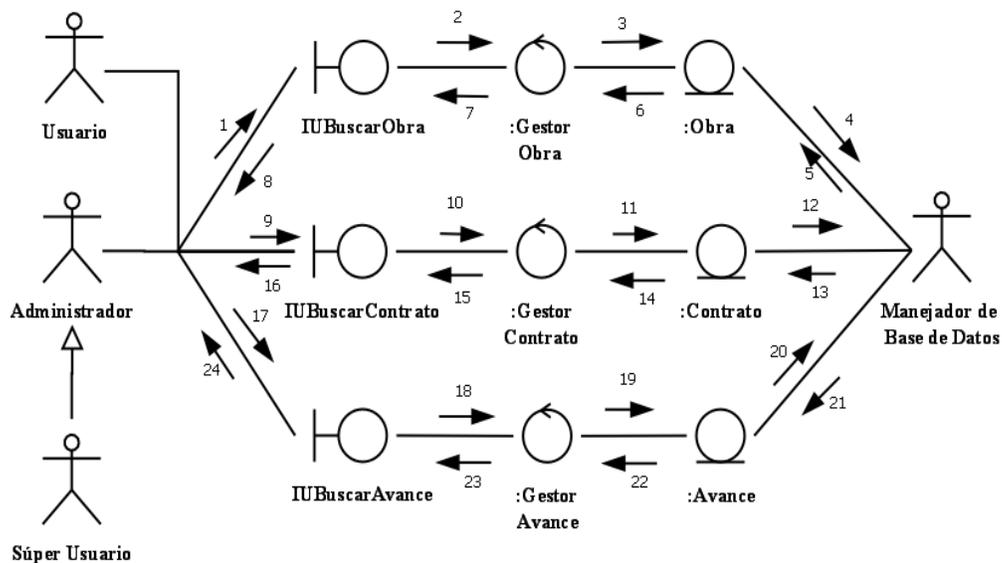


Figura 3.19 Diagrama de colaboración detallado del caso de uso Visualizar Puntos (Fuente: Dima Rodríguez)

En la figura 3.20 se describe el diagrama de colaboración detallado del caso de uso Consultar Obra. Aquí se mostrará la interacción entre los objetos que intervendrán en la Consulta de obras, contratos o avances.

CAPÍTULO III: Fase de Inicio



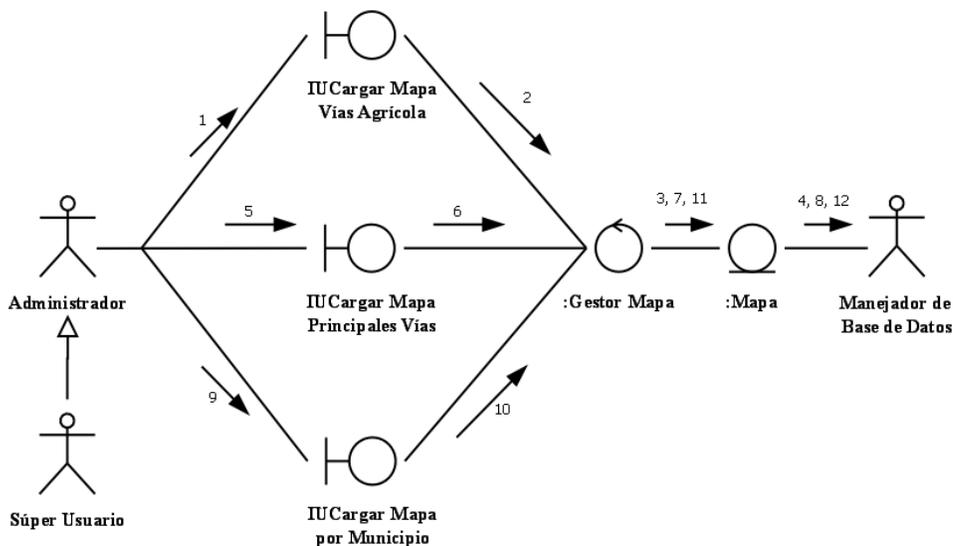
- | | | |
|--|--|---|
| 1.- Solicitar Búsqueda de obras | 9.- Solicitar Búsqueda de contratos | 17.- Solicitar Búsqueda de avances |
| 2.- Consultar obras | 10.- Consultar contratos | 18.- Consultar avances |
| 3.- Buscar obras | 11.- Buscar contratos | 19.- Buscar avances |
| 4.- Recupera obras | 12.- Recupera contratos | 20.- Recupera avances |
| 5.- Obtener obras | 13.- Obtener contratos | 21.- Obtener avances |
| 6.- Retornar obras | 14.- Retornar contratos | 22.- Retornar avances |
| 7.- Mostrar obras | 15.- Mostrar contratos | 23.- Mostrar avances |
| 8.- Ver consulta de obra | 16.- Ver consulta de contratos | 24.- Ver consulta de avances |

Figura 3.20 Diagrama de colaboración detallado del caso de uso Consultar Obra (Fuente: Dima Rodríguez)

Tanto el usuario como el administrador y el súper usuario podrán (1, 9, 17) solicitar una búsqueda en el sistema por medio de la IUBuscarObra, IUBuscarContrato, IUBuscarAvance. Estos podrán realizar la (2, 10, 18) consulta de Obras, Contratos o Avances. Luego se procederá a (3, 11, 19) Buscar las Obras, Contratos o Avances, (4, 12, 20) recuperándola de la base de datos. Luego de esta búsqueda se (5, 13, 21) obtienen las Obras, los Contratos o los Avances. Ahora se retornan (6, 14, 22) los objetos para ser mostrados (7, 15, 23) en sus respectivas interfaces y así los usuarios del sistema podrán ver la consulta (8, 16, 24).

CAPÍTULO III: Fase de Inicio

En la figura 3.21 se describe el diagrama de colaboración detallado del caso de uso *Cargar Mapa*. Aquí se podrá observar la interacción entre los objetos que actuarán en el ingreso de las imágenes al sistema.



- | | |
|--|--|
| 1.- Solicitud de guardar mapa de vía agrícola | 7.- Almacenar Datos del mapa de principales vías |
| 2.- Cargar mapa de vía agrícola | 8.- Guardar Mapa de principales vías en la base de datos |
| 3.- Almacenar Datos del mapa de vía agrícola | 9.- Solicitud de guardar mapa por municipios |
| 4.- Guardar mapa de vía agrícola en la base de datos | 10.- Cargar mapa por municipios |
| 5.- Solicitud de guardar mapa de principales vías | 11.- Almacenar Datos del mapa por municipios |
| 6.- Cargar mapa de principales vías | 12.- Guardar Mapa por municipios en la base de datos |

Figura 3.21 Diagrama de colaboración detallado del caso de uso *Cargar Mapa* (Fuente: Dima Rodríguez)

En la figura precedente se podrá ver como se describe la carga de los mapas. Primero se realiza la (1, 5, 9) solicitud de guardar la imagen del mapa, bien sea de vía agrícola, de vías principales o por municipio, luego se (2, 6, 10) cargar el mapa seleccionado que se va a guardar, el de vía agrícola, el de principales vías o el mapa por municipio. Luego se procede a (3, 7, 11) almacenar los datos del mapa seleccionado, para ser finalmente (4, 8, 12) guardado en la base de datos.

CAPÍTULO III: Fase de Inicio

En la figura 3.22 se describe el diagrama de colaboración detallado del caso de uso *Gestionar Obra*. Aquí se mostrara la interacción entre los objetos que actuaran en el ingreso del sistema.

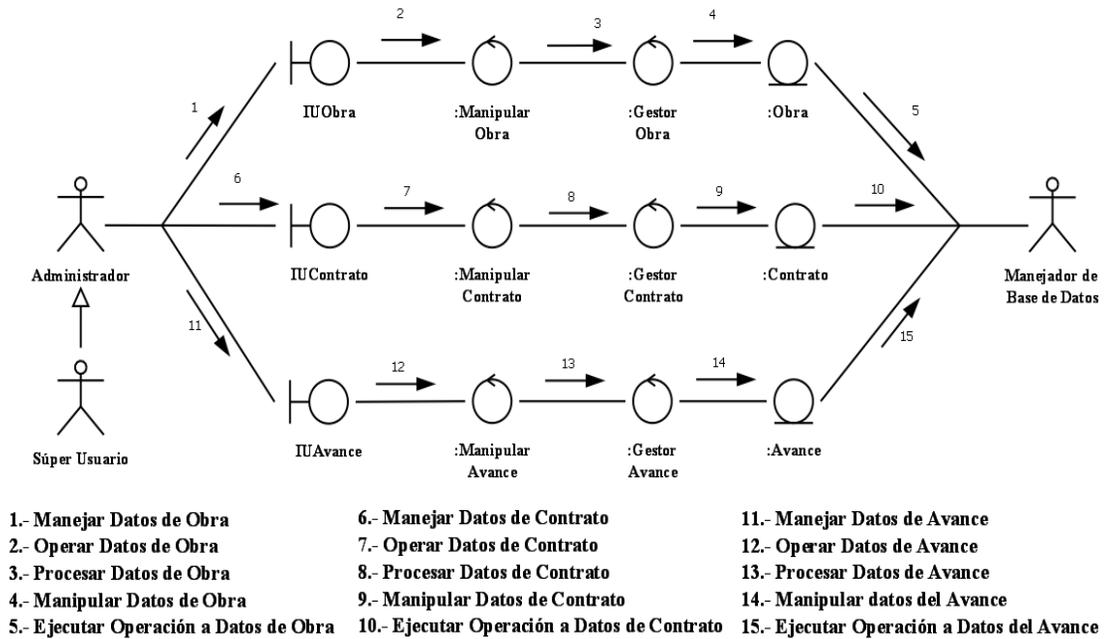
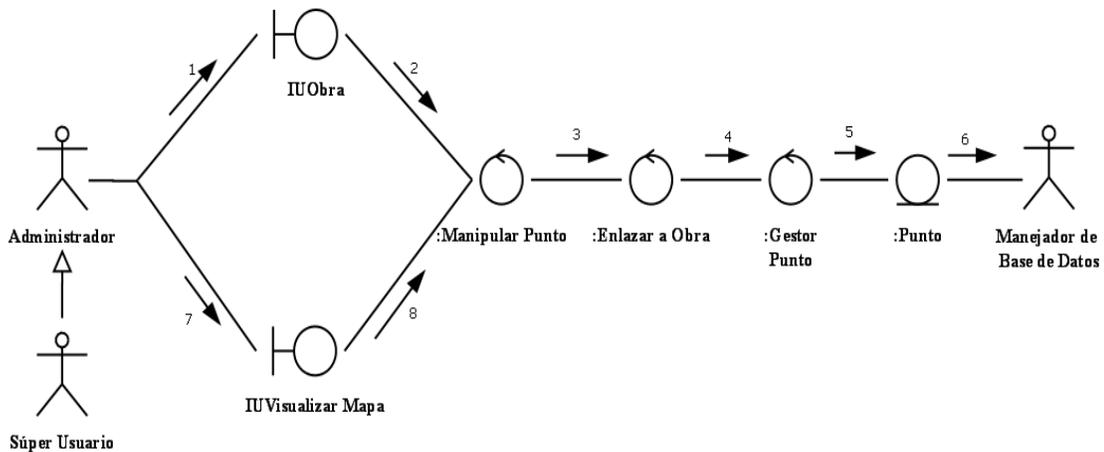


Figura 3.22 Diagrama de colaboración detallado del caso de uso *Gestionar Obra* (Fuente: Dima Rodríguez)

Se puede observar como se representan las operaciones al momento de Gestionar o administrar los datos de las obra, contrato o avance. Dichas operaciones se podrán manejar por medio de IUObra, IUContrato o IUAvance (1, 6, 11), luego determinamos que operación vamos a realizar (2, 7, 12) a la obra, contrato o avance., luego se procesaran los datos (3, 8, 13) y dependiendo de la operación a realizar se procederá a manipular los datos (4, 9, 14) para finalmente ejecutar las operaciones (5, 10, 15) en la base de datos.

CAPÍTULO III: Fase de Inicio

En la figura 3.23 se describe el diagrama de colaboración detallado del caso de uso *Crear Punto de Referencia a Obra*. Aquí se ilustrará la interacción entre los objetos que actuarán en la creación del punto con su respectiva referencia.



- | | |
|---|---|
| 1.- Solicitud de manejar Datos | 5.- Manipular Datos de Punto |
| 2.- Solicitud de manejar Datos de Punto | 6.- Procesar Datos de punto |
| 3.- Vincular Punto con Obra | 7.- Seleccionar Mapa con Punto a Mover |
| 4.- Operar datos de Punto | 8.- Solicitud de manejar Datos de Punto |

Figura 3.23 Diagrama de colaboración detallado del caso de uso *Crear Punto de Referencia a Obra*

(Fuente: Dima Rodríguez)

De la figura 3.23, se puede observar que para manipular los puntos existen 2 formas: hacemos la (1) solicitud de manejar los datos por la IUObra o se puede (7) seleccionar el mapa por medio de la IUVisualizarMapa para poder reubicar un punto. Posterior a esto se realiza la (2, 8) solicitud para manejar los datos del punto. Se debe (3) vincular el punto con la obras. Luego se procede a (4) operar con los datos del punto y así (5) manipular los datos de forma que puedan ser (6) procesados en la base de datos.

En la figura 3.24 se describe el diagrama de colaboración detallado del caso de uso *Configurar Sistema*. Aquí se mostrara la interacción entre los objetos que actuaran en la configuración del sistema.

CAPÍTULO III: Fase de Inicio

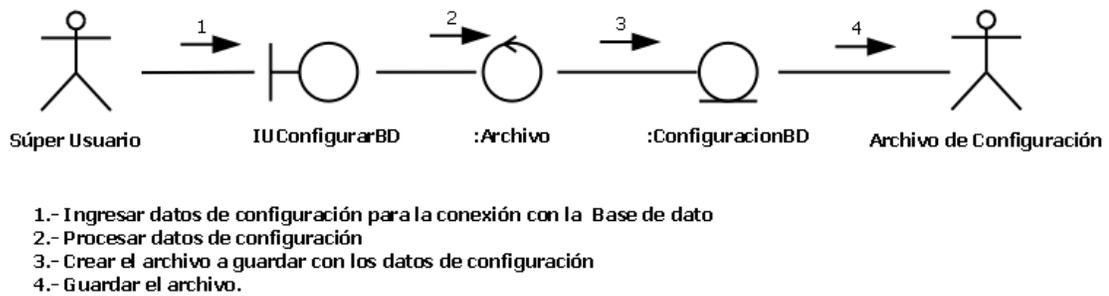


Figura 3.24 Diagrama de colaboración detallado del caso de uso Configurar Sistema (Fuente: Dima Rodríguez)

Como se puede apreciar en la imagen antes expuesta el súper usuario es el encargado de (1) ingresar los datos para configurar para la conexión con la base de datos, luego se han de (2) procesar los datos para (3) crear el archivo y finalmente guardar el archivo.

3.6.3. Diagrama de Paquete de Análisis

Los paquetes de análisis proporcionan un medio para organizar los artefactos del modelo de análisis en piezas manejables. Un paquete de análisis puede constar de clases de análisis, de realizaciones de casos de uso, y de otros paquetes del análisis recursivamente.

Los paquetes del análisis son particionamientos funcionales del sistema basados en el dominio del problema y debería ser reconocible por las personas con conocimiento del dominio.

En la figura 3.25 se muestra el paquete de análisis donde agruparemos un número de casos de uso en un paquete concreto.

CAPÍTULO III: Fase de Inicio

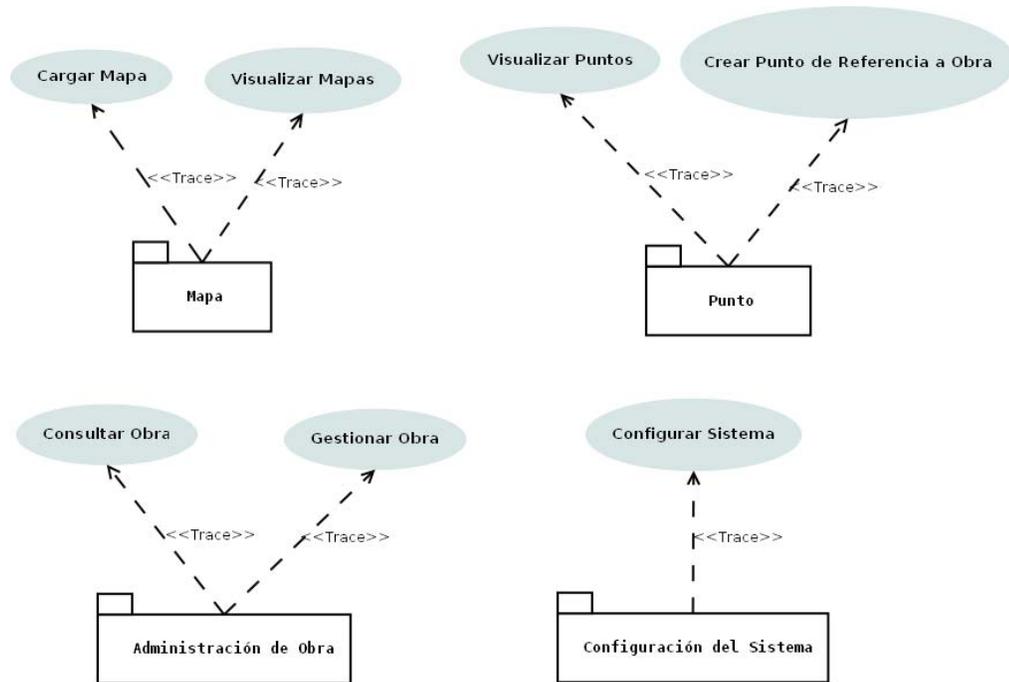


Figura 3.25 Identificación de Paquetes de análisis a partir de los Casos de Uso (Fuente: Dima Rodríguez)

En el siguiente diagrama se ve la relación que hay entre los paquetes ver figura 3.26.

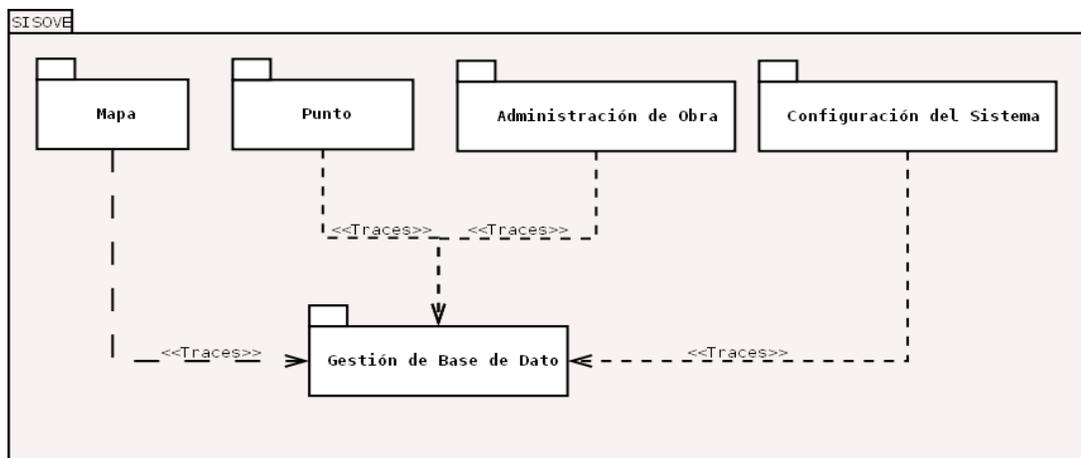


Figura 3.26 Diagrama de Paquete de Análisis (Fuente: Dima Rodríguez)

3.7. RESUMEN DE LA FASE DE INICIO

Con la realización del modelo de dominio del sistema presentado, se alcanzo de forma notable la comprensión del contexto del sistema. Con este modelo se lograron identificar los actores principales, tanto humanos como sistemas externos y la relación de éstos en el sistema “SISOVE”.

Por medio del diagrama de casos de uso se logro estructurar la aplicación en módulos que se encargan de cumplir con cada uno de los requerimientos del sistema detectados y por medio de los diagramas de clase de análisis y colaboración se alcanzo un estudio mas detallado de los casos de uso.

Se pudo apreciar a través del encapsulamiento de los diversos casos de uso detectados utilizando para su representación el diagrama de paquetes se obtuvo una abstracción mas clara de los diversos componentes de software que harán vida en el sistema y serán de utilidad para las posteriores etapas de desarrollo en este trabajo de grado.

Para finalizar se puede acotar que en virtud de la naturaleza de esta fase implementada, no fue necesario realizar tareas de diseño, implementación, y prueba.

CAPÍTULO IV: FASE DE ELABORACIÓN

4.1. INTRODUCCIÓN

Luego de haber culminado la fase de inicio se iniciara la segunda fase del desarrollo del software, denominada fase de elaboración. Teniendo esta como objetivo establecer la línea base de la arquitectura, la cuál, guiará el trabajo de desarrollo durante las fases siguientes, como también a lo largo del ciclo de vida del software.

La fase de elaboración hace énfasis en establecer la arquitectura del sistema software, aunque también recaen en esta fase tareas tan importantes como son el diseño de la interfaz de usuario y el diseño de la base de datos.

4.2. DIAGRAMA DE CLASE DE DISEÑO

Partiendo de análisis previos se definirá mediante diagramas la arquitectura y el diseño del software. Existen varios diagramas que se utilizan para representar dicha estructura, y uno de los más importantes es el diagrama de clase de diseño, en este se representaran las clases, que en adelante instanciarán los objetos. En la Figura 4.1 correspondiente al diagrama de clase de diseño del sistema SISOVE se observa que las clases están representada mediante rectángulos y las relaciones se representan a través de las líneas que las conectan a dichas clases, también se nota el uso de ciertos conectores que sirven para identificar la relación entre las mismas. Las clases fundamentales para este sistema son: Obra, Contrato, ImagenMapa, PuntoReferencia, Avance. Así mismo se representan las clases de interfaz que permiten comunicar al sistema con los entes externos, y los gestores de bases de datos que sirven para coleccionar la data requerida.

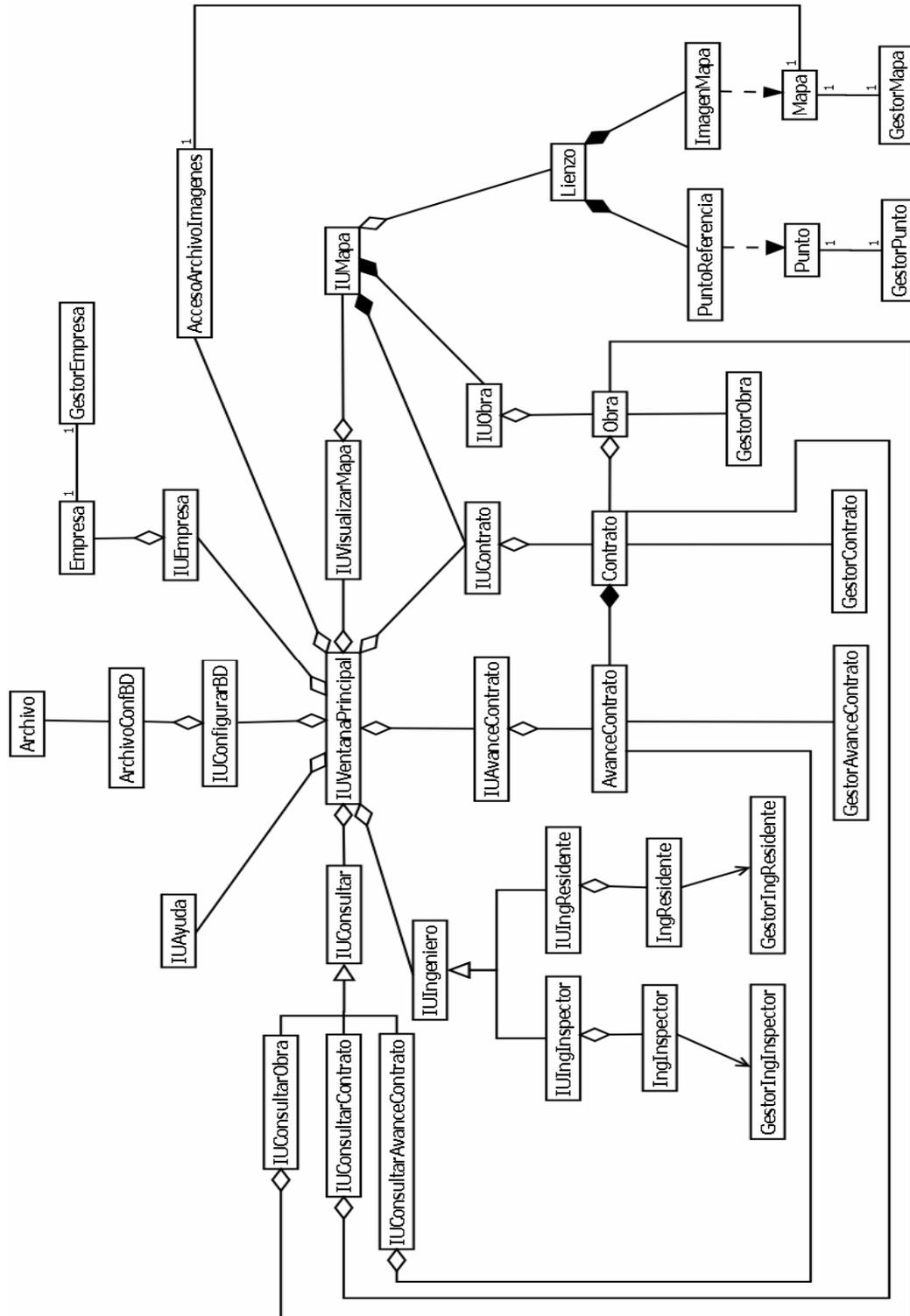


Figura 4.1 Diagrama de Clase de Diseño del Sistema SISOVE (Fuente: Dima Rodríguez)

4.2.1. Diseño de las Clases

Las clases de diseño identificadas al realizar los casos de uso y las interacciones con sus respectivos objetos, se determinaron en la fase anterior. Sin embargo, no se definieron los atributos y responsabilidades en el sistema.

A continuación se definirán las responsabilidades y atributos de las clases identificadas en el sistema SISOVE. Tabla 4.1

Tabla 4.1. Clases que componen al sistema (1/7)

Clase	Responsabilidad	Atributo
IUConfigurarBD	Permite al súper usuario ingresar datos para la configuración de la conexión entre el sistema y la base de datos.	
IUEmpresa	Por medio de esta interfaz el usuario carga los datos de la empresa encargada de la obra	
IUAyuda	Muestra la ayuda del sistema.	
IUVentanaPrincipal	Es la encargada de ejecutar las interfaces necesarias para que el usuario pueda interactuar con el sistema.	
IUVisualizarMapa	Utilizada para determinar que tipo de mapa desea ver el usuario.	
IUMapa	Utilizado para mostrar la imagen del mapa y los puntos que harán referencia a una obra.	

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.2. Clases que componen al sistema (2/7)

Clase	Responsabilidad	Atributo
IUConsultar	Interfaz utilizada por el usuario para la búsquedas de obras, contratos o lo avance de los contratos.	
IUObra	Permite al usuario cargar, modificar o eliminar los datos de una obra y sus enlaces por medio de su interfaz.	
IUContrato	Permite al usuario cargar, modificar o eliminar los datos de un contrato y sus enlaces por medio de su interfaz.	
IUAvance	Permite al usuario cargar, modificar o eliminar los datos del Avance de un contrato por medio de su interfaz.	
IUIngeniero	Interfaz utilizada para cargar los datos del ingeniero inspector y residente, asignados a una obra.	
Archivo	Clase encargada de realizar todos los métodos referentes a las operaciones con archivo.	
ArchivoConfBD	Clase contenedora de los datos utilizados para la conexión con la base de datos.	Login. Password. IP(Servidor) Base de datos

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.3. Clases que componen al sistema (3/7)

Clase	Responsabilidad	Atributo
Avance	Clase contenedora de los datos del avance de los contratos.	<p>Código del avance.</p> <p>Nro. de la Semana de supervisión.</p> <p>Fecha Inicio de supervisión.</p> <p>Fecha fin de supervisión.</p> <p>Monto ejecutado del contrato.</p> <p>Porcentaje físico ejecutado.</p> <p>Estado físico.</p> <p>Estado administrativo.</p> <p>Fecha de culminación.</p> <p>Numero de prorrogas.</p> <p>Fecha de paralización.</p> <p>Motivo paralización.</p> <p>Fecha de reinicio.</p> <p>Numero de prorrogas.</p> <p>Tipo de prorrogas.</p> <p>Fecha de inicio de prorrogas.</p> <p>Fecha de finalización de la prorrogas.</p> <p>Estatus físico de la obra.</p> <p>Estatus administrativo de la obra.</p> <p>Fecha de de la acta de aceptación provisional.</p> <p>Fecha de de la acta de aceptación definitiva.</p> <p>Amortizaciones.</p>

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.4. Clases que componen al sistema (4/7)

Clase	Responsabilidad	Atributo
AccesoArchivoImagen	Clase utilizada para obtener la imagen de un tipo de archivo específico	fchCargardeArchivo.
IngResidente	Clase contenedora de los datos del Ingeniero residente.	C.I.V. Cedula Nombre. Apellido. Nro. Teléfono oficina. Nro. Teléfono celular. Email.
Empresa	Clase contenedora de los datos de la empresa.	R.I.F. Nombre de la empresa. Dirección de la empresa. Teléfono de oficina. Nro. de fax Email. Nro. de cuenta a depositar. Banco a depositar.
IngInspector	Clase contenedora de los datos del Ingeniero inspector	C.I.V. Cedula Nombre. Apellido. Nro. Teléfono oficina. Nro. Teléfono celular. Email.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.5. Clases que componen al sistema (5/7)

Clase	Responsabilidad	Atributo
Contrato	Clase contenedora de los datos del contrato.	Código del contrato. Objetivo Unidad Ejecutora. Monto del contrato Ubicación geográfica. Órgano rector. Ente ejecutor. Fecha de inicio. Fecha de culminación.
Obra	Clase contenedora de los datos de la obra.	Código de la obra. Denominación. Unidad ejecutora. Monto de la obra. Ubicación geográfica. Órgano rector. Ente ejecutor.
Lienzo	Clase utilizada para el manejo de las imágenes, carga las imágenes de los mapas y los puntos que hacen referencia a obra.	
PuntoReferencia	Clase utilizada para la creación de los puntos que hacen referencia a la obra.	

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.6. Clases que componen al sistema (6/7).

Clase	Responsabilidad	Atributo
ImagenMapa	Clase encargada de recuperar las imágenes tanto del archivo como de la base de datos para luego ser pintado en el lienzo	
Mapa	Clase contenedora de la imagen del mapa.	Identidad del mapa. Imagen.
Punto	Clase contenedora del punto que hace referencia a la obra.	Punto de referencia. Código de referencia a la obra. Identidad del mapa. Tipo de obra. Color. Finalidad de la obra.
GestorEmpresa	Encargada de realizar las operaciones en la base de datos para los datos de la empresa.	
GestorIngInspector	Encargada de realizar las operaciones en la base de datos para los datos del ingeniero inspector.	
GestorIngResidente	Encargada de realizar las operaciones en la base de datos para los datos del ingeniero residente.	

(Fuente: Dima Rodríguez)

Tabla 4.7. Clases que componen al sistema (7/7)

Clase	Responsabilidad	Atributo
GestorAvance	Encargada de realizar las operaciones en la base de datos para los datos del avance del contrato.	
GestorObra	Encargada de realizar las operaciones en la base de datos para los datos de la obra.	
GestorContrato	Encargada de realizar las operaciones en la base de datos para los datos del contrato.	
GestorMapa	Encargada de realizar las operaciones en la base de datos para los datos de la imagen del mapa.	

(Fuente: Dima Rodríguez)

4.3. DIAGRAMA DE CAPA

El diagrama de capas es un diagrama que permite representar las diferentes capas de software que contendrá el sistema SISOVE.

El diseño de la arquitectura del sistema involucra cuatro capas: la capa específica de la aplicación, la capa general de la aplicación, la capa intermedia y finalmente la capa de software del sistema.

CAPÍTULO IV: Fase de Elaboración

El diagrama de capas mostrado en la siguiente figura 4.2 representa todas las capas del software en las que se distribuye la funcionalidad del sistema y que se requieren para que se lleve a cabo el sistema y en un futuro sea implementado.

En cada capa del diagrama se muestran las relaciones existentes entre los paquetes mediante las trazas. Se puede observar el paquete “Canaima GNU/Linux” y “Windows XP”, que representan los sistemas operativos usados durante su desarrollo, las herramientas de software: “JDK 6”, el sistema manejador de base de datos “PostgreSQL”. En la capa general de la aplicación se puede apreciar los paquetes del sistema “SISOVE”. En la capa correspondiente la capa específica de la aplicación se puede apreciar los paquetes encargados de la “Mapa”, “Punto”, “Administración de Obra” y “Configuración del sistema”.

4.4. DIAGRAMA DE SECUENCIA

El diagrama de secuencia se encarga de mostrar las interacciones necesarias entre los objetos y los actores del sistema para llevar a cabo la funcionalidad de los casos de uso. En este tipo de diagrama también intervienen los mensajes, que es el medio de comunicación de los objetos. A continuación presentaremos el diagrama de secuencia del caso de uso “*Configurar Sistema*”. En la figura 4.3 se puede observar como interactúa el súper usuario con el sistema. Aquí observamos como es la Configuración del sistema para tener conexión con la base de datos.

Primero se solicita la configuración del sistema por medio de un objeto instanciado por la clase IUVentanaPrincipal, el cual, nos permite crear el archivo que se va a guardar, luego procede a crear el objeto de tipo ArchivoConfBD y cargar el mismo por medio del objeto archivo, para posteriormente guardar el Archivo.

CAPÍTULO IV: Fase de Elaboración

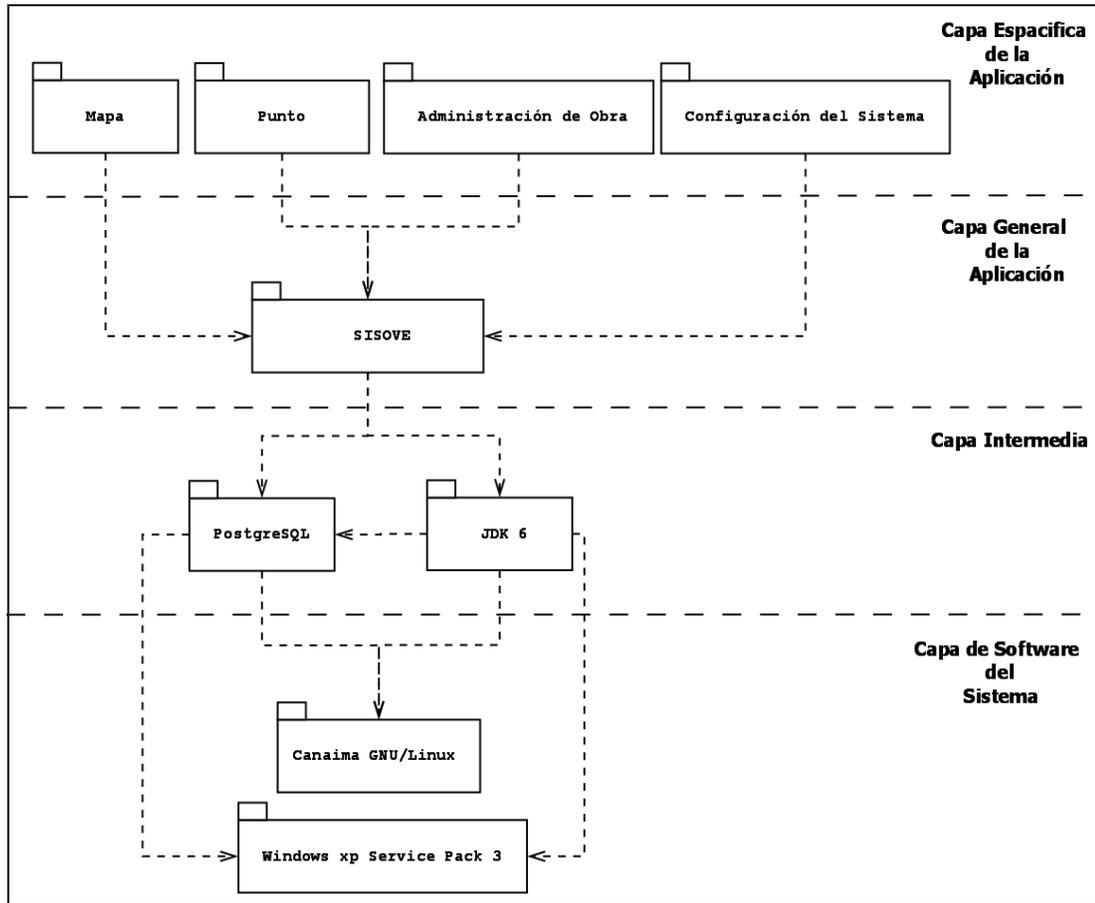


Figura 4.2 Diagrama de capas del sistema (Fuente: Dima Rodríguez)

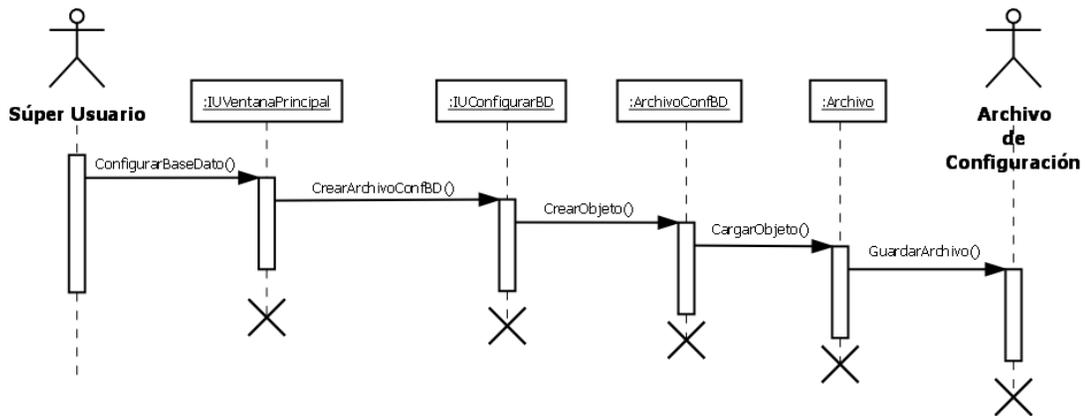


Figura 4.3 Diagrama de Secuencia del Caso de Uso Configurar Sistema. (Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

En la figura 4.4, se observa el diagrama de secuencia del caso de uso “Visualizar Mapas”. Para este caso se puede ver como el usuario, puede visualizar la imagen del mapa. Primero se debe seleccionar el tipo de mapa a mostrar, esto se hace por medio de la IUVisualizarMapas, y que se puede visualizar, a través del enlace de la IUVentanaPrincipal. Una vez seleccionado el tipo de mapa, se va a recuperar la imagen de la base de datos a través del objeto ImagenMapa para ser cargada en el objeto de tipo mapa, pintado en un objeto de tipo lienzo y mostrado en un objeto de tipo IUMapa.

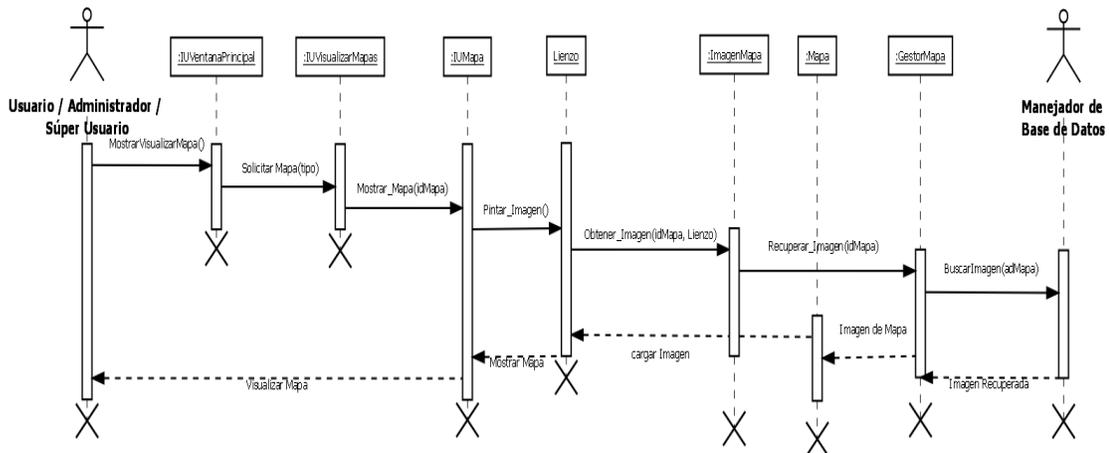


Figura 4.4 Diagrama de Secuencia del Caso de Uso Visualizar Mapas (Fuente: Dima Rodríguez)

Diagrama de secuencia del caso de uso *Visualizar puntos*. Mostrado en la figura 4.5, el usuario en este caso, pueda visualizar la imagen del mapa. Primero se debe seleccionar el tipo de mapa a mostrar, esto se hace por medio de la IUVisualizarMapa, y que se puede visualizar, por medio del enlace de la IUVentanaPrincipal. Una vez seleccionado el tipo de mapa, se va a recuperar los puntos de la base de datos para ser cargada en objetos de tipo punto, pintado en un objeto de tipo lienzo, con ayuda del objeto Puntoreferencia y mostrado en un objeto de tipo IUMapa.

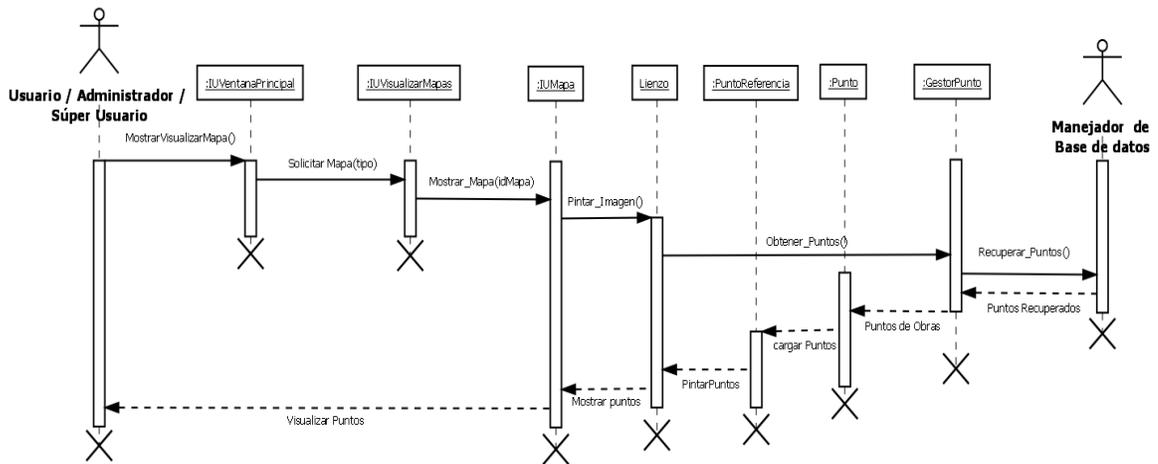


Figura 4.5 Diagrama de Secuencia del Caso de Uso Visualizar Puntos (Fuente: Dima Rodríguez)

El Diagrama de secuencia del caso de uso *Cargar Mapa*, representado en la figura 4.6, muestra como se obtiene y se guardan las imágenes de los mapas a guardar y mostraran en el sistema por el administrador de avance. Al igual que muchos de los diagramas anteriores se hace la petición por medio de la *IUVentanaPrincipal*, mostrar la ubicación de la imagen a ser guardada en la base de datos, esto le corresponde a la *IUAccesoArchivoImagenes*. Luego de seleccionada es cargada en un objeto de tipo *Mapa*, para ser guardada en la Base de datos por medio del *GestorMapa*.

En la figura 4.7, se visualizara el diagrama de secuencia del caso de uso *Consultar Obra*. Para realizar una consulta bien sea de obra, contrato o Avance de contrato en este caso por el usuario, se debe pedir por medio de la *IUVentanaPrincipal*, realizado este pedido, se procede a buscar los datos en la base de datos por medio de los gestores de obra, contrato o Avance de contrato, recibiendo los mismos, si son encontrados. Posteriormente se cargan a los objetos de tipo de dato obra, contrato o Avance de contrato, dependiendo el caso. Y por último se muestran en sus respectivas interfaces, como lo son: *IUConsultarObra*, *IUConsultarContrato* y *IUConsultarAvanceContrato*.

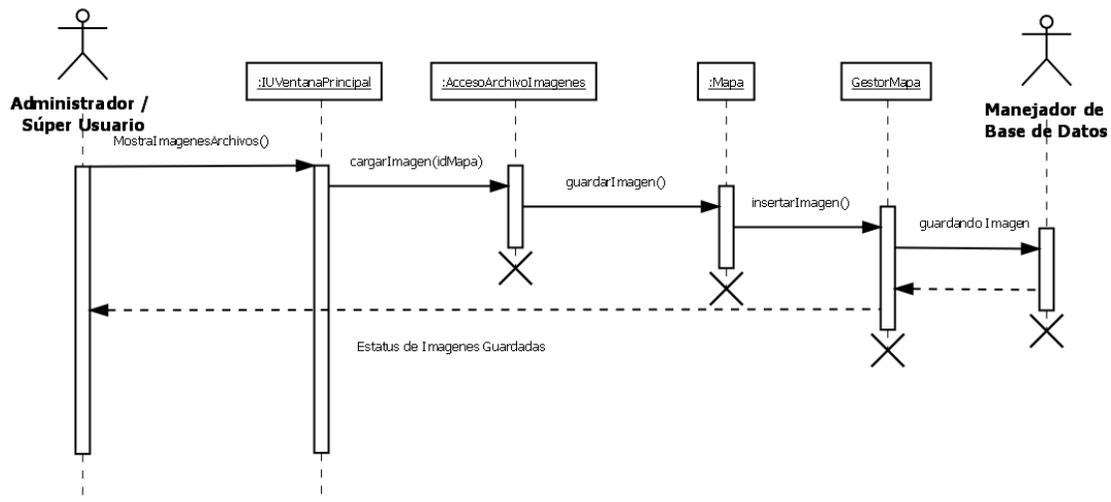


Figura 4.6 Diagrama de Secuencia del Caso de Uso Cargar Mapa (Fuente: Dima Rodríguez)

Diagrama de secuencia del caso de uso *Gestiona Obra*, y es representado en la figura 4.8, la cual abarca parte de lo que es la inserción, modificación y eliminación de una obra, contrato o avance de un contrato. Esto es regido por el administrador de avance de obra, el cual, hace la solicitud, bien sea, para agregar, modificar o eliminar una obra, contrato o avance de contrato. Estos se gestionan por medio de la interfaz IUObra, IUContrato o IUAvanceContrato. En estas interfaces asignamos los datos a ser agregado, modificado o eliminado, para posteriormente ser cargados en sus respectivos objetos de tipo: Obra, Contrato o AvanceContrato. Y así por medio de los gestores (GestorObra, GestorContrato o GestorAvanceContrato) realizar las operaciones en la base de datos.

El Diagrama de secuencia del caso de uso Crear Punto de Referencia a Obra, es uno de los más importantes, ya que, aquí se observa como se realiza el enlace de la obra con el punto que hace referencia al mismo. Como se puede ver, en la figura 4.9, el enlace se realiza tanto para una obra, como para un contrato no asociado.

CAPÍTULO IV: Fase de Elaboración

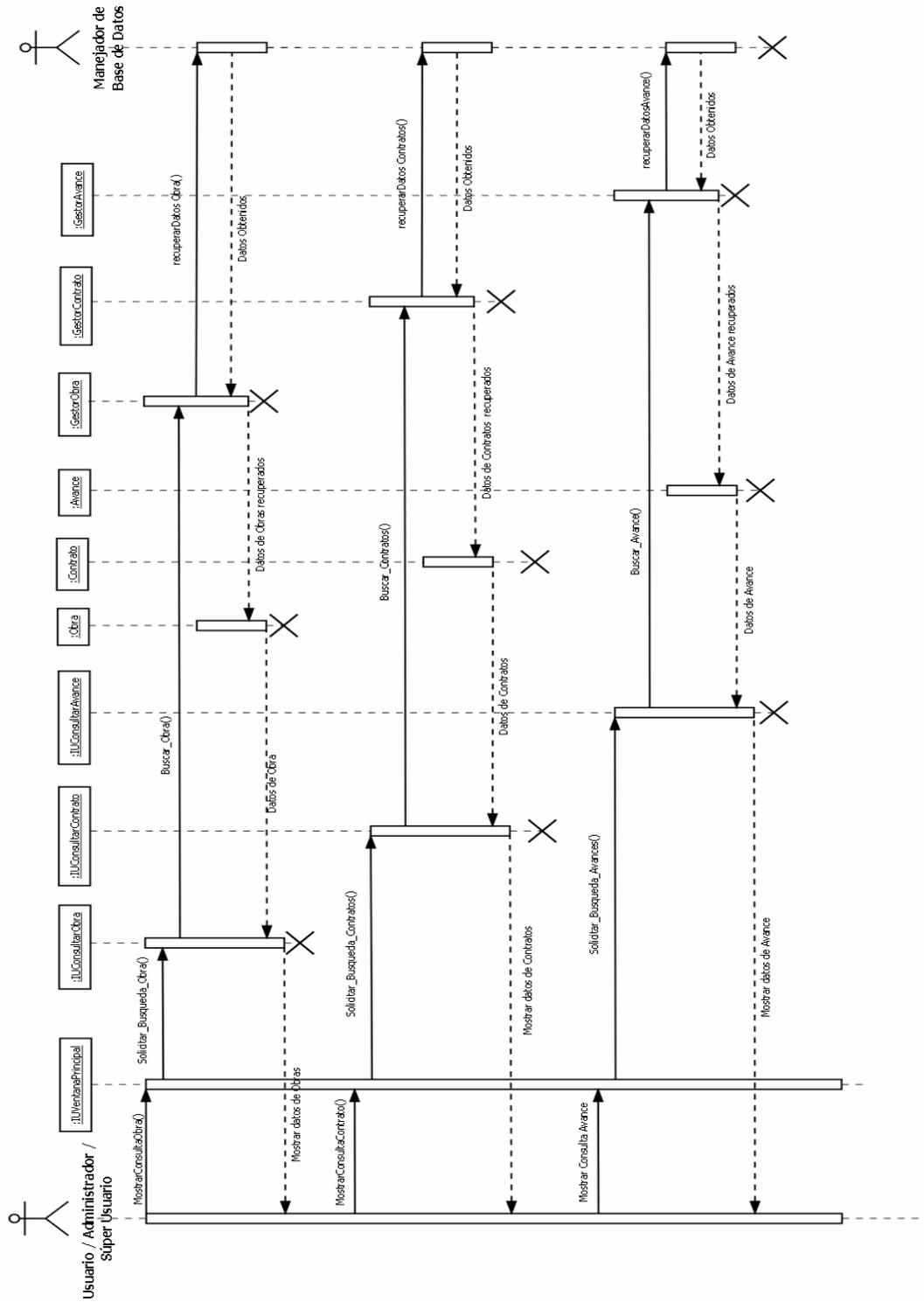


Figura 4.7 Diagrama de Secuencia del Caso de Uso Consultar Obra (Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

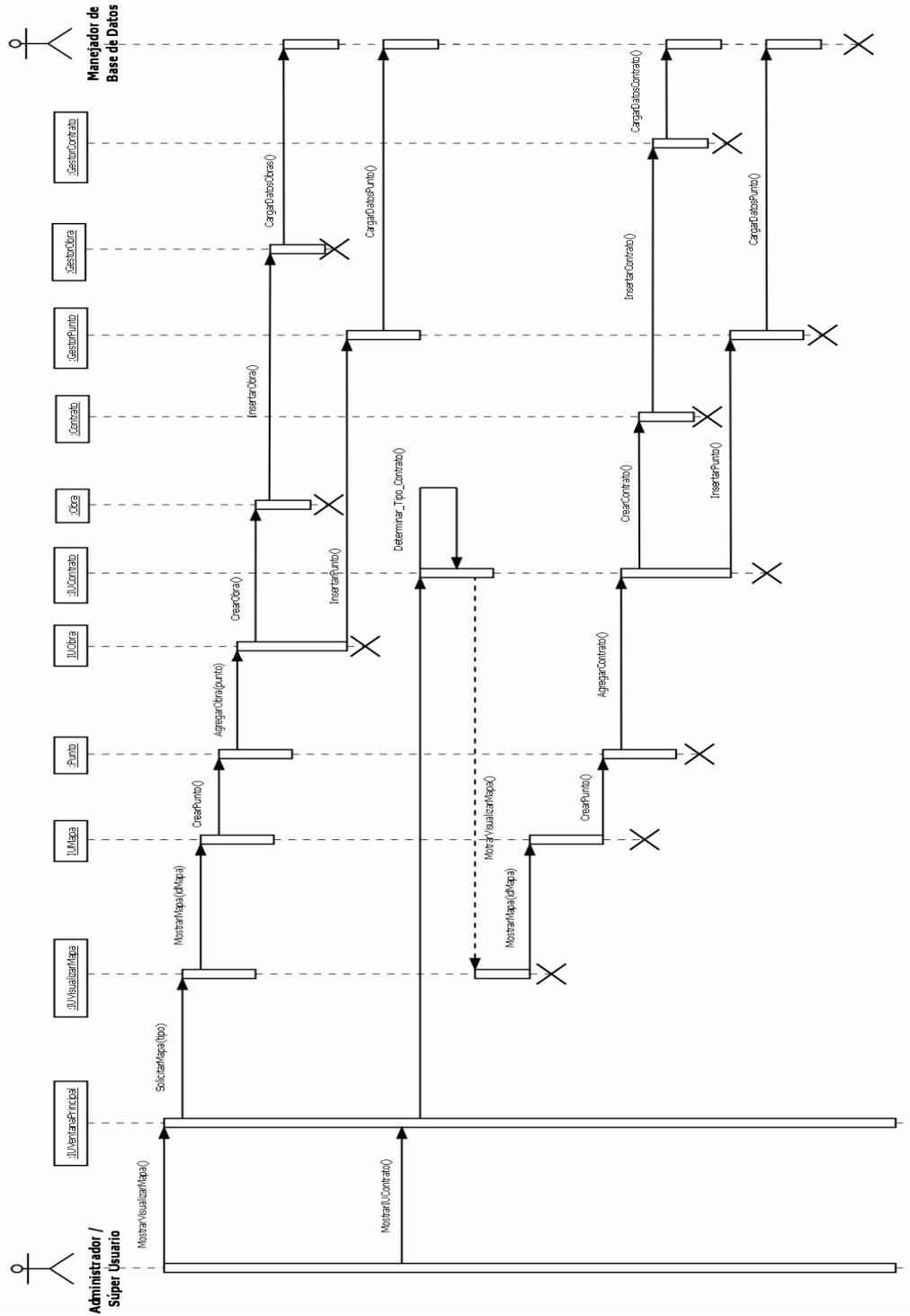


Figura 4.9 Diagrama de Secuencia del Caso de Uso Crear Punto de Referencia a Obra (Fuente: Dima Rodríguez.)

CAPÍTULO IV: Fase de Elaboración

En el primer caso se observa que el administrador de Avance de obra pide por medio de la IUVentanaPrincipal la solicitud de visualizar la IUVisualizarMapa, para determinar a que mapa vamos a anexar el punto, se selecciona el tipo de mapa a mostrar por la interfaz IUMapa .

Luego se procede a crear el punto, para luego ser enviado a la IUObra en donde se procederá a crear la Obra, finalmente por medio de los gestores (GestorPunto y GestorObra), se lleva a cabo las operaciones en la base de datos. Como lo es insertar tanto la obra como el punto.

Cuando la referencia va dirigida a un contrato (No asociado a una obra) la interacción sería de parte del administrador de avance de obra, mostrar la IUContrato para determinar que tipo de contrato sería, de ser No asociado a una obra, mostraría la IUVisualizarMapa, para determinar a que mapa vamos a anexar el punto, se selecciona el tipo de mapa a mostrar en la IUMapa y se procede a crear el punto, rápidamente se envía a la IUContrato en donde se procede a crear el contrato, finalmente por medio de los gestores (GestorPunto y GestorContrato), se llevan a cabo las operaciones en la base de datos. Como lo es insertar tanto el contrato como el punto.

4.5. DISEÑO DE LA BASE DE DATOS

El diseño de la base de datos es una de las partes más importante en el esquema del sistema, ya que permite que la base de datos resultante sea capaz de almacenar toda la información necesaria, a demás, ayuda a garantizar en todo momento la integridad de los datos y evitar errores tales como lo son la conexión errada de las tablas y la duplicidad de la información.

CAPÍTULO IV: Fase de Elaboración

Cada una de las tablas identificadas y que se describen están vinculadas entre sí por alguno de sus atributos. A esta relación se le llama dependencia funcional.

A continuación se podrá ver en la figura 4.10, la base de datos “BD_SISOVE”, el cual, contendrá las respectivas relaciones entre las tablas garantizando así la normalización de la misma.

4.5.1. Descripción de la base de datos

La base de datos del sistema SISOVE (denominada “BD_SISOVE”) está formada por un conjunto de tablas que serán descritas a continuación.

- Tabla Punto: contiene los puntos asociados a las obras.

Tabla 4.8. Tabla Punto

Atributo	Dominio	Longitud	Descripción
idImagenMapa	Texto	3	Identidad del mapa donde está ubicado el punto.
codRefObra	Texto	30	Código de la obra el cual es referido por el punto.
puntoRef	Punto	(8 bytes, 8 bytes)	Coordenadas X,Y para ubicar el punto en la imagen.
tipoObra	Texto	10	Tipo de obra (obra o contrato)
Finalidad	Texto	300	Finalidad de la obra.
Visible	Si/No	1	Visibilidad del punto en el sistema.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

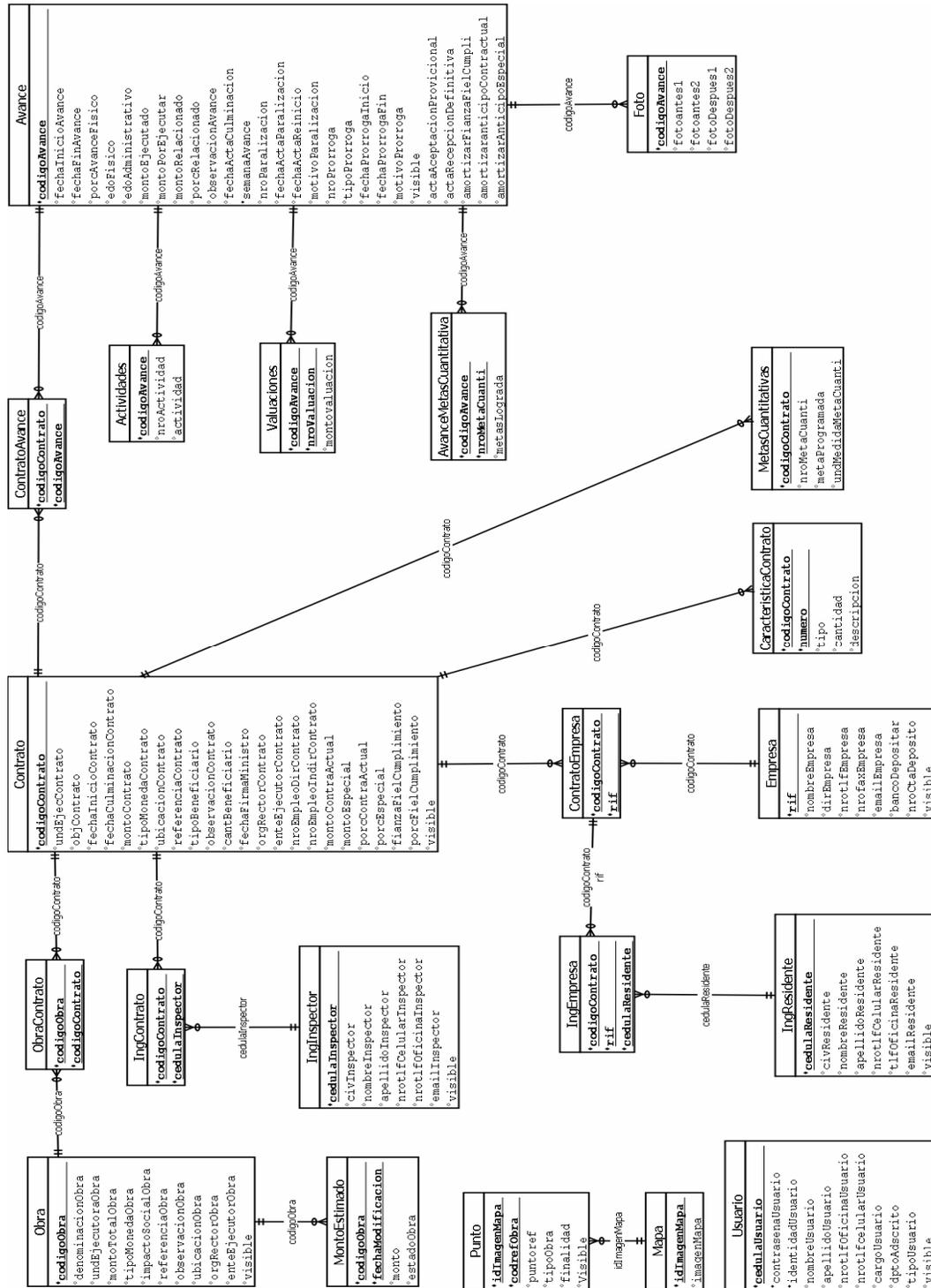


Figura 4.10 Modelo de la Base de Datos BD_SISOVE. (Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

- Tabla Usuario: En esta tabla el sistema almacena los datos de los usuarios que tendrán acceso al sistema.

Tabla 4.9. Tabla Usuario

Atributo	Dominio	Longitud	Descripción
cedulaUsuario	Texto	20	Cedula del usuario.
contrasenaUsuario	Texto	50	Contraseña del usuario.
identidadUsuario	Texto	50	Identidad del usuario.
nombreUsuario	Texto	50	Nombre del usuario.
apellidoUsuario	Texto	50	Apellido del usuario.
nroTlfOficinaUsuario	Texto	30	Teléfono de oficina o casa del usuario.
nroTlfCelularUsuario	Texto	30	Teléfono de celular del usuario.
cargoUsuario	Texto	50	Cargo del usuario en la institución.
tipoUsuario	Texto	20	Tipo de usuario registrado para acceder al sistema.
dptoadscrito	Texto	50	Departamento adscrito al usuario.
visible	Si/No	1	Visibilidad del usuario en el sistema.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

- Tabla Mapa: esta tabla contiene las imágenes de los mapas a mostrar.

Tabla 4.10. Tabla Mapa

Atributo	Dominio	Longitud	Descripción
idImagenMapa	Texto	3	Enlace con la tabla punto.
imagenMapa	Byte	n	Imagen del mapa.

(Fuente: Dima Rodríguez)

- Tabla Obra: La presente tabla se encarga de almacenar los datos de la obra.

Tabla 4.11. Tabla Obra

Atributo	Dominio	Longitud	Descripción
codigoObra	Texto	30	Código de la obra.
denominacionObra	Texto	500	Descripción de la Obra.
undEjecutoraObra	Texto	50	Unidad ejecutora de la obra.
montoTotalObra	Decimal	8 bytes	Monto de la obra.
tipoMonedaObra	Texto	10	Tipo de moneda para representar los montos.
impactoSocialObra	Texto	500	Beneficio que trae la obra.
referenciaObra	Texto	100	Lugar de referencia de la obra.
observacionObra	Texto	500	Observación hecha a la obra.
ubicacionObra	Texto	500	Ubicación física de la obra.
visible	Si/No	1	Visibilidad de la obra en el sistema.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

- Tabla MontoEstimado: contiene los datos de la obra que se pueden modificar a medida que va avanzando la obra.

Tabla 4.12. MontoEstimado

Atributo	Dominio	Longitud	Descripción
codigoObra	Texto	30	Código de la obra con el cual se relacionan los avances de la obra.
fechaModificacion	Fecha	aaaa-mm-dd	Fecha en que se modifico el monto.
monto	Decimal	8 bytes	Monto estimado para terminar la obra.
estadoObra	Texto	30	Estado físico de la obra.

(Fuente: Dima Rodríguez)

- Tabla ObraContrato: Esta tabla enlaza las obras con los contratos que serán asociados a ella.

Tabla 4.13. Tabla ObraContrato

Atributo	Dominio	Longitud	Descripción
CodigoContrato	Texto	30	Código del contrato que relaciona la obra.
CodigoObra	Texto	30	Código de la obra que relaciona los contratos.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

- Tabla Contrato: Esta tabla relaciona los contratos asignados a la obra.

Tabla 4.14. Tabla Contrato (1/3)

Atributo	Dominio	Longitud	Descripción
UbicacionContrato	Texto	500	Ubicación de la obra.
ReferenciaContrato	Texto	400	Punto de referencia de la obra.
TipoBeneficiario	Texto	30	Beneficiarios del contrato.
ObservacionContrato	Texto	500	Observaciones del contrato.
CantBeneficiario	Numérico	4 bytes	Cantidad de beneficiarios del contrato.
FechaFirmaMinistro	Dato	aaaa-mm-dd	Fecha en que el ministro firmo el contrato.
OrgRectorContrato	Texto	100	Órgano rector del contrato
EnteEjecutorContrato	Texto	100	Ente ejecutor de del contrato.
NroEmpleoDirContrato	Decimal	8 bytes	Número estimado de empleos directos.
NroEmpleoIndirContrato	Decimal	8 bytes	Número estimado de empleos indirectos.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.15. Tabla Contrato (2/3)

Atributo	Dominio	Longitud	Descripción
MontoContactual	Decimal	8 bytes	Monto de financiamiento otorgado para iniciar la obra.
MontoEspecial	Decimal	8 bytes	Monto de financiamiento otorgado de manera especial.
PorcEspecial	Decimal	8 bytes	Porcentaje del monto especial.
FianzaAnticipo	Decimal	8 bytes	Porcentaje del monto de financiamiento otorgado de manera especial.
FianzaFielCumplimiento	Decimal	8 bytes	Fondo requerido para garantizar el cumplimiento del contrato.
FianzaAnticipoEspecial	Decimal	8 bytes	Fondo requerido para garantizar el pago de anticipos especiales.
PorcAnticipo	Decimal	8 bytes	Porcentaje del anticipo.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.16. Tabla Contrato (3/3)

Atributo	Dominio	Longitud	Descripción
PorcFielCumplimiento	Decimal	8 bytes	Porcentaje por la fianza de fiel cumplimiento.
PorcAnticipoEspecial	Decimal	8 bytes	Porcentaje por el anticipo especial.
Visible	Si/No	1	Visibilidad en el sistema.

(Fuente: Dima Rodríguez)

- Tabla MetasCuantitativas: Contiene las diferentes Metas programadas del contrato.

Tabla 4.17. Tabla MetasCuantitativas

Atributo	Dominio	Longitud	Descripción
CodigoContrato	Texto	30	Código del contrato.
NroMetaCuanti	Numérico	4 bytes	Número de la meta cuantitativa.
MetaProgramada	Decimal	8 bytes	Longitud de la meta
UndMedidaMetaCuanti	Texto	50	Unidad de medida.

(Fuente: Dima Rodríguez)

- Tabla CaracteristicaContrato: En esta tabla se almacena las características del contrato.

Tabla 4.18. Tabla CaracteristicaContrato(1/2)

Atributo	Dominio	Longitud	Descripción
CodigoContrato	Texto	30	Código del contrato.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.19. Tabla CaracteristicaContrato(2/2)

Atributo	Dominio	Longitud	Descripción
Numero	Numérico	4 bytes	Número de la memoria descriptiva.
Tipo	Texto	300	Tipo de memoria descriptiva.
Cantidad	Numérico	4 bytes	Cantidad de memoria descriptiva.
Descripcion	Texto	500	Descripción de memoria descriptiva.

(Fuente: Dima Rodríguez)

- Tabla Empresa: Contiene los datos de las empresas que ejecutan los contratos.

Tabla 4.20. Tabla Empresa

Atributo	Dominio	Longitud	Descripción
RIF	Texto	50	Registro de información fiscal.
NombreEmpresa	Texto	500	Nombre de la Empresa
DirEmpresa	Texto	500	Dirección de la Empresa.
NroTlfEmpresa	Texto	30	Número telefónico de la Empresa.
NroFaxEmpresa	Texto	30	Número de fax de la Empresa.
EmailEmpresa	Texto	50	Dirección de correo de la Empresa.
BancoDepositar	Texto	100	Nombre del banco a depositar.
NroCtaDeposito	Texto	100	Numero de cuenta a depositar.
Visible	Si/No	1	Visibilidad en el sistema.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

- Tabla ContratoEmpresa: Tabla que enlaza la empresa con el contrato que ejecuta.

Tabla 4.21. Tabla ContratoEmpresa

Atributo	Dominio	Longitud	Descripción
CodigoContrato	Texto	30	Código del Contrato.
RIF	Texto	50	Registro de información fiscal.

(Fuente: Dima Rodríguez)

- Tabla IngEmpresa: Esta tabla enlaza a la empresa con el ingeniero residente.

Tabla 4.22. Tabla IngEmpresa

Atributo	Dominio	Longitud	Descripción
CedulaResidente	Texto	40	Cedula del Ingeniero residente.
RIF	Texto	50	Registro de información fiscal.
CodigoContrato	Texto	30	Código del Contrato.

(Fuente: Dima Rodríguez)

- Tabla IngResidente: Contiene los datos del ingeniero residente.

Tabla 4.23. Tabla IngResidente(1/2)

Atributo	Dominio	Longitud	Descripción
CedulaResidente	Texto	40	Cedula del Ingeniero residente.
CivResidente	Texto	30	Número asignado por el colegio de Ingenieros de Venezuela.
NombreResidente	Texto	40	Nombre del Ingeniero residente.
ApellidoResidente	Texto	40	Apellido del Ingeniero residente.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.24. Tabla IngResidente(2/2)

Atributo	Dominio	Longitud	Descripción
NroTlfCelularResidente	Texto	30	Número celular del Ingeniero residente.
NroTlfOficinaResidente	Texto	30	Número de Oficina del Ingeniero residente.
EmailResidente	Texto	50	Correo electrónico del Ingeniero residente.
Visible	Si/No	1	Visibilidad en el sistema.

(Fuente: Dima Rodríguez)

- Tabla ContratoAvance: Tabla que enlaza el contrato con el avance del mismo.

Tabla 4.25. Tabla ContratoAvance

Atributo	Dominio	Longitud	Descripción
CodigoContrato	Texto	30	Código del Contrato.
CodigoAvance	Texto	30	Código del Avance del contrato.

(Fuente: Dima Rodríguez)

- Tabla Avance: contiene los datos del avance de un contrato

Tabla 4.26. Tabla Avance (1/3)

Atributo	Dominio	Longitud	Descripción
CodigoAvance	Texto	30	Código del Avance del contrato.
FechaInicioAvance	Dato	aaaa-mm-dd	Fecha en que se inicia la inspección de la obra.
FechaFinAvance	Dato	aaaa-mm-dd	Fecha en que se culmina la inspección de la obra.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.27. Tabla Avance (2/3)

Atributo	Dominio	Longitud	Descripción
PorcAvanceFisico	Decimal	8 bytes	Porcentaje de avance físico.
EdoFisico	Texto	25	Estado físico de la obra.
EdoAdministrativo	Texto	20	Estado administrativo de la obra.
MontoEjecutado	Decimal	8 bytes	Monto ejecutado del contrato.
MontoPorEjecutar	Decimal	8 bytes	Monto por ejecutar del contrato.
MontoRelacionado	Decimal	8 bytes	Monto relacionada a la fecha.
PorcRelacionado	Decimal	8 bytes	Porcentaje relacionado.
ObservacionAvance	Texto	1000	Observaciones hechas al avance.
FechaActaCulminacion	Dato	aaaa-mm-dd	Fecha de culminación del contrato.
SemanaAvance	Numérico	4 bytes	Semana de avance.
NroParalizacion	Numérico	4 bytes	Número de paralización.
FechaActaParalizacion	Dato	aaaa-mm-dd	Fecha de paralización.
FechaActaReinicio	Dato	aaaa-mm-dd	Fecha de reinicio.
MotivoParalizacion	Texto	100	Motivo de la paralización.
NroProrroga	Numérico	4 bytes	Número de prórroga.
TipoProrroga	Texto	15	Tipo de prórroga.
FechaProrrogaInicio	Dato	aaaa-mm-dd	Fecha de inicio de prórroga.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.28. Tabla Avance(3/3)

Atributo	Dominio	Longitud	Descripción
FechaProrrogaFin	Dato	aaaa-mm-dd	Fecha de finalización de la prórroga.
MotivoProrroga	Texto	100	Motivo de la prórroga.
Visible	Si/No	1	Visibilidad en el sistema.
ActaAceptacionProvisional	Dato	aaaa-mm-dd	Fecha en que se acepto provisionalmente la obra.
ActaRecepcionDefinitiva	Dato	aaaa-mm-dd	Fecha en que se recibió definitivamente la obra.
AmortizarFianzaFielCumpli	Decimal	8 bytes	Monto que amortizan la fianza fiel cumplimiento.
AmortizarFianzaAnticipo	Decimal	8 bytes	Monto que amortizan la fianza de anticipo.
AmortizarFianzaAnticipoEspecial	Decimal	8 bytes	Monto que amortizan la fianza de anticipo especial.
AmortizarAnticipoContractual	Decimal	8 bytes	Monto que amortizan la cantidad dada por el Anticipo Contractual.
AmortizarAnticipoEspecial	Decimal	8 bytes	Monto que amortizan la cantidad dada por el Anticipo especial.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

- Tabla IngContrato: Tabla que enlaza al ingeniero Inspector con el contrato que supervisa.

Tabla 4.29. Tabla IngContrato

Atributo	Dominio	Longitud	Descripción
CodigoContrato	Texto	30	Código del contrato.
CedulaInspector	Texto	40	Cedula del ingeniero inspector.

(Fuente: Dima Rodríguez)

- Tabla IngInspector: contiene los datos del ingeniero Inspector.

Tabla 4.30. Tabla IngInspector (1/2)

Atributo	Dominio	Longitud	Descripción
CedulaInspector	Texto	40	Cedula del Ingeniero inspector.
CivInspector	Texto	30	Número asignado por el colegio de Ingenieros de Venezuela.
NombreInspector	Texto	40	Nombre del Ingeniero inspector.
ApellidoInspector	Texto	40	Apellido del Ingeniero inspector.
NroTlfCelularInspector	Texto	30	Número celular del Ingeniero inspector.
NroTlfOficinaInspector	Texto	30	Número de Oficina del Ingeniero inspector.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.31. Tabla IngInspector (2/2)

Atributo	Dominio	Longitud	Descripción
EmailInspector	Texto	50	Correo electrónico del Ingeniero inspector.
Visible	Si/No	1	Visibilidad en el sistema.

(Fuente: Dima Rodríguez)

- Tabla AvanceMetasCuantitativa: contiene las metas cuantitativas logradas.

Tabla 4.32. Tabla AvanceMetasCuantitativa

Atributo	Dominio	Longitud	Descripción
CodigoAvance	Texto	30	Código del Avance del contrato.
NroMetaCuanti	Numérico	4 bytes	Número de la meta cuantitativa.
MetasLograda	Decimal	8 bytes	Meta lograda.

(Fuente: Dima Rodríguez)

- Tabla Foto: Contiene las fotos obtenidas en la medidas que se va ejecutando el contrato.

Tabla 4.33. Tabla Foto (1/2)

Atributo	Dominio	Longitud	Descripción
CodigoAvance	Texto	30	Código del Avance del contrato.
FotoAntes1	Byte	n	Foto 1 tomada antes de la actividad.

(Fuente: Dima Rodríguez)

CAPÍTULO IV: Fase de Elaboración

Tabla 4.34. Tabla Foto (2/2)

Atributo	Dominio	Longitud	Descripción
FotoAntes2	Byte	n	Foto 2 tomada antes de la actividad.
FotoDespues1	Byte	n	Foto 1 tomada después de la actividad.
FotoDespues2	Byte	n	Foto 2 tomada después de la actividad.

(Fuente: Dima Rodríguez)

- Tabla Actividades: contiene las actividades ejecutadas del contrato.

Tabla 4.35. Tabla Actividades

Atributo	Dominio	Longitud	Descripción
CodigoAvance	Texto	30	Código del Avance del contrato.
NroActividad	Numérico	4 bytes	Número de la actividad.
Actividad	Texto	500	Descripción de la actividad.

(Fuente: Dima Rodríguez)

- Tabla Valuaciones: contiene las valuaciones realizadas al contrato.

Tabla 4.36. Tabla Valuaciones

Atributo	Dominio	Longitud	Descripción
CodigoAvance	Texto	30	Código del Avance del contrato.
NroValuacion	Numérico	4 bytes	Número de la valuación.
MontoValuacion	Decimal	8 bytes	Monto de la valuación.

(Fuente: Dima Rodríguez)

4.6. DISEÑO DE LA INTERFAZ

Uno de los aspectos más importantes del proceso de desarrollo de software es el diseño de la interfaz de usuario, su importancia radica en que, por medio de esta se hace posible la interacción entre el usuario y el sistema.

El buen diseño de las interfaces de software tiene un propósito, debido a que el objetivo de éstas es mantener una consistencia con la realidad, es decir, deben proporcionar al usuario toda la información que éstos requieran para el autoaprendizaje en relación al software y la realidad a la que modela, además deben ofrecer de una manera razonable información acerca de las acciones que éste ejecuta. Las interfaces permiten un ambiente agradable al usuario, para que así éste se sienta a gusto al interactuar con el sistema.

Se deben considerar aspectos asociados a facilitar dicha interacción, tales como: realizar un diseño de pantallas que cumplan con las siguientes normas; deben ser claras, sencillas, no se deben recargar de información, las pantallas deben ser consistentes, se deben usar los colores adecuados y con una buena selección de menús de usuarios.

4.6.1. Diseño de la ventana de Validación

En el siguiente diseño se muestra cómo serán ubicados los bloques principales que conformaran la interfaz para entrar al sistema, esta será la primera pantalla con la que el usuario interactuará, lo puede observar en la siguiente figura 4.11.

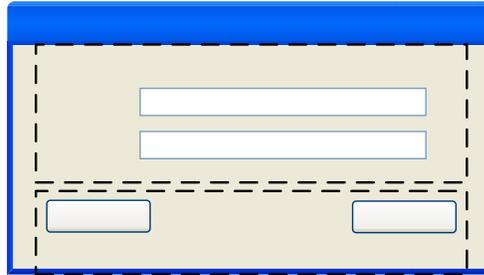


Figura 4.11 Ventana de Validación (Fuente: Dima Rodríguez)

4.6.2. Diseño de la Ventana Principal

En el diseño mostrado a continuación se observan los bloques principales que van a componer la Interfaz Principal del sistema.

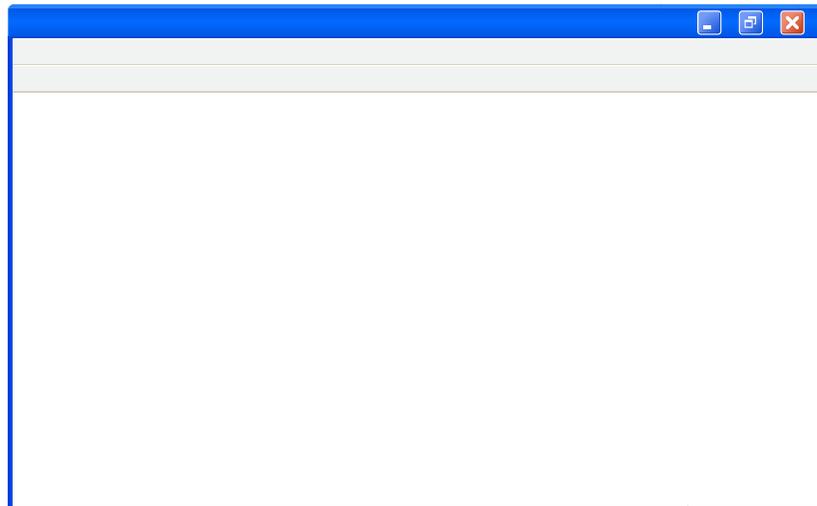


Figura 4.12 Ventana principal del sistema (Fuente: Dima Rodríguez)

4.6.3. Diseño de las Ventanas Internas o Hijas

En el diseño mostrado a continuación se observan los bloques principales que van a componer la Interfaz de las ventanas hijas que contendrán información relevante para el sistema.

Título de la ven

Área c

Identidad Escr

Clave Escr

Aceptar

Botó

Inicialmente se tendrá una zona donde se cargará al sistema la información general, y se puede observar en la figura 4.13. Posteriormente se tiene un conjunto de fichas para seguir cargando más información al sistema.

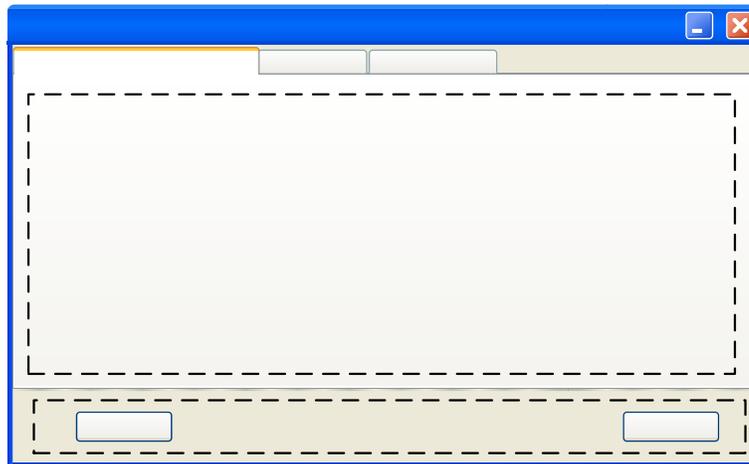


Figura 4.13 Ventanas hijas (Fuente: Dima Rodríguez)

4.7. DIAGRAMA DE COMPONENTES

Título de la Ventana Interna Fichas (JTabbedPane)

El diagrama de componentes se encarga de mostrar el modelado de la estructura del software, incluyendo las dependencias entre los componentes de software. Cada componente es un módulo de software y hace parte de la vista física del sistema, el diagrama de componentes del sistema lo se puede apreciar en la figura 4.14

4.8. DIAGRAMA DE DESPLIEGUE

En este diagrama se describe la arquitectura física del sistema durante la ejecución de la aplicación, su topología, la estructura de los elementos de hardware y el software que ejecuta cada uno de ellos. En la figura 4.15 se puede observar.

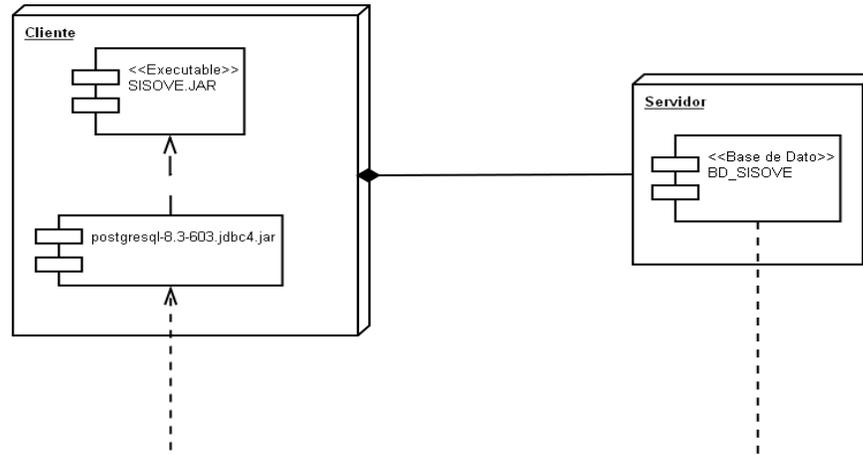


Figura 4.15 Diagrama de despliegue del sistema (Fuente: Dima Rodríguez)

4.9. RESUMEN DE LA FASE DE ELABORACIÓN

Durante la presente fase en el desarrollo de este proyecto se logró concretar de manera significativa la estructura del software a implementar, además de definir el diseño de estructuras básicas como lo son el diagrama de clases, modelo de la base de datos y las interfaces de usuarios, con esto se logra alcanzar una visión bastante aproximada de la estructura final del sistema.

En este capítulo se obvió la implementación, ya que, el trabajo se centró mayormente en la actividad de diseño, como consecuencia no hubo implementación. Tampoco se realizaron pruebas en esta fase pues, tal actividad quedará para la fase posterior.

CAPÍTULO V: FASE DE CONSTRUCCIÓN

5.1. INTRODUCCIÓN

En este capítulo se iniciará la fase de construcción, y tiene como propósito la obtención de un producto en su versión operativa inicial, el cuál deberá tener la calidad adecuada para su aplicación y cumplir con todos los requisitos establecidos en las fases anteriores. Esto significa describir los requisitos restantes, refinando el diseño y completando la implementación y las pruebas de software.

En esta fase los distintos modelos del sistema van en aumento hasta completarse. Sin embargo, la descripción de la arquitectura no crece significativamente debido a que la mayor parte de esta arquitectura se definió durante la fase de elaboración.

5.2. HERRAMIENTAS DE DESARROLLO UTILIZADAS

Para la construcción del sistema se tomo en cuenta un conjunto de herramientas seleccionadas según las necesidades y tomando en consideración características que fueron definidas previamente por la empresa.

- ▣ **Herramientas software libre:** la totalidad del software utilizado en este proyecto debe cumplir con las especificaciones descritas para ser considerado como “software libre”.
- ▣ **Sistemas operativos GNU/Linux o Windows:** Las herramientas así como el software resultante deben proporcionar la completa funcionalidad tanto en el sistema operativo GNU/Linux como en Windows.

5.2.1. Lenguaje de programación Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre.

5.2.2. Manejador de base de datos PostgreSQL

Postgres o PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

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.

Sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 15 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.

5.3. COMPONENTES DEL SISTEMA

En esta fase se detalla el diagrama de componentes presentado en el capítulo anterior (ver Figura 5.1) y se profundizara en la codificación de los componentes más importantes.

5.3.1. Codificación de los componentes

A continuación se presenta la codificación de los principales componentes necesarios para la ejecución de la aplicación, estas serian las referentes a Mapas, Lienzo y Punto de referencia a las obras.

5.3.1.1. Codificación de los componentes relacionados a los mapas y puntos de referencia a las obras.

A continuación presentaremos la construcción de las diversas clases que comprenden a los mapas y los puntos de referencia a las obras, como lo son: IUMapa, ImagenMapa, Lienzo, Mapa, GestorMapa, PuntoReferencia, Punto y GestorPunto.

CAPITULO V: Fase de Construcción

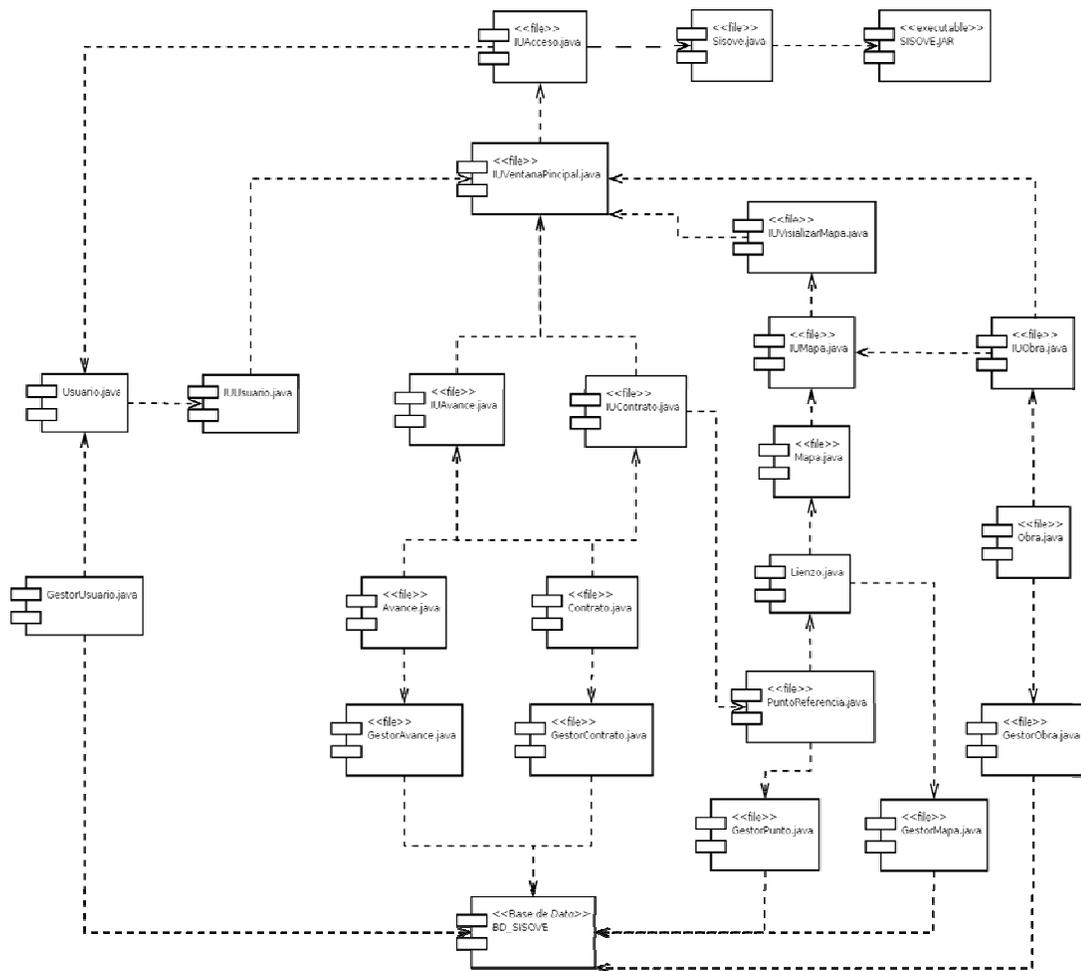


Figura 5.1 Diagrama de Componente del sistema SISOVE. (Fuente: Dima Rodríguez)

UIMapa.java

import interfaz.InterfazVentanaPadre;

import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.Cursor;

import java.awt.Point;

import java.awt.event.MouseEvent;

import java.awt.event.MouseListener;

import java.awt.event.MouseMotionListener;

CAPITULO V: Fase de Construcción

```
import java.util.ArrayList;
import java.util.Collection;
import java.util.Iterator;

import javax.swing.BorderFactory;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JInternalFrame;

import componeMapa.Lienzo;
import componeMapa.PuntoReferencia;

import javax.swing.JScrollPane;

import modelo.AvanceContrato;
import modelo.MontoEstimado;
import modelo.Contrato;
import modelo.Obra;
import modelo.Punto;

import org.postgresql.geometric.PGpoint;

import principal.SISOVE;
import validaciones.Validar;
import ventanasDialogos.VDInforPuntos;

import controlador.GestorAvanceContrato;
import controlador.GestorMontoEstimado;
import controlador.GestorContrato;
import controlador.GestorObra;
import controlador.GestorObraContrato;
import controlador.GestorPunto;
import controlador.SAVEException;

import formatoDatos.Formatos;
import javax.swing.ImageIcon;

/**
 * Clase IUMapa: utilizada para crear objetos de tipo JInternalFrame
 * Ésta muestra una ventana que es de utilidad a la hora de mostrar
 * la imagen de los mapas y los puntos de referencia a las obras.
 */
```

CAPITULO V: Fase de Construcción

```

    *@author Dima Rodríguez
    *
    *@version 1.0.0
    *
    */
    @SuppressWarnings("serial")
    public class IUMapa extends JFrame {

        /**
         * Contador que se incrementa cada vez que se abre una ventana
         */
        static int CountSubVentAbiertas = 0;

        /**
         * Constantes Utilizados para mover la localización de las ventanas
         */
        static final int moverVentX = 30, moverVentY = 30;

        /**
         * Utilizado para capturar la identificación del mapa
         */
        private String idMapa=null; // @jve:decl-index=0:

        /**
         * Utilizado para crear el contenedor principal
         */
        private JPanel jContentPane = null;

        /**
         * Utilizado para la creación de un objeto de tipo lienzo para la ventana
         */
        private Lienzo lienzo = null;

        /**
         * Utilizado principalmente para identificar el tipo de usuario que utiliza el sistema
         */
        private IUVentanaPrincipal principal=null;

        private JScrollPane spLienzo = null;

        private ManejadorUIMapa manejadorUIMapa = null;
    }

```

CAPITULO V: Fase de Construcción

```

private ArrayList<PuntoReferencia> puntos;

final String ESTADO_1 = "SIN CONTRATO";

final String ESTADO_2 = "SIN AVANCE";

public IUVisualizarMapa mapa=null;

/**
 * Constructor de la clase
 * @param principal
 * @param mapa
 * @param idMapa: utilizada para referenciar a un tipo de mapa
 * en especial
 */
public IUMapa(String idMapa,String tituloMapa, IUVentanaPrincipal principal, IUVisualizarMapa mapa) {
    this.principal=principal;
    this.setIdMapa(idMapa);
    this.setName(idMapa);
    //coloca el titulo a la ventana
    this.setTitle(tituloMapa);
    this.mapa=mapa;
    initialize();
}

/**
 * Método que llama al recolector de basura
 * cuando se ha dejado de utilizar este objeto
 */
protected void finalize(){
    setCountSubVentAbiertas(--CountSubVentAbiertas);
    lienzo.setImagenMapa(null);
    lienzo = null;
    /**
     * Realiza la llamada al finalizador del objeto IUMapa
     */
    System.gc();
}

/**
 * Metodo donde se inicializan y se cargan

```

CAPITULO V: Fase de Construcción

```

* los atributos de la clase
* @return void
*/
private void initialize() {
    //incrementa el contador a medida que se van abriendo
    //subventanas
    setCountSubVentAbiertas(++CountSubVentAbiertas);
    //tamaño inicial de la ventana
    this.setSize(387, 249);
    //crea el objeto arreglo punto
    this.setIcon(new ImageIcon(getClass().getResource("/iconos/map.png")));
    //arreglo de los puntos a pintar en el mapa
    this.puntos = new ArrayList<PuntoReferencia>();
    //carga los puntos de las obras en el mapa
    cargarPuntos(idMapa);
    //agrega el panel principal a al ventana
    this.setContentPane(getContentPane());
    //Selecciona u obtiene si el frame interno puede ser minimizado.
    this.setIconifiable(true);
    //Selecciona u obtiene si el frame interno puede ser redimensionado.
    this.setResizable(true);
    //Selecciona u obtiene si el usuario puede maximizar el frame interno.
    this.setMaximizable(true);
    //Selecciona u obtiene si el usuario puede cerrar el frame interno.
    this.setClosable(true);
    //Muestra las ventanas en cascada
    this.setLocation(getMoverVentX()*getCountSubVentAbiertas(),
getMoverVentY()*getCountSubVentAbiertas());
    //Hace visible la Ventana
    this.setVisible(true);
}

/**
* Método que crea el panel principal
* y se agrega a él las barras espaciadoras
* y dentro de éstas el objeto lienzo.
*
* @return javax.swing.JPanel
*/
private JPanel getContentPane() {
    if (jContentPane == null) {
        jContentPane = new JPanel();
        jContentPane.setLayout(new BorderLayout());
    }
}

```

CAPITULO V: Fase de Construcción

```
        jPanelContentPane.add(getSpLienzo(), BorderLayout.CENTER);
    }
    return jPanelContentPane;
}

/**
 * Método que me retorna cuanto se debe mover la
 * ventana con respecto a X
 * @return moverVentX
 */
public static int getMoverVentX() {
    return moverVentX;
}

/**
 * Método que me retorna cuanto se debe mover la
 * ventana con respecto a Y
 * @return moverVentY
 */
public static int getMoverVentY() {
    return moverVentY;
}

/**
 * Método para actualizar el atributo CountSubVentAbiertas
 * con respecto a la cantidad de las ventanas abiertas
 * @param countSubVentAbiertas
 */
public static void setCountSubVentAbiertas(int countSubVentAbiertas) {
    CountSubVentAbiertas = countSubVentAbiertas;
}

/**
 * Método que me retorna el numero de ventanas
 * abiertas
 * @return
 */
public static int getCountSubVentAbiertas() {
    return CountSubVentAbiertas;
}

/**
 * Método para crear un objeto de tipo lienzo
```

CAPITULO V: Fase de Construcción

```

* en donde se mostrara el mapa y los puntos
* de referencia a la obra
*
* @return componentVentMapa.Lienzo
*/
public Lienzo getLienzo() {
    if (lienzo == null) {
        lienzo = new Lienzo(idMapa);
    }
    lienzo.setBorder(BorderFactory.createBevelBorder(0));
    lienzo.addMouseListener(getManejadorUIMapa());
    lienzo.addMouseMotionListener(getManejadorUIMapa());
    return lienzo;
}

/**
 * Método para inicializar o actualizar la
 * identidad de la imagen del mapa
 * @param idMapa
 */
public void setIdMapa(String idMapa) {
    this.idMapa = idMapa;
}

/**
 * Retorna la identidad del mapa
 * @return
 */
public String getIdMapa() {
    return idMapa;
}

/**
 * Método utilizado para crear las barras
 * desplazadoras para el objeto lienzo
 *
 * @return javax.swing.JScrollPane
 */
private JScrollPane getSpLienzo() {
    if (spLienzo == null) {
        spLienzo = new JScrollPane();
        spLienzo.setViewportView(getLienzo());
    }
}

```

CAPITULO V: Fase de Construcción

```

    }
    return spLienzo;
}

/**
 * Método para cargar los puntos guardados en la BD
 * @param idmapa
 */
@SuppressWarnings("unchecked")
public void cargarPuntos(String idmapa) {

    GestorPunto gestorPunto = new GestorPunto();
    Collection grupoPuntos=null;
    Iterator iterar = null;
    Punto puntoMapa = null;

    try {
        //recupera la colección de puntos para un determinado mapa
        grupoPuntos = (Collection) gestorPunto.buscarPorIdentidadMapa(idmapa);

    } catch (SAVException e) {

        e.printStackTrace();
    }

    //valida que existan puntos en el mapa
    if(grupoPuntos == null){

        //Inicializa el arreglo
        this.puntos = new ArrayList<PuntoReferencia>();
        //va a cargar los puntos en el mapa
    }else{

        this.puntos = new ArrayList<PuntoReferencia>();

        iterar = grupoPuntos.iterator();

        //recorre la lista
        while(iterar.hasNext()){

            puntoMapa = new Punto();

```

CAPITULO V: Fase de Construcción

```

        //carga el punto de la lista
        puntoMapa = (Punto) iterar.next();
        //pinta el punto
        PuntoReferencia punto = new
PuntoReferencia((int)puntoMapa.getPuntoReferencia().x,(int)puntoMapa.getPuntoReferencia().y, puntoMapa.getCodRefObra(),
puntoMapa.getFinalidad());
        //carga la finalidad de la obra
        punto.setFinalidad(puntoMapa.getFinalidad());
        //carga el tipo de obra
        punto.setTipoObra(puntoMapa.getTipoObra());
        //determina el tipo de obra para determinar la coloración
        if(punto.getTipoObra().equals("Obra")){ //NOTA: cambiar el string por un entero para determinar la obra y el
contrato
            //pinta el color del punto dependiendo el estado de la obra
            punto.setColorPunto(colorEstadoObra(punto.getCodRefObra()));
        }else{
            //pinta el color del punto dependiendo el estado del contrato
            punto.setColorPunto(colorEstadoContrato(punto.getCodRefObra()));
        }
        //agrega los puntos a la lista
        puntos.add(punto);
        //envia los puntos al lienzo
        getLienzo().setPuntos(puntos);
    }
}

/**
 * retorna el color dependiendo el estado del contrato
 *
 * @param codRefObra
 *
 * @return Color
 */
private Color colorEstadoContrato(String codRefObra) {

    String estadoContrato="";

    //obtiene el estado físico del contrato
    estadoContrato = estadoFisicoContrato(codRefObra);

```

CAPITULO V: Fase de Construcción

```

//determina si no tiene avance
if(estadoContrato == ESTADO_2){

    return (new Color(153, 204, 255));

}else{

    //EJECUTANDOSE
    if(estadoContrato.equals(SISOVE.ESTATUS_FISICO[0])){
        return (new Color(255, 255, 102));

    //PARALIZADO
    }else if(estadoContrato.equals(SISOVE.ESTATUS_FISICO[1])){
        return (new Color(255, 51, 0));

    //CULMINADO
    }else if(estadoContrato.equals(SISOVE.ESTATUS_FISICO[2])){
        return (new Color(0, 255, 204));

    }

}

//color inicial
return (new Color(153, 204, 255));

}

/**
 * Determina el estado fisico del contrato no asociado
 * a obra (OBRA)
 *
 * @param codigoContrato
 *
 * @return
 */
public String estadoFisicoContrato(String codigoContrato){

    AvanceContrato avance = new AvanceContrato();

    final String ESTADO_1 = "SIN AVANCE";

    try {

        avance = (AvanceContrato) new GestorAvanceContrato().buscarPorOtro(codigoContrato, 1);

```

CAPITULO V: Fase de Construcción

```
    } catch (SAVException e) {  
  
        e.printStackTrace();  
    }  
  
    if(avance!=null){  
  
        return avance.getEdoFisico();  
    }  
  
    return ESTADO_1;  
}  
  
/**  
 * retorna el color dependiendo el estado de la obra  
 *  
 * @param codRefObra  
 *  
 * @return Color  
 */  
private Color colorEstadoObra(String codRefObra) {  
  
    String estado = obtenerAvanceObra(codRefObra).getEstadoObra().toString();  
  
    String estadoObra = estatusFisicoObra(codRefObra);  
  
    //Sin contrato  
    if(estadoObra == ESTADO_1){  
  
        return (new Color(0, 0, 255));  
    }  
  
    //Sin avance  
    if(estadoObra == ESTADO_2){  
  
        return (new Color(102, 102, 102));  
    }  
  
    //con avance pero obra no culminada  
    } else if(estado.equals("No")){
```

CAPITULO V: Fase de Construcción

```

        //EJECUTANDOSE
        if(estadoObra.equals(SISOVE.ESTATUS_FISICO[0])){
            return (new Color(255, 255, 0));

            //PARALIZADO si esta culminado o paralizado
        }else if(estadoObra.equals(SISOVE.ESTATUS_FISICO[1]) ||
estadoObra.equals(SISOVE.ESTATUS_FISICO[2])){

            return (new Color(255, 0, 0));

        }

        //obra culminada
    }else if(estado.equals("Si")){
        return (new Color(0, 255, 0));
    }

    //color inicial
    return (new Color(0, 0, 255));

}

/**
 * Determina el estado físico de la obra.
 *
 * @param codigoObra
 *
 * @return
 */
@SuppressWarnings("unchecked")
public String estatusFisicoObra(String codigoObra){

    GestorObra gestorObra = new GestorObra();

    Collection obra = null;

    Collection contratoObra = null;

    final String ESTADO_1 = "SIN CONTRATO";

    final String ESTADO_2 = "SIN AVANCE";

    try {

```

CAPITULO V: Fase de Construcción

```

        contratoObra = new GestorObraContrato().buscarPorCodigoObra(codigoObra);

        if(contratoObra.size() > 0){

            obra = gestorObra.buscarPorCodigoObraEstatusFisico(codigoObra,
SISOVE.ESTATUS_FISICO[0]);
            if(obra.size() > 0){

                return SISOVE.ESTATUS_FISICO[0];
            }
            obra = gestorObra.buscarPorCodigoObraEstatusFisico(codigoObra,
SISOVE.ESTATUS_FISICO[1]);
            if(obra.size() > 0){

                return SISOVE.ESTATUS_FISICO[1];
            }

            obra = gestorObra.buscarPorCodigoObraEstatusFisico(codigoObra,
SISOVE.ESTATUS_FISICO[2]);

            if(obra.size() > 0){

                return SISOVE.ESTATUS_FISICO[2];
            }

            return ESTADO_2;

        }
    } catch (SAVEException e) {
        e.printStackTrace();
    }

    return ESTADO_1;
}

/**
 * Método que inicializa manejadorUIMapa
 * para capturar los eventos de los objetos
 *
 * @return
 */
public ManejadorUIMapa getManejadorUIMapa() {

```

CAPITULO V: Fase de Construcción

```

        if(manejadorUIMapa==null){
            manejadorUIMapa= new ManejadorUIMapa(this,principal,puntos,mapa);
        }
        return manejadorUIMapa;
    }

/**
 * TODO: Clase ManejadorUIMapa
 *
 * Clase ManejadorUIMapa: Esta clase implementa la interfaz
 * ActionListener y MouseMotionListener, la cual, tiene como método a
 * actionPerformed, mouseMoved y mouseDragged encargada de escuchar
 * los eventos generados por los objetos que originan la acción.
 *
 * @author Dima Rodríguez
 *
 * @version 1.0.0
 */

public class ManejadorUIMapa implements MouseListener, MouseMotionListener {

    private PuntoReferencia punt[];

    private IUMapa uiMmapa = null;

    private IUObra obra = null;

    private IUContrato contrato = null;

    Point puntoInicial = null;

/**
 * constructor de la clase
 *
 * @param mapa
 * @param principal
 * @param puntos
 * @param opcionesMapa
 */
    public ManejadorUIMapa(IUMapa mapa,IUVentanaPrincipal principal, ArrayList<PuntoReferencia>
puntos, IUVisualizarMapa opcionesMapa) {

```

CAPITULO V: Fase de Construcción

```

        this.uiMmapa=mapa;

    }

    /**
     * Metodo que captura el evento cuando se oprime y se suelta
     * un boton del raton
     */
    public void mouseClicked(MouseEvent eventoRat) {

        try {

            String    codigo;

            //Valida que el evento de cliqueo sea para crear una obra
            boolean esObra = eventoRat.getClickCount() == 2 &&
            (principal.getUsuario().getTipoUsuario().equals(SISOVE.TIPO_USUARIO[1])||
            principal.getUsuario().getTipoUsuario().equals(SISOVE.TIPO_USUARIO[2]))&&
            mapa.getOpcionMenu()==SISOVE.CREAR_OBRA;

            //Valida que el evento de cliqueo sea para mostrar información del punto
            boolean esMostrar = eventoRat.getClickCount() == 2 &&
            (principal.getUsuario().getTipoUsuario().equals(SISOVE.TIPO_USUARIO[0])||
            principal.getUsuario().getTipoUsuario().equals(SISOVE.TIPO_USUARIO[1])||
            principal.getUsuario().getTipoUsuario().equals(SISOVE.TIPO_USUARIO[2]))&&
            mapa.getOpcionMenu()==SISOVE.VISUALIZAR_MAPA;

            //Valida que el evento de cliqueo sea para crear un contrato no asociado a obra
            boolean esContrato = eventoRat.getClickCount() == 2 &&
            (principal.getUsuario().getTipoUsuario().equals(SISOVE.TIPO_USUARIO[1])||
            principal.getUsuario().getTipoUsuario().equals(SISOVE.TIPO_USUARIO[2]))&&
            mapa.getOpcionMenu()==SISOVE.CREAR_CONTRATO;

            //Determina que sea una obra
            if (esObra){

                do {

                    codigo = (JOptionPane.showInputDialog(null,"Ingrese el Código de la Obra"));
                    if(codigo == null){
                        break;
                    }
                } while (!(new Validar().validarBlanco(codigo)));

                //Se crea un objeto punto donde se le envia como parametro la ubicacion y el nombre

```

CAPITULO V: Fase de Construcción

```

        PuntoReferencia punto = new PuntoReferencia((int) eventoRat.getPoint().getX() - 5, (int)
eventoRat.getPoint().getY() - 5, codigo);
        //se le indica cual va a hacer la identidad del mapa donde se pinta los puntos
        punto.setIdImagenMapa(idMapa);
        //carga el tipo de obra
        punto.setTipoObra("Obra");
        //carga el color
        punto.setColorPunto(new Color(0, 0, 255));
        //determina la visibilidad
        punto.setVisible(true);
        //valida que se haya ingresado el codigo
        if(codigo != null){
            //Crea un objeto UIObra
            crearObra(codigo,punto.getPuntos(),uiMmapa);
            //pinta los puntos en la imagen del mapa
            getLienzo().setPuntos(puntos);
        }

        //determina que de van a mostrar la información de los puntos
    }else if(esMostrar){

        //determina donde esta ubicado el punto
        if(ubicarPunto(eventoRat)!=null){

            //muestra la información del punto
            informacionObra(eventoRat);

        }else{
            JOptionPane.showMessageDialog(null, "No hay Obra Asociada a esta Ubicación");
        }
    }
    //determina que sea un contrato
    } else if(esContrato){

        //valida la entrada del codigo de contrato
        do {
            codigo = (JOptionPane.showInputDialog(null,"Ingrese el Código del Contrato"));
            if(codigo == null){
                break;
            }
        } while (!(new Validar().validarBlanco(codigo)));

        //Se crea un objeto punto donde se le envia como parametro la ubicacion y el nombre

```

CAPITULO V: Fase de Construcción

```

        PuntoReferencia punto = new PuntoReferencia((int) eventoRat.getPoint().getX() - 5, (int)
eventoRat.getPoint().getY() - 5, codigo);
        //se le indica cual va a hacer la identidad del mapa donde se pinta los puntos
        punto.setIdImagenMapa(idMapa);

        punto.setTipoObra("Contrato");

        punto.setColorPunto(new Color(153, 204, 255));

        punto.setVisible(true);

        if(codigo != null){
            //Crea un objeto UIObra
            crearContrato(codigo,punto.getPuntos(),uiMmapa);
            //pinta los puntos en la imagen
            getLienzo().setPuntos(puntos);
        }
    }

    } catch (Exception e) {
        e.printStackTrace();
    }
    //Utilizado para que no muestre a cada rato el tooltip
    getLienzo().setToolTipText("");

}

/**
 * Metodo que captura el evento cuando el cursor del raton entra
 * en los limites de un componente
 */
public void mouseEntered(MouseEvent me) {

}

/**
 * Metodo que captura el evento cuando el cursor del raton sale
 * en los limites de un componente
 */
public void mouseExited(MouseEvent me) {

}

```

CAPITULO V: Fase de Construcción

```

/**
 *Metodo que captura el evento cuando se mantiene presionado el boton
 *del raton
 */
public void mousePressed(MouseEvent eventoRat) {

    int contador = 0;

    //valida que se quiera modificar el punto
    if(mapa.getOpcionMenu()==SISOVE.MODIFICAR_PUNTO){

        //recorre la lista de puntos
        for (int i = 0; i < puntos.size(); i++){
            //determina que el puntero se encuentre sobre el punto
            if (puntos.get(i).ecuacionDeCirculo(eventoRat.getPoint())){

                //Captura el punto inicial del punto por si el usuario no desea moverlo
                puntoInicial = puntos.get(i).getUbicacionPunto();
                //pinta el punto
                puntos.get(i).pintarPunto(getLienzo().getGraphics(), Color.RED);

                getPunt()[contador] = puntos.get(i);

            }
        }

        contador=0;
        //repintar la imagen y los puntos
        getLienzo().repaint();
    }

}

/**
 *Metodo que captura el evento cuando se suelta el boton del raton
 *despues de ser oprimido
 */
public void mouseReleased(MouseEvent eventoRat) {

    //valida que se halla movido un punto
    if(getPunt()[0]!=null){
        getPunt()[0].pintarPunto(getLienzo().getGraphics());
    }
}

```

CAPITULO V: Fase de Construcción

```

    }

    getLienzo().repaint();

    //valida que se quiera modificar el punto
    if(mapa.getOpcionMenu()==SISOVE.MODIFICAR_PUNTO){

        //Determina que se haya hubicado el punto
        if(ubicarPunto(eventoRat)!=null){

            //confirma que se desea mover el punto
            int confirmar = JOptionPane.showConfirmDialog(null, "¿Acepta la nueva referencia
del punto?", "Confirmar", JOptionPane.YES_NO_OPTION);

            //confirma el movimiento
            if(confirmar == 0){
                moverPunto(ubicarPunto(eventoRat),eventoRat);

            }else{
                //Devuelve el punto a su ubicación original
                if(getPunt()[0]!=null){

                    getPunt()[0].setUbicacionPunto(puntoInicial);
                    getPunt()[0].pintarPunto(getLienzo().getGraphics());
                }
                getLienzo().repaint();
            }
            //no selecciono algún punto
        }else{
            JOptionPane.showMessageDialog(null, "Debe Seleccionar un Punto");
        }

    }

    getPunt()[0]=null;
    getPunt()[1]=null;
}

/**
 *Metodo que captura el evento cuando el puntero es movido mientras el
 *boton del raton esta oprimido
 */
public void mouseDragged(MouseEvent eventoRat) {

```

CAPITULO V: Fase de Construcción

```

if(getPunt()[0]!=null){

    //repinta el punto mientras es arastrado
    Point ubicacion=new Point(eventoRat.getX()-5,eventoRat.getY()-5);
    getPunt()[0].setUbicacionPunto(ubicacion);
    getLienzo().repaint();
}

}

/**
 *Metodo que captura el evento cuando el puntero es movido
 */
public void mouseMoved(MouseEvent mm) {

    //valida que el cursor se encuentre en el área de la ventana
    if(isSelected()){
        //cambia el cursos a uno especial
        getLienzo().setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));

    }else{

        //cambia al cursos normal
        getLienzo().setCursor(Cursor.getPredefinedCursor(Cursor.DEFAULT_CURSOR));
    }

    //recorre la lista de puntos
    for (int i = 0; i < puntos.size()&&isSelected(); i++){

        //identifica el punto en la lista
        if (puntos.get(i).ecuacionDeCirculo(mm.getPoint())){

            //cambia el color al estar el puntero encima del punto
            puntos.get(i).pintarPunto(getLienzo().getGraphics(), Color.magenta);
            //muestra su contenido o información del punto
            getLienzo().setToolTipText(puntos.get(i).getFinalidad());
            getLienzo().getToolTipText();

        }else{
            puntos.get(i).pintarPunto(getLienzo().getGraphics());
        }
    }
}

```

CAPITULO V: Fase de Construcción

```

    }

    /**
     *
     * @return
     */
    public ArrayList<PuntoReferencia> getPuntos() {
        return puntos;
    }

    /**
     *
     * @return
     */
    public PuntoReferencia[] getPunt() {

        if(punt==null){
            punt = new PuntoReferencia[2];
punt[0]=null;
punt[1]=null;
        }
        return punt;
    }

    public PuntoReferencia ubicarPunto(MouseEvent eventoRat){

        for (int i = 0; i < puntos.size()&&isSelected(); i++){

            if (puntos.get(i).ecuacionDeCirculo(eventoRat.getPoint())){

                return puntos.get(i);
            }
        }
        return null;
    }

    /**
     * Metodo para mostrar la ventana a la cual se le va
     * a ingresar todos los datos de la obra.
     * @param mapa
     * @param codigo: Código que tendrá la obra
     * @param punto: Ubicacion del punto en el mapa

```

CAPITULO V: Fase de Construcción

```

    * @param agregarPunto: arreglo donde se guardan los puntos
    * creados
    * @return
    */
    public void crearObra(String codigo, PuntoReferencia punto, ArrayList<PuntoReferencia> agregarPunto,
IUMapa mapa) {

        obra= new IUObra(codigo,punto,agregarPunto,mapa,SISOVE.CREAR);
        //Se refiere al titulo de la Ventana
        obra.setTitle("Agregar Obra");
        //Agregar la nueva ventana hija a la ventana padre
        new InterfazVentanaPadre(principal.getDpVentanaPadre()).agregarAVentanaPadre(obra);
    }

    private void crearContrato(String codigo, PuntoReferencia punto, ArrayList<PuntoReferencia>
agregarPunto, IUMapa mapa) {

        contrato = new IUContrato(codigo,punto,agregarPunto,mapa, principal, SISOVE.CREAR);
        contrato.setTitle("Agregar Contrato No asociado a Obra");
        new InterfazVentanaPadre(principal.getDpVentanaPadre()).agregarAVentanaPadre(contrato);
    }

    //metodo utilizado para modificar de ubicación el punto
    public void moverPunto(PuntoReferencia punto, MouseEvent eventoRat){

        Punto puntoAux = new Punto();
        GestorPunto gestorPunto = new GestorPunto();
        try {
            //se buscan los datos asociados a la obra y el punto
            puntoAux=(Punto)gestorPunto.buscarPorClaveprimaria(punto.getCodRefObra());
            //se insertan los nuevos datos
            puntoAux.setUbicacionPunto(new Point((int) eventoRat.getPoint().getX() - 5, (int)
eventoRat.getPoint().getY() - 5));

            puntoAux.setPuntoReferencia(new
PGpoint(puntoAux.getUbicacionPunto().x,puntoAux.getUbicacionPunto().y));
            //se actualizan los datos en la base de datos
            gestorPunto.actualizar(punto.getCodRefObra(), puntoAux);
        } catch (SQLException e) {

            e.printStackTrace();
        }
    }
}

```

CAPITULO V: Fase de Construcción

```

private void informacionObra(MouseEvent eventoRat) {

    GestorObra gestorObra = new GestorObra();
    GestorContrato gestorContrato = new GestorContrato();

    Contrato contrato = new Contrato();
    Obra obra = new Obra();

    @SuppressWarnings("unused")
    Formatos formato = new Formatos();
    @SuppressWarnings("unused")
    VDIforPuntos dialogoInforPuntos = null;

    String[] otrosDatos = new String[2];

    String informacion = "";

    try {
        obra = (Obra)gestorObra.buscarPorClaveprimaria(ubicarPunto(eventoRat).getCodRefObra());
        contrato = gestorContrato.buscarPorClaveprimaria(ubicarPunto(eventoRat).getCodRefObra());
    } catch (SAVException e) {
        e.printStackTrace();
    }

    if(obra!=null){
        informacion = "-CODIGO: " + obra.getCodigoObra()+
            "\n.-DENOMINACIÓN: " + obra.getDenominacionObra()+
            "\n.-MONTO TOTAL: " +
            formato.formatearMoneda((obra.getMontoTotalObra()))+
            "\n.-MONTO EJECUTADO: " + (montoEjecutado(obra.getCodigoObra())) + " " +
            obra.getTipoMonedaObra() +
            "\n.-(%) EJECUTADO: " + porcentajeAvanceObra(obra.getCodigoObra());

        dialogoInforPuntos = new VDIforPuntos(null, true, SISOVE.OBRA,
            informacion,obra.getCodigoObra());
    }
}

```

CAPITULO V: Fase de Construcción

```

        }else if(contrato!=null){

            otrosDatos = monto_PorcentajeContrato(contrato.getCodigoContrato());
            informacion = ".-CODIGO: " + contrato.getCodigoContrato()+
                "\n.-OBJETO: " + contrato.getObjContrato()+
                "\n.-Monto Total: " + formato.formatearMoneda((contrato.getMontoContrato()))+
                "\n.-MONTO EJECUTADO: " + otrosDatos[0] + " " +
                contrato.getTipoMonedaContrato() +
                "\n.-(%) EJECUTADO: " + otrosDatos[1];
            dialogoInforPuntos = new VDIforPuntos(null, true, SISOVE.CONTRATO,
informacion, contrato.getCodigoContrato());
        }
    }

/**
 * calcula el monto ejecutado de la obra
 * @param codigoObra
 * @return
 */
@SuppressWarnings("unchecked")
public String montoEjecutado(String codigoObra){

    Formatos formatos = new Formatos();

    GestorAvanceContrato gestorAvance = new GestorAvanceContrato();

    double subtotal= 0;

    ArrayList<AvanceContrato> mtoEjecutado = new ArrayList<AvanceContrato>();

    try {

        mtoEjecutado = (ArrayList<AvanceContrato>)
gestorAvance.buscarPorcentajeObra(codigoObra);
    } catch (SAVEException e) {
        e.printStackTrace();
    }
}

```

CAPITULO V: Fase de Construcción

```

        for (Iterator iter = mtoEjecutado.iterator(); iter.hasNext();) {
            AvanceContrato avance = (AvanceContrato) iter.next();

            subtotal += avance.getMontoEjecutado();

        }

        return formatos.formatearMoneda(subtotal);
    }

    /**
     * Se obtiene los avance para determinar el estado de la obra
     * @param codigoObra
     *
     * @return AvanceObra
     */
    public MontoEstimado obtenerAvanceObra(String codigoObra){

        GestorMontoEstimado gestorMontoEstimado = new GestorMontoEstimado();
        MontoEstimado montoEstimado = new MontoEstimado();

        String[] key = new String[2];

        key[0] = codigoObra;
        key[1] = codigoObra;

        try {
            montoEstimado = (MontoEstimado) gestorMontoEstimado.buscarPorOtro(key, 1);
        } catch (SAVException e) {
            e.printStackTrace();
            return null;
        }

        return montoEstimado;
    }

    /**
     *
     * @param codigoObra
     * @return
     */
    @SuppressWarnings("unchecked")

```

CAPITULO V: Fase de Construcción

```

private String porcentajeAvanceObra(String codigoObra){

    ArrayList<AvanceContrato> mtoEjecutado = new ArrayList<AvanceContrato>();
    MontoEstimado montoEstimado = new MontoEstimado();
    String[] key = new String[2];

    double subtotal= 0;
    double porcentaje =0;

    key[0] = codigoObra;
    key[1] = codigoObra;

    try {
        montoEstimado = (MontoEstimado) new GestorMontoEstimado().buscarPorOtro(key, 1);
        mtoEjecutado = (ArrayList<AvanceContrato>) new
GestorAvanceContrato().buscarPorcentajeObra(codigoObra);

    } catch (SAVException e) {
        e.printStackTrace();
    }

    for (Iterator iter = mtoEjecutado.iterator(); iter.hasNext(); ) {
        AvanceContrato avance = (AvanceContrato) iter.next();

        subtotal += avance.getMontoEjecutado();

    }

    if(montoEstimado.getMonto() > 0){
        porcentaje = subtotal/montoEstimado.getMonto();
    }

    return new Formatos().formatearPorcentaje(porcentaje);
}

/**
 * calcula el porcentaje y monto del contrato.
 *
 * @param codigoContrato
 *
 * @return String[]

```

CAPITULO V: Fase de Construcción

```

*/
public String[] monto_PorcentajeContrato(String codigoContrato){

    GestorAvanceContrato gestorAvance = new GestorAvanceContrato();
    AvanceContrato avance = new AvanceContrato();
    Formatos formatos = new Formatos();

    String[] datos = new String[2];

    try {
        avance = (AvanceContrato) gestorAvance.buscarPorOtro(codigoContrato, 1);
    } catch (SAVException e) {
        e.printStackTrace();
    }

    if(avance!=null){
        datos[0] = formatos.formatearMoneda(avance.getMontoEjecutado());
        datos[1] = formatos.formatearPorcentaje(avance.getPorcAvanceFisico());
    }else{
        datos[0] = "0,00";
        datos[1] = "0,00%";
    }

    return datos;
}
}

```

ImagenMapa.java

```

import java.awt.Dimension;
import java.awt.Image;
import java.awt.MediaTracker;

import javax.swing.ImageIcon;
import javax.swing.JComponent;
import javax.swing.JOptionPane;

import modelo.Mapa;
import controlador.GestorMapa;

```

CAPITULO V: Fase de Construcción

```
import controlador.SAVEException;

/**
 *
 * Clase utilizada para cargar la
 * imagen del mapa en un objeto tipo
 * JComponet, que luego será cargado
 * en una ventana interna.
 *
 * @author Dima Rodríguez
 *
 * @version 1.0.0
 */

public class ImagenMapa {

    /**
     * Imagen del mapa
     */
    private Image imagen;

    /**
     * Tamaño de la imagen del mapa
     */
    private Dimension tamano;

    /**
     * Constructor que recibe como parámetro idMapa
     * y lienzo. Utilizando el primer parámetro para obtener
     * la imagen de la base de datos por medio del
     * método recuperarImagen y es cargado al objeto
     * lienzo mediante el objeto rastreo instanciado
     * de la clase MediaTracker.
     *
     * @param idMapa: identidad de la imagen del mapa
     * en la base de datos
     * @param lienzo: objeto de tipo JComponet donde
     * se cargara la imagen
     * @exception InterruptedException:
     */

    public ImagenMapa(String idMapa,JComponent lienzo) {
```

CAPITULO V: Fase de Construcción

```

        this.imagen = recuperarImagen(idMapa);

        if(this.imagen!=null){
            MediaTracker rastreo = new MediaTracker(lienzo);
            rastreo.addImage(getImagen(), 0);
            try {
                rastreo.waitForAll();
            } catch (InterruptedException e) {

                e.printStackTrace();
            }
            tamaño = new Dimension(getImagen().getWidth(null),getImagen().getHeight(null));
        }
    }

    /**
     * utilizado para recuperar la imagen de la
     * base de datos y transformarla a un tipo de
     * imagen utilizable por el programa.
     * recibe como parametro idMapa
     *
     * @param idMapa: identidad de la imagen del mapa
     * en la base de datos
     * @exception SAVException
     * @return Image
     */
    private Image recuperarImagen(String idMapa){

        Mapa imagenRecuperada = new Mapa();

        try {
            imagenRecuperada = (Mapa) new GestorMapa().buscarPorClaveprimaria(idMapa);
        } catch (SAVException e) {

            e.printStackTrace();
        }
    }

```

CAPITULO V: Fase de Construcción

```

        if(imagenRecuperada!=null){

            return new ImageIcon(imagenRecuperada.getRetornoImagenMapa()).getImage();

        }else{
            JOptionPane.showMessageDialog(null, "Error al cargar la imagen de la Base de Datos",
"Error", JOptionPane.ERROR_MESSAGE);
            return null;
        }
    }

    /**
     * se hace uso de este para obtener la
     * imagen fuera de su ámbito interno,
     * o sea pueda ser utilizado en otras
     * clases.
     *
     * @return Image
     */
    public Image getImagen() {
        return imagen;
    }

    /**
     *se hace uso de este para obtener
     *el tamaño de la imagen fuera de
     *su ámbito interno, o sea pueda
     *ser utilizado en otras clases.
     *
     * @return Dimension
     */
    public Dimension getTamano() {
        return tamano;
    }
}

```

Lienzo.java

```
import java.awt.Dimension;
```

CAPITULO V: Fase de Construcción

```
import java.awt.Graphics;
import java.util.ArrayList;

import javax.swing.JComponent;

/**
 * Clase que extiende de JComponent utilizada
 * para el manejo de imágenes, en este caso,
 * la carga de la imagen de los mapas y los
 * puntos que referencia a las obras.
 *
 * @author Dima Rodríguez
 *
 * @version 1.0.0
 */
@SuppressWarnings("serial")
public class Lienzo extends JComponent {

    /**
     * Arreglo de puntos ubicados en los mapas
     */
    private ArrayList<PuntoReferencia> puntos;

    /**
     * Imagen del mapa a mostrar
     */
    private ImagenMapa imagenMapa;

    /**
     * Identidad única del mapa a mostrar o buscar
     */
    private String idMapa;

    /**
     * Constructor donde se inicializan
     * los atributos.
     *
     * @param idMapa: código de la imagen.
     */

    public Lienzo(String idMapa) {
```

CAPITULO V: Fase de Construcción

```
this.puntos = new ArrayList<PuntoReferencia>();  
this.imagenMapa = null;  
this.idMapa = idMapa;  
  
}  
  
/**  
 *  
 * atributo de la clase JComponent que  
 * recibe como parámetro un objeto de  
 * tipo Graphics, utilizado para cargar  
 * la imagen de los mapas y los puntos.  
 *  
 * @param g: objeto utilizado para para  
 * cargar la imagen de los mapas y los  
 * puntos.  
 *  
 * @exception OutOfMemoryError  
 */  
  
public void paintComponent(Graphics g){  
  
    try{  
  
        g.drawImage(getImagenMapa().getImagen(), 0, 0, this);  
  
        for(int i = 0; i < puntos.size(); i++){  
  
            final PuntoReferencia punto = (PuntoReferencia) puntos.get(i);  
            punto.pintarPunto(g);  
  
        }  
  
    } catch(OutOfMemoryError e){  
        e.printStackTrace();  
    }  
  
}  
  
/**
```

CAPITULO V: Fase de Construcción

```

*
*por medio de este podemos obtener los
*datos de los puntos que han sido
*guardados en la base de datos.
*
* @param puntos: arreglo de puntos
* obtenidos tanto de la base de datos
* como de la creacion de los mismos.
*/
public void setPuntos(final ArrayList<PuntoReferencia> puntos){
    this.puntos = puntos;
}

/**
 *retorna el tamaño de la imagen.
 *
 * @return Dimension
 */

public Dimension getPreferredSize(){

    return getImagenMapa().getTamano();

}

/**
 *
 *se inicializa el atributo imagenMapa
 *y el método es es utilizado internamente.
 *
 * @return ImagenMapa
 */

public ImagenMapa getImagenMapa(){
    if(imagenMapa==null){
        imagenMapa = new ImagenMapa(this.idMapa,this);
    }
    return imagenMapa;
}

```

CAPITULO V: Fase de Construcción

```
        public void setImagenMapa(ImagenMapa imagenMapa) {  
            this.imagenMapa = imagenMapa;  
        }  
    }  
}
```

Mapa.java

```
import java.io.File;  
import java.io.FileInputStream;  
  
/**  
 * Clase Imagen la cual contiene los atributos  
 * y los metodos de las imagenes de los mapas  
 *  
 * @author Dima Rodríguez  
 *  
 * @version 1.0.0  
 */  
public class Mapa {  
  
    /**  
     * Identificacion unica de la imagen  
     */  
    private String idImagenMapa;  
    /**  
     * recuperar informacion de la imagen  
     */  
    private File archivoImagenMapa;  
    /**  
     * Entrada en byte del archivo de la imagen  
     */  
    private FileInputStream imagenMapa;  
    /**  
     * retorno de la imagen en bytes  
     */  
    private byte[] retornoImagenMapa;  
  
    /**  
     * constructor por defecto
```

CAPITULO V: Fase de Construcción

```
*
*/
public Mapa() {

    idImagenMapa="";
    archivoImagenMapa=null;
    imagenMapa=null;
    retornoImagenMapa=null;

}

/**
 * devolver el archivo imagen mapa
 *
 * @return File
 */
public File getArchivoImagenMapa() {
    return archivoImagenMapa;
}

/**
 * establecer el atributo archivo
 *
 * @param archivo
 */
public void setArchivoImagenMapa(File archivo) {
    this.archivoImagenMapa = archivo;
}

/**
 * devolver la identidad de la imagen
 *
 * @return String
 */
public String getIdImagenMapa() {
    return idImagenMapa;
}

/**
 * establecer la identidad de la imagen.
 *
 * @param idImagenMapa
 */
```

CAPITULO V: Fase de Construcción

```
public void setIdImagenMapa(String idimagen) {  
    this.idImagenMapa = idimagen;  
}  
  
/**  
 * devolver la imagen  
 *  
 * @return FileInputStream  
 */  
public FileInputStream getImagenMapa() {  
    return imagenMapa;  
}  
  
/**  
 * establecer la imagen  
 *  
 * @param imagen  
 */  
public void setImagenMapa(FileInputStream imagen) {  
    this.imagenMapa = imagen;  
}  
  
/**  
 * devolver la imagen en  
 * un arreglo bytes  
 *  
 * @return byte[]  
 */  
public byte[] getRetornoImagenMapa() {  
    return retornoImagenMapa;  
}  
  
/**  
 * establecer el arreglo de bytes de la imagen  
 *  
 * @param retornoImagen  
 */  
public void setRetornoImagenMapa(byte[] retornoImagen) {  
    this.retornoImagenMapa = retornoImagen;  
}  
  
}
```

GestorMapa.java

```

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Collection;

import modelo.Mapa;

/**
 * Clase que extiende de la clase BaseDatoAbstracto
 * donde hereda sus métodos y atributos utilizados
 * para la conexión a la base de dato. Aquí tendrá
 * acceso a la tabla Imagen para insertar, actualizar,
 * eliminar o buscar datos.
 *
 * @author Dima Rodríguez
 *
 * @version 1.0.0
 */
public class GestorMapa extends BaseDatoAbstracto {

    public GestorMapa() {
        super();

        this.instruccion_insertar = "INSERT INTO mapa (idimagenmapa, imagenmapa) VALUES (?,?)";
        this.instruccion_actualizar = "UPDATE mapa SET imagenmapa = ? WHERE idimagenmapa = ?";
        this.instruccion_Eliminar = "DELETE FROM mapa WHERE idimagenmapa = ?";
        this.instruccion_cargar_clave_Prim = "SELECT idimagenmapa, imagenmapa FROM mapa WHERE
idimagenmapa = ?";
    }

    /**
     * Es hereda de la clase abstracta
     */
    protected Object cargar(ResultSet rs) throws SQLException {

        Mapa mapa = new Mapa();

        if(rs.next()){

```

CAPITULO V: Fase de Construcción

```

        mapa.setRetornoImagenMapa(rs.getBytes("imagenmapa"));

        return mapa;
    }

    return null;
}

/**
 * Es heredao de la clase abstracta
 */
protected void setPreparedStatementDelete(PreparedStatement stm, Object clave) throws SQLException {

    if (!(clave instanceof String)){
        return;
    }

    stm.setString(1,(String)clave);

}

/**
 * Es heredao de la clase abstracta
 */
@SuppressWarnings("unchecked")
protected void setPreparedStatementFinder(int tipoBuaqueda, PreparedStatement stm, Collection sets) throws
SQLException {

}

/**
 * Es heredao de la clase abstracta
 */
protected void setPreparedStatementInsert(PreparedStatement stm, Object obj) throws SQLException {

    if (!(obj instanceof Mapa)){
        return;
    }

    Mapa imagen=(Mapa) obj;

    stm.setString(1,imagen.getIdImagenMapa());
}

```

CAPITULO V: Fase de Construcción

```

        stm.setBinaryStream(2, imagen.getImagenMapa(), (int)imagen.getArchivoImagenMapa().length());
    }

    /**
     * Es heredao de la clase abstracta
     */
    protected void setPreparedStatementLoadPK(PreparedStatement stm, Object clave) throws SQLException {

        if (!(clave instanceof String)){
            return;
        }

        stm.setString (1, (String) clave);

    }

    /**
     * Es heredao de la clase abstracta
     */
    protected void setPreparedStatementUpdate(PreparedStatement stm, Object clave, Object obj) throws
    SQLException {

        if (!(obj instanceof Mapa)){
            return;
        }
        if (!(clave instanceof String)){
            return;
        }

        Mapa imagen =(Mapa) obj;

        stm.setBinaryStream(1, imagen.getImagenMapa(), (int)imagen.getArchivoImagenMapa().length());
        stm.setString(2,(String) clave);

    }

}

```

PuntoReferencia.java

```

import java.awt.Color;

```

CAPITULO V: Fase de Construcción

```
import java.awt.Graphics;
import java.awt.Point;

import modelo.Punto;

import org.postgresql.geometric.PGpoint;

/**
 *Clase utilizada para la creación de los
 *puntos que hacen referencia a las obras.
 *
 * @author Dima Rodríguez
 *
 * @version 1.0.0
 */

public class PuntoReferencia extends Punto{

    /**
     * Constante que indica el radio que debe tener el punto
     */
    private static final int RADIO = 10;

    /**
     * Constante que indica el Tamaño del punto
     */
    private final int TAMANO = 10;

    /**
     * constructor utilizado para crear objetos
     * de tipo punto sin inicializar los parámetros.
     *
     */
    public PuntoReferencia(){

    }

    /**
     * utilizado para cargar los puntos con datos de la base de datos
     *
     * @param x: ubicación en la posición x
     * @param y: ubicación en la posición y
     */
}
```

CAPITULO V: Fase de Construcción

```

* @param nombre: código de referencia
* @param finalidad: finalidad del contrato u Obra
*/

public PuntoReferencia(int x, int y, String nombre, String finalidadObra){

    puntoReferencia = new PGpoint (x,y);
    ubicacionPunto = new Point((int)puntoReferencia.x,(int)puntoReferencia.y);
    codRefObra = nombre;
    visible = true;
    finalidad = finalidadObra;

}

/**
 * utilizable para crear los puntos
 * de las obras o contratos.
 *
 * @param x: ubicación en la posición x
 * @param y: ubicación en la posición y
 * @param nombre: código de referencia
 */

public PuntoReferencia(int x, int y, String nombre){

    puntoReferencia = new PGpoint (x,y);
    ubicacionPunto = new Point((int)puntoReferencia.x,(int)puntoReferencia.y);
    codRefObra = nombre;
    visible = true;
}

/**
 * método para pintar el punto a la
 * imagen del mapa y recibe como para
 * metro un objeto Graphics.
 *
 * @param g: objeto Graphics útil a
 * la hora de pintar los puntos
 */

public void pintarPunto(Graphics g){
    g.setColor(Color.BLACK);
    g.drawOval(ubicacionPunto.x, ubicacionPunto.y, TAMANO, TAMANO);
}

```

CAPITULO V: Fase de Construcción

```

        g.setColor(colorPunto);
        g.fillOval(ubicacionPunto.x, ubicacionPunto.y, TAMANO, TAMANO);
    }

    /**
     * método sobrecargado que se utiliza a la
     * hora de interactuar con estos, cambiando
     * el color del mismo.
     *
     * @param g: objeto Graphics útil a
     * la hora de pintar los puntos
     * @param color: color que se le da al punto
     */

    public void pintarPunto(Graphics g, Color color){
        g.setColor(Color.BLACK);
        g.drawOval(ubicacionPunto.x, ubicacionPunto.y, TAMANO, TAMANO);
        g.setColor(color);
        g.fillOval(ubicacionPunto.x, ubicacionPunto.y, TAMANO, TAMANO);
    }

    /**
     * utilizado para determinar la ecuación
     * de la circunferencia con que se dibuja
     * el círculo.
     *
     * @param punto: datos de la coordenada de la
     * ubicación de los puntos
     * @return boolean
     */

    public boolean ecuacionDeCirculo(Point punto){

        return (((punto.x - ubicacionPunto.x) * (punto.x - ubicacionPunto.x) + (punto.y - ubicacionPunto.y) * (punto.y -
        ubicacionPunto.y)) <= (RADIO * RADIO));
    }
}

```

Punto.java

```
import java.awt.Color;
```

CAPITULO V: Fase de Construcción

```
import java.awt.Point;

import org.postgresql.geometric.PGpoint;

/**
 * Clase utilizada para crear objetos de tipo punto
 * contiene información del punto que hace referencia
 * a la obra.
 *
 * @author Dima Rodríguez
 *
 * @version 1.0.0
 */
public class Punto {

    /**
     * Captura la referencia del punto en la imagen
     * pero de tipo Punto para la base de dato Postgres
     */
    protected PGpoint puntoReferencia;

    /**
     * Código de referencia de la obra.
     */
    protected String codRefObra;

    /**
     * Identificación de la imagen del mapa
     */
    protected String idImagenMapa;

    /**
     * Captura la referencia del punto en la imagen
     * para el sistema de tipo point (awt.point)
     */
    protected Point ubicacionPunto;

    /**
     * Tipo Obra o Contrato No asociado
     */
    protected String tipoObra;

    /**
```

CAPITULO V: Fase de Construcción

```
* Visibilidad del punto en el sistema.
*/
protected boolean visible;

/**
 * Finalidad de la Obra
 */
protected String finalidad;

/**
 * Color del punto dependiendo del estado de la obra.
 */
protected Color colorPunto;

/**
 * Constructor por defecto encargado de inicializar el objeto.
 */

public Punto() {

    puntoReferencia = null;
    codRefObra = "";
    idImagenMapa = "";
    ubicacionPunto = null;
    tipoObra = "";
    visible = true;
    finalidad = "";
    colorPunto = null;
}

/**
 * devolver el código de la obra o contrato.
 *
 * @return codRefObra: código de la obra o contrato.
 */
public String getCodRefObra() {
    return codRefObra;
}

/**
 * establecer el código de la obra o contrato.
 *
```

CAPITULO V: Fase de Construcción

```
* @param codigo: código de la obra o contrato.
*/
public void setCodRefObra(String codRefObra) {
    this.codRefObra = codRefObra;
}

/**
 * devolver el color de los puntos
 * en otras clases se utiliza este método.
 *
 * @return colorPunto: de tipo Color
 */
public Color getColorPunto() {
    return colorPunto;
}

/**
 * establecer el color de los puntos
 * en otras clases se utiliza este método.
 *
 * @param colorPunto: color del punto
 */
public void setColorPunto(Color colorPunto) {
    this.colorPunto = colorPunto;
}

/**
 * devolver la finalidad u objeto de la obra o contrato.
 *
 * @return finalidad: String
 */
public String getFinalidad() {
    return finalidad;
}

/**
 * establecer la finalidad u objeto de la obra o contrato.
 *
 * @param finalidad: finalidad u objeto de la obra o contrato.
 */
public void setFinalidad(String finalidad) {
    this.finalidad = finalidad;
}
```

CAPITULO V: Fase de Construcción

```
/**
 * devolver código de la imagen del mapa.
 *
 * @return idImagenMapa: String
 */
public String getIdImagenMapa() {
    return idImagenMapa;
}

/**
 * establecer código de la imagen del mapa.
 *
 * @param idImagenMapa: código de la imagen del mapa
 */
public void setIdImagenMapa(String idImagenMapa) {
    this.idImagenMapa = idImagenMapa;
}

/**
 * devolver posición x, y, para la base
 * de dato postgres.
 *
 * @return puntoReferencia: de tipo PGpoint
 */
public PGpoint getPuntoReferencia() {
    return puntoReferencia;
}

/**
 * establecer posición x, y.
 *
 * @param puntoReferencia: posición x, y de los puntos
 */
public void setPuntoReferencia(PGpoint puntoReferencia) {
    this.puntoReferencia = puntoReferencia;
}

/**
 * devolver el tipo de obra que referencia la obra.
 *
 * @return tipoObra: String
 */
```

CAPITULO V: Fase de Construcción

```

public String getTipoObra() {
    return tipoObra;
}

/**
 * establecer el tipo de obra que referencia la obra.
 *
 * @param tipoObra: tipo de obra que referencia la obra.
 */
public void setTipoObra(String tipoObra) {
    this.tipoObra = tipoObra;
}

/**
 * devolver la ubicación de los
 * puntos en otras clases se utiliza
 * este método.
 *
 * @return ubicacionPunto: de tipo Point
 */
public Point getUbicacionPunto() {
    return ubicacionPunto;
}

/**
 * establecer la ubicación de los
 * puntos en otras clases se utiliza
 * este método.
 *
 * @param ubicacionMapa: posicion x, y
 */
public void setUbicacionPunto(Point ubicacionMapa) {
    this.ubicacionPunto = ubicacionMapa;
}

/**
 * devolver el estatus de visibilidad de los puntos
 *
 * @return visibilidad: estatus de visibilidad de los puntos
 */
public boolean isVisible() {
    return visible;
}

```

CAPITULO V: Fase de Construcción

```

/**
 * establecer el estatus de visibilidad de los puntos.
 *
 * @param visibilidad: estatus de visibilidad
 */
public void setVisible(boolean visibilidad) {
    this.visible = visibilidad;
}
}

```

GestorPunto.java

```

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.Collection;

import modelo.Punto;

import org.postgresql.geometric.PGpoint;

/**
 * Clase que extiende de la clase
 * BaseDatoAbstracto donde hereda
 * sus métodos y atributos utilizados
 * para la conexión a la base de dato.
 * Aquí tendrá acceso a la tabla punto
 * para insertar, actualizar, eliminar
 * o buscar datos.
 *
 * @author Dima Rodríguez
 *
 * @version 1.0.0
 */
public class GestorPunto extends BaseDatoAbstracto {

    private final int BUSCAR_ID_MAPA = 1;

```

CAPITULO V: Fase de Construcción

```

private final String CONSULTA_ID_MAPA = "SELECT idimagenmapa, codrefobra, puntoref, tipoobra, finalidad,
visible FROM punto WHERE idimagenmapa = ? AND visible = true";

```

```

public GestorPunto() {
    super();

    this.instruccion_insertar = "INSERT INTO punto (idimagenmapa, codrefobra, puntoref, tipoobra,
finalidad, visible) VALUES (?, ?, ?, ?, ?)";
    this.instruccion_actualizar = "UPDATE punto SET puntoref = ? , tipoobra = ?, finalidad = ?, visible = ?
WHERE codrefobra = ?";
    this.instruccion_Eliminar = "DELETE FROM punto WHERE codrefobra = ?";
    this.instruccion_cargar_clave_Prim = "SELECT idimagenmapa, codrefobra, puntoref, tipoobra, finalidad,
visible FROM punto WHERE codrefobra = ? AND visible = true";
    //Recuperar punto no visible
    this.instruccion_cargar_clave_secundaria = "SELECT idimagenmapa, codrefobra, puntoref, tipoobra,
finalidad, visible FROM punto WHERE codrefobra = ? AND visible = false";
}

```

```

/**
 * recuperar los puntos asociados a una identidad de una imagen de mapa.
 *
 * @param identMapa
 * @return Collection
 * @throws SAVException
 */
@SuppressWarnings("unchecked")
public Collection buscarPorIdentidadMapa(String identMapa) throws SAVException{
    ArrayList<String> parametrosDeQueryList = new ArrayList<String>();
    Collection resultadoBusqueda;
    parametrosDeQueryList.add(identMapa);
    resultadoBusqueda
this.getBuscar(BUSCAR_ID_MAPA,CONSULTA_ID_MAPA,parametrosDeQueryList);
    return resultadoBusqueda;
}

```

```

/**
 * Es heredao de la clase abstracta
 */
protected Object cargar(ResultSet rs) throws SQLException {

    Punto punto = new Punto();

```

CAPITULO V: Fase de Construcción

```

        if(rs.next()){

            punto.setCodRefObra(rs.getString("codrefobra"));
            punto.setIdImagenMapa(rs.getString("idimagenmapa"));
            punto.setPuntoReferencia(((PGpoint) rs.getObject("puntoref")));
            punto.setTipoObra(rs.getString("tipoobra"));
            punto.setFinalidad(rs.getString("finalidad"));
            punto.setVisible(rs.getBoolean("visible"));

            return punto;
        }else{
            return null;
        }
    }

    /**
     * Es herdao de la clase abstracta
     */
    protected void setPreparedStatementDelete(PreparedStatement stm, Object key) throws SQLException {

        if (!(key instanceof String)){
            return;
        }

        stm.setString(1,(String)key);
    }

    /**
     * Es herdao de la clase abstracta
     */
    @SuppressWarnings("unchecked")
    protected void setPreparedStatementFinder(int tipoBusqueda, PreparedStatement stm, Collection sets) throws
    SQLException {

        switch (tipoBusqueda){

            case BUSCAR_ID_MAPA:

                stm.setString(1,(String)sets.toArray()[0]);
                break;

        }
    }

```

CAPITULO V: Fase de Construcción

```

}

/**
 * Es heredao de la clase abstracta
 */
protected void setPreparedStatementInsert(PreparedStatement stm, Object obj) throws SQLException {

    if (!(obj instanceof Punto)){
        return;
    }

    Punto punto = (Punto) obj;

    stm.setString(1,punto.getIdImagenMapa());
    stm.setString(2,punto.getCodRefObra());
    stm.setObject(3,punto.getPuntoReferencia());
    stm.setString(4,punto.getTipoObra());
    stm.setString(5,punto.getFinalidad());
    stm.setBoolean(6, punto.isVisible());

}

/**
 * Es heredao de la clase abstracta
 */
protected void setPreparedStatementLoadPK(PreparedStatement stm, Object key) throws SQLException {

    if (!(key instanceof String)){
        return;
    }

    stm.setString (1, (String) key);

}

/**
 * Es heredao de la clase abstracta
 */
protected void setPreparedStatementUpdate(PreparedStatement stm, Object key, Object obj) throws SQLException
{

    if (!(key instanceof String)){

```

CAPITULO V: Fase de Construcción

```

        return;
    }

    if(!(obj instanceof Punto)){
        return;
    }

    Punto punto = (Punto) obj;

    /*Se crea un objeto de tipo punto pero de postgres, utilizamos
    los puntos de la clase Punto*/

    stm.setObject(1,punto.getPuntoReferencia());
    stm.setString(2,punto.getTipoObra());
    stm.setString(3,punto.getFinalidad());
    stm.setBoolean(4, punto.isVisible());
    stm.setString(5,(String) key);

    }
}

```

5.4. PRUEBAS

El flujo de trabajo de pruebas, constituye la sección final de esta fase. En ésta se pone a prueba todo el sistema, con el objetivo de depurar la implementación de los casos de uso a través de la utilización de la pruebas por unidad y pruebas de integración.

Las pruebas son una actividad fundamental de esta fase, debido a que prepara casos y procedimientos de pruebas que permiten comprobar la sucesiva integración de subsistemas hasta acabar con la línea base completa. Es decir, se implementan los casos de uso como componentes y se comprueba tanto de forma individual como en construcciones. Si una prueba no alcanza sus objetivos, los casos y procedimientos de prueba deberán ser modificados para lograrlos.

5.4.1. Pruebas por Unidad

Las pruebas por unidad evalúan los componentes establecidos como unidades individuales, verificando la interacción interna entre los ellos. Estas pruebas se aplicarán a través de las pruebas de “caja negra”, a los componentes que integran el subsistema.

Para ello se identifican un conjunto de valores que pueden introducirse por un actor y se expresaron como clases para poder abarcar la totalidad de las ocurrencias de un evento de inserción de datos.

El caso de uso *Administrar Obra* proporciona entradas que deben ser ingresadas por el usuario a las cuales se les aplicarán las pruebas de “caja negra”, para el cual se han identificado las clases de equivalencia que muestra la tabla 5.1

Tabla 5.1. Clases de equivalencia para las pruebas de caja negra de “Administrar obra”. (1/2)

Nº	Dato	Clase de Equivalencia	Válido	No Válido
1	Código Obra	no es Alfanumérico		X
2	Código Obra	Alfanumérico y el caracteres ‘ - ’ menos al inicio	X	
3	Código Obra	es Alfanumérico	X	
4	Código Obra	0 >Longitud de caracteres< 30	X	
5	Código Obra	Longitud de caracteres> 30		X
6	Código Obra	0<Longitud de caracteres		X
7	Código Obra	En blanco		X
8	Denominación	Alfanumérico con caracteres especiales	X	
9	Denominación	En blanco		X

(Fuente: Dima Rodríguez)

CAPITULO V: Fase de Construcción

Tabla 5.2. Clases de equivalencia para las pruebas de caja negra de “Administrar obra”. (2/2)

Nº	Dato	Clase de Equivalencia	Válido	No Válido
10	Monto Inicial	Es numérico	X	
11	Monto Inicial	es Alfanumérico		X
12	Monto Inicial	es decimal	X	
13	Monto Inicial	En blanco		X
14	Ubicación	Alfanumérico con caracteres especiales	X	
15	Ubicación	En blanco	X	
16	Referencia	Alfanumérico con caracteres especiales	X	
17	Referencia	En blanco		X
18	Observaciones	Alfanumérico con caracteres especiales	X	
19	Observaciones	En blanco		X
20	Impacto Social	Alfanumérico con caracteres especiales	X	
21	Impacto Social	En blanco		X

(Fuente: Dima Rodríguez)

A partir de las clases de equivalencia expuestas anteriormente, se elaboran los casos de prueba que se presentan en la tabla 5.2.

Tabla 5.3. Caso de prueba de caja negra de “Incluir obra” (1/2)

Nº	Dato	Caso de prueba	Salida	Clases Cubiertas
1	Código Obra	'01_DET*15465'	Inválida	1
2	Código Obra	'12345689DEU111111111112222156634'	Inválida	5

(Fuente: Dima Rodríguez)

CAPITULO V: Fase de Construcción

Tabla 5.4. Caso de prueba de caja negra de “Incluir obra” (2/2)

Nº	Dato	Caso de prueba	Salida	Clases Cubiertas
3	Código Obra	‘01-DEU-11111’	Válida	2,4
4	Código Obra	‘01DET15465’	Válida	3,4
5	Código Obra	‘’	Inválida	6,7
6	Denominación	‘Reparación de la vía pariaguan – el pao. Prog. 3+500 a 13+000.’	Válida	8
7	Denominación	‘’	Válida	9
8	Monto Inicial	‘112144’	Válida	10
9	Monto Inicial	‘trescientos 300’	Inválida	11
10	Monto Inicial	155645,78	Válida	12
11	Monto Inicial	‘’	Inválida	13
12	Ubicación	‘El rincón - la toma, municipio Bolívar, Edo. Anzoátegui ’	Valida	14
13	Ubicación	‘’	Inválida	15
14	Referencia	‘Obra cercana a la catedral de Barcelona.’	Válida	16
15	Referencia	‘’	Inválida	17
16	Observaciones	‘Obra a realizar con carácter de urgencia.’	Válida	18
17	Observaciones	‘’	Inválida	19
18	Impacto Social	‘Obra que beneficiara a 500 familias.’	Válida	20
	Impacto Social	‘’	Válida	21

(Fuente: Dima Rodríguez)

5.4.2. Pruebas de Integración

Las pruebas de integración se utilizan para verificar que los componentes interaccionan entre si de la forma apropiada después de haber sido integrados. Para la ejecución de estas pruebas se delimitara nuestra área de estudio a la interacción entre dos módulos de vital importancia como lo son “Crear Punto de referencia a obra” y “Gestionar Obra”.

La secuencia de acciones realizadas durante las pruebas de integración consistieron en incluir una Obra al sistema con sus respectivas características, posteriormente se agregara un contrato relacionado a esta obra, de esta manera se buscará demostrar la completa integración entre los dos módulos en estudio. A continuación procederemos a mostrar las acciones a realizar durante las pruebas.

El sistema SISOVE mediante su ventana interna “Consultar Obra” (ver Figura 5.2) muestra que no se a ingresado al sistema ni una obra luego de realizar una búsqueda presionando el botón ubicado en la parte inferior izquierda.

Posteriormente se inicia el proceso de agregar una nueva obra, para ello se inicia el proceso de “crear Punto de referencia a obra” y “Gestionar Obra”, específicamente el módulo de agregar una Obra y agregar Contrato con todas sus correspondientes características.

Como veremos Inicialmente se crea el punto de referencia a obra (ver Figura 5.3) colocando un código de obra valido. Una vez ingresado este se pinta el punto en el mapa.

CAPITULO V: Fase de Construcción

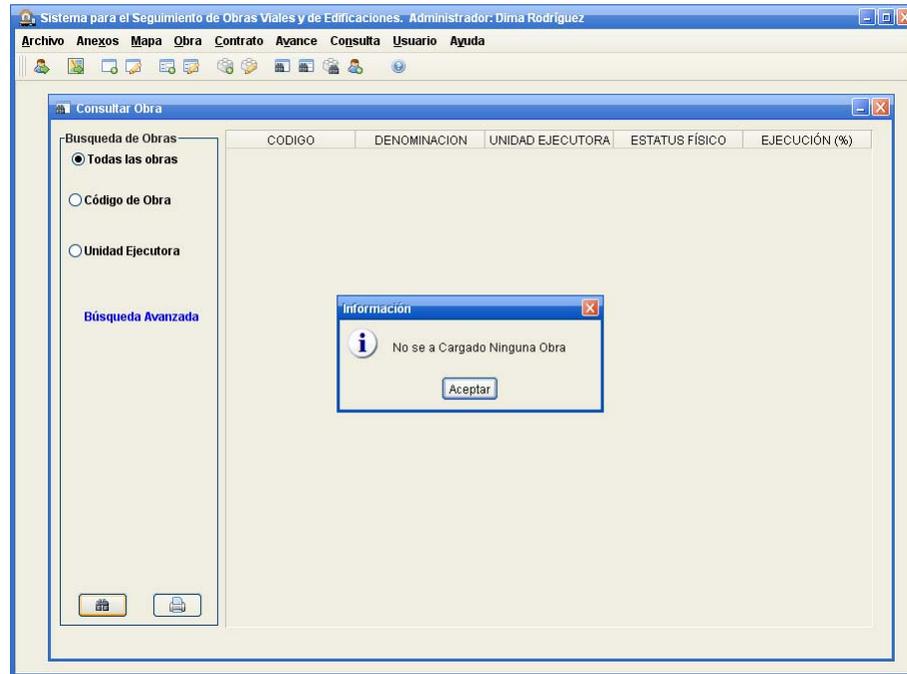


Figura 5.2 Ventana Interna “Consultar Obra”. (Fuente: Dima Rodríguez)

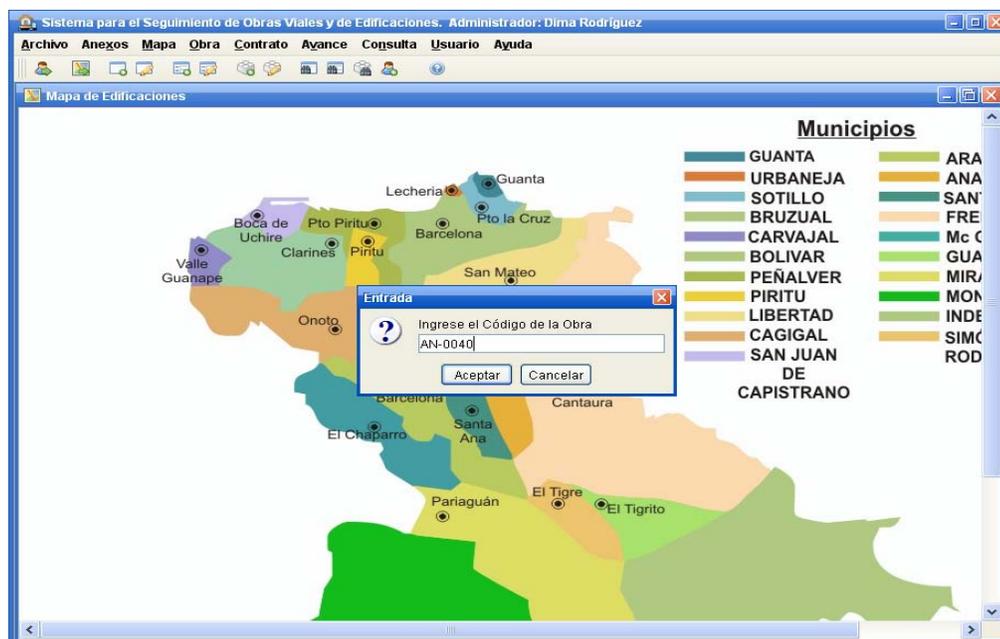


Figura 5.3 Crear Punto de Referencia a Obra. (Fuente: Dima Rodríguez)

CAPITULO V: Fase de Construcción

A la vez que se crea el punto, también se crea la ventana interna “Agregar Obra” (ver Figura 5.4) la cual será cargada con los datos necesarios para su ingreso al sistema.

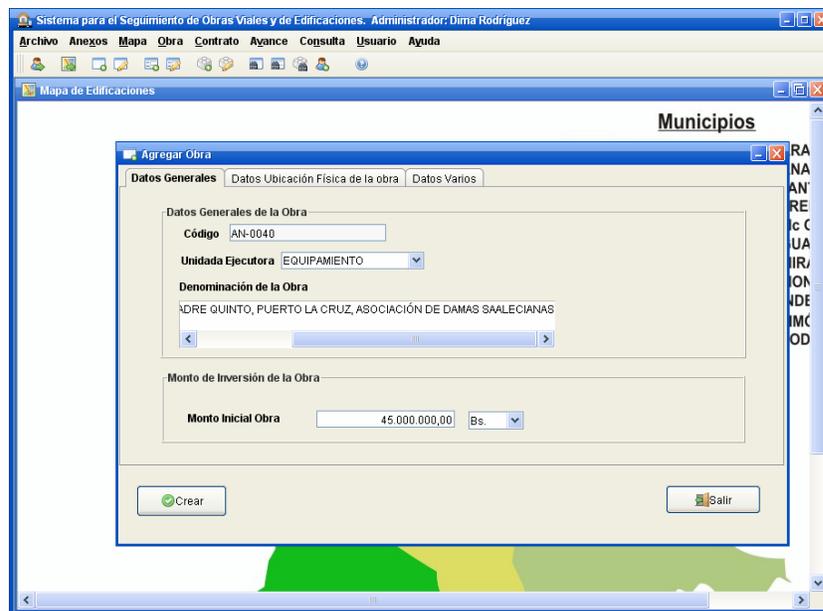


Figura 5.4 Ventana interna “Agregar Obra”. (Fuente: Dima Rodríguez)

Una vez cargado los datos de la obra se acepta ingresar los datos mostrando un mensaje de carga exitosa (ver Figura 5.5).

Luego de haber concluido exitosamente el proceso de agregar Obra, podemos apreciar que el mismo se incorpora al sistema por medio de la ventana interna Buscar Obra (ver Figura 5.6), con esto se logra alcanzar el objetivo deseado al agregar los datos de una obra.

Usando el manejador de PgAdmin3 se puede verificar la inserción correcta de la información en la base de datos “BDSISOVE”, específicamente en la tabla “Punto” (ver Figura 5.7) y “Obra” (ver Figura 5.8).

CAPITULO V: Fase de Construcción

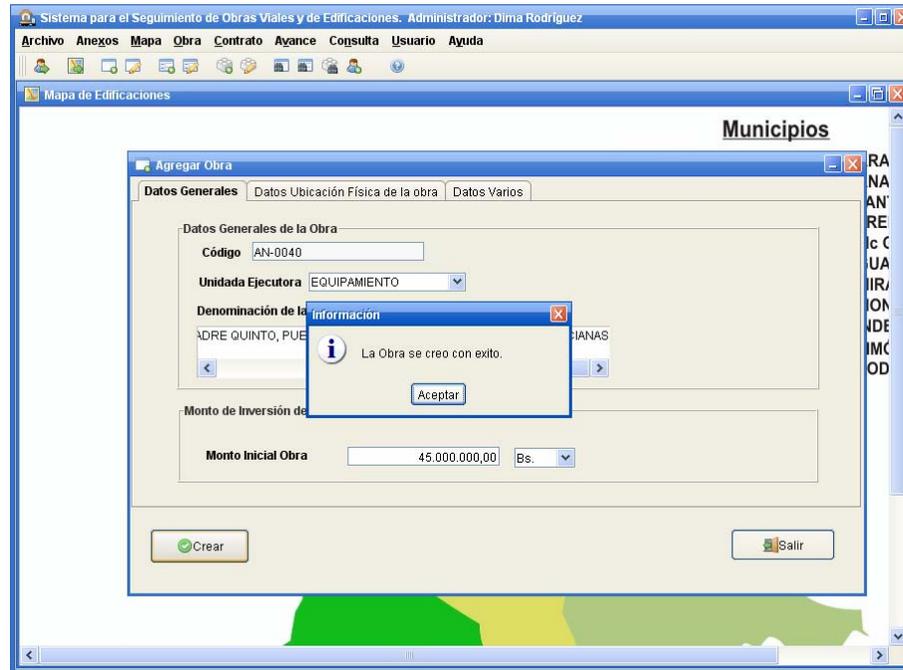


Figura 5.5 Ventana de dialogo con mensaje de Carga Exitosa de datos (Fuente: Dima Rodríguez)

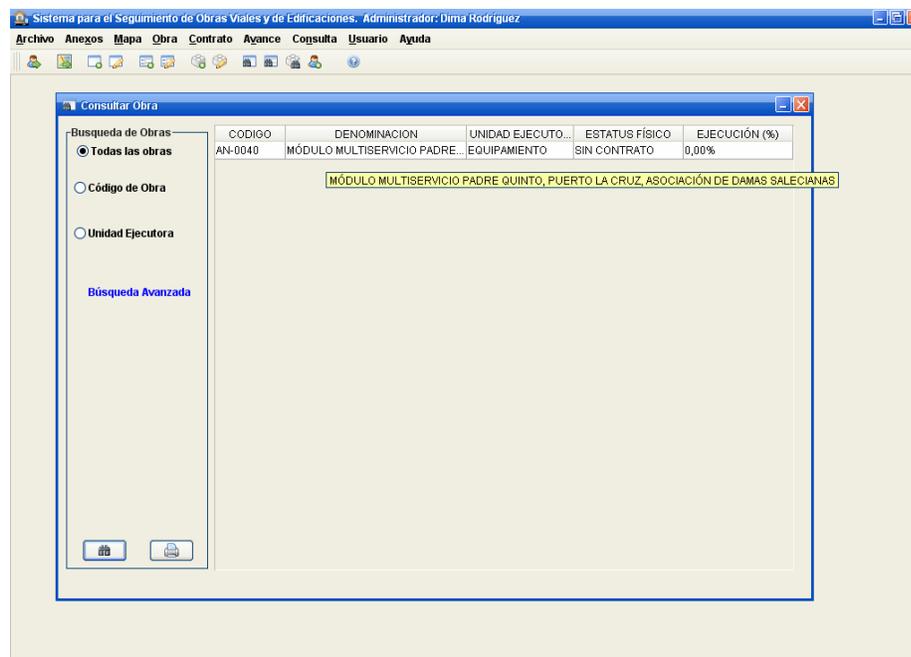


Figura 5.6 Ventana Interna "Consultar Obra". (Fuente: Dima Rodríguez)

CAPITULO V: Fase de Construcción

	idimagenmap character var	codrefobra [PK] caracte	puntoref point	tipoobra character var	finalidad character var	visible boolean
1	003	AN-0040	(423,98)	Obra	MÓDULO MULTI	TRUE
*						

1 rows.

Figura 5.7 Tabla Punto (Fuente: Dima Rodríguez)

	codigoobra [PK] caracte	denominacion character var	undejecutora character var	montototal double precis	tipomoneda character var	impactosocia	visible boolean
1	AN-0040	MÓDULO MULTI EQUIPAMIENTO	45000000	Bs.	LA OBRA BENEF	CERCA DEL TER "	CALLE DON BOS TRUE
*							

1 rows.

Figura 5.8 Tabla Obra (Fuente: Dima Rodríguez)

Antes de agregar el contrato asociado a la obra ingresada anteriormente realizamos la búsqueda de los contratos asociados a la obra primeramente ingresada por medio de la ventana interna Consultar Contrato (ver Figura 5.9).

CAPITULO V: Fase de Construcción

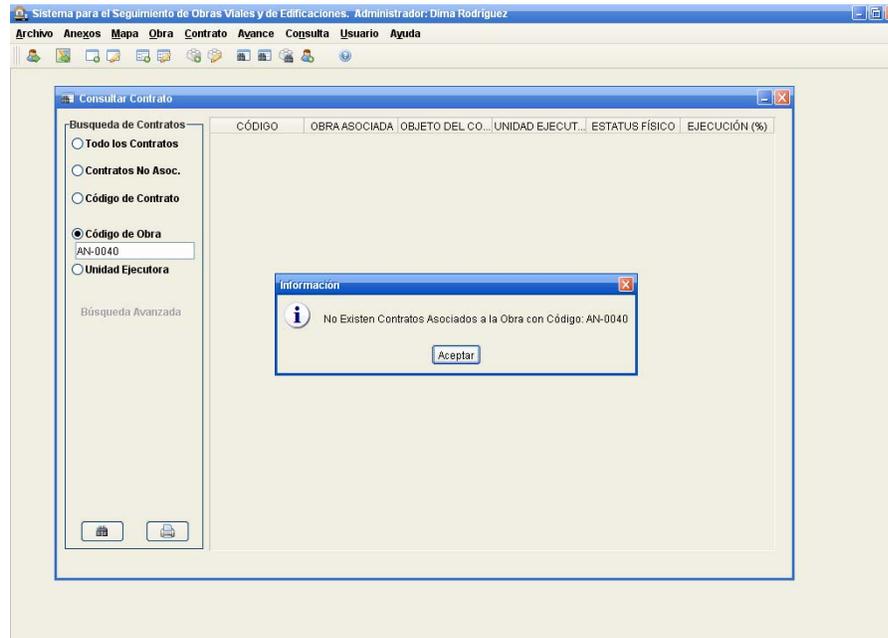


Figura 5.9 Ventana interna “Consultar Contrato” (Fuente: Dima Rodríguez)

Una vez ingresada la obra se procederá a agregar un contrato asociado a esta obra, se hace por medio de la ventana interna “Agregar Contrato” (ver Figura 5.10) y que permite comprobar la integración entre los módulos en estudio.

Como se podrá verificar en la siguiente imagen (ver Figura 5.11) que la información del contrato ingresado ha sido registrado en el sistema, con esto se logra comprobar que la información del contrato se ha asociado correctamente a la obra ingresada por el Administrador o Súper Usuario.

Obtenidos los resultados deseados se puede concluir que se ha alcanzado una correcta integración entre los módulos de “crear Punto de referencia a obra” y “Gestionar Obra”, estos resultados pueden ser comprobados directamente en la base de datos “BDSISOVE”, específicamente en la tabla “Contrato” (ver Figura 5.12).

CAPITULO V: Fase de Construcción

Sistema para el Seguimiento de Obras Viales y de Edificaciones. Administrador: Dima Rodríguez

Archivo Anexos Mapa Obra Contrato Avance Consulta Usuario Ayuda

Agregar contrato

Datos Generales Datos Físicos Datos Empresa Datos de Ing. Inspector Observaciones

Datos Generales del Contrato

Código del Contrato: AN-95-021 Asociado a Obra No Asociada a Obra

Unidad Ejecutora: EQUIPAMIENTO Obra Asociada: AN-0040

Órgano Rector: Ente Ejecutor:

Objeto del Contrato: CIO PADRE QUINTO, PUERTO LA CRUZ, ASOCIACIÓN DE DAMAS SALECIANAS

Fechas: Firma Ministro: 28/12/1995 Terminación: 09/03/1996

Beneficiarios: Tipo de Beneficiario: USUARIO Cantidad de Beneficiario: 200

Monto: Monto del Contrato: 4.000.000,00 Bs.

Crear Salir

Información: El Contrato se Creo con exito. Aceptar

Figura 5.10 Ventana interna "Agregar Contrato" (Fuente: Dima Rodríguez)

Sistema para el Seguimiento de Obras Viales y de Edificaciones. Administrador: Dima Rodríguez

Archivo Anexos Mapa Obra Contrato Avance Consulta Usuario Ayuda

Consultar Contrato

-Búsqueda de Contratos-

Todo los Contratos

Contratos No Asoc.

Código de Contrato

Código de Obra

AN-0040

Unidad Ejecutora

Búsqueda Avanzada

CÓDIGO	OBRA ASOCIADA	OBJETO DEL CO...	UNIDAD EJECUT...	ESTATUS FÍSICO	EJECUCIÓN (%)
AN-95-021	AN-0040	MÓDULO MULTI...	EQUIPAMIENTO	SIN AVANCE	0,00%

Figura 5.11 Ventana interna "Consultar Contrato" (Fuente: Dima Rodríguez)

CAPITULO V: Fase de Construcción

	codigocontral [PK] character	undejecontr character var	objcontrato character var	fechainicio date	fechaculmina date	montocontral double precis	tipomoneda character var	ubicacioncon character var	referenciacor character var	tipobeneficia character var	observacion character var	cantbeneficia integer
1	AN-95-021	EQUIPAMIENTO	MÓDULO MULTI	1996-01-11	1996-03-09	4000000	Bs.	CALLE DON BOS	CERCA DEL TER	USUARIO	"	200
*												

Figura 5.12 Tabla Contrato (Fuente: Dima Rodríguez)

5.5. RESUMEN DE LA FASE DE CONSTRUCCIÓN

Después de haber realizado las pruebas, tanto de unidad como de integración, y evaluando los resultados obtenidos en las mismas se determinó que el software a superado las pruebas del sistema y que ha alcanzado una capacidad operativa inicial, optima para su inclusión en las diversas pruebas que serán realizadas por los usuarios finales del sistema SISOVE.

CONCLUSIONES

- ✓ El empleo del Proceso Unificado de Desarrollo de Software conjuntamente con UML representa una herramienta muy eficaz en la construcción de un software, en virtud de que, el primero garantiza la obtención de un software de calidad y el segundo plantea los modelos que dan representación abstracta al sistema.
- ✓ Gracias a la clara información proporcionada por los usuarios y la documentación suministrada, se obtuvo un buen análisis del dominio del sistema en la elaboración de este proyecto, esto permitió una mayor fluidez en las etapas posteriores del desarrollo del software.
- ✓ El diseño previo de la arquitectura del sistema facilitó la visualización del ambiente con el cual se relaciona el sistema, proporcionando así una herramienta con la cual se lograba apreciar los errores o inconvenientes a solucionar, obteniendo una etapa de construcción más limpia y rápida.
- ✓ Se seleccionó el sistema manejador de Bases de datos PostgreSQL, por las distintas características que este posee, una de la principal es que es licenciado bajo la GPL de GNU, cumpliendo uno de los principales requisitos del sistema, también es comparable con un sistema comercial por sus múltiples atributos, a demás proporciona conectores para trabajar con el java
- ✓ El adecuado diseño del modelo de la base de datos resulta fundamental en todo desarrollo de software, ya que éste permite una adecuada esquematización del conocimiento empleado en el sistema.
- ✓ La correcta codificación de un sistema dependerá en gran parte de un correcto diseño de su arquitectura
- ✓ La integración de cada uno de los módulos programados mediante el lenguaje Java se realizó sin ningún inconveniente lo que permitió obtener una aplicación robusta, confiable y segura.

CONCLUSIONES

- ✓ La realización de las pruebas fue de mucha importancia, ya que permitió la comprobación del código y el correcto funcionamiento, evaluando así la calidad del software y garantizando su eficiencia
- ✓ La elaboración del manual de usuario permitió una rápida familiarización del usuario con el sistema.

RECOMENDACIONES

- ✓ Mantener suficiente espacio en disco duro, dado el posible incremento de la base de datos.
- ✓ Consultar el manual de usuario al momento de surgir alguna duda con respecto al manejo del sistema.
- ✓ No utilizar imágenes de mapas de grandes dimensiones, ya que, este consume mucho recurso y acarrearía problemas al sistema.
- ✓ Se recomienda realizar un módulo para que el sistema pueda realizar alejamiento y acercamiento (zoom) en las imágenes de los mapas y puntos, y así aprovechar aun más el espacio de la imagen y ser más específico en la ubicación de la obra con respecto a los puntos.
- ✓ Se sugiere realizar un módulo para que el sistema pueda refrescar (repintar) los puntos ubicados en los mapas mientras, la ventana del mapa este abierta. Para saber en tiempo real el estado actual de una obra.
- ✓ Se puede crear un módulo para llevar el seguimiento de los proyectos en estado de aprobación.
- ✓ Si es necesario que la aplicación utilice imágenes de grandes dimensiones, es preciso crear un archivo especial, que contendrá información necesaria para aumentar la memoria de la máquina virtual java. Pero esto implica que el programa deja de ser portable.
- ✓ Actualizar el sistema para que éste siempre esté acorde con las necesidades de los usuarios.

BIBLIOGRAFÍA

Fairley, R. (1987), “**Ingeniería de Software**”. 2da Edición, Editorial McGraw-Hill, México.

Grady, B. Y James, R. (1996), “**El Lenguaje Unificado en Modelado**”. 2da Edición, Editorial Addison Wesley, México.

Lockhart, T. (1996). “**Manual del usuario de PostgreSQL**”.
<http://fcp.unach.mx/manuales/download/Postgres-User.pdf>

Ivar, J.; Grady, B. y James, R. (2000), “**El Proceso Unificado de Desarrollo de Software**”. 1ra Edición, Editorial Pearson Addison-Wesley, España

Charlis, R. (2001), “**Base de Datos**”.
<http://www.monografias.com/trabajos7/bada/bada.shtml>.

Leonardo, M. (2003). “**Manual de Inspección y Residencia de Obras**”. 1ra Edición, Caracas.

Dimas, R. (2003), “**Desarrollo de un sistema para la administración y control de suministros y equipos perteneciente a un almacén de una empresa petrolera**”. Universidad de Oriente. Trabajo de Grado, Ingeniería en Computación, Puerto La Cruz, Venezuela.

Gaceta Oficial de la república Bolivariana de Venezuela, N° 38095, (2004) Decreto de software libre en la administración pública. (Decreto N° 3390)

BIBLIOGRAFÍA

Frank, Álvarez. (2004), “**Desarrollo de un Software para la Automatización de la Gestión y Control de Proyectos de una Empresa que Presta Servicios Petroleros**”. Universidad de Oriente. Trabajo de Grado, Ingeniería en Computación, Puerto La Cruz, Venezuela.

Carlos, M. (2005), “**Desarrollo de un software para el seguimiento de los bienes ubicados en el área de control de almacenamiento de una institución aduanera, ubicada en Guanta, estado Anzoátegui**”. Universidad de Oriente. Trabajo de Grado, Ingeniería en Computación, Puerto La Cruz, Venezuela.

Deomary, V. y Jorge R. (2005), “**Desarrollo de un Software para el Avance, Control y Valuaciones de Obras Civiles y Mecánicas, Aplicado al departamento de planificación de una empresa de construcción**”. Universidad de Oriente. Trabajo de Grado, Ingeniería en Computación, Puerto La Cruz, Venezuela.

Ministerio del poder popular para la Infraestructura. (2007). “**Manual del usuario del Sistema Automatizado De Seguimiento Y Evaluación De Obras Durante Las Etapas De: Contratación, Ejecución Y Culminación De MINFRA (SASEOMINFRA)**”. Caracas

Ana, O. y Stefano N. (2007), “**Desarrollo de un software para la automatización y seguimiento de las actividades que se realiza en el almacén de un taller de mantenimiento aeronáutico**”. Universidad de Oriente. Trabajo de Grado, Ingeniería en Computación, Puerto La Cruz, Venezuela.

Fuad, N. (2008), “**Desarrollo de un software para el control de inventario del material importado de una ensambladora de vehículos**”. Universidad de Oriente. Trabajo de Grado, Ingeniería en Computación, Puerto La Cruz, Venezuela.

BIBLIOGRAFÍA

Ministerio del poder popular para las obras públicas y vivienda. (2009). “**Misión**”.
<http://www.mopvi.gob.ve>

Ministerio del poder popular para las obras públicas y vivienda. (2009). “**Visión**”.
<http://www.mopvi.gob.ve>

Ministerio del poder popular para las obras públicas y vivienda. (2009). “**Ubicación**”.
<http://www.mopvi.gob.ve>

Wikipedia la enciclopedia libre (2010), “**Programación Orientada a Objeto**”.
http://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos

Wikipedia la enciclopedia libre (2010), “**Lenguaje de programación Java**”.
http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java

Sun Microsystems (2010), “**JAVA 2D**”.
<http://java.sun.com/products/java-media/2D/index.jsp>

ANEXO A**Decreto N° 3390****HUGO CHÁVEZ FRÍAS****PRESIDENTE DE LA REPÚBLICA****23 de Diciembre de 2004**

De conformidad con lo dispuesto en los artículos 110 y 226 de la Constitución de la República Bolivariana de Venezuela, 12 y 47 de la Ley Orgánica de la Administración Pública y, 2º, 19 y 22 del Decreto con Rango y Fuerza de Ley Orgánica de Ciencia, Tecnología e Innovación, en Consejo de Ministros,

CONSIDERANDO

Que es prioridad del Estado incentivar y fomentar la producción de bienes y servicios para satisfacer las necesidades de la población,

CONSIDERANDO

Que el uso del Software Libre desarrollado con Estándares Abiertos fortalecerá la industria del software nacional, aumentando y fortaleciendo sus capacidades,

CONSIDERANDO

Que la reducción de la brecha social y tecnológica en el menor tiempo y costo posibles, con calidad de servicio, se facilita con el uso de Software Libre desarrollado con Estándares Abiertos,

CONSIDERANDO

Que la adopción del Software Libre desarrollado con Estándares Abiertos en la Administración Pública y en los servicios públicos facilitará la interoperabilidad de los sistemas de información del Estado, contribuyendo a dar respuestas rápidas y oportunas a los ciudadanos, mejorando la gobernabilidad,

CONSIDERANDO

Que el Software Libre desarrollado con Estándares Abiertos, permite mayor participación de los usuarios en el mantenimiento de los niveles de seguridad e interoperatividad,

DECRETA

Artículo 1. La Administración Pública Nacional empleará prioritariamente Software Libre desarrollado con Estándares Abiertos, en sus sistemas, proyectos y servicios informáticos. A tales fines, todos los órganos y entes de la Administración Pública Nacional iniciarán los procesos de migración gradual y progresiva de éstos hacia el Software Libre desarrollado con Estándares Abiertos.

Artículo 2. A los efectos del presente Decreto se entenderá por:

Software Libre: Programa de computación cuya licencia garantiza al usuario acceso al código fuente del programa y lo autoriza a ejecutarlo con cualquier propósito, modificarlo y redistribuir tanto el programa original como sus modificaciones en las mismas condiciones de licenciamiento acordadas al programa original, sin tener que pagar regalías a los desarrolladores previos.

Estándares Abiertos: Especificaciones técnicas, publicadas y controladas por alguna organización que se encarga de su desarrollo, las cuales han sido aceptadas por la industria, estando a disposición de cualquier usuario para ser implementadas en un software libre u otro, promoviendo la competitividad, interoperatividad o flexibilidad.

Software Propietario: Programa de computación cuya licencia establece restricciones de uso, redistribución o modificación por parte de los usuarios, o requiere de autorización expresa del Licenciador.

Distribución Software Libre desarrollado con Estándares Abiertos para el Estado Venezolano: Un paquete de programas y aplicaciones de Informática elaborado utilizando Software Libre con Estándares Abiertos para ser utilizados y distribuidos entre distintos usuarios.

Artículo 3. En los casos que no se puedan desarrollar o adquirir aplicaciones en Software Libre bajo Estándares Abiertos, los órganos y entes de la Administración Pública Nacional deberán solicitar ante el Ministerio de Ciencia y Tecnología autorización para adoptar otro tipo de soluciones bajo las normas y criterios establecidos por ese Ministerio.

Artículo 4. El Ministerio de Ciencia y Tecnología, adelantará los programas de capacitación de los funcionarios públicos, en el uso del Software Libre desarrollado con Estándares Abiertos, haciendo especial énfasis en los responsables de las áreas de tecnologías de información y comunicación, para lo cual establecerá con los demás órganos y entes de la Administración Pública Nacional los mecanismos que se requieran.

Artículo 5. El Ejecutivo Nacional fomentará la investigación y desarrollo de software bajo modelo Software Libre desarrollado con Estándares Abiertos, procurando incentivos especiales para desarrolladores.

Artículo 6. El Ejecutivo Nacional fortalecerá el desarrollo de la industria nacional del software, mediante el establecimiento de una red de formación, de servicios especializados en Software Libre desarrollado con Estándares Abiertos y desarrolladores.

Artículo 7. El Ministerio de Ciencia y Tecnología será responsable de proveer la Distribución Software Libre desarrollado con Estándares Abiertos para el Estado Venezolano, para lo cual implementará los mecanismos que se requieran.

Artículo 8. El Ejecutivo Nacional promoverá el uso generalizado del Software Libre desarrollado con Estándares Abiertos en la sociedad, para lo cual desarrollará mecanismos orientados a capacitar e instruir a los usuarios en la utilización del Software Libre desarrollado con Estándares Abiertos.

Artículo 9. El Ejecutivo Nacional promoverá la cooperación internacional en materia de Software Libre desarrollado con Estándares Abiertos, con especial énfasis en la cooperación regional a través del MERCOSUR, CAN, CARICOM y la cooperación SUR-SUR.

Artículo 10. El Ministerio de Educación y Deportes, en coordinación con el Ministerio de Ciencia y Tecnología, establecerá las políticas para incluir el Software Libre desarrollado con Estándares Abiertos, en los programas de educación básica y diversificada.

Artículo 11. En un plazo no mayor de noventa (90) días continuos, contados a partir de la publicación del presente Decreto en la Gaceta Oficial de la República Bolivariana de Venezuela, el Ministerio de Ciencia y Tecnología deberá presentar ante la Presidencia de la República, los planes y programas que servirán de plataforma para la ejecución progresiva del presente Decreto.

Artículo 12. Cada Ministro en coordinación con la Ministra de Ciencia y Tecnología, en un plazo no mayor de noventa (90) días continuos, contados a partir de la aprobación por parte de la Presidencia de la República de los planes y programas referidos en el artículo anterior, publicará en la Gaceta Oficial de la República Bolivariana de Venezuela su respectivo plan de implantación progresiva del Software Libre desarrollado con Estándares Abiertos, acogiendo a los lineamientos contenidos en aquellos, incluyendo estudios de financiamiento e incentivos fiscales a quienes desarrollen Software Libre con Estándares Abiertos destinados a la aplicación de los objetivos previstos en el presente Decreto. Igualmente, las máximas autoridades de sus entes adscritos publicaran a través del Ministerio de adscripción sus respectivos planes.

Los planes de implantación progresiva del Software Libre desarrollado con Estándares Abiertos de los distintos órganos y entes de la Administración Pública Nacional, deberán ejecutarse en un plazo no mayor de veinticuatro (24) meses, dependiendo de las características propias de sus sistemas de información. Los Ministros mediante Resolución y las máximas autoridades de los entes que le estén adscritos a través de sus respectivos actos, determinarán las fases de ejecución del referido Plan, así como las razones de índole técnico que imposibiliten la implantación progresiva del Software Libre en los casos excepcionales, de acuerdo a lo establecido en el artículo 3 del presente Decreto.

Artículo 13. El Ministerio de Ciencia y Tecnología establecerá dentro de los planes y programas contemplados en el presente Decreto, mecanismos que preserven la identidad y necesidades culturales del país, incluyendo a sus grupos indígenas, para lo cual procurará que los sistemas operativos y aplicaciones que se desarrollen se adecuen a su cultura.

Artículo 14. Todos los Ministros quedan encargados de la ejecución del presente Decreto, bajo la coordinación de la Ministra de Ciencia y Tecnología.

Dado en Caracas, a los 23 días del mes de Diciembre de dos mil cuatro. Año 194° de la Independencia y 145° de la Federación.

(L.S)

HUGO CHAVEZ FRIAS

Refrendado:

El Vicepresidente de la República

(L.S)

JOSÉ VICENTE RANGEL

Todos los Ministros

ANEXO B

Sistema para el Seguimiento de Obras Viales Y de Edificaciones (SISOVE)

Manual del usuario:

Versión 1.0



INTRODUCCIÓN

El objetivo principal de la aplicación SISOVE (Sistema para el Seguimiento de Obras Viales y de Edificaciones), es permitir tanto a la Dirección regional MOPVI-Anzoátegui, como a otras oficinas de esta entidad, llevara de manera rápida y sencilla el seguimiento de las obras, ya que, podrán localizar las mismas dependiendo de su ubicación en el mapa, e indicando su estatus dependiendo su color. De esta forma se podrán atender las obras con mayor prioridad y tener conocimiento general del estado de todas las obras.

Requisitos del sistema.

- Sistema Operativo: Windows, Linux, Mac.
- Procesador: mínimo procesador Pentium III en adelante.
- Memoria: 512 en adelante
- Java: JRE 6 en adelante.

ACCESO AL SISTEMA

Al iniciar el sistema se muestra primero una ventana con el logo del sistema (Fig. 1), luego se abrirá una ventana de validación de usuario (Fig. 2) con la cual se permite ingresar el correspondiente nombre de usuario y la contraseña, si estos datos corresponden a un usuario existente en el sistema se permite su acceso con las restricciones definidas para el mismo.



Figura 1 Logo del sistema.

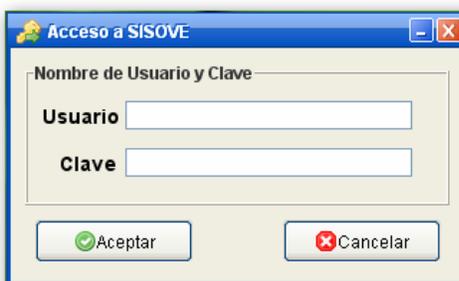


Figura 2 Ingreso del nombre de usuario y la contraseña.

Si la validación del nombre del usuario y contraseña es correcta se permite el acceso al sistema, en caso contrario se muestra el siguiente mensaje de error (Fig. 3).

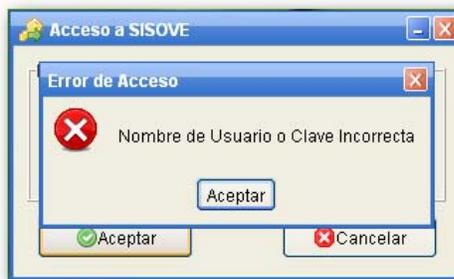


Figura 3 Mensaje de error.

Existe otro tipo de acceso, este es exclusivo para el súper usuario. Y para acceder al sistema éste necesita una clave extra para el ingreso. En la siguiente imagen se mostrara la ventana extra (fig. 4)

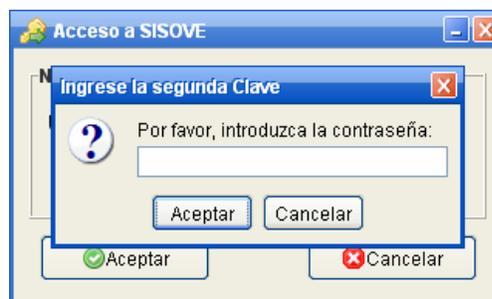


Figura 4 Segunda Clave para Súper Usuario.

TIPOS DE USUARIOS

Actualmente existen tres (3) tipo de usuarios que tendrán acceso al sistema, y que poseerán distintos privilegios.

✓ *Súper Usuario*: este tendrá el acceso completo al sistema, teniendo como tarea principal configurar la aplicación para que se conecte con el servidor, también puede restaurar obras eliminadas del sistema, más no de la Base de dato, entre otras funciones.

Anexo B: Manual de Usuario

- ✓ *Administrador*: no tiene tantos privilegios como el súper usuario, pero si los suficientes para administrar el sistema. Teniendo como tarea principal cargar los datos al sistema, así como, crear los usuario que van acceder al mismo.
- ✓ *Usuario*: es aquel que solo le interesa la información que se ha cargado al sistema.

VENTANAS PRINCIPALES

Existen tres tipos de ventanas (fig. 5) dependiendo del tipo de usuario que lo va a utilizar. Pero para explicar las funciones de las mismas vamos a utilizar la ventana principal del súper usuario (fig. 6).

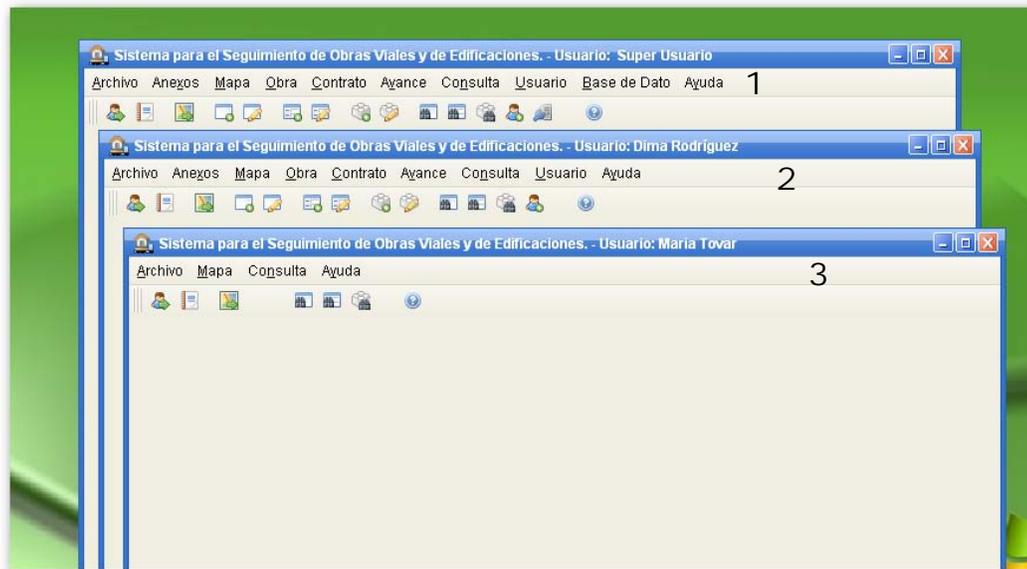


Figura 5 Ventanas principales para Súper Usuario (1), Administrador (2) y Usuario (3).

Anexo B: Manual de Usuario

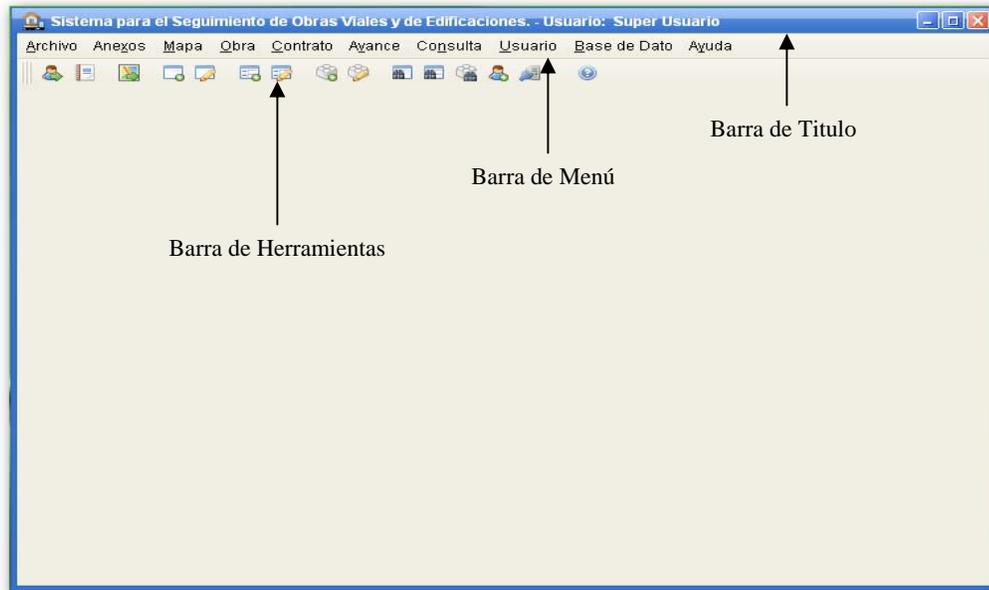


Figura 6 Ventana Principal del súper usuario.

- *Barra de título:* muestra tanto el nombre del sistema como el del usuario que accede al mismo.
- *Barra de menú:* contiene los menús que se pueden utilizar para realizar acciones en el sistema.
- *Barra de herramienta:* proporciona las acciones principales y más comunes a efectuar con el sistema.

BARRAS DE HERRAMIENTAS

Las barras de herramientas SISOVE (Fig. 7, 8 y 9) proporciona las siguientes acciones según el tipo de usuario:



Figura 7 Barra de Herramienta Súper Usuario.



Figura 8 Barra de Herramienta Administrador.



Figura 9 Barra de Herramienta Usuario.

 *Cambiar de Usuario*: permite al usuario cambiar de ambiente.

 *Visualizar Mapas*: muestra una ventana para seleccionar el o los mapas a mostrar.

 *Agregar Obra*: utilizado para crear una obra nueva.

 *Modificar Obra*: útil a la hora de modificar los datos de una obra.

 *Agregar Contrato* (asociado o no asociado a una obra): Creación de contratos nuevos asociados obra ⁽¹⁾ o no asociado a una obra ⁽²⁾

¹ Una vez ingresada la obra en el sistema, esta opción permite ingresar un contrato y asignarlo a la obra creada.

 *Modificar Contrato*: A la hora de modificar un contrato tenemos este enlace.

 *Agregar Avance*: indica que por medio de este enlace se puede agregar un avance del contrato.

 *Modificar Avance*: acceso directo a modificar un avance de un contrato.

 *Consultar Obras*: enlace directo para la consulta de las obras.

 *Consultar Contrato*: enlace directo para la consulta de los contratos

 *Consultar Avances*: enlace directo para la consulta de los avances

 *Crear Usuario*: atajo para la creación de un nuevo usuario.

 *Configurar Conexión*: va a la configuración de la conexión a la base de dato.

 *Ayuda*: Muestra la ayuda del sistema.

BARRA DE MENÚ

Las barras de menú SISOVE (Fig. 10, 11 y 12) proporciona las siguientes acciones según el tipo de usuario:

² Permite ingresar al sistema un contrato el cual por su tamaño, relevancia o naturaleza no forma parte de una gran obra. A nivel interno del sistema, una vez que se ingresa un contrato “sin obra asociada”, este contrato pasará a ser una *Obra* como tal.



Figura 10 Barra de menú para el Súper Usuario.



Figura 11 Barra de menú para el administrador.



Figura 12 Barra de menú para el usuario.

ARCHIVO

Utilizado para la realización de tareas de cambio de usuario, cambio de tema, así como, la salida del sistema.

Si se selecciona el menú “Archivo” se despliegan tres operaciones posibles (fig. 13)



Figura 13 Menú “Archivo”.

Cambiar usuario

Como ya se sabe el sistema cuenta con tres (3) tipos de usuario, y por ende, el sistema cuenta con una opción de cambiar de usuario.

Si el usuario selecciona el submenú “Cambiar Usuario” se mostrara una ventana de dialogo (fig. 14) que validara si de verdad desea cambiar de usuario.

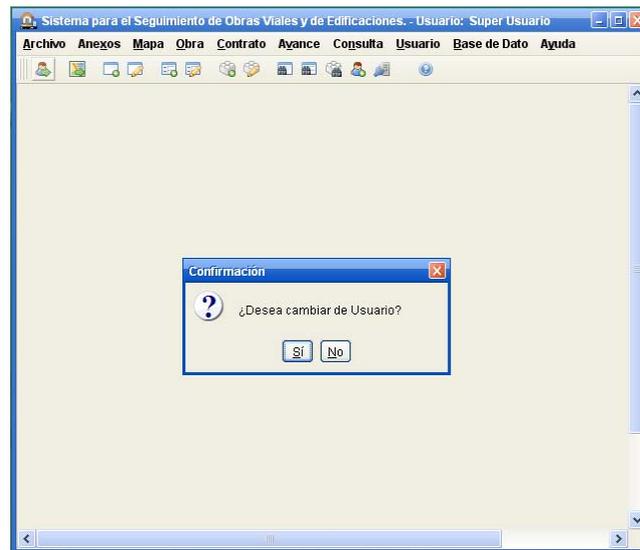


Figura 14 Ventana de dialogo confirmación.

Cambiar tema

Para personalizar la interfaz del sistema el usuario podrá cambiar el aspecto del sistema, esto lo puede hacer si se elige el submenú "Cambiar Tema" se mostrara un ventana de dialogo (fig. 15) indicándole una lista de temas a escoger.

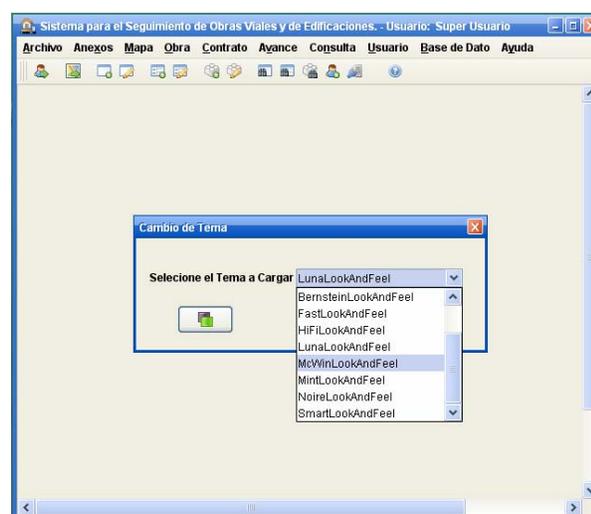


Figura 15 Ventana de dialogo con lista de temas.

Una vez seleccionado el tema se realizara el cambio de aspecto de todo el sistema (fig. 16) ⁽³⁾.

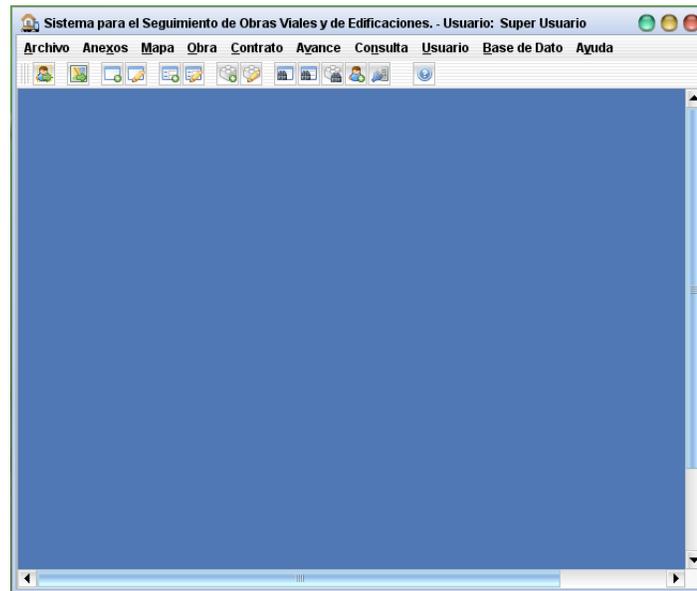


Figura 16 Ventana principal con nuevo aspecto.

Salir

Para salir del sistema seleccionamos el submenú “Salir”, pero antes del que sistema se cierre se le preguntara al usuario que confirme el cierre del mismo (fig. 17).

³ Para que el cambio se realice con éxito se debe reiniciar el sistema.

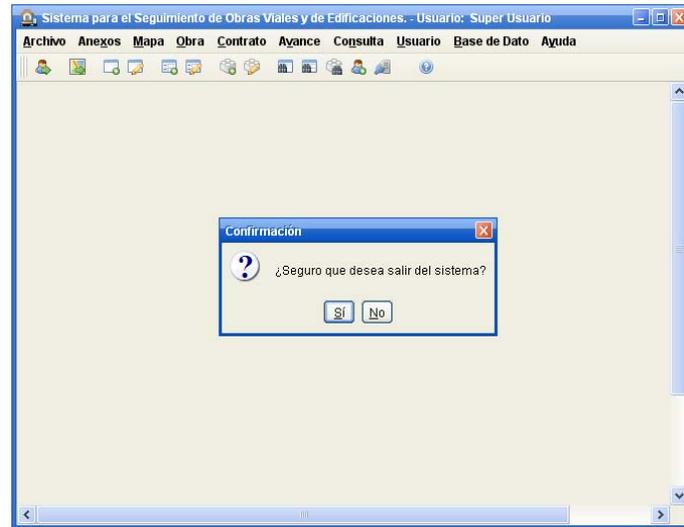


Figura 17 Ventana de dialogo Confirmación de la salida.

ANEXOS

Por medio de estos podremos ver y cargar la información requerida por el sistema de las empresas que ejecutaran un contrato. Así como también la de los ingenieros inspectores y residentes.

Si el usuario selecciona el menú "Anexos" (fig. 18) desplegara las opciones de Agregar al sistema datos de las empresas, ingenieros inspectores e ingenieros residentes.

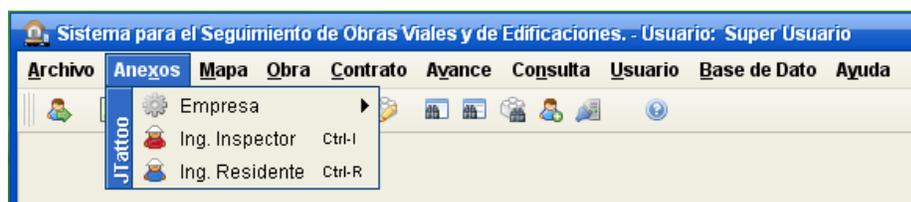


Figura 18 Menú "Anexos".

Empresa.

Es aquella que asume legalmente ante el Ente Contratante, la obligación y responsabilidad de realizar una obra o parte de ella a través de una relación contractual en un lapso establecido y por un monto acordado.

Si se selecciona el submenú “Empresa” (fig. 19) esta desplegará un conjunto de operaciones relacionadas con este, a demás de mostrar una lista de las empresas ya ingresadas al sistema.



Figura 19 Submenú “Empresa”.

Al elegir el submenú “Agregar” se mostrará la siguiente ventana (ver fig. 20) la cual contendrá un formulario con los datos que se requieren para cargar una empresa al sistema.

Figura 20 Ventana Agregar Empresa.

Anexo B: Manual de Usuario

- *R.I.F.:* registro fiscal de la empresa.
- *Nombre:* nombre de la empresa.
- *Dirección:* dirección fiscal de la empresa.
- *Teléfono:* número de teléfono de la empresa.
- *Fax:* número de fax de la empresa.
- *Correo electrónico:* dirección de Correo electrónico de la empresa.
- *Entidad bancaria:* nombre del banco a donde se le va a realizar los depósitos.
- *Número de cuenta:* número de cuenta del banco a donde se realizaran los depósitos.

Estos datos se pueden modificar seleccionando el submenú “Modificar”, mostrando la siguiente ventana (fig. 21)



Figura 21 Ventana Modificar Empresa.

O simplemente se pueden eliminar seleccionando el submenú “Eliminar” mostrando una ventana como esta (fig. 20). Aunque la empresa no será eliminada si está siendo utilizada en una obra.



Eliminar Empresa

Empresa Contratada

RIF

Nombre

Dirección

Teléfono

Fax

Correo Electrónico

Entidad Bancaria

Número de Cuenta

Figura 22 Ventana Eliminar Empresa.

Si queremos hacer una consulta de cuáles son las empresas que se han cargado al sistema se debe seleccionar “lista” en el submenú empresa y por medio de esta se mostrara la siguiente ventana (fig. 22).



Lista de Empresas

R.I.F.	NOMBRE	TELEFONO	E-MAIL	BANCO	NRO. CUENTA
--------	--------	----------	--------	-------	-------------

Figura 23 Ventana Lista de Empresas.

Ingeniero Inspector.

El Ingeniero Inspector es el representante del Ente Contratante y generalmente es el único autorizado para comunicar al contratista, las posibles modificaciones (Autorizadas por el contratante) o nuevas propuestas técnicas que modifiquen el proyecto o el presupuesto.

Para cargar, modificar o eliminar los datos del ingeniero inspector que va a supervisar una obra se debe seleccionar el submenú “Ing. Inspector” e inmediatamente se mostrará una ventana con la lista de los Ing. inspectores que se han agregado al sistema (fig. 23)



Figura 24 Ventana Ing. Inspector.

Por medio de los botones inferiores se podrá realizar las operaciones antes descrita. Se mostrará una ventana con los datos a ingresar al sistema (fig. 24)

Figura 25 Ventana Crear Ing. Inspector.

- *Cedula*: cedula del ingeniero Inspector.
- *C.I.V.*: código del colegio de ingeniero del ingeniero Inspector.
- *Nombre*: nombre del ingeniero Inspector.
- *Apellido*: apellido del ingeniero Inspector.
- *Teléfono de Oficina*: número de teléfono casa u oficia ingeniero Inspector.
- *Celular*: número del teléfono celular del ingeniero Inspector.
- *Correo Electrónico*: dirección del correo electrónico del ingeniero Inspector.

Para actualizar los datos de un ingeniero, se debe de escoger uno de la lista y presionar el botón de modificar . Y para eliminar igualmente se escoger uno y se presiona el botón eliminar .

Ingeniero Residente

El Ingeniero Residente es el representante técnico del Contratista en la obra y es el encargado de la planificación, ejecución de la obra y de las actividades de control, tales como **calidad**, organización del personal, actas, mediciones, valuaciones y demás actos administrativos similares.

Se muestra una ventana similar (fig. 25) si se selecciona el submenú “Ing. residente”, del menú Anexos.

Como vera las ventanas y formularios son similares al del Ing. Inspector.



Figura 26 Ventana Ing. Residente.

MAPA.

Utilizado de manera referencial para las obras, existen tres (3) imágenes de mapas que servirán para ubicar las obras en ellas, por medio de puntos que indicaran el estado de las obras e información relevante para el seguimiento de las mismas.

Si se selecciona el menú “Mapa” se despliega un conjunto de submenú referente a las imágenes de los mapas y puntos de referencias a las obras. (fig. 25)

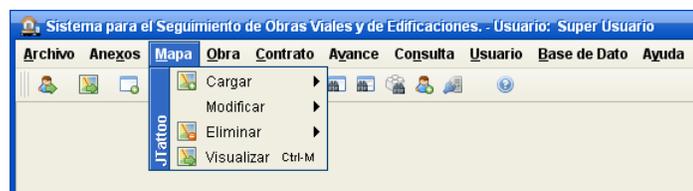


Figura 27 Menú “Mapa”.

Cargar Mapa.

Para mayor seguridad las imágenes de los mapas utilizados en el sistema han de guardada en la base de dato, por esto se ubica el archivo de la imagen y se resguarda en la base de dato.

Al seleccionar el submenú “Cargar” (fig. 27) se muestra una ventana de archivos mostrando las imágenes de los mapas a cargar al sistema (fig. 28).



Figura 28 Submenú “Cargar”.

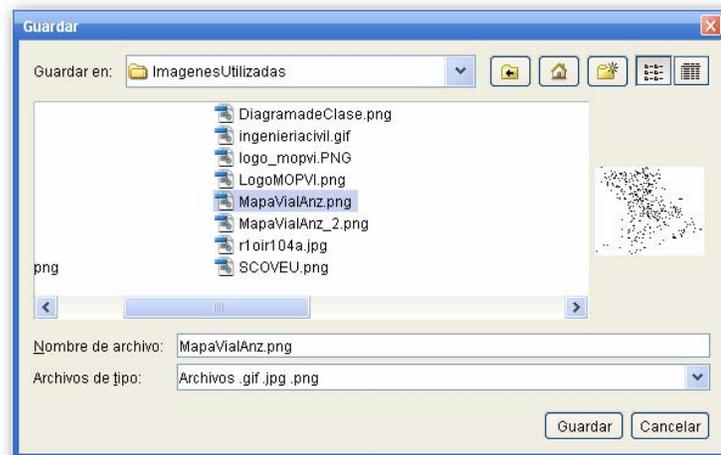


Figura 29 Ventana de archivos.

Modificar Mapa o Punto

En un futuro podríamos necesitar una nueva imagen de un mapa y por ende cabria la necesidad de reubicar de nuevos los puntos de las obras a su sitio original.

Anexo B: Manual de Usuario

Se debe elegir el submenú Modificar. Para mover el punto en los mapas o cambiar la imagen de los mapas (fig. 29).

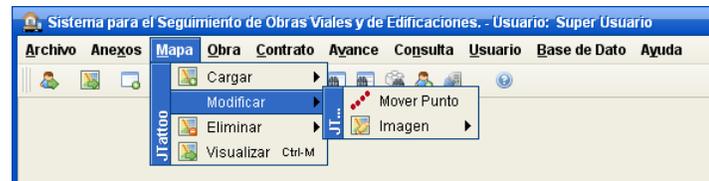


Figura 30 Submenú "Modificar".

Al seleccionar el submenú "Mover Punto" se debe de elegir el punto que se va a reubicar (fig. 30) y moverlo a su nueva ubicación, pero no sin antes preguntarle al usuario si esta de acuerdo con la reubicación (fig. 31).

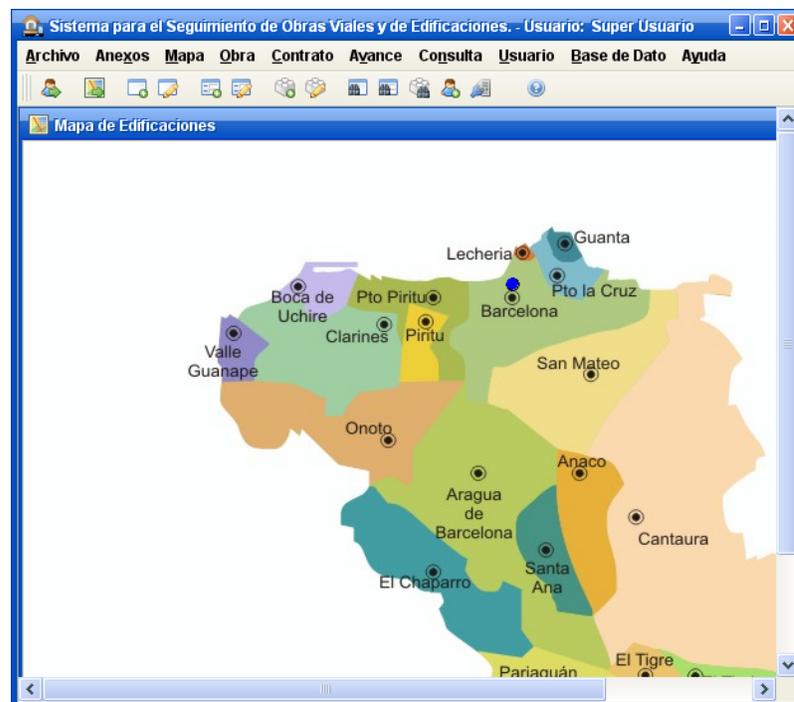


Figura 31 Punto a seleccionar.

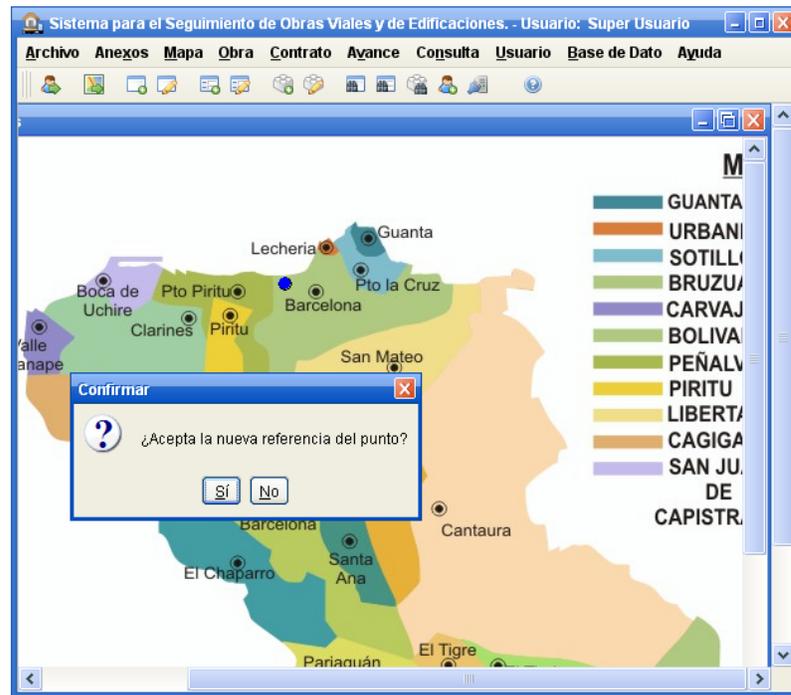


Figura 32 Reubicación y Mensaje

Eliminar Mapa

Para eliminar un mapa se debe escoger en el submenú eliminar (fig. 32) y elegir cual será la imagen del mapa a eliminar, teniendo en cuenta que no debe de existir ningún punto de referencia a obras sobre dicha imagen del mapa.

Luego se mostrara una ventana donde se preguntara si se desea realizar la eliminación (fig. 33) y se certifica la eliminación una ve eliminada (fig. 34).



Figura 33 Submenú "Eliminar".

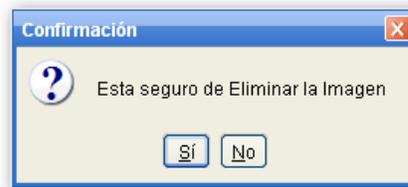


Figura 34 Ventana de dialogo Confirmación

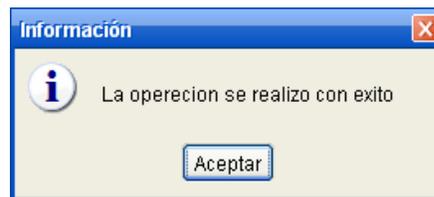


Figura 35 Ventana informativa.

Visualizar Mapa

De manera informativa, se pueden visualizar los mapas, con los puntos de referencias a las obras, pudiendo obtener más información referente a la misma.

Se pueden visualizar los mapas por medio del submenú "Visualizar", mostrándole una ventana de selección (fig. 35), para determinar cuales mapas se van a mostrar (fig. 36).



Figura 36 Ventana de selección.

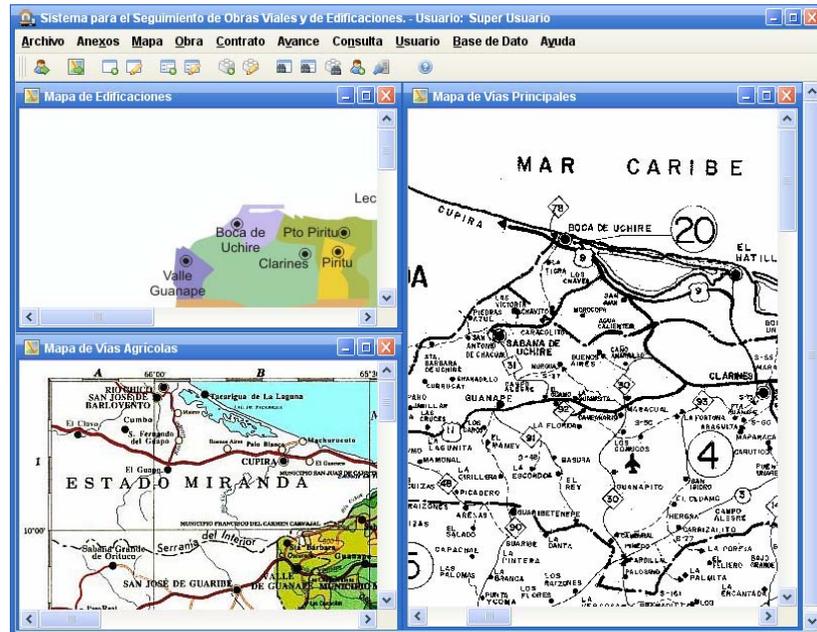


Figura 37 Mapas seleccionados.

OBRA

Para crear y editar la información general de una obra, se selecciona el menú “Obra” (fig. 37). Se debe considerar que en el sistema la entidad “obra” es de carácter “referencial”, es decir, sólo contiene la información de identificación básica. Gran parte del detalle de la obra es ingresado en los formularios de Contrato y de Avance los cuales serán explicados en lo sucesivo. ⁽⁴⁾

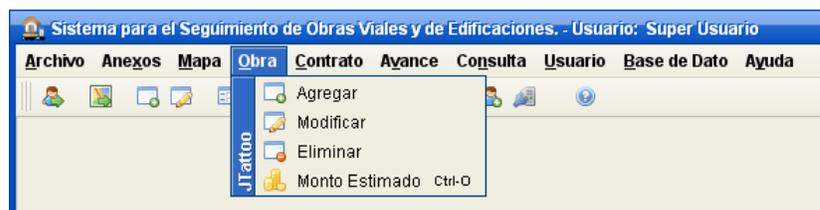


Figura 38 Menú “Obra”.

⁴ Es importante señalar que una obra podría tener uno (1) o más contratos asociados, pero si la obra es el contrato mismo, se recomienda seleccionar la opción “Crear un nuevo contrato sin obra asociada”.

Agregar Obra

Para agregar una obra, hay que seleccionar el mapa donde será creado el punto de referencia a la obra, para luego puntar el punto, una vez hecho eso, se mostrara una ventana entrada (fig. 38) donde se colocara el código de la obra.

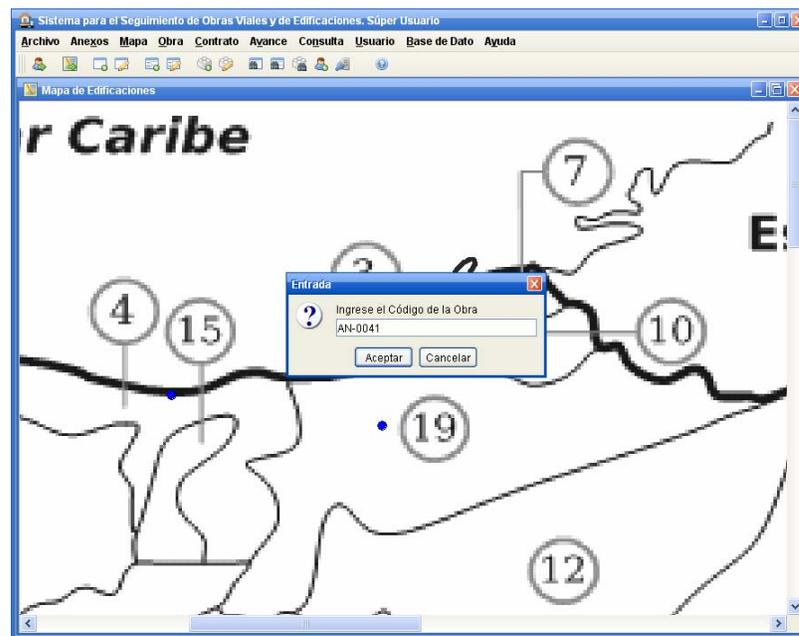


Figura 39 Ventana Entrada

Ingresado el código de la obra se muestra la ventana “Agregar Obra” (fig. 39). Esta ventana esta dividida en tres grupos que son “Datos Generales” (fig. 40), “Datos Ubicación Física de la Obra” (fig. 41), “Datos Varios” (fig. 42).

Anexo B: Manual de Usuario

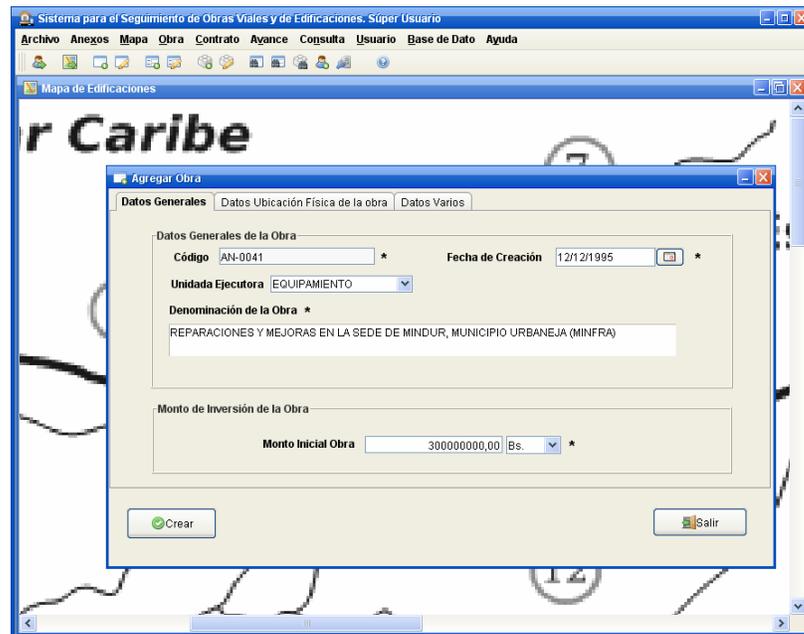


Figura 40 Ventana Agregar Obra.

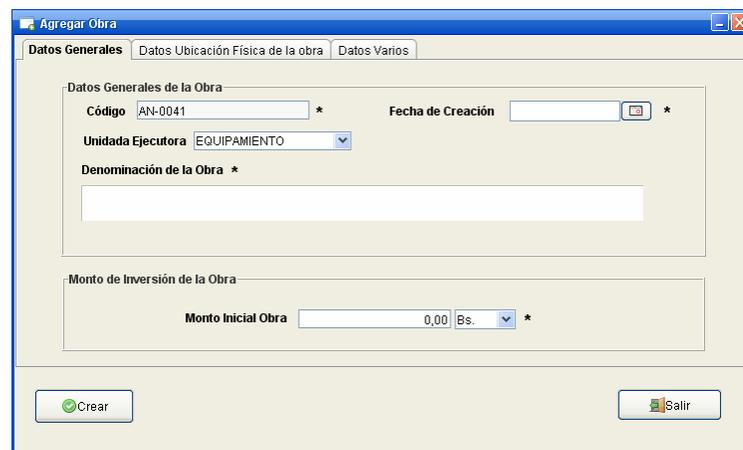


Figura 41 Ventana Agregar Obra pestaña "Datos Generales".

- *Código de la obra*: número o código único que identifica la obra.
- *Fecha de Creación*: fecha en la cual se crea la obra.
- *Unidad ejecutora*: unidad encargada de llevar todo lo referente a la obra.

Anexo B: Manual de Usuario

- *Denominación de la obra*: descripción de la obra según lo especificado en la carátula del contrato, adicione la “acción” para mayor especificidad de la descripción.
- *Monto Inicial de la obra*: monto estimado para realizar la obra.

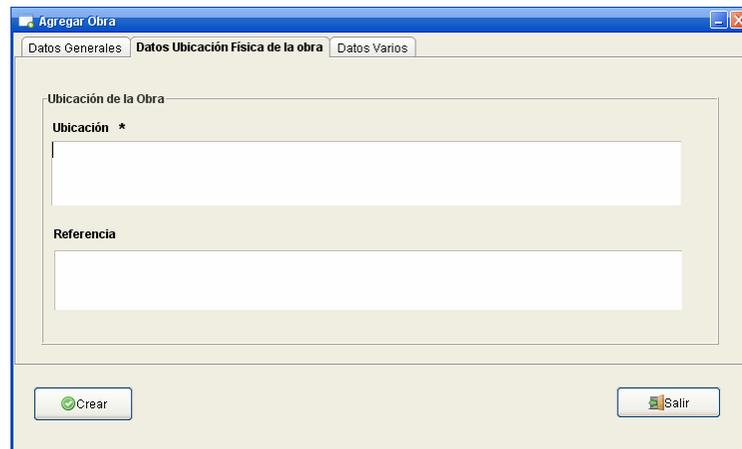
The screenshot shows a software window titled 'Agregar Obra'. It has three tabs: 'Datos Generales', 'Datos Ubicación Física de la obra' (which is selected), and 'Datos Varios'. The main area contains two text input fields. The first is labeled 'Ubicación de la Obra' and contains a sub-label 'Ubicación *'. The second is labeled 'Referencia'. At the bottom of the window, there are two buttons: 'Crear' with a green plus icon and 'Salir' with a red X icon.

Figura 42 Ventana Agregar Obra pestaña “Datos Ubicación Física de la Obra”.

- *Ubicación*: se coloca la dirección en donde se va a ejecutar la obra.
- *Referencia*: Es un punto de referencia para poder ubicar la dirección fácilmente

The screenshot shows the same 'Agregar Obra' window, but with the 'Datos Varios' tab selected. The main area contains two text input fields. The first is labeled 'Observaciones e Impacto' and contains a sub-label 'Observaciones'. The second is labeled 'Impacto Social de la Obra'. At the bottom of the window, there are two buttons: 'Crear' with a green plus icon and 'Salir' with a red X icon.

Figura 43 Ventana Agregar Obra pestaña “Datos Varios”.

Anexo B: Manual de Usuario

- *Observaciones*: aquí se colocara cualquier observación hecha a la obra.
- *Impacto social*: se debe indicar el beneficio social que trae la obra a las comunidades usuarias o cercanas, en términos de mejoramiento de la calidad de vida de los habitantes, en el acceso a los servicios básicos como alimentación, salud, educación, recreación, cultura.

Eliminar Obra

Antes de eliminar una obra, el sistema muestra una advertencia (fig. 43) una vez aceptada, se muestra la ventana eliminar obra (fig. 44). Luego de verificar los datos de la obra a eliminar y se acepta a eliminación se muestra una ventana de confirmación (fig. 45), la cual, el usuario determinara si la obra se elimina parcial o totalmente del sistema.

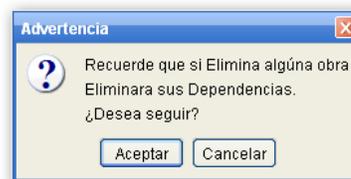


Figura 44 Ventana Advertencia.

A window titled "Eliminar Obra" with three tabs: "Datos Generales", "Datos Ubicación Física de la obra", and "Datos Varios". The "Datos Generales" tab is active. It contains several input fields: "Código" (text), "Fecha de Creación" (calendar icon), "Unidad Ejecutora" (dropdown menu with "VALIDIDAD" selected), "Denominación de la Obra" (text area), and "Monto Inicial Obra" (text field with "0,00 Bs." and a dropdown). At the bottom, there are two buttons: "Eliminar" and "Salir".

Figura 45 Ventana Eliminar Obra.

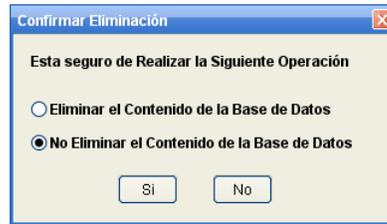


Figura 46 Ventana Confirmar Eliminación.

Monto Estimado

Para la ejecución de una obra es posible que variara en el tiempo el monto necesario para culminar una obra de gran escala, es por esto que se debe ingresar datos relacionados con los montos de inversión de la obra.

Por medio de la presente ventana monto estimado de la obra (fig. 46) se muestran una lista de los ajustes realizados al monto de una obra para su culminación. Estos datos son ingresados por medio de la ventana cargar monto (fig. 47).

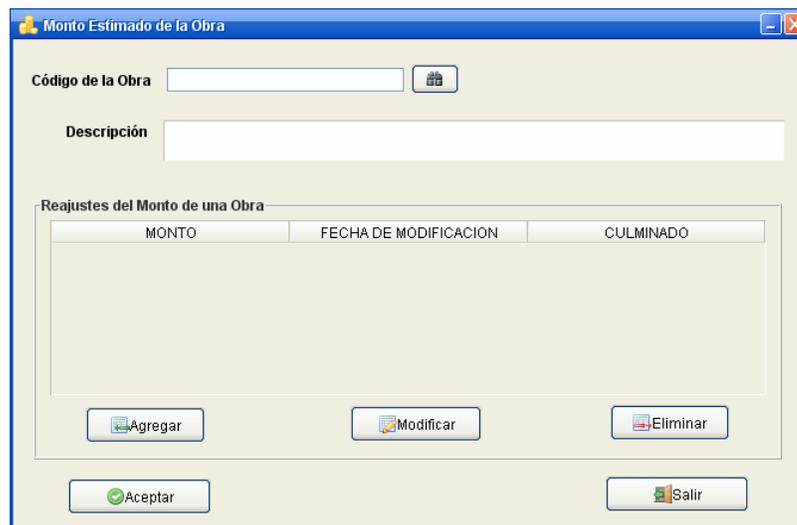


Figura 47 Ventana Monto Estimado de la Obra.

Figura 48 Ventana Cargar nuevo monto.

- *Monto Estimado*: Monto estimado para culminar la obra.
- *Fecha de Modificación*: fecha en que se modifico el monto.
- *Obra Culminada*: Si la obra en general se ha culminado se debe seleccionar “SI”, de lo contrario se debe mantener en “NO”.

En el sistema se pueden registrar dos clases de contratos:

- *Contrato asociado a una obra*: se emplea cuando se trata de un contrato que está inmerso dentro de una gran obra, tratándose ésta de una obra que ejecuta varios contratos en forma independiente.
- *Contrato no asociado a una obra*: se emplea cuando los contratos son pequeños y no están circunscritos dentro de una gran obra o cuando se trata de obras donde no se desea reportar cada uno contratos por separado, puesto que son demasiados y la información sería muy extensa. En este caso, toda la información de la obra es vaciada en los formularios de contratos bajo la figura de “Contrato sin obra asociada” esta pasaría a ser una obra.

CONTRATO

El presente menú Contrato (fig. 48) es utilizado por el usuario para cargar, modificar o eliminar tanto los contratos asociados a una obra, como a los no asociados (se asume como una Obra).



Figura 49 Menú "Contrato".

Agregar Contrato Asociado a una Obra

Esta ventana compuesta por un conjunto de fichas esta dividida en cinco conjuntos generales más dos subconjuntos. Las fichas que componen esta ventana son "Datos Generales" (fig. 49), "Ubicación y características" (fig. 50), "Metas" (fig. 51), "Datos Empresa"⁽⁵⁾ (52), "Datos Ing. Residente"⁽⁶⁾ (53), "Datos Ingeniero Inspector"⁽⁷⁾ (fig. 54) y "Observaciones" (fig. 55).

⁵ Para cargar los datos de la empresa se hace clic en el botón buscar empresa mostrando una lista de las empresas cargadas al sistema, hacemos doble clic a una fila y se cargara los datos.

⁶ Para cargar los datos del Ing. Residente se hace clic en el botón buscar Ing. Residente mostrando una lista de los Ing. Residente cargados al sistema, hacemos doble clic a una fila y se cargara los datos.

⁷ Para cargar los datos del Ing. Inspector se hace clic en el botón buscar Ing. Inspector mostrando una lista de los Ing. Inspector cargados al sistema, hacemos doble clic a una fila y se cargara los datos.

The screenshot shows a software window titled "Agregar contrato" with a tabbed interface. The active tab is "Datos Generales". The form contains the following fields and controls:

- Datos Generales del Contrato:**
 - Código del Contrato:** Text input field with an asterisk (*).
 - Asociado a Obra / No Asociada a Obra:** Radio buttons.
 - Unidad Ejecutora:** Dropdown menu with "VALIDIDAD" selected.
 - Obra Asociada:** Text input field with a search icon and an asterisk (*).
 - Órgano Rector:** Text input field with an asterisk (*).
 - Ente Ejecutor:** Text input field with an asterisk (*).
 - Objeto del Contrato:** Large text input field with an asterisk (*).
- Fechas:**
 - Firma Ministro:** Text input field with a calendar icon and an asterisk (*).
 - Inicio:** Text input field with a calendar icon and an asterisk (*).
 - Culminacion:** Text input field with a calendar icon and an asterisk (*).
- Beneficiarios:**
 - Tipo de Beneficiario:** Text input field with an asterisk (*).
 - Cantidad de Beneficiario:** Text input field with "0" and an asterisk (*).
- Monto:**
 - Monto del Contrato:** Text input field with "0,00" and a currency dropdown set to "Bs." and an asterisk (*).

At the bottom of the window, there are two buttons: "Crear" (with a green checkmark icon) and "Salir" (with a red X icon).

Figura 50 Ventana Agregar Contrato ficha "Datos Generales".

- **Código del Contrato:** Se trata del código que identifica al contrato. Este viene especificado en la carátula del contrato y es único e inequívoco para cada contrato.
- **Tipo de contrato:** indica si el contrato esta asociado a una obra o no.
- **Unidad ejecutora:** unidad encargada de llevar todo lo referente a la obra.
- **Obra Asociada:** permite que el contrato que se ingresa sea vinculado o asociado a una obra previamente cargada.
- **Órgano rector:** Es el encargado de llevar a cabo todo lo concerniente a una obra.
- **Ente Ejecutor:** entidad encargada de llevar adelante una determinada obra con la aplicación de las medidas de control e inspección.
- **Objeto del Contrato:** indica el nombre o descripción del contrato de obra.
- **Fecha firma del Ministro:** Fecha en la cual el ministro firma el contrato
- **Fecha Inicio:** fecha en la cual las obras deben iniciar.
- **Fecha Finalización:** fecha en la cual se estipula terminar la obra.
- **Tipo de Beneficiario:** beneficiario que tendrá el contrato (la obra) una vez culminada, por ejemplo: paciente, alumnos, pasajeros, etc. según el tipo de obra que se construye.

Anexo B: Manual de Usuario

- *Cantidad de Beneficiario*: número de personas que se beneficiarán de la ejecución del contrato (obra).
- *Monto del contrato*: monto especificado en el contrato de obra.
- *Tipo de Moneda*: tipo de moneda para representar los montos.

Figura 51 Ventana Agregar Contrato ficha “Ubicación y características”.

- *Ubicación*: dirección en donde se va a ejecutar el contrato.
- *Referencia*: Es un punto de referencia para poder ubicar la dirección fácilmente.
- *Características*: permite especificar elementos caracterizadores de la obra para documentar sus aspectos diferenciadores y adicionar aspectos de capacidad.
 - *Tipo*: tipo de características. Ejemplo estación, puente, túnel.
 - *Numero*: cantidad del tipo de característica seleccionada. Por ejemplo 2 puentes.
 - *Descripción*: descripción del tipo de característica seleccionada. Ejemplo: Puentes de concreto armado con soportes de acero.

Figura 52 Ventana Agregar Contrato ficha “Metas”.

- *Metas Cuantitativas*: se ingresan las metas programadas del contrato. El sistema ofrece un máximo de cinco (5) metas programadas.
 - *Meta programada*: Indica la longitud de la meta Ejemplo: 10,12. Emplee solo números.
 - *Unidad de Medida*: Especifique la unidad de medida para la meta programada, por ejemplo: Metros, Kilómetros, etc.

Figura 53 Ventana Agregar Contrato ficha “Datos Generales Empresa”.

Anexo B: Manual de Usuario

The screenshot shows a software window titled "Agregar contrato" with a blue title bar. It features a tabbed interface with the following tabs: "Datos Generales", "Datos Físicos", "Datos Empresa", "Datos de Ing. Inspector", and "Observaciones". The "Datos de Ing. Residente" tab is currently selected. The main content area is titled "Ingeniero Residente" and contains a form with the following fields: "C.I.V." (with a green checkmark icon), "Cedula", "Nombre", "Apellido", "Teléfono de Oficina", "Celular", and "Correo Electrónico". At the bottom of the window, there are two buttons: "Crear" (with a green checkmark icon) and "Salir" (with a trash can icon).

Figura 54 Ventana Agregar Contrato ficha "Datos de Ing. Residente".

The screenshot shows the same "Agregar contrato" window, but with the "Datos de Ing. Inspector" tab selected. The main content area is titled "Ingeniero Inspector" and contains a form with the following fields: "C.I.V." (with a green checkmark icon), "Cedula", "Nombre", "Apellido", "Teléfono de Oficina", "Teléfono Celular", and "Correo Electrónico". The "Crear" and "Salir" buttons are visible at the bottom of the window.

Figura 55 Ventana Agregar Contrato ficha "Datos de Ing. Inspector".

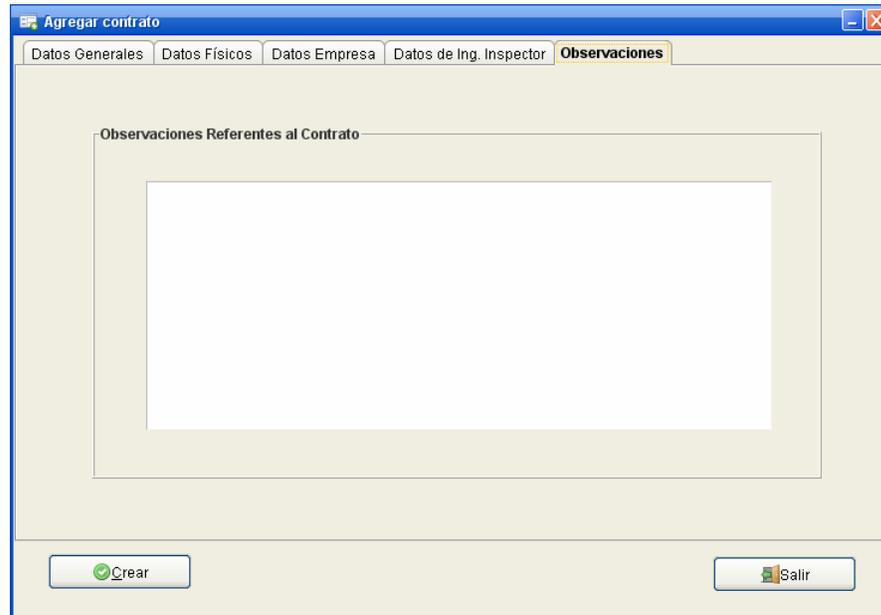


Figura 56 Ventana Agregar Contrato ficha "Observaciones".

- *Observaciones*: se ingresan las observaciones realizadas al contrato.

Agregar Contrato No Asociado a obra

Para la creación de un contrato No asociado (se asume como una obra) se debe seleccionar la opción "No asociado a obra" de la ventana "Agregar Contrato", una vez seleccionado sigue los mismos pasos para agregar una obra. Al final se muestra la ventana de "Agregar Contrato No Asociado a Obra" (fig. 56)

Anexo B: Manual de Usuario

Figura 57 Ventana Agregar Contrato No Asociado a Obra.

AVANCE

Para determinar la evolución de un contrato, necesitamos evaluar continuamente la ejecución del mismo, es por esto que se realizan un conjunto de avances referentes al contrato y así ver su progreso.

Con el menú “Avance” (fig. 57) el usuario puede ingresar datos de la evolución tanto de los contratos asociados o No a una obra.



Figura 58 Menú “Avance”.

Agregar Avance

Esta estará compuesta por un conjunto de fichas que a la vez contendrán otras fichas internas. Inicialmente se vera la ficha “Avance Físico” compuestas por las fichas “General” (fig. 58), “Metas Logradas” (fig. 59), “descripción Obra” (fig. 60), “Reporte Fotográfico” (fig. 61). Luego tendremos la ficha “Avance Administrativo”, el cual contendrá las siguientes fichas “Valuaciones del Contrato” (fig. 62), “Prorroga o Paralizaciones” (fig. 63). Seguidamente se tiene la Ficha “Avance Financiero” formado por las fichas “Monto” (fig. 64) “Fianzas y Anticipos” (fig. 65), “Amortizaciones” (fig. 66). Y por ultimo se tiene la ficha “Observaciones” (fig. 67).

The screenshot shows a software window titled "Agregar Avance". It has a main tab "Avance Físico" and sub-tabs "General", "Metas Logradas", "Descripción Obra", and "Reporte Fotográfico". The "General" sub-tab is active. The form contains the following fields:

- Datos del Contrato:**
 - Código del Contrato: [text box] *
 - Número de Control: [text box]
 - Objeto del Contrato: [text box]
 - Lapso de Ejecución del Contrato: [text box]
 - Porcentaje del Avance Físico: [text box] 0,00%
 - Número de Empleos Directos: [text box] 0
 - Número de Empleos Indirectos: [text box] 0
- Periodo de Ejecución:**
 - Desde: [calendar icon] [text box] *
 - Hasta: [calendar icon] [text box] *
 - Semana N°: [text box] *
- Estatus Físico:**
 - Estatus Físico: EJECUTANDOSE (dropdown menu)

At the bottom, there are two buttons: "Agregar" (with a green plus icon) and "Salir" (with a green minus icon).

Figura 59 Ventana Avance de contrato ficha “General”.

- *Código del contrato:* código del contrato al cual se le agregara un avance.
- *Número de control:* Número de control de los avances.
- *Objeto del contrato:* se llena automáticamente al cargar los datos del contrato.
- *Lapso de ejecución del Contrato:* se carga automáticamente determinada por la fecha de inicio y fecha de culminación estimada del contrato.

Anexo B: Manual de Usuario

- *Porcentaje de Avance Físico*: se determina automáticamente al ingresar el monto ejecutado.
- *Número de empleos Directo*: número estimado de empleos directos que demanda o demandará el contrato (asociado a la obra) durante su ejecución.
- *Número de empleos Indirecto*: número estimado de empleos indirectos que demanda o demandará el contrato (asociado a la obra) durante su ejecución.
- *Desde*: Fecha en que se inicia la supervisión.
- *Hasta*: Fecha en que inicializa la supervisión.
- *Semana N°*: número de la semana del año.
- *Estatus Físico*: estado físico de la obra.

META PROGRAMADA	META LOGRADA	UNIDAD DE MEDIDA

Figura 60 Ventana Avance de contrato ficha “Metas Logradas”.

Metas cuantitativas Logradas: se va actualizando las metas cuantitativas logradas hasta la fecha.

Anexo B: Manual de Usuario

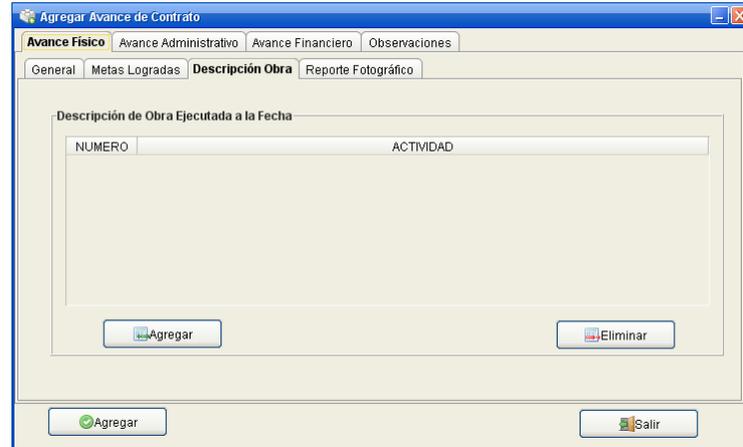


Figura 61 Ventana Avance de contrato ficha “Descripción Obra”.

- *Descripción de Obra*: se enumeran los trabajos realizados hasta esa fecha.
 - *Número*: número de actividad.
 - *Actividad*: actividad realizada.



Figura 62 Ventana Avance de contrato ficha “Reporte Fotográfico”.

- *Antes*: se ingresan al sistema al menos dos (2) imágenes del avance antes de ejecutadas las actividades.

Anexo B: Manual de Usuario

- *Después*: se ingresan al sistema al menos dos (2) imágenes del avance después de ejecutadas las actividades.

Figura 63 Ventana Avance de contrato ficha “Valuaciones del Contrato”.

- *Valuaciones*: precios correspondientes a los trabajos realizados, a los fines del pago de la obra ejecutada.
 - *Numero*: número de la valuación.
 - *Monto Bruto*: monto bruto valuado.
 - *Total*: suma de los montos brutos.
- *Fecha de Inicio*: fecha se carga automáticamente, y es la fecha en que se debe iniciar la obra.
- *Fecha de Terminación*: se habilitara una vez se hayan concluido los trabajos, y es la fecha en que se firma el acta de terminación.
- *Aceptación Provisional*: fecha en la cual se firma el acta de aceptación provisional.
- *Recepción definitiva*: fecha en que se formo el acta de recepción definitiva.

Figura 64 Ventana Avance de contrato ficha “Prorroga o Paralizaciones”.

- *Estatus Administrativo*: estado en que se encuentra administrativamente el contrato.
- *Prorrogas*: lapso que se le otorga al ente contratado bien sea para que inicie o termine una obra por causas ajenas a él.
 - *Tipo de Prorrogas*: existen dos (2) tipos de prorrogas:
 - *Inicio*: Si existen inconvenientes para dar inicio a la obra en el lapso indicado en el contrato, se le otorgara una prorrogas.
 - *Finalización*: Si existen inconvenientes para finalizar la obra en el lapso indicado en el contrato, se le otorgara una prorrogas.
 - *Desde*: fecha en que se inicio la prorrogas.
 - *Hasta*: fecha de culminación de la prorrogas.
 - *Número de Prorrogas*: número de prorrogas.
 - *Justificación*: motivos por el cual se pide la prorrogas.
- *Paralización y Reinicio*: estados que puede tomar una obra independientemente de las causas que la produjeron, y por ende una vez subsanadas estas cusas la obras será reiniciada.
 - *Fecha Acta Paralización*: Fecha en que se firmo el acta de paralización.

Anexo B: Manual de Usuario

- *Número de Paralización*: número de paralización
- *Fecha Acta Reinicio*: Fecha de reinicio de la obra.
- *Causales*: causas que provocaron la paralización.

The screenshot shows a software window titled "Agregar Avance" with a tabbed interface. The active tab is "Avance Financiero", and the sub-tab is "Montos". Under "Variaciones del Contrato", there are five input fields: "Monto Contratado", "Monto Ejecutado", "Monto por Ejecutar", "Monto Relacionado", and "% Relacionado", each containing the value "0,00". To the right of these fields is a pie chart that is entirely blue. A legend below the chart indicates that blue represents "Porcentaje sin Ejecutar" and red represents "Porcentaje Ejecutado". At the bottom of the window, there are two buttons: "Agregar" (with a green checkmark icon) and "Salir" (with a green exit icon).

Figura 65 Ventana Avance de contrato ficha "Monto".

- *Monto contratado*: monto que se carga automáticamente, es el monto del contrato.
- *Monto Ejecutado*: monto que se ha ejecutado hasta la fecha.
- *Monto por Ejecutar*: monto que falta por ejecutar.
- *Monto Relacionado*: debe corresponder al monto total retenido y que la relación administrativa del contrato se corresponda con el cuadro demostrativo de obra ejecutada y que se hayan amortizado la totalidad de los anticipos otorgados.
- *Porcentaje Relacionado*: como dice porcentaje relacionado.

Anexo B: Manual de Usuario

The screenshot shows a software window titled "Agregar Avance" with a tabbed interface. The active tab is "Avance Financiero", which contains a sub-tab "Fianzas y Anticipos". Under this sub-tab, there are three rows of input fields:

- Row 1: "Fianza Fiel Cumplimiento %" (0,00%) and "Monto Fianza Fiel Cumplimiento" (0,00)
- Row 2: "Anticipo Contractual %" (0,00%) and "Monto Anticipo Contractual" (0,00)
- Row 3: "Anticipo Especial %" (0,00%) and "Monto Anticipo Especial" (0,00)

At the bottom of the window, there are two buttons: "Agregar" (with a green plus icon) and "Salir" (with a red X icon).

Figura 66 Ventana Avance de contrato ficha “Fianzas y Anticipos”.

- Fianza Fiel Cumplimiento: porcentaje del monto otorgado por concepto de fianza fiel cumplimiento.
- Anticipo Contractual: porcentaje del monto otorgado por concepto de anticipo contractual.
- Anticipo Especial: porcentaje del monto otorgado por concepto de anticipo especial.

The screenshot shows the same "Agregar Avance" window, but with the "Amortizaciones" sub-tab selected. It displays three rows of input fields for amortization:

- Row 1: "Amortización Fianza Fiel Cumplimiento" (0,00) and "0,00% Amortizado"
- Row 2: "Amortización de Anticipo Contractual" (0,00) and "0,00% Amortizado"
- Row 3: "Amortización de Anticipo Especial" (0,00) and "0,00% Amortizado"

The "Agregar" and "Salir" buttons are also present at the bottom.

Figura 67 Ventana Avance de contrato ficha “Amortizaciones”.

- *Amortización Fianza Fiel Cumplimiento*: retención por concepto de fianza fiel cumplimiento.

- *Amortización de Anticipo Contractual*: retención por concepto de de anticipo contractual.
- *Amortización de Anticipo Especial*: retención por concepto de de anticipo especial.

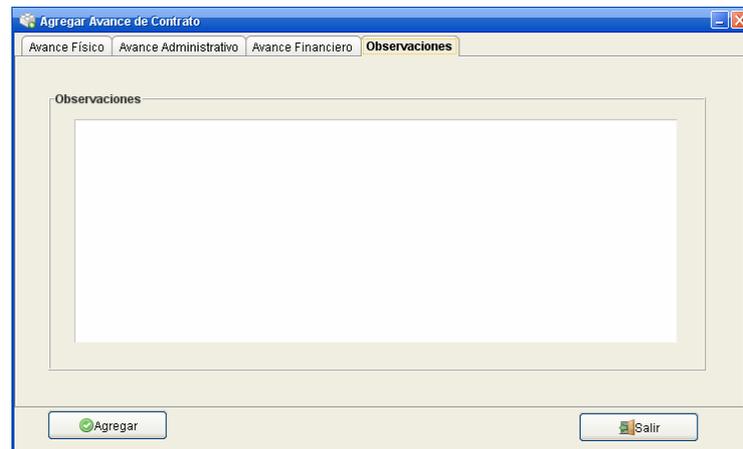


Figura 68 Ventana Avance de contrato ficha "Observaciones".

- *Observaciones*: observaciones hechas mediante la supervisión.

Lista de Avances.

A continuación se mostrara la ventana lista de avance (fig. 67), por medio de esta podemos tener acceso al conjunto de avance realizado a una obra, pudiendo así, imprimirlos.

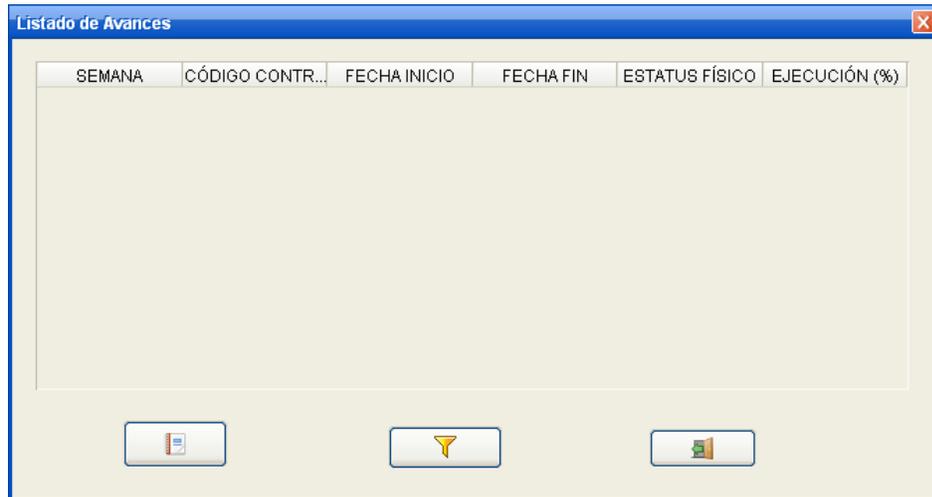


Figura 69 Ventana Lista de Avance.

CONSULTA

Por medio del menú “Consulta” (fig. 69) se permite se accede al módulo de consultas, donde el usuario podrán ver las obras, contratos o avance de los contratos, ingresados al sistema.



Figura 70 Menú “Consulta”.

Consultar Obra

Por medio de esta ventana (fig. 70) el usuario podrá ver todas las obras ingresadas al sistema, o si desea obtener un conjunto de las mismas filtrando la información.

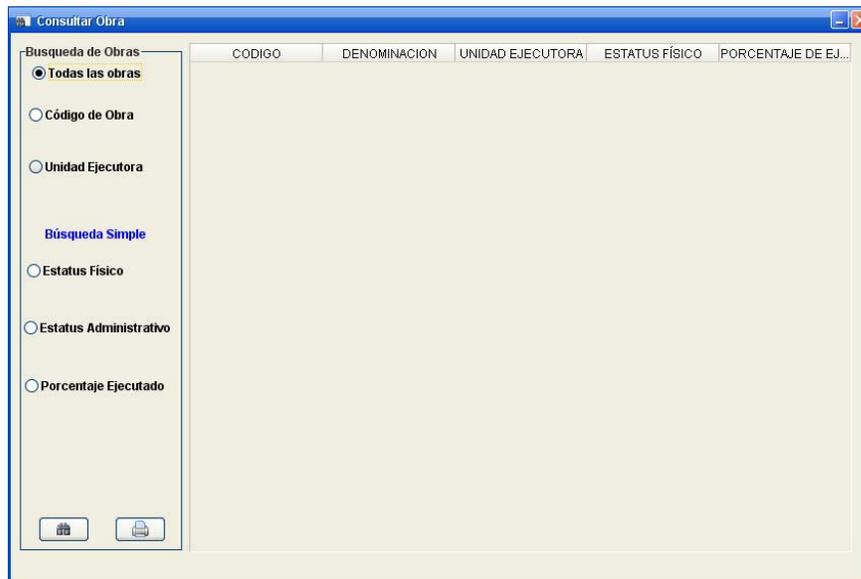


Figura 71 Ventana Consultar Obra.

Consultar Contrato

La presente ventana (fig. 71) es de utilidad en la búsqueda de los contratos asociados o no a una obra.



Figura 72 Ventana Consultar Contrato.

Consultar Avance

Para visualizar el avance de un contrato tenemos la ventana “Consultar Avance Contrato” (fig. 72) la cual tendrá información del como va el avance de un contrato.

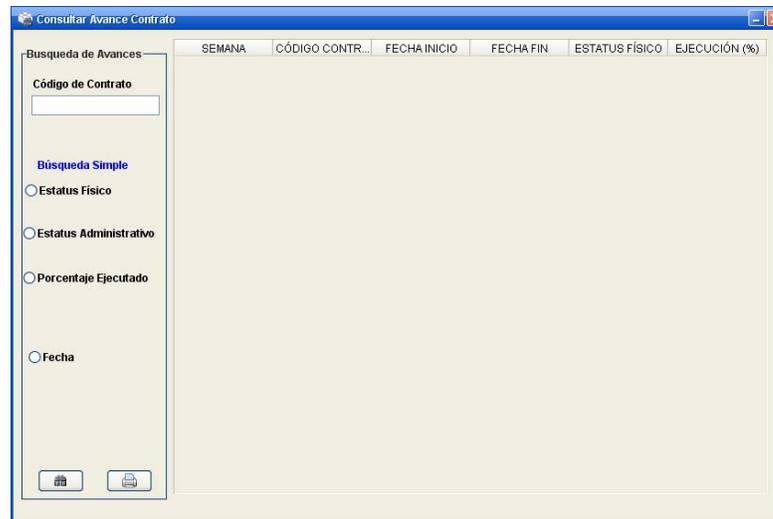


Figura 73 Ventana Consultar Avance Contrato.

USUARIO

Serán aquellos que tendrán acceso al sistema, y dependiendo del usuario, éste tendrá acceso restringido.

Por medio del menú “Usuario” (fig. 73) se puede ingresar al sistema los usuarios que tendrán acceso al mismo y sus alcances según su tipo.



Figura 74 Menú “Usuario”.

Crear Usuario

La ventana “Crear Usuario” es utilizada para el ingreso de los datos de los usuarios al sistema, determinando así el alcance del mismo. Esta ventan esta constituidas por tres fichas: “Datos Personales” (fig. 74), “Dato Institucional” (fig. 75) y “Datos para Acceso al Sistema” (fig. 76).

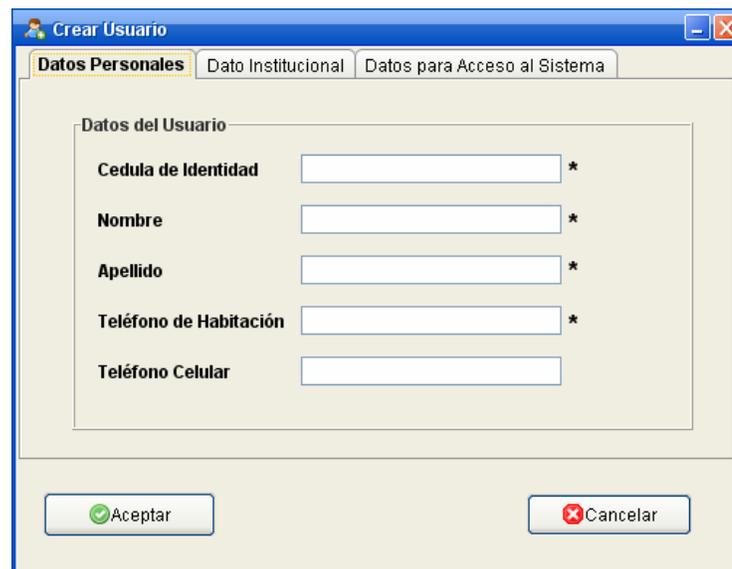
The image shows a software window titled "Crear Usuario" with a blue title bar. It contains three tabs: "Datos Personales" (selected), "Dato Institucional", and "Datos para Acceso al Sistema". The "Datos Personales" tab displays a form titled "Datos del Usuario" with five input fields: "Cedula de Identidad", "Nombre", "Apellido", "Teléfono de Habitación", and "Teléfono Celular". Each of the first four fields has an asterisk (*) to its right, indicating they are required. At the bottom of the window, there are two buttons: "Aceptar" with a green checkmark icon and "Cancelar" with a red X icon.

Figura 75 Ventana Crear Usuario ficha “Datos Personales”.

- *Cedula de Identidad*: cedula de identidad del usuario que utilizara el sistema.
- *Nombre*: nombre del usuario que utilizara el sistema.
- *Apellido*: Apellido del usuario que utilizara el sistema
- *Teléfono de Habitación*: número de teléfono de habitación u oficina.
- *Teléfono Celular*: número de teléfono celular.

Anexo B: Manual de Usuario

Figura 76 Ventana Crear Usuario ficha “Dato Institucional”.

- *Cargo*: cargo que desempeña en la institución.
- *Dpto. Adscrito*: departamento al cual fue asignado.

Figura 77 Ventana Crear Usuario ficha “Datos para Acceso al Sistema”.

- *Nombre de Usuario*: nombre de usuario para sistema
- *Contraseña de Usuario*: contraseña de usuario para sistema

- *Validar contraseña:* de nuevo la contraseña de usuario del sistema
- *Tipo de Usuario:* tipo de usuario que utilizara el sistema.

Lista de Usuarios

Se pueden ver los usuarios que tienen accesos al sistema por medio de la ventana “Lista de Usuarios” (fig. 77)



CEDULA	NOMBRE	DEPARTAMENTO	CARGO
14838994	yani, gonza	Gestión	jefa
14531689	Dima, Rodríguez	Dirección	Pasante

Figura 78 Ventana Lista de Usuario.

BASE DE DATO

Es de utilidad para enlazar la aplicación con el servidor de base de dato, así como también recupera datos eliminados del sistema.⁽⁸⁾

Este menú (fig. 78) proporciona las opciones de configurar la conexión con la base de dato y el restablecimiento de obras eliminadas del sistema, pero no de la base de dato.

⁸ Eliminado del sistema más no de la base de dato.



Figura 79 Menú "Base de Dato".

Configurar Conexión

La ventana mostrada a continuación (fig. 79) es la encargada de recabar los datos necesarios para conexión con la base de dato.

 The image shows a dialog box titled "Configurar Conexión". It contains a section titled "Datos para Configurar el Acceso al Servidor" with four input fields, each followed by an asterisk:

- Login:** postgres
- Password:** *****
- IP (Servidor):** localhost
- Base de Dato:** BDMOPVIANZ

 At the bottom of the dialog, there are two buttons: a refresh button (circular arrows) and a save button (floppy disk).

Figura 80 Ventana Configurar Conexión.

- *Login:* nombre del súper usuario del servidor.
- *Password:* clave para acceder al servidor.
- *IP (Servidor):* dirección IP del la PC donde se encuentra el servidor de base de dato Postgres.
- *Base de Dato:* Nombre de la base de dato.

Restaurar Obra

Las obras que han sido eliminada del sistema, más no de la base de dato pueden ser recuperada por el súper usuario.

La ventana “Restaurar Obra” (fig. 80) es utilizada para restituir la obra eliminada del sistema, pero no de la base de dato.

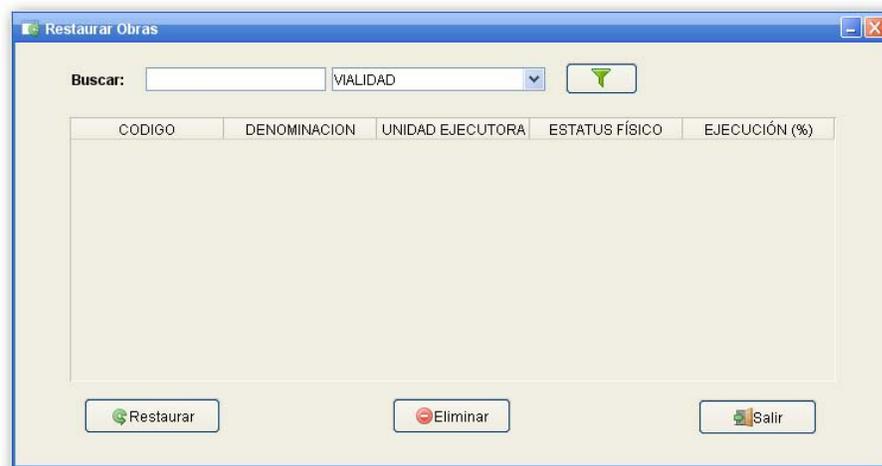


Figura 81 Ventana Restaurar Obras.

Restaurar Contrato

Los contrato no asociados a obras que han sido eliminada del sistema, más no de la base de dato pueden ser recuperada por el súper usuario.

La Ventana “Restaurar Contrato” (fig. 81) es utilizada para restituir el contrato no asociado eliminado del sistema, pero no de la base de dato.

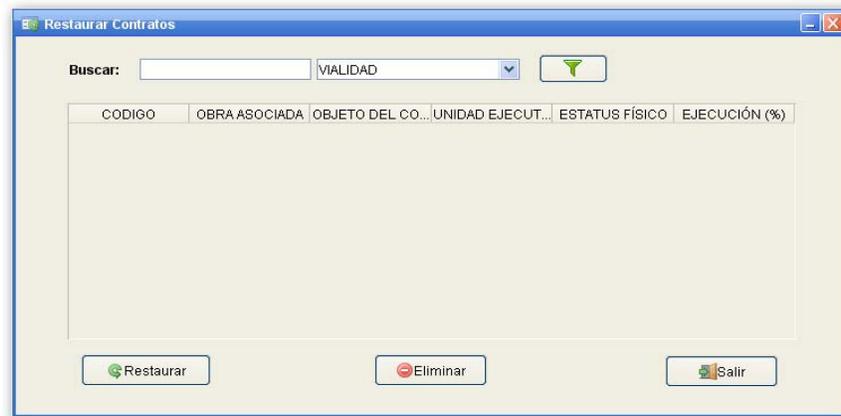


Figura 82 Ventana Restaurar Contratos.

AYUDA

Por medio del menú “Ayuda” (fig. 82) el usuario tiene acceso al sistema de ayuda desarrollado para asistirlo durante el uso del sistema.



Figura 83 Submenú “Ayuda”.

Ayuda SISOVE

El sistema contara con una ayuda para asistir al usuario en cualquier duda que pueda tener a la hora de realizar una operación en el sistema.

Para tener acceso a la ventana de ayuda (fig. 83), bien sea por la barra de menú, barra de herramienta o simplemente por medio de la tecla “F1”



Figura 84 Ventana de Ayuda del sistema.

Color Estado

Los puntos que hacen referencia a las obras, pueden cambiar de un momento a otro dependiendo del estatus en que se encuentre. Para esto se muestra una ventana informativa indicando el color dependiendo el estado de la obra o contrato No asociado (fig. 84).

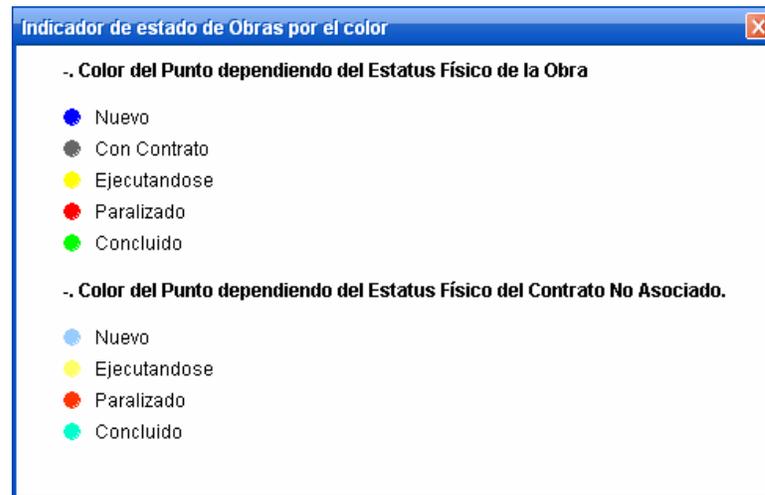


Figura 85 Ventana Indicadora del estado de la obra por color.

Acerca de SISOVE

En la ventana Acerca (fig.85) podrá ver información de las herramientas utilizadas en la elaboración del sistema.



Figura 86 Ventana Acerca de SISOVE.

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

TÍTULO	DESARROLLO DE UN SOFTWARE PARA EL SEGUIMIENTO DE LAS OBRAS POR PARTE DE LA DIRECCIÓN ESTADAL DEL MINISTERIO DEL PODER POPULAR PARA LAS OBRAS PÚBLICAS Y VIVIENDA - ESTADO ANZOATEGUI
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
RODRÍGUEZ CH., DIMA J.	CVLAC: 14.531.689 E MAIL: dimajoserodriguez@gmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

Desarrollo, Software, Aplicación, Base de Datos, Software libre, MOPVI, Seguimiento de Obra, automatización, Java, PostgreSQL, Computación.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ÀREA	SUBÀREA
INGENIERÍA Y CIENCIAS APLICADAS	INGENIERÍA EN COMPUTACIÓN

RESUMEN (ABSTRACT):

El siguiente proyecto de investigación se basa en el desarrollo de un sistema para el seguimiento de las obras viales y de edificaciones (SISOVE) para el centro regional del ministerio del poder popular para las obras públicas y vivienda del estado Anzoátegui. El software se encarga de proporcionar una interfaz amigable y de fácil manejo al personal que laboran en los distintos departamentos de la institución y muy especialmente al director del centro, que tiene la tarea de llevar el seguimiento de las obras ejecutadas en el estado, este trabajo lo realiza mediante consultas al sistema, el cual, le mostrara la imagen de un mapa referencial del estado y puntos que representaran a cada obra ingresada en el sistema. Esta información relacionada con las obras es guardada directamente en la base de datos ubicada en el departamento de informática de la institución, lo que mejorara la comunicación entre los departamentos. Este proyecto se construyó siguiendo el Proceso Unificado de Desarrollo de Software, y el Lenguaje de Modelado UML. Y para la construcción del sistema se utilizo el lenguaje de programación java, contenido en el entorno de programación Easyclipse y finalmente como sistema manejador de base de datos se utilizo PostgreSQL.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

CONTRIBUIDORES:

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Vásquez, Damelis	ROL	CA	AS X	TU	JU
	CVLAC:	8.304.463			
	E_MAIL				
	E_MAIL				
Veracierta T., Gabriela M.	ROL	CA	AS	TU X	JU
	CVLAC:	14.616.683			
	E_MAIL	gabrielaveracierta@hotmail.com			
	E_MAIL				
Rodríguez M., Rhonald E.	ROL	CA	AS	TU	JU X
	CVLAC:	14.077.185			
	E_MAIL	rhoen@hotmail.com			
	E_MAIL				
Dorta B., Pedro J.	ROL	CA	AS	TU	JU X
	CVLAC:	12.914.617			
	E_MAIL	dortap@hotmail.com			
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2010	08	12
AÑO	MES	DÍA

LENGUAJE. SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

ARCHIVO (S):

NOMBRE DE ARCHIVO	TIPO MIME
TESISDesarrolloSoftware.doc	application/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H I J K L M N
O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t u v w x y
z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPACIAL: Ministerio del Poder Popular para las Obras Públicas y
Vivienda - Estado Anzoátegui (OPCIONAL)

TEMPORAL: _____(OPCIONAL)

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

Ingeniero en Computación

NIVEL ASOCIADO CON EL TRABAJO:

Pregrado

ÁREA DE ESTUDIO:

Departamento de Computación y Sistemas

INSTITUCIÓN:

Universidad de Oriente, Núcleo de Anzoátegui

METADATOS PARA TRABAJOS DE GRADO, TESIS Y ASCENSO:

DERECHOS

De acuerdo al artículo 44 del reglamento de trabajo de grado

“Los trabajos de grado son de exclusiva propiedad de la Universidad y sólo podrán ser utilizados para otros fines con el conocimiento del Consejo de Núcleo respectivo, quién lo participará al Consejo Universitario”

Rodríguez, Dima

AUTOR

Veracierta, Gabriela

TUTOR

Rodríguez, Rhonald

JURADO

Dorta, Pedro

JURADO

POR LA SUBCOMISION DE TESIS