

APLICACIÓN DE LA ARQUITECTURA DE SOFTWARE BASADO EN COMPONENTES
Caso de estudio: prototipo para la gestión administrativa del acueducto rural Ayacucho La
Buitrera, Palmira

Derlyn Johanna Lourido Quintero
201356388
derlyn.lourido@correounivalle.edu.co

Hernán Darío González Patiño
201356430
hernan.gonzalez@correounivalle.edu.co

Director:
Antonio José Vélez Quintero, Ing.
antonio.velez@correounivalle.edu.co

Universidad del Valle
Sede Tuluá
Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación
Ingeniería de Sistemas
Octubre 10 de 2016

Agradecimientos

Agradezco a los que estuvieron conmigo en la realización de este proyecto, a mis padres, amigos, familiares y compañeros. Gracias a todos por sus consejos y colaboraciones.

Derlyn J. Lourido Quintero.

Agradezco a cada uno de los docentes y compañeros que me ayudaron a crecer como persona y adquirir nuevos conocimientos cada día, muchas gracias al Profesor Antonio Vélez por su acompañamiento en todo este proceso desde la Tecnología hasta la culminación de la Ingeniería en sistemas y también a los profesores que apoyaron esta propuesta de trabajo de grado desde el momento que se presentó como anteproyecto, Profesora Diana Lorena Velandia, Profesor Carlos Andrés Delgado y Profesor Julián Rodas Valverde.

Agradezco a mi padres que siempre estuvieron apoyándome a culminar mis estudios profesionales, haciendo sacrificios, mis hermanas Valentina y Angélica que nunca dejaron caer mis ánimos sin importar los momentos difíciles, a Grethel por acompañarme en este camino que fue un poco difícil, y mi perrunos Dante, Manolo, Ati y en especial a Gaia que siempre esperaba mi llegada y se quedaba conmigo en esas noches eternas besos a ella en el cielo.

Por ultimo Gracias a la Universidad Del Valle por prestarme sus herramientas para poder hacer posible la culminación de esta meta.

Hernán Darío González Patiño.

TABLA DE CONTENIDO

RESUMEN	0
ABSTRACT	0
INTRODUCCIÓN	1
PLANTEAMIENTO DEL PROBLEMA	2
Antecedente del Caso de Estudio	5
Formulación de la pregunta	6
ESTADO DEL ARTE	7
OBJETIVOS	9
Objetivo General	9
Objetivos Específicos	9
MARCO REFERENCIAL	10
INGENIERÍA DEL SOFTWARE	10
¿QUÉ ES LA ARQUITECTURA DE SOFTWARE?	11
¿Por qué es importante?	12
ESTILOS Y PATRONES ARQUITECTÓNICOS	14
LENGUAJES DE DESCRIPCIÓN DE ARQUITECTURA (ADL)	16
Modelo UML 4+1	16
Arquitectura y UML	18
ARQUITECTURA DE SOFTWARE BASADO EN COMPONENTES	18
COMPONENTE DE SOFTWARE	19
RELACIÓN ENTRE COMPONENTES Y OBJETOS	20
ESTÁNDARES DE COMPONENTES MÁS COMUNES	21
COM (Component Object Model)	21
EJB (Enterprise JavaBeans)	22
RUP: Rational Unified Process - Definición	22
Fases	23
Disciplinas	24
Artefactos	25

Actividades	25
Roles	26
UML: UNIFIED MODELING LANGUAGE	26
Organización de la superestructura	26
ACTIVIDADES PARA LA DESCRIPCION DE LA ARQUITECTURA DE SOFTWARE BASADA EN COMPONENTES	33
1. Gestionar el alcance del sistema	34
1.1. Priorizar los casos de uso y escenarios	34
2. Definir una arquitectura candidata	36
Punto crítico 1: Pensar en componentes	37
2.1. Análisis arquitectónico	39
3. Refinar la arquitectura	40
3.1. Identificar mecanismos de diseño	40
3.2. Identificar elementos de diseño	41
3.3. Incorporar elementos de diseño existentes	41
3.4. Describir la arquitectura en tiempo de ejecución	41
3.5. Describir la distribución	41
Punto crítico 2: Problemas típicos del DSBC	42
Documento de Arquitectura de Software (Prototipo)	43
STAKEHOLDERS	51
METODOLOGÍA	52
RESULTADOS OBTENIDOS	53
Comparación con otros desarrollos realizado por los estudiantes.	55
CONCLUSIONES	57
REFERENCIA BIBLIOGRÁFICA	58

ÍNDICE DE FIGURAS

Figura 1 Proceso de definición de la arquitectura de software	4
Figura 2 Presentación típica, pero no informativa, de una arquitectura de software	12
Figura 3 Objetivos del Negocio, Usuario y del Sistema.....	13
Figura 4 Vistas Modelo 4 + 1	17
Figura 5 Elaboración del diseño de un Componente	21
Figura 6 Dimensiones de tiempo y contenido de RUP	23
Figura 7 Estructura diagramas UML 2.0	27
Figura 8 Diagrama de clase	28
Figura 9 Diagrama de paquetes.....	28
Figura 10 Diagrama de Despliegue.....	29
Figura 11 Diagrama de Componentes	30
Figura 12 Diagrama de Actividad.....	31
Figura 13 Casos de Uso.....	32
Figura 14 Diagrama de Secuencia	33
Figura 15 Gestionar el Alcance del Sistema.....	36
Figura 16 Ciclo de vida basado en componentes	38
Figura 17 Definir arquitectura candidata	40
Figura 18 Refinar la arquitectura.....	42
Figura 19 Captura pantalla Inicio aplicación.....	53
Figura 20 Captura Pantalla Menús	54
Figura 21 Captura pantalla Facturación.....	54
Figura 22 Estructura prototipo Ayacucho La Buitrera - Palmira	55

ÍNDICE DE TABLAS

Tabla 1 Diagramas UML para describir la arquitectura	18
Tabla 2 Comparación con otros proyectos realizados	55

RESUMEN

Este trabajo de grado se fundamenta en la adquisición de nuevos conocimientos orientados a la arquitectura de software basado en componentes, sus características, actividades, beneficios e inconvenientes. Por consiguiente, se aplicaron dichos conocimientos en la elaboración de un prototipo, que busca dar solución a las necesidades del acueducto rural Ayacucho descritas en el caso de estudio.

ABSTRACT

The upgrade for this degree work is to have new knowledge for software architecture looking for their constituents, features, activity, advantage and disadvantages. Therefore, such knowledge applied in the elaboration of a prototype. This will be used to solve neediness to the Ayacucho's rural aqueduct. The needs were described in the study case.

INTRODUCCIÓN

En este proyecto se explican los principios fundamentales de la arquitectura de software basada en componentes, los métodos y lineamientos que se deben tener en cuenta para abordar un problema y solucionarlo con base a los criterios de esta arquitectura.

Se abordan las principales actividades y tareas, que se llevan a cabo durante el desarrollo del documento de arquitectura de software basada en componentes y se identifican puntos críticos presentes durante la realización de estas.

El caso de estudio, *“prototipo para la gestión administrativa del acueducto rural Ayacucho, La Buitrera – Palmira”*, fue seleccionado teniendo en cuenta la dinámica de la organización, ya que este cuenta con elementos comunes a otras organizaciones donde lo implementado puede usarse de igual forma o adaptarse según las necesidades.

PLANTEAMIENTO DEL PROBLEMA

En el año 1968, durante la primera conferencia de la OTAN¹ sobre desarrollo de software se acuñó el término “**Crisis del software**” [1]. El término expresa las dificultades del desarrollo frente al rápido crecimiento de la demanda, la complejidad de los problemas a ser resueltos y de la inexistencia de técnicas establecidas para el desarrollo de sistemas que funcionarán adecuadamente o pudieran ser validados.

Según Sommerville [2], los métodos, metodologías y técnicas en ingeniería de software actuales, han permitido que la construcción de sistemas grandes y complejos sea mejor, pero aun así se siguen presentando proyectos que se retrasan, sobrepasan del presupuesto o presentan falencias al satisfacer las necesidades del cliente.

Muchos estudiantes y profesionales no son conscientes de los métodos modernos, de las principales tendencias, tanto técnicas como metodológicas. Santiago Urrizola, director de Flux IT, comenta que *“los cambios actuales son extremos por lo que no se puede tener una estructura rígida, dogmática, que no se pueda cambiar, que no mute de acuerdo a lo que el negocio demande”* [3].

Ahora, se sabe que no hay un enfoque «ideal» en la ingeniería del software. La amplia diversidad de diferentes tipos de sistemas y organizaciones que existen, conlleva a la existencia de una diversidad de enfoques de desarrollo de software. Actualmente la demanda de los negocios es poder escalar no solamente en cantidad de usuarios o transacciones, sino poder escalar en funcionalidad de manera ágil y dinámica, pero sobre todo, económica y con bajo riesgo [3].

Uno de los enfoques que está tomando fuerza es el orientado a la reutilización, puesto que en la mayoría de los proyectos de software existe algo de este paradigma. Según Sommerville [2], la reutilización sucede generalmente cuando las personas que trabajan en el proyecto conocen diseños o códigos similares al requerido. Los buscan, los modifican según las necesidades y los incorporan en el sistema. Sametinger [4], describe algunos indicadores sobre la reutilización:

- Entre el 40 y 60% del código fuente de una aplicación es reutilizable en otra similar.
- Aproximadamente el 60% del diseño y del código de aplicaciones administrativas es reutilizable.
- Aproximadamente el 75% de las funciones son comunes a más de un programa.
- Sólo el 15% del código encontrado en muchos sistemas es único y novedoso a una aplicación específica. El rango general de uso recurrente potencial está entre el 15% y el 85%.

¹ OTAN: Organización del Tratado del Atlántico Norte. También denominada Alianza del Atlántico.

La ingeniería de software basada en componentes es un acercamiento a la reutilización. Esta técnica permite tomar partes del sistema ya existentes, permitiendo reutilizar piezas de código pre-elaborado dando beneficios en la mejora de la calidad, reducción del ciclo de desarrollo y tener un mayor retorno sobre la inversión [2]. El proceso de desarrollo del sistema se centra en la integración de esas partes más que desarrollarlas desde el inicio; esta ingeniería adopta la filosofía de “compra, no hagas” propuesta por Fred Brooks y otros.

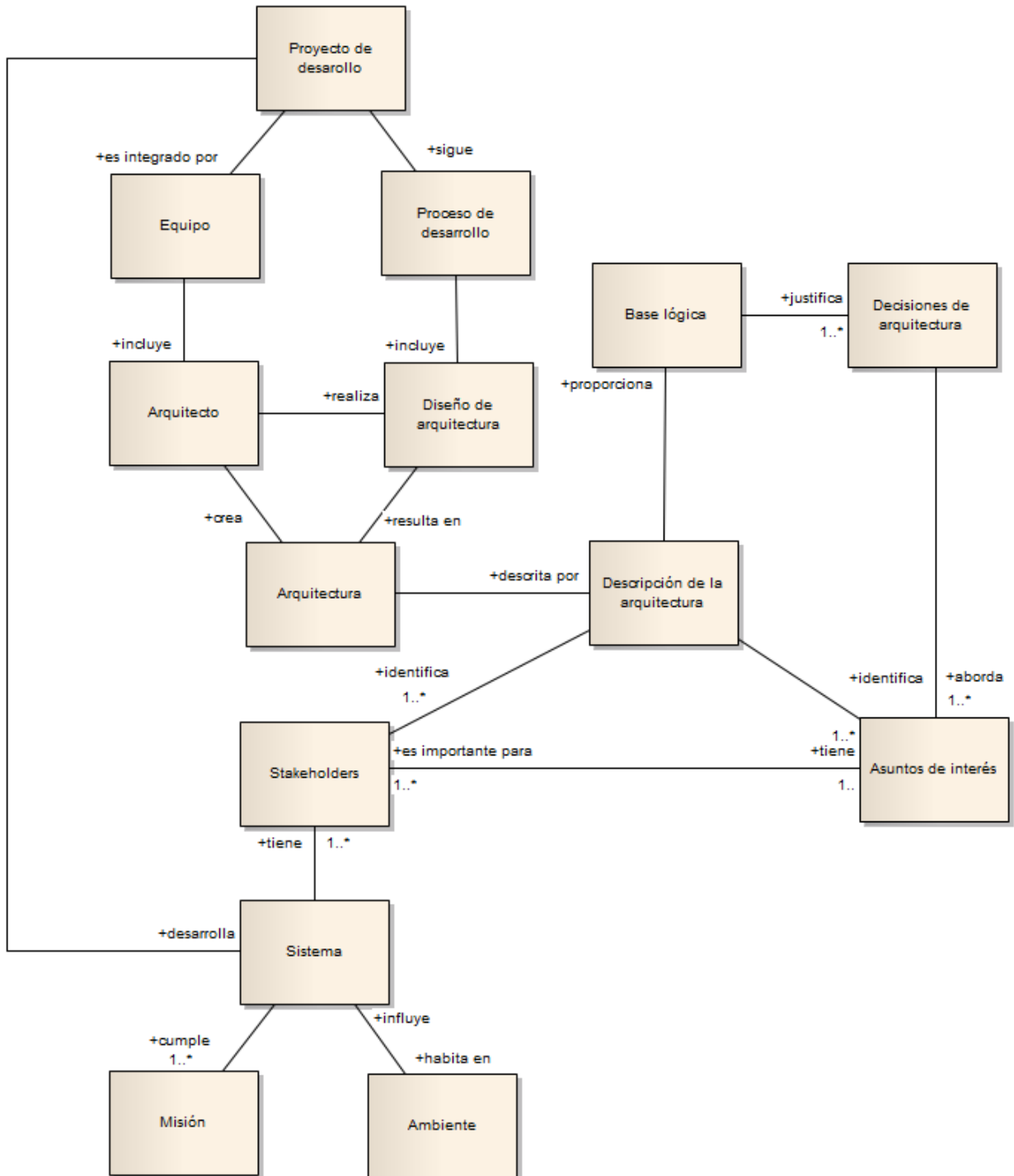
En el ámbito de la ingeniería de software cada vez es más común escuchar el término “arquitectura de software”. Este concepto tiende a ser malentendido y la falta de comprensión de sus principios, frecuentemente repercute de manera negativa en la construcción de sistemas de software.

Según Martin Fowler, la arquitectura de software es una comprensión compartida del diseño del sistema. Esta comprensión incluye cómo el sistema se divide en componentes y cómo los componentes interactúan a través de interfaces. Es una estructuración del sistema que, idealmente, se debe realizar en etapas tempranas del desarrollo. El no crear este diseño en etapas tempranas puede limitar severamente que el producto final satisfaga las necesidades de los clientes [5].

“Dinos lo que es importante” La arquitectura es sobre las cosas importantes. Sea lo que sea.

Con el fin de ilustrar los elementos que conforman el diseño de la arquitectura de software, se muestra en la Figura 1: Proceso de definición de la arquitectura de software, un meta-modelo de términos relacionados con el diseño de la arquitectura del software [19]. Esta figura muestra los elementos que son importantes únicamente para el diseño de la arquitectura de software, se omiten detalles sobre otros roles y actividades que no son realizadas por el arquitecto de software.

Figura 1 Proceso de definición de la arquitectura de software



Tomado de: http://danimaniarqsoft.com/?page_id=193

De la anterior figura podemos decir lo siguiente:

- Un sistema tiene una arquitectura, aun cuando no se describe
- Un sistema cumple con una o más misiones.
- Un sistema tiene uno o más *stakeholders*.
- Un sistema habita en un ambiente (entorno).
- Un ambiente influye en un sistema.
- Una arquitectura es descrita por una descripción de arquitectura.
- Una descripción de arquitectura identifica uno o más stakeholders.
- Una descripción de arquitectura identifica uno o más asuntos de intereses.
- Una descripción de arquitectura provee una base lógica.
- Un *stakeholder* tiene uno o más asuntos de interés.
- Un asunto de interés es importante para uno o más stakeholders.
- Un proyecto de desarrollo es integrado por un equipo.
- Un proyecto de desarrollo sigue un proceso de desarrollo.
- Un proyecto de desarrollo desarrolla un sistema.
- Un proceso de desarrollo incluye un diseño de arquitectura.
- Un equipo incluye a un arquitecto.
- El arquitecto realiza el diseño de arquitectura.
- El arquitecto crea una arquitectura.
- El diseño de arquitectura resulta en una arquitectura.
- La base lógica justifica una o más decisiones de arquitectura.
- Una decisión de arquitectura aborda uno o más asuntos de interés.

Antecedente del Caso de Estudio

La gran riqueza que tiene el corregimiento La Buitrera del municipio de Palmira es el agua: ríos, quebradas, acequias y chorreras dan cuenta de ésta gran riqueza natural generadora de vida.

Desde la Asociación del acueducto Ayacucho La Buitrera se está haciendo un importante trabajo para mejorar las condiciones de potabilidad del agua mediante un sistema de cloración con gas cloro, con el cual se potabiliza el agua las 24 horas del día durante los 365 días del año, colocando éste acueducto a la vanguardia de los acueductos rurales del Valle del Cauca como ejemplo y modelo administrativo.

El acueducto rural Ayacucho La Buitrera desea automatizar los procesos administrativos que este realiza. Muchas actividades son registradas en formularios de papel u hojas de cálculo y la obtención de datos estadísticos consume gran cantidad de recursos.

Cada vez que un suscriptor realiza un pago en la caja del acueducto se genera un **recibo de pago** por el valor aportado; se debe tener en cuenta que el suscriptor puede realizar abonos por valores inferiores o superiores al saldo actual de su cuenta. Algunos suscriptores realizan sus pagos en el a través de consignaciones en entidades financieras.

El acueducto no cuenta con micro-medidores que permiten medir la cantidad de agua consumida por los suscriptores por lo cual han definido diferentes tarifas para el cobro del servicio, los cuales depende de la actividad económica del predio.

La generación de los recibos de cobro se realiza cada dos meses. Se debe tener en cuenta que un suscriptor puede tener varias acometidas y se pueden generar varios recibos para la misma acometida (este caso se presenta cuando en el predio existen varias casas habitadas por diferentes familias y el suscriptor lo solicita).

Se registra diariamente el flujo de entrada y salida de efectivo que se presenta en la caja teniendo en cuenta el detalle de la operación.

Los procesos contables a tener en cuenta son:

- **Libro diario:** es un libro contable donde se recogen, día a día, los hechos económicos de una empresa. La anotación de un hecho económico en el **Libro Diario** se llama asiento; es decir en él se registran todas las transacciones realizadas por una empresa [13].
- **Cartera:** La cartera son las deudas que los clientes tienen con la empresa, y están originadas en las ventas a crédito, en este caso se genera por el atraso en el pago del servicio prestado. Lo que busca la provisión de cartera, es precisamente prever y reconocer ese riesgo constante de perder parte de lo vendido a crédito, de no poder recuperar esa cartera y no poder cobrar esas deudas [14].

Formulación de la pregunta

¿Cuáles son las actividades / tareas que se deben tener en cuenta cuando se desea desarrollar un producto de software usando la arquitectura basada en componentes frente a uno que no la cumple?

Replantear la pregunta

ESTADO DEL ARTE

A continuación se mencionan algunas investigaciones en el área de la ingeniería de software basado en componentes.

Use of Component-Based Software Architectures in Industrial Control Systems: es un documento realizado por Frank Lüders donde expone el proceso de adaptar la arquitectura de software basado en componentes para el desarrollo de sistemas de control industrial.

Esta tesis investiga los beneficios y los problemas relacionados con el uso de arquitecturas en sistemas de control industrial, que son sistemas basados en computadoras que controlan los procesos físicos y equipos. La investigación se lleva a cabo principalmente a través de un estudio a una empresa global de desarrollo industrial para una nueva generación de sistemas de control, destinados a sustituir varios sistemas existentes. Para aprovechar sus recursos de desarrollo global y la competencia de los diferentes centros de desarrollo, la empresa decidió adoptar una arquitectura de software basada en componentes que permite cierta funcionalidad para desarrollar componentes de forma independiente [11].

Desarrollo de Software Basado en Componentes: Los autores de este texto explican conceptos claves para el desarrollo de software basado en componentes.

Este documento presenta los modelos, conceptos y mecanismos fundamentales sobre los que se apoya actualmente el desarrollo de aplicaciones software basado en componentes reutilizables. En primer lugar, las arquitecturas software y los marcos de trabajo intentan ofrecer soluciones de diseño desde el punto de vista estructural de las aplicaciones, y de las relaciones entre sus componentes. A otro nivel se encuentra la programación orientada a componentes, un paradigma que propugna la construcción de componentes reutilizables en entornos abiertos y distribuidos, con el objetivo de lograr un mercado global de software. Basados en ellos, los modelos y plataformas de componentes proporcionan los mecanismos adecuados para tratar la complejidad de los problemas que aparecen en los sistemas abiertos y distribuidos. Finalmente, se describen las dificultades que encuentran las metodologías tradicionales para construir aplicaciones en estos nuevos ambientes, y los retos a los que se enfrenta la Ingeniería del Software para poder hablar realmente de “Ingeniería del Software Basada en Componentes” [12].

Propuesta de vistas de la arquitectura de software: La presente investigación habla acerca de la importancia de contar con unas vistas de arquitectura de software bien definidas. Se da a conocer una propuesta de vistas de la arquitectura de software totalmente novedosa y que rompe con algunas de sus concepciones actuales. Se resalta la importancia de las vistas a la hora de desarrollar una arquitectura de software, definiéndose y propugnando cada una como una parte indispensable a la hora de desarrollar un proyecto [13].

Desarrollo de Agentes Software sobre una Arquitectura Basada en Componentes: El desarrollo de sistemas multi-agente se aborda desde los métodos, técnicas y herramientas que ofrece la

Ingeniería del Software Orientada a Agentes. Sin embargo, a pesar de la existencia de diversas metodologías y plataformas de desarrollo de agentes software, el trabajo del desarrollador se ve incrementado por una falta de arquitecturas de agentes flexibles. Actualmente las arquitecturas de agente proporcionadas por estas metodologías y plataformas no ofrecen la flexibilidad necesaria para desarrollar agentes software adaptable, poniendo poco énfasis en la reutilización. Este trabajo presenta una arquitectura composicional de agentes software en la que, aplicando la tecnología de componentes y el principio de separación de aspectos en el diseño de agentes software, la funcionalidad de los agentes se encuentra distribuida en componentes independientes y reutilizables.

En estos documentos se aborda con diferentes puntos de vista la ingeniería de software basado en componentes y sus procesos para llevar a cabo diferentes tipos de desarrollo. Los procesos de la ingeniería son extensos y complejos, por lo cual los autores recalcan los puntos más importantes de aquellos que describen en su respectivo trabajo.

Con este proyecto se pretende analizar los procesos para definir la arquitectura de software en el marco de Ingeniería de software basado en componentes y cómo esta afecta el desarrollo de un sistema. También se busca mostrar las actividades/tareas que se realizan de más durante el proceso de definir la arquitectura de software basado en componentes frente a otra que no la cumple.

OBJETIVOS

Objetivo General

Aplicar la arquitectura de software basada en componentes y sus implicaciones, como estrategia para diseñar y especificar la estructura de un sistema.

Objetivos Específicos

1. Describir que es una arquitectura de software basada en componentes, y qué se busca al definir una arquitectura para un sistema.
2. Identificar el proceso de definir la arquitectura de software basada en Componentes catalogando los puntos críticos.
3. Identificar los artefactos de entrada necesarios para el proceso de definir una arquitectura de software.
4. Realizar los artefactos de entrada y salida necesarios para el proceso de definir la arquitectura de software del prototipo, para el apoyo a la gestión administrativa del acueducto rural Ayacucho La Buitrera ubicado en el municipio de Palmira.
5. Aplicar la arquitectura de software basado en componentes al diseño y especificación de la estructura del prototipo, para el apoyo a la gestión administrativa del acueducto rural Ayacucho La Buitrera ubicado en el municipio de Palmira.

MARCO REFERENCIAL

INGENIERÍA DEL SOFTWARE

Para Sommerville, la ingeniería del software es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza [2]. Pero la definición dada por Fritz Bauer [17] sirve como base para el análisis:

[La ingeniería de software es] el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales.

Pressman, plantea algunas preguntas partiendo de la definición propuesta por Bauer [7]:

¿Cuáles son los “principios fundamentales de la ingeniería” que pueden aplicarse al desarrollo del software de computadora? ¿Cómo se desarrolla software “en forma económica” y que sea “confiable”? ¿Qué se requiere para crear programas de cómputo que trabajen con “eficiencia”, no en una sino en muchas “máquinas reales” diferentes?

En el libro *How to solve it*, escrito mucho antes de que las computadoras existieran, se describió la esencia de la solución de problemas, y según Pressman, la esencia de la práctica de la ingeniería de software [7]:

1. Entender el problema (Comunicación y análisis)
2. Planear la solución (modelado y diseño del software)
3. Ejecutar el plan (generación del código)
4. Examinar la exactitud del resultado (probar y asegurar la calidad)

Teniendo en cuenta la esencia de la práctica de la ingeniería de software, la arquitectura está embebida en el segundo paso **planear la solución**, ya que en esta se debe definir todo lo importante y crítico del sistema.

A continuación, se plantea la definición de arquitectura en la ingeniería civil, la cual se acerca mucho y da idea de qué es la arquitectura de software. Muchos de los conceptos de la ingeniería de software son traídos de la ingeniería civil, es por ello que esta definición nos acerca a un concepto paralelo.

Cuando se piensa en la arquitectura de una construcción, llegan a la mente muchos atributos distintos. En el nivel más sencillo, se considera la forma general de la estructura física. Pero, en realidad, la arquitectura es mucho más que eso. Es la manera en la que los distintos componentes del edificio se integran para formar un todo cohesivo. Es la forma en la que la construcción se adapta a su ambiente y se integra a los demás edificios en la vecindad. Es el grado en el que el edificio cumple

con su propósito y en el que satisface las necesidades del propietario. Es la sensación estética de la estructura —el efecto visual de la edificación— y el modo en el que se combinan texturas, colores y materiales para crear la fachada en el exterior y el “ambiente de vida” en el interior. Es los pequeños detalles: diseño de las lámparas, tipo de piso, color de las cortinas... la lista es casi interminable. Y, finalmente, es arte.

Son los “miles de decisiones, tanto grandes como pequeñas”. Algunas de éstas se toman en una etapa temprana del diseño y tienen un efecto profundo en todas las demás acciones. [7].

¿QUÉ ES LA ARQUITECTURA DE SOFTWARE?

No hay duda que el mundo cada vez se está volviendo más dependiente del software, por tal motivo, este debe proporcionar la capacidad requerida, ser de calidad suficiente, tener alta disponibilidad y entregarse a un precio aceptable. Todas estas características están influidas por la arquitectura de software.

Bass, Clements y Kazman [18] definen el término arquitectura de software como: *“ la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos.”*

La arquitectura no es el software operativo. Pressman [7] describe que es una representación que permite 1) analizar la efectividad del diseño para cumplir los requerimientos establecidos, 2) considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y 3) reducir los riesgos asociados con la construcción del software.

Teniendo en cuenta lo planteado por Sommerville [2], los grandes sistemas siempre se descomponen en subsistemas que proporcionan algún conjunto de servicios relacionados. Se llama diseño arquitectónico al proceso de diseño inicial que identifica estos subsistemas y establece un marco para el control y comunicación de estos. El resultado de este proceso de diseño es una descripción de la arquitectura del software.

El medio ambiente, o el contexto, determinan el entorno y las circunstancias del desarrollo, las condiciones de servicio, políticas y otras influencias propias de cada sistema. Kruchten [22] expone que el conjunto de decisiones importantes sobre la organización de un sistema de software, la selección de los elementos estructurales y sus interfaces por el cual el sistema se compone, junto con su comportamiento, cómo se especifica las colaboraciones entre esos elementos, la composición de estos elementos en subsistemas progresivamente más grandes, y el estilo arquitectónico que guía esta organización, es lo que compone la arquitectura de un software.

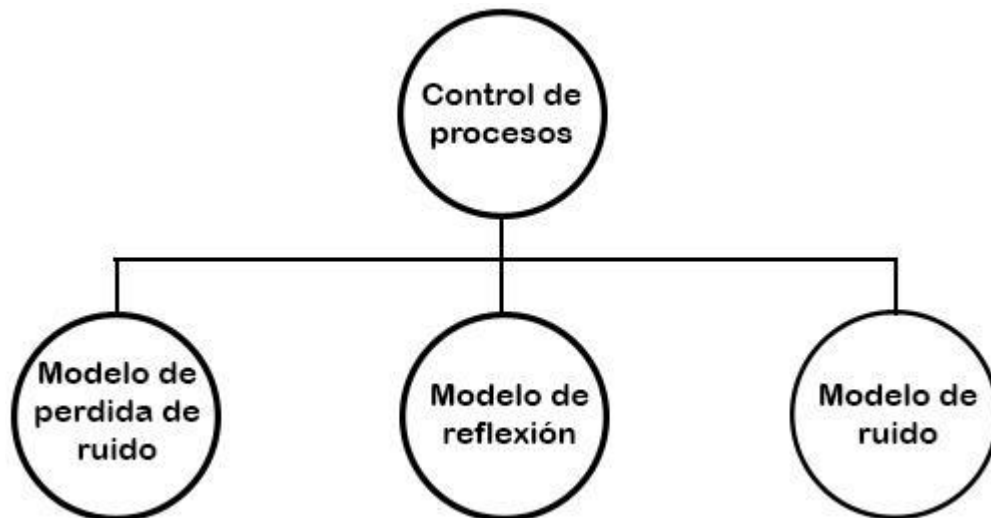
Por ello la definición de la ISO/IEC/IEEE 42010 describe que el diseño de la arquitectura es *“el proceso de concebir, definir, expresar, documentar, comunicar, certificar la correcta ejecución de mantener y mejorar una arquitectura a lo largo del ciclo de vida de un sistema”* [44].

Tomando en cuenta las descripciones anteriores para arquitectura de software la ISO/IEC/IEEE 42010 compacta la definición de esta como “*los conceptos fundamentales o las propiedades de un sistema en su entorno, plasmado en sus elementos, relaciones y principios de su diseño y evolución*” [44].

¿Por qué es importante?

Si se le preguntará a cualquier persona que describa qué es arquitectura, nueve de cada diez, va a hacer referencia a la estructura. Esto muy a menudo en relación a un edificio u obra civil. Aunque existen otras características tales como el comportamiento, la aptitud para el propósito e incluso la estética, la característica estructural es la más conocida y mencionada. Partiendo de esta suposición, se podría mencionar que la Figura 2 Presentación típica, pero no informativa, de una arquitectura de software, hace referencia a la arquitectura de un software.

Figura 2 Presentación típica, pero no informativa, de una arquitectura de software



Tomado de <http://www.ece.ubc.ca/~matei/EECE417/BASS/ch02lev1sec1.html>

Según Garlan y Shaw [8] se puede notar que estas condiciones no son suficientes. A partir de la figura 1 no se puede tener en cuenta ¿Qué hace cada elemento?, ¿Cómo se relaciona con los otros elementos?, ¿Depende de otros elementos? ¿Cuál es la naturaleza de los elementos? ¿Cuál es el significado de su separación? ¿Su ejecución se realiza en diferentes instantes? ¿Contienen programas, procesos, o ambos? ¿Son tareas, objetos, procesos, funciones? y esto sería por nombrar algunas preguntas.

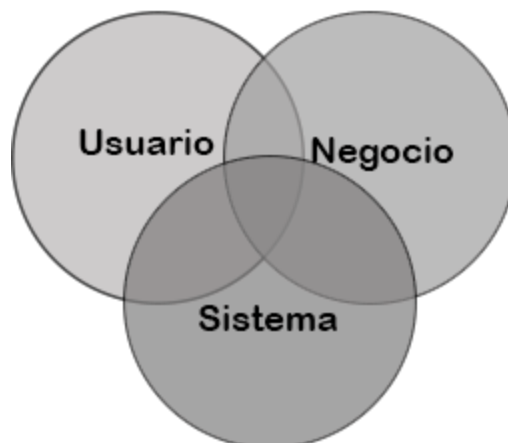
Para Pressman [7], un objetivo clave del negocio de cualquier arquitectura de software es permitir la adición de características sin gran esfuerzo, y en general, la capacidad de sostener los cambios y garantizar que la aplicación construida sobre esta arquitectura se mantenga a prueba a lo largo de su ciclo de vida.

Bass, Clements y Kazman [18] señalan tres ventajas de diseñar explícitamente y documentar la arquitectura del software:

1. **Comunicación con los stakeholders (todas las partes interesadas, véase pág. 44):** La arquitectura constituye una presentación de alto nivel del sistema que puede usarse como punto de discusión por varios stakeholders. Puede usarse como base para la comprensión mutua, la negociación, el consenso y la comunicación. Proporciona un lenguaje común en el que las diferentes preocupaciones se pueden expresar, negociar y resolverse en un nivel que es intelectualmente manejable, incluso para sistemas grandes y complejos.
2. **Análisis del sistema:** Hacer explícita la arquitectura del sistema en una etapa temprana del desarrollo requiere realizar algún análisis. Las decisiones de diseño arquitectónico tienen un gran efecto sobre si el sistema puede cumplir los requerimientos críticos, tales como rendimiento, fiabilidad y mantenibilidad.
3. **Reutilización a gran escala:** Un modelo de arquitectura del sistema es una descripción compacta y manejable de cómo se organiza y cómo inter-operan sus componentes. La arquitectura del sistema es a menudo la misma para sistemas con requerimientos similares y, por lo tanto, pueden soportar reutilización del software a gran escala. No sólo el código se puede volver a utilizar, también puede hacerse con los requisitos que dieron lugar a la arquitectura en primer lugar, así como la experiencia de la construcción de la arquitectura reutilizada. Cuando las decisiones arquitectónicas pueden ser reutilizados en múltiples sistemas, todas las consecuencias de decisiones tempranas que se acaban de describir también se transfieren.

Terreros [10], hace énfasis en que los sistemas deben ser diseñados con la consideración para el usuario, el sistema (la infraestructura de TI), y los objetivos de negocio. Para cada una de estas áreas, se debe esbozar escenarios clave e identificar los atributos de calidad importantes (por ejemplo, la fiabilidad o la escalabilidad) y áreas clave de la satisfacción y la insatisfacción, Figura 3: Objetivos del Negocio, Usuario y del Sistema.

Figura 3 Objetivos del Negocio, Usuario y del Sistema



Tomado de: <https://msdn.microsoft.com/en-us/library/ee658098.aspx>

La arquitectura se centra en cómo los principales elementos y componentes dentro de una aplicación son utilizados, o interactúan, con otros elementos y componentes más importantes dentro de la aplicación. Según Fowler [5], la arquitectura de software busca ser el puente entre los requerimientos del negocio y los requisitos técnicos, buscando la forma de aplicarlos al sistema mediante la comprensión de las necesidades de las partes implicadas.

El objetivo de la arquitectura es identificar los requisitos que afectan la estructura de la aplicación. Un buen diseño es suficientemente flexible para ser capaz de manejar los cambios que se producirán con el tiempo en hardware y tecnología de software, así como en escenarios y necesidades de los usuarios [2].

A continuación se mencionan algunas consideraciones a tener en cuenta cuando se piensa en la arquitectura de software según Terreros [10]:

- ¿Cómo van los usuarios a usar la aplicación?
- ¿Cómo se desplegará la aplicación en producción y en administración?
- ¿Cuáles son los requisitos de los atributos de calidad de la aplicación, tales como seguridad, rendimiento, simultaneidad, internacionalización, configuración, entre otros?
- ¿Cómo se puede diseñar la aplicación para ser flexible y fácil de mantener en el tiempo?
- ¿Cuáles son las tendencias arquitectónicas que pueden afectar su solicitud ahora o después de que se ha desplegado?

La arquitectura de un sistema software puede basarse en un modelo o estilo arquitectónico particular. Garlan y Shaw [8], describen que un estilo arquitectónico es un patrón de organización de un sistema tal como una arquitectura cliente-servidor o una arquitectura por capas. Es importante un conocimiento de los diferentes estilos arquitectónicos, sus aplicaciones, y sus ventajas e inconvenientes debido a que la mayoría de los sistemas grandes no utilizan un único estilo. Pueden diseñarse diferentes partes del sistema utilizando distintos estilos arquitectónicos. En algunos casos, la totalidad de la arquitectura del sistema puede ser una arquitectura compuesta creada mediante la combinación de diferentes estilos arquitectónicos.

ESTILOS Y PATRONES ARQUITECTÓNICOS

Pressman [7] describe que un estilo arquitectónico es una transformación que se impone al diseño de todo el sistema, cuyo objetivo es establecer una estructura para todos los componentes que lo comprenden.

Arquitecturas orientadas a objetos: *“Los componentes de un sistema incluyen datos y las operaciones que deben aplicarse para manipularlos. La comunicación y coordinación entre los componentes se consigue mediante la transmisión de mensajes.*

Los componentes de este estilo son los objetos, o más bien, instancias de los tipos de datos abstractos. Se basan sobre los principios de POO²: encapsulamiento, herencia y polimorfismo. Son asimismo las unidades de modelado, diseño e implementación, y los objetos y sus interacciones son el centro de las incumbencias en el diseño de la arquitectura y en la estructura de la aplicación.

Las interfaces están separadas de las implementaciones. En general la distribución de objetos es transparente, y no importa si los objetos son locales o remotos.

Restricciones: una interfaz pueda ser implementada por múltiples clases. Los componentes y objetos interactúan a través de invocaciones de funciones y procedimientos.” Somerville [2].

Mientras que un patrón (modelo) arquitectónico, como un estilo de arquitectura, impone la transformación del diseño de una arquitectura.

Cliente - Servidor: *“es un modelo de sistema en el que dicho sistema se organiza como un conjunto de servicios y servidores asociados, más unos clientes que acceden y usan los servicios. Los principales componentes de este modelo son:*

- 1. Un conjunto de servidores que ofrecen servicios a otros subsistemas. Ejemplos de servidores son servidores de impresoras que ofrecen servicios de impresión, servidores de ficheros que ofrecen servicios de gestión de ficheros y servidores de compilación, que ofrecen servicios de compilación de lenguajes de programación.*
- 2. Un conjunto de clientes que llaman a los servicios ofrecidos por los servidores. Estos son normalmente subsistemas en sí mismos. Puede haber varias instancias de un programa cliente ejecutándose concurrentemente.*
- 3. Una red que permite a los clientes acceder a estos servicios. Esto no es estrictamente necesario ya que los clientes y los servidores podrían ejecutarse sobre una única máquina. En la práctica, sin embargo, la mayoría de los sistemas cliente-servidor se implementan como sistemas distribuidos.*

La ventaja más importante del modelo cliente-servidor es que es una arquitectura distribuida. Se puede hacer un uso efectivo de los sistemas en red con muchos procesadores distribuidos. Es fácil añadir un nuevo servidor e integrarlo con el resto del sistema o actualizar los servidores de forma transparente sin afectar al resto del sistema.” Somerville [2].

Modelo – Vista – Controlador (MVC): *El modelo–vista–controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las*

² POO: Programación Orientada a Objetos

comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.^{1 2} Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento [49].

Sin embargo, un patrón difiere de un estilo en varias formas fundamentales:

1. *El alcance del patrón es menos amplio y se centra en un aspecto de la arquitectura más que en el total de ésta*
2. *Un patrón impone una regla a la arquitectura, describe la manera en la que el software manejará ciertos aspectos de su funcionalidad en el nivel de la infraestructura*
3. *Los patrones arquitectónicos tienden a abocarse a aspectos específicos del comportamiento en el contexto de la arquitectura (por ejemplo, cómo manejarán la sincronización o las interrupciones las aplicaciones en tiempo real).* [7]

Los patrones se utilizan junto con un estilo arquitectónico para dar forma a la estructura general de un sistema.

LENGUAJES DE DESCRIPCIÓN DE ARQUITECTURA (ADL)

Una vez que el arquitecto de software, tras conocer el requerimiento, se decide a delinear su estrategia y a articular los patrones que se le ofrecen, se supone que debería expresar las características de su sistema, o en otras palabras, modelarlo, aplicando una convención gráfica o algún lenguaje avanzado de alto nivel de abstracción.

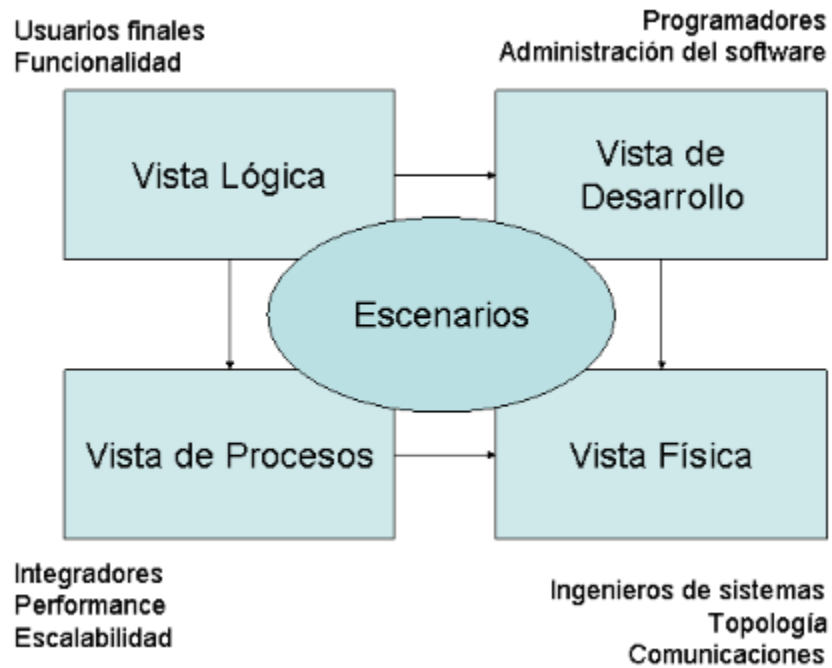
*El **ADL** (Architecture Description Language) es un lenguaje que ayuda a abstraer, conceptualizar, modelar y concretar de una manera más amplia y óptima un software, esto lo hace a través de las herramientas, los modelos y los patrones para aplicar estas en su desarrollo o construcción.* Reynoso y Kicillof [23].

No existe hasta hoy una definición consensuada y unívoca de ADL, pero comúnmente se acepta que un ADL debe proporcionar un modelo explícito de componentes, conectores y sus respectivas configuraciones. Se estima deseable, además, que un ADL suministre soporte de herramientas para el desarrollo de soluciones basadas en arquitectura y su posterior evolución. Canal [24]

Modelo UML 4+1

Es un modelo de vistas diseñado por el profesor Philippe Kruchten [25], que encaja con el estándar IEEE 1471-2000, ahora conocido como ISO/IEC/IEEE 42010:2011, “*Recommended Practice for Architecture Description of Software-Intensive Systems*” utilizado para describir la arquitectura de un sistema de software intensivo basado en el uso de múltiples puntos de vista, Figura 4: Vistas Modelo 4+1.

Figura 4 Vistas Modelo 4 + 1



Tomado de: <http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf>

Vista Lógica: describe el modelo de objetos del diseño cuando se usa un método orientado a objetos. Esta vista se apoya principalmente en los requisitos funcionales - la funcionalidad que el sistema proporcionará a los usuarios finales, es decir, se ha de representar lo que el sistema debe hacer, las opciones y servicios que ofrece.

Aquí se aplican los principios de abstracción, encapsulamiento y herencia. Esta descomposición no solo se hace para potenciar el análisis funcional, sino también sirve para identificar mecanismos y elementos de diseño comunes a diversas partes del sistema.

Vista de Procesos: toma en cuenta algunos requisitos no funcionales tales como el desempeño y la disponibilidad. Se enfoca en asuntos de concurrencia y distribución, integridad del sistema, de tolerancia a fallas. La vista de procesos también especifica en cuál hilo de control se ejecuta efectivamente una operación de una clase identificada en la vista lógica.

Vista de Desarrollo: se centra en la organización real de los módulos de software en el ambiente de desarrollo del software. El software se empaqueta en partes pequeñas – bibliotecas de programas o subsistemas– que pueden ser desarrollados por uno o un grupo pequeño de desarrolladores.

La vista de desarrollo tiene en cuenta los requisitos internos relativos a la facilidad de desarrollo, administración del software, reutilización y elementos comunes, y restricciones

impuestas por las herramientas o el lenguaje de programación que se use. También apoya la asignación de requisitos y trabajo al equipo de desarrollo, y apoya la evaluación de costos, la planificación, el monitoreo de progreso del proyecto, y también como base para analizar re-uso, portabilidad y seguridad. Es la base para establecer una línea de productos.

Vista Física: toma en cuenta primeramente los requisitos no funcionales del sistema tales como la disponibilidad, confiabilidad (tolerancia a fallas), rendimiento y escalabilidad. Se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema.

Vista Escenarios: Esta vista es representada por los casos de uso y tiene la función de unir y relacionar las otras 4 vistas, esto quiere decir que desde un caso de uso se puede ver cómo se van ligando las otras 4 vistas, con lo que se tendrá una trazabilidad de componentes, clases, equipos, paquetes, etc., para realizar cada caso de uso.

Arquitectura y UML

A continuación se describe los diagramas UML usados para describir la arquitectura del software y su relación con las vistas propuestas en el modelo 4+1 del profesor Philippe Kruchten [25]

Tabla 1 Diagramas UML para describir la arquitectura

Vista	Diagrama UML
Escenarios	Casos de uso
Lógica	Clase y Colaboración
Desarrollo	Componentes y Paquetes
Física	Despliegue
Procesos	Actividad, Estados y Secuencia

ARQUITECTURA DE SOFTWARE BASADO EN COMPONENTES

La arquitectura basada en componentes es una rama de la Ingeniería de software que se enfoca en la descomposición del software en componentes funcionales. Esta descomposición permite convertir componentes pre-existentes en piezas más grandes de software.

Cuando se habla de los componentes se basa en la teoría, la cual consiste en diseñar, construir y documentar partes de software que puedan ser probadas y re-utilizadas como unidades independientes, de manera tal que su integración a sistemas más amplios sea posible. Los componentes de software se construyen mediante lenguajes de programación que tienen una gramática de integración definida explícitamente por reglas bien formadas semánticas y sintácticamente propias de dichos lenguajes, lo cual permite la construcción de interfaces para un uso futuro de los mismos. [30, 32].

Algunas definiciones de componentes de software importantes son:

- Los componentes de software son unidades binarias de producción independiente que interactúan para formar un sistema funcional [33]
- *“Un componente es una implementación de software que puede ser ejecutada sobre un dispositivo lógico o físico. Un componente implementa una o más interfaces que deben obedecer ciertas reglas, de manera que componentes desarrollados independientemente puedan interactuar de manera predecible”* [35].

Un componente puede tomar la forma de un archivo ejecutable o una biblioteca dinámica, que usualmente cobra vida a través de objetos, pero no es este un requisito indispensable. De hecho, los primeros componentes conocidos (aunque en su momento no se los haya definido así) fueron las bibliotecas de procedimientos [34].

Algunas características de los componentes son:

- Tiene un conjunto de operaciones que puede realizar, ocultando detalles de su implementación.
- Sus servicios son inmutables, es decir, los servicios que ofrece un componente a través de sus interfaces no deben variar.
- La implementación física de estos servicios pueden ser modificadas, pero no deben afectar la interfaz a la que está asociado.
- Debe tener una documentación adecuada que facilite:
 - La recuperación del componente desde un repositorio,
 - La evaluación del componente,
 - La adaptación a un nuevo ambiente.
 - Su integración con otros componentes.
- Debe ser reemplazable por una nueva versión, o por otro componente que proporcione los mismos servicios. [37,34,35]

COMPONENTE DE SOFTWARE

La *Especificación OMG del Lenguaje de Modelado Unificado* [21] define un componente como *“una parte modular, desplegable y sustituible de un sistema, que incluye la implantación y expone un conjunto de interfaces”*. Los componentes forman la arquitectura del software y,

en consecuencia, juegan un papel en el logro de los objetivos y de los requerimientos del sistema que se va a construir. Como los componentes se encuentran en la arquitectura del software, deben comunicarse y colaborar con otros componentes y con entidades (otros sistemas, dispositivos, personas, etc.) que existen fuera de las fronteras del software [7].

RELACIÓN ENTRE COMPONENTES Y OBJETOS

Se ha relacionado mucho los componentes con objetos, pero la programación orientada a objetos se ha mostrado insuficiente al tratar de aplicar sus técnicas para el desarrollo de aplicaciones basadas en sistemas de componentes. En particular se ha observado que no permite expresar claramente la distinción entre los aspectos composicionales y los meramente computacionales de las aplicaciones, y que hace prevalecer la visión de objeto sobre la de componente, estos últimos como unidades de composición independientemente de las aplicaciones. Por este motivo, surge la llamada programación orientada a componentes como una extensión natural de la orientación a objetos [41,42].

De hecho, la orientación a objetos no utiliza los objetos como unidad elemental del software, al menos durante el desarrollo de las aplicaciones, sino que esta unidad la constituyen las clases, siendo los objetos instancias de una clase y estando su existencia ligada normalmente a un programa en ejecución.

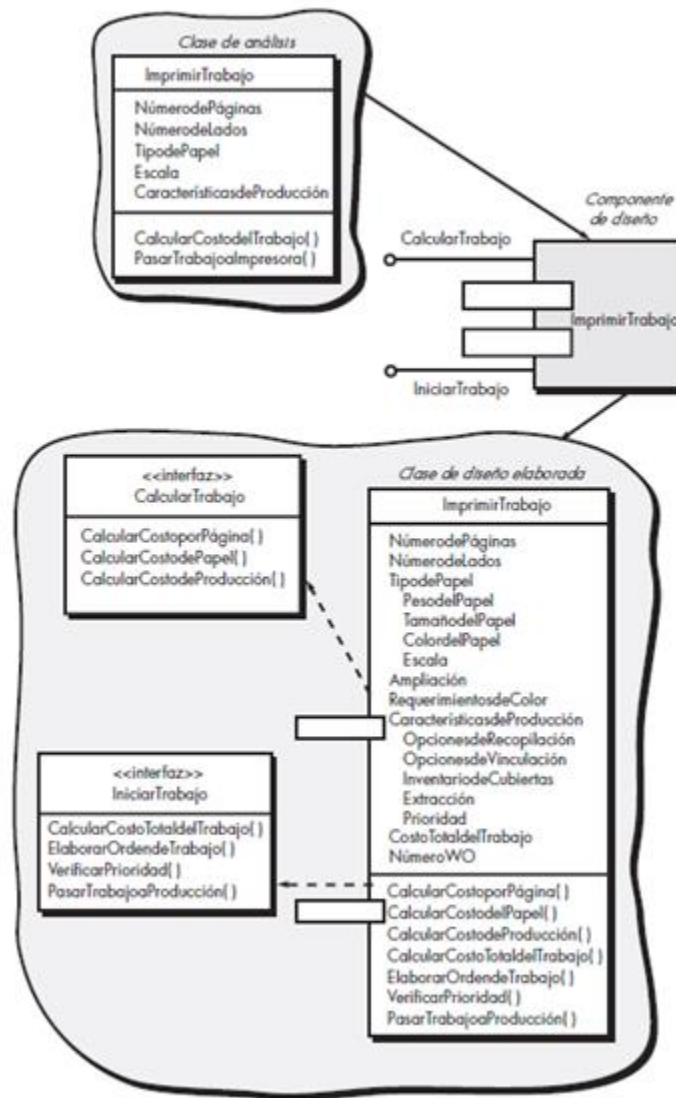
Otra diferencia fundamental entre las tecnologías de componentes y de objetos radica en los diferentes mecanismos que ambas proponen para la composición y reutilización, y que están relacionados con la visibilidad de la implementación de los elementos a través de su interfaz.

Según Szyperski, la reutilización en los sistemas de componentes debe basarse exclusivamente en la composición dinámica de interfaces y en el reenvío de mensajes entre objetos. Los componentes son cajas negras que ocultan totalmente los detalles de su implementación, siendo su interfaz lo único visible para sus clientes. La reutilización consiste, de esta forma, en la implementación de un componente que proporciona la interfaz requerida por otro o el uso de un componente en función de la interfaz que este ofrece [43].

Al momento de elaborar un componente según Pressman, deben elaborarse los detalles del componente para que den información suficiente que guíe la Implementación [7].

La clase de análisis original se lleva a cabo para dar cuerpo a todos los atributos y operaciones requeridos para implementar la clase así como el componente. En la parte inferior derecha de la Figura 5: Elaboración del diseño de un componente, la clase de diseño elaborada contiene información más detallada de los atributos, así como una descripción amplia de las operaciones que se necesitan para implementar el componente. Las interfaces implican comunicación y colaboración con otros componentes (que no se muestran) [7].

Figura 5 Elaboración del diseño de un Componente



Tomado de Ingeniería de software: un enfoque práctico [7].

ESTÁNDARES DE COMPONENTES MÁS COMUNES

COM (Component Object Model)

COM es una manera de implementar objetos neutrales con respecto al lenguaje, de manera que pueden ser usados en entornos distintos de aquel en que fueron creados, a través de fronteras entre máquinas. Para componentes bien creados, COM permite la reutilización de objetos sin conocimiento de su implementación interna, porque fuerza a los implementadores de componentes a proveer interfaces bien definidas que están separados de la implementación. Las diferentes semánticas de reserva de memoria están

acomodadas haciendo a los objetos responsables de su propia creación y destrucción por medio del contador de referencias. [39]

La familia de tecnologías COM incluye COM +, COM distribuido (DCOM) y controles ActiveX.

EJB (Enterprise JavaBeans)

Originado por Sun Microsystems, Enterprise JavaBeans es equivalente a Microsoft Component Object Model / Distributed Component Object Model, pero, al igual que todas las arquitecturas basadas en Java, los programas se pueden implementar en diferentes sistemas operativos. Está basado en la tecnología JavaBeans la cual permite distribuir los componentes de programas a los clientes en una red. Este ofrece la ventaja de ser capaz de controlar los cambios en el servidor en lugar de tener que actualizar cada equipo con los nuevos componentes que se añadieron o se modificaron [38].

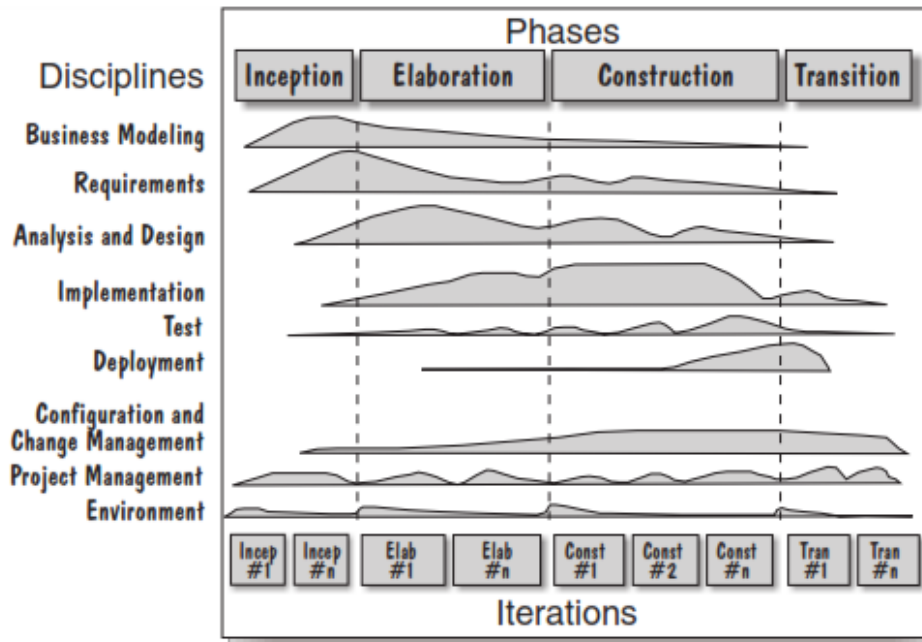
RUP: Rational Unified Process - Definición

“El RUP es un marco de proceso de desarrollo de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización de desarrollo” [47].

RUP es un marco de proceso adaptable, previsto para ser a la medida de las organizaciones de desarrollo y equipos de proyecto de software, estos seleccionarán los elementos del proceso que son apropiados para sus necesidades. Fue desarrollado por Rational Software y adquirido por IBM en 2003.

RUP puede ser descrito en términos de dos dimensiones: **tiempo** y **contenido**. En la Figura 6 Dimensiones de tiempo y contenido de RUP, el eje horizontal representa el tiempo y demuestra los aspectos del ciclo de vida del proceso. Esta dimensión se describe en términos de fases e iteraciones. El eje vertical representa el contenido y muestra las disciplinas, que agrupan lógicamente el contenido del proceso [47].

Figura 6 Dimensiones de tiempo y contenido de RUP



Tomado de: <http://homes.dcc.ufba.br/~luizromario/EngSoftII/IBM%20Rational%20Unified%20Process.pdf>

Fases

El RUP ha determinado un ciclo de vida del proyecto que consta de cuatro fases las cuales siguen una estrategia iterativa e incremental. La clave del proceso radica en las iteraciones de desarrollo que se encuentran en todas las fases. Además, cada fase tiene un objetivo clave y un hito que denota si el objetivo se llevó a cabo [47].

- **Inicio (Inception):** En esta fase se establece el modelo de negocio que incluye el contexto empresarial, factores de éxito (ingresos esperados, reconocimiento de mercado, etc.) y proyecciones financieras.
- **Elaboración (Elaboration):** El objetivo es mitigar los elementos de riesgo clave identificados por el análisis hasta el final de esta fase. La fase de elaboración es donde el proyecto empieza a tomar forma. En esta fase se realiza el análisis del dominio de problema y la arquitectura del proyecto obtiene su forma básica.
- **Construcción (Construction):** El objetivo principal es construir el sistema de software. En esta fase, el foco principal está en el desarrollo de componentes y otras características del sistema. Esta es la fase cuando ocurre la mayor parte de la codificación. En proyectos más grandes, varias iteraciones de construcción pueden ser desarrolladas en un esfuerzo por dividir los casos de uso en segmentos manejables, con el objetivo de producir prototipos demostrables.
- **Transición (Transition):** El objetivo principal es la 'transición' del sistema desarrollado a producción, lo que está disponible y es entendido por el usuario final. Las actividades de esta fase incluyen capacitación de usuarios finales, desarrollo y realización de pruebas

beta al sistema para validarlo contra las expectativas de los usuarios finales. El producto también se comprueba con el nivel de calidad en la fase inicial.

Disciplinas

- **Modelado del negocio (Business Modeling):** describe una visión de la organización en la que se implementará el sistema y cómo entonces utilizar esta visión como base para delinear el proceso, roles y responsabilidades [47].

Los propósitos que tiene el modelado del negocio son:

- Entender los problemas que la organización desea solucionar e identificar mejoras potenciales.
 - Medir el impacto del cambio organizacional.
 - Asegurar que clientes, usuarios finales, desarrolladores y los otros participantes tengan un entendimiento compartido del problema.
 - Derivar los requerimientos del sistema de software, necesarios para dar soporte a los objetivos de la organización.
 - Entender cómo el sistema a ser desarrollado entra dentro de la organización.
- **Requerimientos (Requirements):** explican cómo obtener las solicitudes de los interesados y transformarlas en un conjunto de requisitos de trabajo para construir y proporcionar requisitos detallados de lo que debe hacer el sistema [47].

Esta disciplina tiene el propósito de:

- Establecer y mantener un acuerdo con los clientes y los otros interesados acerca de la importancia de hacer el sistema.
 - Proveer a los desarrolladores del sistema de un mejor entendimiento de los requerimientos del sistema.
 - Definir los límites (o delimitar) del sistema.
 - Proveer una base para la planeación de los contenidos técnicos de las iteraciones.
 - Proveer una base para la estimación de costo y tiempo necesarios para desarrollar el sistema.
 - Definir una interfaz de usuario para el sistema, enfocada en las necesidades y objetivos del usuario.
- **Análisis y diseño (Analysis and Design):** el objetivo del análisis y diseño es [47]:
 - Transformar los requerimientos a diseños del sistema.
 - Desarrollar una arquitectura robusta para el sistema.
 - Adaptar el diseño para hacerlo corresponder con el ambiente de implementación y ajustarlo para un desempeño esperado.
 - **Implementación (Implementation):** el proceso describe cómo reutilizar los componentes existentes o implementar nuevos componentes con responsabilidad bien definida, haciendo que el sistema sea más fácil de mantener y aumentar las posibilidades de reutilización [47].

Sus propósitos son:

- Definir la organización del código, en términos de la implementación de los subsistemas organizados en capas.
- Implementar el diseño de elementos en términos de los elementos (archivos fuente, binarios, ejecutables y otros)
- Probar los componentes desarrollados como unidades.
- Integrar los resultados individuales en un sistema ejecutable.
- **Pruebas (Test):** los objetivos de las pruebas son [47]:
 - Verificar la interacción entre objetos
 - Verificar la correcta integración de todos los componentes del software
 - Verificar que todos los requisitos se han implementado correctamente
 - Identificar y asegurar que los defectos se tratan antes de la implementación del software
 - Asegúrese de que todos los defectos son fijos, re-probarlos y cerrarlos
- **Despliegue (Deployment):** el propósito del despliegue es producir con éxito el lanzamiento de los productos y entregar el software a sus usuarios finales [47].

Artefactos

“Un artefacto es una pieza de información que es producida, modificada o usada por un proceso. Los artefactos son productos tangibles del proyecto, las cosas que el proyecto produce o utiliza en el trabajo hacia el producto final” [45].

Un artefacto puede ser:

- **Un documento:** como un Caso de Negocio o un documento de la arquitectura del Software.
- **Un modelo:** como un modelo de caso de uso.
- **Un elemento** de un modelo: como una sola clase de todo el Diagrama de Clases.
- **Código fuente**
- **Ejecutable**

Los artefactos se utilizan como entrada por los trabajadores para llevar a cabo una actividad, y son el resultado o salida de tales actividades. Cada uno de los roles está encargado de crear y actualizar los artefactos que deben ser objeto de control de versiones durante el ciclo de vida del proyecto [46].

Actividades

Un proceso de desarrollo de software define quién hace qué, cómo y cuándo. RUP define cuatro elementos [46]:

- Los roles que responden a la pregunta ¿Quién?
- Las actividades que responden a la pregunta ¿Cómo?
- Los artefactos que responden a la pregunta ¿Qué?
- Los flujos de trabajo de las disciplinas que responde a la pregunta ¿Cuándo?

El grupo de actividades que se encuentran dentro de una disciplina principalmente son una ayuda para entender el proyecto desde la perspectiva clásica de cascada.

Roles

Un rol define el comportamiento y las responsabilidades de un individuo, o un conjunto de individuos para trabajar juntos como un equipo, en el contexto de una organización de desarrollo de software. Una persona puede desempeñar diversos roles, así como un mismo rol puede ser desempeñado por varias personas [46].

Un rol es responsable de uno o más artefactos y lleva a cabo un conjunto de actividades.

UML: UNIFIED MODELING LANGUAGE

El Lenguaje de Modelamiento Unificado (UML) es un lenguaje para especificación, visualización, construcción y documentación de elementos de sistemas de software, modelamiento de negocios; y además puede ser usado para modelar otros sistemas que no sean software [26].

Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group).

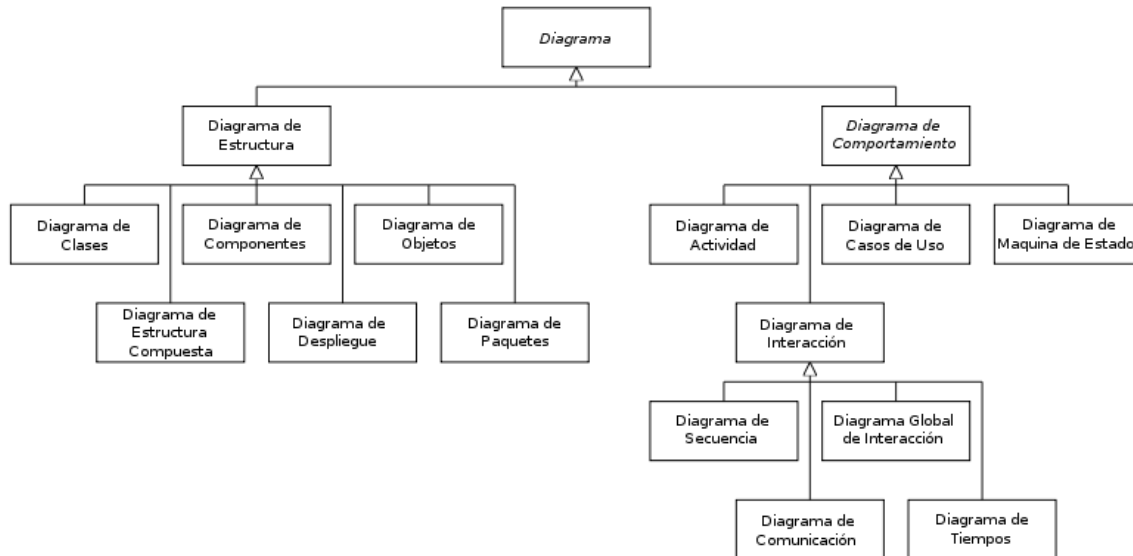
UML son las siglas para Unified Modeling Language, que en castellano quiere decir: Lenguaje de Modelado Unificado. Para comprender qué es el UML, basta con analizar cada una de las palabras que lo componen por separado.

- **Lenguaje:** el UML como lenguaje es mecanismo de comunicación. Lo que implica que éste cuenta con una sintaxis y una semántica. Por lo tanto, al modelar un concepto en UML, existen reglas sobre cómo deben agruparse los elementos del lenguaje y el significado de esta agrupación.
- **Modelado:** el UML es visual. Mediante su sintaxis se modelan distintos aspectos del mundo real, que permiten una mejor interpretación y entendimiento de éste.
- **Unificado:** unifica varias técnicas de modelado en una única.

Organización de la superestructura

El bloque de construcción básico del UML es un diagrama. La estructura de los diagramas UML está reflejado por el diagrama de la Figura 7: Estructura diagramas UML 2.0, de acuerdo con la especificación del UML 2.0 del Object Development Group. Los detalles sobre estos diagramas específicos se organizan de acuerdo a esta estructura taxonómica, que da la perspectiva a los diagramas y a sus interrelaciones [27].

Figura 7 Estructura diagramas UML 2.0



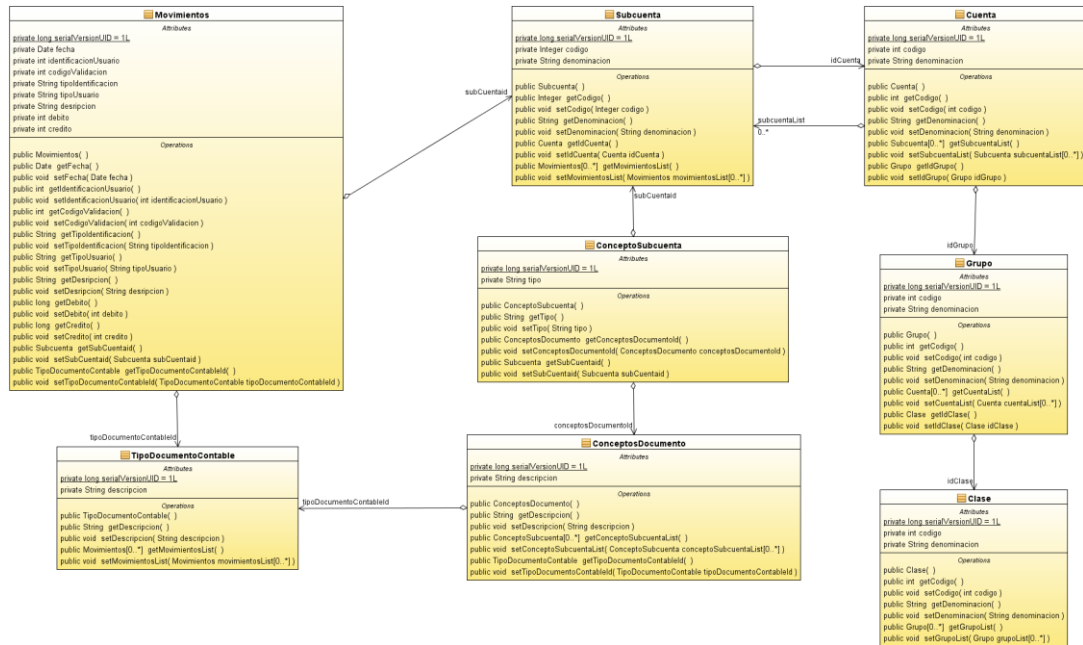
Tomado de: http://www.epidataconsulting.com/tikiwiki/tiki-read_article.php?articleId=15

Diagramas de Estructura

Los diagramas estructurales representan elementos y así componen un sistema o una función. Estos diagramas pueden reflejar las relaciones estáticas de una estructura, como lo hacen los diagramas de clases o de paquetes, o arquitecturas en tiempo de ejecución, tales como diagramas de Objetos o de Estructura Compuesta.

- **Diagramas de Clase:** Los diagramas de clases, [Figura 8: Diagrama de clase](#), muestran los bloques de construcción de cualquier sistema orientado a objetos. Se dice que los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas. Las generalizaciones, agregaciones, y asociaciones son todas valiosas al reflejar herencias, composición o uso, y conexiones respectivamente [27].

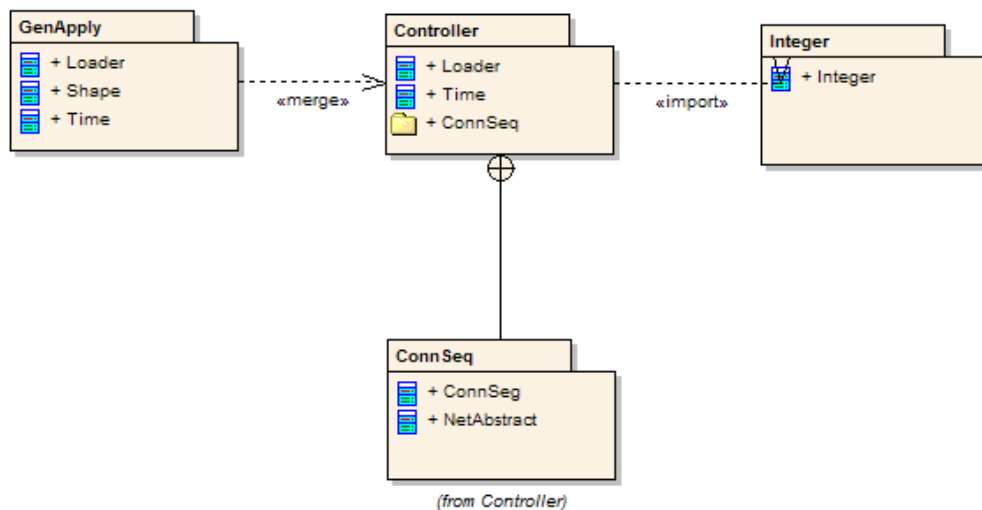
Figura 8 Diagrama de clase



Tomado de Diagrama de clases – Prototipo Ayacucho, La Buitrera

- **Diagrama de Paquetes:** Un diagrama de paquetes, Figura 9: Diagrama de paquetes, muestra como un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre esas agrupaciones. Dado que normalmente un paquete está pensado como un directorio, los diagramas de paquetes suministran una descomposición de la jerarquía lógica de un sistema [28].

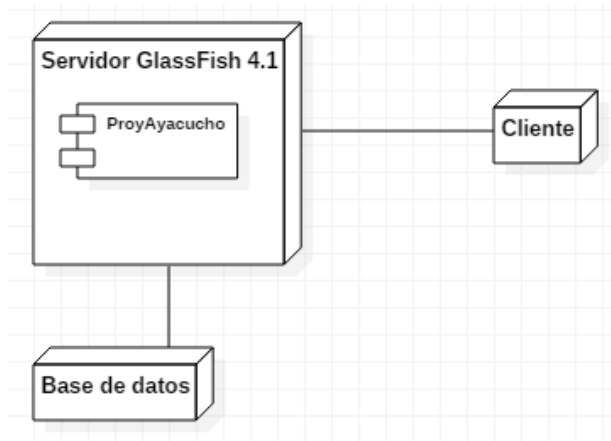
Figura 9 Diagrama de paquetes



Tomado de <https://docs.kde.org/trunk4/es/kdesdk/umbrello/uml-basics.html>

- **Diagrama de Despliegue:** Un Diagrama de Despliegue, Figura 10: Diagrama de Despliegue, modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos [27].

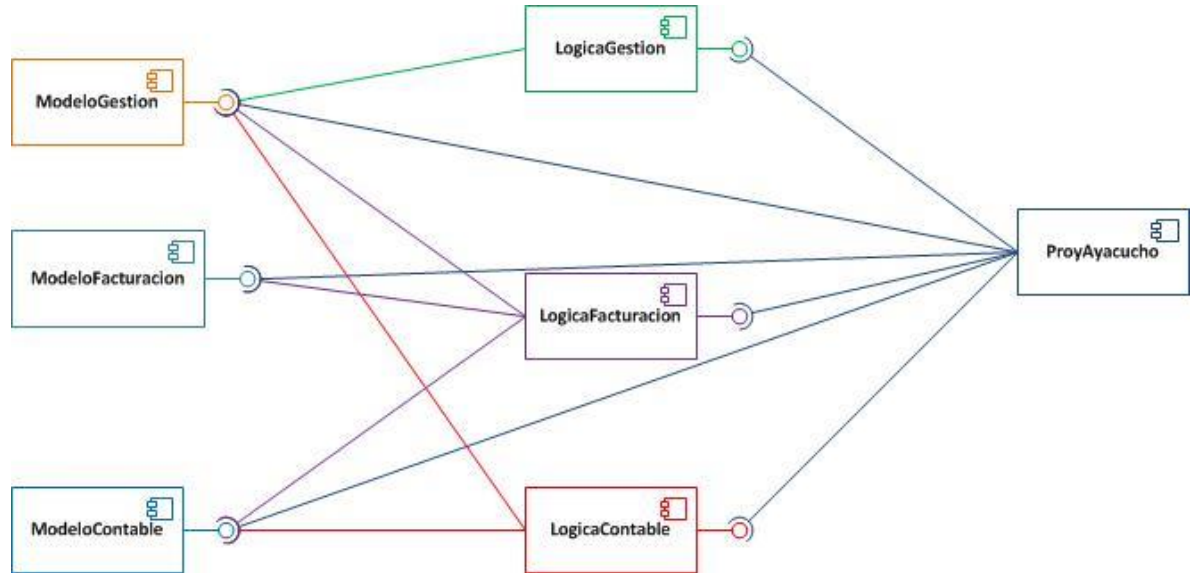
Figura 10 Diagrama de Despliegue



Tomado de Documento Arquitectura de Software – Prototipo Ayacucho, La Buitrera

- **Diagramas de Componentes:** Los Diagramas de Componentes, Figura 11: Diagrama de componentes, ilustran las piezas del software (ya sea las tecnologías que lo forman como Kparts, componentes CORBA, Java Beans o simplemente secciones del sistema claramente distintas) que conformarán el sistema. Un diagrama de componentes tiene un nivel más alto de abstracción que un diagrama de clase – usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema [27].

Figura 11 Diagrama de Componentes



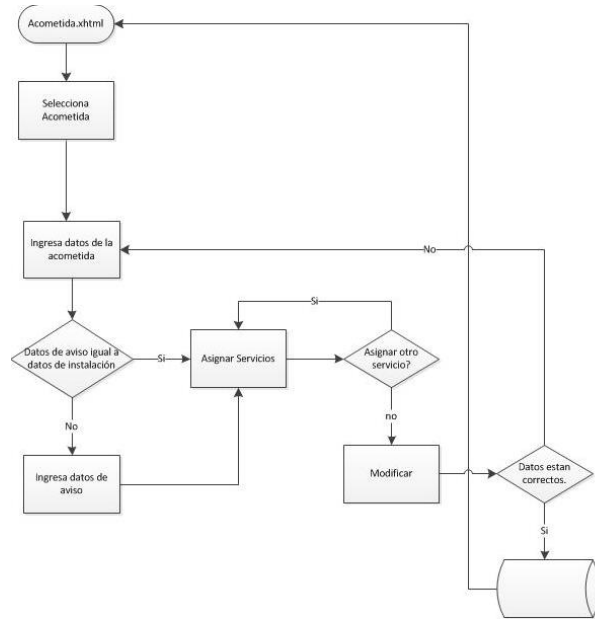
Tomado de Documento Arquitectura de Software – Prototipo Ayacucho, La Buitrera

Diagramas de Comportamiento

Los diagramas de comportamiento representan las características de comportamiento de un sistema o proceso de negocios y, a su vez, incluyen a los diagramas de: actividades, casos de uso, máquinas de estados, tiempos, secuencias, repaso de interacciones y comunicaciones [27].

- **Diagramas de Actividad:** Los diagramas de actividades, Figura 12: Diagrama de Actividad, muestran el flujo de trabajo desde el punto de inicio hasta el punto final detallando muchas de las rutas de decisiones que existen en el progreso de eventos contenidos en la actividad. Estos también pueden usarse para detallar situaciones donde el proceso paralelo puede ocurrir en la ejecución de algunas actividades [28].

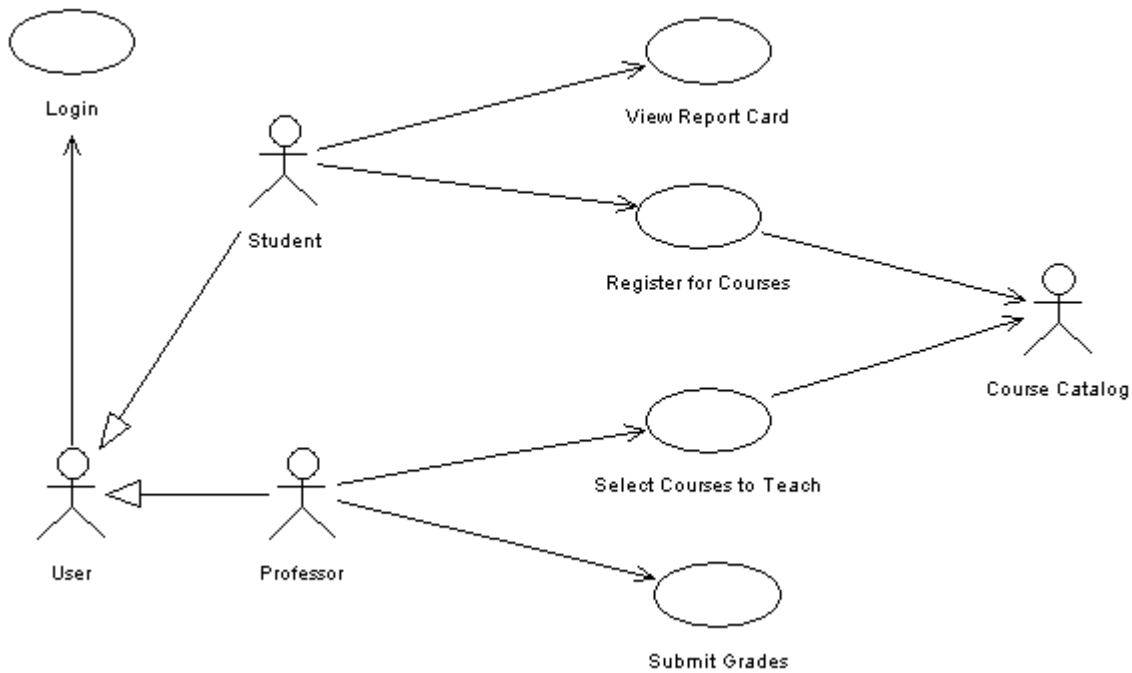
Figura 12 Diagrama de Actividad



Tomado de Documento Diagrama de Actividades – Prototipo Ayacucho, La Buitrera

- **Diagramas de Casos de Uso:** Un caso de uso, Figura 13: Casos de uso, expresa todas las formas de usar un sistema para alcanzar una meta particular para un usuario. En conjunto, los casos de uso le proporcionan todos los caminos útiles de usar el sistema e ilustran el valor que este provee [28].

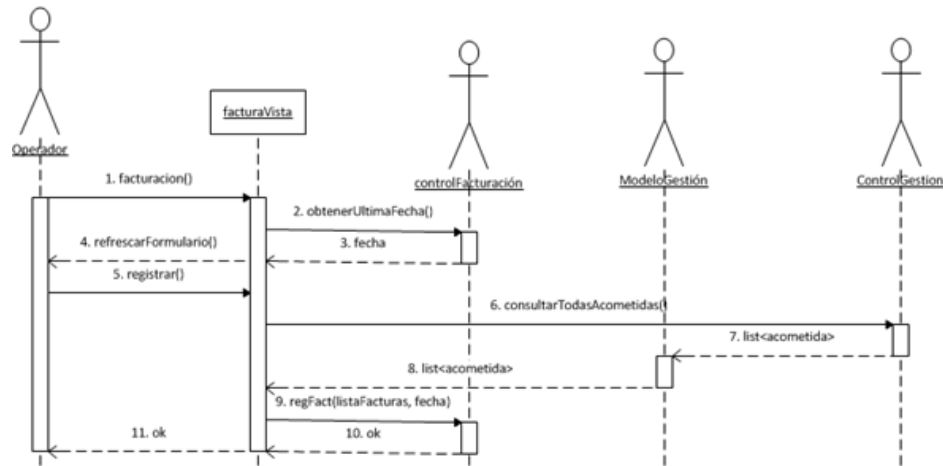
Figura 13 Casos de Uso



Tomado de <https://docs.kde.org/trunk4/es/kdesdk/umbrello/uml-basics.html>

- **Diagramas de interacción:** Los Diagramas de Interacción son modelos que describen la manera en que colaboran grupos de objetos para cierto comportamiento. Se deberán usar diagramas de interacción si se quiere analizar el comportamiento de un grupo de objetos en un mismo caso de uso [27].
 - **Diagramas de Secuencia:** Un diagrama de secuencia, Figura 14: Diagramas de Secuencia, es una forma de diagrama de interacción que muestra los distintos estados del objeto y la secuencia de eventos que conllevan al cambio en el estado del mismo durante su tiempo de vida. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con objetos y qué mensajes disparan esas comunicaciones. Los diagramas de secuencia no están pensados para mostrar lógicas de procedimientos complejos [28].

Figura 14 Diagrama de Secuencia



Tomado de Documento Arquitectura de Software – Prototipo Ayacucho, La Buitrera

ACTIVIDADES PARA LA DESCRIPCION DE LA ARQUITECTURA DE SOFTWARE BASADA EN COMPONENTES

Cabe resaltar que el arquitecto de software debe ser una persona equilibrada, debe poseer madurez, visión y una experiencia profunda que le permita comprender la problemática de manera rápida y realizar juicio cultural ante la ausencia de información completa. Más específicamente, el arquitecto de software, o miembros del equipo de arquitectura, deben combinar estas habilidades [46]:

- Experiencia en el dominio del problema, a través de un conocimiento profundo de los requisitos y el dominio de la ingeniería de software. Si hay un equipo, al menos un arquitecto de software debe proporcionar la visión global del proyecto.
- Liderazgo con el fin de impulsar el esfuerzo técnico a través de los distintos equipos, y para tomar decisiones críticas a presión y hacer que esas decisiones se mantengan.
- Comunicación para ganar confianza, persuadir, motivar, y para guiar. La arquitectura de software no puede conducir por decreto, sólo por el consentimiento del resto del proyecto. Para ser eficaz, el arquitecto de software debe ganarse el respeto del equipo del proyecto, el director del proyecto, el cliente, y la comunidad de usuarios, así como al equipo de gestión.
- El arquitecto de software es la técnica impulsora del proyecto, no un visionario o soñador. La carrera exitosa de un arquitecto de software es una larga serie de decisiones óptimas en la incertidumbre y bajo presión.

El Rational Unified Process o RUP, es una metodología iterativa – incremental implementada a partir del desarrollo en espiral, por tal motivo en cada iteración las partes que conforman el producto final se van refinando. No existe una única versión de un artefacto. Cada iteración realiza

procesos basados en los resultados obtenidos de la anterior iteración y da como resultado un artefacto refinado [46].

A continuación se describen las actividades que se llevan a cabo por el arquitecto con el objetivo de realizar el documento de la arquitectura de software basado en componentes, según lo planteado por RUP. También se describen puntos críticos presentes en algunas actividades que son esenciales para la arquitectura basada en componentes y no necesariamente para otras arquitecturas. El primer punto crítico se encuentra en la actividad Definir una arquitectura candidata y el segundo punto crítico se encuentra en la actividad Refinar la arquitectura.

1. Gestionar el alcance del sistema

El alcance de un proyecto se define por el conjunto de requisitos que se le asignen. La gestión del alcance del proyecto debe adaptarse a los recursos disponibles (tiempo, personas y dinero) que es la clave para la gestión de proyectos exitosos.

Gestionar el alcance es una actividad continua que requiere que el proyecto se divida en piezas más pequeñas y manejables, para así poder controlarlo durante todo el ciclo del proyecto.

Durante el grupo de actividades realizadas para gestionar el alcance del sistema, el arquitecto realiza la siguiente actividad:

1.1. Priorizar los casos de uso y escenarios

El arquitecto de software propone el contenido técnico y el orden de las sucesivas iteraciones mediante la selección de un determinado número de escenarios y casos de uso para ser analizados y diseñados, Figura 15: Gestionar el alcance del sistema. Esta propuesta técnica es completada y refinada por los diferentes equipos de desarrollo basados en la disponibilidad de personal, requerimientos del cliente en términos de entregas, disponibilidad de herramientas y las necesidades de otros proyectos [47].

La selección de escenarios y casos de uso es impulsada por lo siguiente [47]:

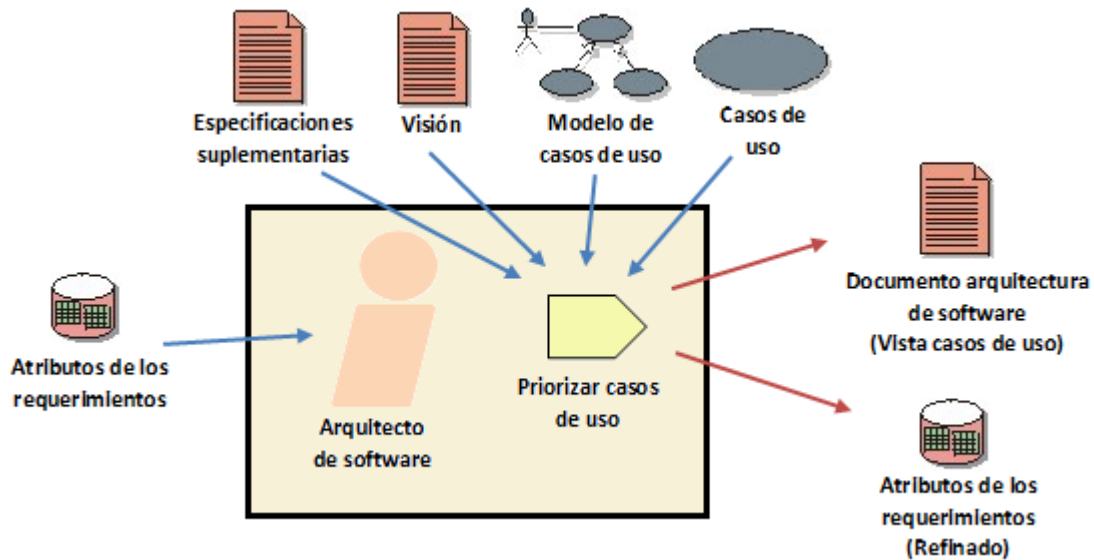
- El beneficio del escenario a las partes interesadas: crítica, importante, útil
- El impacto arquitectónico del escenario: ninguno, se extiende, modifica
- Los riesgos a ser mitigados (rendimiento, la disponibilidad de un producto, y la idoneidad de un componente)
- La total cobertura de la arquitectura (asegurándose de que al final de la fase de elaboración, cada pieza de software a desarrollar ha encontrado su lugar en la implementación).

La sección de vista de casos de uso del documento de la arquitectura de software contiene una lista de los casos de uso y escenarios significativos dentro de cada paquete del modelo de casos de uso junto con las propiedades importantes, tales como descripciones del flujo de eventos, relaciones, diagramas de casos de uso y requisitos especiales relacionados con cada caso de uso [47].

Para realizar esta actividad, el arquitecto de software requiere de los siguientes artefactos [47]:

- **Visión:** Define la vista de los interesados del producto a desarrollar, especificada en términos de las características y necesidades principales de las partes interesadas. Contiene un resumen de los requisitos fundamentales previstos y proporciona la base contractual para los requisitos técnicos más detallados. Responsable: Analista de sistemas.
- **Especificaciones complementarias:** reúne los requisitos del sistema que no son fácilmente capturados como casos de uso del modelo de caso de uso. Tales requisitos incluyen:
 - Requisitos legales y reglamentarios, y normas de aplicación
 - Atributos de calidad del sistema a construir, incluyendo requisitos de usabilidad, fiabilidad, rendimiento y compatibilidad
 - Otros requisitos tales como sistemas operativos y entornos, requisitos de compatibilidad y restricciones de diseño.Responsable: Analista de sistemas.
- **Modelo de casos de uso:** es un modelo de funciones del sistema y su entorno, y sirve como un contrato entre el cliente y los desarrolladores. El modelo de caso de uso se utiliza como insumo esencial para actividades de análisis, diseño y prueba. Responsable: Analista de sistemas.
- **Casos de uso:** define un conjunto de instancias de casos de uso, donde cada instancia es una secuencia de acciones que realiza un sistema que produce un resultado observable de valor para un actor particular. Responsable: Ingeniero de requerimientos.
- **Atributos de los requerimientos:** Este artefacto contiene un repositorio de los requisitos del proyecto, los atributos y dependencias para realizar un seguimiento desde una perspectiva de gestión de requisitos. Responsable: Analista de sistemas.

Figura 15 Gestionar el Alcance del Sistema



Tomado de: http://sce.uhcl.edu/helm/rationalunifiedprocess/process/workflow/requirem/wfs_sco.htm

2. Definir una arquitectura candidata

Esta actividad se realiza mejor con un grupo pequeño de miembros que tengan funciones cruzadas. Los asuntos que son de gran importancia arquitectónica típicamente incluyen el rendimiento, la escalabilidad, el proceso y la sincronización de subprocessos, y la distribución. El equipo tiene que ser capaz de tirar de todos los hilos de una forma cohesiva y coherente [47].

Esta actividad tiene como objetivo:

- Crear un esbozo inicial de la arquitectura del sistema
 - Definir un conjunto inicial de elementos de gran importancia arquitectónica que se utilizarán como base para el análisis
 - Definir un conjunto inicial de mecanismos de análisis
 - Definir la estratificación inicial y organización del sistema
 - Definir las realizaciones de casos de uso que se abordarán en la iteración actual
 - Definir la estrategia de reutilización
- Identificar las clases de análisis de los casos de uso de gran importancia arquitectónica
- Actualizar las realizaciones de casos de uso con las interacciones de las clases de análisis

Esta actividad se realiza mejor en varias sesiones con la iteración entre el análisis arquitectónico y el análisis de los casos de uso. Realizar una pasada inicial de la arquitectura y a continuación, elegir los casos de uso de gran importancia arquitectónica y realizar el análisis a cada uno. Después de analizar cada caso de uso, actualizar la arquitectura tanto sea necesario para reflejar las adaptaciones que definen el nuevo comportamiento del sistema y para hacer frente a los posibles problemas que se identifiquen en la arquitectura [47].

Punto crítico 1: Pensar en componentes

Durante la realización de las actividades para definir una arquitectura candidata, el equipo participante debe tener en cuenta lo siguiente [46]:

“El desarrollo basado en componentes enfatiza en la separación de asuntos”

El equipo debe examinar los requerimientos para determinar cuál(es) sub-sistemas están dispuestos para la composición y no para la construcción. Siguiendo este enfoque, el equipo debe hacerse preguntas orientadoras tales como [46]:

- ¿Se dispone de componentes reutilizables desarrollados internamente para implementar el requerimiento?
- ¿Es posible disponer de componentes comerciales ya desarrollados para implementar el requerimiento?
- ¿Son compatibles las interfaces de los componentes que están disponibles dentro de la arquitectura del sistema a construir?

En el ejercicio de determinar los subsistemas que pueden ser satisfechos con el uso de componentes existentes, también se analizan los sub-sistemas candidatos a reutilización. Existen algunas preguntas orientadoras para realizar la identificación de estos [46]:

- ¿Es la funcionalidad del componente requerida en futuras implementaciones?
- ¿Qué tan común es la función del componente dentro del dominio?
- ¿Hay duplicidad de la función del componente dentro del dominio?
- ¿Se puede reutilizar el componente en varias implementaciones con sólo cambios menores?
- ¿Podría un componente no reutilizable ser descompuesto en sub-componentes reutilizables? ¿Qué tan válido sería?

Se debe tener en cuenta que el dominio representa, en términos generales, todos los aspectos del problema del usuario relacionado de forma directa con la funcionalidad que apoya el componente [48].

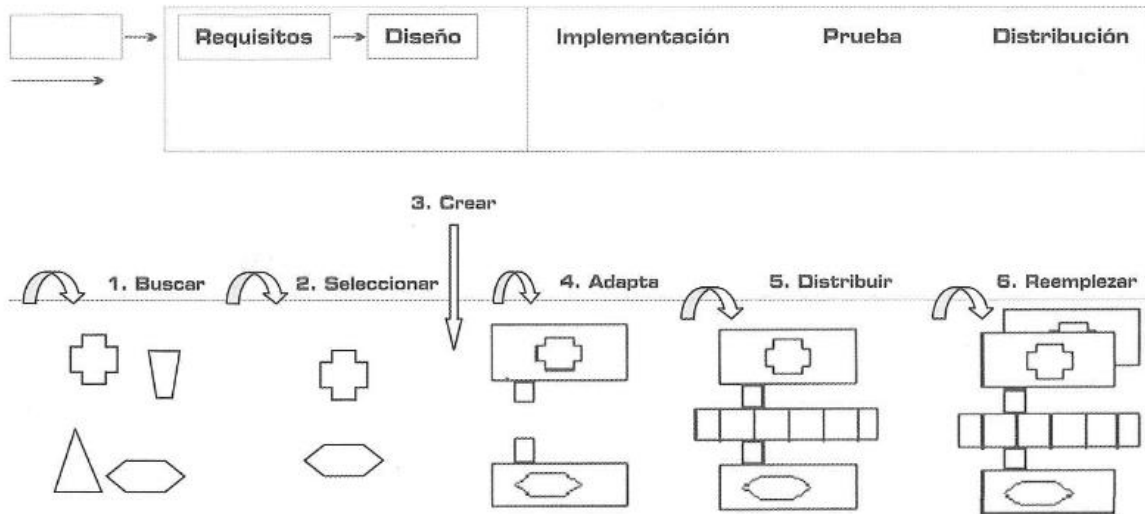
Cuando se piensa en desarrollo basado en componentes se debe tener claro que existen dos conceptos ligados: desarrollar un componente y usar un componente disponible.

- Desarrollo de componentes: En caso de que hayan requerimientos que no se puedan suplir con componentes comerciales o de desarrollo interno, y estos son altamente reutilizables en otros proyectos, se aplican métodos de desarrollo de software con el objetivo de construirlos para dar solución a los requerimientos exigidos.
- Uso de componentes disponibles: Los requerimientos que se abordan con los componentes disponibles, dan inicio a las actividades de selección y adaptación, descritas en la Figura 16: Ciclo de vida basado en componentes, con el objetivo de implementar la mejor solución.

En la Figura 16: Ciclo de vida basado en componentes, se puede apreciar el ciclo de vida de la ingeniería basada en componentes en contraste con otras ingenierías. Este ciclo permite ser

incorporado en las fases del ciclo de desarrollo que esté usando el equipo, con el objetivo de incorporar componentes al proyecto.

Figura 16 Ciclo de vida basado en componentes



Tomado de:

http://www.uelbosque.edu.co/sites/default/files/publicaciones/revistas/revista_tecnologia/volumen4_numero1/ingenieria_software_basada_componentes4-1.pdf

Debe incorporarse ahora al proceso de desarrollo las actividades correspondientes a:

Selección de componentes

Esta actividad se realiza en dos fases:

1. **Búsqueda y filtrado:** involucra un conjunto de actividades que intentan identificar y encontrar componentes candidatos que sean altamente reutilizables y cumplan con algunos de los requerimientos exigidos.
2. **Análisis:** Se analizan un conjunto de alternativas ponderadas, donde la alternativa con mayor ponderación se somete a la decisión de “*comprar o desarrollar*”.

Adaptación de componentes

Integración de componentes

Tan importante como determinar las incompatibilidades arquitectónicas es calcular el esfuerzo de integración. En esta actividad se debe examinar, evaluar y determinar cómo va a ser la comunicación y la coordinación entre los componentes que harán parte del sistema. Luego debe ensamblarse el sistema y proseguir con una serie de pruebas que determinarán si los componentes seleccionados son los adecuados.

Pruebas a los componentes en el sistema

Posterior a la integración de los componentes se debe proceder con las pruebas, esto implica evaluar el funcionamiento de los componentes que fueron integrados en el sistema, si algún componente demuestra no estar funcionando de forma correcta se debe pensar en la posibilidad de reemplazarlo o modificarlo para luego proceder con la re – integración.

Durante las actividades para definir una arquitectura candidata, el arquitecto realiza un análisis con el objeto de definir y limitar la arquitectura con la que contará el sistema.

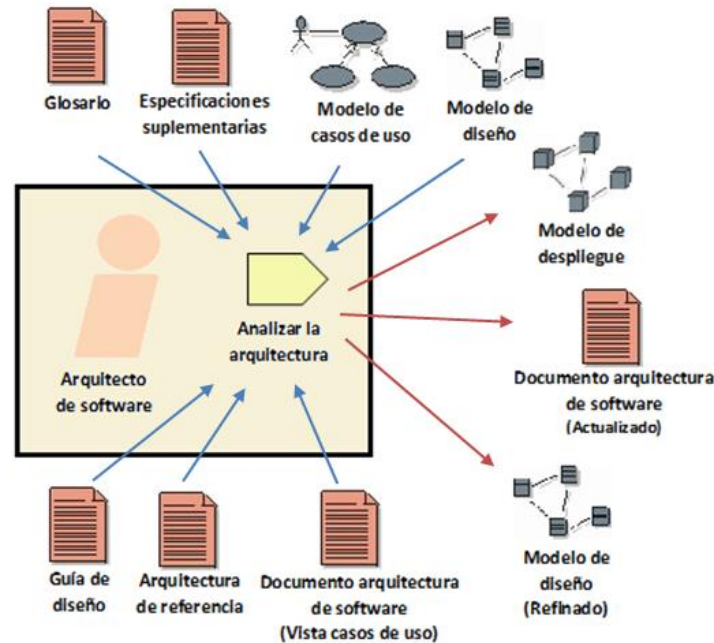
2.1. Análisis arquitectónico

El análisis arquitectónico se centra en definir una arquitectura candidata y limitar las técnicas arquitectónicas a ser usadas en el sistema, Figura 17: Definir una arquitectura candidata. Se basa en la recopilación de experiencia adquirida en sistemas similares o dominios de problemas para limitar y enfocar la arquitectura de manera que el esfuerzo no se desperdicie en el redescubrimiento de la arquitectura [47].

Para las tareas realizadas con el fin de definir una arquitectura candidata, el arquitecto requiere los siguientes artefactos [46]:

- Especificaciones suplementarias
- Modelo de casos de uso
- Guía de diseño
- Arquitectura de referencia
- Documento de arquitectura de software: Vistas de casos de uso
- **Glosario:** Se definen los términos más importantes utilizados por el proyecto. Responsable: Analista de sistemas.
- **Modelo de diseño:** es un modelo de objetos que describe la realización de casos de uso, y sirve como una abstracción del modelo de implementación y su código fuente. El modelo de diseño se utiliza como insumo esencial para las actividades de implementación y prueba. Responsable: Arquitecto de software.

Figura 17 Definir arquitectura candidata



Tomado de: http://sce.uhcl.edu/helm/rationalunifiedprocess/process/workflow/ana_desi/wfs_and1.htm

3. Refinar la arquitectura

Durante las actividades que se llevan a cabo en la disciplina de análisis y diseño, se realiza un refinamiento a la arquitectura, Figura 18: Refinar la arquitectura. Esto con el propósito de [46]:

- Proporcionar la transición natural de las actividades de análisis a las actividades de diseño.
- Mantener la consistencia e integridad de la arquitectura.
- Describir la organización de la arquitectura del sistema en tiempo de ejecución e implementación.
- Organizar el modelo de aplicación para hacer una transición transparente entre el diseño y ejecución.

Las tareas para refinar la arquitectura son:

3.1. Identificar mecanismos de diseño

Se realiza con el propósito de perfeccionar los mecanismos de análisis sobre los mecanismos de diseño basados en las limitaciones impuestas por el entorno de la aplicación. Proporcionan los conjuntos conceptuales de servicios que son utilizados por los objetos de análisis [46].

Para ello, se realiza un inventario de los mecanismos de aplicación que se tienen a disposición:

- Mecanismos que ofrece un producto de middleware
- Mecanismos ofrecidos por los sistemas operativos

- Mecanismos ofrecidos por un componente
- Mecanismos que ofrece una biblioteca de clases
- Código heredado
- Paquetes de propósito especial: constructor de interfaz gráfica de usuario, Sistema de Información Geográfica, etc.

3.2. Identificar elementos de diseño

Se realiza con el objetivo de analizar las interacciones de las clases de análisis para identificar los elementos del modelo de diseño. El análisis a los casos de uso resulta en clases de análisis que representan los conceptos de los comportamientos que se pueden presentar [47]. Esta actividad da como resultado diferentes tipos de elementos de diseño, tales como:

- Clases: que representan un conjunto de responsabilidades de grano fino.
- Subsistemas: que representan un conjunto de responsabilidades de grano grueso. Puede estar compuesto de una serie de subsistemas, pero a la final es una instancia de un conjunto de clases.
- Hilos de control en el sistema
- Interfaces: representan declaraciones abstractas de responsabilidades proporcionadas por una clase o subsistema.
- Eventos: son especificaciones de sucesos interesantes en el tiempo y espacio que normalmente requieren de una respuesta del sistema.
- Señales: representan mecanismos asíncronos utilizados para comunicar tipos de eventos dentro del sistema.

3.3. Incorporar elementos de diseño existentes

Se realiza con el objetivo de refinar la arquitectura (la incorporación de la reutilización cuando sea posible) y para identificar soluciones comunes a los problemas más frecuentes de diseño.

Esta actividad permite identificar dónde existen subsistemas y/o componentes que pueden ser reutilizados en función de sus interfaces [47].

3.4. Describir la arquitectura en tiempo de ejecución

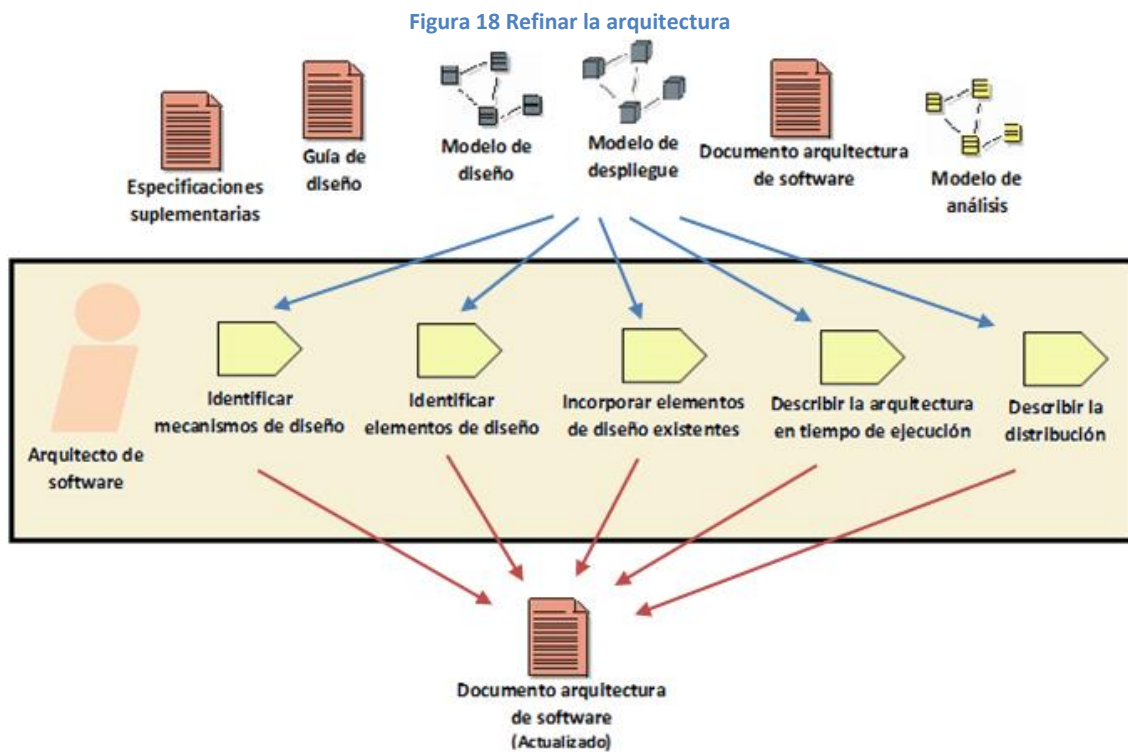
Se realiza con el propósito de analizar los requerimientos de concurrencia, para identificar los procesos, los mecanismos de comunicación entre procesos, asignar los recursos de coordinación entre procesos, identificar a los ciclos de vida del proceso, y distribuir los elementos del modelo entre los procesos [47].

3.5. Describir la distribución

Se realiza para describir la forma en que la funcionalidad del sistema se distribuye a través de nodos físicos (Sólo es necesario para sistemas distribuidos) [47].

Para estas actividades cuyo objetivo es refinar la arquitectura se requieren los siguientes artefactos [46]:

- Especificaciones suplementarias
- Guía de diseño
- Modelo de diseño
- Modelo de despliegue
- Documento arquitectura de software
- **Modelo de análisis:** describe la realización de casos de uso, y sirve como una abstracción del Artefacto: Modelo de diseño. El modelo de análisis es un artefacto opcional.



Tomado de: http://sce.uhcl.edu/helm/rationalunifiedprocess/process/workflow/ana_desi/wfs_and2.htm

Kruchten [47], define que el documento de arquitectura de Software proporciona una descripción comprensiva de la arquitectura del sistema, utilizando diferentes puntos de vista arquitectónicos para describir diversos aspectos del sistema. Este sirve como un medio de comunicación entre los arquitectos de software y otros miembros del equipo del proyecto sobre las decisiones arquitectónicas importantes que se han tomado para el proyecto.

Punto crítico 2: Problemas típicos del DSBC

El desarrollo de software basado en componentes (DSBC) es una disciplina muy joven y por tanto en la que los resultados obtenidos hasta ahora se centran más en la identificación de los problemas que en la resolución de los mismos [12].

Algunos de los problemas descritos por L. Fuentes y compañía [12] con los que se enfrenta el DSBC son:

- **La clarividencia:** se refiere a la dificultad con la que se encuentra un diseñador de un componente para realizar su diseño, pues no conoce quien lo utilizará, ni cómo, ni en qué entorno, ni para qué aplicación. Surge debido a que se busca que un componente sea altamente reutilizado por diversos programas.
- **Evolución de los componentes:** La gestión de la evolución es un problema serio, pues en los sistemas grandes han de poder coexistir varias versiones de un mismo componente. Además, las actualizaciones **de** los componentes adquiridos no están en manos de los desarrolladores del sistema.
- **Particularización:** se presenta cuando se trata de particularizar los servicios ofrecidos por un componente para adaptarlo a las necesidades y requerimientos concretos de la aplicación, sin poder manipular su implementación.
- **Falta de soporte formal:** el DSBC se encuentra con otro reto añadido, como es la dificultad que encuentran los métodos formales para trabajar con las peculiaridades, como puede ser la composición tardía, el polimorfismo o la evolución de los componentes.
- **Interoperabilidad:** *Actualmente, los contratos de los componentes se reducen a la definición de sus interfaces a nivel sintáctico, y la interoperabilidad se reduce a la comprobación de los nombres y perfiles de los métodos. Sin embargo, es necesario ser capaces de referirnos y buscar los servicios que necesitamos por algo más que sus nombres, y poder utilizar los métodos ofrecidos en una interfaz en el orden adecuado. El estudio de la interoperabilidad de componentes a nivel de protocolos (orden entre los mensajes) o semántico (“significado” de las operaciones) es otro de los problemas abiertos [12].*

Documento de Arquitectura de Software (Prototipo)

El resultado de realizar las actividades anteriormente descritas para el prototipo del caso de estudio se ve reflejado en el documento de arquitectura del software, incluido en los anexos. A continuación se mostrarán algunos puntos clave de este documento.

5. Vista Lógica

5.1. Introducción

Se presentan en este punto los sucesivos refinamientos que definen las diferentes unidades lógicas que componen la arquitectura del sistema para la gestión administrativa rural Ayacucho La Buitrera, Palmira.

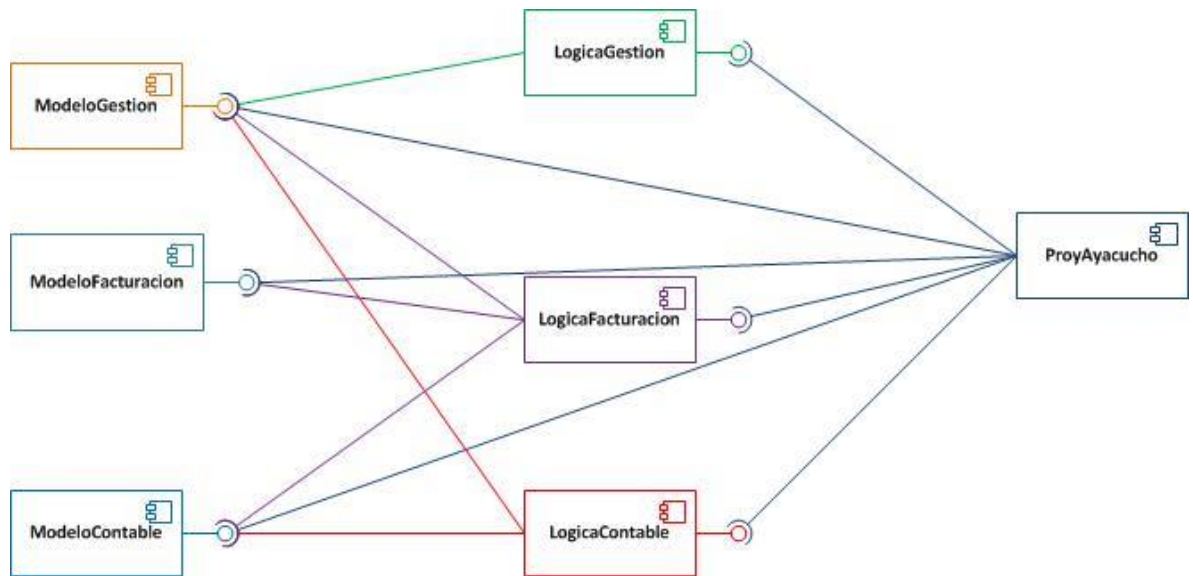
El primer refinamiento realizado consiste en la descripción de los componentes. Cada componente consiste en el agrupamiento de diferentes funcionalidades relacionadas entre sí y posee la capacidad de funcionar como un sistema en sí mismo.

Posteriormente se explora la composición de cada uno de los componentes.

Finalmente se incluye la realización de los casos de uso descritos en la sección anterior mediante los componentes arquitectónicos definidos.

5.2. Definición de los componentes

El desarrollo basado en componentes permite identificar subsistemas que pueden ser reutilizables en otros proyectos que estén dentro del dominio del proyecto a realizarse.



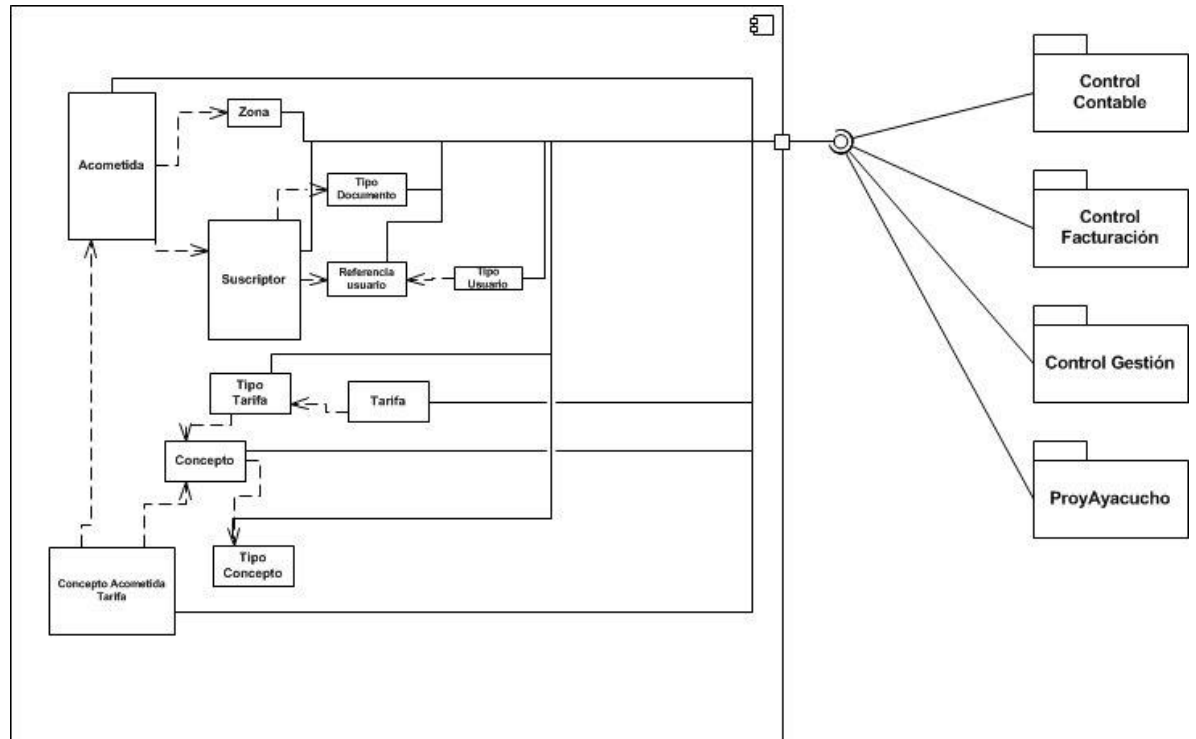
5.3. Descripción de los componentes

- **Modelo Gestión:** Este componente debe contener las entidades relacionadas a la administración del acueducto. Debe permitir una manipulación sencilla al sistema de la información relacionada a:
 - Acometida
 - Suscriptor
 - Zonas
 - Concepto – Acometida – Tarifa
 - Concepto
 - Tipo Concepto
 - Tipo Tarifa
 - Tarifa
 - Tipo Documento
 - Referencia usuario
 - Tipo Usuario

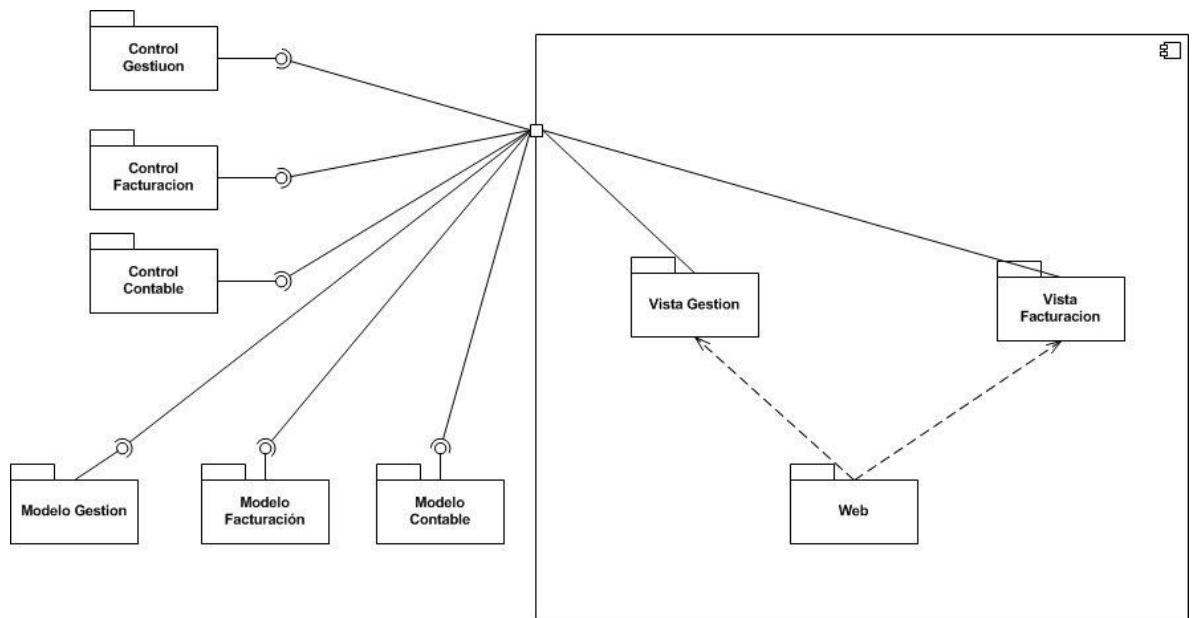
- **Modelo Facturación:** Este componente debe contener las entidades relacionadas a la facturación. Debe permitir al sistema la manipulación de la información correspondiente a:
 - Factura
 - Detalle de la factura
 - Fecha Registro Facturas
- **Modelo Contable:** Este componente debe contener las entidades relacionadas a la gestión contable. Debe permitir al sistema la manipulación de la información correspondiente a:
 - Clase
 - Grupo
 - Cuenta
 - Subcuenta
 - Concepto – Subcuenta
 - Concepto – Documentos
 - Tipo Documento Contable
 - Movimiento
- **Control Gestión:** Este componente es el encargado de proporcionar las herramientas adecuadas que le permitan al sistema manipular la información almacenada en la base de datos, relacionada con la información administrativa propia del acueducto. Debe garantizar una comunicación segura con la base de datos.
- **Control Facturación:** Este componente debe permitirle al sistema manipular la información almacenada en la base de datos, correspondiente al proceso de facturación de los servicios prestados mensualmente a las acometidas instaladas. Debe garantizar una comunicación segura con la base de datos.
- **Control Contable:** Este componente debe permitirle al sistema la manipulación de la información almacenada en la base de datos, correspondiente a los movimientos contables realizados en el acueducto. Debe garantizar una comunicación segura con la base de datos.
- **ProyAyacucho:** Este es el proyecto principal, el cual consume todos los componentes antes mencionados y se encarga de realizar el control de las vistas que se le presentan al usuario final.

5.4. Diseño de los componentes

5.4.1. Componente: Modelo Gestión



5.4.7. Componente: ProyAyacucho



5.5. Realización de los casos de uso relevantes para la arquitectura

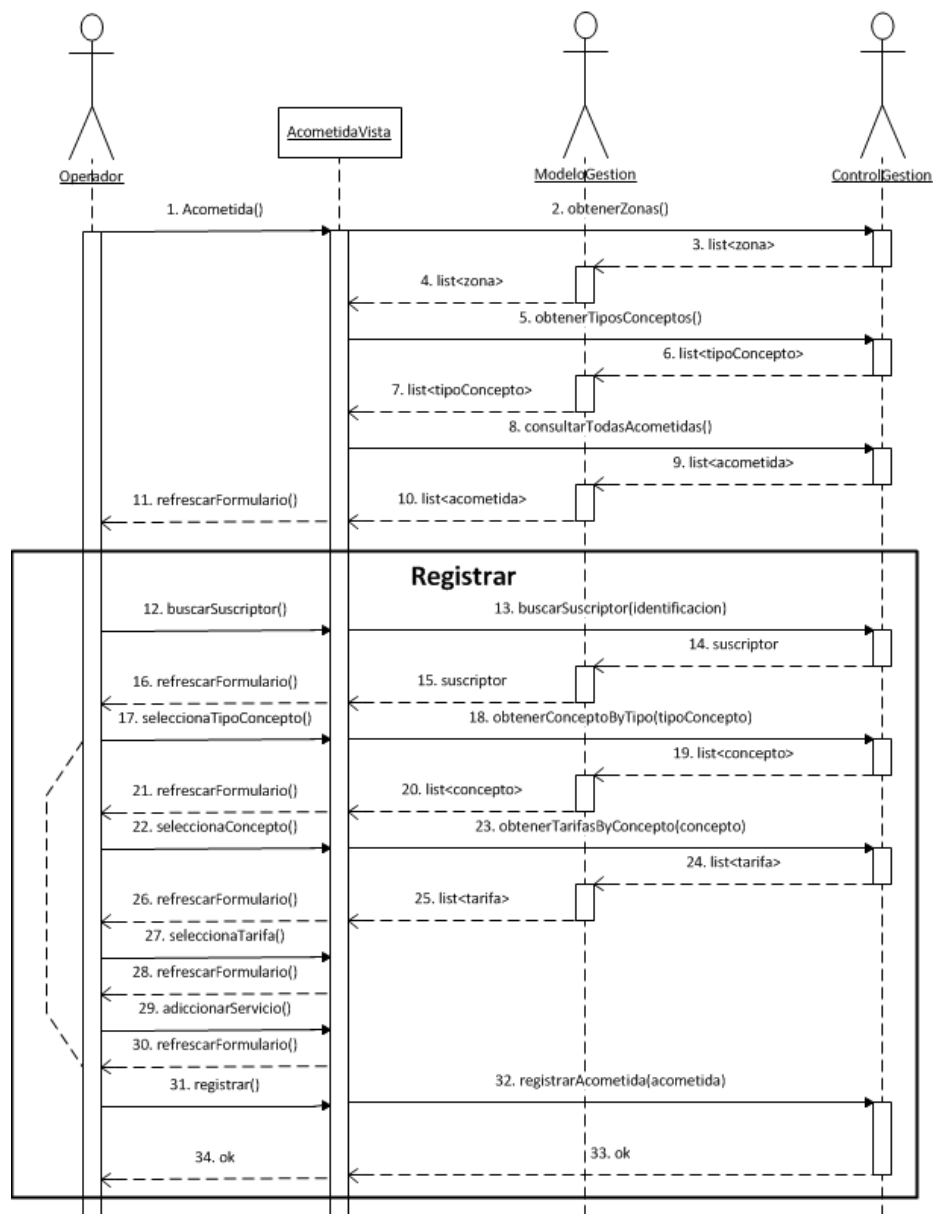
5.5.1. Gestión Acometidas

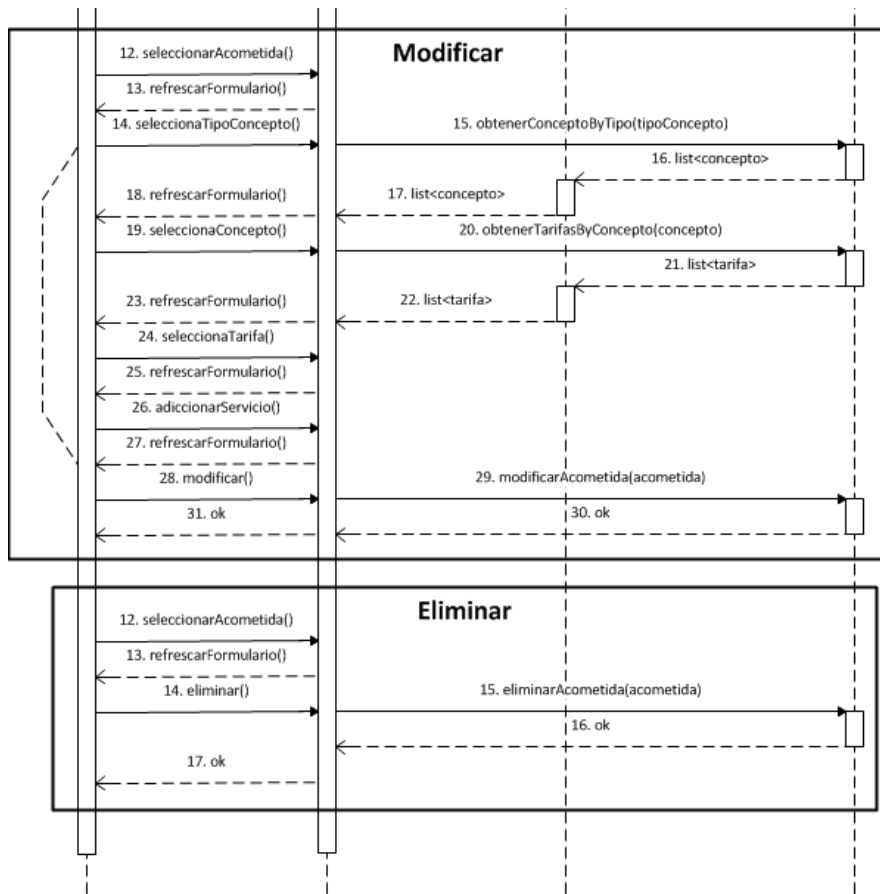
A continuación se describe las secuencias para la comunicación entre componentes que se llevan a cabo para la gestión de las acometidas

- **Gestionar Acometida (ProyAyacucho)**

Descripción del proceso que se realiza para gestionar las acometidas en el sistema ProyAyacucho.

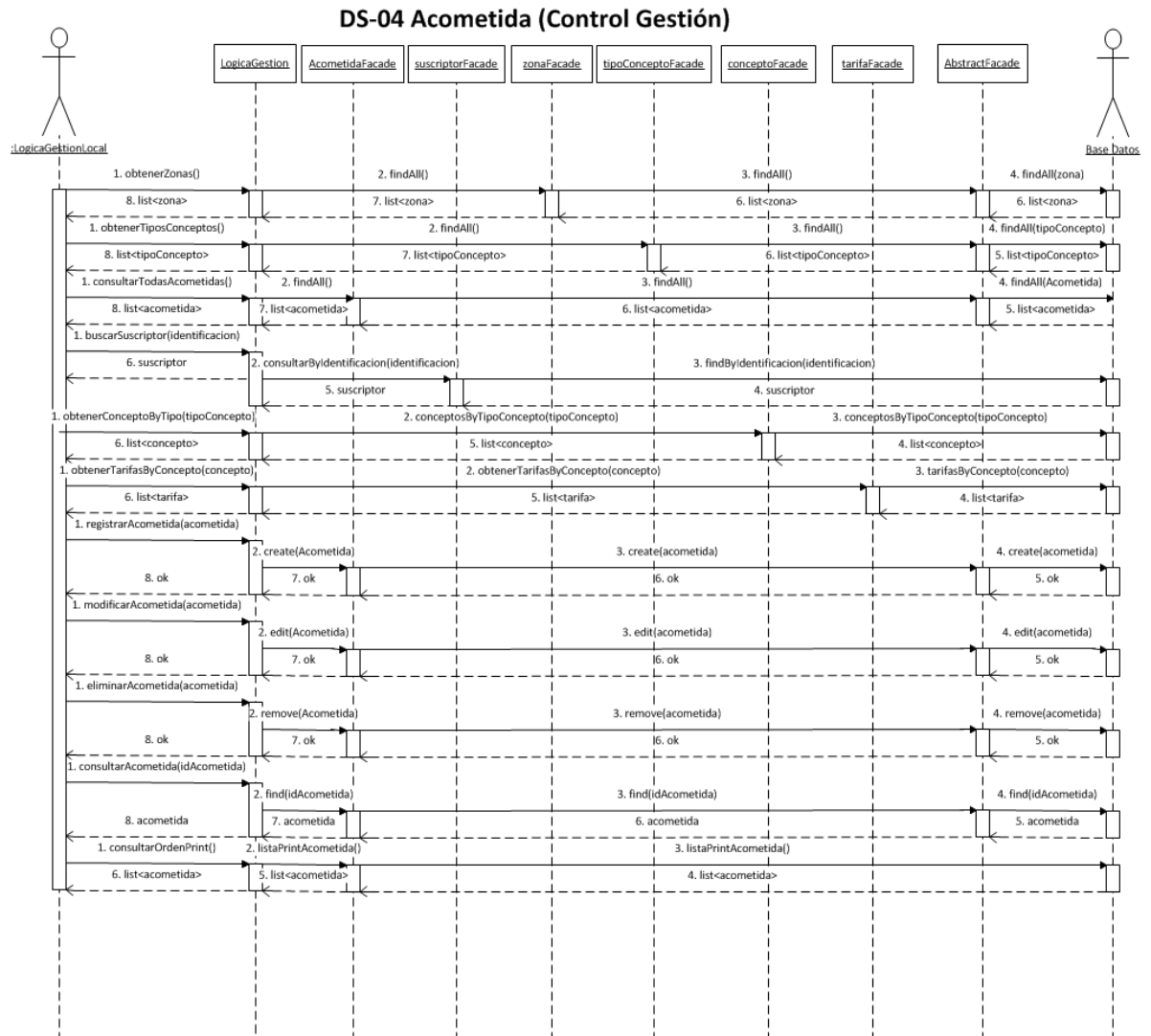
DS-03 Acometida (ProyAyacucho)





- **Gestionar Acometida (Control Gestión)**

Descripción del proceso de gestionar las acometidas desde el componente Control Gestión.



6. Vista de despliegue

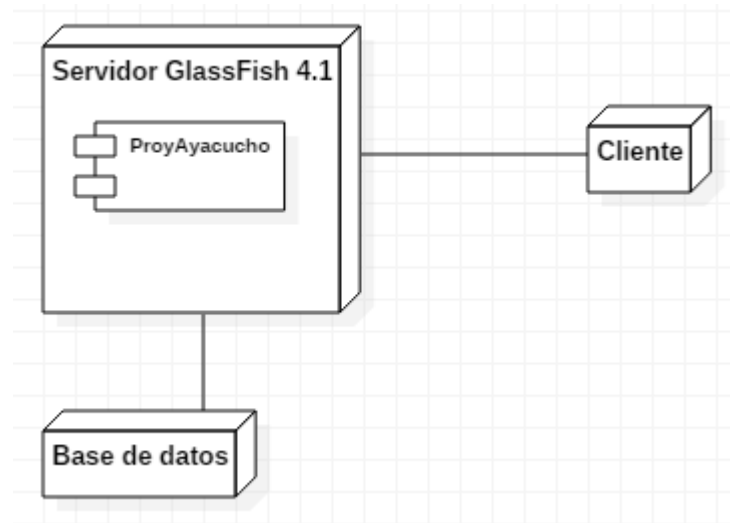
6.1. Introducción

Esta sección describe una o más configuraciones físicas sobre las cuales se realiza el despliegue del software y como es ejecutado, así como la infraestructura necesaria para su instalación.

Para el caso del sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera se describe el escenario general de distribución esperado para los componentes de software antes descritos, las características de los nodos presentados y la comunicación entre los mismos.

6.2. Distribución y despliegue

La figura xx presenta el escenario de distribución esperado para la instalación del sistema. El mismo se ubica en el contexto de una organización sobre una LAN privada.



A continuación se describen los nodos presentes en la figura xx:

- **Servidor GlassFish 4.1:** Es un servidor de aplicaciones desarrollado por Sun Microsystems, que implementa las tecnologías definidas en la plataforma Java Enterprise Edition y permite ejecutar aplicaciones que siguen esta especificación.
En este servidor estará alojada la aplicación ProyAyacucho.
- **Base de datos:** Es un servidor MySQL 10.1.10 para gestionar las bases de datos relacionales.
- **Cliente:** Representa el medio por el cual el operador accederá a la aplicación. Puede realizarlo por los browser Chrome 53.0, Mozilla Firefox 49.0.

Para visualizar este documento completo debe dirigirse al documento de arquitectura incluido en los anexos, allí se encontraran todos los documentos elaborados para el proceso de desarrollo aplicando la arquitectura al prototipo.

STAKEHOLDERS

El desarrollo de software tradicional ha sido impulsado por la necesidad del software entregado para satisfacer las necesidades de los usuarios. Sin embargo, las personas afectadas por un sistema de software no se limitan a aquellos que lo utilizan. Los sistemas de software no sólo son utilizados: tienen que ser construidos y probados, tienen que ser operados, pueden que tengan que ser reparados, por lo general se han mejorado, y por supuesto que tendrán que ser pagados. Cada una de estas actividades implica una serie - posiblemente un número significativo - de personas, además de los usuarios.

Cada uno de estos grupos de personas tiene sus propios requisitos, intereses que deben ser cumplidos por el sistema de software. Se refieren colectivamente a estas personas como **las partes interesadas** o **Stakeholders**. La comprensión del papel de la parte interesada es fundamental para entender el papel del arquitecto en el desarrollo de un producto de software o sistema [20].

***Definición:** Un **Stakeholder**, en la arquitectura de un sistema, es un individuo, equipo, organización, o clases de los mismos, que tienen un interés en la realización del sistema [20].*

METODOLOGÍA

El tipo de estudio realizado en este trabajo de grado es exploratorio, puesto que sus resultados constituyen una visión aproximada acerca de las arquitecturas de software basadas en componentes.

Los estudios exploratorios permiten aumentar el grado de familiaridad con temas desconocidos. Se busca analizar las actividades a realizar al definir una arquitectura basada en componentes al desarrollo de un software y a partir de los resultados obtenidos aplicarlos a la solución de una problemática.

El método de investigación está dividido en dos fases:

- La primera fase fue inductiva: En esta fase se identificó el concepto de arquitectura de software haciendo revisión de la literatura disponible sobre el tema objeto de este trabajo. La información recopilada fue analizada y contrastada con experiencias previas en otros proyectos realizados por los estudiantes.
- La segunda fase fue de síntesis: En esta fase se buscó consolidar el conocimiento adquirido en la fase anterior, aplicando los resultados obtenidos en dicha actividad al caso de estudio, definiendo así una arquitectura de software basada en componentes y realizando la implementación en el prototipo para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira.

RESULTADOS OBTENIDOS

Al finalizar este proyecto se cuenta con un documento que contiene:

1. Una descripción puntual y concisa de la arquitectura de software basado en componentes.
2. La identificación de los procesos que se llevan a cabo durante la definición de una arquitectura, para el desarrollo de un software basado en componentes.
3. El análisis de cada uno de estos procesos resaltando los puntos críticos.
4. Los artefactos de entrada para realizar la definición de la arquitectura de software basado en componentes, para el desarrollo del prototipo para el acueducto Ayacucho La Buitrera.
5. La indicación del proceso de diseño de la arquitectura de software basado en componentes aplicada al prototipo para el acueducto Ayacucho La Buitrera.
6. Los artefactos generados al definir una arquitectura de software prototipo para el acueducto Ayacucho La Buitrera.

A continuación se puede evidenciar mediante ilustraciones los resultados obtenidos en el prototipo para la gestión administrativa del acueducto rural Ayacucho, La Buitrera – Palmira:

Figura 19 Captura pantalla Inicio aplicación



Tomado de: Prototipo para la gestión administrativa del acueducto rural Ayacucho, La Buitrera – Palmira

Figura 20 Captura Pantalla Menús



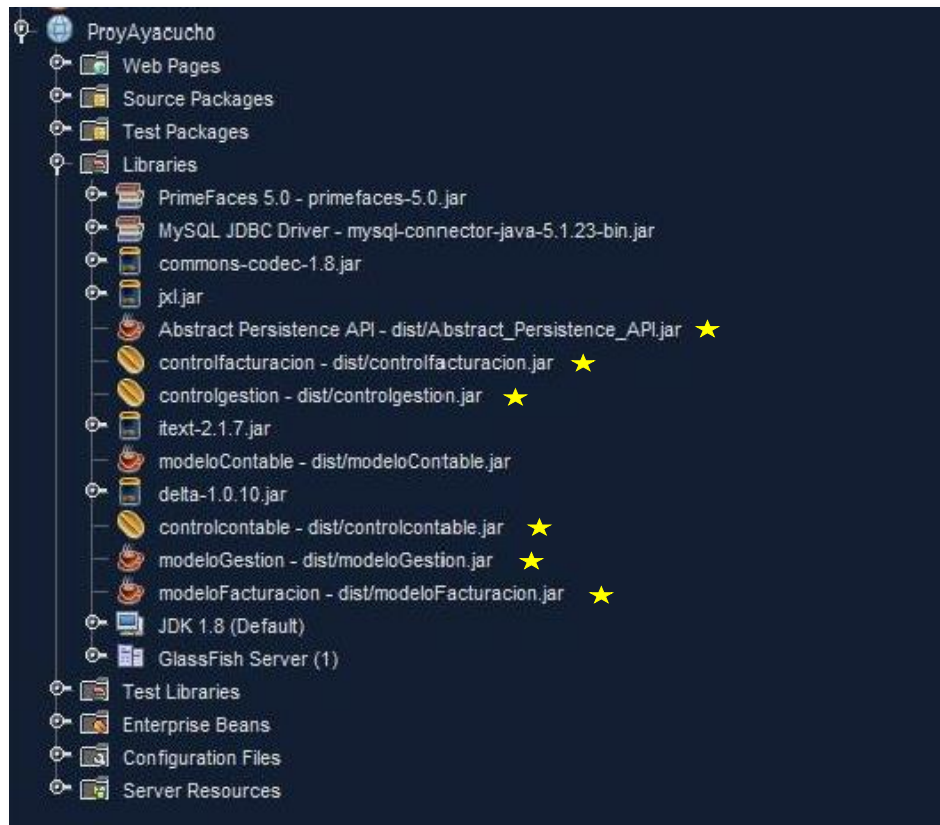
Tomado de: Prototipo para la gestión administrativa del acueducto rural Ayacucho, La Buitrera – Palmira

Figura 21 Captura pantalla Facturación



Tomado de: Prototipo para la gestión administrativa del acueducto rural Ayacucho, La Buitrera – Palmira

Figura 22 Estructura prototipo Ayacucho La Buitrera - Palmira



Tomado de: Prototipo para la gestión administrativa del acueducto rural Ayacucho, La Buitrera – Palmira

Comparación con otros desarrollos realizado por los estudiantes.

Teniendo en cuenta las actividades realizadas para el desarrollo de software usando la metodología aprendida en la academia, se pudo encontrar algunas similitudes y diferencias con lo aprendido en la realización de este trabajo. A continuación se describen algunas de ellas:

Tabla 2 Comparación con otros proyectos realizados

Otros Desarrollos	Basado en Componentes
La reutilización de subsistemas, se basa en copiar y pegar las clases y métodos que lo componen	Para reutilizar un subsistema basta con integrar el componente que lo describe.
La reutilización de código no ahorra costos y tiempo en el ciclo de desarrollo, debido a que este debe realizarse completamente	Para la reutilización de un componente este no se incorpora en el proceso de desarrollo, por lo tanto ahorra costos y tiempo.
Los ciclos de análisis son más cortos y no requieren de experiencia previa	Los ciclos de análisis son más largos y complejos y requieren que el equipo tenga conocimiento de los principios del desarrollo basado en componentes.
El acoplamiento con otras arquitecturas es complicado, y algunos casos no se pueden acoplar	Su acoplamiento con otras arquitecturas es relativamente sencillo.

No siempre se realiza la descripción de la arquitectura con la que contara el sistema y aun así se lleva acabo su implementación	Requiere la descripción de la arquitectura para identificar como estarán compuestos los componentes y su comportamiento en el sistema.
--	--

CONCLUSIONES

- Disponer de una arquitectura permite tener una visión general del software a desarrollarse. Cuando se desarrollan proyectos con características similares, se puede reutilizar las decisiones que se llevaron a cabo durante la definición de la arquitectura, por ello, todas las consecuencias de las decisiones tomadas se transfieren al proyecto.
- Al disponer de diversos enfoques, durante el análisis y diseño del software llena de mucha satisfacción los diferentes resultados que se pueden obtener. El desarrollo de software basado en componentes nos plantea ahora la posibilidad de pensar en crear o reutilizar los diferentes elementos que componen el software a desarrollarse. Con la experiencia obtenida, en el enfoque dado por la academia para los diversos proyectos realizados y el conocimiento adquirido durante la realización de este proyecto, el equipo cuenta ahora con una visión más amplia para plantear soluciones más eficientes, orientadas a la reutilización y a la separación de asuntos.
- El proceso de análisis en el desarrollo basado en componentes es muy complejo, y requiere de cierta experiencia en el área, haciéndolo más demorado y costoso en comparación al realizado en otras arquitecturas. Para obtener los beneficios de los componentes, el equipo debe contar con la experiencia para separar los asuntos del problema y prever su reutilización en el futuro, y así poder desarrollar componentes que cumplan con las características establecidas.
- Al desarrollar un componente se deben sortear los problemas típicos al desarrollo de software basado en componentes (clarividencia, evolución, particularización, entre otros), haciendo que sea más costoso si solo se tiene en cuenta para un proyecto, es decir, si el componente a desarrollar sólo será usado en ese proyecto, se realizarán actividades fundamentales para el desarrollo basado en componentes que no son esenciales o necesarias en otros tipos de desarrollo aumentado así el tiempo y el costo que tendrá el proyecto.
- El desarrollo basado en componentes, por ser una disciplina tan joven, aún no cuenta con un amplio mercado para la comercialización de componentes, dejando así al equipo de desarrollo con la opción de usar los componentes desarrollados internamente.
- Para ampliar el uso de la reutilización y mejorar la calidad de los resultados obtenidos, los equipos de desarrollo deberían conocer los principios del desarrollo basado en componentes. Principios que permiten hacer separación de asuntos, facilitando así el mantenimiento del sistema. Además, permite identificar subsistemas que podrían *“tomarse del estante”* y ahorrarse el costo que implica desarrollarlos.
- Al finalizar el desarrollo del prototipo para la gestión administrativa del acueducto rural Ayacucho, La Buitrera - Palmira, se analizaron los componentes desarrollados y nos encontramos con la dificultad para integrar un nuevo componente al proyecto, esto se presentó debido a la falta de experiencia para diseñar los componentes y la dependencia que tendrían entre ellos. Por ello, es importante que todo el equipo que participa en el

desarrollo del proyecto cuenta con diversos recursos y enfoques para tomar mejores decisiones que enriquezcan el software a desarrollarse.

REFERENCIA BIBLIOGRÁFICA

- [1] Scrum Manager (2013). Crisis del software. [En Línea]. Disponible: http://www.scrummanager.net/bok/index.php?title=Crisis_del_software Accedido: 22 Abril de 2015
- [2] I. Sommerville, Ingeniería del software. Séptima Edición. Reino Unido: Pearson Addison Wesley, 2005.
- [3] C. Herrera. ¿Hacia dónde va el futuro de la arquitectura de TI en Argentina?, 2014 <http://searchdatacenter.techtarget.com/es/cronica/Hacia-donde-va-el-futuro-de-la-arquitectura-de-TI-en-Argentina> Accedido: 22 Abril de 2015
- [4] J. Sametinger. Software engineering with reusable components. Primera Edición. Springer Verlag, Agosto 1997.
- [5] M. Fowler. Who Needs an Architect?. IEEE SOFTWARE [En Línea]. 2003. Disponible: <http://martinfowler.com/ieeeSoftware/whoNeedsArchitect.pdf> Accedido: 29 Abril de 2015
- [6] Meyer. Object-Oriented Software Construction. Segunda Edición. 1998
- [7] R. S. Pressman. Ingeniería de software: un enfoque práctico. Séptima Edición. New York: McGraw-Hill, 2010.
- [8] D. Garlan, M. Shaw. An Introduction to Software Architecture [En Línea]. World Scientific Publishing Company, New Jersey, 1993. Disponible: http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf Accedido: 29 Abril de 2015
- [9] EISC, Escuela de Ingeniería y Computacion. Universidad del Valle. [En Línea]. Disponible: <http://ingenieria.univalle.edu.co/pregrados/sistemas.php> Accedido: 15 Mayo de 2015
- [10] J.C. Terreros. Desarrollo de Software basado en Componentes. Microsoft. [En Línea]. Disponible: <https://msdn.microsoft.com/es-es/library/bb972268.aspx> Accedido: 24 Abril de 2015

- [11] F. Lüunders. Use of Component-Based Software Architectures in Industrial Control Systems. Department of Computer Science and Engineering. Mälardalen University Licentiate Thesis. No. 18, 2003. [En Línea] Disponible: https://scholar.google.se/citations?view_op=view_citation&hl=sv&user=kpAoMLIAAAAJ&citation_for_view=kpAoMLIAAAAJ:d1gkVwhDpl0C Accedido: 27 Abril de 2015
- [12] L. Fuentes; J.M Troya y A. Vallecillo. Desarrollo de Software Basado en Componentes. Depto. Lenguajes y Ciencias de la Computación. Universidad de Málaga. [En Línea] Disponible: <http://www.lcc.uma.es/~av/Docencia/Doctorado/tema1.pdf> Accedido: 12 Abril de 2015
- [13] Wikipedia. Libro diario (Contabilidad). [En Línea]. Disponible: http://es.wikipedia.org/wiki/Libro_Diario_%28Contabilidad%29 Accedido: 15 Mayo de 2015
- [14] Gerencie.com ¿Qué es la provisión de cartera?. [En Línea]. Disponible: <http://www.gerencie.com/que-es-la-provision-de-cartera.html> Accedido: 15 Mayo 2015
- [15] Debitoor. Definición de conciliación bancaria. [En Línea]. Disponible: <https://debitoor.es/definicion-de-conciliacion-bancaria> Accedido: 15 Mayo 2015
- [16] Debitoor. Definición de nómina. [En Línea]. Disponible: <https://debitoor.es/glosario/definicion-nomina> Accedido: 15 Mayo 2015
- [17] P. Naur, B. Randalls. Software Engineering: A Report on a Conference Sponsored by the NATO Science Committee. NATO. 1969. [En Línea]. Disponible: <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF> Accedido: 16 Mayo de 2015
- [18] L. Bass, P. Clements y R. Kazman. Software Architecture in Practice. 2a. ed., Addison-Wesley, 2003.
- [19] DanimaniaARQISOFT. El proceso de diseño de la arquitectura de software. [En Línea]. Disponible: http://danimaniarqsoft.com/?page_id=193 Accedido: 15 Mayo de 2015
- [20] N. Rozanski, E. Woods. Software System Architectures. "Stakeholders". [En línea]. Disponible: <http://www.viewpoints-and-perspectives.info/home/stakeholders/> Accedido: 15 Mayo 2015
- [21] Object Management Group. OMG Unified Modeling Language Specification, versión 1.5. Marzo 2003
- [22] Philippe Kruchten. The *Rational Unified Process: An introduction*, tercera edición. Addison-Wesley Professional. 2003

- [23] C. Canal. Un Lenguaje para la Especificación y Validación de Arquitecturas de Software. Tesis Doctoral, Universidad de Málaga. Diciembre 2000 [En Línea]. Disponible: http://www.lcc.uma.es/~canal/papers/tesis/canal_tesis.pdf Accedido: 20 de Octubre de 2015
- [24] C. Reynoso, N. Kicillof. Lenguajes de Descripción de Arquitectura (ADL), Versión 1.0. Marzo 2004. [En línea]. Disponible: <http://carlosreynoso.com.ar/archivos/arquitectura/ADL.PDF> Accedido: 20 Octubre de 2015.
- [25] P. Kruchten. Architectural Blueprints—The “4+1” View Model of Software Architecture. Paper published in IEEE Software 12 (6). Noviembre 1995. [En Línea]. Disponible: <http://www.cs.ubc.ca/~gregor/teaching/papers/4+1view-architecture.pdf> Accedido: 25 Octubre de 2015
- [26] Object Management Group. Unified Modeling Language™ (UML®) Resource Page. Mayo del 2015. [En Línea]. Disponible: <http://www.uml.org/> Accedido: 25 Octubre de 2015
- [27] KDE Documentation. [En Línea]. Disponible: <https://docs.kde.org/trunk4/es/kdesdk/umbrello/uml-basics.htm> Accedido: 27 Octubre de 2015
- [28] Sparx Systems. [En Línea]. Disponible: <http://www.sparxsystems.com.ar/resources/> Accedido: 27 Octubre de 2015.
- [29] I. Jacobson, I. Spence, K. Bittner. Ivar Jacobson International SA. Use Case 2.0. Agosto 2013. [En Línea]. Disponible: https://www.ivarjacobson.com/sites/default/files/field_iji_file/article/use_case_2.0_-_spanish_translation.pdf Accedido: 1 Noviembre de 2015
- [30] D. Garlan, M. Shaw. An Introduction to Software Architecture [En Línea]. World Scientific Publishing Company, New Jersey, 1993. Disponible: http://www.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf Accedido: 29 Abril de 2015.
- [31] L. Chung, B. Nixon, E. Yu, J. Mylopoulos. "Non-Functional Requirements in Software Engineering". Kluwer Academic Publisher, 2000.
- [32] C. Szyperski. "Component Software: Beyond Object-Oriented Programming". Addison-Wesley, 1999.

[33] B. Abraham, J. Aguilar, E. Leiss. "Middleware for Improving Security in a Component Based Software Architecture". Networked Computing and Advanced Information Management, IEEE Computer Society, 2008. Vol. 1. pp.502-509.

[34] P. Turmero. Componentes [En Línea]. Disponible: <http://www.monografias.com/trabajos16/componentes/componentes.html> Accedido: Agosto 2015.

[35] F. Bachman. "Technical Concepts of Component-Based Software Engineering". Software Engineering Institute, Volume II. Carnegie Mellon Internal Research and Development. 2000.

[36] P. Herzum, O. Sims. "Business component factory: A comprehensive overview of component-based development for the enterprise". John Wiley & Sons, Primera Edición 2000.

[37] A. Brown, K. Wallnau. The current state of CBSE. IEEE Software, 1998. Vol. 15, pp. 37-46

[38] Enterprise JavaBeans (EJB). Disponible:

<http://searchsoa.techtarget.com/definition/Enterprise-JavaBeans> Accedido: 18 Marzo de 2016

[39] MSDN. component object model [En Línea]. Disponible: <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/componentobjectmodelanchor.asp> Accedido: 5 diciembre de 2016.

[40] Rick Kazman, Len Bass, Paul Clements, "Software Architecture in Practice", Addison-Wesley Professional. Third Edition 2012.

[41] C. Szyperski, C. Pfister, Summary of the Workshop on Component Oriented Programming (WCOP'96) 1996.

[42] O. Nierstrasz. Requirements for a Composition Language. En ECOOP'94 1995.

[43] C. Szyperski, Component Software: Beyond Object-Oriented Programming. Addison Wesley 1998.

[44] ISO/IEC/IEEE 42010. Systems and software engineering - Architecture description [En Línea]. First Edition. 2011. Disponible en: <http://cabibbo.dia.uniroma3.it/asw/altrui/iso-iec-ieee-42010-2011.pdf> Accedido: Marzo de 2016

[45] International Business Machines Corp (IBM). Rational Unified Process Best Practices for Software Development Teams [En Línea]. Disponible en:

https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf Accedido: Marzo de 2016

[46] Rational Software Corporation. Rational Unified Process: Overview. [En Línea]. Disponible en: <http://sce.uhcl.edu/helm/rationalunifiedprocess/> Accedido: Marzo de 2016

[47] P. Kruchten. The Rational Unified Process: An Introduction. Addison Wesley. Second Edition 2000.

[48] M. Ariza, J. Molina. Introducción y principios básicos del desarrollo basado en componentes. [En línea]. Disponible en: <http://pegasus.javeriana.edu.co/~jcpymes/Docs/DSBC.pdf> Accedido: Marzo de 2016

[49] J. Pavón. El patron Modelo – Vista - Controlador (MVC). [En Línea]. Disponible en: <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.MVC.pdf> Accedido: Marzo de 2016

**Sistema para la gestión
administrativa del acueducto rural
Ayacucho La Buitrera, Palmira**
Documento arquitectura de software

**Versión 2.0
08/12/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
30/Abril/2016	1.0	Elaboración del documento	Derlyn Johanna Lourido Quintero
8/Dic/2016	2.0	Correcciones	Hernan Darío Gonzalez Patiño

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

Tabla de contenido

1. Introducción	1
1.1. Propósito	1
1.2. Alcance	1
1.3. Definiciones, acrónimos y abreviaciones.....	1
2. Representación de la arquitectura	1
3. Objetivos y restricciones	2
4. Vistas de casos de uso	2
4.1. Introducción	2
4.2. Identificación de los Casos de Uso relevantes para la arquitectura	2
4.3. Descripción de los Casos de uso relevantes para la arquitectura	3
4.3.1. Gestionar Acometidas	4
4.3.2. Gestionar Facturación	4
4.3.3. Gestionar Recaudo	4
5. Vista Lógica.....	4
5.1. Introducción	4
5.2. Definición de los componentes	5
5.3. Descripción de los componentes.....	5
5.4. Diseño de los componentes	7
5.4.1. Componente: Control Gestión.....	9
5.4.2. Componente: Control Facturación	10
5.4.3. Componente: Control Contable.....	11
5.4.4. Componente: Modelo Gestión	7
5.4.5. Componente: Modelo Facturación	7
5.4.6. Componente: Modelo Contable	8
5.4.7. Componente: ProyAyacucho	12
5.5. Realización de los casos de uso relevantes para la arquitectura	12
5.5.1. Gestión Acometidas.....	12
6. Vista de despliegue	17
6.1. Introducción	17

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

6.2. Distribución y despliegue	17
7. Referencias.....	18

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

DOCUMENTO ARQUITECTURA DE SOFTWARE

1. Introducción

1.1. Propósito

El Documento de Arquitectura de Software presenta la arquitectura del sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera a través de diferentes vistas, cada una de las cuales ilustra un aspecto en particular del software desarrollado. Se pretende de esta forma que el documento brinde al lector una visión global y comprensible del diseño general del sistema desarrollado.

1.2. Alcance

El documento se centra en el desarrollo de la vista lógica del sistema. Se incluyen los aspectos fundamentales del resto de las vistas y se omiten aquellas que no se consideren pertinentes como ser el caso de la vista de procesos.

En cuanto a los componentes externos que se mencionen, se incluye una descripción de los mismos en el nivel que se considere apropiado y se indican las referencias donde consultar más información sobre los mismos.

1.3. Definiciones, acrónimos y abreviaciones

Ver glosario [1].

2. Representación de la arquitectura

El modelo propuesto por RUP [2] para representar la arquitectura utiliza el siguiente conjunto de vistas:

- Vista de Casos de Uso: lista los casos de uso o escenarios del modelo de casos de uso que representen funcionalidades centrales del sistema final, que requieran una gran cobertura arquitectónica o aquellos que impliquen algún punto especialmente delicado de la arquitectura.
- Vista Lógica: describe las partes arquitectónicamente significativas del modelo de diseño, como ser la descomposición en capas, subsistemas o paquetes. Una vez presentadas estas unidades lógicas principales, se profundiza en ellas hasta el nivel que se considere adecuado.
- Vista de Procesos: describe la descomposición del sistema en hilos y procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican.
- Vista de Despliegue: describe uno o más escenarios de distribución física del sistema sobre los cuales se ejecutará y desplegará del mismo. Muestra la comunicación entre los diferentes nodos que componen los escenarios antes mencionados, así como el mapeo de los elementos de la Vista de Procesos en dichos nodos.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

- Vista de Implementación: describe la estructura general del Modelo de Implementación y el mapeo de los subsistemas, paquetes y clases de la Vista Lógica a subsistemas y componentes de implementación.
- Vista de Datos: describe los elementos principales del Modelo de Datos, brindando un panorama general de dicho modelo en términos de tablas, vistas, índices, etc.

3. Objetivos y restricciones

El prototipo para la gestión administrativa cumple con el modelo propuesto en el documento de requerimientos [3]. Dentro de esta base de desarrollo son propiedades esenciales para la arquitectura a definir:

- Diseño basado en componentes de propósito claro y concreto y con alto grado de cohesión
- Desacoplamiento entre componentes que permita el fácil reemplazo de los mismos
- Componentes altamente reutilizables

4. Vistas de casos de uso

4.1. Introducción

La Vista de Casos de Uso presenta un subconjunto del Modelo de Casos de Uso. Describe los casos de uso o escenarios que representen funcionalidades centrales del sistema final, que requieran una gran cobertura arquitectónica o aquellos que impliquen algún punto especialmente delicado de la arquitectura. Estos casos de uso, en conjunto con los requerimientos no funcionales, permiten descubrir y diseñar la arquitectura del sistema.

4.2. Identificación de los Casos de Uso relevantes para la arquitectura

Para el diseño del sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, se identifican como los casos de uso relevantes desde el punto de vista de la arquitectura, los abajo mencionados:

- **Gestionar Acometidas:**
Gestiona la información del punto central del acueducto, las acometidas.
Permite registrar, modificar, consultar y eliminar la información correspondiente a la acometida propia y a los servicios que se le presta a esta.
- **Realizar Facturación:**
Gestiona el proceso de generar la facturación de los servicios prestados a las acometidas instaladas.
Genera un documento con la totalidad de las facturas realizadas en el mes ordenadas según la distribución asignada por el personal a cargo de esta operación. También permite generar copia de una factura seleccionada del mes o periodo actual.

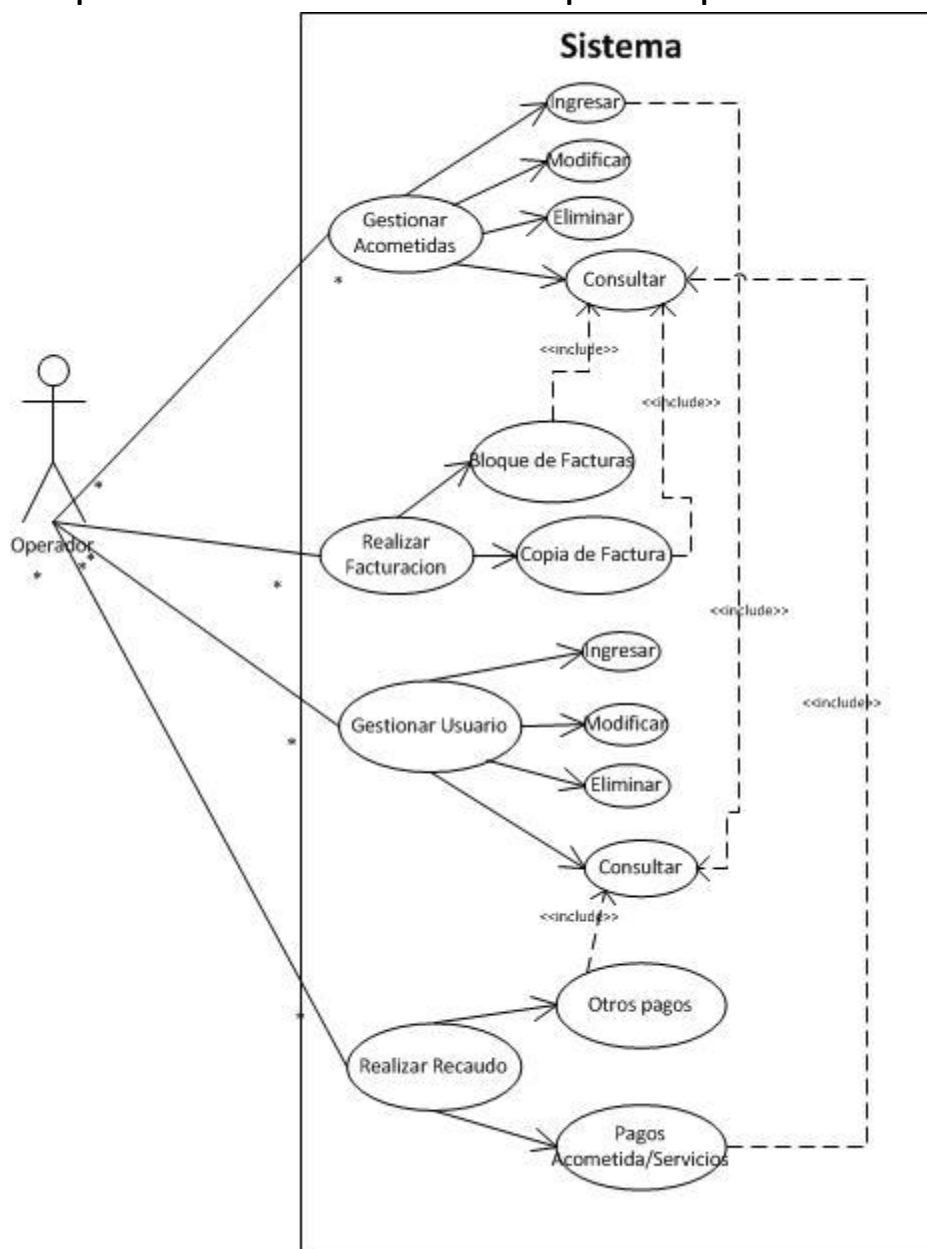
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

- **Realizar Recaudo:**

Gestiona el proceso de recaudar el dinero por los servicios prestados y/o derecho a las acometidas y otros tipos de ingreso con los que pueda contar el acueducto.

Cada movimiento realizado depende de una cuenta contable asignada, por ello es indispensable que el operario asigne las cuentas correctas al movimiento que está realizando.

4.3. Descripción de los Casos de uso relevantes para la arquitectura



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

4.3.1. Gestionar Acometidas

Nombre	Gestionar Acometidas
Actores	Operario
Sinopsis	El caso de uso comienza cuando el operario indica al sistema que desea gestionar la información registrada de las acometidas. El sistema proporciona a tales efectos, una lista total de las acometidas registradas y un formulario con la información básica necesaria para realizar la gestión de las acometidas. El operario puede realizar las siguientes acciones: • Registrar una nueva acometida • Modificar una acometida seleccionada • Eliminar una acometida seleccionada • Consultar una acometida seleccionada

4.3.2. Realizar Facturación

Nombre	Realizar Facturación
Actores	Operario
Sinopsis	Mensualmente debe realizarse el proceso de facturar los servicios prestados a la totalidad de las acometidas instaladas. Para ello el operario debe estar al día con el orden de distribución de las facturas y procede a realizar la generación de la facturación activando el proceso. El sistema automáticamente guarda la información correspondiente al bloque de facturas generado. El operario puede realizar dos acciones posteriores a la facturación: 1). Generar un archivo con la información del total de las facturas del periodo actual en el orden de distribución asignado. 2) Realizar copia de una factura seleccionada.

4.3.3. Realizar Recaudo

Nombre	Realizar Recaudo
Actores	Operario
Sinopsis	El acueducto recibe durante el mes los dineros recaudados por los conceptos de servicios prestados a la comunidad. A si mismo, existen otros tipos de ingresos al acueducto, los cuales también deben tener su registro diario. El caso de uso inicia cuando el operario lanza el proceso de recaudo. Seguido debe seleccionar el tipo de recaudo a generar e ingresar la información solicitada según sea el caso. Al finalizar el sistema registra automáticamente la información del movimiento realizado.

5. Vista Lógica

5.1. Introducción

Se presentan en este punto los sucesivos refinamientos que definen las diferentes unidades lógicas que componen la arquitectura del sistema para la gestión administrativa rural Ayacucho La Buitrera, Palmira.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

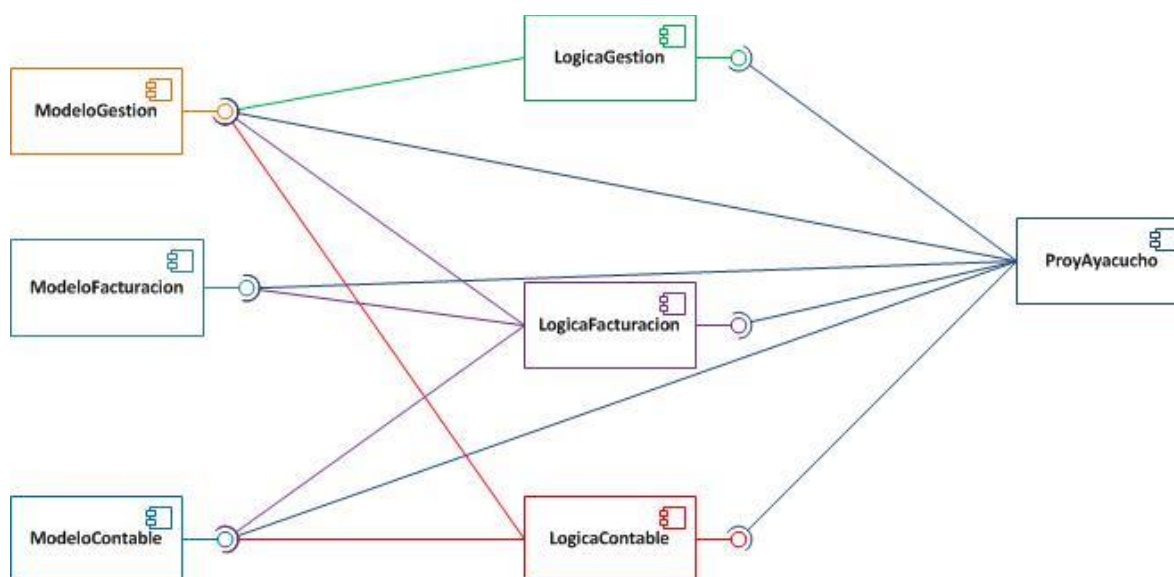
El primer refinamiento realizado consiste en la descripción de los componentes.. Cada componente consiste en el agrupamiento de diferentes funcionalidades relacionadas entre sí y posee la capacidad de funcionar como un sistema en sí mismo.

Posteriormente se explora la composición de cada uno de los componentes.

Finalmente se incluye la realización de los casos de uso descritos en la sección anterior mediante los componentes arquitectónicos definidos.

5.2. Definición de los componentes

El desarrollo basado en componentes permite identificar subsistemas que pueden ser reutilizables en otros proyectos que estén dentro del dominio del proyecto a realizarse.



5.3. Descripción de los componentes

- **Modelo Gestión:** Este componente debe contener las entidades relacionadas a la administración del acueducto. Debe permitir una manipulación sencilla al sistema de la información relacionada a:
 - Acometida
 - Suscriptor
 - Zonas
 - Concepto – Acometida – Tarifa
 - Concepto
 - Tipo Concepto
 - Tipo Tarifa
 - Tarifa
 - Tipo Documento

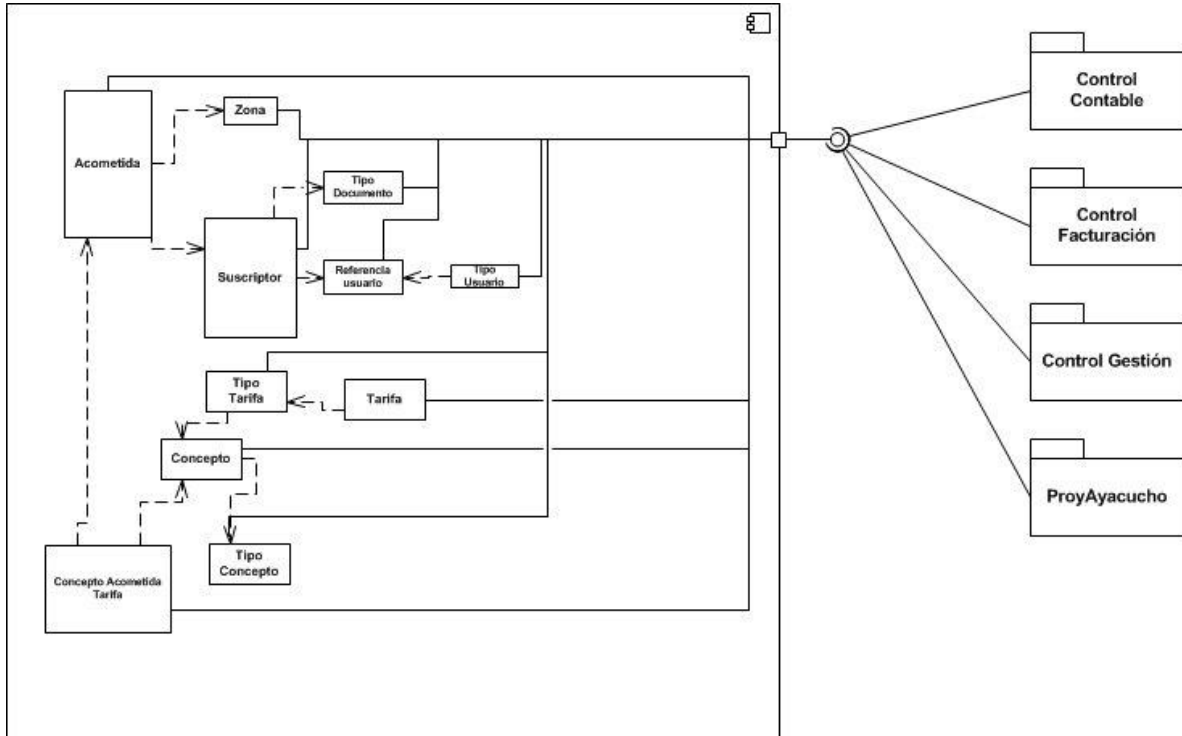
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

- Referencia usuario
- Tipo Usuario
- **Modelo Facturación:** Este componente debe contener las entidades relacionadas a la facturación. Debe permitir al sistema la manipulación de la información correspondiente a:
 - Factura
 - Detalle de la factura
 - Fecha Registro Facturas
- **Modelo Contable:** Este componente debe contener las entidades relacionadas a la gestión contable. Debe permitir al sistema la manipulación de la información correspondiente a:
 - Clase
 - Grupo
 - Cuenta
 - Subcuenta
 - Concepto – Subcuenta
 - Concepto – Documentos
 - Tipo Documento Contable
 - Movimiento
- **Control Gestión:** Este componente es el encargado de proporcionar las herramientas adecuadas que le permitan al sistema manipular la información almacenada en la base de datos, relacionada con la información administrativa propia del acueducto. Debe garantizar una comunicación segura con la base de datos.
- **Control Facturación:** Este componente debe permitirle al sistema manipular la información almacenada en la base de datos, correspondiente al proceso de facturación de los servicios prestados mensualmente a las acometidas instaladas. Debe garantizar una comunicación segura con la base de datos.
- **Control Contable:** Este componente debe permitirle al sistema la manipulación de la información almacenada en la base de datos, correspondiente a los movimientos contables realizados en el acueducto. Debe garantizar una comunicación segura con la base de datos.
- **ProyAyacucho:** Este es el proyecto principal, el cual consume todos los componentes antes mencionados y se encarga de realizar el control de las vistas que se le presentan al usuario final.

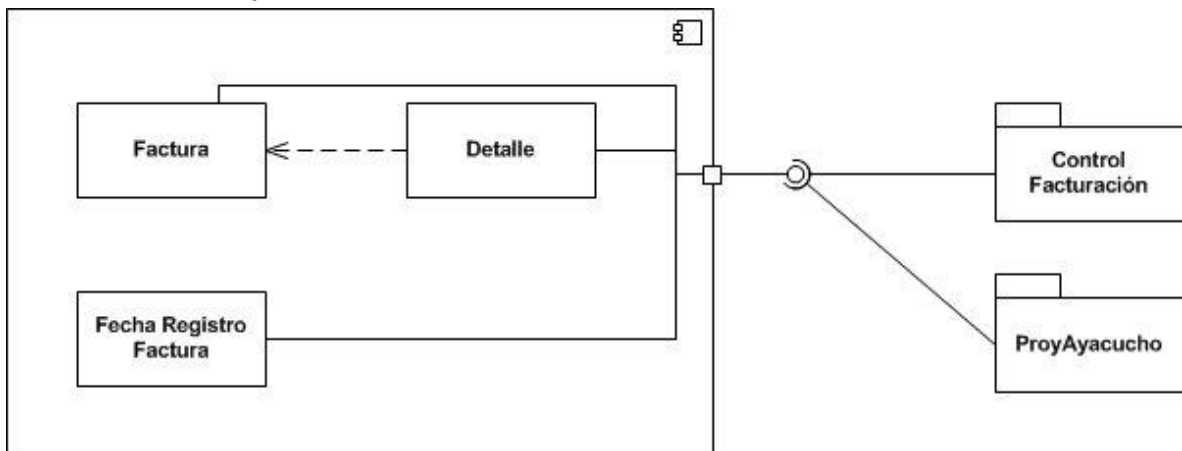
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

5.4. Diseño de los componentes

5.4.1. Componente: Modelo Gestión

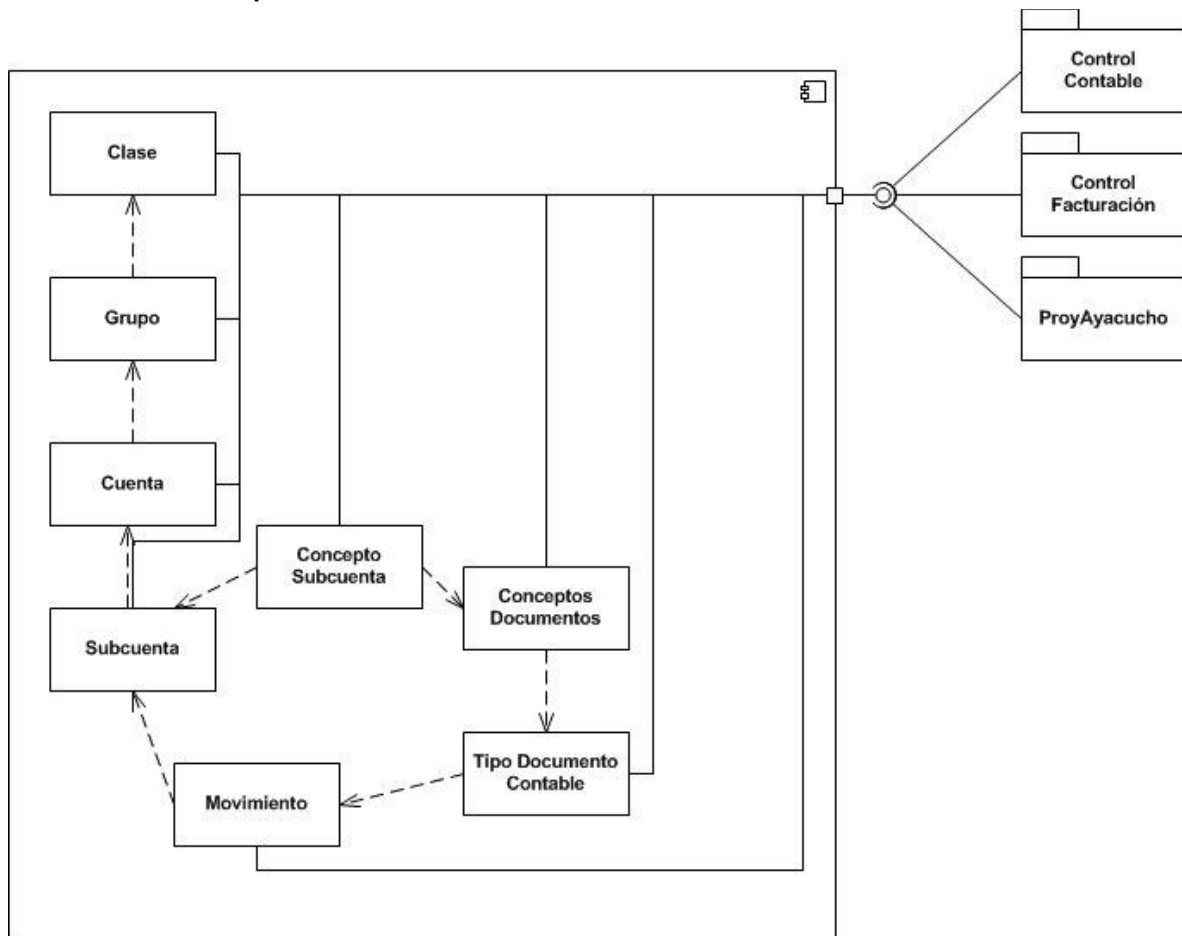


5.4.2. Componente: Modelo Facturación



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

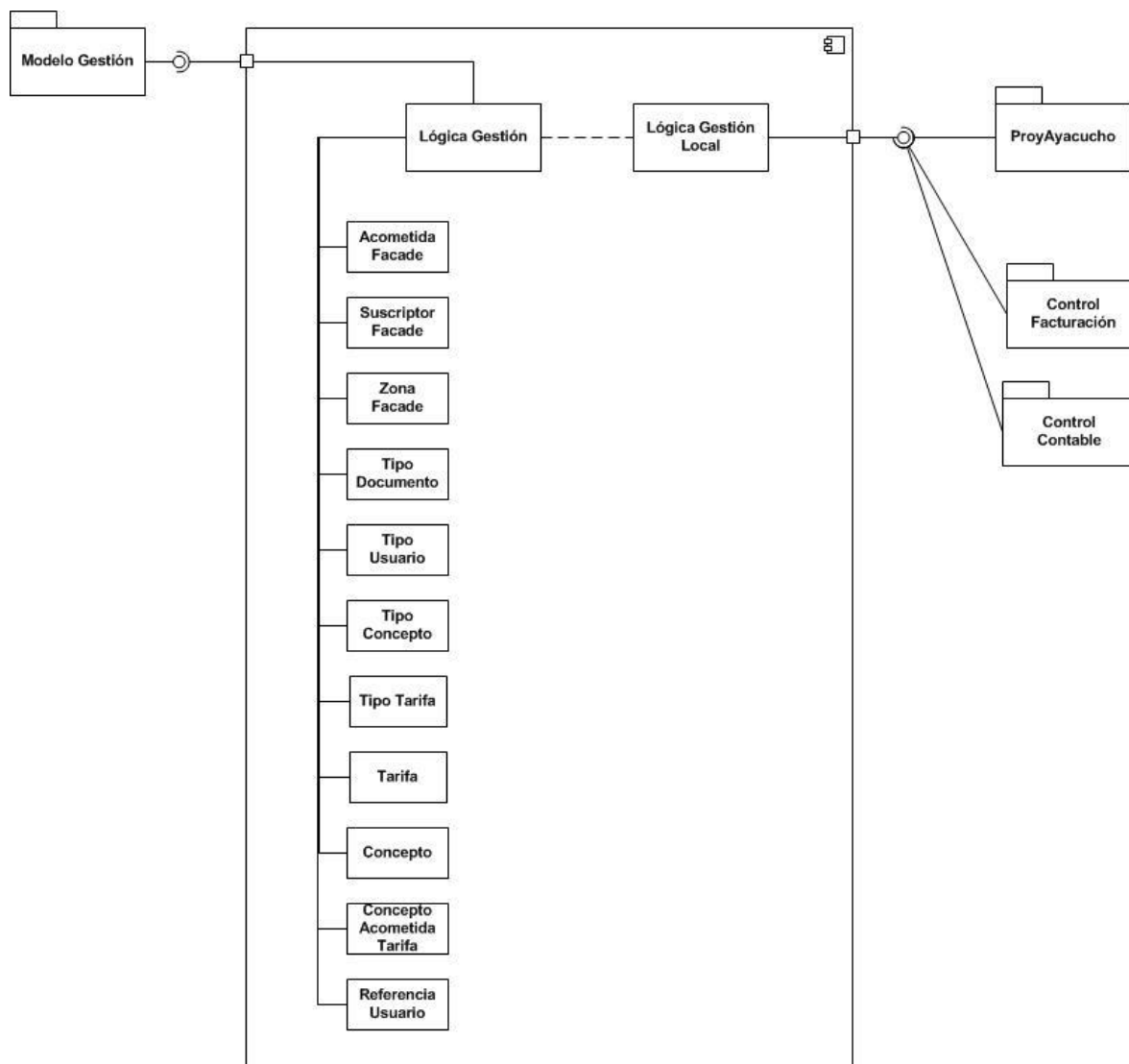
5.4.3. Componente: Modelo Contable



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

5.4.4. Componente: Control Gestión

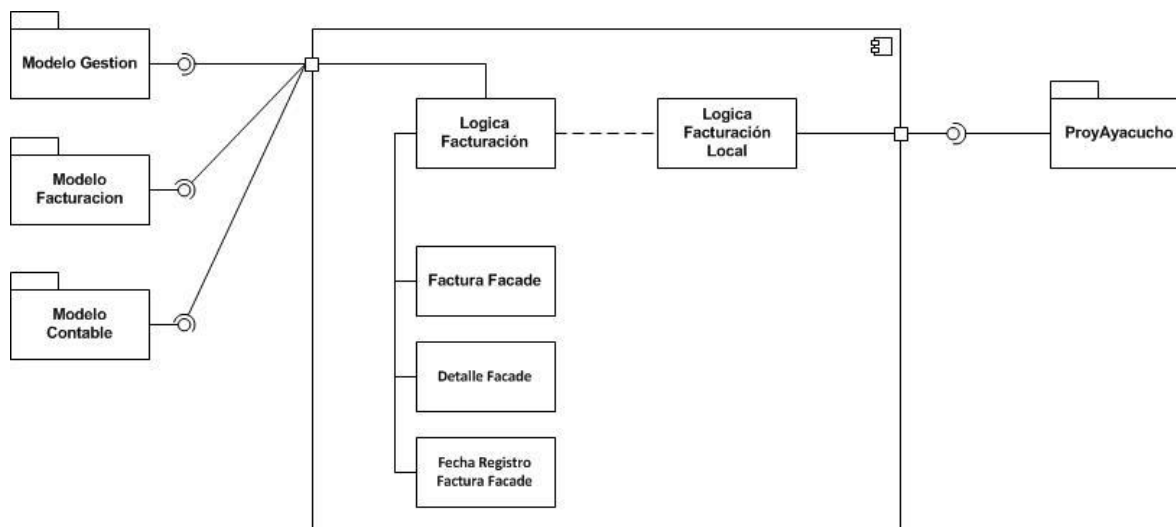
Este componente requiere de los servicios ofrecidos por el componente **Modelo Gestión** y así poder realizar de manera más sencilla la manipulación de la información administrativa del acueducto, Figura xx.



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

5.4.5. Componente: Control Facturación

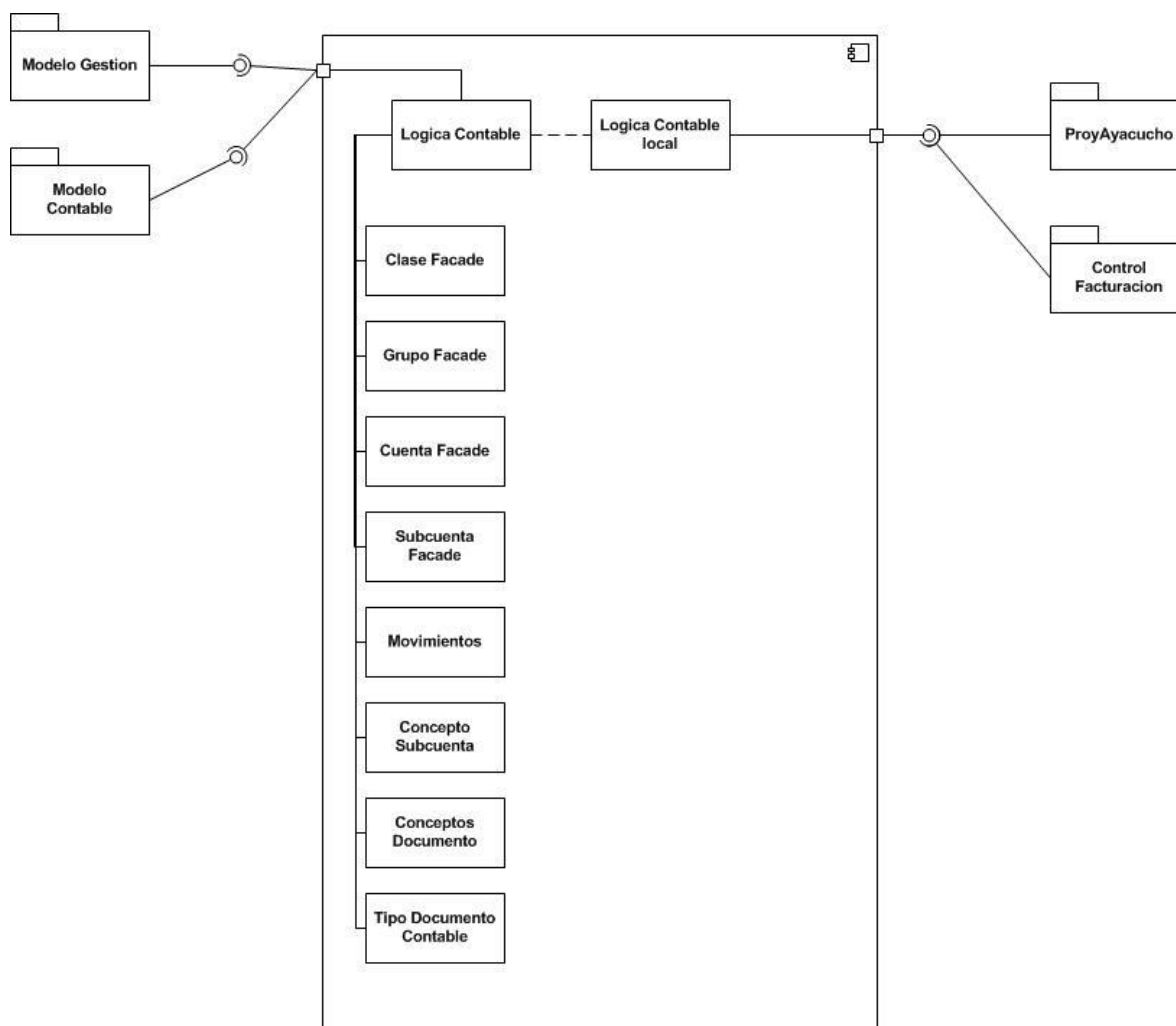
Este componente requiere de los servicios ofrecidos por los componentes: **Modelo Facturación**, **Modelo Gestión**, **Modelo Contable**.



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

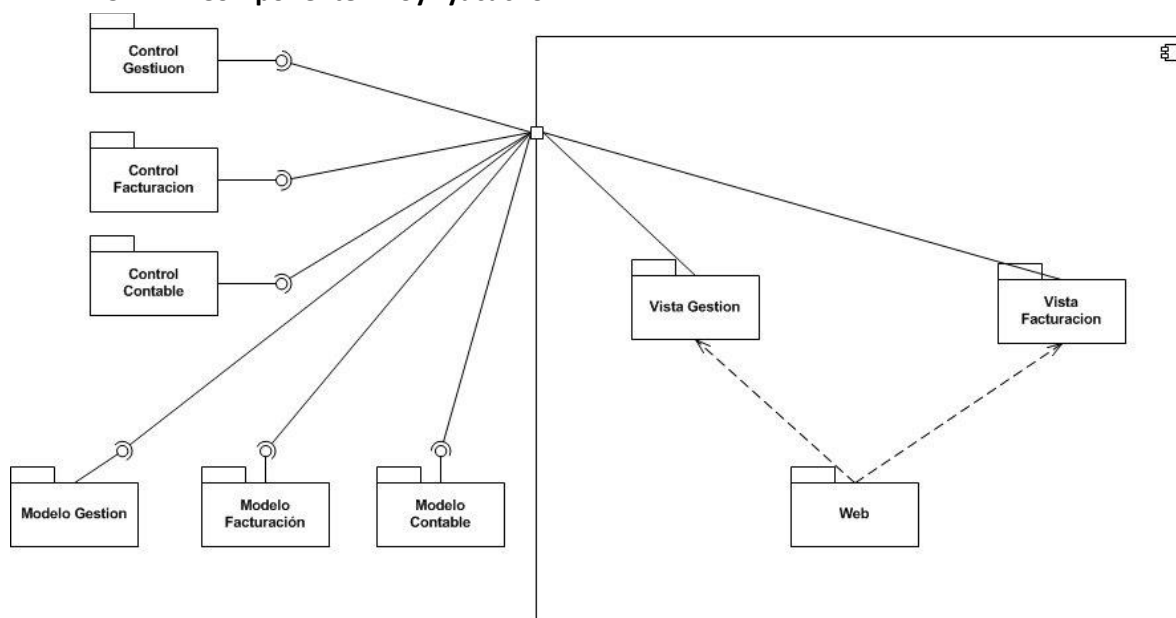
5.4.6. Componente: Control Contable

Este componente requiere de los servicios ofrecidos por los componentes: **Modelo Gestión, Modelo Contable.**



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

5.4.7. Componente: ProyAyacucho



5.5. Realización de los casos de uso relevantes para la arquitectura

5.5.1. Gestión Acometidas

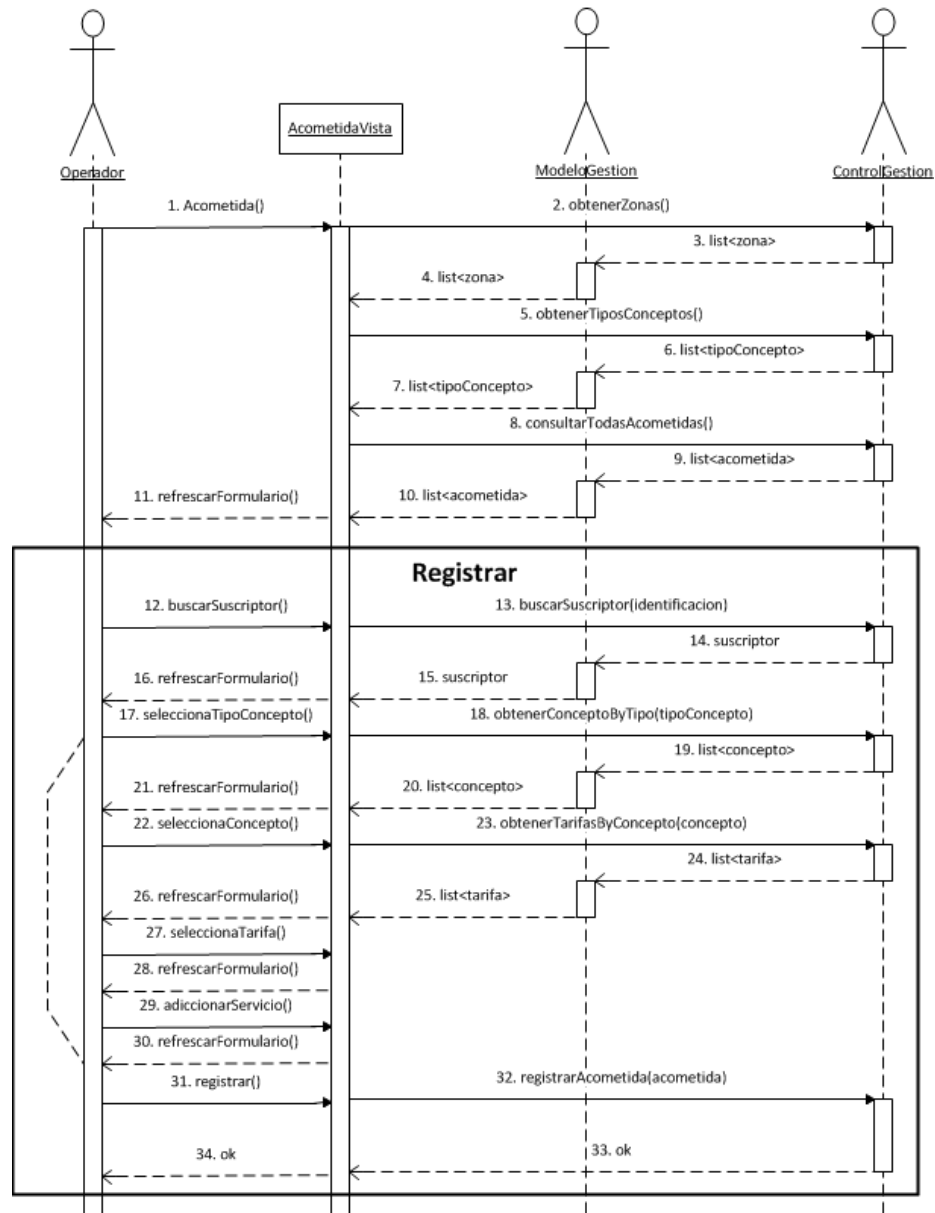
A continuación se describe las secuencias para la comunicación entre componentes que se llevan a cabo para la gestión de las acometidas

- **Gestionar Acometida (ProyAyacucho)**

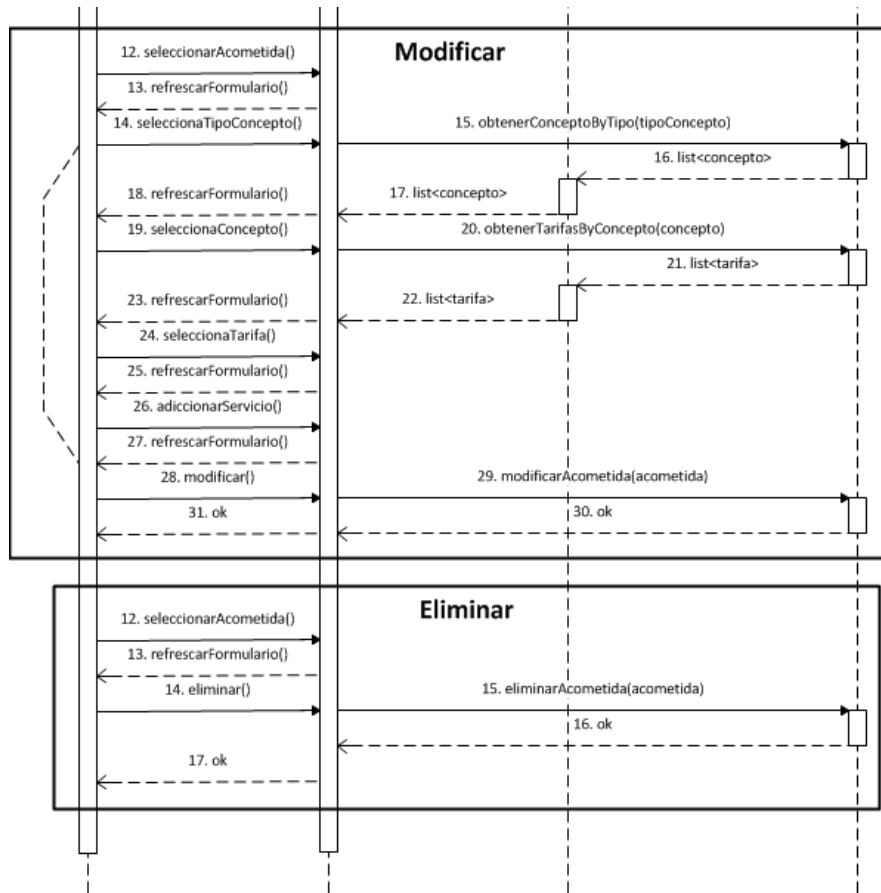
Descripción del proceso que se realiza para gestionar las acometidas en el sistema ProyAyacucho.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

DS-03 Acometida (ProyAyacucho)



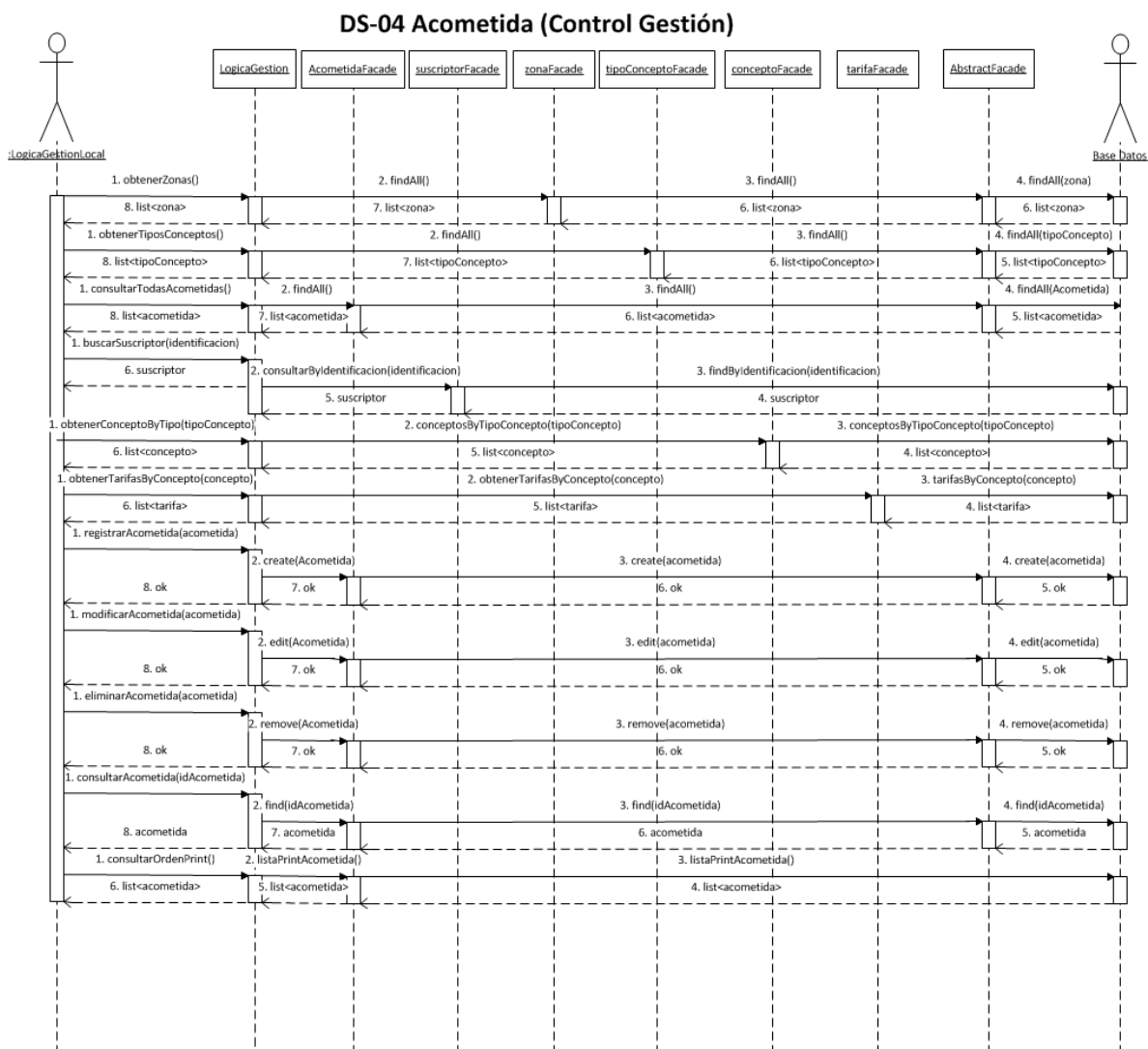
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016



- **Gestionar Acometida (Control Gestión)**

Descripción del proceso de gestionar las acometidas desde el componente Control Gestión.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016



5.5.2. Gestionar Facturación

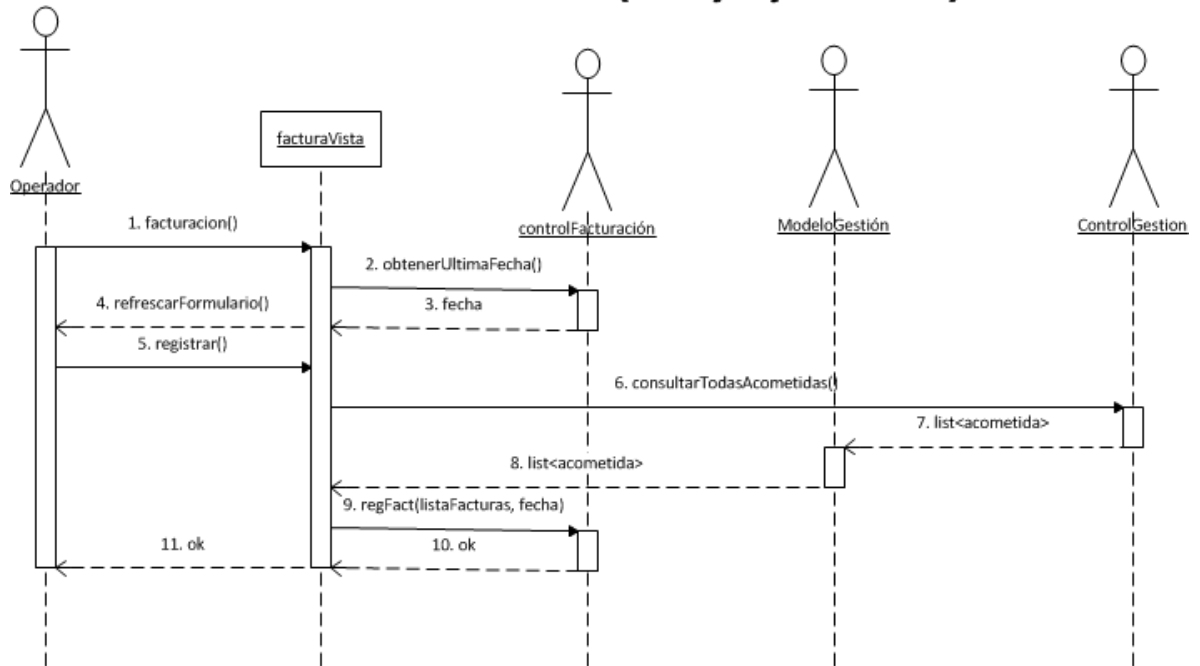
A continuación se describe las secuencias para la comunicación entre componentes que se llevan a cabo para la gestión de la facturación

- **Gestionar Facturación (ProyAyacucho)**

Descripción del proceso que se realiza para gestionar la facturación en el sistema ProyAyacucho.

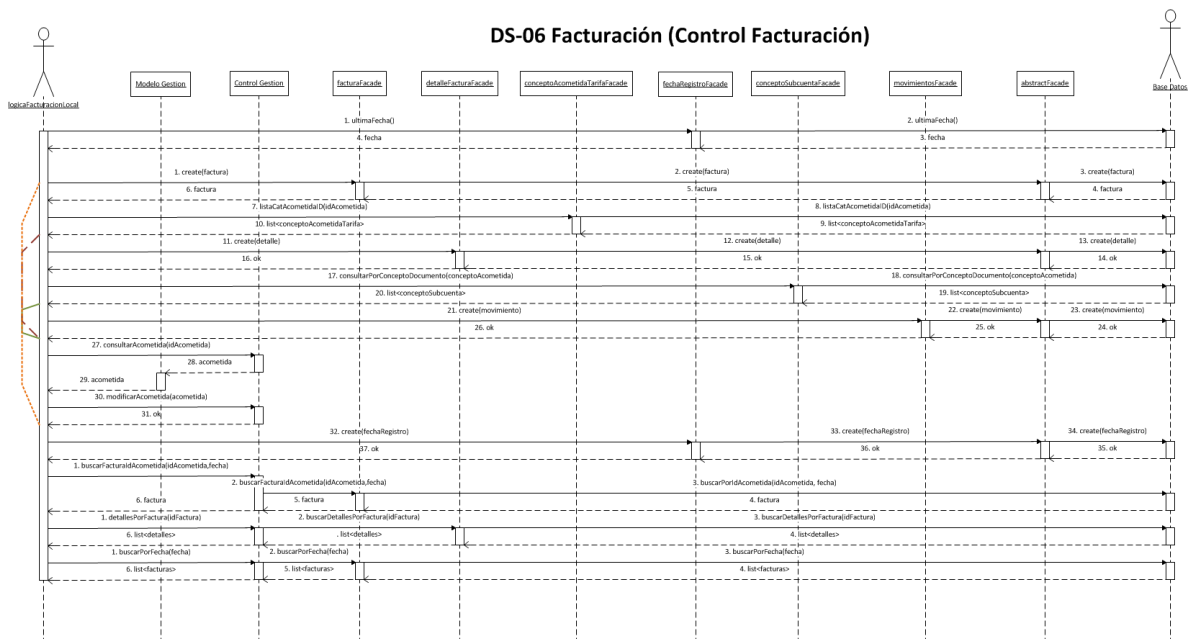
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

DS-05 Facturación (ProyAyacucho)



- Gestionar Facturación (Control Facturación)**

Descripción del proceso de gestionar la facturación desde el componente Control Facturación.



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

6. Vista de despliegue

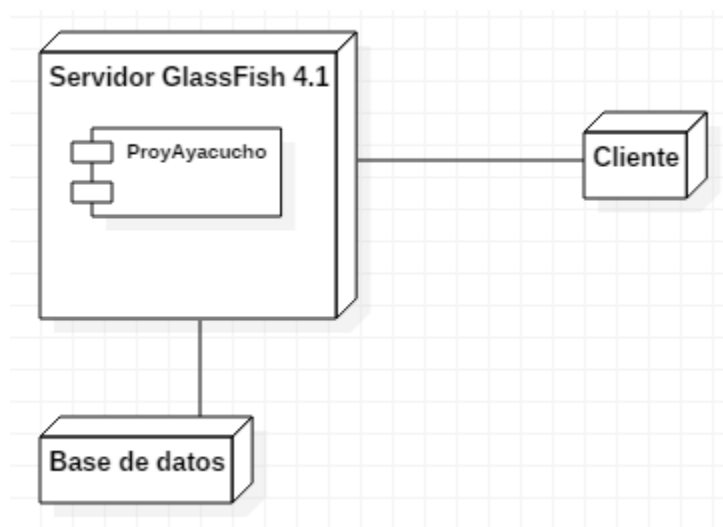
6.1. Introducción

Esta sección describe una o más configuraciones físicas sobre las cuales se realiza el despliegue del software y como es ejecutado, así como la infraestructura necesaria para su instalación.

Para el caso del sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera se describe el escenario general de distribución esperado para los componentes de software antes descritos, las características de los nodos presentados y la comunicación entre los mismos.

6.2. Distribución y despliegue

La figura siguiente presenta el escenario de distribución esperado para la instalación del sistema. El mismo se ubica en el contexto de una organización sobre una LAN privada.



A continuación se describen los nodos presentes en la figura:

- **Servidor GlassFish 4.1:** Es un servidor de aplicaciones desarrollado por Sun Microsystems, que implementa las tecnologías definidas en la plataforma Java Enterprise Edition y permite ejecutar aplicaciones que siguen esta especificación. En este servidor estará alojada la aplicación ProyAyacucho.
- **Base de datos:** Es un servidor MySQL 10.1.10 para gestionar las bases de datos relacionales.
- **Cliente:** Representa el medio por el cual el operador accederá a la aplicación. Puede realizarlo por los browser Chrome 53.0, Mozilla Firefox 49.0.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Documento arquitectura de software	Fecha: 30/Abril/2016

7. Referencias

[1] H. González, D. Lourido. Sistema para la gestión administrativa del acueducto rural Ayacucho, La Buitrera Palmira – Glosario. 2016.

[2] Rational Software Corporation. Rational Unified Process: Overview. [En Línea]. Disponible en: <http://sce.uhcl.edu/helm/rationalunifiedprocess/> Accedido: Marzo de 2016.

[3] H. González, D. Lourido. Sistema para la gestión administrativa del acueducto rural Ayacucho, La Buitrera Palmira – Requerimientos. 2016.

**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

Visión

**Versión 1.0
10/04/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Visión	Fecha: 10/Abril/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
10/Abril/2016	1.0	Propuesta inicial del documento Visión con las primeras capturas de requisitos funcionales del sistema.	Derlyn Johanna Lourido Quintero

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Visión	Fecha: 10/Abril/2016

Tabla de contenido

1. Introducción	3
1.1. Propósito	3
1.2. Referencias	3
2. Posicionamiento	3
2.1. Definición del problema	3
2.2. Definición del posicionamiento del producto	3
3. Descripción de los usuarios y partes interesadas	4
3.1. Partes interesadas (Stakeholders)	4
3.2. Usuarios	5
3.3. Necesidades de la parte interesada y los usuarios	5
3.4. Alternativas y competencias	6
4. Descripción del producto	6
4.1. Perspectiva del producto	6
4.2. Asunciones y dependencias	7
5. Características del producto	7
6. Otros requerimientos del producto	7

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Visión	Fecha: 10/Abril/2016

1. Introducción

1.1. Propósito

El propósito de este documento es recoger, analizar y definir las necesidades y características de alto nivel del Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira. Se centra en las capacidades necesarias de las partes interesadas y los usuarios objetivo, y por qué existen estas necesidades. Los detalles de cómo el sistema satisface estas necesidades se detallan en el caso de uso y las especificaciones complementarias.

1.2. Referencias

Glosario

RUP

Diagrama casos de uso del negocio

2. Posicionamiento

2.1. Definición del problema

El problema de	No contar con la gestión de la información administrativa y contable que maneja el acueducto de forma automatizada, haciendo que los procesos que se llevan a cabo tomen mucho tiempo en realizarse y la información se pierda o este duplicada.
Que afecta a	Los usuarios encargados de realizar los procesos de facturación y recaudo Al administrador para realizar los procesos de gestión del acueducto Al contador para realizar los procesos de rendición de cuentas
El impacto de ellos	No existe el control de los procesos realizados por los usuarios. Realizar los informes que solicitan los entes de vigilancia del país toma mucho tiempo. La información está duplicada y puede perderse porque no se cuenta con un respaldo. No contar con la información del acueducto oportunamente.
Una solución exitosa debería	Garantizar la idoneidad de los datos del acueducto. Facilitar y optimizar los procesos de facturación y recaudo. Agilizar los procesos de rendimiento de cuentas.

2.2. Definición del posicionamiento del producto

Para	Secretaria de acueductos rurales Área administrativa de acueductos rurales Contador de acueductos rurales
Quien	Gestionan la información de los usuarios del acueducto, información administrativa del acueducto, generan el proceso de facturación y recaudo, generan el control

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Visión	Fecha: 10/Abril/2016

	de la información contable
	Es un software para la gestión administrativa de acueductos rurales
Que	Almacena y procesa la información necesaria para la gestión administrativa y control contable de una empresa presta el servicio de acueducto.
A diferencia de	El sistema actual
Nuestro producto	Permite gestionar las distintas actividades relacionados con la administración de la empresa mediante una interfaz gráfica sencilla y amigable. Además proporciona un acceso rápido y actualizado a la información desde cualquier punto que tenga acceso a la web.

3. Descripción de los usuarios y partes interesadas

Para proveer de una forma efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que éstos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que éstos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por qué estos requisitos son necesarios.

3.1. Partes interesadas (Stakeholders)

Nombre	Descripción	Responsabilidades
Administrador	Es el encargado de administrar, organizar, dirigir y controlar los procesos y recursos del acueducto	• Responsable por el cumplimiento de los lineamientos que plantea la junta directiva. • Representante legal del acueducto • Responsable de supervisar los procesos que se llevan a cabo en el acueducto
Contador	Es el encargado de controlar y supervisar los procesos contables que se realizan en el acueducto	• Responsable de supervisar los movimientos contables que se realizan en el acueducto • Responsable de rendir cuentas a los entes

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Visión	Fecha: 10/Abril/2016

		gubernamentales
Junta Directiva	Es la encargada de tomar las decisiones administrativas del acueducto	• Responsable de crear y controlar los lineamientos del acueducto

3.2. Usuarios

Nombre	Descripción	Responsabilidades	Stakeholder
Operador	Es quien realiza las actividades de las procesos que se llevan a cabo en el acueducto	• Responsable de realizar la facturación y recaudo • Responsable de llevar el libro diario contable • Responsable de administrar la información de las acometidas y los suscriptores • Responsable de asignar las tarifas • Responsable de ordenar la entrega de las facturas	

3.3. Necesidades de la parte interesada y los usuarios

Necesidad	Prioridad	Preocupaciones	Solución actual	Solución propuesta
Gestión de las zonas del municipio y la distribución de las facturas	Baja		Se realiza el orden de la distribución manualmente	Modulo para la gestión de las zonas y distribución de las facturas por cada una de estas
Gestión de las tarifas	Media	Demora en obtener información de las tarifas asignadas a los suscriptores del acueducto	Se realiza en una hoja de Excel	Modulo para la gestión de las tarifas
Gestión de los servicios prestados	Media		Se llevan los controles en diferentes hojas de Excel	Modulo para la gestión de los servicios prestados
Gestión de los	Alta	Información	Se lleva en una	Modulo para la

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Visión	Fecha: 10/Abril/2016

Necesidad	Prioridad	Preocupaciones	Solución actual	Solución propuesta
suscriptores		duplicada o inexistente	hoja de Excel	gestión de los suscriptores
Gestión de las acometidas	Alta	Información duplicada o inexistente	Se lleva en una hoja de Excel	Modulo para la gestión de las acometidas
Facturación	Alta	Asignación de valores incorrectos en la factura	Se lleva a cabo en una hoja de Excel	Modulo que realice la facturación automáticamente con la información relacionada de la acometida y los servicios asignados a esta
Recaudo	Alta	Asignación de valores incorrectos al generar el pago	Se realiza manualmente	Modulo que gestione la información del recaudo y realice los movimientos contables automáticamente
Control contable	Alta	Perdida en la veracidad de los datos registrados	Se realiza en una hoja Excel	Modulo que permita supervisar los movimientos realizados por el operador

3.4. Alternativas y competencias

En el mercado existen numerosas herramientas para la gestión de acueductos. Los inconvenientes encontrados con estas herramientas es que no fueron pensadas para las necesidades que tienen los acueductos rurales. Por lo general, en el área rural no se cuenta con micro-medidores, las direcciones son referencias a puntos conocidos por la población, las opciones de pago son variadas según el caso, entre otros.

4. Descripción del producto

4.1. Perspectiva del producto

El producto a desarrollar es un sistema que permitirá la gestión de los procesos administrativos y control contable que se realiza en el acueducto rural Ayacucho La Buitrera Palmira. Las áreas a tratar por el sistema son: administración, facturación y recaudo, y control contable.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Visión	Fecha: 10/Abril/2016

4.2. Asunciones y dependencias

- Se contará con acceso a toda la información solicitada por el equipo del proyecto para la elaboración del sistema
- Los documentos o información que necesite el equipo de proyecto deberán ser solicitados de manera formal a los encargados de proporcionarla.
- El equipo de proyecto será independiente de las reglas organizacionales de la empresa y rendirá informes directamente al administrador del acueducto.
- Existe dependencia de que la computadora donde será instalada y ejecutada la aplicación deberá contar con acceso a internet.
- Los riesgos del sistema están relacionados con la consistencia de las comunicaciones de los diferentes usuarios a través de la Web, pudiendo establecerse fallas de conexión que den como resultado que las operaciones no se realicen de manera adecuada

5. Características del producto

- **Alta disponibilidad:** el sistema estará disponible 24/7 durante los 365 días del año en los cuales los usuarios podrán acceder desde cualquier lugar con conexión a internet.
- **Información actualizada y oportuna:** Las cargas se las realiza diariamente, por lo que la información se mantiene en actualización constante.
- **Detección temprana de errores:** En base a la verificación de información se puede detectar oportunamente los errores encontrados y reportarlos al usuario para que realice los cambios.

6. Otros requerimientos del producto

Se requiere de un servidor que despliegue los siguientes servicios:

- Servidor web: Glassfish 4.1
- Servidor de base de datos: MySQL 10.1.10

**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

Glosario

**Versión 1.0
05/04/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Glosario	Fecha: 05/Abril/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
10/Abril/2016	1.0	Propuesta inicial del documento Glosario	Hernán D. González

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Glosario	Fecha: 05/Abril/2016

Tabla de contenido

1. Introducción.....	1
1.1. Propósito.....	1
1.2. Alcance.....	1
1.3. Referencias.....	1
1.4. Organización del Glosario.....	1
2. Definiciones.....	1
2.1. Acreedor.....	1
2.2. Acueducto.....	1
2.3. Administradora.....	1
2.4. Asignar Movimiento.....	2
2.5. Auxiliar administrativo.....	2
2.6. Barrio.....	2
2.7. Concepto.....	2
2.8. Contador.....	2
2.9. Cuenta Contable.....	2
2.10. Cuenta Crédito.....	2
2.11. Cuenta Debito.....	2
2.12. Descargar Archivo.....	2
2.13. Facturación.....	2
2.14. Fontanero.....	2
2.15. Generar Archivo.....	2
2.16. Junta Directiva.....	3
2.17. Micro-medidor.....	3
2.18. Movimiento.....	3
2.19. Operador.....	3
2.20. Orden de distribución.....	3
2.21. Otros Pagos.....	3
2.22. Predio.....	3
2.23. Proveedor.....	3

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Glosario	Fecha: 05/Abril/2016

2.24.	Recaudo.....	3
2.25.	Saldo Acometida.....	3
2.26.	Servicios.....	3
2.27.	Suscriptor.	3
2.28.	Tarifa.	4
2.29.	Tipo Concepto.	4
2.30.	Tipo de Pago.	4
2.31.	Tipo de Predio.	4
2.32.	Ultimo registro de factura.	4
2.33.	Uso.	4
2.34.	Usuario.	4
2.35.	Vereda.	4
2.36.	Zona.....	4

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Glosario	Fecha: 05/Abril/2016

GLOSARIO

1. Introducción.

Este documento abarca todos los términos utilizados a lo largo del proyecto AYACUCHO. Se busca tener un diccionario de datos y definiciones que manejan tratando de tener un lenguaje unificado a lo largo del proyecto.

1.1. Propósito

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto de desarrollo de un sistema para la gestión del acueducto AYACUCHO. También sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecedores del proyecto.

1.2. Alcance

El alcance del presente documento se extiende a todos los subsistemas definidos para el acueducto AYACUCHO. De tal modo que la terminología empleada se refleja con claridad en este documento.

1.3. Referencias.

El presente glosario hace referencia a los siguientes documentos:

1.4. Organización del Glosario.

El presente documento está organizado por definiciones de términos ordenados de forma ascendente según la ordenación alfabética tradicional del español.

2. Definiciones.

2.1. Acreedor.

Es aquella persona física o jurídica que legítimamente está autorizada para exigir el pago o cumplimiento de una obligación contraída con anterioridad. Como ejemplo está el fontanero, la contadora, la auxiliar administrativa.

2.2. Acueducto.

Es un sistema o conjunto de sistemas de irrigación que permite transportar agua en forma de flujo continuo desde un lugar en el que está accesible en la naturaleza hasta un punto de consumo distante, generalmente una ciudad o poblado.

2.3. Administradora.

Es la persona encargada de administrar, organizar, dirigir y controlar los procesos y recursos del acueducto encargada de dirigir el acueducto.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Glosario	Fecha: 05/Abril/2016

2.4. Asignar Movimiento.

Es el evento que se genera mientras se está creando otro pago, este consta de un número de cuenta contable, una descripción, un valor y un número de identificación de un usuario.

2.5. Auxiliar administrativo.

Es la persona encargada en ayudar a la administradora, lleva el control de los pagos que se hacen a las acometidas, redacta los diferentes documentos manejados en el acueducto.

2.6. Barrio.

Es una subdivisión territorial, también es llamada vereda ya que hace referencia a un área rural.

2.7. Concepto.

Es el nombre que recibe la descripción del servicio que se está prestando.

2.8. Contador.

Es el encargado de controlar y supervisar los procesos contables que se realizan en el acueducto.

2.9. Cuenta Contable.

Hace parte el PUC (Plan único de cuentas) la cual consta de un código y una descripción.

2.10. Cuenta Crédito.

Hace referencia al movimiento contable que va a ser acreditado.

2.11. Cuenta Debito.

Hace referencia al movimiento contable que va a ser debitado.

2.12. Descargar Archivo.

Es el evento que se encarga de descargar el documento del bloque de factura o copia de una factura específica para después ser impreso por la auxiliar contable.

2.13. Facturación.

Es el documento donde se registran todos los servicios que se han prestado durante el mes a una acometida.

2.14. Fontanero.

Persona que tiene por oficio colocar, conservar y reparar las conducciones de agua e instalaciones sanitarias que regulan, canalizan y distribuyen el agua en un edificio.

2.15. Generar Archivo.

Es la consolidación de los registros de una factura, se puede generar ya sea por un bloque de facturas de un mes o por copia de la factura de una acometida.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Glosario	Fecha: 05/Abril/2016

2.16. Junta Directiva.

Es la encargada de tomar las decisiones administrativas del acueducto.

2.17. Micro-medidor.

Es el dispositivo a través del cual se mide el consumo de agua que genera un predio conocido popularmente como contador.

2.18. Movimiento.

Hacemos referencia al movimiento contable en el cual se registran cada vez se genera un ingreso o un egreso de dinero al acueducto.

2.19. Operador.

Persona encargada de manipular el sistema de gestión del acueducto APZU.

2.20. Orden de distribución.

Es el orden según la ruta de distribución organizada por la persona encargada de distribuir las facturas de las acometidas en los predios, estas tienen un orden según la zona donde se encuentra el predio.

2.21. Otros Pagos.

Son los pagos diferentes a acometida o a servicios, estos siempre se deben configurar los movimientos manualmente indicando la cuenta que se va a debitar y la cuenta que se va acreditar.

2.22. Predio.

Es un terreno delimitado que pertenece a una persona, en este es donde se hará uso del servicio del agua. Conocido también como lote, casa, hacienda... etc.

2.23. Proveedor.

Son personas o empresas que prestan bienes o servicios a quienes lo necesitan.

2.24. Recaudo.

Hace referencia a recibir el dinero en forma de pago por la prestación de un servicio.

2.25. Saldo Acometida.

Cantidad a favor o en contra de dinero que tiene una acometida.

2.26. Servicios.

Conjunto de actividades que busca resolver las necesidades de otra persona.

2.27. Suscriptor.

Es la persona responsable de pagar los servicios prestados a un predio por el acueducto.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Glosario	Fecha: 05/Abril/2016

2.28. Tarifa.

Es el monto establecido por la junta directiva para la prestación del servicio del agua, el acueducto maneja diferentes tarifas según las características del predio.

2.29. Tipo Concepto.

Es la catalogación de los conceptos establecidos por la junta directiva.

2.30. Tipo de Pago.

Es la catalogación de los pagos establecidos por la junta directiva.

2.31. Tipo de Predio.

Es la catalogación de los predios según sus características.

2.32. Ultimo registro de factura.

Es la fecha que se lleva como control para saber la última vez que se generó el cobro de facturas a las acometidas.

2.33. Uso.

Hace referencia a la función que se le da al predio que puede ser comercial, empresarial o residencial.

2.34. Usuario.

Es la persona que va a tener relación con el acueducto según los roles establecido que son Suscriptor, Proveedor o Acreedor. Un usuario debe tener mínimo un rol en el sistema.

2.35. Vereda.

Es un tipo de subdivisión territorial que consta de mínimo 50 y máximo 1200 personas.

2.36. Zona.

Es la división establecida por un grupo de personas, en este caso las diferentes zonas son establecidas por la junta directiva.

**Sistema para la gestión
administrativa del acueducto rural
Ayacucho La Buitrera, Palmira**
Atributos de los requerimientos

Versión 1.0
15/Abril/2016

Confidencial

@Ayacucho, La Buitrera

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
10/Abril/2016	1.0	Propuesta inicial del documento de Requerimientos.	Hernan Darío Gonzalez

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

Tabla de contenido

Registrar usuarios	1
Modificar usuarios	2
Eliminar usuarios	3
Registrar acometidas	4
Modificar acometidas	5
Eliminar acometidas	6
Registrar zonas.....	7
Modificar zonas	8
Eliminar zonas.....	9
Ordenar distribución de las facturas.....	10
Registrar tipos de concepto	11
Modificar tipos de concepto	12
Eliminar tipos de concepto	13
Registrar conceptos	14
Modificar conceptos	15
Eliminar conceptos	16
Registrar tarifas.....	17
Modificar tarifas	18
Eliminar tarifas.....	19
Generar bloque de facturas	21
Generar copia de la factura del último mes	22
Registrar pagos a conceptos por acometida.....	23
Registrar pago por conceptos de servicios.....	24
Registrar otros pagos	25

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Registrar usuarios	
IDENTIFICADOR	RF-01	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Documento de identificación • Tipo de identificación • Tipo de usuario • Nombres • Apellidos • Correo electrónico • Teléfono fijo • Teléfono móvil	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la creación del suscriptor	
DESCRIPCIÓN	El sistema debe permitir registrar y almacenar la información de un nuevo usuario.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Modificar usuarios	
IDENTIFICADOR	RF-02	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Documento de identificación • Tipo de identificación • Tipo de usuario • Nombres • Apellidos • Correo electrónico • Teléfono fijo • Teléfono móvil	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la modificación del usuario	
DESCRIPCIÓN	El sistema debe permitir modificar la información de un usuario seleccionado. Se puede buscar al usuario por los campos visibles en la lista de los usuarios: • Documento de identificación • Nombres y Apellidos • Teléfono fijo • Teléfono celular • Correo Una vez seleccionado el usuario requerido, se pueden modificar todos los datos relacionados a este.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Eliminar usuarios	
IDENTIFICADOR	RF-03	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Documento de identificación • Nombres y Apellidos • Teléfono fijo • Teléfono celular • Correo	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la eliminación del usuario	
DESCRIPCIÓN	El sistema debe permitir eliminar la información de un usuario seleccionado. Se debe permitir buscarlo por los campos visibles en la lista de los usuarios. Una vez seleccionado el usuario requerido, se debe confirmar para proceder con la eliminación.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Registrar acometidas	
IDENTIFICADOR	RF-04	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	- Número de documento del suscriptor - Dirección del predio - Número catastral del predio - Barrio o Vereda - Zona de instalación - Tipo de predio - Uso del predio - Dirección de envío de la factura - Barrio o vereda de envío de la factura - Zona de envío de la factura - Servicios asignados	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la creación de la acometida.	
DESCRIPCIÓN	El sistema debe permitir registrar y almacenar la información de una nueva acometida.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Modificar acometidas	
IDENTIFICADOR	RF-05	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	- Número de documento del suscriptor - Dirección del predio - Número catastral del predio - Barrio o Vereda - Zona de instalación - Tipo de predio - Uso del predio - Dirección de envío de la factura - Barrio o vereda de envío de la factura - Zona de envío de la factura - Servicios asignados	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la modificación de la acometida	
DESCRIPCIÓN	El sistema debe permitir modificar la información de una acometida seleccionada. Se puede buscar la acometida por los campos visibles en la lista de las acometidas: - Nombres y/o Apellidos del suscriptor - Dirección del predio - Barrio o Vereda - Zona de instalación - Tipo de predio - Uso del predio Una vez seleccionada la acometida requerida, se pueden modificar todos los datos relacionados a esta.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Eliminar acometidas	
IDENTIFICADOR	RF-06	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Nombres y/o Apellidos del suscriptor • Dirección del predio • Barrio o Vereda • Zona de instalación • Tipo de predio • Uso del predio	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la eliminación de la acometida.	
DESCRIPCIÓN	El sistema debe permitir eliminar la información de una acometida seleccionada. Se debe permitir buscarla por los campos visibles en la lista de las acometidas. Una vez seleccionada la acometida requerida, se debe confirmar para proceder con la eliminación.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Registrar zonas	
IDENTIFICADOR	RF-07	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	Descripción de la zona.	
SALIDA	El sistema mostrara un mensaje de informado el éxito de la creación de la zona.	
DESCRIPCIÓN	El sistema debe permitir registrar y almacenar la información de una nueva zona.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Modificar zonas	
IDENTIFICADOR	RF-08	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	Descripción de la zona	
SALIDA	El sistema debe mostrar un mensaje de información indicando el éxito de la modificación de la zona	
DESCRIPCIÓN	El sistema debe permitir modificar la información de una zona seleccionada. Se puede buscar la zona por los campos visibles en la lista de las zonas: Descripción Una vez seleccionada la zona requerida, se pueden modificar todos los datos relacionados a esta.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Eliminar zonas	
IDENTIFICADOR	RF-09	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	Descripción de la zona	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la eliminación de la zona	
DESCRIPCIÓN	El sistema debe permitir eliminar la información de una zona seleccionada. Se debe permitir buscarla por los campos visibles en la lista de las zonas. Una vez seleccionado la zona requerida, se debe confirmar para proceder con la eliminación.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Ordenar distribución de las facturas	
IDENTIFICADOR	RF-10	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	Descripción de la zona.	
SALIDA	El sistema debe mostrar un mensaje indicando el éxito de la asignación del nuevo orden de distribución de las facturas de la zona.	
DESCRIPCIÓN	El sistema debe permitir ordenar la distribución de las facturas de una zona seleccionada. El operador desplaza (Arriba/Abajo) la posición de las acometidas asociadas a la zona según las pautas del repartidor.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Registrar tipos de concepto	
IDENTIFICADOR	RF-11	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	Descripción del tipo de concepto	
SALIDA	El sistema debe mostrar un mensaje indicando el éxito de la creación del tipo de concepto.	
DESCRIPCIÓN	El sistema debe permitir registrar y almacenar información de un nuevo tipo de concepto.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Modificar tipos de concepto	
IDENTIFICADOR	RF-12	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	Descripción del tipo de concepto	
SALIDA	El sistema debe mostrar un mensaje indicando el éxito de la modificación del tipo de concepto.	
DESCRIPCIÓN	El sistema debe permitir modificar la información de un tipo de concepto seleccionado. Se puede buscar el tipo de concepto por los campos visibles en la lista de los tipos de concepto: Descripción del tipo de concepto Una vez seleccionado el tipo de concepto requerido, se pueden modificar todos los datos relacionados a este.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Eliminar tipos de concepto	
IDENTIFICADOR	RF-13	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	Descripción tipo de concepto	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la eliminación del tipo de concepto.	
DESCRIPCIÓN	El sistema debe permitir la eliminación de un tipo de concepto seleccionado. Se debe permitir buscarlo por los campos visibles en la lista de los tipos de concepto. Una vez seleccionado el tipo de concepto requerido, se debe confirmar para proceder con la eliminación.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Registrar conceptos	
IDENTIFICADOR	RF-14	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Descripción del concepto • Tipo de concepto	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la creación del concepto	
DESCRIPCIÓN	El sistema debe permitir registrar y almacenar la información de un nuevo concepto.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Modificar conceptos	
IDENTIFICADOR	RF15	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	- Descripción del concepto - Tipo de concepto	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la modificación del concepto	
DESCRIPCIÓN	El sistema debe permitir modificar la información de un concepto seleccionado. Se puede buscar el concepto por los campos visibles en la lista de los conceptos: - Descripción del concepto - Tipo de concepto Una vez seleccionado el concepto requerido, se pueden modificar todos los datos relacionados a este.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Eliminar conceptos	
IDENTIFICADOR	RF-16	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Descripción del concepto • Tipo de concepto	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la eliminación del concepto	
DESCRIPCIÓN	El sistema debe permitir eliminar la información de un concepto seleccionado. Se debe permitir buscarlo por los campos visibles en la lista de los conceptos. Una vez seleccionado el concepto requerido, se debe confirmar para proceder con la eliminación.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Registrar tarifas	
IDENTIFICADOR	RF-17	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Descripción de la tarifa • Valor • Concepto	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la creación de la tarifa	
DESCRIPCIÓN	El sistema permitirá ingresar toda la información requerida para registrar la tarifa	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Modificar tarifas	
IDENTIFICADOR	RF18	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Descripción de la tarifa • Valor • Concepto	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la modificación de la tarifa	
DESCRIPCIÓN	El sistema debe permitir modificar la información de una tarifa seleccionada. Se puede buscar la tarifa por los campos visibles en la lista de las tarifas: • Descripción de la tarifa • Concepto Una vez seleccionada la tarifa requerida, se pueden modificar todos los datos relacionados a esta.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Eliminar tarifas	
IDENTIFICADOR	RF-19	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Descripción de la tarifa • Concepto	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la eliminación de la tarifa	
DESCRIPCIÓN	El sistema debe permitir eliminar la información de una tarifa seleccionada. Se debe permitir buscarla por los campos visibles en la lista de las tarifas. Una vez seleccionado la tarifa requerida, se debe confirmar para proceder con la eliminación.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Realizar facturación	
IDENTIFICADOR	RF-20	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	Acometidas	
SALIDA	El sistema debe mostrar un mensaje informando el éxito de la facturación.	
DESCRIPCIÓN	El sistema debe permitir generar un periodo de facturación con la información almacenada de la lista total de acometidas. Los movimientos contables se deben generar según los servicios asociados a la factura.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Generar bloque de facturas	
IDENTIFICADOR	RF-21	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA		
SALIDA	- El sistema debe mostrar un mensaje informando el éxito de la generación del bloque de facturas. - El sistema debe generar un archivo .pdf con todas las facturas de esa fecha.	
DESCRIPCIÓN	El sistema debe permitir generar copia del bloque total de las facturas generadas en el periodo de facturación.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Generar copia de la factura del periodo actual	
IDENTIFICADOR	RF-22	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	Acometida	
SALIDA	- El sistema debe generar un documento en formato .pdf con la información de la factura asociada a la acometida seleccionada - El sistema debe mostrar un mensaje informando el éxito de la generación de la factura	
DESCRIPCIÓN	El sistema debe permitir generar una copia de la última factura generada a una acometida seleccionada. Se debe permitir buscar la acometida por los campos visibles en la lista de acometidas: - Nombres y/o Apellidos del suscriptor - Dirección del predio - Barrio o Vereda - Zona de instalación - Tipo de predio - Uso del predio Una vez seleccionada la acometida requerida se procede a descargar el documento.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Registrar pagos a conceptos por acometida	
IDENTIFICADOR	RF23	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Acometida • Valor a pagar	
SALIDA	El sistema debe mostrar un mensaje informando que el pago se generó con éxito.	
DESCRIPCIÓN	El sistema debe permitir realizar pagos por conceptos de derechos de acometida. Se debe permitir buscar la acometida requerida con los campos visibles en la lista de acometidas: • Nombres y/o Apellidos del suscriptor • Dirección del predio • Barrio o Vereda • Zona de instalación • Tipo de predio • Uso del predio Una vez seleccionada la acometida requerida, se debe permitir registrar el valor a pagar y generarse el pago. El sistema debe generar los movimientos a las cuentas contables asociadas a los conceptos por acometida.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Registrar pago por conceptos de servicios	
IDENTIFICADOR	RF-24	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Acometida • Valor a pagar	
SALIDA	El sistema debe mostrar un mensaje informando que el pago se generó con éxito.	
DESCRIPCIÓN	El sistema debe permitir realizar pagos por conceptos de servicios. Se debe permitir buscar la acometida requerida con los campos visibles en la lista de acometidas: • Nombres y/o Apellidos del suscriptor • Dirección del predio • Barrio o Vereda • Zona de instalación • Tipo de predio • Uso del predio Una vez seleccionada la acometida requerida, se debe permitir registrar el valor a pagar y generarse el pago. El sistema debe generar los movimientos a las cuentas contables asociadas a los conceptos por servicios.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Requerimientos	Fecha: 10/Abril/2016

REQUERIMIENTO	Registrar otros pagos	
IDENTIFICADOR	RF-25	
TIPO DE REQUERIMIENTO	Funcional	
ENTRADA	• Número de documento de identificación del usuario • Tipo de usuario • Cuenta débito • Cuenta crédito • Valor a pagar • Descripción	
SALIDA	El sistema mostrara un mensaje informando que el pago se generó exitosamente	
DESCRIPCIÓN	El sistema debe permitir generar pagos por otros conceptos. Al ser pagos a conceptos no establecidos, el sistema debe permitir asignar las respectivas cuentas débito y crédito que estarán asociadas al pago. Estas cuentas se pueden buscar por el código de cuenta o su descripción. Una vez asociados todos los movimientos requeridos se genera el pago.	
Realizado por	Firma	Fecha
Hernán Darío González Patiño		14/04/2016
Aprobado por	Firma	Fecha
Derlyn Johanna Lourido Quintero		20/04/2016



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira

Atributos de los requerimientos

**Versión 1.0
11/04/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Atributos de los requerimientos	Fecha: 11/Abril/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
11/Abril/2016	1.0	Propuesta inicial del documento atributos de los requerimientos	Derlyn Johanna Lourido Quintero

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Atributos de los requerimientos	Fecha: 11/Abril/2016

Tabla de contenido

Gestionar usuarios	1
Gestionar acometidas	2
Gestionar zonas.....	3
Gestionar tipos de conceptos.....	3
Gestionar conceptos	4
Gestionar tarifas.....	4
Gestionar facturas.....	5
Gestionar recaudo.....	6

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Atributos de los requerimientos	Fecha: 11/Abril/2016

AR-01	Gestionar usuarios
Casos de uso asociados	CU-01 Gestionar Usuarios
Requerimientos asociados	RF-01 Registrar usuarios RF-02 Modificar usuarios RF-03 Eliminar usuarios
Descripción	La información almacenada por el sistema deberá satisfacer las siguientes restricciones: • El número de documento de identificación del usuario debe ser único, no puede existir dos usuarios con la misma identificación • El usuario debe tener asignado como mínimo un tipo de usuario • Para registrar un usuario previamente debe existir en la base de datos: ○ Tipos de usuario ○ Tipos documento de identificación • Para eliminar un usuario se debe validar que este no tenga acometidas asociadas
Datos obligatorios	• Documento de identificación • Tipo de identificación • Tipo de usuario • Nombres • Apellidos
Comentarios	En caso de que no se cumpla alguna obligación o se viole alguna restricción, el sistema debe mostrar un mensaje informando el error cometido.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Atributos de los requerimientos	Fecha: 11/Abril/2016

AR-02	Gestionar acometidas
Casos de uso asociados	CU-02 Gestionar acometidas
Requerimientos asociados	RF-04 Registrar acometidas RF-05 Modificar acometidas RF-06 Eliminar acometidas RF-07 Registrar zonas RF-11 Tipos de conceptos RF-14 Registrar conceptos RF-17 Registrar tarifas
Descripción	La información almacenada por el sistema deberá satisfacer las siguientes restricciones: • Para registrar una acometida previamente debe existir en la base de datos: ○ Usuarios ○ Zonas ○ Tipos de concepto ○ Conceptos ○ Tarifas • La acometida debe tener asignado como mínimo un servicio
Datos obligatorios	• Documento de identificación del suscriptor • Dirección del predio • Número catastral del predio • Barrio o Vereda • Zona de instalación • Tipo de predio • Uso del predio • Dirección de envío de la factura • Barrio o vereda de envío de la factura • Zona de envío de la factura • Servicios asignados
Comentarios	En caso de que no se cumpla alguna obligación o se viole alguna restricción, el sistema debe mostrar un mensaje informando el error cometido.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Atributos de los requerimientos	Fecha: 11/Abril/2016

AR-03	Gestionar zonas
Casos de uso asociados	
Requerimientos asociados	RF-07 Registrar zonas RF-08 Modificar zonas RF-09 Eliminar zonas RF-10 Ordenar distribución de las facturas
Descripción	La información almacenada por el sistema deberá satisfacer las siguientes restricciones: • Para eliminar una zona, esta no debe estar asociada a ninguna acometida • Para ordenar la distribución de las facturas debe existir previamente acometidas en la base de datos
Datos obligatorios	• Descripción de la zona
Comentarios	En caso de que no se cumpla alguna obligación o se viole alguna restricción, el sistema debe mostrar un mensaje informando el error cometido.

AR-04	Gestionar tipos de conceptos
Casos de uso asociados	
Requerimientos asociados	RF-11 Registrar tipos de concepto RF-12 Modificar tipos de concepto RF-13 Eliminar tipos de concepto
Descripción	La información almacenada por el sistema deberá satisfacer las siguientes restricciones: • Para eliminar un tipos de concepto, este no debe estar asociado a ningún concepto
Datos obligatorios	• Descripción del tipo de concepto
Comentarios	En caso de que no se cumpla alguna obligación o se viole alguna restricción, el sistema debe mostrar un mensaje informando el error cometido.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Atributos de los requerimientos	Fecha: 11/Abril/2016

AR-05	Gestionar conceptos
Casos de uso asociados	
Requerimientos asociados	RF-14 Registrar conceptos RF-15 Modificar conceptos RF-16 Eliminar conceptos RF-11 Registrar tipos de concepto
Descripción	La información almacenada por el sistema deberá satisfacer las siguientes restricciones: • Para registrar un concepto deben existir previamente tipos de concepto almacenados en la base de datos • Para eliminar un concepto, este no debe estar asociado a ninguna tarifa
Datos obligatorios	• Descripción del concepto • Tipo de concepto
Comentarios	En caso de que no se cumpla alguna obligación o se viole alguna restricción, el sistema debe mostrar un mensaje informando el error cometido.

AR-06	Gestionar tarifas
Casos de uso asociados	
Requerimientos asociados	RF-17 Registrar tarifas RF-18 Modificar tarifas RF-19 Eliminar tarifas RF-11 Registrar tipos de concepto RF-14 Registrar conceptos
Descripción	La información almacenada por el sistema deberá satisfacer las siguientes restricciones: • Para registrar una tarifa deben existir previamente conceptos almacenados en la base de datos • Para eliminar una tarifa, esta no debe estar asociada a ninguna acometida
Datos obligatorios	• Descripción del concepto • Concepto
Comentarios	En caso de que no se cumpla alguna obligación o se viole alguna restricción, el sistema debe mostrar un mensaje informando el error cometido.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Atributos de los requerimientos	Fecha: 11/Abril/2016

AR-07	Facturación
Casos de uso asociados	
Requerimientos asociados	RF-04 Registrar acometidas RF-20 Facturación
Descripción	La información almacenada por el sistema deberá satisfacer las siguientes restricciones: • Sólo debe permitirse generar la facturación al total de las acometidas registradas una vez al mes. • Las acometidas que aún no se han instalado, no se les debe generar facturación de servicios • Para realizar la facturación debe existir previamente en la base de datos: ○ Acometidas ○ Tipos de documento contable ○ Conceptos contables ○ PUC
Datos obligatorios	• Acometidas
Comentarios	En caso de que no se cumpla alguna obligación o se viole alguna restricción, el sistema debe mostrar un mensaje informando el error cometido.

AR-08	Gestionar facturas
Casos de uso asociados	
Requerimientos asociados	RF-21 Generar bloque de facturas RF-22 Generar copia de la factura del periodo actual RF-20 Facturación
Descripción	La información almacenada por el sistema deberá satisfacer las siguientes restricciones: • Para generar un bloque de facturas, debe haberse realizado previamente la facturación del periodo actual. • Para generar copia de la factura del último mes debe haberse realizado la facturación del periodo actual.
Datos obligatorios	• Acometidas
Comentarios	En caso de que no se cumpla alguna obligación o se viole alguna restricción, el sistema debe mostrar un mensaje informando el error cometido.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Atributos de los requerimientos	Fecha: 11/Abril/2016

AR-08	Gestionar recaudo
Casos de uso asociados	
Requerimientos asociados	RF-21 Generar bloque de facturas RF-22 Generar copia de la factura del periodo actual RF-20 Facturación
Descripción	La información almacenada por el sistema deberá satisfacer las siguientes restricciones: • Para generar un bloque de facturas, debe haberse realizado previamente la facturación del periodo actual. • Para generar copia de la factura del último mes debe haberse realizado la facturación del periodo actual.
Datos obligatorios	• Acometidas
Comentarios	En caso de que no se cumpla alguna obligación o se viole alguna restricción, el sistema debe mostrar un mensaje informando el error cometido.

**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

Especificaciones Suplementarias

**Versión 1.0
10/04/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Especificaciones suplementarias	Fecha: 10/Abril/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
10/Abril/2016	1.0	Propuesta inicial del documento Especificaciones suplementarias.	Derlyn Johanna Lourido Quintero

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Especificaciones suplementarias	Fecha: 10/Abril/2016

Contenido

1.	Introducción	3
2.	Funcionalidad	3
2.1.	Prestación del servicio	3
2.1.1.	Gestionar zonas	3
2.1.2.	Gestionar Tipo Conceptos	3
2.1.3.	Gestionar Conceptos	3
2.1.4.	Gestionar Tarifas	3
2.1.5.	Gestionar Orden de distribución	3
3.	Usabilidad.....	3
4.	Confiabilidad	4
5.	Rendimiento.....	4
6.	Compatibilidad	4
7.	Restricciones de diseño	4
8.	Requerimientos de documentación en línea y ayudas del sistema	4
9.	Compra de componentes	4
10.	Interfaces	4
11.	Licenciamiento	4
12.	Avisos legales, Derechos de autor y Otros.....	5
13.	Normas aplicables	5

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Especificaciones suplementarias	Fecha: 10/Abril/2016

1. Introducción

Este documento de especificación suplementaria describe las características no funcionales de alto nivel a implementarse en el proyecto en términos de necesidades de usuarios finales, capturando todos los requisitos que no han sido incluidos como parte de los casos de uso.

2. Funcionalidad

2.1. Prestación del servicio

2.1.1. Gestionar zonas

El sistema debe permitir al operador registrar, modificar y eliminar las zonas (sectores geográficos) en los cuales el acueducto presta el servicio.

La zona está representada por una descripción.

2.1.2. Gestionar Tipo Conceptos

El sistema debe permitir al operador registrar, modificar y eliminar los diferentes tipos de conceptos que se usaran para la prestación del servicio.

El Tipo Concepto está representado por una descripción.

2.1.3. Gestionar Conceptos

El sistema debe permitir al operador registrar, modificar y eliminar los conceptos de servicios que el acueducto presta.

El concepto requiere ser asignado a un tipo de concepto.

El concepto está representado por una descripción.

2.1.4. Gestionar Tarifas

El sistema debe permitir al operador registrar, modificar y eliminar las tarifas que el acueducto cobra por la prestación del servicio.

La tarifa requiere ser asignada a un concepto.

La tarifa está representada por una descripción y tiene un valor a cobrar.

2.1.5. Gestionar Orden de distribución

El sistema debe permitir al operador gestionar el orden de la distribución de las facturas. Este orden se asigna según las recomendaciones del repartidor.

3. Usabilidad

Al ser un software desarrollado para personas que no tienen por qué mantener una relación directa con el mundo de la informática, el producto software debe ser construido de forma que sea fácil su aprendizaje y utilización, para mejorar la eficiencia en el trabajo diario del operador.

El idioma utilizado será el castellano. Se podría pensar en exhibir la información en otros idiomas aunque hoy por hoy sería una opción de poca utilidad dado el destino de su utilización.

El sistema debe contar con iconos estándar que ayuden a deducir la funcionalidad y no causen desorientación al usuario.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Especificaciones suplementarias	Fecha: 10/Abril/2016

El sistema debe contar con una interfaz con colores que eviten el cansancio visual durante el uso prolongado.

4. Confiabilidad

Propiedad que tiene el sistema de realizar las funciones para las que fue diseñado sin fallos.

5. Rendimiento

El sistema debe tener un buen rendimiento sirviendo a los usuarios toda la funcionalidad que el sistema les ofrece de una forma rápida y eficiente

6. Compatibilidad

El sistema debe ser compatible con los navegadores Mozilla Firefox 40.0 y Google Chrome 50.0.

7. Restricciones de diseño

El sistema deberá ser diseñado en Java para entornos Web usando la tecnología de Java Server Faces.

Se podrá hacer uso de herramientas, componentes, servidores y demás, que cuenten con licencias open source, free, entre otras. No hacer uso si cuentan con licencias privadas o comerciales.

8. Requerimientos de documentación en línea y ayudas del sistema

El operario dispondrá de una guía de ayuda que intentará solucionar cualquier problema que le pueda surgir. El operario, por tanto, no necesitará poseer grandes conocimientos de informática para la utilización del software.

9. Compra de componentes

El sistema se desarrollará con herramientas, componentes, servidores y demás que cuenten con licencias open source, free, entre otras.

10. Interfaces

El software no tiene interfaces con sistemas (hardware o software) externos.

11. Licenciamiento

El sistema contará con una licencia de software privado. El acueducto podrá hacer uso del sistema sin costo alguno. No se entregarán los artefactos finales del sistema. No se podrán hacer modificaciones, ni se podrá redistribuir sin autorización de la universidad.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Especificaciones suplementarias	Fecha: 10/Abril/2016

12. Avisos legales, Derechos de autor y Otros

Los derechos de autor del software y los artefactos relacionados a este pertenecen a la Universidad del Valle.

13. Normas aplicables

**Sistema para la gestión
administrativa del acueducto rural
Ayacucho La Buitrera, Palmira
CU-01 Gestionar Usuarios**

**Versión 1.0
10/04/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-01 Gestionar Usuarios	

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
10/Abril/2016	1.0	Elaboración del documento	Hernan Darío Gonzalez

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-01 Gestionar Usuarios	

Tabla de contenido

1. Descripción Breve.....	1
2. Flujo Básico de Eventos	1
2.1. Registrar Usuario	1
2.2. Modificar Usuario	1
2.3. Eliminar Usuario	1
2.4. Fin del Caso de Uso.	2
3. Flujos Alternos.....	2
3.1. Información requerida incompleta: Registrar Usuario	2
3.2. Información requerida incompleta: Modificar Usuario.	2
3.3. Usuario con acometidas: Eliminar Usuario.	2
4. Sub-flujos	2
4.1. Buscar usuario	2
5. Precondiciones	2
6. Poscondiciones.....	2

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-01 Gestionar Usuarios	

ESPECIFICACIÓN DEL CASO DE USO: CU-01 GESTIONAR USUARIOS

1. Descripción Breve

Este caso de uso es iniciado por el Operador y en él se permite:

- Registrar un usuario
- Modificar un usuario
- Eliminar un usuario

2. Flujo Básico de Eventos

2.1. Registrar Usuario

2.1.1 El operador ingresa la información del usuario, la cual consta de:

- Documento de identificación
- Tipo de identificación
- Nombres
- Apellidos
- Correo electrónico
- Teléfono fijo
- Teléfono móvil

2.1.2 El operador asigna como mínimo un tipo de usuario

2.1.3 Se registra el usuario.

2.2. Modificar Usuario

2.2.1. El operador selecciona un usuario de la lista total de usuarios.

2.2.2. El operador modifica la información que requiere de los siguientes datos:

- Documento de identificación
- Tipo de identificación
- Nombres
- Apellidos
- Correo electrónico
- Teléfono fijo
- Teléfono móvil
- Tipo de usuario

2.2.3. Se modifica el usuario

2.3. Eliminar Usuario

2.3.1. El operador selecciona un usuario de la lista total de usuarios.

2.3.2. El operador ejecuta el proceso de eliminar el usuario.

2.3.3. El operador debe confirmar la acción de eliminar el usuario.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-01 Gestionar Usuarios	

2.4. Fin del Caso de Uso.

2.4.1. El caso de uso termina.

3. Flujos Alternos

3.1. Información requerida incompleta: Registrar Usuario

En el paso 2.1.3 Registrar usuario, si la información requerida no se ha ingresado, el sistema muestra un mensaje detallando la información que falta.

El caso de uso continúa en el paso 2.1.1.

3.2. Información requerida incompleta: Modificar Usuario.

En el paso 2.2.2 Modificar usuario, si la información requerida no se ha ingresado, el sistema muestra un mensaje detallando la información que falta.

El caso de uso continúa en el paso 2.2.1.

3.3. Usuario con acometidas: Eliminar Usuario.

En el paso 2.3.3 Eliminar usuario, si el usuario tiene asignada una o más acometidas, el sistema muestra un mensaje informando que el usuario no puede ser eliminado por que tiene una o más acometidas asignadas.

4. Sub-flujos

4.1. Buscar usuario

En los flujos 2.2.1 Modificar usuario y 2.3.1 Eliminar usuario, el operador puede realizar un filtro a la lista de usuarios por los siguientes datos:

- Documento de identificación
- Nombres
- Apellidos
- Correo electrónico
- Teléfono fijo
- Teléfono móvil

5. Precondiciones

5.1. El operador ha realizado correctamente el inicio de sesión en el sistema.

6. Poscondiciones

6.1. En caso de haberse registrado un usuario exitosamente, los datos de este deben quedar almacenados en la base de datos.

6.2. En caso de haberse modificado un usuario exitosamente, los datos de este deben estar actualizados en la base de datos.

6.3. En caso de haberse eliminado exitosamente un usuario, los datos de esta deben ser borrados de la base de datos.



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira

CU-02 Gestionar Acometidas

**Versión 1.0
10/04/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-02 Gestionar Acometidas	

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
10/Abril/2016	1.0	Elaboración del documento	Hernan Darío Gonzalez

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-02 Gestionar Acometidas	

Tabla de contenido

1. Descripción breve:.....	1
2. Flujo Básico de Eventos	1
2.1. Registrar Acometida.	1
2.2. Modificar Acometida	1
2.3. Eliminar Acometida	2
2.4. Fin del caso de uso	2
3. Flujos Alternos.....	2
3.1. Información requerida incompleta: Registrar Acometida	2
3.2. Información requerida incompleta: Modificar Acometida.	2
4. Sub-flujos	2
4.1. Seleccionar dirección de facturación diferente de la dirección de instalación	2
4.2. Buscar acometida	2
4.3. Adicionar servicios.....	2
4.4. Quitar servicios.....	3
5. Precondiciones	3
6. Post-condiciones	3

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-02 Gestionar Acometidas	

ESPECIFICACION DEL CASO DE USO: CU-02 GESTIONAR ACOMETIDAS

1. Descripción breve:

Este caso de uso es iniciado por el Operador y en él se permite:

- Registrar la acometida
- Modificar la acometida
- Eliminar la acometida

2. Flujo Básico de Eventos

2.1. Registrar Acometida.

2.1.1. El Operador digita el número del documento de identificación del suscriptor al cual se le asignará la acometida.

2.1.2. El operador ingresa la información correspondiente a la acometida, la cual consta de:

- Dirección del predio
- Número catastral del predio
- Barrio o Vereda
- Zona de instalación
- Tipo de predio
- Uso del predio

2.1.3. El operador asigna los servicios y tarifas que se le prestarán a la acometida.

2.1.4. Se registra la acometida.

2.2. Modificar Acometida

2.2.1. El operador selecciona una acometida de la lista de acometidas.

2.2.2. El operador modifica la información que requiere de los siguientes datos:

- Suscriptor
- Dirección del predio
- Número catastral del predio
- Barrio o Vereda
- Zona de instalación
- Tipo de predio
- Uso del predio
- Dirección de envío de la factura
- Barrio o vereda de envío de la factura
- Zona de envío de la factura
- Asignar/Quitar servicios

2.2.3. Se modifica la acometida.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-02 Gestionar Acometidas	

2.3. Eliminar Acometida

2.3.1. El operador selecciona una acometida de la lista total de acometidas.

2.3.2. El operador ejecuta el proceso de eliminar la acometida.

2.3.3. El operador debe confirmar la acción de eliminar la acometida

2.4. Fin del caso de uso

El caso de uso finaliza.

3. Flujos Alternos

3.1. Información requerida incompleta: Registrar Acometida

En el paso 2.1.2 Registrar acometida, si la información requerida no se ha ingresado, el sistema muestra un mensaje detallando la información que falta.

El caso de uso continúa en el paso 2.1.1.

3.2. Información requerida incompleta: Modificar Acometida.

En el paso 2.2.2 Modificar acometida, si la información requerida no se ha ingresado, el sistema muestra un mensaje detallando la información que falta.

El caso de uso continúa en el paso 2.2.1.

4. Sub-flujos

4.1. Seleccionar dirección de facturación diferente de la dirección de instalación

En el flujo 2.1.2 Registrar acometida y 2.2.2 Modificar acometida si el operario selecciona enviar factura a otra dirección, se deben ingresar los siguientes datos:

- Dirección de envío de la factura
- Barrio o Vereda de envío de la factura
- Zona de envío de la factura

Continúa en el flujo 2.1.3 Registrar acometida o 2.2.3 Modificar acometida.

4.2. Buscar acometida

En el flujo 2.2.1 Modificar acometida se puede buscar la acometida por los campos visibles en la lista de las acometidas:

- Nombres y/o Apellidos del suscriptor
- Dirección del predio
- Barrio o Vereda
- Zona de instalación
- Tipo de predio
- Uso del predio

4.3. Adicionar servicios

En el flujo 2.1.3 Registrar acometida y 2.2.2 Modificar acometida para adicionar servicios el operario debe:

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-02 Gestionar Acometidas	

- Seleccionar el Tipo de concepto
- Seleccionar el Concepto
- Seleccionar la Tarifa
- Agregar el servicio

Este proceso se repite las veces que se requiera agregar un servicio

4.4. Quitar servicios

En el flujo 2.1.3 Registrar acometida y 2.2.2 Modificar acometida para quitar servicios el operario debe:

- Seleccionar el servicio de la lista de servicios asignados a la acometida
- Seleccionar “Quitar” el servicio

5. Precondiciones

- 5.1.** El operador ha realizado correctamente el inicio de sesión en el sistema.
- 5.2.** El suscriptor al cual se le asignará la acometida debe estar registrado previamente en el sistema

6. Post-condiciones

- 6.1.** En caso de haberse registrado una acometida exitosamente los datos de esta deben quedar almacenados en la base de datos.
- 6.2.** En caso de haberse modificado la acometida exitosamente, los datos de esta deben estar actualizados en la base de datos.
- 6.3.** En caso de haberse eliminado exitosamente una acometida, los datos de esta deben ser borrados de la base de datos.

**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

CU-03 Gestionar Facturación

**Versión 1.0
10/04/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-03 Gestionar Facturación	

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
10/Abril/2016	1.0	Elaboración del documento	Hernan Darío Gonzalez

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-03 Gestionar Facturación	

Tabla de contenido

1. Descripción breve.....	1
2. Flujo básico de eventos	1
2.1. Registrar facturación del mes	1
2.2. Generar documento de las facturas	1
2.3. Generar copia de la última factura de una acometida	1
2.4. Fin del caso de uso	1
3. Flujos Alternos.....	1
3.1. Facturas del mas ya generadas.....	1
3.2. Descargar archivo desactivado	1
4. Sub-flujos	2
4.1. Buscar acometida	2
5. Precondiciones	2
6. Post-condiciones	2

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-03 Gestionar Facturación	

ESPECIFICACION DEL CASO DE USO: CU-03 GESTIONAR FACTURACIÓN

1. Descripción breve

Este caso de uso es iniciado por el Operador y en él se permite:

- Realizar la facturación del periodo actual
- Generar documento de las facturas
- Generar copia de la última factura de una acometida

2. Flujo básico de eventos

2.1. Registrar facturación del mes

2.1.1. El operador lanza la operación de facturación

2.1.2. El sistema registra las facturas de cada acometida con la información de los servicios asignados correspondientes al mes actual.

2.2. Generar documento de las facturas

2.2.1. El operador lanza la operación de generar archivo.

2.2.2. El sistema crea un archivo pdf con todas las facturas correspondientes al mes actual.

2.2.3. El sistema muestra un mensaje informando el éxito de la generación del archivo

2.2.4. El operador descarga el archivo

2.3. Generar copia de la última factura de una acometida

2.3.1. El operador selecciona la acometida requerida de la tabla de registros.

2.3.2. Se mostrara un mensaje informado el éxito de la generación del archivo.

2.3.3. El operador descarga el archivo

2.4. Fin del caso de uso

2.4.1. El caso de uso finaliza

3. Flujos Alternos

3.1. Facturas del mas ya generadas

En el paso 2.1.1 Registrar facturación del mes si ya se ha realizado la facturación del periodo actual, no estará disponible la opción hasta el día 1 del siguiente mes.

3.2. Descargar archivo desactivado

En el paso 2.2.4 Generar documento de las facturas y 2.3.3 Generar copia de la última factura de una acometida no se permite generar la descarga debido a error con la creación del archivo.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-03 Gestionar Facturación	

4. Sub-flujos

4.1. Buscar acometida

En el flujo 2.3.1 Generar copia de la última factura de una acometida se puede buscar la acometida por los campos visibles en la lista de las acometidas:

- Nombres y/o Apellidos del suscriptor
- Dirección del predio
- Barrio o Vereda
- Zona de instalación
- Tipo de predio
- Uso del predio

5. Precondiciones

- 5.1. El operador ha realizado correctamente el inicio de sesión en el sistema.
- 5.2. No haberse realizado Las acometidas al cual se le asignará la acometida debe estar registrado previamente en el sistema

6. Post-condiciones

- 6.1. En caso de haberse ejecutado la facturación exitosamente, los datos de las facturas deben quedar almacenados en la base de datos.

**Sistema para la gestión
administrativa del acueducto rural
Ayacucho La Buitrera, Palmira
CU-02 Gestionar Recaudo**

**Versión 1.0
10/04/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-04 Gestionar Recaudos	

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
10/Abril/2016	1.0	Elaboración del documento	Hernan Darío Gonzalez

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-04 Gestionar Recaudos	

Tabla de contenido

1. Descripción Breve.....	1
2. Flujo Básico de Eventos.	1
2.1. Registrar Pago por Acometida.	1
2.2. Registrar Pago por Servicio.....	1
2.3. Registrar otros pagos.....	1
2.4. Fin caso de uso	1
3. Flujos Alternos.....	1
3.1. Información requerida incompleta.....	1
4. Sub-flujos.	1
4.1. Buscar acometida	1
5. Precondiciones.	2
5.1. Registrar pagos acometida.	2
5.2. Registrar Pagos Servicios	2
5.3. Registrar Otros Pagos.	2
6. Post-condiciones.	2

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-04 Gestionar Recaudos	

ESPECIFICACION DEL CASO DE USO: CU-04 GESTIONAR RECAUDOS

1. Descripción Breve

Este caso de uso es iniciado por el Operador y en él se permite:

- Registrar pago por Acometida
- Registrar pago por Servicio
- Registrar otros pagos

2. Flujo Básico de Eventos.

2.1. Registrar Pago por Acometida.

2.1.1. El operador selecciona la acometida de la lista de acometidas

2.1.2. El operador ingresa el monto a pagar

2.1.3. Se registra el pago por acometida

2.2. Registrar Pago por Servicio.

2.2.1. El operador selecciona la acometida de la lista de acometidas

2.2.2. El operador ingresa el monto a pagar

2.2.3. Se registra el pago por servicios

2.3. Registrar otros pagos.

2.3.1. El operador ingresa el número del documento de identificación del usuario

2.3.2. El operador selecciona el tipo de usuario con el que se va a registrar el pago

2.3.3. El operador ingresa la descripción del movimiento y el monto a pagar

2.3.4. El operador selecciona la cuenta que se debitará y la cuenta que se acreditará en ese movimiento

2.3.5. Se repite desde 2.3.3 las veces que se requieran adicionar movimientos

2.3.6. Se registra el pago.

2.4. Fin caso de uso

El caso de uso finaliza

3. Flujos Alternos.

3.1. Información requerida incompleta

3.1.1. En el paso 2.1.3 Registrar pago acometida, 2.2.3 Registrar pago servicios y 2.3.5 Registrar otros pagos si la información requerida no se ha ingresado el sistema indica que información falta para registrar el pago.

4. Sub-flujos.

4.1. Buscar acometida

En el paso 2.1.3 Registrar pago acometida y 2.2.3 Registrar pago servicios, se puede buscar la acometida por los campos visibles en la lista de las acometidas:

- Nombres y/o Apellidos del suscriptor

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Casos de Uso	Fecha: 10/Abril/2016
CU-04 Gestionar Recaudos	

- Dirección del predio
- Barrio o Vereda
- Zona de instalación
- Tipo de predio
- Uso del predio

5. Precondiciones.

5.1. Registrar pagos acometida.

5.1.1. El operador ha realizado correctamente inicio de sesión en el sistema

5.1.2. La acometida a la que se le va a registrar el pago ya debe existir

5.2. Registrar Pagos Servicios

5.2.1. El operador ha realizado correctamente inicio de sesión en el sistema

5.2.2. La acometida a la que se le va a registrar el pago ya debe existir en el sistema.

5.3. Registrar Otros Pagos.

5.3.1. El operador ha realizado correctamente inicio de sesión en el sistema

5.3.2. El usuario al que se le va a generar el pago ya debe existir en el sistema.

6. Post-condiciones.



Sistema

Gestionar Usuario

Gestionar Acometida

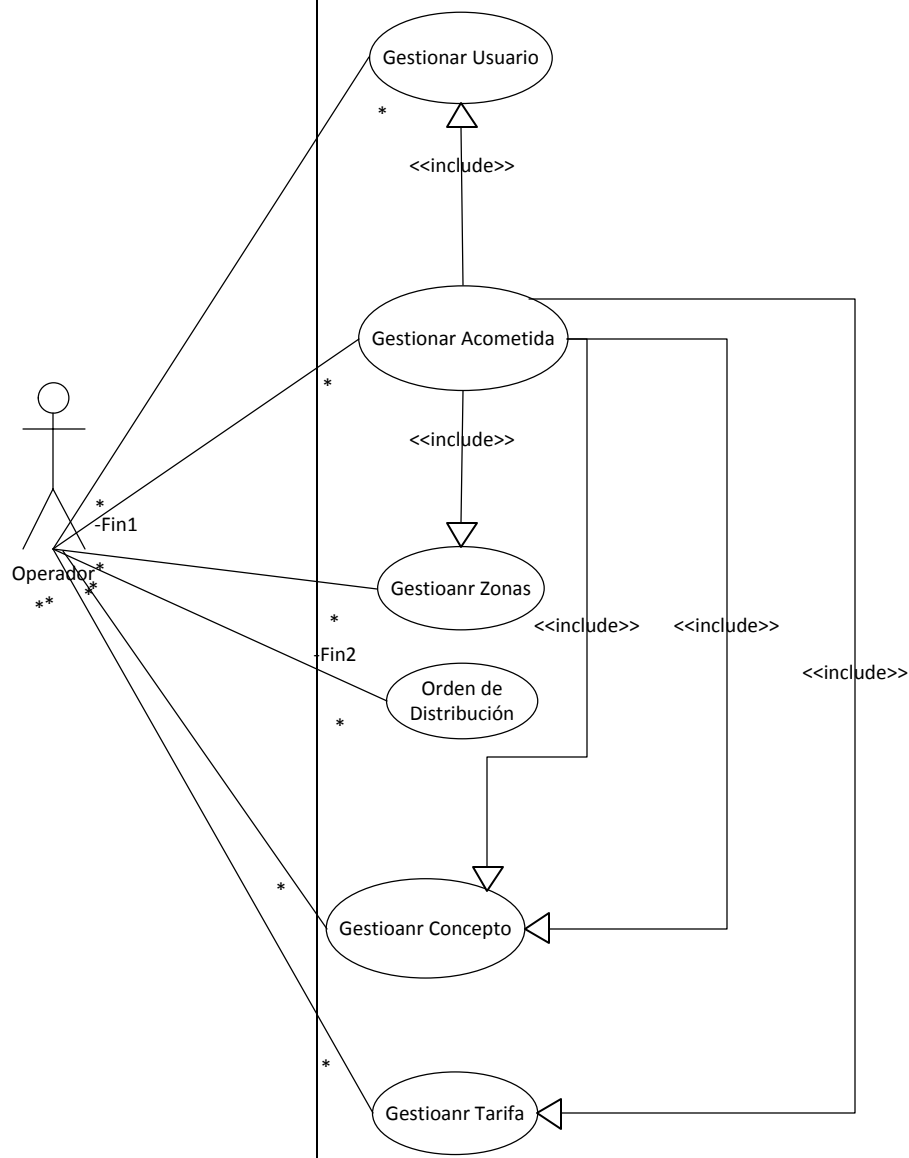
Gestionar Zonas

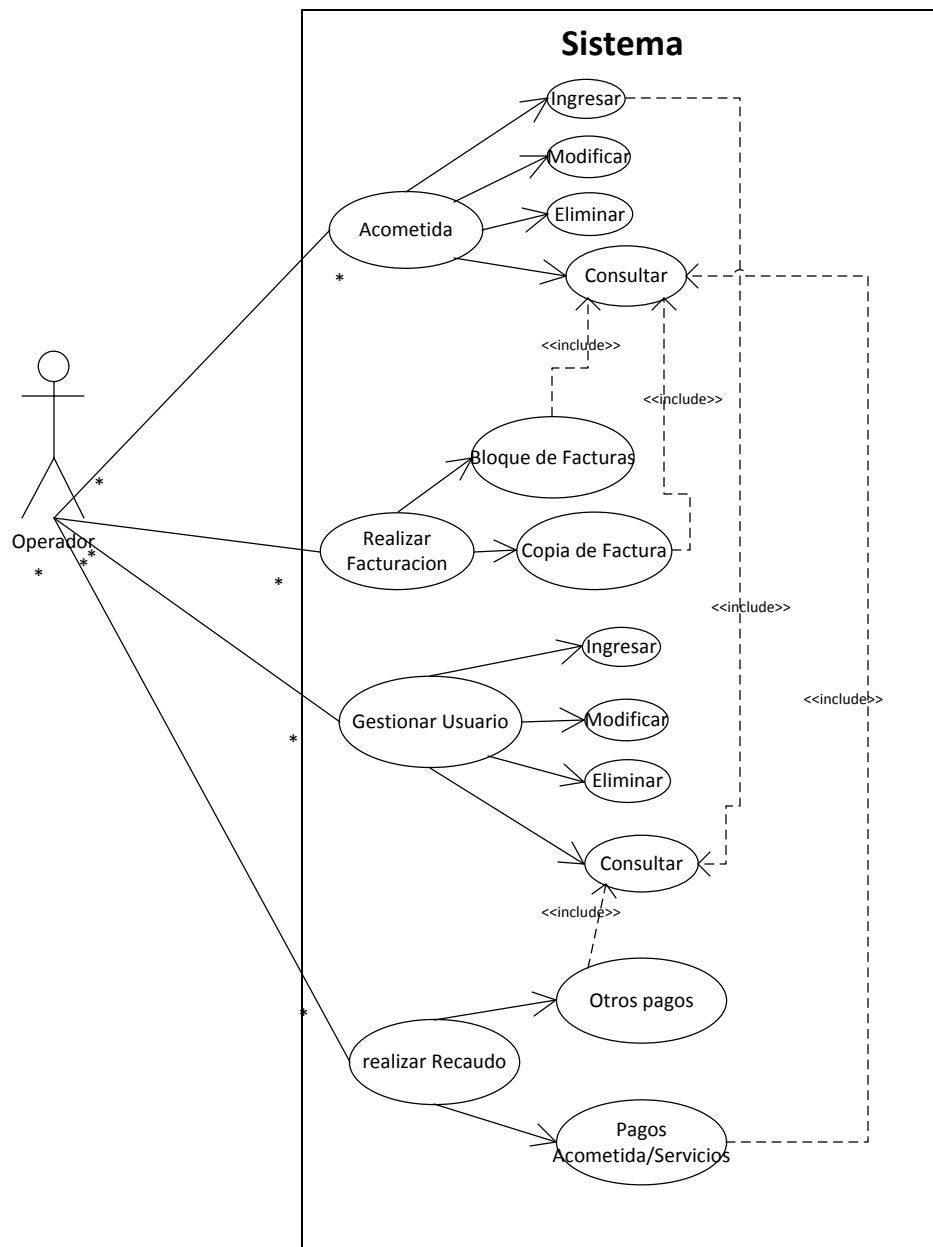
Gestionar Concepto

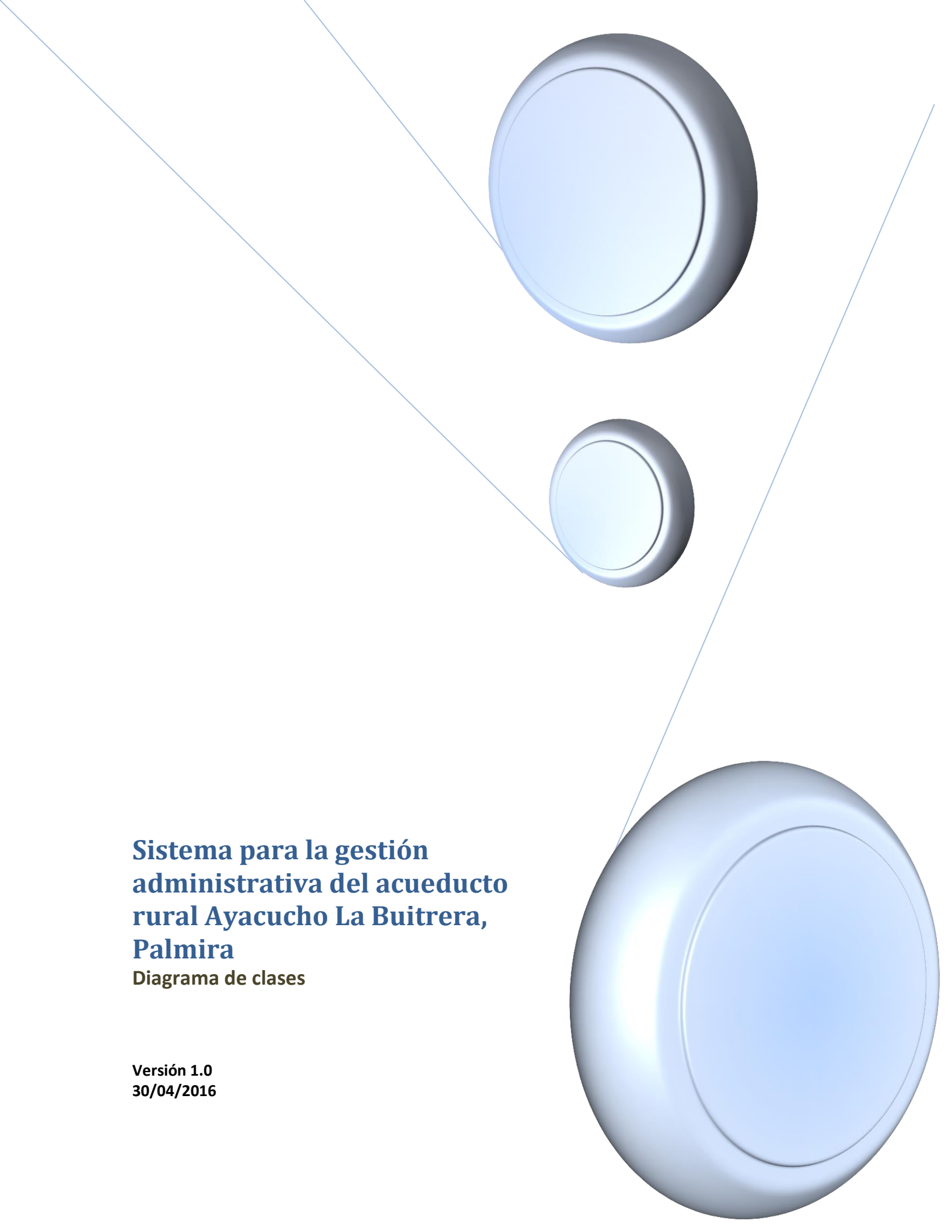
Gestionar Tarifa

Gestionar
Facturacion

Gestionar Recaudo







Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira

Diagrama de clases

**Versión 1.0
30/04/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Diagrama de clases	Fecha: 30/Abril/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
30/Abril/2016	1.0	Propuesta inicial del documento diagrama de clases	Hernán Darío González

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Diagrama de clases	Fecha: 30/Abril/2016

Tabla de contenido

1. Introducción	1
COMPONENTE GESTION.....	2
COMPONENTE FACTURACION.....	3
COMPONENTE CONTABLE.....	4

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Diagrama de clases	Fecha: 30/Abril/2016

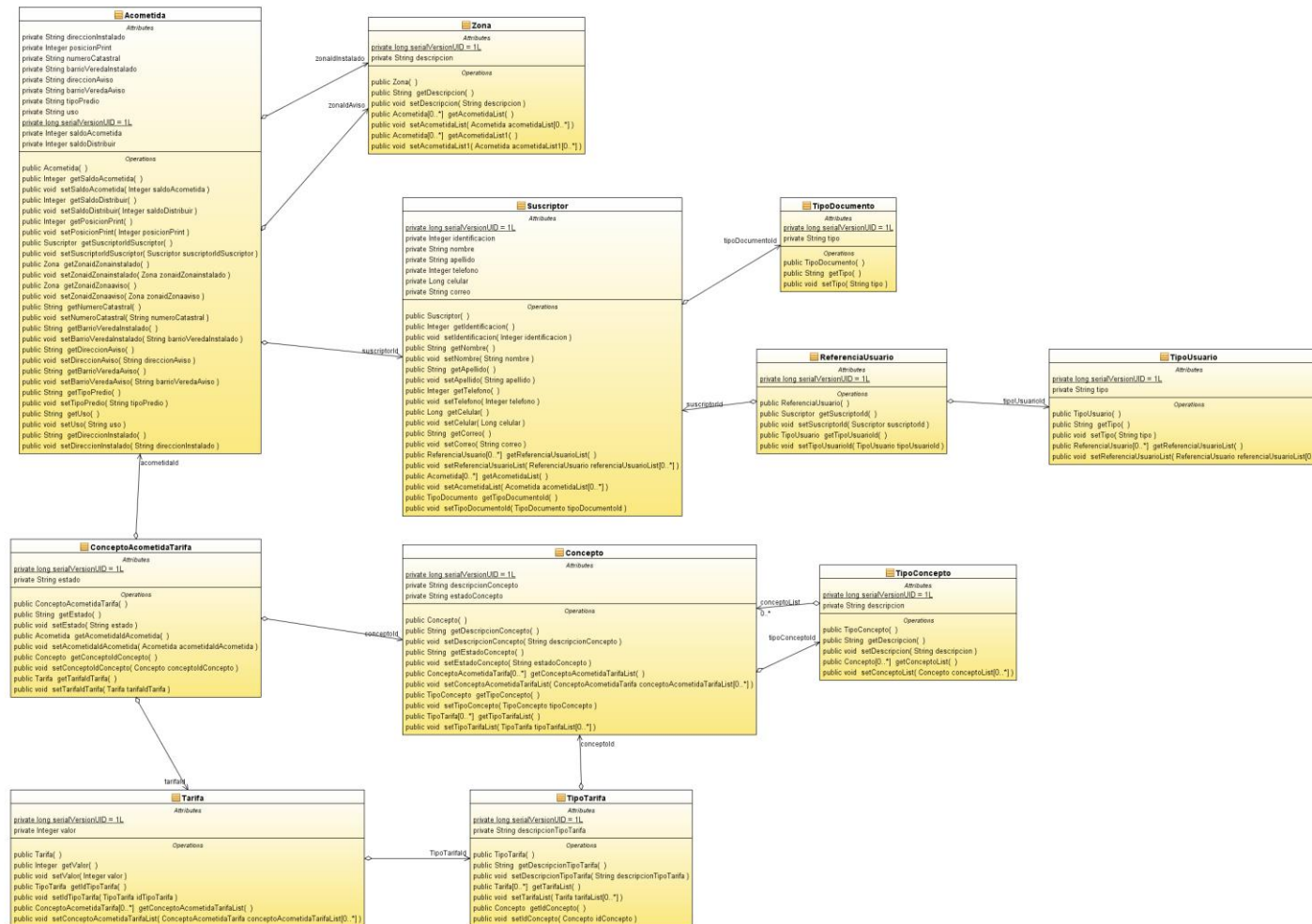
DOCUMENTO DIAGRAMA DE CLASES

1. Introducción

En este documento podremos encontrar la información de cada clase, sus métodos y la relación con las otras clases del componente.

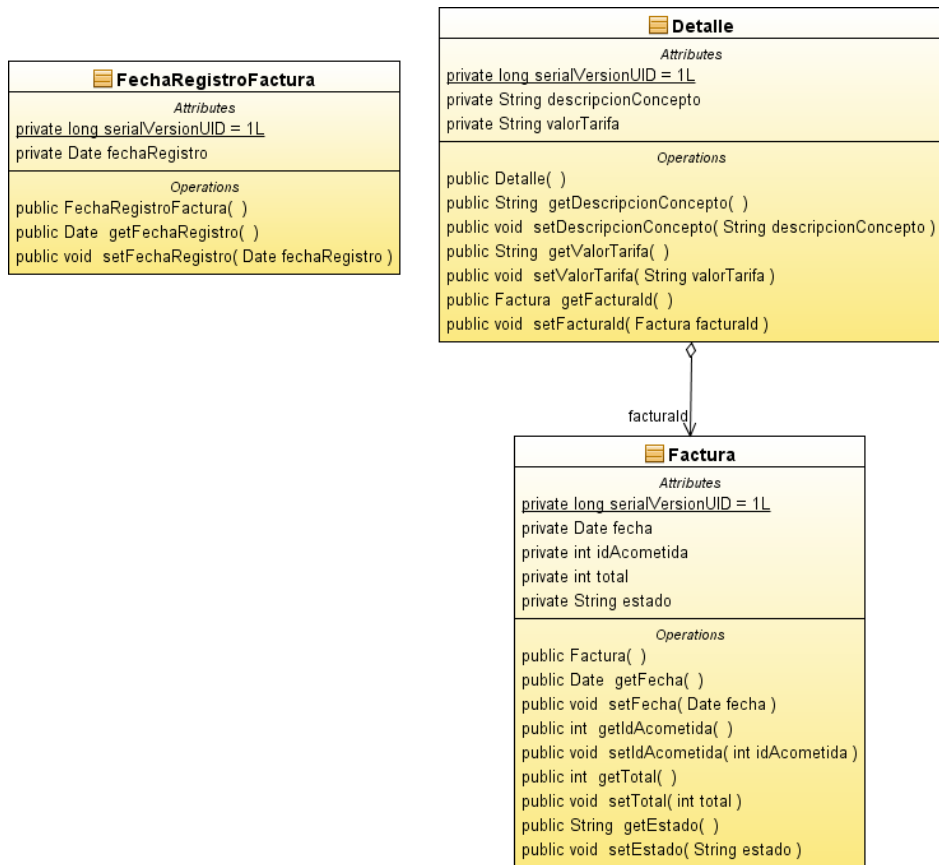
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Diagrama de clases	Fecha: 30/Abril/2016

COMPONENTE GESTION



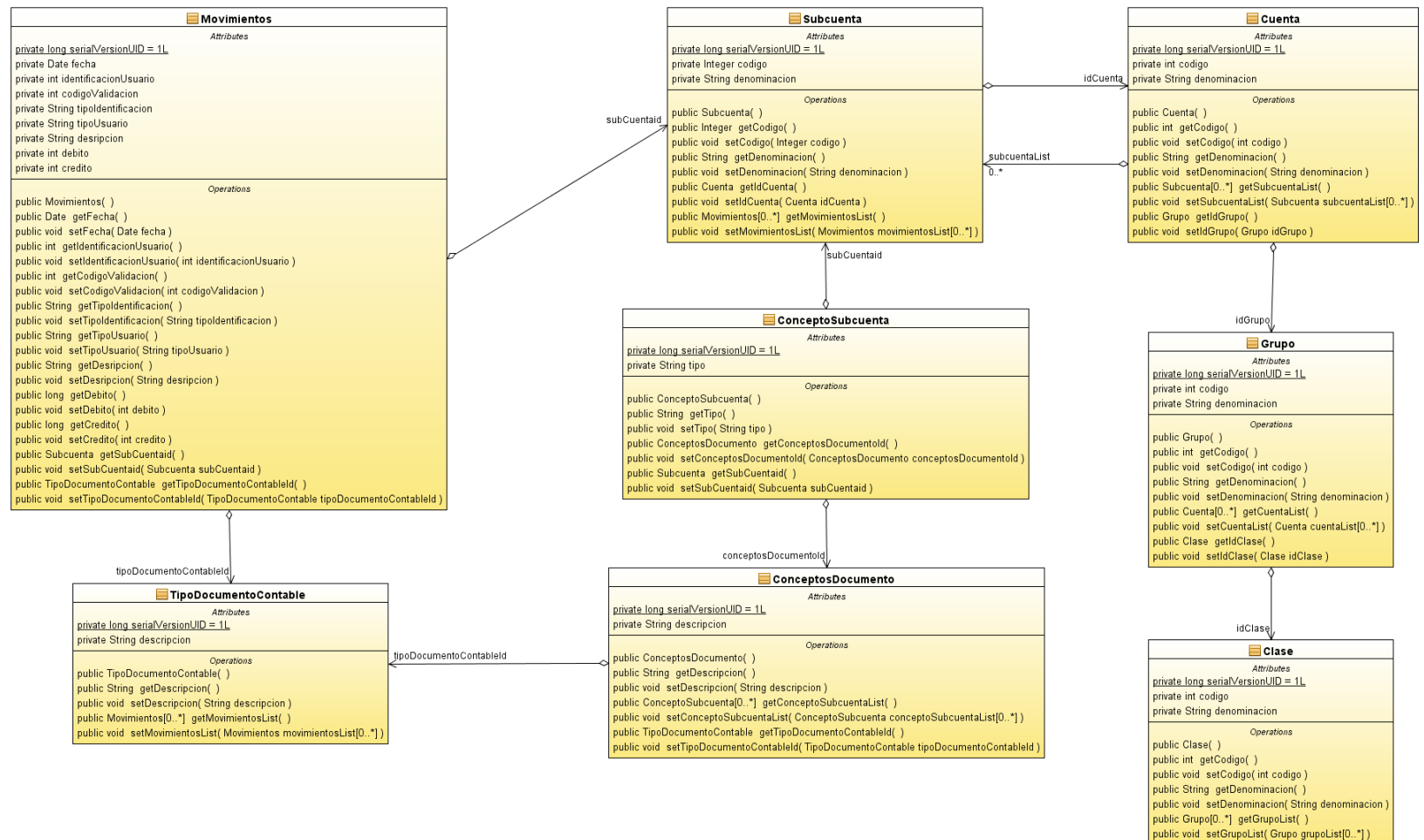
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Diagrama de clases	Fecha: 30/Abril/2016

COMPONENTE FACTURACION



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
Diagrama de clases	Fecha: 30/Abril/2016

COMPONENTE CONTABLE

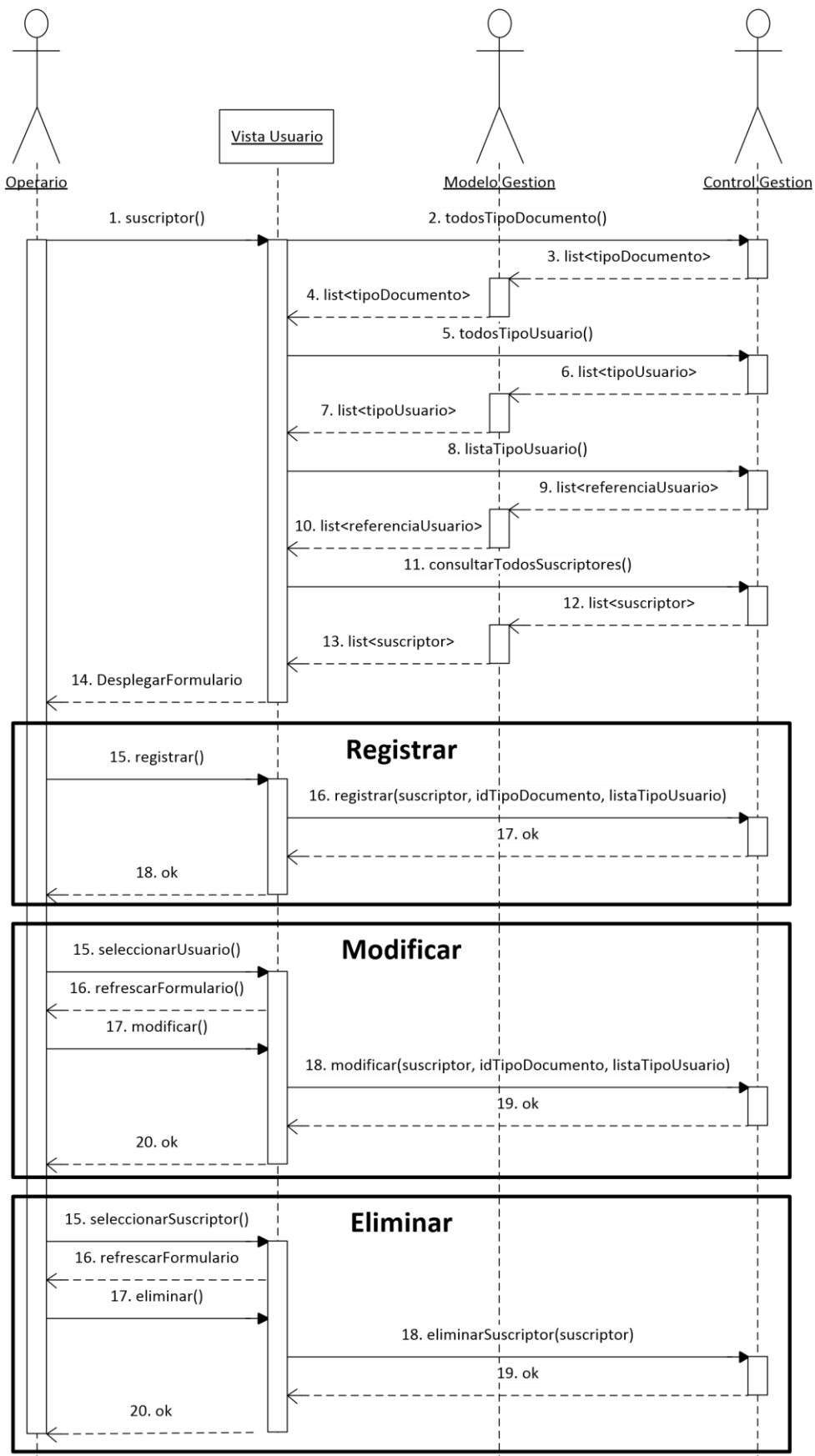


Sistema para la gestión
administrativa del acueducto rural
Ayacucho La Buitrera, Palmira
Diagrama de Secuencias

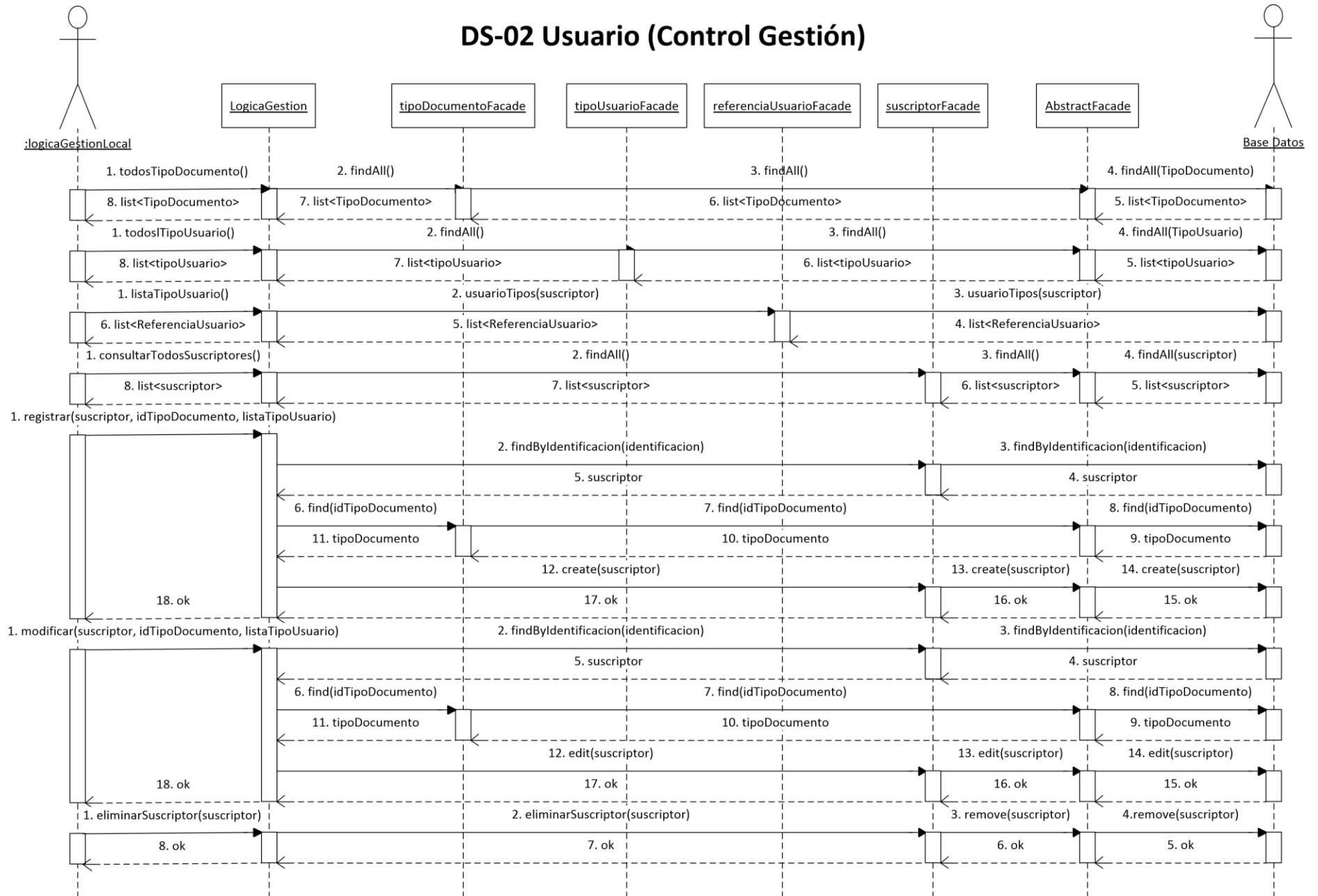
Versión 1.0 30/04/2016



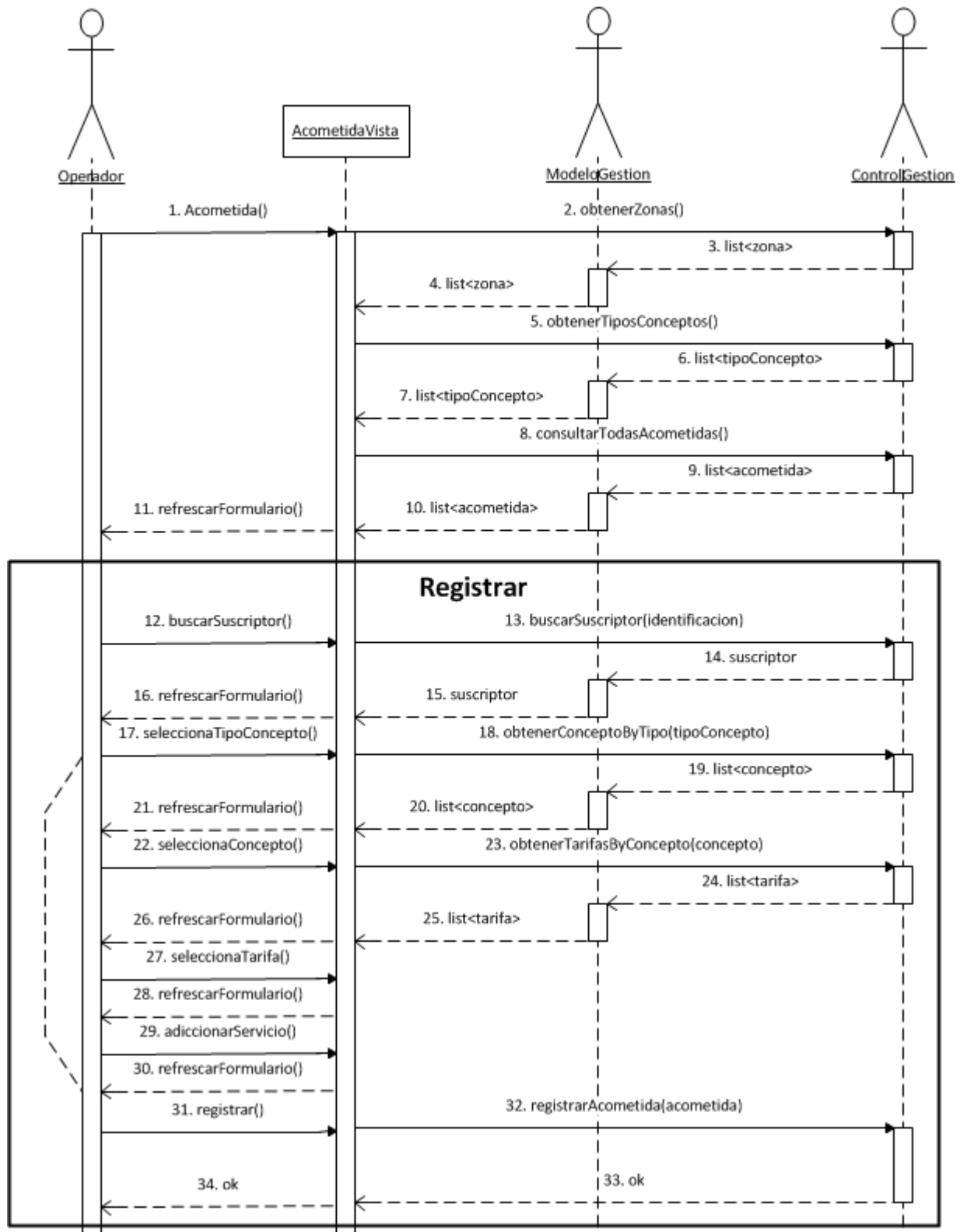
DS-01 Usuario (ProyAyacucho)

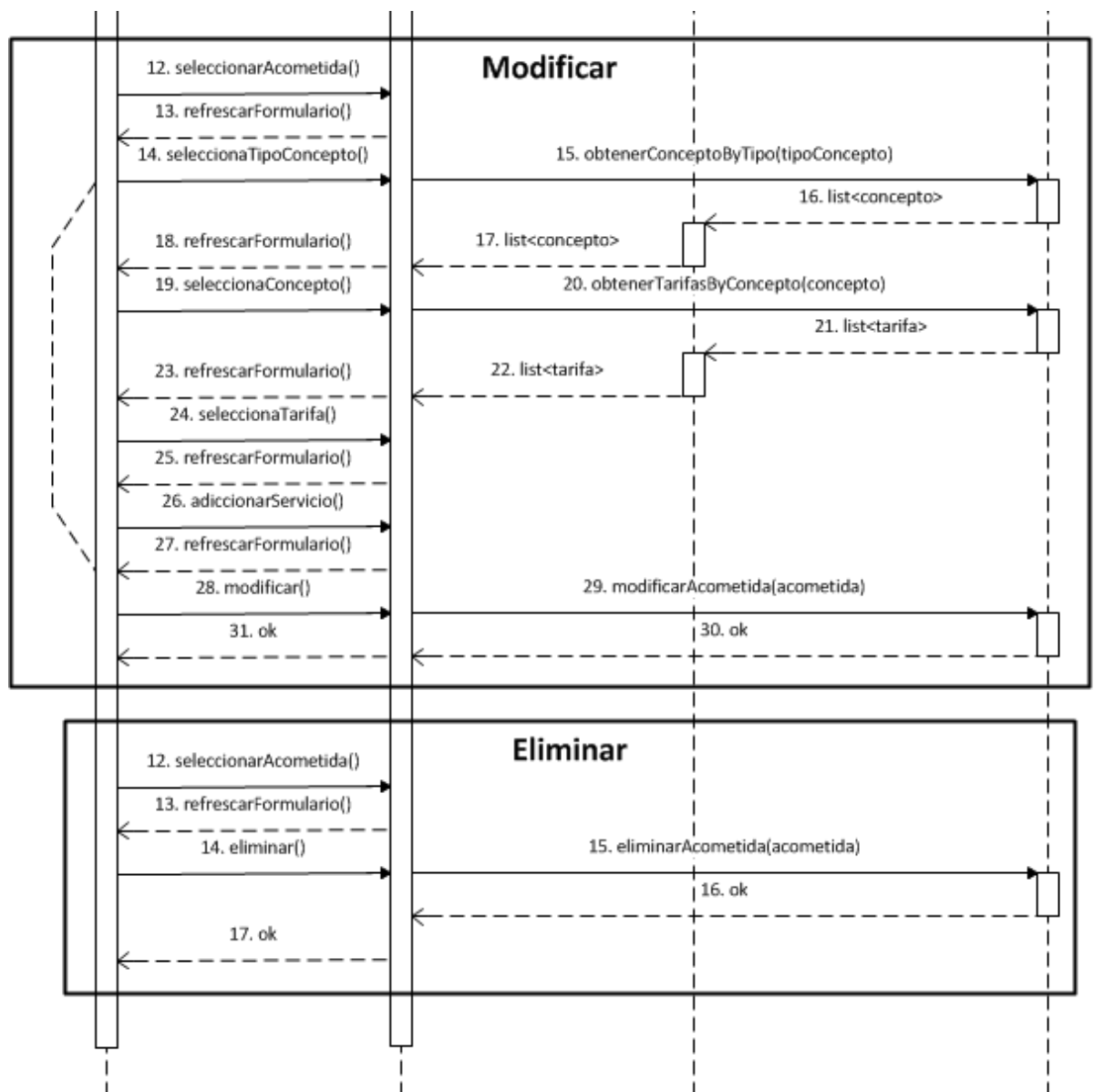


DS-02 Usuario (Control Gestión)

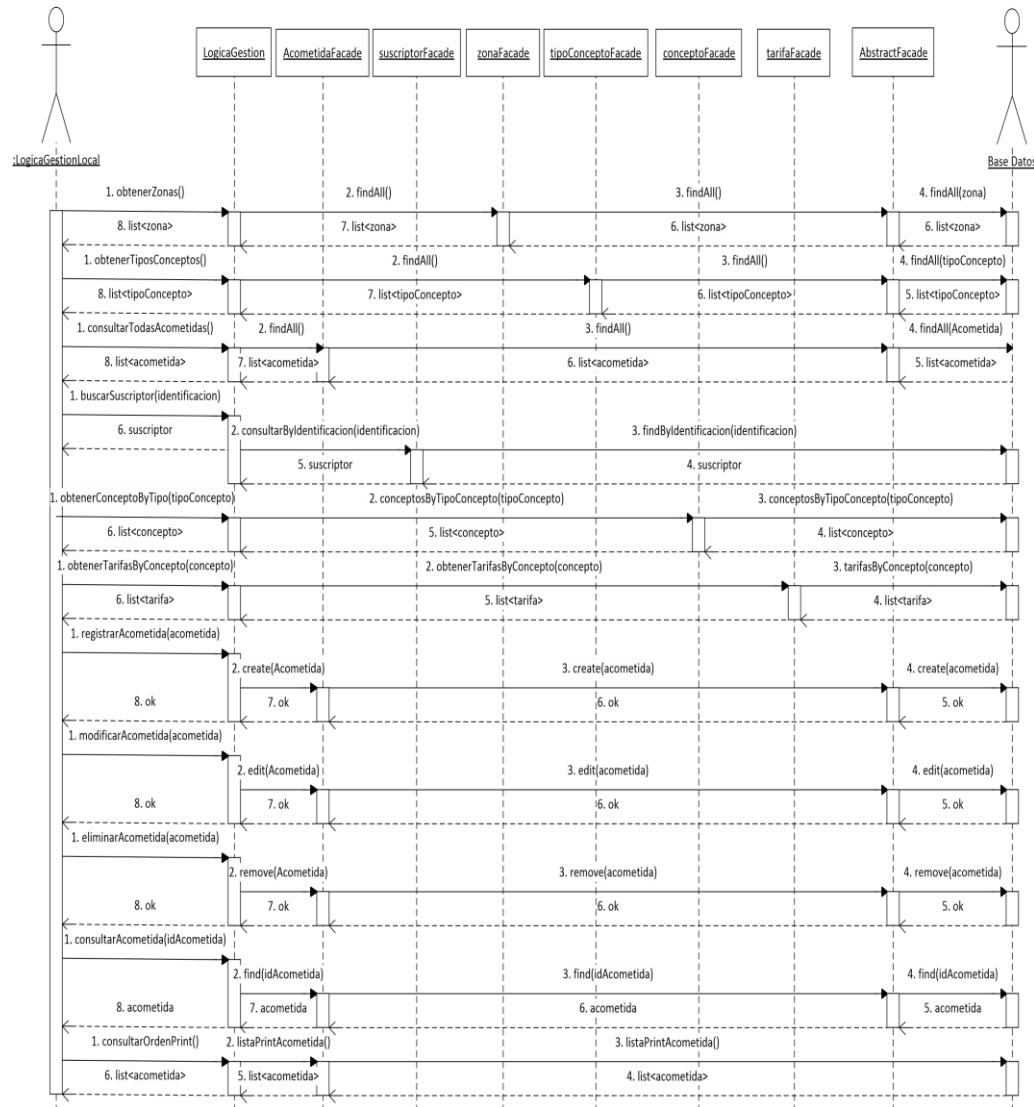


DS-03 Acometida (ProyAyacucho)

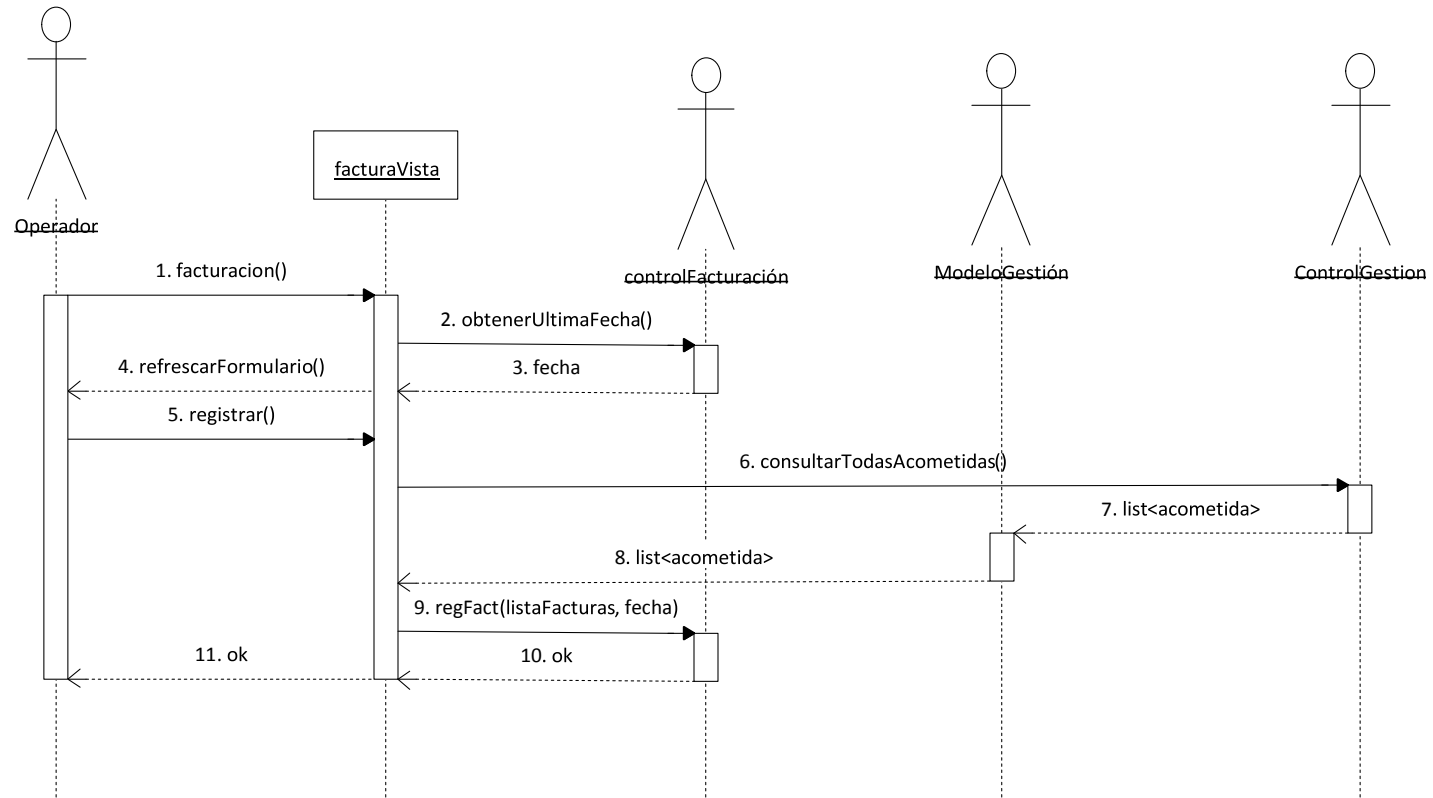




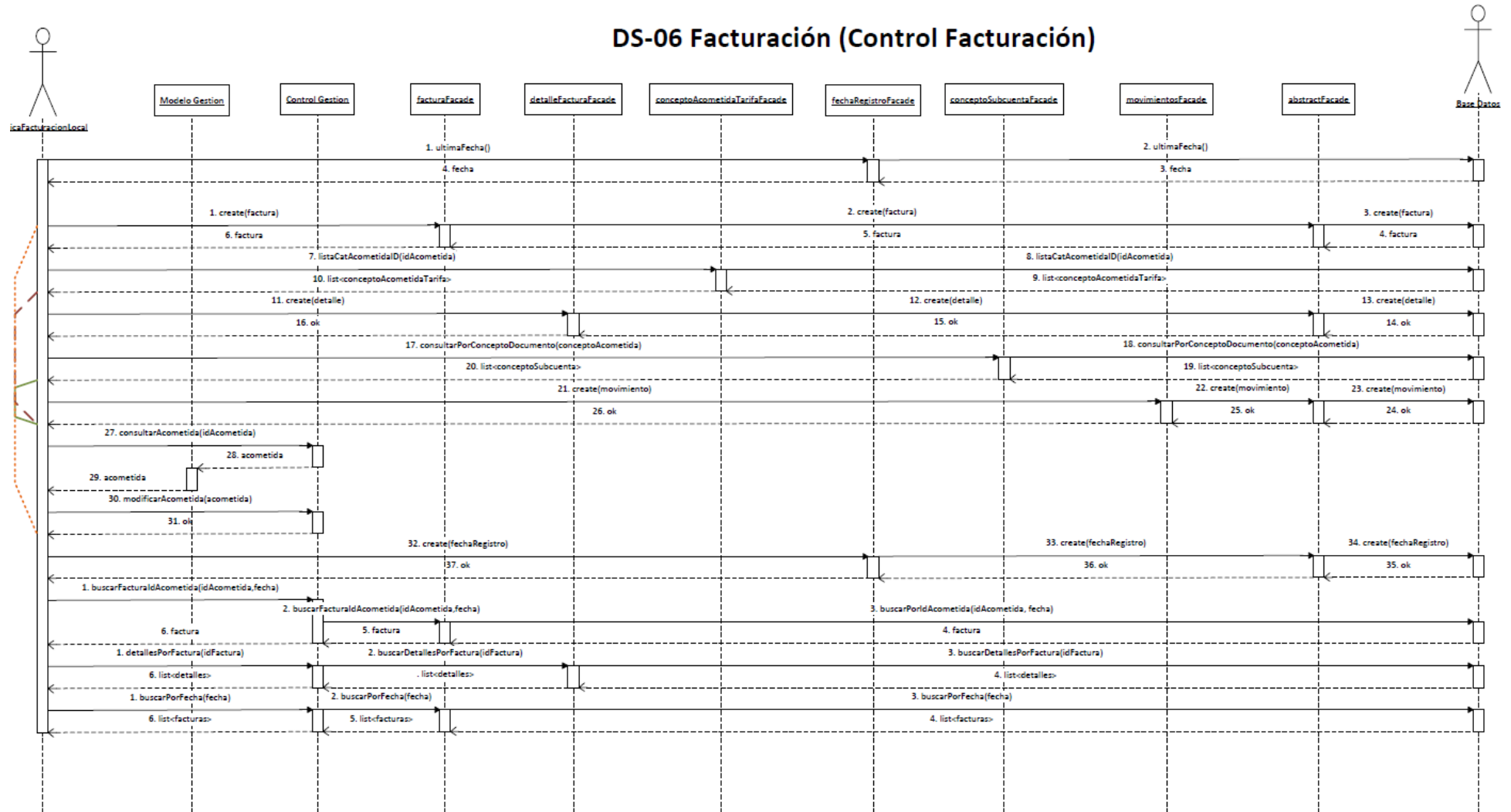
DS-04 Acometida (Control Gestión)



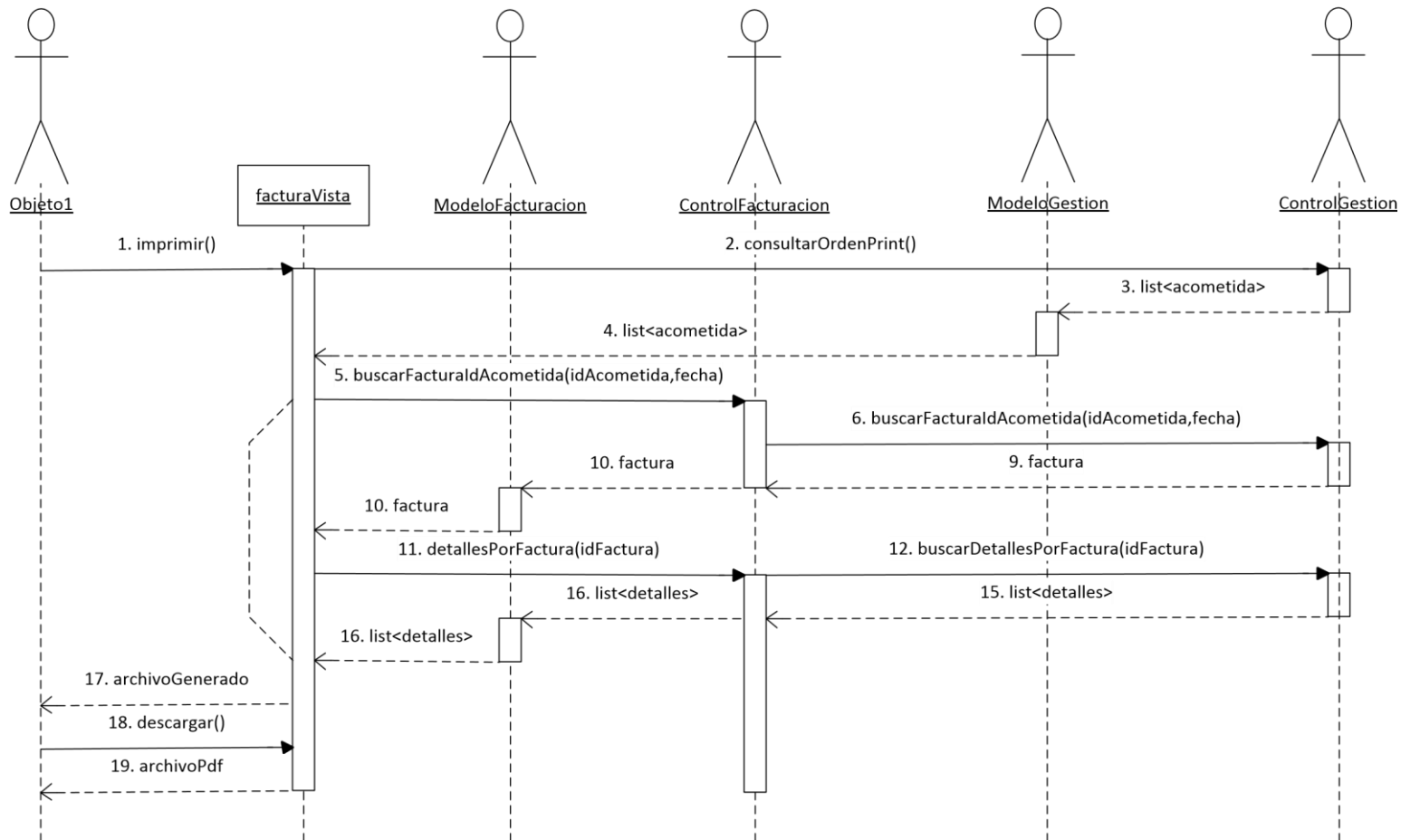
DS-05 Facturación (ProyAyacucho)



DS-06 Facturación (Control Facturación)



DS-08 Generar documento Facturas



**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

DC-01 Control contable

**Versión 1.0
03/05/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-01 Control Contable	Fecha: 3/Mayo/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
3/Mayo/2016	1.0	Propuesta inicial del documento	Hernán Darío Gonzalez

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-01 Control Contable	Fecha: 3/Mayo/2016

Tabla de contenido

1. Introducción	1
2. Modelo Componente	1
3. Clases que conforman el Componente.....	1
3.1. LogicaContable	1
3.1.1. Variables	2
3.1.2. Métodos	2
3.1.3. Componentes Externos	2
3.2. ConceptosDocumentoFacade.....	2
3.2.1. Variables	2
3.2.2. Métodos	2
3.2.3. Componentes Externos.	2
3.3. ConceptoSubcuentaFacade.	3
3.3.1. Variables	3
3.3.2. Métodos	3
3.3.3. Componentes Externos.	3
3.4. CuentaFacade	3
3.4.1. Variables	3
3.4.2. Métodos	3
3.4.3. Componentes Externos	3
3.5. GrupoFacade	3
3.5.1. Variables	3
3.5.2. Métodos	3
3.5.3. Componentes Externos	3
3.6. MovimientosFacade	4
3.6.1. Variables	4
3.6.2. Métodos	4
3.6.3. Componentes Externos	4
3.7. SubcuentaFacade	4
3.7.1. Variables	4

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-01 Control Contable	Fecha: 3/Mayo/2016

3.7.2.	Métodos	4
3.7.3.	Componentes Externos	4
3.8.	TipoDocumentoContableFacade	4
3.8.1.	Variables	4
3.8.2.	Métodos	4
3.8.3.	Componentes Externos.	4

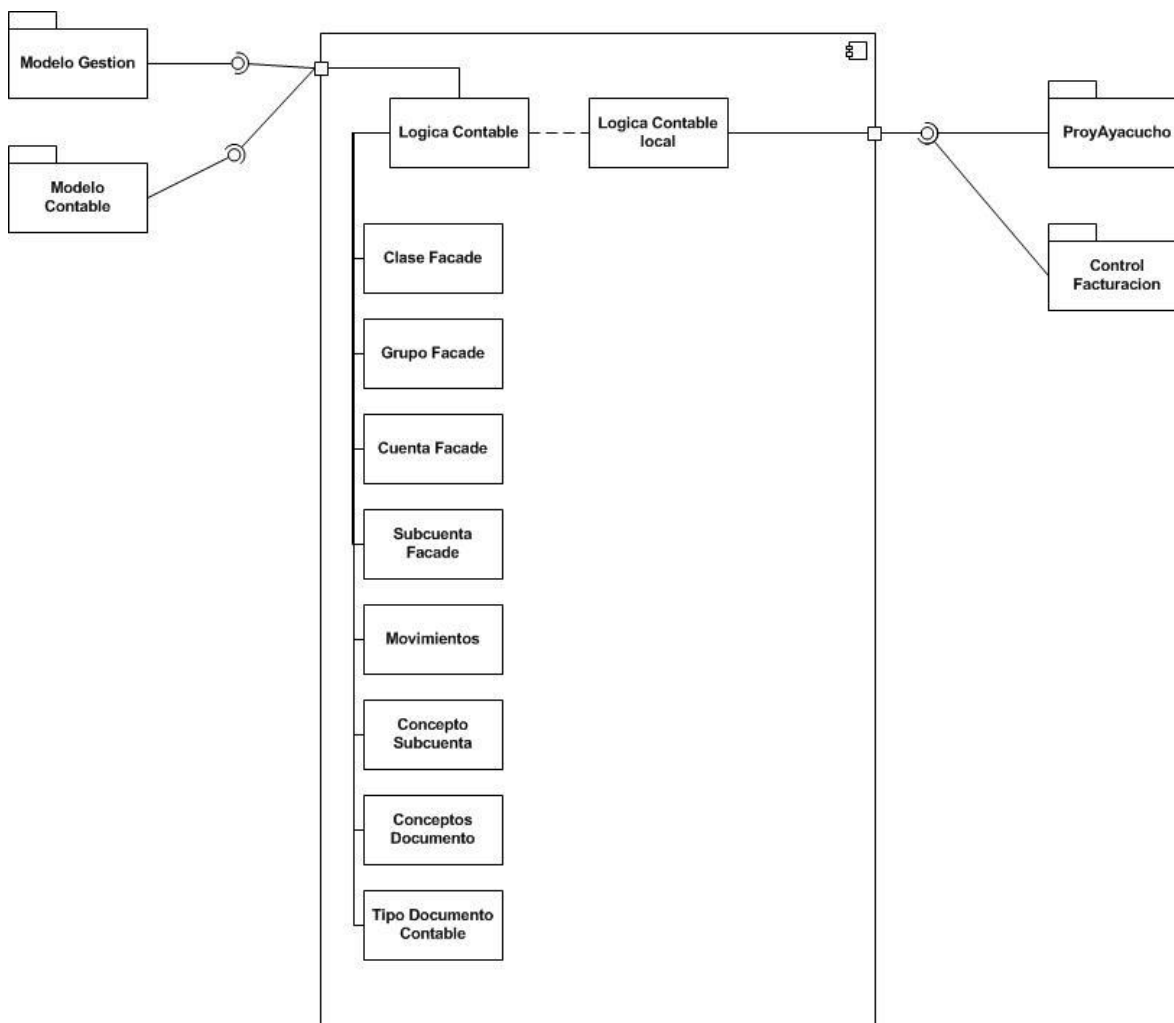
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-01 Control Contable	Fecha: 3/Mayo/2016

CONTROL CONTABLE

1. Introducción

Este documento busca describir que ofrece el componente y que necesita para su funcionamiento.

2. Modelo Componente



3. Clases que conforman el Componente.

3.1. LogicaContable.

En esta clase se encontraran todos los métodos que permitirán realizar las distintas operaciones relacionadas con el componente, permitiendo crear, modificar, eliminar los

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-01 Control Contable	Fecha: 3/Mayo/2016

distintos registros en la base de datos, y realizar las consultas necesarias para la gestión de los datos.

3.1.1. Variables

tiDoCoDAO
subCuentaDAO
movimientosDAO
tipoUsuarioDAO
conceptoSubcuentaDAO
acometidaDAO
conceptosDocumentosDAO

3.1.2. Métodos

List<ConceptoSubcuenta> consultarPorConcepto(String conceptosDocumentoDescripcion)
List<ConceptosDocumento> consultarPorConcepto(Integer conceptosDocumentId)
void registrarMovimiento(String tipo, Date fecha, Integer idAcometida, String descripcion, Integer pago)
void registrarMovimientos(List<Movimientos> movimientos)
List<Subcuenta> consultarTodasSubcuentas()
Subcuenta buscarCuentaId(Integer id)
TipoDocumentoContable buscarTipoDocumentoContableId(Integer id);

3.1.3. Componentes Externos

PersistenciaAPI
Modelo Gestion
Modelo Contable.

3.2. ConceptosDocumentoFacade

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.2.1. Variables

em

3.2.2. Métodos

getEntityManager()
ConceptosDocumentoFacade()

3.2.3. Componentes Externos.

PersistenciaAPI
ModeloContable

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-01 Control Contable	Fecha: 3/Mayo/2016

3.3. ConceptoSubcuentaFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.3.1. Variables

em

3.3.2. Métodos

getEntityManager()

ConceptoSubcuentaFacade()

consultarPorConceptoDocumento()

3.3.3. Componentes Externos.

PersistenciaAPI

ModeloContable

3.4. CuentaFacade

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.4.1. Variables

Em

3.4.2. Métodos

getEntityManager()

CuentaFacade()

3.4.3. Componentes Externos

PersistenciaAPI

ModeloContable

3.5. GrupoFacade

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.5.1. Variables

em

3.5.2. Métodos

getEntityManager()

GrupoFacade()

3.5.3. Componentes Externos

PersistenciaAPI

ModeloContable

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-01 Control Contable	Fecha: 3/Mayo/2016

3.6. MovimientosFacade

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.6.1. Variables

em

3.6.2. Métodos

getEntityManager()

MovimientosFacade()

3.6.3. Componentes Externos

PersistenciaAPI

ModeloContable

3.7. SubcuentaFacade

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.7.1. Variables

em

3.7.2. Métodos

getEntityManager()

SubcuentaFacade()

3.7.3. Componentes Externos

PersistenciaAPI

ModeloContable

3.8. TipoDocumentoContableFacade

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.8.1. Variables

em

3.8.2. Métodos

getEntityManager()

TipoDocumentoContableFacade()

3.8.3. Componentes Externos.

PersistenciaAPI

ModeloContable

**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

DC-02 Control Facturación

**Versión 1.0
05/05/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-02 Control Facturación	Fecha: 5/Mayo/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
5/Mayo/2016	1.0	Propuesta inicial del documento	Hernán Darío González

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-02 Control Facturación	Fecha: 5/Mayo/2016

Tabla de contenido

1. Introducción	1
2. Modelo Componentes.....	1
3. Clases que conforman el Componente.....	1
3.1. LogicaFacturacion	1
3.1.1. Variables	1
3.1.2. Métodos	2
3.1.3. Componentes Externos	2
3.2. DetalleFacade	2
3.2.1. Variables	2
3.2.2. Métodos	2
3.2.3. Componentes Externos	2
3.3. FacturaFacade	2
3.3.1. Variables	2
3.3.2. Métodos	3
3.3.3. Componentes Externos	3
3.4. FechaRegistroFacturaFacade	3
3.4.1. Variables	3
3.4.2. Métodos	3
3.4.3. Componentes Externos.	3

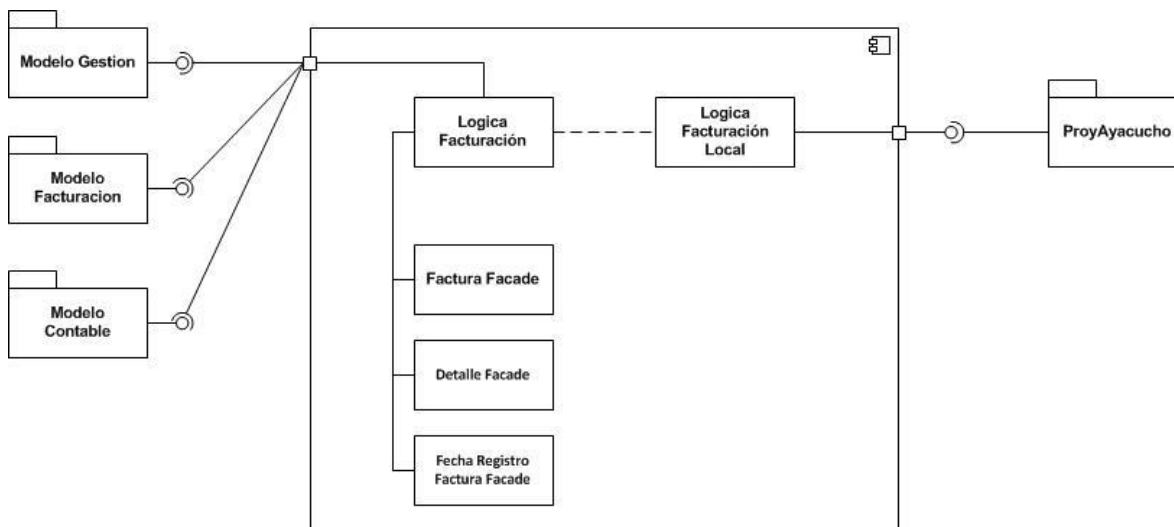
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-02 Control Facturación	Fecha: 5/Mayo/2016

CONTROL FACTURACION

1. Introducción

Este documento busca describir que ofrece el componente y que necesita para su funcionamiento.

2. Modelo Componentes



3. Clases que conforman el Componente

3.1. LogicaFacturacion

En esta clase se encontraran todos los métodos que permitirán realizar las distintas operaciones relacionadas con el componente, permitiendo crear, modificar, eliminar los distintos registros en la base de datos, y realizar las consultas necesarias para la gestión de los datos.

3.1.1. Variables

FacturaDAO
DetalleDAO
CatDAO
RegfecDAO
AcometidaDAO
TipoUsuarioDAO
ConceptoSubcuentaDAO
MovimientosDAO
regfecDAO

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-02 Control Facturación	Fecha: 5/Mayo/2016

3.1.2. Métodos

```

void regFact(List<Factura> listaFacturas, Date fecha)
void registrar(Factura factura)
Factura consultarPorId(Integer id)
List<Detalle> detallesPorFactura(int id)
List<Factura> consultarTodasFacturas()
List<Factura> buscarPorFecha(String fecha)
Factura buscarFacturaIdAcometida(Integer idAcometida, Date f)
List<Factura> buscarFacturaIdAcometidaNP(Integer idAcometida)
void modificarFactura(Factura factura)
void generarPago(Integer idAcometida, int pago)
boolean existeFecha(Date fecha);
String ultimaFecha();

```

3.1.3. Componentes Externos

```

PersistenciaAPI
Modelo Gestion
Modelo Facturacion
Modelo Contable

```

3.2. DetalleFacade

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.2.1. Variables

```
em
```

3.2.2. Métodos

```

getEntityManager()
DetalleFacade()
buscarDetallesPorFactura()

```

3.2.3. Componentes Externos

```

PersistenciaAPI
Modelo Facturacion

```

3.3. FacturaFacade

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.3.1. Variables

```
em
```

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-02 Control Facturación	Fecha: 5/Mayo/2016

3.3.2. Métodos

getEntityManager()
 FacturaFacade()
 buscarPorFecha()
 buscarPorIdAcometida()
 buscarPorIdAcometidaNP()

3.3.3. Componentes Externos

PersistenciaAPI
 Modelo Facturacion

3.4. FechaRegistroFacturaFacade

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.4.1. Variables

em

3.4.2. Métodos

getEntityManager()
 FechaRegistroFacturaFacade()
 ultimaFecha()

3.4.3. Componentes Externos.

PersistenciaAPI
 Modelo Facturacion

**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

DC-03 Control Gestión

**Versión 1.0
05/05/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
5/Mayo/2016	1.0	Propuesta inicial del documento	Hernán Darío Gonzalez

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

Tabla de contenido

1. Introducción.....	1
2. Modelo Componente.....	1
3. Clases que conforman el Componente.....	2
3.1. LogicaGestion.....	2
3.1.1. Variables.....	2
3.1.2. Métodos.....	2
3.1.3. Componentes Externos.....	3
3.2. AcometidaFacade.....	3
3.2.1. Variables.....	3
3.2.2. Metodos.....	3
3.2.3. Componentes Externos.....	4
3.3. ConceptoAcometidaTarifaFacade.....	4
3.3.1. Variables.....	4
3.3.2. Metodos.....	4
3.3.3. Componentes Externos.....	4
3.4. ConceptoFacade.....	4
3.4.1. Variables.....	4
3.4.2. Metodos.....	4
3.4.3. Componentes Externos.....	4
3.5. ReferenciaUsuarioFacade.....	4
3.5.1. Variables.....	5
3.5.2. Metodos.....	5
3.5.3. Componentes Externos.....	5
3.6. SuscriptorFacade.....	5
3.6.1. Variables.....	5
3.6.2. Metodos.....	5
3.6.3. Componentes Externos.....	5
3.7. TarifaFacade.....	5
3.7.1. Variables.....	5

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

3.7.2.	Metodos.	5
3.7.3.	Componentes Externos.	5
3.8.	TipoConceptoFacade.	6
3.8.1.	Variables.	6
3.8.2.	Metodos.	6
3.8.3.	Componentes Externos.	6
3.9.	TipoDocumentoFacade.	6
3.9.1.	Variables.	6
3.9.2.	Metodos.	6
3.9.3.	Componentes Externos.	6
3.10.	TipoTarifaFacade.	6
3.10.1.	Variables.	6
3.10.2.	Metodos.	6
3.10.3.	Componentes Externos.	6
3.11.	TipoUsuarioFacade.	6
3.11.1.	Variables.	6
3.11.2.	Metodos.	7
3.11.3.	Componentes Externos.	7
3.12.	ZonaFacade.	7
3.12.1.	Variables.	7
3.12.2.	Metodos.	7
3.12.3.	Componentes Externos.	7

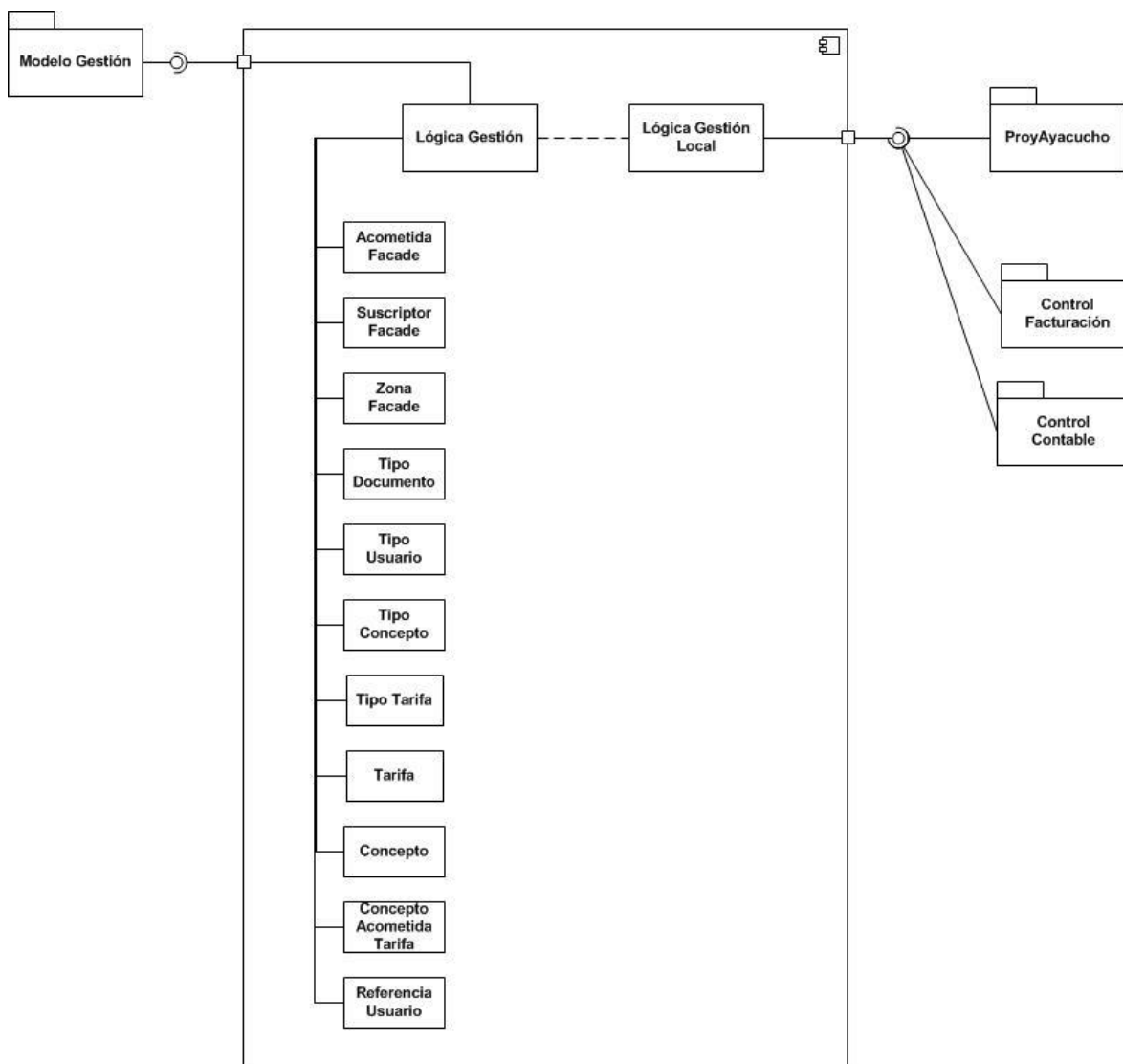
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

CONTROL GESTIÓN

1. Introducción.

Este documento busca describir que ofrece el componente y que necesita para su funcionamiento.

2. Modelo Componente.



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

3. Clases que conforman el Componente.

3.1. LogicaGestion.

En esta clase se encontraran todos los métodos que permitirán realizar las distintas operaciones relacionadas con el componente, permitiendo crear, modificar, eliminar los distintos registros en la base de datos, y realizar las consultas necesarias para la gestión de los datos.

3.1.1. Variables.

acometidaDAO.
 catDAO.
 conceptoDAO.
 referenciaUsuarioDAO.
 suscriptorDao.
 tarifaDAO.
 tipocptoDAO.
 tipoDocumentoDao.
 tipoTarifaDAO.
 tipoUsuarioDao.
 zonaDAO.

3.1.2. Métodos

void registrarAcometida(Acometida acometida)
 void modificarAcometida(Acometida acometida)
 void eliminarAcometida(Acometida acometida)
 Acometida consultarAcometida(int id)
 List<Acometida> consultarTodasAcometida()
 List<Acometida> consultarOrdenPrint()
 List<Acometida> consultarAcometidaZona(Integer idZona)
 void modificarPosicionPrint(Acometida acometida)
 void registrar(ConceptoAcometidaTarifa cat)
 void modificar(ConceptoAcometidaTarifa cat)
 void eliminar(ConceptoAcometidaTarifa cat)
 List<ConceptoAcometidaTarifa> consultarTodosCAT()
 List<ConceptoAcometidaTarifa> listaCatAcometida(Acometida acometida)
 List<ConceptoAcometidaTarifa> listaCatAcometidaID(int id)
 void registrarConcepto(Concepto cpto)
 void modificarConcepto(Concepto cpto)
 void eliminarConcepto(Concepto cpto)
 List<Concepto> obtenerConceptos()
 List<Concepto> obtenerConceptoByTipo(Integer idTipoConcepto)
 void registrarReferenciaUsuario(ReferenciaUsuario referenciaUsuaio)

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

```

List<TipoUsuario> listaTipoUsuario(Suscriptor suscriptor)
List<Suscriptor> consultaraTodosSuscriptor()
void eliminarSuscriptor(Suscriptor suscriptor)
Suscriptor consultarByIdentificacion(Integer identificacion)
List<TipoDocumento> todosTipoDocumento()
List<TipoUsuario> todosTipoUsuario()
void registrarSuscriptor(Suscriptor suscriptor, String idTDocumento, List<String> ItUsuario)
void modificarSuscriptor(Suscriptor suscriptor, String idTDocumento, List<String> ItUsuario,
Boolean condicion)
void registrarTarifa(Tarifa tarifa)
void modificarTarifa(Tarifa tarifa)
void eliminarTarifa(Tarifa tarifa)
List<Tarifa> consultarTodasTarifas()
List<Tarifa> obtenerTarifasByConcepto(Integer concepto)
void registrarTarifaCompleta(Tarifa tarifa, TipoTarifa tipoTarifa)
void registrarTipoConcepto(TipoConcepto tc)
void modificarTipoConcepto(TipoConcepto tc)
void eliminarTipoConcepto(TipoConcepto tc)
List<TipoConcepto> obtenerTipoConceptos()
void registrarTipoTarifa(TipoTarifa tipoTarifa)
void modificarTipoTarifa(TipoTarifa tipoTarifa)
void eliminarTipoTarifa(TipoTarifa tipoTarifa)
List<TipoTarifa> consultarTiposTarifa()
void registrarZona(Zona zona)
void modificarZona(Zona zona)
void eliminarZona(Zona zona)
List<Zona> obtenerZonas()

```

3.1.3. Componentes Externos.

PersistenciaAPI.

Modelo Gestion.

3.2. AcometidaFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.2.1. Variables.

em.

3.2.2. Metodos.

getEntityManager().

AcometidaFacade().

listaPrintAcometida().

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

listaAcometidaZona().

3.2.3. Componentes Externos.

PersistenciaAPI.

Modelo Gestion.

3.3. ConceptoAcometidaTarifaFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.3.1. Variables.

em.

3.3.2. Metodos.

getEntityManager().

ConceptoAcometidaTarifaFacade().

listaCatAcometida().

listaCatAcometidaID().

3.3.3. Componentes Externos.

PersitenciaAPI.

Modelo Gestion.

3.4. ConceptoFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.4.1. Variables.

em.

3.4.2. Metodos.

getEntityManager().

ConceptoFacade().

listaCatAcometida().

conceptosByTipoConcepto().

3.4.3. Componentes Externos.

PersistenciaAPI.

Modelo Gestion.

3.5. ReferenciaUsuarioFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

3.5.1. Variables.

em.

3.5.2. Metodos.

getEntityManager().
ReferenciaUsuarioFacade().
usuarioTipos().
deletAllUsuario().

3.5.3. Componentes Externos.

PersistenciaAPI.
Modelo Gestion.

3.6. SuscriptorFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.6.1. Variables.

em.

3.6.2. Metodos.

getEntityManager().
SuscriptorFacade().
findByIdentificacion().

3.6.3. Componentes Externos.

PersistenciaAPI.
ModeloGestion.

3.7. TarifaFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.7.1. Variables.

em.

3.7.2. Metodos.

getEntityManager().
TarifaFacade().
tarifasByConcepto().

3.7.3. Componentes Externos.

PersistenciaAPI.
ModeloGestion.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

3.8. TipoConceptoFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.8.1. Variables.

em.

3.8.2. Metodos.

TipoConceptoFacade().

3.8.3. Componentes Externos.

PersistenciaAPI.

ModeloGestion.

3.9. TipoDocumentoFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.9.1. Variables.

em.

3.9.2. Metodos.

TipoDocumentoFacade().

3.9.3. Componentes Externos.

PersistenciaAPI.

ModeloGestion.

3.10. TipoTarifaFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.10.1. Variables.

em.

3.10.2. Metodos.

TipoTarifaFacade().

3.10.3. Componentes Externos.

PersistenciaAPI.

ModeloGestion.

3.11. TipoUsuarioFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.11.1. Variables.

em.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-03 Control Gestión	Fecha: 5/Mayo/2016

3.11.2. Metodos.

TipoUsuarioFacade().

3.11.3. Componentes Externos.

PersistenciaAPI.

ModeloGestion.

3.12. ZonaFacade.

En esta clase se encontrara los métodos de comunicación con la base de datos.

3.12.1. Variables.

em.

3.12.2. Metodos.

getEntityManager()

ZonaFacade()

obtenerZonas()

obtenerZonas2()

3.12.3. Componentes Externos.

PersistenciaAPI.

ModeloGestion.

**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

DC-04 Modelo Contable

**Versión 1.0
05/05/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-04 Modelo Contable	Fecha: 5/Mayo/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
5/Mayo/2016	1.0	Propuesta inicial del documento	Hernán Darío González

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-04 Modelo Contable	Fecha: 5/Mayo/2016

Tabla de contenido

1. Introducción.....	1
2. Modelo Componente.	1
3. Clases que conforman el Componente.....	1
3.1. Clase.....	1
3.1.1. Variables.....	1
3.1.2. Métodos.	2
3.1.3. Componentes Externos.	2
3.2. ConceptoDocumento.	2
3.2.1. Variables.....	2
3.2.2. Metodos.	2
3.2.3. Componentes Externos.	2
3.3. ConceptoSubcuenta.	3
3.3.1. Variables.....	3
3.3.2. Metodos.	3
3.3.3. Componentes Externos.	3
3.4. Cuenta.....	3
3.4.1. Variables.....	3
3.4.2. Métodos.	3
3.4.3. Componentes externos.	4
3.5. Grupo.....	4
3.5.1. Variables.....	4
3.5.2. Métodos.	4
3.5.3. Componentes Externos.	4
3.6. Movimiento.....	4
3.6.1. Variables.....	4
3.6.2. Métodos.	5
3.6.3. Componentes Externos.	5
3.7. Subcuenta.....	5
3.7.1. Variables.....	5

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-04 Modelo Contable	Fecha: 5/Mayo/2016

3.7.2.	Métodos.	5
3.7.3.	Componentes Externos.	6
3.8.	TipoDocumentoContable.	6
3.8.1.	Variables.	6
3.8.2.	Métodos.	6
3.8.3.	Componentes Externos.	6

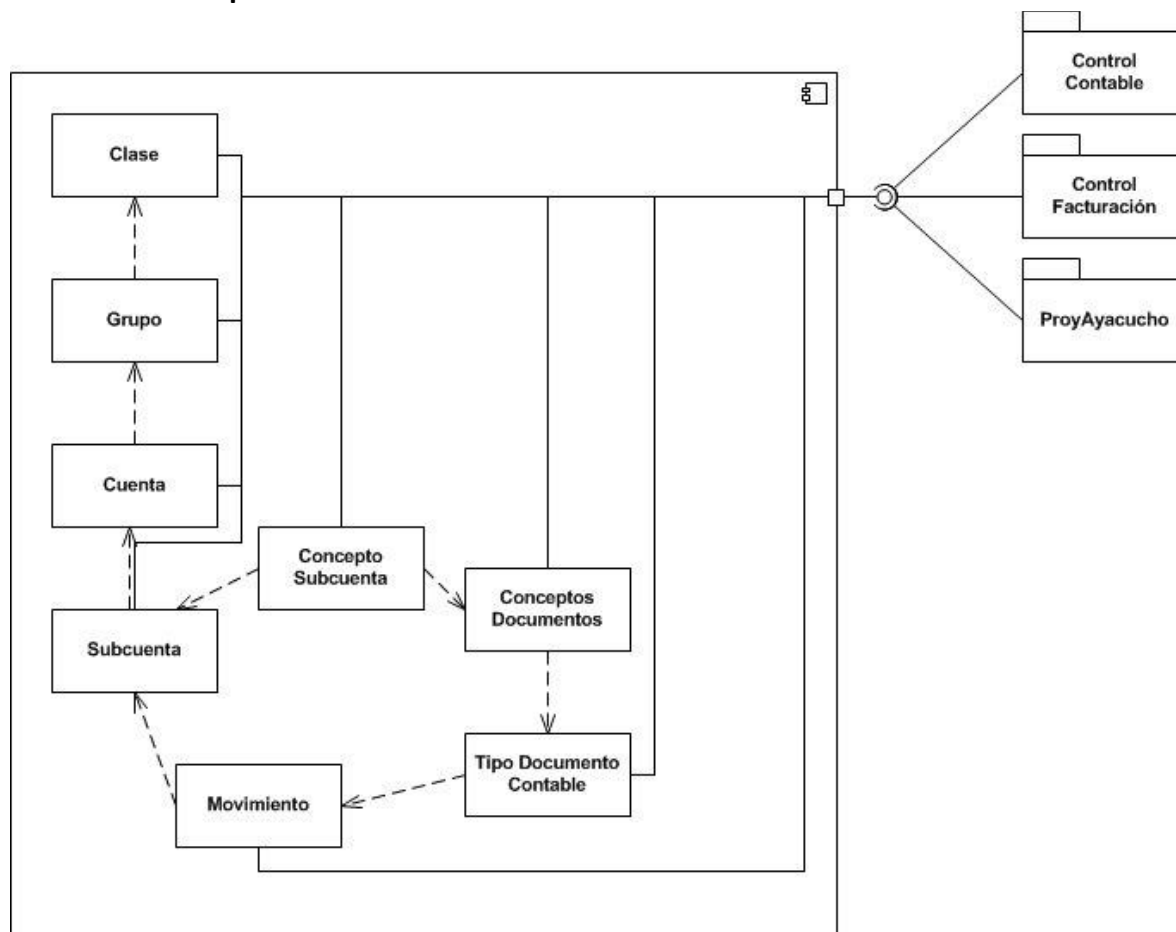
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-04 Modelo Contable	Fecha: 5/Mayo/2016

MODELO CONTABLE

1. Introducción.

Este documento busca describir que ofrece el componente y que necesita para su funcionamiento.

2. Modelo Componente.



3. Clases que conforman el Componente.

3.1. Clase.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.1.1. Variables.

codigo.

denominacion.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-04 Modelo Contable	Fecha: 5/Mayo/2016

grupoList.

3.1.2. Métodos.

Clase().
 getCodigo().
 setCodigo().
 getDenominacion().
 setDenominacion().
 getGrupoList().
 setGrupoList().

```
int getCodigo()
void setCodigo(int codigo)
String getDenominacion()
void setDenominacion(String denominacion)
List<Grupo> getGrupoList()
void setGrupoList(List<Grupo> grupoList)
```

3.1.3. Componentes Externos.

PersistenciaAPI.

3.2. ConceptoDocumento.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.2.1. Variables.

descripcion.
 conceptoSubcuentaList.
 tipoDocumentoContableId.

3.2.2. Metodos.

```
String getDescripcion()
void setDescripcion(String descripcion)
List<ConceptoSubcuenta> getConceptoSubcuentaList()
void setConceptoSubcuentaList(List<ConceptoSubcuenta> conceptoSubcuentaList)
TipoDocumentoContable getTipoDocumentoContableId()
void setTipoDocumentoContableId(TipoDocumentoContable tipoDocumentoContableId)
```

3.2.3. Componentes Externos.

PersistenciaAPI.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-04 Modelo Contable	Fecha: 5/Mayo/2016

3.3. ConceptoSubcuenta.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.3.1. Variables.

tipo.
conceptosDocumentold.
subCuentaId.

3.3.2. Metodos.

String getTipo()
void setTipo(String tipo)
ConceptosDocumento getConceptosDocumentold()
void setConceptosDocumentold(ConceptosDocumento conceptosDocumentold)
Subcuenta getSubCuentaId()
void setSubCuentaId(Subcuenta subCuentaId)

3.3.3. Componentes Externos.

PersistenciaAPI.

3.4. Cuenta.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.4.1. Variables.

codigo.
denominacion.
subcuentaList.
idGrupo.

3.4.2. Métodos.

int getCodigo()
void setCodigo(int codigo)
String getDenominacion()
void setDenominacion(String denominacion)
List<Subcuenta> getSubcuentaList()
void setSubcuentaList(List<Subcuenta> subcuentaList)
Grupo getIdGrupo()
void setIdGrupo(Gruo idGrupo)

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-04 Modelo Contable	Fecha: 5/Mayo/2016

3.4.3. Componentes externos.

PersistenciaAPI.

3.5. Grupo.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.5.1. Variables.

codigo.
denominacion.
cuentaList.
idClase.

3.5.2. Métodos.

```
int getCodigo()
void setCodigo(int codigo)
String getDenominacion()
void setDenominacion(String denominacion)
List<Cuenta> getCuentaList()
void setCuentaList(List<Cuenta> cuentaList)
Clase getIdClase()
void setIdClase(Clase idClase)
```

3.5.3. Componentes Externos.

PersistenciaAPI.

3.6. Movimiento.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.6.1. Variables.

fecha.
identificacionUsuario.
codigoValidacion.
tipoidentificacion.
tipoUsuario.
desripcion.
debito.
credito.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-04 Modelo Contable	Fecha: 5/Mayo/2016

subCuentaId.
tipoDocumentoContableId.

3.6.2. Métodos.

```
Date getFecha()
void setFecha(Date fecha)
int getIdentificacionUsuario()
void setIdentificacionUsuario(int identificacionUsuario)
int getCodigoValidacion()
void setCodigoValidacion(int codigoValidacion)
String getTipoidentificacion()
void setTipoidentificacion(String tipoidentificacion)
String getTipoUsuario()
void setTipoUsuario(String tipoUsuario)
String getDescripcion()
void setDescripcion(String descripcion)
long getDebito()
void setDebito(int debito)
long getCredito()
void setCredito(int credito)
Subcuenta getSubCuentaId()
void setSubCuentaId(Subcuenta subCuentaId)
TipoDocumentoContable getTipoDocumentoContableId()
void setTipoDocumentoContableId(TipoDocumentoContable tipoDocumentoContableId)
```

3.6.3. Componentes Externos.

PersistenciaAPI.

3.7. Subcuenta.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.7.1. Variables.

codigo.
denominacion.
idCuenta.
movimientosList.

3.7.2. Métodos.

```
Integer getCodigo()
void setCodigo(Integer codigo)
```

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-04 Modelo Contable	Fecha: 5/Mayo/2016

```

String getDenominacion()
void setDenominacion(String denominacion)
Cuenta getIdCuenta()
void setIdCuenta(Cuenta idCuenta)
List<Movimientos> getMovimientosList()
void setMovimientosList(List<Movimientos> movimientosList)

```

3.7.3. Componentes Externos.

PersistenciaAPI.

3.8. TipoDocumentoContable.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.8.1. Variables.

```

descripcion.
movimientosList.

```

3.8.2. Métodos.

```

String getDescripcion()
void setDescripcion(String descripcion)
List<Movimientos> getMovimientosList()
void setMovimientosList(List<Movimientos> movimientosList)

```

3.8.3. Componentes Externos.

PersistenciaAPI.

**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

DC-05 Modelo Facturación

**Versión 1.0
05/05/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-05 Modelo Facturación	Fecha: 5/Mayo/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
5/Mayo/2016	1.0	Propuesta inicial del documento	Hernán Darío González

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-05 Modelo Facturación	Fecha: 5/Mayo/2016

Tabla de contenido

1. Introducción.....	1
2. Modelo Componente.....	1
3. Clases que conforman el Componente.....	1
3.1. Detalle.....	1
3.1.1. Variables.....	1
3.1.2. Métodos.....	1
3.1.3. Componentes Externos.....	2
3.2. Factura.....	2
3.2.1. Variables.....	2
3.2.2. Metodos.....	2
3.2.3. Componentes Externos.....	2
3.3. FechaRegistro.....	2
3.3.1. Variables.....	2
3.3.2. Metodos.....	2
3.3.3. Componentes Externos.....	2

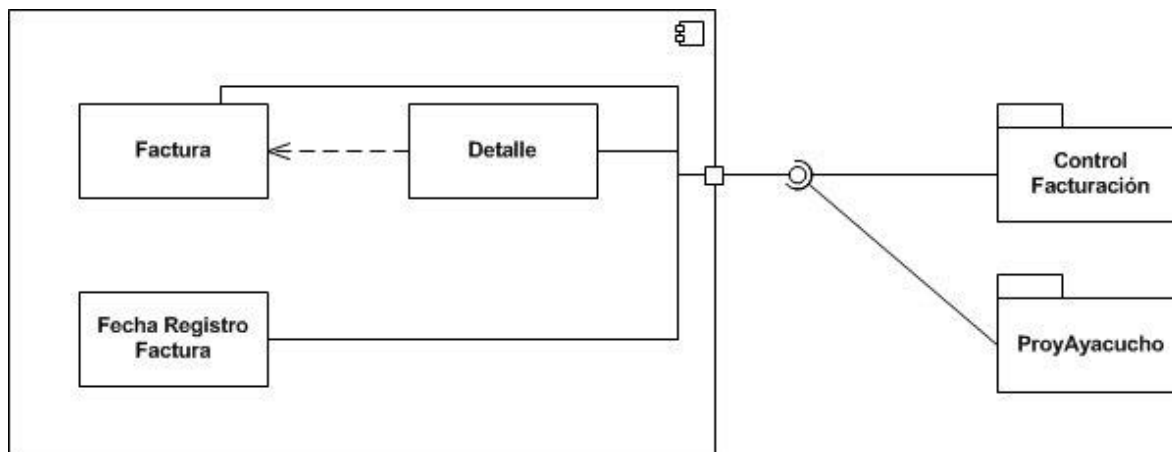
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-05 Modelo Facturación	Fecha: 5/Mayo/2016

MODELO FACTURACIÓN

1. Introducción.

Este documento busca describir que ofrece el componente y que necesita para su funcionamiento.

2. Modelo Componente.



3. Clases que conforman el Componente.

3.1. Detalle.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.1.1. Variables.

descripcionConcepto.
valorTarifa.
facturald.

3.1.2. Métodos.

```

String getDescripcionConcepto()
void setDescripcionConcepto(String descripcionConcepto)
String getValorTarifa()
void setValorTarifa(String valorTarifa)
Factura getFacturald()
void setFacturald(Factura facturald)
  
```

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-05 Modelo Facturación	Fecha: 5/Mayo/2016

3.1.3. Componentes Externos.

PersistenciaAPI.

3.2. Factura.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.2.1. Variables.

fecha.
idAcometida.
total.
estado.

3.2.2. Metodos.

Date getFecha()
void setFecha(Date fecha)
int getIdAcometida()
void setIdAcometida(int idAcometida)
int getTotal()
void setTotal(int total)
String getEstado()
void setEstado(String estado)

3.2.3. Componentes Externos.

PersistenciaAPI

3.3. FechaRegistro.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.3.1. Variables.

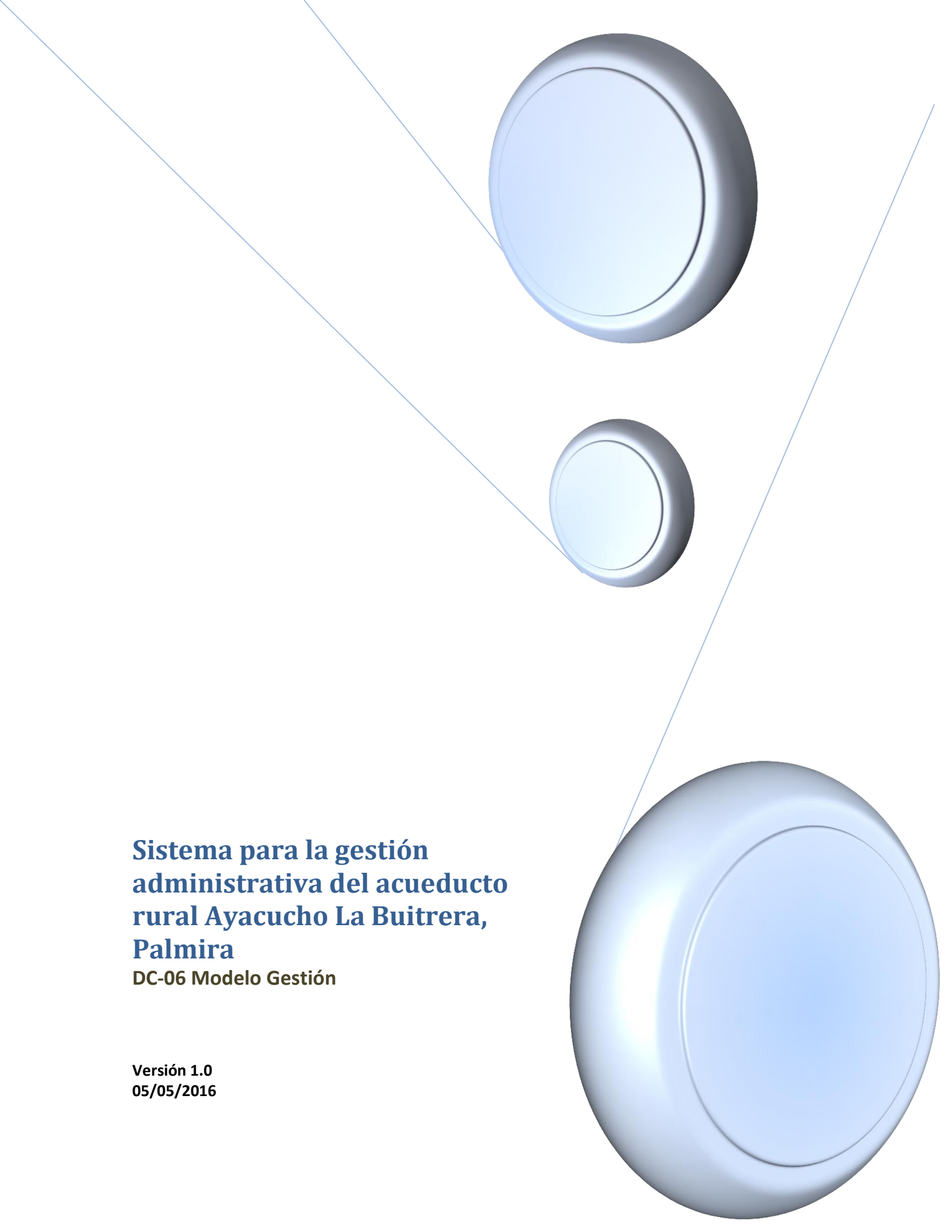
fechaRegistro.

3.3.2. Metodos.

Date getFechaRegistro()
setFechaRegistro(Date fechaRegistro)

3.3.3. Componentes Externos.

PersistenciaAPI.



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira

DC-06 Modelo Gestión

**Versión 1.0
05/05/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
5/Mayo/2016	1.0	Propuesta inicial del documento	Hernán Darío González

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

Tabla de contenido

1. Introducción.....	1
2. Modelo del componente.....	1
3. Clases que conforman el Componente.....	1
3.1. Acometida.....	1
3.1.1. Variables.....	1
3.1.2. Métodos.....	2
3.1.3. Componentes Externos.....	2
3.2. Concepto.....	3
3.2.1. Variables.....	3
3.2.2. Métodos.....	3
3.2.3. Componentes Externos.....	3
3.3. ConceptoAcometidaTarifa.....	3
3.3.1. Variables.....	3
3.3.2. Métodos.....	3
3.3.3. Componentes Externos.....	4
3.4. ReferenciaUsuario.....	4
3.4.1. Variables.....	4
3.4.2. Métodos.....	4
3.5. Suscriptor.....	4
3.5.1. Variables.....	4
3.5.2. Métodos.....	5
3.5.3. Componentes Externos.....	5
3.6. Tarifa.....	5
3.6.1. Variables.....	5
3.6.2. Métodos.....	5
3.6.3. Componentes Requeridos.....	6
3.7. TipoConcepto.....	6
3.7.1. Variables.....	6
3.7.2. Métodos.....	6

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

3.7.3. Componentes Externos.	6
3.8. TipoDocumento.....	6
3.8.1. Variables.....	6
3.8.2. Métodos	6
3.8.3. Componentes Externos.	6
3.9. TipoTarifa.	6
3.9.1. Variables	6
3.9.2. Métodos	7
3.9.3. Componentes Externos	7
3.10. TipoUsuario	7
3.10.1. Variables	7
3.10.2. Métodos	7
3.10.3. Componentes Externos	7
3.11. Zona.....	7
3.11.1. Variables	7
3.11.2. Métodos	8
3.11.3. Componentes Externos	8

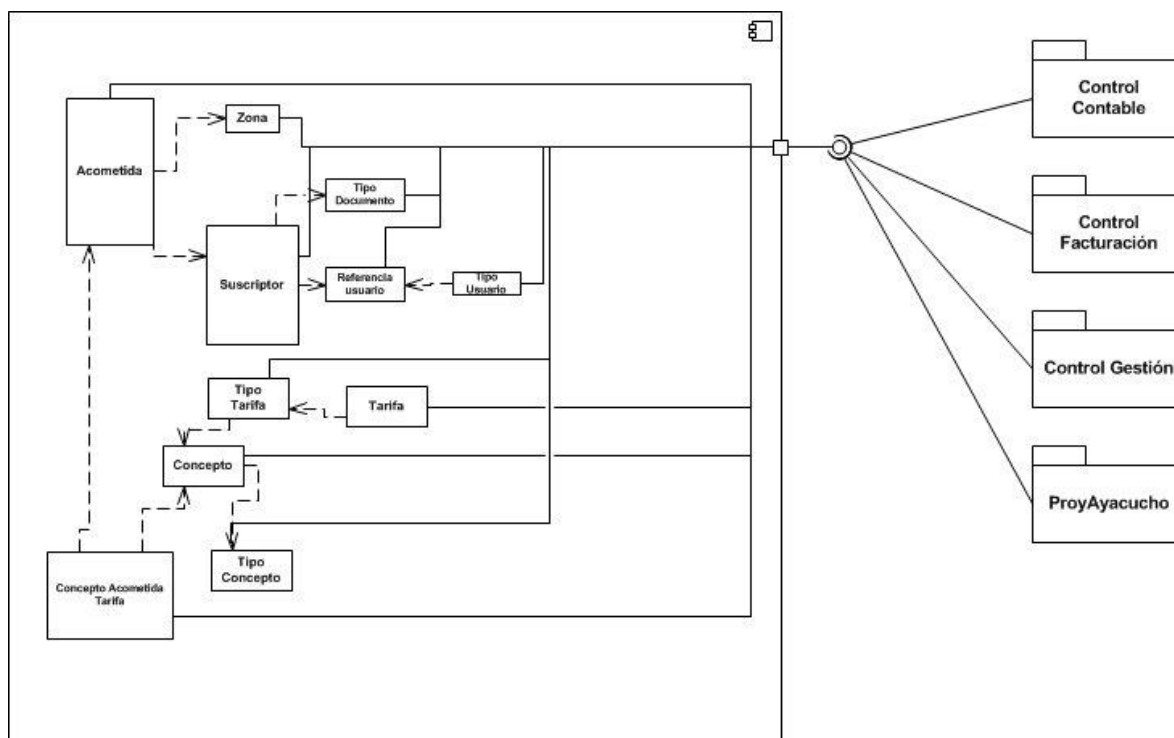
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

CONTENIDO

1. Introducción.

Este documento busca describir que ofrece el componente y que necesita para su funcionamiento.

2. Modelo del componente.



3. Clases que conforman el Componente.

3.1. Acometida.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.1.1. Variables.

direccionInstalado.
posicionPrint.
numeroCatastral.
barrioVeredaInstalado.
direccionAviso.
barrioVeredaAviso.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

tipoPredio.
 uso.
 saldoAcometida.
 saldoDistribuir.
 conceptoAcometidaTarifaList.
 suscriptorId.
 zonaIdInstalado.
 zonaIdAviso.

3.1.2. Métodos

```

Integer getSaldoAcometida()
void setSaldoAcometida(Integer saldoAcometida)
Integer getSaldoDistribuir()
void setSaldoDistribuir(Integer saldoDistribuir)
Integer getPosicionPrint()
void setPosicionPrint(Integer posicionPrint)
List<ConceptoAcometidaTarifa> getConceptoAcometidaTarifaList()
Void setConceptoAcometidaTarifaList(List<ConceptoAcometidaTarifa> conceptoAcometidaTarifaList)
Suscriptor getSuscriptorIdSuscriptor()
void setSuscriptorIdSuscriptor(Suscriptor suscriptorIdSuscriptor)
Zona getZonaIdZonaInstalado()
void setZonaIdZonaInstalado(Zona zonaIdZonaInstalado)
Zona getZonaIdZonaAviso()
void setZonaIdZonaAviso(Zona zonaIdZonaAviso)
void addConceptoTarifa(ConceptoAcometidaTarifa conceptoAcometidaTarifa)
String getNumeroCatastral()
void setNumeroCatastral(String numeroCatastral)
String getBarrioVeredaInstalado()
void setBarrioVeredaInstalado(String barrioVeredaInstalado)
String getDireccionAviso()
void setDireccionAviso(String direccionAviso)
String getBarrioVeredaAviso()
void setBarrioVeredaAviso(String barrioVeredaAviso)
String getTipoPredio()
void setTipoPredio(String tipoPredio)
String getUso()
void setUso(String uso)
String getDireccionInstalado()
void setDireccionInstalado(String direccionInstalado)
  
```

3.1.3. Componentes Externos.

PersistenciaAPI.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

3.2. Concepto.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.2.1. Variables.

descripcionConcepto.
estadoConcepto.
conceptoAcometidaTarifaList.
tipoConceptoId.
tipoTarifaList.

3.2.2. Métodos

```
String getDescripcionConcepto()
void setDescripcionConcepto(String descripcionConcepto)
String getEstadoConcepto()
void setEstadoConcepto(String estadoConcepto)
List<ConceptoAcometidaTarifa> getConceptoAcometidaTarifaList()
void setConceptoAcometidaTarifaList(List<ConceptoAcometidaTarifa> conceptoAcometidaTarifaList)
TipoConcepto getTipoConcepto()
void setTipoConcepto(TipoConcepto tipoConcepto)
List<TipoTarifa> getTipoTarifaList()
void setTipoTarifaList(List<TipoTarifa> tipoTarifaList)
```

3.2.3. Componentes Externos.

PersistenciaAPI

3.3. ConceptoAcometidaTarifa.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.3.1. Variables.

estado.
acometidaId.
conceptoId.
tarifaId.

3.3.2. Métodos

```
String getEstado()
void setEstado(String estado)
Acometida getAcometidaIdAcometida()
```

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

```

void setAcometidaIdAcometida(Acometida acometidaIdAcometida)
Concepto getConceptoIdConcepto()
void setConceptoIdConcepto(Concepto conceptoIdConcepto)
Tarifa getTarifaIdTarifa()
void setTarifaIdTarifa(Tarifa tarifaIdTarifa)

```

3.3.3. Componentes Externos.

PersistenciaAPI.

3.4. ReferenciaUsuario.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.4.1. Variables

```

suscriptorId.
tipoUsuarioId.

```

3.4.2. Métodos

```

Suscriptor getSuscriptorId()
void setSuscriptorId(Suscriptor suscriptorId)
TipoUsuario getTipoUsuarioId()
void setTipoUsuarioId(TipoUsuario tipoUsuarioId)

```

3.5. Suscriptor.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.5.1. Variables

```

identificacion.
nombre.
apellido.
teléfono.
celular.
correo.
acometidaList.
referenciaUsuarioList.
tipoDocumentoId.

```

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

3.5.2. Métodos

```

Integer getIdentificacion()
void setIdentificacion(Integer identificacion)
String getNombre()
void setNombre(String nombre)
String getApellido()
void setApellido(String apellido)
Integer getTelefono()
void setTelefono(Integer telefono)
Long getCelular()
void setCelular(Long celular)
String getCorreo()
void setCorreo(String correo)
List<ReferenciaUsuario> getReferenciaUsuarioList()
void setReferenciaUsuarioList(List<ReferenciaUsuario> referenciaUsuarioList)
List<Acometida> getAcometidaList()
void setAcometidaList(List<Acometida> acometidaList)
TipoDocumento getTipoDocumentoId()
void setTipoDocumentoId(TipoDocumento tipoDocumentoId)

```

3.5.3. Componentes Externos.

PersistenciaAPI.

3.6. Tarifa.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.6.1. Variables

valor.
TipoTarifaId.
ConceptoAcometidaTarifaList.

3.6.2. Métodos

```

Integer getValor()
void setValor(Integer valor)
TipoTarifa getIdTipoTarifa()
void setIdTipoTarifa(TipoTarifa idTipoTarifa)
List<ConceptoAcometidaTarifa> getConceptoAcometidaTarifaList()
void setConceptoAcometidaTarifaList(List<ConceptoAcometidaTarifa> conceptoAcometidaTarifaList)

```

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

3.6.3. Componentes Requeridos.

PersistenciaAPI.

3.7. TipoConcepto.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.7.1. Variables.

descripcion.
conceptoList.

3.7.2. Métodos

```
String getDescripcion()
void setDescripcion(String descripcion)
List<Concepto> getConceptoList()
void setConceptoList(List<Concepto> conceptoList)
```

3.7.3. Componentes Externos.

PersistenciaAPI.

3.8. TipoDocumento.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.8.1. Variables.

Tipo.

3.8.2. Métodos

```
String getTipo()
setTipo(String tipo)
```

3.8.3. Componentes Externos.

PersistenciaAPI.

3.9. TipoTarifa.

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.9.1. Variables

descripcionTipoTarifa.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

tarifaList.
conceptoId.

3.9.2. Métodos

String getDescripcionTipoTarifa()
void setDescripcionTipoTarifa(String descripcionTipoTarifa)
List<Tarifa> getTarifaList()
void setTarifaList(List<Tarifa> tarifaList)
Concepto getIdConcepto()
void setIdConcepto(Concepto idConcepto)

3.9.3. Componentes Externos

PersistenciaAPI.

3.10. TipoUsuario

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.10.1. Variables

tipo.
referenciaUsuarioList.

3.10.2. Métodos

String getTipo()
void setTipo(String tipo)
List<ReferenciaUsuario> getReferenciaUsuarioList()
void setReferenciaUsuarioList(List<ReferenciaUsuario> referenciaUsuarioList)

3.10.3. Componentes Externos

PersistenciaAPI.

3.11. Zona

En esta estructura se guardara la información que conforma la misma entidad de en la base de datos.

3.11.1. Variables

descripcion.
acometidaList.
acometidaList1.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-06 Modelo Gestión	Fecha: 5/Mayo/2016

3.11.2. Métodos

```
String getDescripcion()
void setDescripcion(String descripcion)
List<Acometida> getAcometidaList()
void setAcometidaList(List<Acometida> acometidaList)
List<Acometida> getAcometidaList1()
void setAcometidaList1(List<Acometida> acometidaList1)
```

3.11.3. Componentes Externos

PersistenciaAPI.



Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira

DC-07 PersistenciaApi

**Versión 1.0
05/05/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-07 PersistenciaApi	Fecha: 5/Mayo/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
5/Mayo/2016	1.0	Propuesta inicial del documento	Hernán Darío González

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-07 PersistenciaApi	Fecha: 5/Mayo/2016

Tabla de contenido

1. Introducción	1
2. Clases que conforman el Componente	1
2.1. AbstractEntity	1
2.1.1. Variables	1
2.1.2. Métodos	1
2.2. AbstractFacade	1
2.2.1. Variables	1
2.2.2. Métodos	1

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-07 PersistenciaApi	Fecha: 5/Mayo/2016

PersistenciaApi

1. Introducción

Este documento busca describir que ofrece el componente y que necesita para su funcionamiento.

2. Clases que conforman el Componente

2.1. AbstractEntity

En esta clase se encontraran lo métodos abstractos de las entidades.

2.1.1. Variables

primaryKey.

2.1.2. Métodos

AbstractEntity().
AbstractEntity().
setPrimaryKey().
hashCode().
equals(Object obj).
toString().

2.2. AbstractFacade

En esta clase se encontraran lo métodos abstractos de los facade.

2.2.1. Variables

entityClass.

2.2.2. Métodos

AbstractFacade().
EntityManager getEntityManager();
edit().
remove().
find().
findAll().
findRange().
count().
save().

**Sistema para la gestión
administrativa del acueducto
rural Ayacucho La Buitrera,
Palmira**

DC-08 ProyAyacucho

**Versión 1.0
05/05/2016**

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

HISTORIAL DE REVISIONES

Fecha	Versión	Descripción	Autor
5/Mayo/2016	1.0	Propuesta inicial del documento	Hernán Darío González

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

Tabla de contenido

1. Introducción.....	1
2. Modelo del componente.....	1
3. Clases que conforman el Componente.....	1
3.1. AcometidaVista.	1
3.1.1. Variables.....	1
3.1.2. Métodos.	2
3.1.3. Componentes Externos.	3
3.2. ConceptoVista.	3
3.2.1. Variables.....	3
3.2.2. Metodos.	4
3.2.3. Componentes Externos.	4
3.3. SuscriptorVista.	4
3.3.1. Variables.....	4
3.3.2. Metodos.	5
3.3.3. Componentes Externos.	6
3.4. TarifaCompletoVista.....	6
3.4.1. Variables.....	6
3.4.2. Metodos.	6
3.4.3. Componentes Externos.	7
3.5. TipoConceptoVista.	7
3.5.1. Variables.....	7
3.5.2. Metodos.	7
3.5.3. Componentes Externos.	7
3.6. ZonaVista.....	8
3.6.1. Variables.....	8
3.6.2. Metodos.	8
3.6.3. Componentes Externos.	8
3.7. FacturaVista.....	8
3.7.1. Variables.....	8

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

3.7.2.	Metodos.	9
3.7.3.	Componentes Externos.	9
3.8.	OrdenImpresionVista.	10
3.8.1.	Variables.	10
3.8.2.	Metodos.	10
3.8.3.	Componentes Externos.	10
3.9.	PagarFacturasVista.	11
3.9.1.	Variables.	11
3.9.2.	Metodos.	11
3.9.3.	Componentes Externos.	12

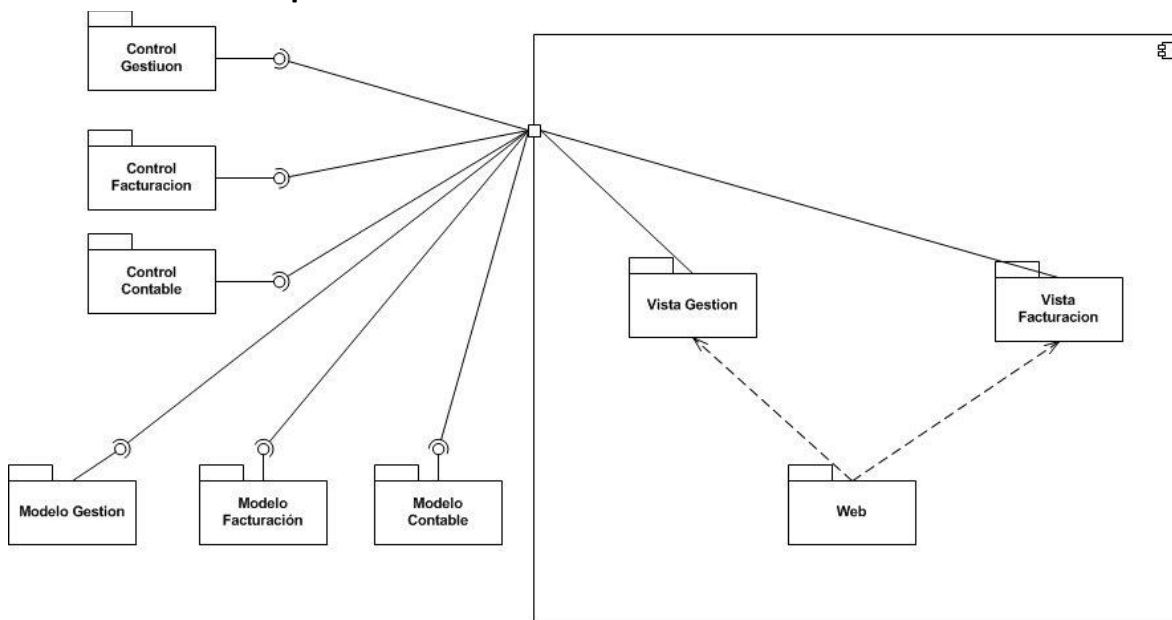
Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

ProyAyacucho

1. Introducción.

Este documento busca describir que ofrece el componente y que necesita para su funcionamiento.

2. Modelo del componente



3. Clases que conforman el Componente.

3.1. AcometidaVista.

Esta clase se encarga de recibir los datos de la interface del operador para ser procesados a través de los componentes de control según la necesidad de la acción ejecutada.

3.1.1. Variables.

listaAcometida.
 listaAcometidaFiltro.
 selectedAcometida.
 strsup.
 comboZonaInstalada.
 itemsZonaInstalada.
 comboZonaAviso.
 itemsZonaAviso.
 listaZonas.
 txtSuscriptor.
 comboTipoConcepto.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

itemsTipoConcepto.
 listaTipoConceptos.
 comboConcepto.
 itemsConcepto.
 listaConceptos.
 comboTarifa.
 itemsTarifa.
 listaTarifas.
 tarifaSelected.
 envioFactura.
 acometida.
 acometidaLogica.
 zonaLogica.
 suscriptorLogica.
 tipoConceptoLogica.
 conceptoLogica.
 tarifaLogica.

3.1.2. Métodos.

AcometidaVista().
 getStrsup().
 setStrsup().
 getListAcometida().
 setListaAcometida().
 getListAcometidaFiltro().
 setListaAcometidaFiltro().
 getSelectedAcometida().
 setSelectedAcometida().
 getAcometida().
 setAcometida().
 getComboZonaInstalada().
 setComboZonaInstalada().
 getItemsZonaInstalada().
 setItemsZonaInstalada().
 getComboZonaAviso().
 setComboZonaAviso().
 getItemsZonaAviso().
 setItemsZonaAviso().
 getTxtSuscriptor().
 setTxtSuscriptor().

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

```

getComboTipoConcepto().
setComboTipoConcepto().
getItemsTipoConcepto().
setItemsTipoConcepto().
getComboConcepto().
setComboConcepto().
getItemsConcepto().
setItemsConcepto().
getComboTarifa().
setComboTarifa().
getItemsTarifa().
setItemsTarifa().
getTarifaSelected().
setTarifaSelected().
buscarSuscriptor().
onRowSelect().
getEnvioFactura().
limpiar().
registrar().
modificar().
agregarConceptoTarifa().
onRowSelectServicios().
quitarConceptoTarifa().

```

3.1.3. Componentes Externos.

```

PersistenciaAPI.
Modelo Gestion.
ControlGestion.
PrimaFaces.

```

3.2. ConceptoVista.

Esta clase se encarga de recibir los datos de la interface del operador para ser procesados a través de los componentes de control según la necesidad de la acción ejecutada.

3.2.1. Variables.

```

cmb_tc.
items_tc.
array_tc.
listaConceptos.
listaConceptosFiltro.
estadoVista.

```

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

selectedConcepto.
cpto.
tipo_concepto_logica.
cpto_logica.

3.2.2. Metodos.

ConceptoVista().
getCpto().
setCpto().
getCmb_tc().
setCmb_tc().
getListaConceptosFiltro().
setListaConceptosFiltro().
getItems_tc().
setItems_tc().
isEstadoVista().
setEstadoVista().
getListaConceptos().
setListaConceptos().
getSelectedConcepto().
setSelectedConcepto().
onRowSelect().
limpiar().
fixSelectTipoConcepto().
registrar().
modificar().
eliminar().

3.2.3. Componentes Externos.

PersistenciaAPI.
LogicaGestion.
ModeloGestion.
Primefaces.

3.3. SuscriptorVista.

Esta clase se encarga de recibir los datos de la interface del operador para ser procesados a través de los componentes de control según la necesidad de la acción ejecutada.

3.3.1. Variables.

listaSuscriptor.
listaSuscriptorFiltro.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

selectedSuscriptor.
 listaTipoDocumento.
 suscriptor.
 tipoDocumento.
 idTipoDocumento.
 tipoUsuario.
 listaTipoUsuario.
 condicion.
 auxList.
 radioTipoDocumento.
 suscriptorLogica.
 referenciaSuscriptorLogica.

3.3.2. Metodos.

SuscriptorVista().
 getSuscriptor().
 setSuscriptor().
 getTipoDocumento().
 getTipoUsuario().
 setTipoUsuario().
 getAuxList().
 setAuxList().
 getRadioTipoDocumento().
 setRadioTipoDocumento().
 getListTipoUsuario().
 setListaTipoUsuario().
 setIdTipoDocumento().
 getIdTipoDocumento().
 getListSuscriptor().
 setListaSuscriptor().
 getListSuscriptorFiltro().
 setListaSuscriptorFiltro().
 getSelectedSuscriptor().
 getListTipoDocumento().
 setListaTipoDocumento().
 setSelectedSuscriptor().
 onRowSelect().
 registrar().
 modificar().
 eliminar().

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

limpiar()).

3.3.3. Componentes Externos.

PersistenciaAPI.

LogicaGestion.

ModeloGestion.

Primefaces.

3.4. TarifaCompletoVista.

Esta clase se encarga de recibir los datos de la interface del operador para ser procesados a través de los componentes de control según la necesidad de la acción ejecutada.

3.4.1. Variables.

tarifa.

ttarifa.

listaTarifa.

listaTarifaFiltro.

listaConcepto.

selected_tarifa.

comboConcepto.

itemsConcepto.

tarifaLogica.

ttarifaLogica.

conceptoLogica.

3.4.2. Metodos.

TarifaCompletoVista().

Tarifa getTarifa().

setTarifa().

getTtarifa().

setTtarifa().

getListTarifa().

setListaTarifa().

getListTarifaFiltro().

setListaTarifaFiltro().

getSelected_tarifa().

setSelected_tarifa().

getComboConcepto().

setComboConcepto().

getItemsConcepto().

setItemsConcepto().

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

onRowSelect().
 fixSelectConcepto().
 registrar().
 modificar().
 eliminar().
 limpiar().

3.4.3. Componentes Externos.

PersistenciaAPI.
 ModeloGestion.
 LogicaGestion.
 PrimeFaces.

3.5. TipoConceptoVista.

Esta clase se encarga de recibir los datos de la interface del operador para ser procesados a través de los componentes de control según la necesidad de la acción ejecutada.

3.5.1. Variables.

tipo_concepto.
 selected_tc.
 listaTipoConceptoFiltro.
 tipo_concepto_logica.

3.5.2. Metodos.

TipoConceptoVista().
 getSelected_tc().
 setSelected_tc().
 getListTipoConcepto().
 getListTipoConceptoFiltro().
 setListaTipoConceptoFiltro().
 getTipo_concepto().
 onRowSelect().
 limpiar().
 registrar().
 modificar().
 eliminar().

3.5.3. Componentes Externos.

PersistenciaAPI.
 ModeloGestion.
 LogicaGestion.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

PrimeFaces.

3.6. ZonaVista

Esta clase se encarga de recibir los datos de la interface del operador para ser procesados a través de los componentes de control según la necesidad de la acción ejecutada.

3.6.1. Variables.

listaZonasFiltro.
zona.
selectedZona.
zona_logica.

3.6.2. Metodos.

ZonaVista().
getListaZonas().
getListaZonasFiltro().
setListaZonasFiltro().
getZona().
setZona().
getSelectedZona().
setSelectedZona().
onRowSelect().
limpiar().
registrar().
modificar().
eliminar().

3.6.3. Componentes Externos.

PersistenciaAPI.
ModeloContable.

3.7. FacturaVista.

Esta clase se encarga de recibir los datos de la interface del operador para ser procesados a través de los componentes de control según la necesidad de la acción ejecutada.

3.7.1. Variables.

listaFacturas.
listaFacturasFiltro.
selectedFactura.
archivoFacturas.
controlFacturacion.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

controlDescarga.
facturaLogica.
catLogica.
acometidaLogica.
regfecLogica.

3.7.2. Metodos.

FacturaVista().
getListaFacturas().
setListaFacturas().
getListaFacturasFiltro().
setListaFacturasFiltro().
getSelectedFactura().
setSelectedFactura().
getArchivoFacturas().
setArchivoFacturas().
getControlFacturacion().
setControlFacturacion().
getControlDescarga().
setControlDescarga().
onRowSelect().
onRowReorder().
fixAcometida().
registrar().
obtenerUltimaFecha().
imprimir().
descargarArchivo().
manipulatePdf().
celdaBold().
celdaBoldCenter().
getWatermarkedImage().
imprimir2().
imprimir3().

3.7.3. Componentes Externos.

PersistenciaAPI.
LogicaGestion.
LogicaFacturacion.
ModeloGestion.
ModeloFacturacion.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

Primefaces.

ltext.

3.8. OrdenImpresionVista.

Esta clase se encarga de recibir los datos de la interface del operador para ser procesados a través de los componentes de control según la necesidad de la acción ejecutada.

3.8.1. Variables.

comboZona.
itemsZona.
array_Zona.
listaAcometidas.
selectedAcometida.
posZona.
zonaLogica.
acometidaLogica.

3.8.2. Metodos.

OrdenImpresionVista().
getComboZona().
setComboZona().
getItemsZona().
setItemsZona().
getArray_Zona().
setArray_Zona().
getListAcometidas().
setListaAcometidas().
getSelectedAcometida().
setSelectedAcometida().
getPosZona().
setPosZona().
onRowSelect().
onRowReorder().
cargar_lista().
confirmar().
cancelar().

3.8.3. Componentes Externos.

PersistenciaAPI.
LogicaGestion.
ModeloGestion.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

Primefaces.

3.9. PagarFacturasVista.

Esta clase se encarga de recibir los datos de la interface del operador para ser procesados a través de los componentes de control según la necesidad de la acción ejecutada.

3.9.1. Variables.

pago.
 pagoOtros.
 idAcometida.
 selectedAcometida.
 listaAcometidas.
 radioTipoPago.
 idTipoPago.
 descripcion.
 txtSuscriptor.
 strsup.
 suscriptor .
 comboTipoUsuario.
 itemsTipoUsuario.
 comboSubcuentaDebito.
 comboSubcuentaCredito.
 itemsSubcuenta.
 listaMoviminetos.
 tipoSuscriptor.
 acometidaLogica.
 facturaLogica.
 movimientoLogica.
 suscriptorLogica.
 subCuentaLogica.
 tiDoCoLogica.

3.9.2. Metodos.

PagarFacturasVista().
 getPago().
 setPago().
 getPagoOtros().
 setPagoOtros().
 getIdAcometida().
 setIdAcometida().
 getSelectedAcometida().

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

setSelectedAcometida().
 getRadioTipoPago().
 setRadioTipoPago().
 getIdTipoPago().
 setIdTipoPago().
 getTxtSuscriptor().
 setTxtSuscriptor().
 getStrsup().
 setStrsup().
 getDescripcion().
 setDescripcion().
 getComboTipoUsuario().
 setComboTipoUsuario().
 getItemsTipoUsuario().
 setItemsTipoUsuario().
 getComboSubcuentaDebito().
 setComboSubcuentaDebito().
 getComboSubcuentaCredito().
 setComboSubcuentaCredito().
 getItemsSubcuenta().
 setItemsSubcuenta().
 getListMoviminetos().
 setListaMoviminetos().
 getSuscriptor().
 setSuscriptor().
 getTipoSuscriptor().
 setTipoSuscriptor().
 getListAcometidas().
 setListaAcometidas().
 onRowSelect().
 buscarSuscriptor().
 generarPago().
 adicionarMovimiento().
 registrarMovimientos().
 limpiar().
 limpiarMovimiento().

3.9.3. Componentes Externos.

PersistenciaAPI.
 LogicaGestion.

Sistema para la gestión administrativa del acueducto rural Ayacucho La Buitrera, Palmira	Versión 1.0
DC-08 ProyAyacucho	Fecha: 5/Mayo/2016

LogicaFacturacion.

LogicaContable.

ModeloContable.

ModeloGestion.

Primefaces.