

UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingenieros de Sistemas**

**TEMA:
ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE PARA LA
GESTIÓN ADMINISTRATIVA DE LAS JUNTAS COMUNITARIAS DE AGUA
DE CONSUMO HUMANO EN LAS PROVINCIAS DE PICHINCHA E
IMBABURA**

**AUTORES:
RICHARD DAMIÁN CARRIÓN JIMÉNEZ
EDGAR STALIN LADINO PINTO**

**TUTOR:
LLERENA PAZ ROBINSON DIMITRI**

Quito, agosto del 2017

CESIÓN DE DERECHOS DE AUTOR

Nosotros CARRIÓN JIMENEZ RICHARD DAMIÁN y, LADINO PINTO EDGAR STALIN con documento de identificación N° 1718401928 y 1721117917 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE PARA LA GESTIÓN ADMINISTRATIVA DE LAS JUNTAS COMUNITARIAS DE AGUA DE CONSUMO HUMANO EN LAS PROVINCIAS DE PICHINCHA E IMBABURA”, mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, agosto del 2017



Richard Damián Carrión Jiménez

CC: 1718401928



Edgar Stalin Ladino Pinto

CC: 1721117917

DECLARATORIA DE COAUTORIA DEL DOCENTE TUTOR/A

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación, “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE PARA LA GESTIÓN ADMINISTRATIVA DE LAS JUNTAS COMUNITARIAS DE AGUA DE CONSUMO HUMANO EN LAS PROVINCIAS DE PICHINCHA E IMBABURA”, realizado por los estudiantes CARRIÓN JIMENEZ RICHARD DAMIÁN con Cl. 1718401928 y LADINO PINTO EDGAR STALIN con Cl.1721117917, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto del 2017



Llerena Paz Robinson Dimitri
C.C. 1707106942.

Dedicatoria

Dedico este trabajo principalmente a Dios, por haberme permitido culminar mi formación profesional.

A mis padres y hermanos por ser el pilar más importante en mi vida y por demostrarme siempre su cariño y apoyo incondicional.

A los docentes, quienes nunca desistieron en transmitirnos sus conocimientos hasta llegar a la culminación de nuestra etapa universitaria.

Edgar Stalin Ladino Pinto

En primer lugar quiero dedicar este trabajo a Dios por permitirme cumplir uno de mis objetivos tan anhelados, por siempre haber guiado mi camino y haberme brindado su sabiduría, en segundo lugar a mis padres, quienes han inculcado los mejores valores, me han brindado su amor, sus consejos y quienes nunca dejaron de creer en mí y siempre me han apoyado a lo largo de mi vida. Gracias a mis hermanos con quienes he compartido los mejores momentos de mi vida y quienes son un motivo de inspiración para cada día salir adelante. A familiares, amigos y conocidos que me han brindado su ayuda en todo momento y que han sido partícipes de este logro. A esta excelente institución y sus docentes por sus esfuerzos para que finalmente pueda ser un buen profesional.

¡Muchas Gracias!

Richard Damián Carrión Jiménez

Agradecimiento

Un especial agradecimiento a la Universidad Politécnica Salesiana, por habernos permitido formar parte de tan prestigiosa institución, a los docentes, partícipes fundamentales de nuestra formación académica que con esmero supieron transmitirnos conocimientos y valores para el camino profesional que nos espera.

Gracias a nuestro tutor, el Ingeniero Robinson Llerena, por la orientación y el apoyo brindado durante la realización de este trabajo, así como a la Unidad de Vinculación, representada por la Lic. Alexandra Moya, por su acertada gestión con la sociedad de Cayambe que permitió llevar a cabo con éxito este proyecto.

Carrión Jiménez Richard Damián & Ladino Pinto Edgar Stalin

ÍNDICE

INTRODUCCIÓN	1
Antecedentes	2
Problema	2
Justificación	4
Objetivo general	5
Objetivos específicos	5
Capítulo 1	6
Marco Teórico	6
1.1 Proceso de recaudación	6
1.1.1 Descripción del proceso.	6
1.2 Herramientas y Tecnología	7
1.2.1 Programación Orientada a Objetos.....	7
1.2.1.1 Abstracción.	8
1.2.2 Base de datos Relacionales.....	8
1.2.3 Aplicativo Web	10
1.2.4 Modelo Vista Controlador.....	11
1.2.5 Android.....	12
1.3 Metodología.....	13
1.3.1 Scrum.	13
1.3.2 Equipo Scrum.	13
1.3.3 Uso de Scrum	15
1.3.4 Uso de UML.....	18
1.3.5 Uso de la PBS.....	19
1.3.6 Uso del Prototipado.	19
Capítulo 2	20
Análisis y diseño	20
2.1 Requerimientos.....	20

2.1.1 Requerimientos Funcionales.	20
2.1.2 Requerimientos no funcionales.	20
2.2 Diagramas de Casos de Uso	21
2.2.1 Diagramas del módulo de Administración.	21
2.2.2 Diagramas del módulo de Cobro y Facturación.	35
2.2.3 Diagramas del módulo de Aplicación Android.	37
2.3 Diagrama de base de datos	40
2.4 Arquitectura del sistema	42
Capítulo 3	44
Construcción y pruebas	44
3.1 Arquitectura	44
3.2 Código Relevante	46
3.2.1 Clase: Conexión.	46
3.2.2 Módulo Login - Clase: loginControler	48
3.2.3 Módulo Usuario - Clase: usuarioController.	49
3.2.4 Módulo Parametrización - Clase: parametrizacionController.	51
3.2.5 Módulo Reportes - Clase: reportesController	54
3.3 Pruebas	55
3.3.1 Ambiente para Pruebas.	55
3.3.2 Pruebas.	55
3.3.2.1 Pruebas funcionales.	56
3.3.2.2 Pruebas de Rendimiento.	58
3.3.2.3 Pruebas de aceptación.	66
CONCLUSIONES	71
RECOMENDACIONES	73
REFERENCIAS BIBLIOGRÁFICAS	74

ÍNDICE DE FIGURAS

Figura 1.-Modelo Vista Controlador.....	11
Figura 2.-Modelo de Caso de uso, Proceso Creación de Usuario	21
Figura 3.-Modelo de Caso de uso, Proceso Actualización y eliminación de Usuarios....	23
Figura 4.-Modelo de Caso de uso, Proceso Creación de Medidores.....	25
Figura 5.- Modelo de Caso de uso, Proceso Actualización y eliminación de Medidores.	27
Figura 6.-Modelo de Caso de uso, Proceso de autenticación	29
Figura 7.-Modelo de Caso de uso, Proceso Creación de Parámetros.....	31
Figura 8.- Modelo de Caso de uso, Proceso Actualización y eliminación de Parámetros.	33
Figura 9.-Modelo de Caso de uso, Proceso de Facturación	35
Figura 10.-Modelo de Caso de uso, Proceso de autenticación Aplicación Android.....	37
Figura 11.-Modelo de Caso de uso, Proceso de ingreso de lecturas	39
Figura 12.- Versión 3.0 de la base de datos	41
Figura 13.- Diagrama inicial de la arquitectura.....	43
Figura 14.- Diagrama de la arquitectura.....	45
Figura 15.- Archivo xml – Clase Conexión	46
Figura 16.- Archivo de configuración – Datasource	47
Figura 17.- Clase: LoginControler – Módulo Loguin	48
Figura 18.- Clase: usuarioController – Módulo Usuario	49
Figura 19.- Método: Validación Login – Módulo Usuario	50
Figura 20.- Clase: parametrizaciónController – Módulo Parametrización	51
Figura 21.- Clase: parametrizaciónController – Módulo Parametrización	52
Figura 22.- Método: Consumo Total – Módulo Parametrización	53
Figura 23.- Clase: reportesCotroller – Módulo Reportes.....	54
Figura 24.-Configuración petición http, Prueba de rendimiento Jmeter	59
Figura 25.-Configuración grupo de hilos, Prueba de rendimiento Jmeter	59
Figura 26.- Resultados árbol de resultados, Módulo de autenticación, Caso 1	60
Figura 27.- Resultados gráfico de resultados, Módulo de autenticación, Caso 1	60

Figura 28.- Resultados informe agregado, Módulo de autenticación, Caso 1	61
Figura 29.- Resultados árbol de resultados, Módulo de facturación, Caso 1	61
Figura 30.- Resultados gráfico de resultados, Módulo de facturación, Caso 1	62
Figura 31.- Resultados informe agregado, Módulo de facturación, Caso 1	62
Figura 32.- Resultados árbol de resultados, Módulo de autenticación, Caso 2	63
Figura 33.- Resultados gráfico de resultados, Módulo de autenticación, Caso 2	63
Figura 34.- Resultados informe agregado, Módulo de autenticación, Caso 2	64
Figura 35.- Resultados árbol de resultados, Módulo de facturación, Caso 1	64
Figura 36.- Resultados gráfico de resultados, Módulo de facturación, Caso 2	65
Figura 37.- Resultados informe agregado, Módulo de facturación, Caso 2	65
Figura 38.- Resultados obtenidos prueba de aceptación, Pregunta 1	67
Figura 39.- Resultados obtenidos prueba de aceptación, Pregunta 2	67
Figura 40.- Resultados obtenidos prueba de aceptación, Pregunta 3	68
Figura 41.- Resultados obtenidos prueba de aceptación, Pregunta 4	69
Figura 42.- Resultados obtenidos prueba de aceptación, Pregunta 5	69
Figura 43.- Resultados generales de las preguntas de las pruebas de aceptación	70

ÍNDICE DE TABLAS

Tabla 1.- Escenario de Caso de Uso, Proceso Creación de Usuario	22
Tabla 2.- Escenario de Caso de Uso, Proceso Actualización o Eliminación de Usuario.	24
Tabla 3.- Escenario de Caso de Uso, Proceso Creación de Medidores	26
Tabla 4.- Escenario de Caso de Uso, Proceso Actualización o Eliminación de Medidores	28
Tabla 5.-Escenario de Caso de Uso, Proceso de Autenticación.	30
Tabla 6.- Escenario de Caso de Uso, Proceso Creación de Parámetros	32
Tabla 7.- Escenario de Caso de Uso, Proceso Actualización o Eliminación de Parámetros	34
Tabla 8.- Escenario de Caso de Uso, Proceso de Facturación	36
Tabla 9.-Escenario de Caso de Uso, Proceso de Autenticación Aplicación Android. ..	38
Tabla 10.-Escenario de Caso de Uso, Proceso de Ingreso de Lecturas	40
Tabla 11.-Resultados pruebas funcionales	56

Resumen

El presente proyecto consiste en el análisis, diseño e implementación de un aplicativo web y móvil que automatice el proceso de gestión administrativa de consumo de agua potable en las Juntas Comunitarias de las provincias de Pichincha e Imbabura que se manejaba por medio de una hoja de cálculo.

La aplicación se diseñó con base a los requerimientos de los usuarios pertenecientes a las regionales y comunidades. Se mantuvo reuniones periódicas con los diferentes representantes para dar a conocer la propuesta de automatización, solventar dudas y levantar la información necesaria para que el sistema creado satisfaga los requerimientos y beneficie a todos los involucrados.

En el desarrollo del sistema se tomó en cuenta, en primer lugar, el proceso de ingreso de datos para la facturación, que debía ser parametrizable debido a que cada comunidad cuenta con su propia tabla de valores de consumo por metro cúbico, para lo cual se presentó una avance de la pantalla de parametrización con el objetivo de que sea validada y se pueda generar la factura con los valores correspondientes a cada una de las tablas de valores de consumo.

El resultado final del proyecto fue un aplicativo web y móvil, que automatiza el proceso de gestión administrativa del consumo de agua potable, bajo los requerimientos levantados en el inicio del proyecto.

Abstract

This research is about analysis, design and implementation of web and mobile application that automates the process of administrative management of consumption drinkable water in the Community Committee of the provinces of Pichincha and Imbabura that was handled through a spreadsheet.

The application was designed according to users requirements belongs to regions and communities. Meetings were held with the different representatives for now the automatization proposal, solve doubts and raise the necessary information so that the system created fulfil the requirements and benefits all involved.

In the system development, the first place was data income for check in, that had to be parametrizable, due to each community have own values table for consuming per cubic metre, which it was presented an advance of the Parameterization screen in order to be validated and can be generated bill with the values corresponding to each of the tables consumption values.

As a final result was web and mobile application, that automates the administrative management of drinkable water consumption, under the requirements raised at the beginning of the project.

INTRODUCCIÓN

El presente proyecto nace después de la necesidad de dar una solución a la gestión administrativa de las Juntas Comunitarias de Pichincha e Imbabura, se tiene como antecedente que anteriormente se realizó un sistema, el cual fue una solución local y que no tuvo mayor continuidad por problemas de requerimientos y de instalación; por tal razón se realizó este proyecto con una orientación web y que sea accesible desde cualquier punto y para cualquier tipo de usuario.

El presente proyecto se encuentra dividido en 4 capítulos en los que se describe de forma clara y concisa todas las etapas que se requirió para optimizar el proceso de facturación en las Juntas Comunitarias y por consecuente gestionar el servicio público de Agua de Consumo Humano de mejor manera.

A continuación se describe de forma generalizada cada uno de los capítulos:

El capítulo uno trata temas relacionados con la estructura del proyecto: título del proyecto, planteamiento del problema, antecedentes, descripción del problema, objetivos planteados, justificación, alcance y la metodología empelada para el desarrollo del proyecto.

En el capítulo dos se aborda los procedimientos, procesos y teorías que constituyen la base del sustento teórico de este proyecto, se detalla el marco desarrollo ágil que se utilizó para la ejecución además de los diversos artefactos que permitieron la ejecución del mismo.

En el capítulo 3 se detalla los módulos y diagramas establecidos en la etapa de análisis y diseño para la elaboración del proyecto.

En el capítulo 4 se describe, finalmente, el código fuente generado a partir del desarrollo de los módulos establecidos en el alcance, todas las pruebas realizadas y los resultados obtenidos en base a las mismas, demostrando el óptimo funcionamiento del proyecto.

Antecedentes

Las Juntas Comunitarias de Pichincha e Imbabura contemplan varios proyectos multidisciplinarios entre los cuales se contemplan proyectos viales, electromecánicos, estructurales, eléctricos, sanitarios, entre otros.

Entre estos se encuentran la rehabilitación de diferentes sistemas de Agua Potable, para poder dotar de este servicio a las comunidades de los diferentes sectores. Para poder llevar a cabo estos proyectos se necesita llevar a cabo una gestión que contemple los diferentes gastos que generan estos cambios.

Entre la gestión que llevan a cabo las Juntas Comunitarias se encuentran el proceso de facturación, el mismo que lo llevan a cabo en Aplicaciones ofimáticas con hojas de cálculo como es “Excel” o de forma manual.

Bajo estos antecedentes surge la necesidad de un Sistema de Gestión Administrativa que les permita llevar un control del consumo y suministro de agua potable de cada uno de los usuarios, que permita llevar una correcta facturación, pero principalmente que permita el desarrollo de estos proyectos que benefician a todos los integrantes de las diferentes comunidades.

Problema

En los sectores rurales de las provincias de Pichincha e Imbabura, existen zonas denominados regionales; las cuales se conforman de Juntas Comunitarias. Estas tienen

una estructura organizacional, es decir, cuentan con dirigentes los cuales se encargan de mantener el bienestar de todos los miembros de su comunidad.

Los dirigentes de cada Junta Comunitaria tienen la responsabilidad de administrar el consumo de agua potable entre todos los miembros de la comunidad, la cual debe ser equitativa y justa, garantizando de esta manera, el derecho humano sobre la misma. Para llevar a cabo esta labor los dirigentes de cada junta están encargados de realizar las siguientes actividades:

Administrar la información hídrica de carácter técnico y administrativo.

Recopilar mensualmente las lecturas del consumo del agua potable de cada una de las familias a las que ha sido asignado el servicio.

Asignar medidores a cada una de las familias que han sido beneficiadas para el acceso a este servicio.

Imponer multas de acuerdo al reglamento de cada una las Juntas Comunitarias.

Realizar el proceso de facturación para el cobro de los valores correspondientes a la prestación del servicio y demás que correspondan.

Las Juntas comunitarias para la administración del agua potable, en principio, registraban el consumo, multas y demás valores por cada miembro de manera manual. La gran cantidad de valores generados provocó dificultad en la administración de los mismos, por lo cual posteriormente, surgió la necesidad de que las Juntas Comunitarias gestionen la información mediante herramientas tecnológicas y por disponibilidad, se usa actualmente la aplicación de ofimática “Excel” que permitió resolver temporalmente el inconveniente de accesibilidad a los datos, sin embargo, dicha herramienta no brinda

facilidad para generar facturas y balances mensuales, provocando retrasos en las entregas de facturas y a la vez malestar entre los usuarios.

Por tales motivos, resulta imprescindible la implementación de un sistema que automatice el proceso actual, a través del desarrollo de un software para la Gestión Administrativa de las Juntas Comunitarias de Agua de Consumo Humano en las Provincias de Pichincha e Imbabura.

Justificación

La implantación de un sistema de gestión administrativa de fácil administración y amigable al usuario, que automatice el proceso de facturación responde a la dificultad que representa el manejo manual de información y valores. Por otra parte, la automatización beneficiará a corto y mediano plazo a todos los miembros de las Juntas Comunitarias.

Los beneficios más significativos que se derivan de la implementación de un sistema son los siguientes:

Reportes estadísticos del consumo de agua mensual por cada familia, brindando información relevante para los encargados de mantener la igualdad en el servicio.

Ubicación precisa de cada junta comunitaria en la provincia de Imbabura y Pichincha.

Pantalla de inicio con información de eventos a generarse por cada junta mancomunada.

Aplicación móvil para ingresar lecturas del consumo de agua potable por cada medidor, facilitando la recolección de la información.

Por otro lado, la arquitectura a ser diseñada brindará facilidad de futuras

implementaciones y actualizaciones de manera eficiente.

Objetivo general

Analizar, diseñar e implementar un sistema de facturación, para mejorar la gestión administrativa de las Juntas Comunitarias de agua de consumo humano en las Provincias de Pichincha e Imbabura.

Objetivos específicos

Consolidar los requerimientos administrativos que puedan existir en cada una de las juntas e implementarlo para dar cumplimiento en un solo producto.

Generar una aplicación genérica que permita a todas las juntas manejar de manera sencilla la administración de los ingresos y egresos del servicio de agua potable.

Identificar qué datos de ingresos y egresos son indispensables, de manera que se permitan satisfacer las necesidades de las juntas.

Determinar los roles a utilizar para mantener la seguridad de la aplicación y de la información generada por cada una de las juntas.

Capacitar a los dirigentes y personal administrativo de las juntas comunitarias en el manejo del sistema, obteniendo mayor productividad y mejorando la gestión administrativa de las juntas.

Realizar la parametrización de variables en función de los requerimientos obtenidos, para obtener un sistema flexible en dependencia de cada junta.

Capítulo 1

Marco Teórico

1.1 Proceso de recaudación

El proceso de recaudación (facturación) es el punto de relación más importante entre el usuario (consumidor) y la empresa, a través de la factura como el principal medio de cobranza del servicio prestado, en este caso del servicio de agua para el consumo humano. En el caso de las comunidades de las Juntas Comunitarias de las provincias de Pichincha e Imbabura llevan un sistema de facturación con relación al consumo generado.

1.1.1 Descripción del proceso.

El proceso de recaudación se realiza de una forma manual, el cual dificulta la exactitud de los datos y errores al momento del cálculo del valor de la factura.

El Operador que es la persona asignada por la Junta acude a cada uno de los domicilios que tiene el servicio de agua, quien al finalizar el mes registra la métrica actual manualmente del consumo en el momento y realiza el mismo proceso con todas las casas del sector asignado.

La Junta según su reglamento tiene establecido las tarifas por consumo doméstico de agua, por lo cual al finalizar la recopilación de los datos de todas las acometidas o medidores se realiza el ingreso manual de los datos mediante la aplicación Excel, de acuerdo a las tarifas establecidas se verifica si el consumo mensual está dentro de lo establecido y se da a conocer al usuario el pago básico que debe realizar, si excede los m³ del consumo básico cada m³ tiene un costo extra el cual se suma al momento de la cancelación de la misma.

Para el proceso de multas no se lleva un control, estos valores no tienen la forma de ser auditados por tal motivo los valores son difíciles de controlar.

Cada junta tiene instalados los medidores de acuerdo al número de usuarios del agua, pero existen juntas donde no existe aún medidores si no que el consumo es abierto y no se puede controlar el consumo.

El proceso manual de facturación de las Juntas Comunitarias de las provincias de Pichincha e Imbabura tiene ciertos inconvenientes al momento de realizar la auditoria correspondiente, ya que el ingreso tiene más demora a veces los datos se pierden y por tal motivo existen retrasos en la entrega del balance.

1.2 Herramientas y Tecnología

Las principales herramientas y tecnología que servirán de base para este proyecto son:

1.2.1 Programación Orientada a Objetos.

La programación Orientada a objetos permite tener un enfoque más cercano al pensamiento del ser humano, además permite tener un equilibrio entre los procesos y los datos. Tiene un gran cambio respecto al anterior paradigma como es la herencia la cual facilita el crecimiento y el mantenimiento de nuestras aplicaciones. (Bernal Bermúdez & Fernández Muñoz , 2012, pág. 7).

Entre los fundamentos más relevantes que se utilizó dentro del presente software se tiene:

1.2.1.1 Abstracción.

La abstracción se trata de un procedimiento muy subjetivo en el cual el observador extrae las características principales, eliminando detalles que pudieran ser redundantes y que aumentarían el trabajo. (Bernal Bermúdez & Fernández Muñoz , 2012, pág. 7)

1.2.1.1.2 Encapsulación.

La Encapsulación trata de un proceso en el cual se almacenan todos los detalles relevantes que describan una abstracción. Se tiene dos partes dentro de la encapsulación, el primer tipo es el que ayuda a la funcionalidad y la segunda la que incorpora detalles internos. (Bernal Bermúdez & Fernández Muñoz , 2012, pág. 7)

1.2.1.1.3 Modularidad.

Permite dividir a un sistema en pequeñas partes. Todas las partes deben estar asociadas y conectadas. (Bernal Bermúdez & Fernández Muñoz , 2012, pág. 7)

2.2.1.1.4 Jerarquización.

La jerarquización no es la que la estructuración de todos los elementos por niveles y dentro de este concepto se incorporan conceptos con ciertos elementos como son: clases y objetos, métodos, atributos y estado y mensajes, herencia y polimorfismo. (Bernal Bermúdez & Fernández Muñoz , 2012, pág. 7)

1.2.2 Base de datos Relacionales.

Una base de datos relacional consiste en una colección de datos interrelacionados y un conjunto de aplicaciones que permitan el acceso a dichos datos. Las bases de datos relacionales permiten el almacenamiento de grandes cantidades de información relevante

para nuestra empresa. (Silberschatz, Korth, & Sudarshan, 2002, pág. 1).

Una base de datos relacional es un repositorio compartido de los datos. Y para hacer accesible la información de estos datos es necesario considerar aspectos como:

La forma que los usuarios hacen peticiones de los datos.

La seguridad e integridad de los datos. (Silberschatz, Korth, & Sudarshan, 2002, pág. 84)

Para el diseño de una base de datos relacional es necesario el diseño de un esquema relacional como primer paso, pero es necesario tomar en cuenta principios que faciliten la eficiencia en la consulta de los datos y que evitan inconsistencias. (Silberschatz, Korth, & Sudarshan, 2002, pág. 84)

2.2.2.2 SQL.

Es un lenguaje que se soporta en el modelo relacional. Contiene estándares que permiten incluir construcciones orientadas a objetos. En cuanto a los lenguajes orientados a objetos como Java y C++, los objetos se relacionan entre sí de manera que permiten al lenguaje soportar problemas complejos que no permitirían resolver los lenguajes tradicionales. (Sheldon & Oppel, 2010, pág. 17)

2.2.2.3 Postgresql.

“PostgreSQL es un gestor de bases de datos orientadas a objetos (SGBDOO o ORDBMS en sus siglas en inglés) muy usado y conocido en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos SGBD comerciales”. (Perez Mora & Gibert Ginestà, 2009, pág. 5)

“PostgreSQL puede funcionar en múltiples plataformas (en general, en todas las modernas basadas en Unix) y, a partir de la próxima versión 8.0 (actualmente en su segunda beta), también en Windows de forma nativa. Para las versiones anteriores existen versiones binarias para este sistema operativo, pero no tienen respaldo oficial”. (Perez Mora & Gibert Ginestà, 2009, pág. 5)

1.2.3 Aplicativo Web

“El éxito espectacular de la web se basa en dos puntales fundamentales: el protocolo HTTP y el lenguaje HTML. Uno permite una implementación simple y sencilla de un sistema de comunicaciones que permite enviar cualquier tipo de ficheros de una forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan miles de peticiones y reduzcan los costes de despliegue”. (Mateu, 2004, pág. 13).

1.2.3.2 Protocolo HTTP.

“El protocolo HTTP (hypertext transfer protocol) es el protocolo base de la WWW. UN protocolo simple, orientado a conexión y sin estado. La forma de que esté orientado a conexión es que emplea para su funcionamiento un protocolo de comunicaciones (TCP, transport control protocol) de modo conectado, un protocolo que establece un canal de comunicaciones de extremo a extremo (entre el cliente y el servidor)”. (Mateu, 2004, págs. 13,14)

1.2.3.3 JSF.

“JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas

en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa JavaServer Pages (JSP) como la tecnología permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL”. (Cédric , 2011, pág. 6)

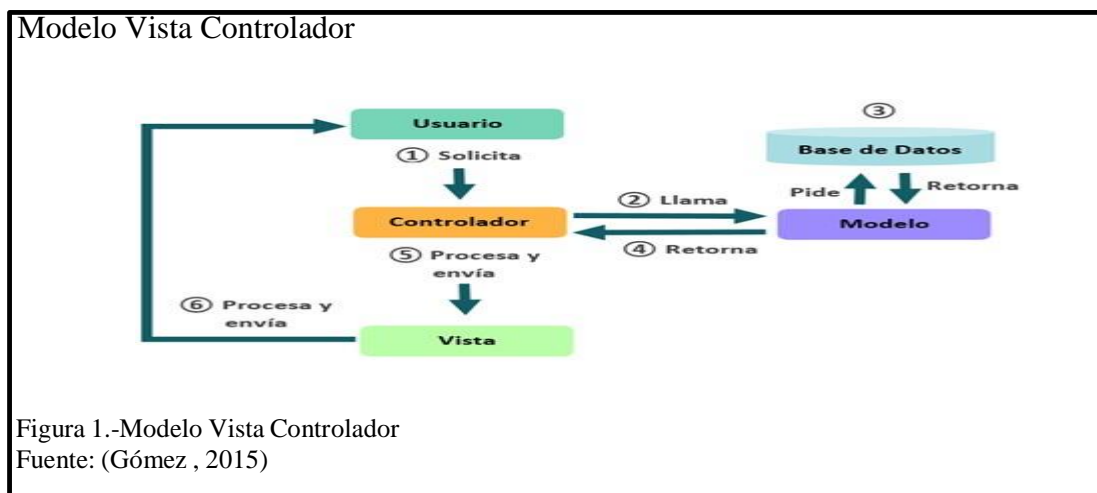
1.2.3.4 Primefaces.

PrimeFaces es un paquete de componentes JSF con varias extensiones. Amplio conjunto de elementos (Editor HTML, Diálogo, Autocompletar, Gráficos y muchos más). De código libre, ligero y con interfaz de aplicaciones móviles web para móviles. Ajax incorporado basado en las API estándar de AJAX JSF 2.0. (Civici, 05, pág. 10).

1.2.4 Modelo Vista Controlador.

EL Modelo Vista Controlador se define como un patrón de diseño en cuanto a la arquitectura de software.

Este patrón se estructura por tres componentes los cuales se puede visualizar en la siguiente figura.



2.2.4.1 Modelo.

El Modelo es el parte que interpreta los datos del programa. Manipula los datos y controla todos sus cambios. El Modelo no tiene conocimiento definido de los Controladores o de las Vistas, ni siquiera contiene información de ellos. (Sáenz Morras, 2008, pág. 12).

2.2.4.2 Vista.

La Vista es la parte que maneja la presentación visual de los datos representados por el Modelo. Realiza una presentación visual del Modelo y despliega los datos al usuario. Se relaciona con el Modelo a través de una relación al propio Modelo. (Sáenz Morras, 2008, pág. 12)

2.2.4.3 Controlador.

El Controlador es la parte que proporciona razón a las órdenes del usuario, interviniendo sobre los datos representados por el Modelo. Cuando se realiza algún cambio, actúa, ya sea por cambios en la información del Modelo o por modificaciones de la Vista. (Sáenz Morras, 2008, pág. 12)

1.2.5 Android.

Android es un sistema operativo y una plataforma software libre, basado en Linux para dispositivos móviles e incluso PC's. Android permite programar en un entorno de trabajo (framework) de Java, aplicaciones sobre una máquina virtual Dalvik (una variación de la máquina de Java con compilación en tiempo de ejecución). (Báez, y otros, 2012, pág. 1)

1.3 Metodología

1.3.1 Scrum.

Scrum al ser una metodología que permite resolver problemas complejos adaptativos. Scrum es una metodología ligera, fácil de entender pero difícil de llegar a dominar, es un marco que permite utilizar varias técnicas y procesos. (Sánchez López & Letelier Torres, 2003)

Scrum presenta la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo. El marco de trabajo Scrum consta de los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Todos son una parte importante para poder conseguir el éxito. (Sánchez López & Letelier Torres, 2003)

La metodología Scrum se basa en el empirismo El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum utiliza un enfoque iterativo e incremental para mejorar la predictibilidad y el control del riesgo. Tres pilares apoyan toda la implementación del control de procesos empírico: inspección, transparencia y adaptación. (Sánchez López & Letelier Torres, 2003)

1.3.2 Equipo Scrum.

El Equipo Scrum se compone un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. Los Equipos Scrum son multifuncionales y organizados. Los equipos organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. (Sánchez López & Letelier Torres, 2003)

1.3.2.1 Dueño de Producto (Product Owner).

El Dueño de Producto es la persona responsable de incrementar el valor del producto y del trabajo del Equipo de Desarrollo. El Dueño de Producto es la única persona responsable de gestionar la lista del Producto, expresar claramente los elementos del producto, asegurar que el Equipo de desarrollo entienda los elementos producto al nivel necesario, y es la única persona responsable del producto así delegue a otra persona sus funciones. (Sánchez López & Letelier Torres, 2003)

2.3.2.2 El Equipo de Desarrollo (Development Team).

El Equipo de Desarrollo consiste en los profesionales que desarrollan el trabajo de entregar un Incremento de producto, que se pueda poner en producción, al final de cada Sprint. Solo los miembros del Equipo de Desarrollo los únicos de ejecutar la tarea de desarrollar cada incremento.

El equipo de trabajo debe ser capaz de organizarse por sí mismo, debe de ser multifuncionales y todos son Desarrolladores, muy independiente del trabajo que realiza; no hay excepciones a esta regla. (Sánchez López & Letelier Torres, 2003)

2.3.2.3 El Scrum Master.

El Scrum Master es la persona que dirige al equipo de desarrollo, es la persona que debe hacer que el equipo de desarrollo entienda la metodología. El Scrum Master apoya a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum. El Scrum Master ayuda a todos a ajustar estas interacciones para incrementar el valor creado por el Equipo Scrum. (Sánchez López & Letelier Torres, 2003)

1.3.3 Uso de Scrum

El paradigma de las metodologías ágiles en la actualidad es un tema muy común de escuchar, puesto que ha tenido una gran aceptación por la mayoría de las organizaciones, debido a que están dando excelentes resultados.

La forma en que estas metodologías permiten adaptarse a cualquier cambio, así como los valores y principios que estos generan es lo que ha hecho que para el presente proyecto se seleccionó una de estas metodologías como es SRUM.

SCRUM es una metodología que se adapta perfectamente a esta clase de proyectos de carácter funcional. Además, que permite hacer seguimiento de cada uno de los avances que se genera y que permite tener un gran apoyo con los clientes al desarrollar las reuniones en las cuales se puede tener una retroalimentación muy importante.

Otra de las razones por las cuales se seleccionó esta metodología es porque tiene claro cada una de las actividades que realiza cada uno de los roles que esta involucra, pudiendo de esta manera manejar de mejor manera el presente proyecto.

1.3.3.1 Roles de SCRUM utilizados.

1.3.3.1.1 Scrum Master.

El Scrum Master aquella persona que lidera al equipo para que este cumpla con todas los procesos y reglas, quien facilita y guía la colaboración entre todo el equipo. Durante el desarrollo de este proyecto el papel de ScrumMaster fue asumido por dos personas primordiales:

Como primera persona se tuvo como Tutor al Ing. Robinson Llerena el cual permitió que se gestione de correcta manera el proyecto, además que dio las pautas de los errores cometidos para para la corrección posterior.

Como segunda persona se tuvo a la Lic. Alexandra Moya quien facilitó las reuniones de Scrum con los diferentes miembros de la Juntas Comunitarias, además que incentivó a mantener el compromiso para que se desarrolle el proyecto en el menor tiempo posible.

1.3.3.1.2 Propietario del producto (Product Owner).

En este caso los propietarios del producto al cual está enfocado el proyecto son las Juntas Comunitarias de las provincias de Pichincha e Imbabura, quienes por medio de sus representantes se vieron en la necesidad de la ejecución de este proyecto y quienes proporcionaron la información de los diferentes procesos que se realizaba en cada uno de los módulos (requerimientos funcionales), además que fueron los que ayudaron en la toma de decisiones por medio de la aprobación de cada uno de los sprints desarrollados.

Por otra parte vendría a participar como propietario del producto la Universidad Politécnica Salesiana quién por medio del área de Vinculación con la Colectividad en la Ciudad de Cayambe, pudo ayudar con la identificación y colaboración para el exitoso desarrollo de este proyecto.

1.3.3.1.3 El Equipo (Team).

El equipo o conjunto de personas que comparten la responsabilidad del desarrollo del proyecto con las diferentes responsabilidades que esto involucra. En este caso fue asumido por nuestras personas quienes adoptamos las responsabilidades del desarrollo de este

proyecto con un fin en común como fue el desarrollo exitoso del mismo.

Priorizando los requerimientos, planificando el trabajo de cada iteración, estimando el esfuerzo de cada tarea, planificando las reuniones con coordinación de los Scrum Master, entre otras actividades para llegar a cumplir nuestra visión.

2.3.3.2 Sprints Realizados.

Durante la ejecución del proyecto se ha planificó los Sprints de acuerdo a la disponibilidad de tiempo de cada uno de los integrantes del equipo debido a sus responsabilidades, en el cual se dispuso de un horario de 3 a 4 horas diarias en la jornada normal (lunes a viernes) y de un horario de 6 a 8 horas en los fines de semana (sábados y domingos). Además que se tomó en cuenta la complejidad de cada una de las tareas, tomando en cuenta las respectivas correcciones y variaciones de las mismas.

De esta manera dio como resultado un total de 8 sprints los cuales tuvieron una duración de 3 a 4 semanas, dependiendo la complejidad de las tareas que cada uno involucraba y el detalle de cada uno se describe a continuación.

2.3.3.3 Planificación de reuniones.

2.3.3.3.1 Daily Sprint Meeting (Reuniones diarias).

Estas reuniones se llevaron a cabo con herramientas las cuales facilitaron la coordinación de las actividades y la retroalimentación de las mismas, herramientas como Hangouts. Además que cuando se tuvo la disponibilidad de tiempo se las realizó presencialmente dentro de los laboratorios de la Universidad para mayor interacción y generación de las

mejores soluciones.

2.3.3.3.2 *Demo y retrospectiva.*

Estas reuniones destinadas al final de cada uno de los Sprints, en este caso se las realizó con coordinación de las Juntas Comunitarias y la representante de Vinculación de Colectividad. Se las realizó de acuerdo a la disponibilidad de los representantes de las Juntas Regionales, en muchos de los casos en Cayambe o en la ciudad de Ibarra.

Estas reuniones se las llevó a cabo con el objetivo de verificar si lo desarrollado cumplía con los requerimientos del Product Owner, para adquirir retroalimentación para futuras mejoras y para establecer claramente los requerimientos para el próximo Sprint.

1.3.4 Uso de UML.

Para mejorar la explicación de cada uno de los módulos y sus funciones de una forma más explícita se utilizó como artefactos los diagramas de casos de uso esto con el propósito de abstraer el comportamiento de cada una de las actividades que requería el usuario de forma gráfica, además que de facilitar el desarrollo.

Como otro artefacto se utilizó el diagrama físico de la base de datos, el cual se desplego posteriormente en el gestor de base de datos Postgresql. Siempre llevando a cabo los cambios que se iban generando en cada uno de los incrementos realizados, teniendo una versión diferente por cada Sprint.

1.3.5 Uso de la PBS.

Esta herramienta de gestión de proyectos benefició de gran manera, puesto que de esta manera se pudo establecer una visión completa del alcance del presente proyecto, de sus componentes y productos que eran necesarios para la ejecución del presente proyecto.

Permitió descomponer en partes más pequeñas y manejables del producto de manera gráfica, convirtiendo la complejidad en simplicidad. Además que permitió priorizar el desarrollo de cada uno de los componentes y establecer la relación que existía en ellos para de esta manera poder reutilizar componentes y disminuir el trabajo.

Esta herramienta como es el PBS es de gran utilidad ya que permite mejorar la calidad, tomar de mejor manera las decisiones y disminuir el número de fallos.

1.3.6 Uso del Prototipado.

Este proyecto al ser de naturaleza cambiante y evolutiva se llevó a cabo con el uso del prototipado con el cuál en primera instancia permitió establecer una idea de los requerimientos y funcionalidades que el Product Owner necesitaba, en segunda instancia permitió al final de cada sprint llevar a cabo el proceso de aceptación y de retroalimentación en cuanto a requerimientos faltantes o que tenían una mala interpretación.

Esto benefició de tal manera que al final de cada sprint se obtuvo un prototipo nuevo que se acercaba más a los requerimientos que el Product Owner deseaba obtener.

Capítulo 2

Análisis y diseño

En este capítulo se detallan los módulos y diagramas establecidos en la etapa de análisis y diseño para la elaboración del proyecto.

2.1 Requerimientos

2.1.1 Requerimientos Funcionales.

Entre los requerimientos funcionales del presente proyecto se pudo destacar los siguientes:

Gestión administrativa de personas, regionales, sectores, medidores.

Operaciones C.R.U.D sobre cada módulo.

Parametrización de valores a cobrar por cada sector ingresado.

Registró de consumo a cobrar por medio de portal administrativo o paliativo móvil

Generar facturas por cada persona con su respectivo valor a cancelar.

Exportación de reportes con información del consumo generado mensual.

2.1.2 Requerimientos no funcionales.

Entre los requerimientos no funcionales del presente proyecto se pudo destacar los siguientes:

Implementación de interfaz usando User Xperience para mejorar manejo del sistema

Manejo de persistencia a base de datos usando JPA, para mejorar acceso y tiempos de respuesta.

Uso de servicios web planos (REST), para conexión con paliativo móvil facilitando su

acceso y beneficiando a su rendimiento.

Generación de reportes en formato PDF, por facilidad de uso e interpretación.

2.2 Diagramas de Casos de Uso

2.2.1 Diagramas del módulo de Administración.

2.2.1.1 Diagrama de la Gestión de Usuarios.

2.2.1.1.1 Diagrama de la creación de Usuarios.

A continuación se muestra el proceso que se sigue para la creación de los diferentes usuarios que procesan información del sistema.

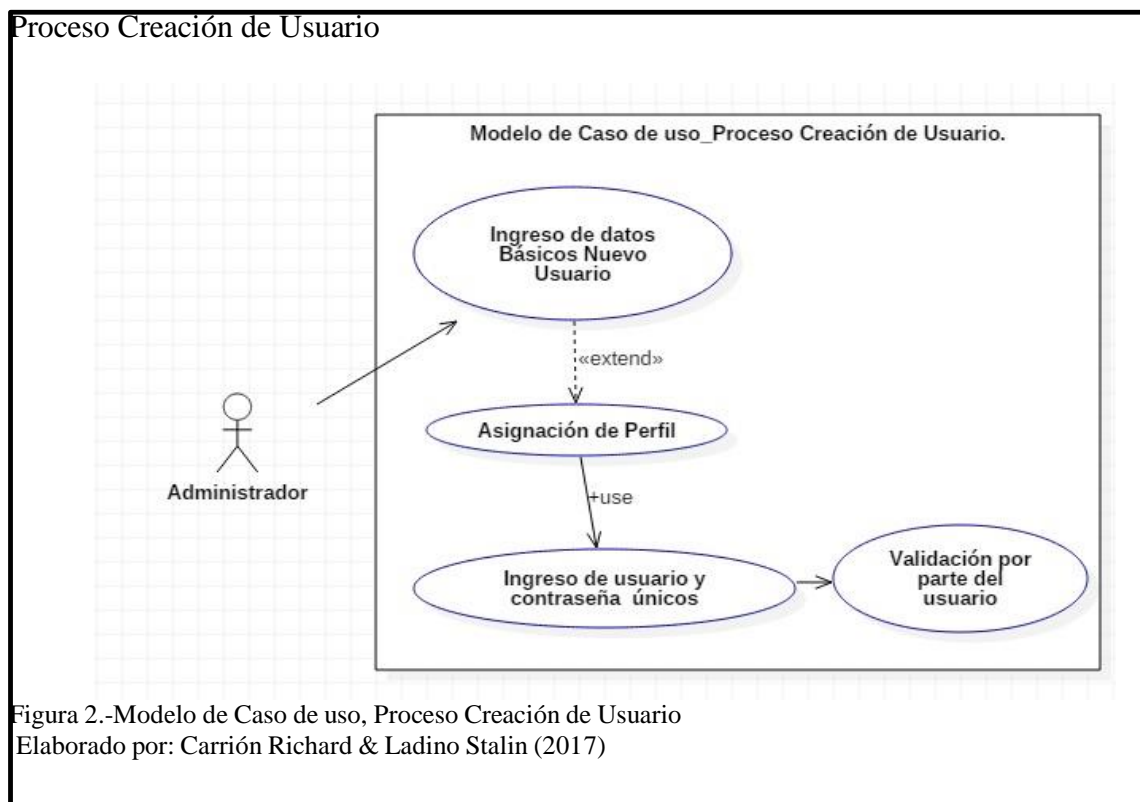


Figura 2.-Modelo de Caso de uso, Proceso Creación de Usuario
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Tabla 1.- Escenario de Caso de Uso, Proceso Creación de Usuario

ACTORES	
Administrador	
PRECONDICIONES	El usuario no debe estar registrado en el sistema.
POSCONDICIONES	El usuario se ingresará al sistema siempre y cuando se ingresen todos los datos correctamente.
FLUJO BÁSICO	<p>El flujo básico debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • El usuario a crear debe ingresar los datos personales (cedula, nombre, dirección, sector, rol, identidad étnica, número de familiares). • Se verifica que los datos estén correctos y se procede con el ingreso respectivo a la base de datos.
FLUJO ALTERNATIVO	<p>El flujo alternativo debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Si los datos no son correctos o no son ingresados, el sistema presenta un mensaje en el que se indica que los datos ingresados por el usuario son incorrectos o se deben ingresar.

Nota: Se presenta el escenario para la creación de usuarios
 Elaborado por: Carrión Richard & Ladino Stalin

2.2.1.1.2 Diagrama de la actualización y eliminación de Usuarios.

A continuación se muestra el proceso que se sigue para la actualización y eliminación de los diferentes usuarios que procesan información del sistema.

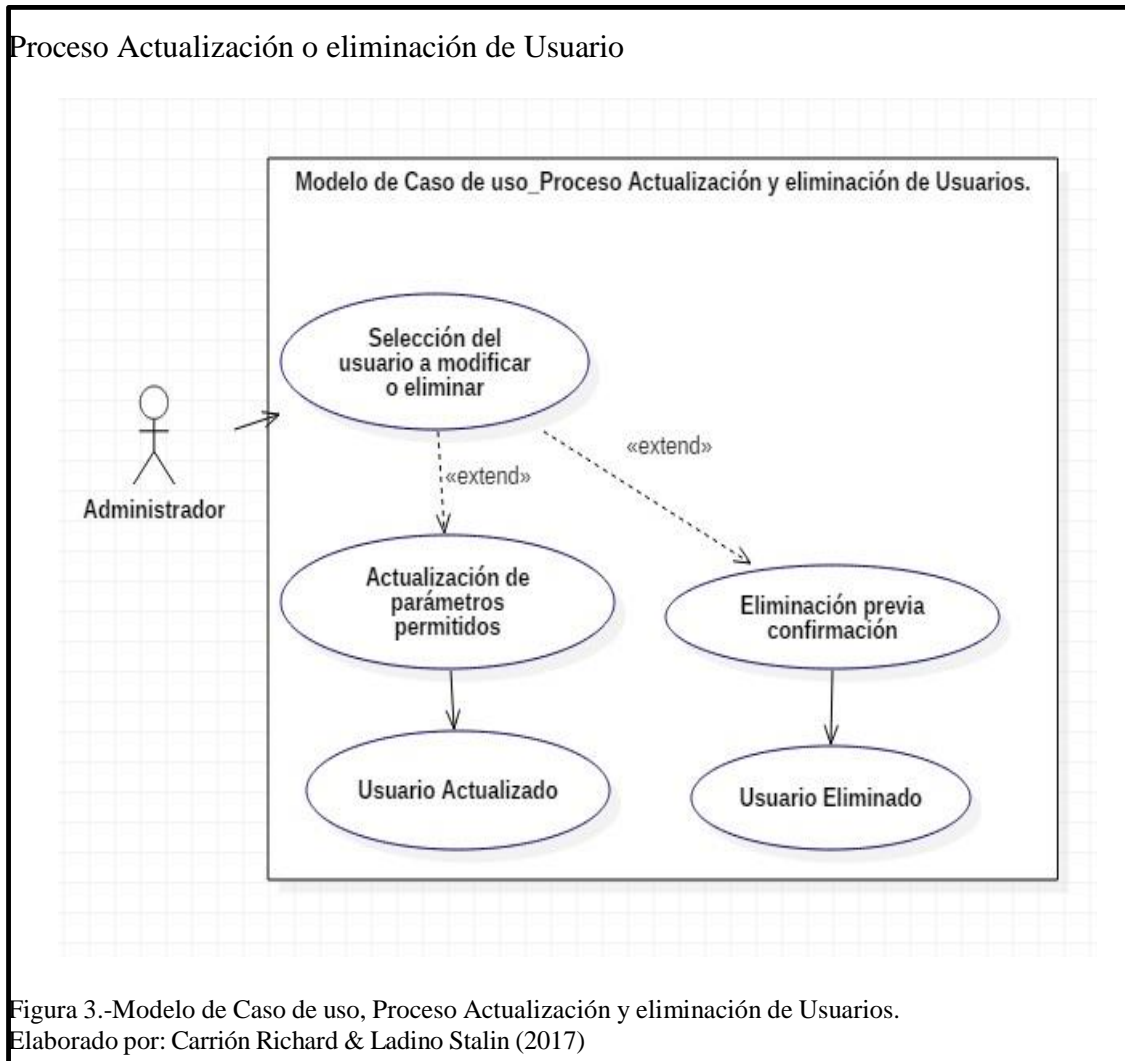


Figura 3.-Modelo de Caso de uso, Proceso Actualización y eliminación de Usuarios.
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Tabla 2.- Escenario de Caso de Uso, Proceso Actualización o Eliminación de Usuario

ACTORES	
	Administrador
PRECONDICIONES	El usuario debe estar registrado en el sistema.
POSCONDICIONES	<p>El usuario se actualizará del sistema, siempre y cuando se actualicen correctamente los datos.</p> <p>El usuario se eliminará del sistema, siempre y cuando se confirme la eliminación del mismo.</p>
FLUJO BÁSICO	<p>El flujo básico debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Se debe seleccionar el usuario a eliminar o actualizar. • En el caso de actualizar, se podrán actualizar solo los campos que son permitidos. • En el caso de eliminar, se eliminará el usuario siempre y cuando se seleccione dicha opción.
FLUJO ALTERNATIVO	<p>El flujo alternativo debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Si los datos a actualizar no son correctos o no son ingresados, el sistema presenta un mensaje indicando que los datos a actualizar son incorrectos o se deben ingresar.

Nota: Se presenta el escenario para la actualización y eliminación de usuarios

Elaborado por: Carrión Richard & Ladino Stalin

2.2.1.2 Diagrama de la Gestión de Medidores.

2.2.1.2.1 Diagrama de la creación de Medidores.

A continuación se muestra el proceso que se sigue para la creación de los diferentes medidores a asignar a cada uno de los usuarios de los diferentes sectores y de acuerdo a la regional a la que pertenecen., consecuentemente para la toma de las lecturas.

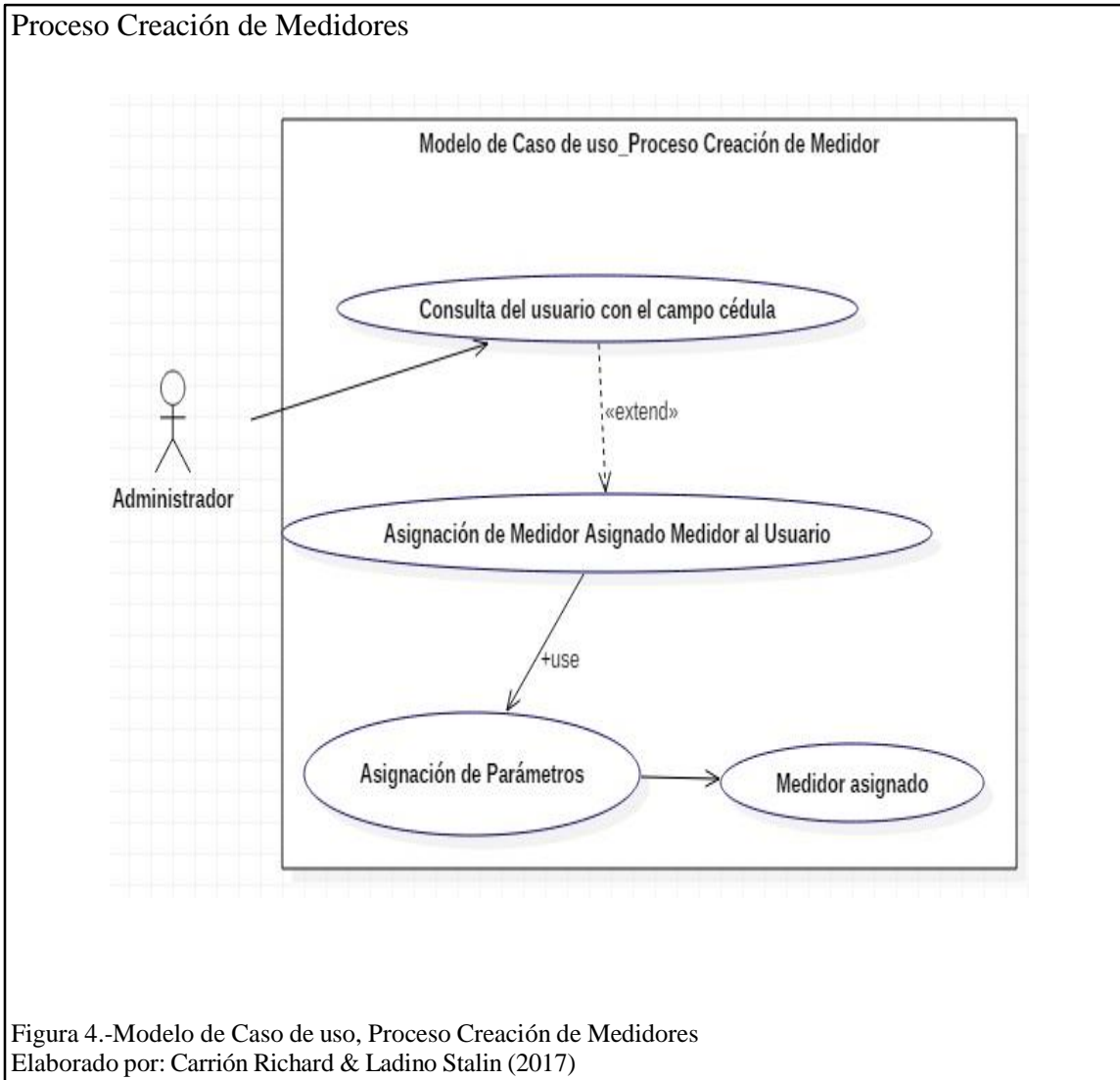


Tabla 3.- Escenario de Caso de Uso, Proceso Creación de Medidores

ACTORES	
	Administrador
PRECONDICIONES	El medidor no debe estar registrado en el sistema.
POSCONDICIONES	El medidor se ingresará al sistema siempre y cuando se ingresen todos los datos correctamente.
FLUJO BÁSICO	<p>El flujo básico debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Para la creación del medidor se debe hacer una consulta si existe el usuario, en el caso de que exista se deben ingresar los datos requeridos (Serie del medidor, modelo, tiempo de vida, contador inicial, fecha de ingreso) y se debe seleccionar el tipo de medidor, la regional y sector al que pertenece. • Se verifica que los datos estén correctos y se procede con el ingreso respectivo a la base de datos.
FLUJO ALTERNATIVO	<p>El flujo alternativo debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Si los datos no son correctos o no son ingresados, el sistema presenta un mensaje indicando que los datos ingresados por el usuario son incorrectos o se deben ingresar. • Si el usuario al cual se quiere asignar el medidor no existe, se mostrará un mensaje de notificación.

Nota: Se presenta el escenario para la creación de Medidores
Elaborado por: Carrión Richard & Ladino Stalin

2.2.1.2.2 Diagrama de la actualización y eliminación de Medidores.

A continuación se muestra el proceso que se sigue para la actualización y eliminación de los diferentes medidores una vez que sea asignado a cada usuario.

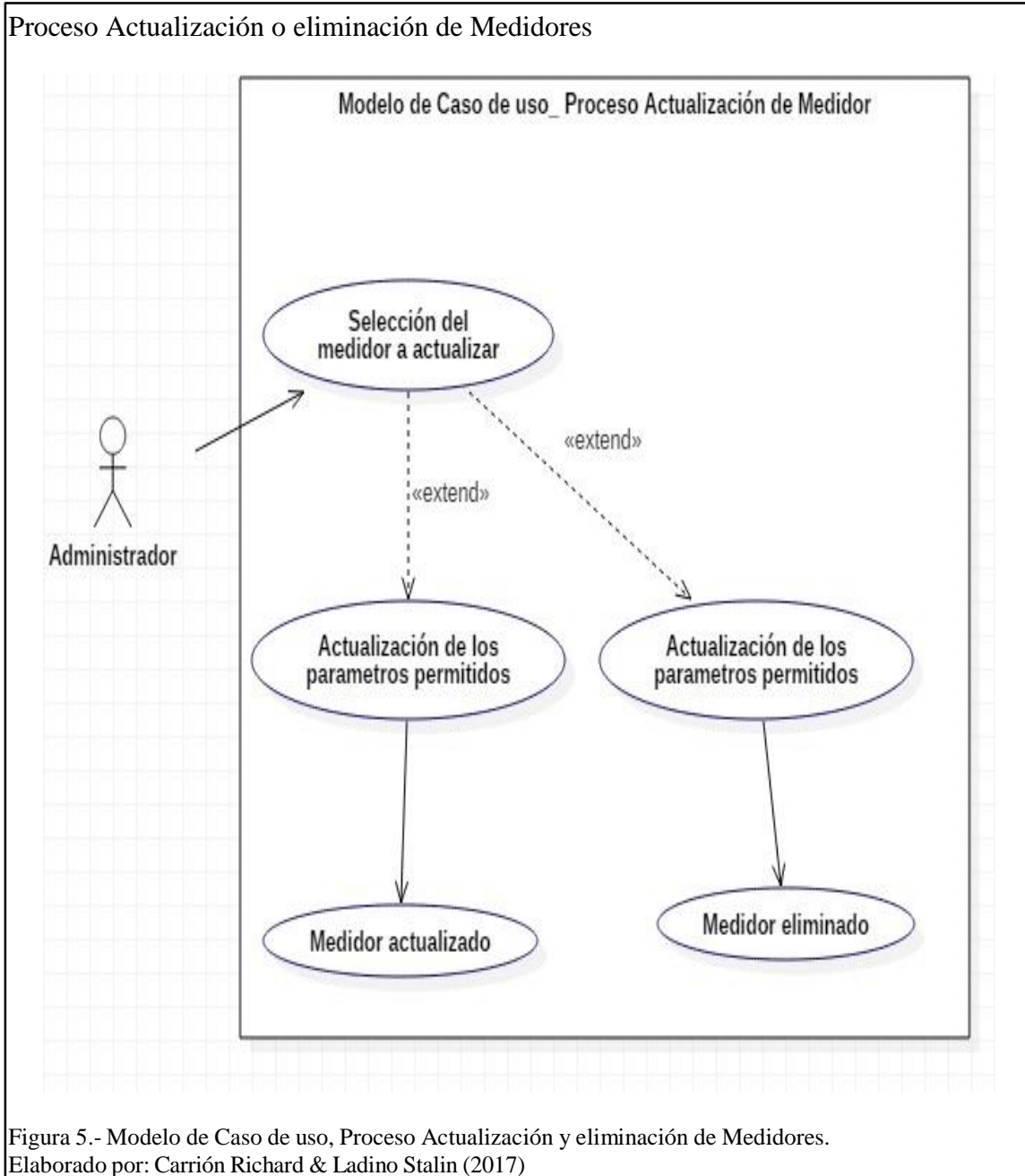


Tabla 4.- Escenario de Caso de Uso, Proceso Actualización o Eliminación de Medidores

ACTORES	Administrador
PRECONDICIONES	El medidor debe estar registrado en el sistema.
POSCONDICIONES	<p>El medidor se actualizará del sistema, siempre y cuando se actualicen correctamente los datos.</p> <p>El medidor se eliminará del sistema, siempre y cuando se confirme la eliminación del mismo.</p>
FLUJO BÁSICO	<p>El flujo básico debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Se debe seleccionar el medidor a eliminar o actualizar. • En el caso de actualizar, se podrán actualizar solo los campos que son permitidos. • En el caso de eliminar, se eliminará el medidor siempre y cuando se seleccione dicha opción.
FLUJO ALTERNATIVO	<p>El flujo alternativo debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Si los datos a actualizar no son correctos o no son ingresados, el sistema presenta un mensaje indicando que los datos a actualizar son incorrectos o se deben ingresar.

Nota: Se presenta el escenario para la actualización y eliminación de Medidores
Elaborado por: Carrión Richard & Ladino Stalin

2.2.1.3 Diagrama de la Gestión de autenticación.

2.2.1.3.1 Diagrama del proceso de autenticación.

A continuación se muestra el proceso que se sigue para la autenticación de cada uno de los diferentes usuarios, para su posterior ingreso al perfil asignado al mismo.

Proceso de autenticación

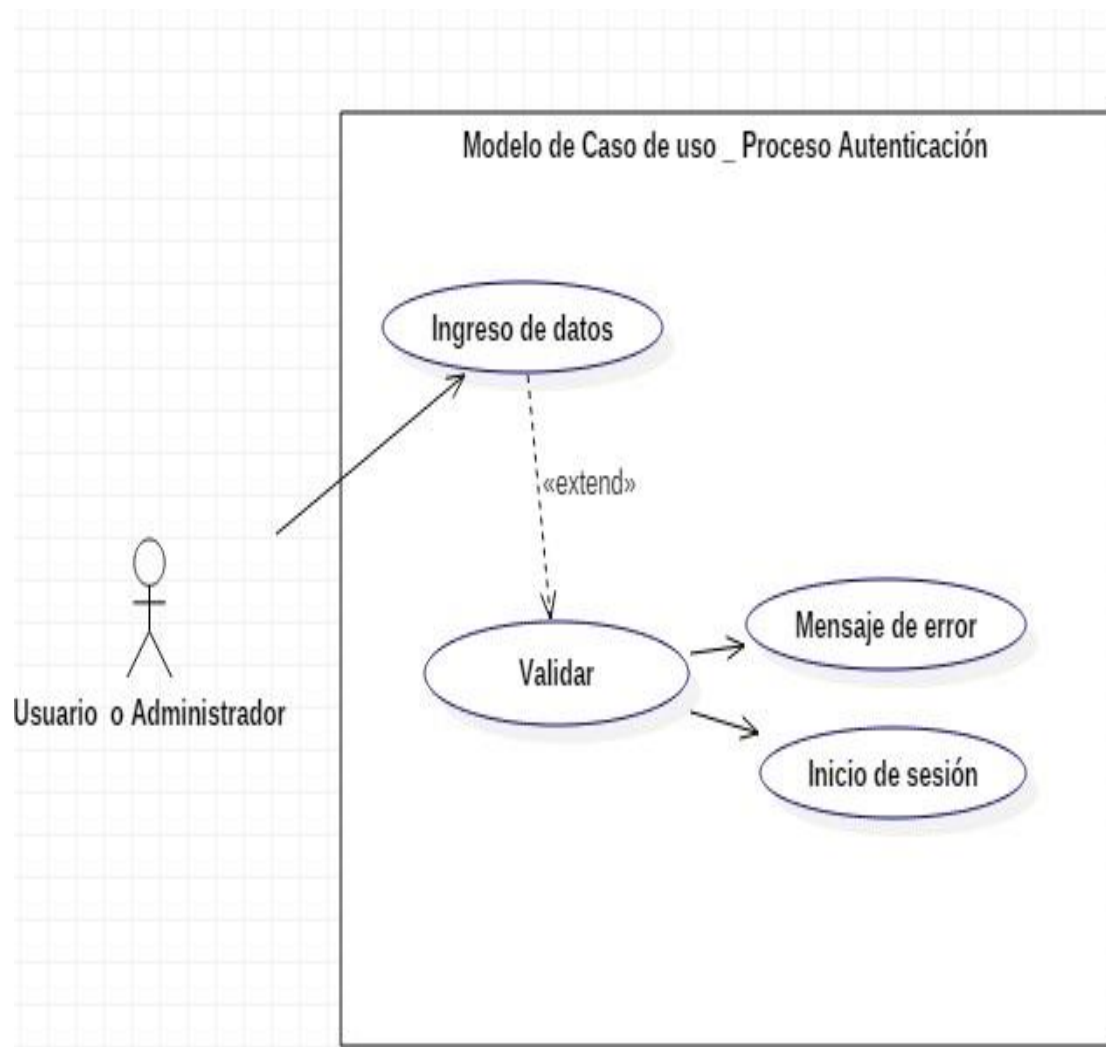


Figura 6.-Modelo de Caso de uso, Proceso de autenticación
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Tabla 5.-Escenario de Caso de Uso, Proceso de Autenticación.

ACTORES Administrador Usuario	
PRECONDICIONES	El usuario debe estar registrado en el sistema.
POSCONDICIONES	El usuario ingresará al sistema de acuerdo al rol asignado. El administrador ingresará con acceso a todos los módulos.
FLUJO BÁSICO	El flujo básico debe realizarse de la siguiente forma: <ul style="list-style-type: none"> • El usuario ingresa al sistema con el usuario y contraseña asignado al momento de la creación. • Se verifica que los datos estén correctos y se procede con el ingreso respectivo a la base de datos.
FLUJO ALTERNATIVO	El flujo alternativo debe realizarse de la siguiente forma: <ul style="list-style-type: none"> • Si los datos son incorrectos se mostrará un mensaje de notificación informando de que los datos son erróneos.

Nota: Se presenta el escenario para la autenticación de los diferentes usuarios.

Elaborado por: Carrión Richard & Ladino Stalin

2.2.1.4 Diagrama de la Gestión de Parámetros.

2.2.1.4.1 Diagrama de la creación de Parámetros.

A continuación se muestra el proceso que se sigue para la creación de los diferentes parámetros a asignar a cada uno de los medidores de los diferentes sectores, dichos parámetros ayudan al momento de la facturación para determinar a qué tipo de tarifa está asignado este medidor.

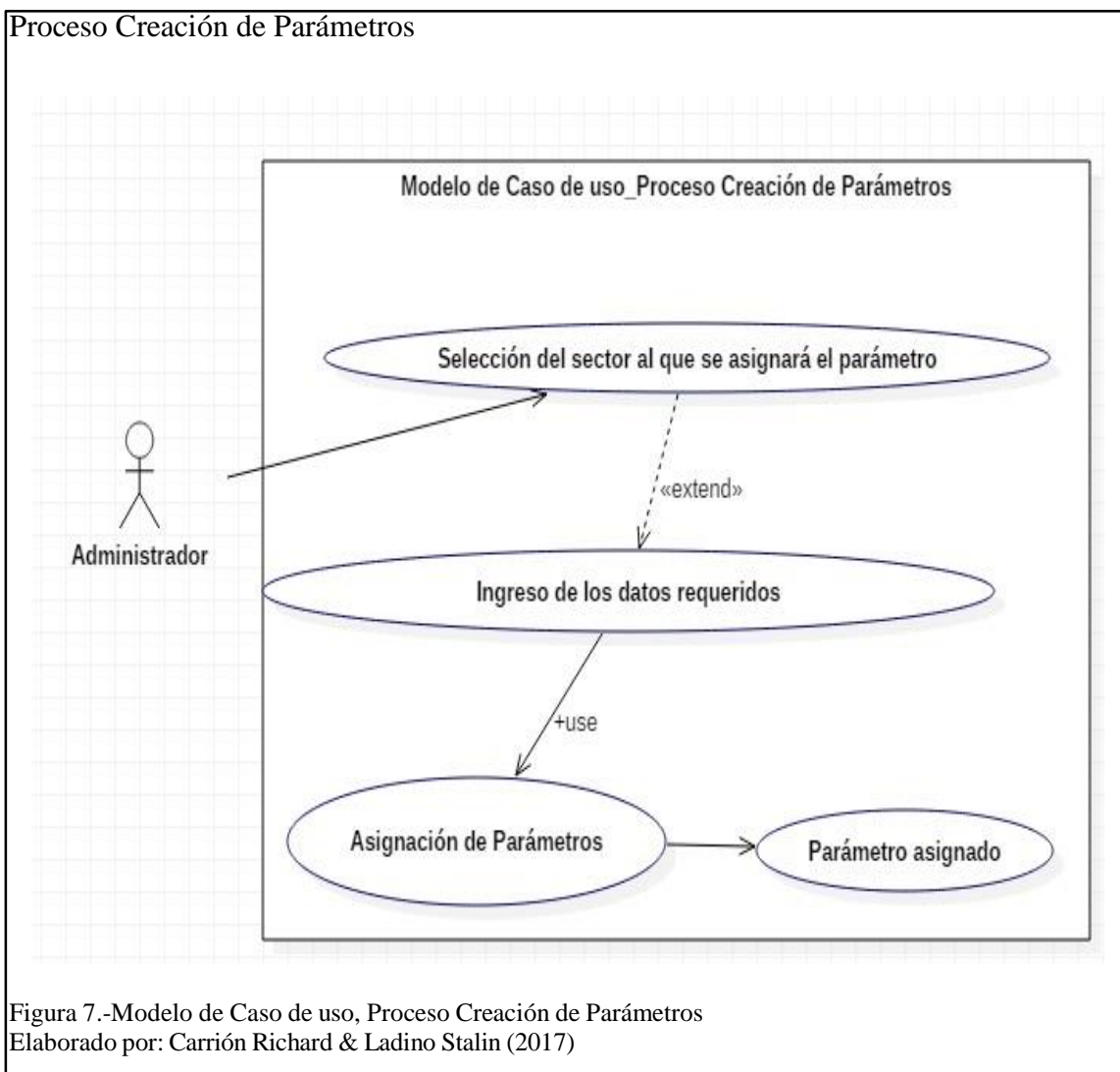


Tabla 6.- Escenario de Caso de Uso, Proceso Creación de Parámetros

ACTORES	
	Administrador
PRECONDICIONES	El parámetro no debe estar registrado en el sistema.
POSCONDICIONES	El parámetro se ingresará al sistema siempre y cuando se ingresen todos los datos correctamente.
FLUJO BÁSICO	<p>El flujo básico debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Para la creación del parámetro se debe seleccionar el sector al cual va ir asignado, se ingresan datos como son: registro mínimo, registro máximo, valor del m³, valor de instalación del medidor, servicio al que va orientado el medidor, y los valores relacionados a los excedentes. • Se verifica que los datos estén correctos y se procede con el ingreso respectivo a la base de datos.
FLUJO ALTERNATIVO	<p>El flujo alternativo debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Si los datos no son correctos o no son ingresados, el sistema presenta un mensaje en el que se indica que los datos ingresados por el usuario son incorrectos o se deben ingresar.

Nota: Se presenta el escenario para la creación de Parámetros
Elaborado por: Carrión Richard & Ladino Stalin

2.2.1.4.2 Diagrama de la actualización y eliminación de Parámetros.

A continuación se muestra el proceso que se sigue para la actualización y eliminación de los diferentes parámetros una vez que sea asignado a cada sector.

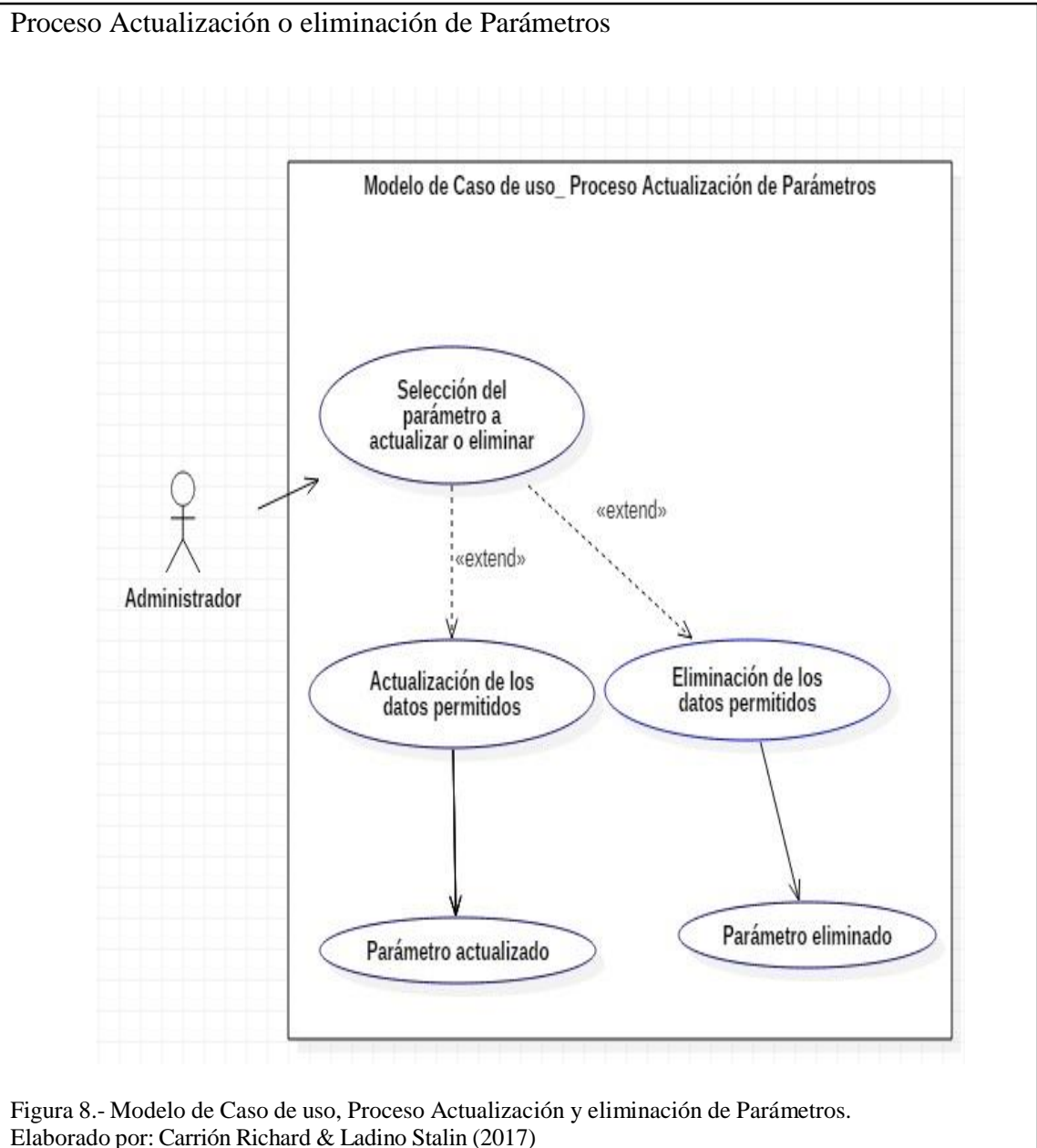


Tabla 7.- Escenario de Caso de Uso, Proceso Actualización o Eliminación de Parámetros

ACTORES	
	Administrador
PRECONDICIONES	El parámetro debe estar registrado en el sistema.
POSCONDICIONES	<p>El parámetro se actualizará del sistema, siempre y cuando se actualicen correctamente los datos.</p> <p>El parámetro se eliminará del sistema, siempre y cuando se confirme la eliminación del mismo.</p>
FLUJO BÁSICO	<p>El flujo básico debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Se debe seleccionar el parámetro a eliminar o actualizar. • En el caso de actualizar, se podrán actualizar solo los campos que son permitidos. • En el caso de eliminar, se eliminará el parámetro siempre y cuando se seleccione dicha opción.
FLUJO ALTERNATIVO	<p>El flujo alternativo debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Si los datos a actualizar no son correctos o no son ingresados, el sistema presenta un mensaje indicando que los datos a actualizar son incorrectos o se deben ingresar.

Nota: Se presenta el escenario para la actualización y eliminación de Parámetros.
Elaborado por: Carrión Richard & Ladino Stalin

2.2.2 Diagramas del módulo de Cobro y Facturación.

2.2.2.1 Diagrama de proceso de facturación.

A continuación se muestra el proceso que se sigue para la facturación de los diferentes usuarios que están asignados a un sector, en este proceso el usuario que dispone de la facultad de realizar la facturación tendrá la facilidad de realizar la búsqueda de diferentes formas, además de revisar que todos los detalles sean correctos y posteriormente emitir la factura correspondiente.

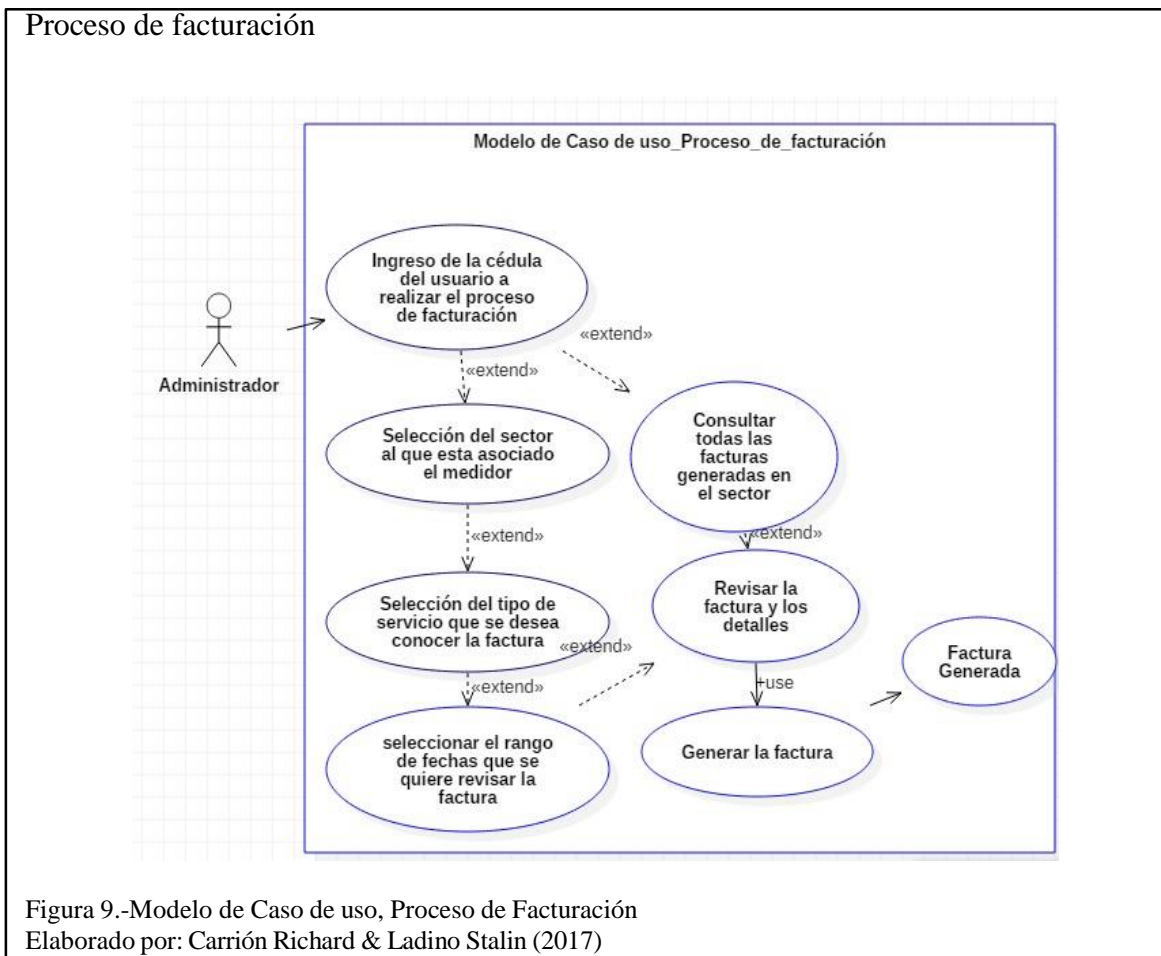


Tabla 8.- Escenario de Caso de Uso, Proceso de Facturación

ACTORES	
Administrador	
PRECONDICIONES	La aplicación móvil debe haber registrado la lectura con anticipación.
POSCONDICIONES	El administrador podrá generar la factura solo y cuando todos los valores sean correctos.
FLUJO BÁSICO	<p>El flujo básico debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Para generar la factura el administrador deberá hacer la consulta de la fecha, usuario al que se entregará la factura. • Una vez que seleccione que factura va a generar, el mismo podrá visualizar los detalles para estar seguro/a podrá continuar. • El Administrador generará la factura y entregará al usuario.
FLUJO ALTERNATIVO	<p>El flujo alternativo debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Si los datos no son correctos o tienen inconsistencias el proceso se cancelará hasta que estos valores se rectifiquen.

Nota: Se presenta el escenario para el proceso de facturación
 Elaborado por: Carrión Richard & Ladino Stalin

2.2.3 Diagramas del módulo de Aplicación Android.

2.2.3.1 Diagrama de la Gestión de autenticación.

2.2.3.1.1 Diagrama de ingreso del proceso de autenticación.

A continuación se muestra el proceso que se sigue para la autenticación de cada uno de los diferentes usuarios que tienen asignado el rol “Operador”, para su posterior ingreso para la toma de lecturas.

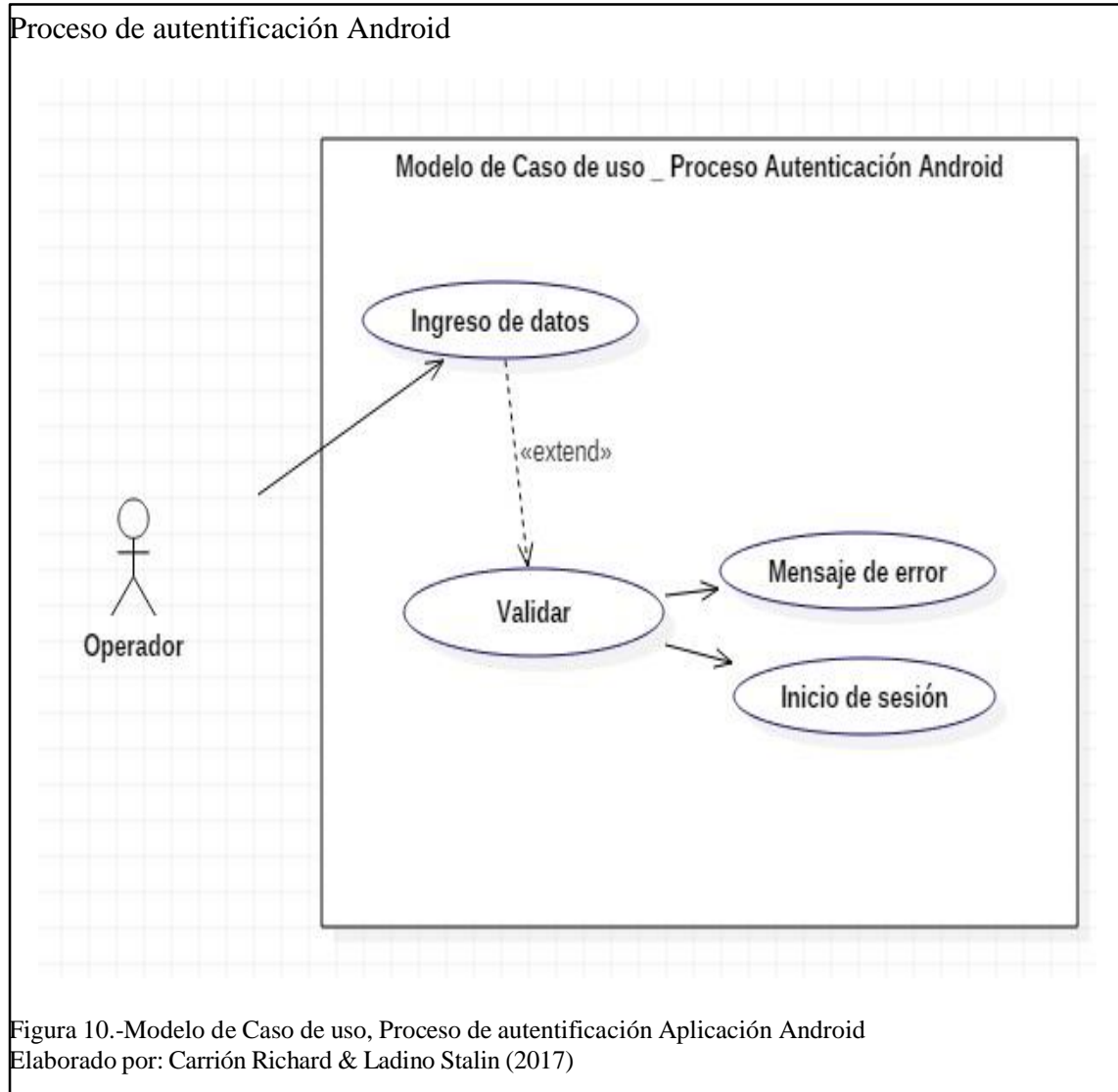


Tabla 9.-Escenario de Caso de Uso, Proceso de Autenticación Aplicación Android.

ACTORES	
Operador	
PRECONDICIONES	El usuario con rol “operador” debe estar registrado en el sistema.
POSCONDICIONES	El usuario ingresará al sistema solo si dispone del rol “operador”.
FLUJO BÁSICO	<p>El flujo básico debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • El usuario ingresa al sistema con el usuario y contraseña asignado al momento de la creación. • Se verifica que los datos estén correctos y se procede con el ingreso respectivo a la base de datos.
FLUJO ALTERNATIVO	<p>El flujo alternativo debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Si los datos son incorrectos se mostrará un mensaje de notificación informando de que los datos son erróneos.

Nota: Se presenta el escenario para la autenticación del usuario que realizará las lecturas de los medidores.
Elaborado por: Carrión Richard & Ladino Stalin

2.2.3.1 Diagrama de la Gestión de Lecturas

2.2.3.1.1 Diagrama de ingreso de lecturas

A continuación se muestra el proceso que se sigue para el ingreso de lecturas de cada uno de los medidores, para el posterior proceso de facturación para cada uno de los usuarios.

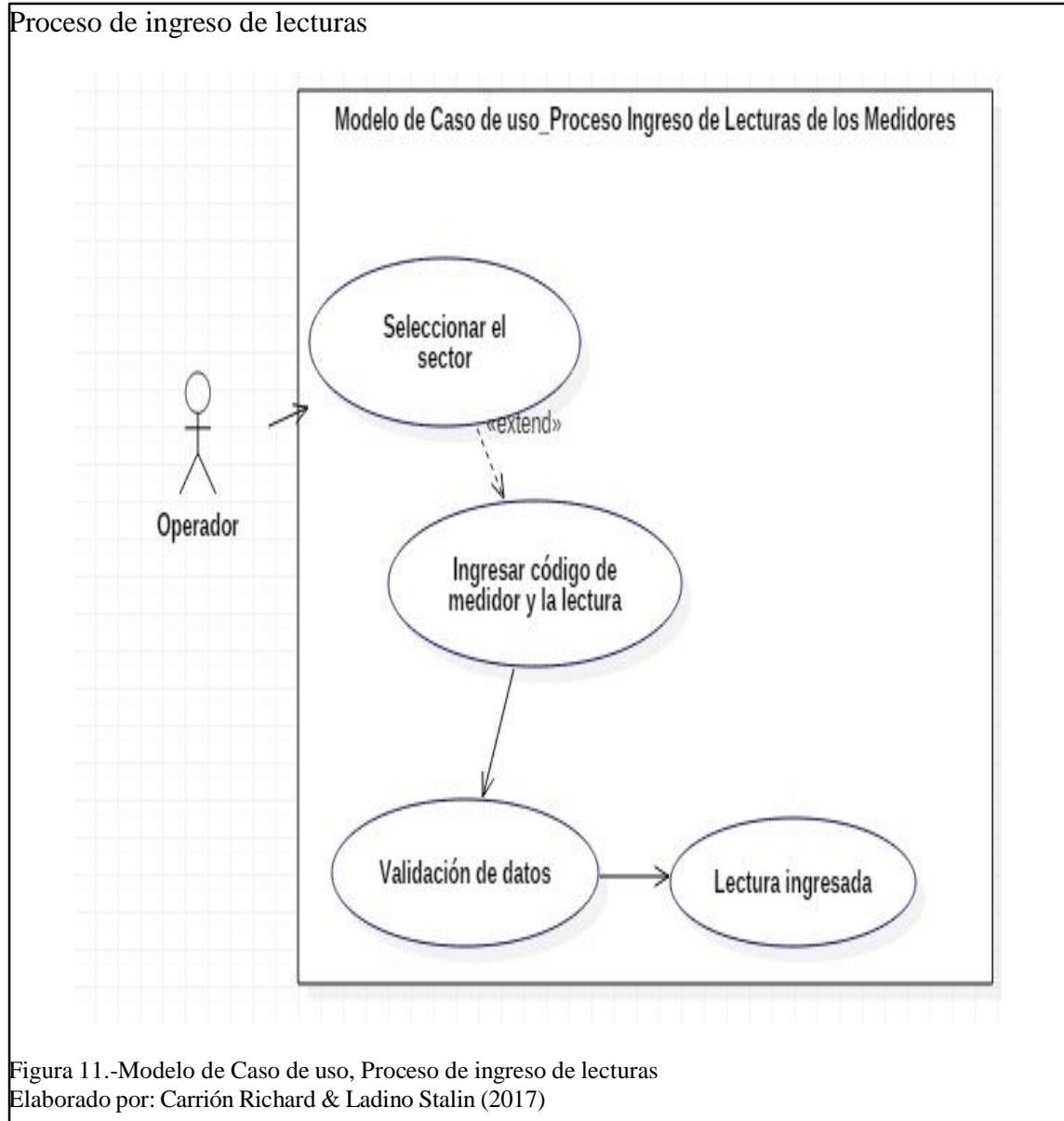


Tabla 10.-Escenario de Caso de Uso, Proceso de Ingreso de Lecturas

ACTORES	Operador
PRECONDICIONES	El usuario con rol “operador” debe estar autenticado.
POSCONDICIONES	El operador ingresará los datos de acuerdo a la lectura del medidor.
FLUJO BÁSICO	<p>El flujo básico debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • El usuario seleccionará el sector al cual va asignado el medidor. • Se ingresa los dos datos esenciales como es el código del medidor y la respectiva lectura. • Se verifica que los datos estén correctos y se procede con el ingreso respectivo a la base de datos.
FLUJO ALTERNATIVO	<p>El flujo alternativo debe realizarse de la siguiente forma:</p> <ul style="list-style-type: none"> • Si los datos son incorrectos se mostrará un mensaje de notificación informando de que los datos son erróneos.

Nota: Se presenta el escenario para la lectura del consumo de agua potable de cada uno de los medidores.
Elaborado por: Carrión Richard & Ladino Stalin

Esto en cuanto a los Casos de Uso que tienen mayor relevancia en nuestra Aplicación los restantes se encuentran desarrollados en el Anexo 3.

2.3 Diagrama de base de datos

Para establecer el diagrama de la base de datos a utilizar y el modelado de los datos, como de las entidades más relevantes se optó por el modelo entidad relación el cuál lo se pudo observar en la figura que se encuentra a continuación.

Diagrama de base de datos

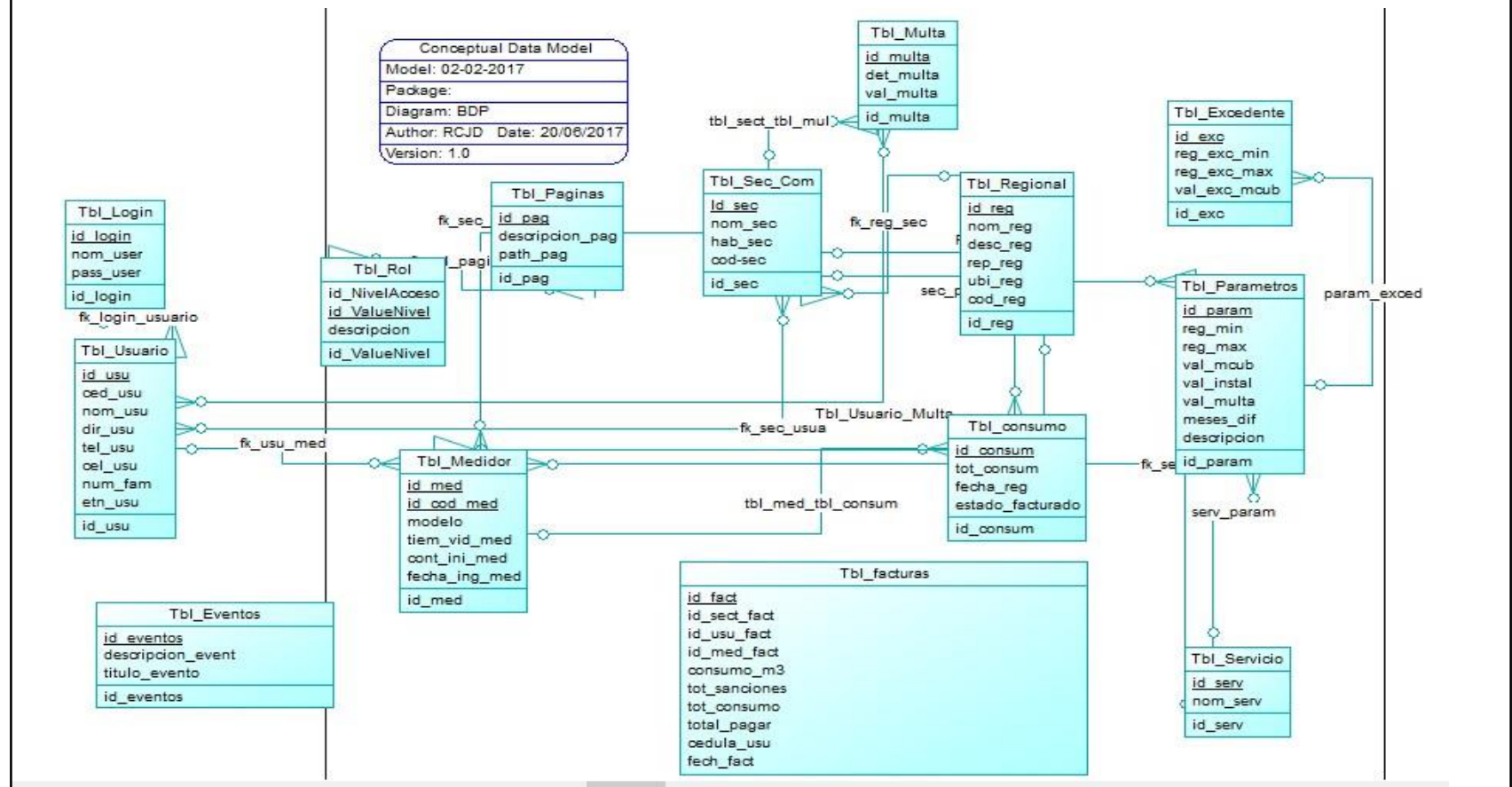


Figura 12.- Versión 3.0 de la base de datos

Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la figura anterior se muestra el diagrama de la base de datos final, el cual después de dos versiones anteriores se pudo llegar a reestructurar para obtener esta versión la cual cumple con los requerimientos necesarios para que nuestra aplicación pueda funcionar de manera óptima.

La primera versión se dio en base a los primeros requerimientos que fueron obtenidos a partir del primer prototipo generado y con el cual se pudo llevar a cabo los primeros Sprints del sistema y que con la retroalimentación y las necesidades se fueron reestructurando para llegar a dar la segunda versión.

La Segunda versión se generó a partir de la necesidad de reestructurar ciertos aspectos relacionados con la distribución de los usuarios dentro de los diferentes sectores.

Y esta versión final se reestructuro para dar solución a las diferentes necesidades e inconvenientes finales. Necesidades que involucraban procesos como el tema relacionado con la facturación. Las diferentes reuniones permitieron refinar la base de datos a tal punto que llegue a cumplir todas las expectativas.

2.4 Arquitectura del sistema

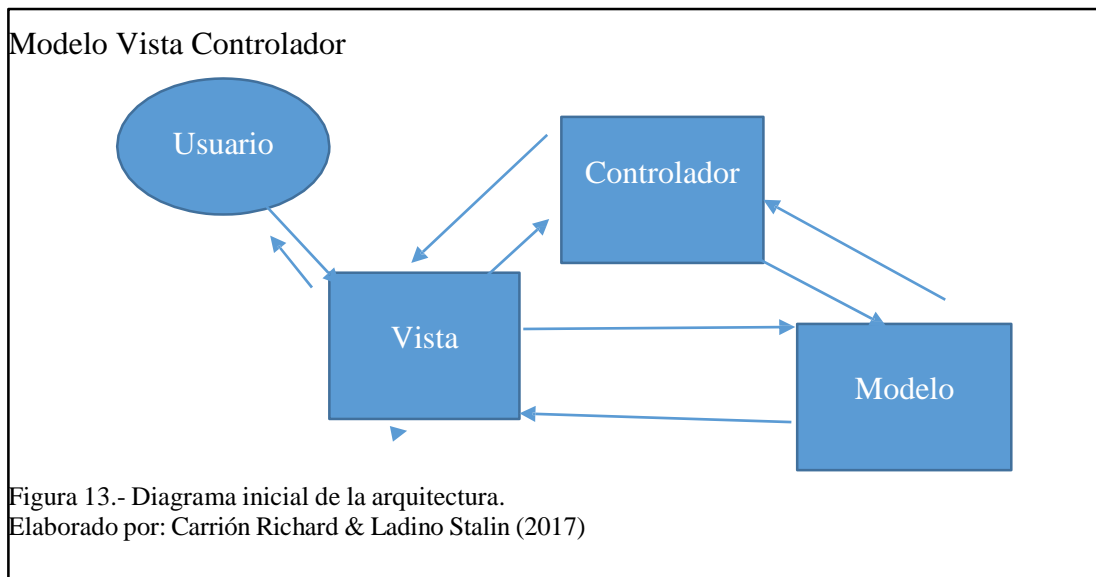
Desde un punto en el cual se inició el proyecto se optó por la arquitectura MVC (Modelo Vista controlador), primeramente porque esta arquitectura facilita el desarrollo de la aplicación con ventajas como la reutilización del código, estructuración correcta de los componentes, escalabilidad del negocio, robustez, etc.

Esta arquitectura distribuye en tres capas a la aplicación:

La primera capa procesa los datos ingresados por el usuario y realiza las diferentes peticiones a la base de datos.

La segunda capa va orientada a la visualización de las interfaces de usuario.

La tercera capa es la encargada de tener el código suficiente para responder a las acciones que se solicitan en la aplicación.



Capítulo 3

Construcción y pruebas

Las Juntas Comunitarias de la provincia de Imbabura y Pichincha disponen como representantes y usuarios a personas con un conocimiento tecnológico de todo nivel, por esta razón este sistema ha sido diseñado con una interfaz muy amigable e intuitiva, para que el usuario pueda tener una navegación sin ningún inconveniente.

Todas las pantallas disponen de un mismo diseño para facilitar la interacción, además dispone de colores que son muy llamativos y que facilitan la visibilidad del usuario.

Se utilizó una arquitectura muy fuerte y muy robusta como es MVC, el cual describe un conjunto de patrones y estándares que permiten una programación muy ordenada separando los datos, la lógica de negocio y la interfaz presentada al usuario. Esto al final ayudó a facilitar el desarrollo y el control de los diferentes cambios.

3.1 Arquitectura

La arquitectura utilizada en el presente sistema como se mencionó anteriormente es el MVC (Modelo-Vista-Controlador), este modelo permite llevar a cabo todas las funciones de manera ordenada tal es el caso que se puede describir el flujo que lleva a cabo el sistema en el gráfico presentado a continuación.

Modelo Vista Controlador

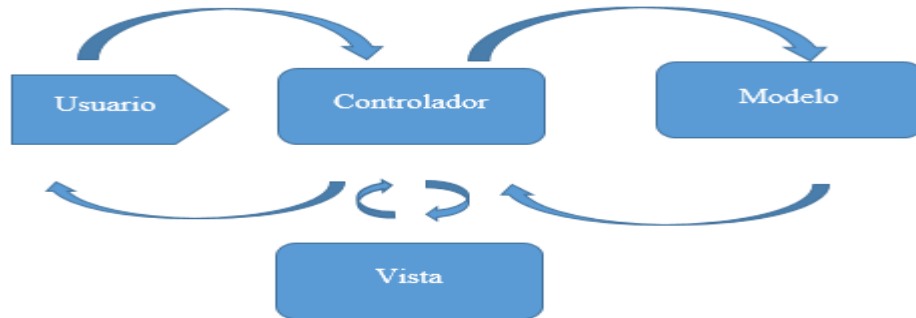


Figura 14.- Diagrama de la arquitectura
Elaborado por: Carrión Richard & Ladino Stalin (2017)

El flujo del gráfico anterior se puede describir en 5 pasos los cuales describen el proceso de nuestra aplicación:

El usuario final interactúa con la interfaz de usuario de alguna manera (Ya sea solicitando el ingreso de un nuevo usuario, regional, etc.).

El controlador recibe la petición que realizó el usuario y gestiona el evento que ingresa.

El controlador ingresa al modelo y gestiona todas las reglas necesarias para el control de los eventos. Llama a los métodos necesarios (ingresar_usuario).

El controlador designa los objetos de la vista a ser mostrados en la interfaz del usuario.

La vista espera más interacciones con el usuario.

3.2 Código Relevante

Describe las partes más importantes del código desarrollado en la implementación del sistema.

3.2.1 Clase: Conexión.

Permite realizar la conexión del servidor de aplicaciones con la aplicación.

En este caso se usó un archivo de propiedades “xml”, identificado en el presente proyecto como “persistence.xml” “el mismo que tiene la siguiente estructura.

Clase Conexión

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1"
  xmlns="http://xmlns.jcp.org/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_1.xsd">
  <persistence-unit name="WebApplicationUPS-EJB" transaction-type="JTA">
    <jta-data-source>java:jboss/datasources/appFact</jta-data-source>
    <class>com.ec.ups.jpa.entidades.loginDTO</class>
    <class>com.ec.ups.jpa.entidades.TblMedidor</class>
    <class>com.ec.ups.jpa.entidades.TblPagina</class>
    <class>com.ec.ups.jpa.entidades.TblRegional</class>
    <class>com.ec.ups.jpa.entidades.rolDTO</class>
    <class>com.ec.ups.jpa.entidades.secComDTO</class>
    <class>com.ec.ups.jpa.entidades.TblServicio</class>
    <class>com.ec.ups.jpa.entidades.usuarioDTO</class>
    <class>com.ec.ups.jpa.entidades.FkRolPagina</class>
    <class>com.ec.ups.jpa.entidades.FkRolPaginaPK</class>
    <class>com.ec.ups.jpa.entidades.TblParametro</class>
    <class>com.ec.ups.jpa.entidades.TblExcedente1</class>
    <class>com.ec.ups.jpa.entidades.TblMulta</class>
    <class>com.ec.ups.jpa.entidades.TblExcedente</class>
    <class>com.ec.ups.jpa.entidades.TblConsumo</class>
    <class>com.ec.ups.jpa.entidades.TblFactura</class>
    <class>com.ec.ups.jpa.entidades.ResultadoFactura</class>
    <class>com.ec.ups.jpa.entidades.TblUsuarioMulta</class>
    <class>com.ec.ups.jpa.entidades.TblSectorUsuario</class>
```

Figura 15.- Archivo xml – Clase Conexión

Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la imagen se pudo identificar el datasource el cual posee la siguiente dirección “java:jboss/datasources/appFact”, la cual identifica la dirección en la cual se encuentra el archivo de configuración dentro de la aplicación que identifica el servidor de aplicaciones

el mismo que debe tener la misma estructura.

Datasource

```
<datasource jta="false" jndi-name="java:jboss/datasources/appFact" pool-name="appFact" enabled="true" use-ccm="false">
  <connection-url>jdbc:postgresql://localhost:5432/facturacionRegional</connection-url>
  <driver-class>org.postgresql.Driver</driver-class>
  <driver>postgresql</driver>
  <pool>
    <min-pool-size>10</min-pool-size>
    <max-pool-size>100</max-pool-size>
    <prefill>true</prefill>
  </pool>
  <security>
    <user-name>postgres</user-name>
    <password>1234</password>
  </security>
  <validation>
    <validate-on-match>false</validate-on-match>
    <background-validation>false</background-validation>
  </validation>
  <statement>
    <share-prepared-statements>false</share-prepared-statements>
  </statement>
</datasource>
```

Figura 16.- Archivo de configuración – Datasource
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En el servidor de aplicaciones “Wildfly”, se tuvo el archivo standalone.xml el cual posee la conexión a la base de datos, así como la ruta en la cual se tuvo un archivo de configuración dentro del aplicativo web para poder realizar la conexión entre el sistema y la base de datos.

3.2.2 Módulo Login - Clase: loginControler.

Clase Login Controler

```
public void ingresar() throws IOException {
    loginDTO login = new loginDTO();
    loginDatamanager.setMapUserSec(new HashMap<>());
    loginDatamanager.setlistSectoros(new ArrayList<secComDTO>());
    login = loginDTOInterface.buscarLogin(
        loginDatamanager.getNombreUsuario(),
        loginDatamanager.getPassword());
    if (null == login) {
        JsUtil.addMessage(ConstantsUtil.MSJ_ERROR, MensajesUtil
            .getString("etiqueta.mensaje.credenciales.incorrectas"));
        return;
    } else {
        loginDatamanager.setSectorSelcUser(new secComDTO());
        loginDatamanager.setLoginObj(login);
        List<TblSectorUsuario> listSec = sectorUsuarioDTOInterface
            .buscarLogin(login.getIdLogin());
        if (!listSec.isEmpty()) {
            if (listSec.get(0).getIdValuenivel().equals(ConstantsUtil.NIVEL_SUPER_ADMIN)) {
                System.out.println("Administrador");
            } else {
                usuarioDTO user = usuarioDTOInterface.findByIdLogin(login.getIdLogin());
                for (TblSectorUsuario listUserSec : listSec) {
                    loginDatamanager.getlistSectoros().add(listUserSec.getSecComDTO());
                    loginDatamanager.getMapUserSec().put(listUserSec.getIdSec(), listUserSec);
                }
                if (null != user) {
                    loginDatamanager.setUsuarioSession(user);
                }
                FacesContext fc = FacesContext.getCurrentInstance();
                fc.getExternalContext().redirect(
                    "/SistemaFacturacionUPS/page/menu.jsf");
            } else {
                loginDatamanager.getLoginObj().setIdValuenivel(ConstantsUtil.NIVEL_SUPER_ADMIN);
                FacesContext fc = FacesContext.getCurrentInstance();
                fc.getExternalContext().redirect(
                    "/SistemaFacturacionUPS/page/menu.jsf");
            }
        }
    }
}
```

Figura 17.- Clase: LoginControler – Módulo Loguin
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Método que verifica el ingreso al sistema si es usuario con rol “Super administrador”, permite visualizar la información de todas las regionales con sus respectivos sectores, caso

contrario muestra un combo cargando los sectores a los cuales el usuario se encuentra asociado restringiendo la visualización de información a la cual no debe tener acceso.

3.2.3 Módulo Usuario - Clase: usuarioController.

Clase Usuario Controller

```
public void guardar() {  
    if (usuarioDatamanager.getAccion().equals(ConstantsUtil.ACCION_NUEVO)) {  
        if (!usuarioDatamanager.getExistUser()) {  
            if (validarContrasena()) {  
                return;  
            }  
        }  
        if (null != usuarioDatamanager.getContrasena()) {  
            usuarioDatamanager.getUsuarioFormul().getLoginDTO()  
                .setPassUser(usuarioDatamanager.getContrasena());  
        }  
        if (!usuarioDatamanager.getExistUser()) {  
            loginDTOInterface.create(usuarioDatamanager.getUsuarioFormul()  
                .getLoginDTO());  
            usuarioDatamanager.getUsuarioFormul().setIdLogin(  
                usuarioDatamanager.getUsuarioFormul().getLoginDTO()  
                    .getIdLogin());  
            usuarioDatamanager.getTblSectorUsuarioFormul().setIdLogin(  
                usuarioDatamanager.getUsuarioFormul().getLoginDTO()  
                    .getIdLogin());  
        }  
    }  
}
```

Figura 18.- Clase: usuarioController – Módulo Usuario
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Método para validar existencia de usuario si es nuevo ingreso Login con su respectiva contraseña y se almacena en la base de datos.

Método validación Login

```
    } else {  
        List<usuarioDTO> list = usuarioDTOInterface  
            .buscarByCedula(usuarioDatamanager.getUsuarioFormul()  
                .getCedUsu());  
  
        List<TblSectorUsuario> listSecUse = sectorUsuarioDTOInterface  
            .buscarPorCedula(usuarioDatamanager.getUsuarioFormul()  
                .getCedUsu());  
        for (TblSectorUsuario secUser : listSecUse) {  
            if (secUser.getIdSec().equals(  
                usuarioDatamanager.getTblSectorUsuarioFormul()  
                    .getIdSec())) {  
                JsUtil.addMessage(ConstantsUtil.MSJ_INFO,  
                    "El usuario ya existe en el sector seleccionado");  
                return;  
            }  
        }  
        usuarioDatamanager.getUsuarioFormul().setIdLogin(  
            usuarioDatamanager.getExistUserObj().getIdLogin());  
  
        usuarioDatamanager.getTblSectorUsuarioFormul().setIdLogin(  
            usuarioDatamanager.getUsuarioFormul().getLoginDTO()  
                .getIdLogin());  
    }  
}
```

Figura 19.- Método: Validación Login – Módulo Usuario
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En caso de ser un usuario existente que se ingresa en un nuevo sector se restringe el ingreso de un nuevo login permitiendo al usuario acceder a los sectores a los cuales se encuentra asociado con la misma contraseña y login previamente registrados.

3.2.4 Módulo Parametrización - Clase: parametrizacionController.

Clase parametrizacionController

```
public void guardar() {  
  
    if (parametrizacionDatamanager.getAccion().equals(  
        ConstantesUtil.ACCION_NUEVO)) {  
  
        parametroDTOInterface.create(parametrizacionDatamanager  
            .getParametroObjs());  
        parametrizacionDatamanager  
            .setVerPantalla(ConstantesUtil.PANTALLA_CONSULTAR);  
  
        for (TblExcedente list : parametrizacionDatamanager  
            .getExcedenteList()) {  
            list.setIdParam(parametrizacionDatamanager.getParametroObjs()  
                .getIdParam());  
            excedenteDTOInterface.create(list);  
        }  
  
        JsfUtil.addMessage(ConstantesUtil.MSJ_INFO,  
            "Parámetro registrado con éxito");  
  
    } else if (parametrizacionDatamanager.getAccion().equals(  
        ConstantesUtil.ACCION_MODIFICAR)) {  
        parametroDTOInterface.edit(parametrizacionDatamanager  
            .getParametroObjs());  
        parametrizacionDatamanager  
            .setVerPantalla(ConstantesUtil.PANTALLA_CONSULTAR);  
        JsfUtil.addMessage(ConstantesUtil.MSJ_INFO,  
            "Parámetro modificado con éxito");  
    }  
    if (loginDatamanager.getLoginObj().getIdValuenivel() != ConstantesUtil.NIVEL_SUPER_ADMI) {  
  
        parametrizacionDatamanager.setListParametros(parametroDTOInterface  
            .buscarParamBySect(loginDatamanager.getSectorSeleSec()  
                .getIdSec()));  
    } else {  
        parametrizacionDatamanager.setListParametros(parametroDTOInterface  
            .buscarParamSec());  
    }  
    parametrizacionDatamanager.setExcedenteFormul(new TblExcedente());  
}
```

Figura 20.- Clase: parametrizaciónController – Módulo Parametrización
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Método para validar la parametrización de cada sector con sus respectivos valores mínimos y máximos, así como el valor del excedente.

Clase parametrizacionController

```
public Boolean calcularValor() {

    List<TblParametro> parametroList = parametroDTOInterface
        .buscarParamBySectServ(facturacionDatamanager
            .getResultadoSelect().getId_sec(),
            facturacionDatamanager.getResultadoSelect()
                .getId_serv());

    Double resp = 0.0;
    Double totalAPagarConsumo=0.0;
    Double totalMultasConsumo = 0.0;

    if (parametroList.size() > 1) {
        JsfUtil.addMessage(ConstantsUtil.MSJ_INFO,
            "Revise tabla de parámetros existen dos"
                + " valores para el servicio seleccionado.");
        return Boolean.FALSE;
    } else {
        if (parametroList.isEmpty()) {
            JsfUtil.addMessage(
                ConstantsUtil.MSJ_INFO,
                "Revise tabla de parámetros no existen valores "
                    + "para realizar el calculo de su valor a cancelar.");
            return Boolean.FALSE;
        }
    }
}
```

Figura 21.- Clase: parametrizaciónController – Módulo Parametrización
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Método para calcular los cálculos de la factura validando la tabla de parametrización por sector realizando validación de existencia de valor para poder realizar el cálculo de la factura así como la existencia de un solo valor para el tipo de servicio a facturar.

Método Consumo Total

```
TblParametro parametro = parametroList.get(0);
if (facturacionDatamanager.getResultadoSelect().getTot_consum() >= parametro
    .getRegMin()
    && facturacionDatamanager.getResultadoSelect()
        .getTot_consum() <= parametro.getRegMax()) {

    resp=parametro.getValMcub();
    totalAPagarConsumo=parametro.getValMcub();

    facturacionDatamanager.setValorPagarMedidor(this.Redondear(resp));
    facturacionDatamanager.setValorPagarExcedente(0.0);

} else {
    resp = formulaFacturacionExc(facturacionDatamanager
        .getResultadoSelect().getTot_consum(),
        parametro.getIdParam());
    facturacionDatamanager.setValorPagarMedidor(this.Redondear(parametro.getValMcub()));
    totalAPagarConsumo=resp+parametro.getValMcub();

}
facturacionDatamanager.setTarifaMedidor(parametro.getValMcub().toString());
facturacionDatamanager.setValorMul(calculoMultas());

facturacionDatamanager.setValorPagar(this.Redondear(totalAPagarConsumo));
totalMultasConsumo = totalAPagarConsumo + calculoMultas();
facturacionDatamanager.setValorTotal(this.Redondear(totalMultasConsumo));
return Boolean.TRUE;
}
```

Figura 22.- Método: Consumo Total – Módulo Parametrización
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Método para validar si el total del consumo se encuentra en los valores base o si se genera el cálculo del excedente por sector.

3.2.5 Módulo Reportes - Clase: reportesController.

Clase reportesController

```
public void generarFactura() throws Exception{

    String url =ConstantesUtil.CADENA_VACIA;
    usuarioDTO user=new usuarioDTO();

    Map<String, Object> parametros = new HashMap<String, Object>();
    parametros.put(ReportServlet.TIPO_REPORT, ReportServlet.REPORTE_PDF);
    parametros.put(ReportServlet.TIPO_SELECCIONADO, ReportServlet.TIPO_FACTURACION);
    parametros.put(ReportServlet.CONEXION, "si");
    parametros.put("fechDesde", reportesDatamanager.getFechaIni());
    parametros.put("fechHasta", reportesDatamanager.getFechaFin());
    parametros.put("idSector", reportesDatamanager.getSecComSelect().getIdSec());
    parametros.put("nomSector", secComDTOInterface.findById(reportesDatamanager.getSecComSelect().getIdSec()).getNomSec());
    if(!reportesDatamanager.getCedula().equals(ConstantesUtil.CADENA_VACIA)){
        user=usuarioDTOInterface.findByCedula(reportesDatamanager.getCedula());
    } else {
        if(reportesDatamanager.getTipReporte()==1){
            JsFUtil.addMessage(ConstantesUtil.MSJ_INFO, "Número de cédula es requerido para reporte detallado");
            return;
        }
    }

    if(reportesDatamanager.getTipReporte()==1){
        if(null!=user){
            parametros.put("cedula", reportesDatamanager.getCedula());
            parametros.put("nombres", user.getNomUsu());
            url = JsFUtil.getContextPath()+ "/reportServlet?reporte=reporteFacturacionDetallado";
        } else {
            JsFUtil.addMessage(ConstantesUtil.MSJ_INFO, "Número de cédula no registrado.");
            return;
        }
    }
}
```

Figura 23.- Clase: reportesCotroller – Módulo Reportes
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Método para generar reporte, validando por tipo de reporte los valores requeridos, así como la existencia del usuario del cual se va realizar el reporte detallado.

3.3 Pruebas

3.3.1 Ambiente para Pruebas.

Para el proceso de pruebas se preparó un ambiente el cual constaba de las siguientes características:

Servidor en la Nube contratado con la Empresa “**Amazon**”

Plan: t2.small.

Número de Cpu: 1.

Ram: 1 gb.

Espacio en disco: 8 Gb.

Créditos por hora: 6

El ambiente se levantó en una máquina virtual con sistema operativo Ubuntu (Linux), se instaló un gestor de base de datos “Postgresql 9.3”, Servidor de aplicaciones Java EE “WildFly”.

3.3.2 Pruebas.

Las pruebas que se ejecutaron en el sistema permitieron asegurar la calidad del producto y por tanto asegurar la satisfacción del cliente, verificar que todos los requerimientos expuestos en el alcance se cumplan, conocer el comportamiento del sistema frente a

diferentes circunstancias como es el rendimiento con la simulación de una situación extrema, corrección de errores, etc.

3.3.2.1 Pruebas funcionales.

Las pruebas funcionales permiten verificar que los requerimientos expuestos en el alcance se cumplan de manera satisfactoria. El comportamiento y la funcionalidad de cada una de las pantallas deben darnos los resultados esperados. Estas pruebas permiten encontrar errores que muchas veces no se pueden observar de manera simple.

Para comprobar que el funcionamiento sea el esperado se pudo observar que cada una de las tareas que realice este sistema cumpla con cada una de las actividades que se recopiló durante la fase del análisis y diseño.

Estas pruebas se las pudo realizar en la etapa de la revisión de los últimos sprints, luego de corregir los errores revisados en las reuniones anteriores.

Ya realizadas las diferentes pruebas funcionales se pudo obtener los resultados descritos en la siguiente tabla.

Tabla 11.-Resultados pruebas funcionales

Satisfactoria				
Código Prueba	Módulo	Si	No	Observación
P001	Loguin	X		Ninguna
P002	Gestión de Usuarios	X		Se debe añadir la identidad étnica ninguna en el caso que se facture a nombre de una institución

P003	Gestión de Regionales	X		Ninguna
P004	Gestión de Sectores	X		Ninguna
P005	Gestión de Medidores	X		Se debe redefinir la restricción que cada usuario solo debe tener un medidor
P006	Gestión de Multas y Sanciones	X		Ninguna
P007	Gestión de la Cuenta	X		Ninguna
P008	Gestón de Parámetros	X		Ninguna
P009	Facturación	X		Ninguna
P010	Gestión de Reportes	X		Ninguna
P011	Gestión de la aplicación móvil	X		Ninguna

Nota: Se presenta los resultados obtenidos durante las pruebas funcionales.

Elaborado por: Carrión Richard & Ladino Stalin

Luego de realizar las diferentes pruebas funcionales y de las pequeñas observaciones que tuvieron los beneficiarios al momento del uso del sistema se pudo determinar que en un 95 % tuvo la aceptación de los mismos.

Por lo cual se pudo decir que se obtuvo la satisfacción del usuario y que se cumple con todos los requerimientos solicitados en el alcance, que los pequeños cambios que se tuvieron que corregir no afectaron ni a la base, ni a la arquitectura de nuestra aplicación. Además se pudo tomar en cuenta que estos pequeños cambios de dieron ya que ciertos

sectores manejan de diferente manera el proceso y por tanto la aplicación debe contemplar estas situaciones.

3.3.2.2 Pruebas de Rendimiento.

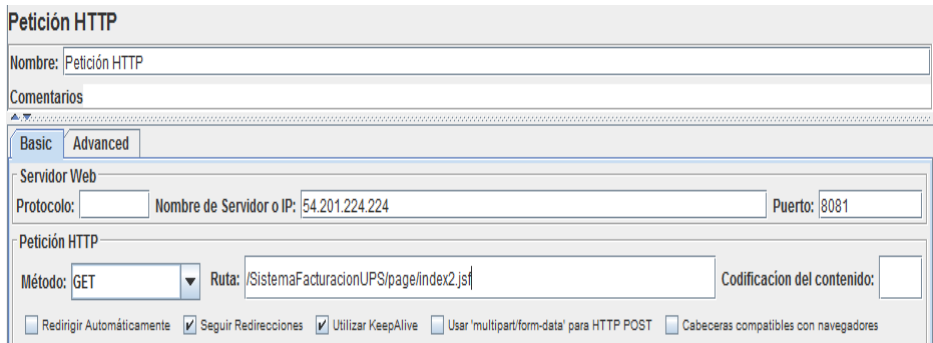
Para realizar las pruebas de rendimiento se optó por utilizar la herramienta “JMeter”, la cual permite ver el comportamiento del sistema frente a diferentes situaciones como es con un número promedio de usuarios durante un tiempo determinado.

Otra de las ventajas por las cuales se optó por esta herramienta es porque despliega diferentes tipos de reportes las cuales permiten interpretar de mejor manera los resultados obtenidos.

Para la preparación del ambiente para este tipo de pruebas se debe configurar el entorno de la petición http y el entorno de grupo de hilos en el cual se definió la muestra a utilizar para nuestra prueba.

En la siguiente figura se muestra la configuración de la petición http en donde se colocó el servidor y el puerto de nuestra aplicación.

Configuración Petición http



The image shows the 'Petición HTTP' (HTTP Request) configuration window in JMeter. It has a 'Basic' tab selected. The 'Nombre' (Name) field is 'Petición HTTP'. The 'Comentarios' (Comments) field is empty. Under the 'Servidor Web' (Web Server) section, the 'Protocolo' (Protocol) is set to 'http', 'Nombre de Servidor o IP' (Server Name or IP) is '54.201.224.224', and 'Puerto' (Port) is '8081'. Under the 'Petición HTTP' (HTTP Request) section, the 'Método' (Method) is 'GET', 'Ruta' (Path) is '/SistemaFacturacionUPS/page/index2.jsf', and 'Codificación del contenido' (Content Encoding) is empty. At the bottom, there are several checkboxes: 'Redirigir Automáticamente' (Redirect Automatically) is unchecked, 'Seguir Redirecciones' (Follow Redirects) is checked, 'Utilizar KeepAlive' (Use KeepAlive) is checked, 'Usar multipart/form-data para HTTP POST' (Use multipart/form-data for HTTP POST) is unchecked, and 'Cabeceras compatibles con navegadores' (Browser-compatible headers) is unchecked.

58

Figura 24.-Configuración petición http, Prueba de rendimiento Jmeter
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Como siguiente paso se debió configurar la pantalla de grupo de hilos, donde se definió la muestra y el número de segundos en los cuales se verificó el comportamiento de nuestra aplicación.

Configuración Grupo de hilos

Nombre: Grupo de Hilos

Comentarios

Acción a tomar después de un error de Muestreador

☒ Continuar ☐ Comenzar siguiente iteración ☐ Parar Hilo ☐ Parar Test ☐ Parar test ahora

Propiedades de Hilo

Número de Hilos: 100

Periodo de Subida (en segundos): 5

Contador del bucle: ☐ Sin fin 1

☐ Retrasar la creación de Hilos hasta que se necesiten

☐ Planificador

Configuración del Planificador

Duración (segundos)

Retardo de arranque (segundos)

Tiempo de Arranque: 2017/06/11 13:51:12

Tiempo de Finalización: 2017/06/11 13:51:12

Figura 25.-Configuración grupo de hilos, Prueba de rendimiento Jmeter
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Luego de tener configurado estos parámetros se procedió a realizar las pruebas en donde se obtuvo:

Árbol de resultados.- Muestra las peticiones que se realizan correcta o incorrectamente.

Gráfico de resultados.- Muestra de manera gráfica los resultados.

Informe agregado.- Muestra los resultados con mayos detalle.

Esta prueba se realizó para dos casos y para dos diferentes módulos los cuales son lo más concurrentes dentro de nuestra aplicación como son el módulo de autenticación (Loguin)

y el módulo de Facturación.

Caso 1. Orientado para un promedio de 100 usuarios en un promedio de 5 segundos.

Módulo de autenticación

Árbol de resultados

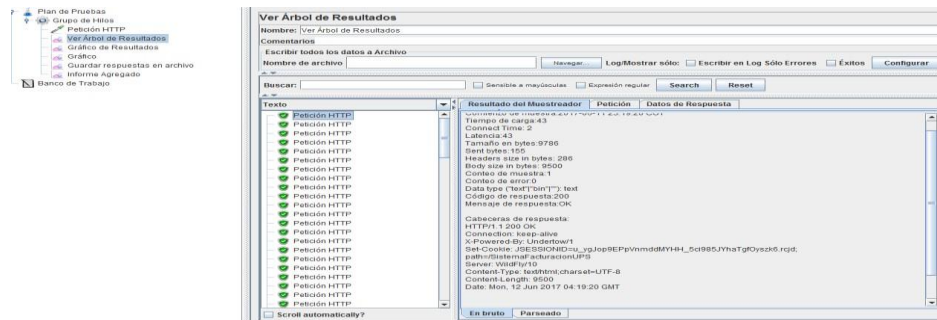


Figura 26.- Resultados árbol de resultados, Módulo de autenticación, Caso 1
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo observar que ninguna de las peticiones tuvo inconvenientes.

Gráfico de resultados

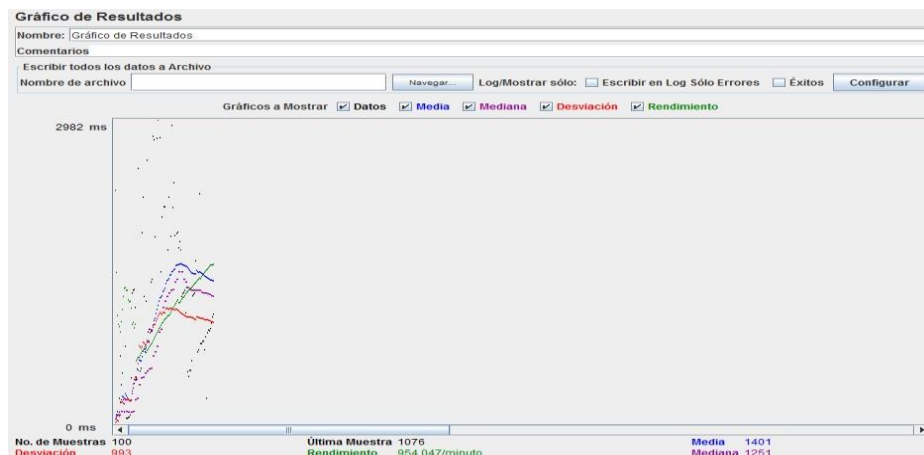


Figura 27.- Resultados gráfico de resultados, Módulo de autenticación, Caso 1
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo ver el comportamiento del sistema frente a la situación propuesta.

Informe agregado

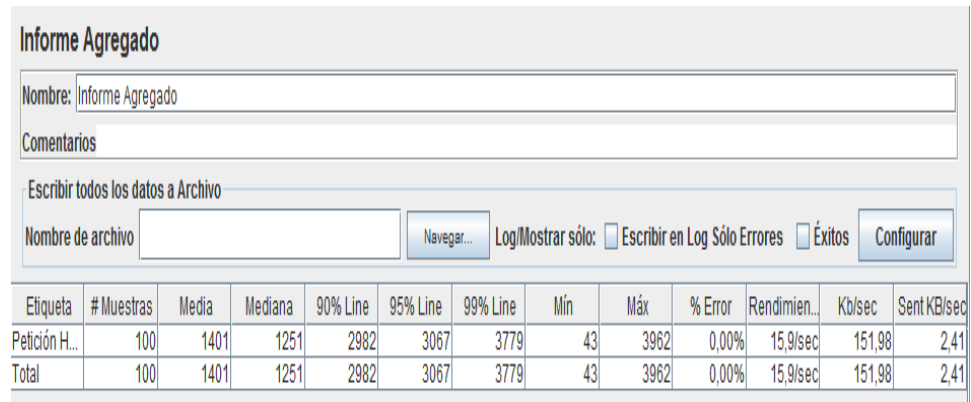


Figura 28.- Resultados informe agregado, Módulo de autenticación, Caso 1
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo observar los resultados obtenidos con detalles específicos.

Módulo de facturación

Árbol de resultados

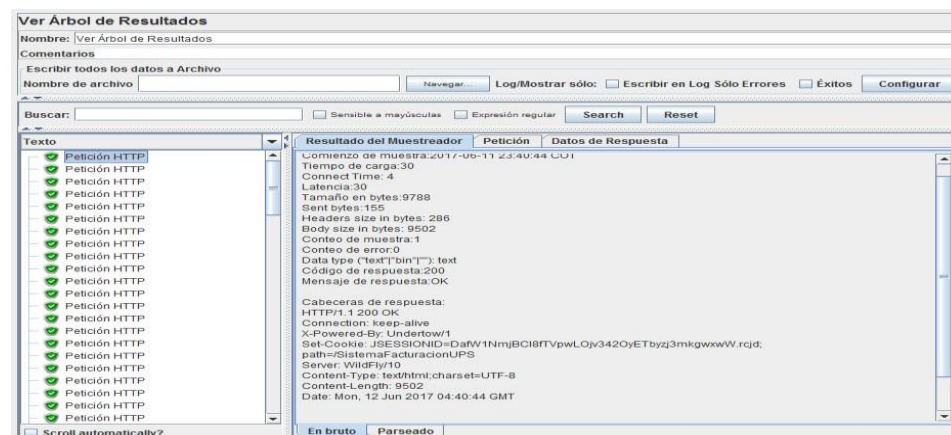


Figura 29.- Resultados árbol de resultados, Módulo de facturación, Caso 1
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo observar que ninguna de las peticiones tuvo inconvenientes

Gráfico de resultados

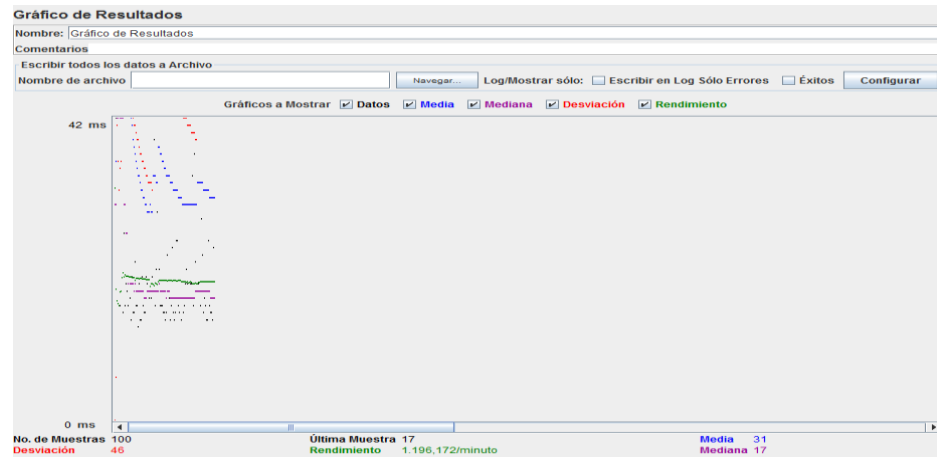


Figura 30.- Resultados gráfico de resultados, Módulo de facturación, Caso 1
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo ver el comportamiento del sistema frente a la situación propuesta.

Informe agregado

Informe Agregado												
Nombre: Informe Agregado												
Comentarios												
Escribir todos los datos a Archivo												
Nombre de archivo		Navegar...		Log/Mostrar sólo:		<input type="checkbox"/> Escribir en Log Sólo Errores		<input type="checkbox"/> Éxitos		Configurar		
Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimien...	Kb/sec	Sent KB/sec
Petición H...	100	31	17	43	81	235	13	327	0,00%	19,9/sec	190,55	3,02
Total	100	31	17	43	81	235	13	327	0,00%	19,9/sec	190,55	3,02

Figura 31.- Resultados informe agregado, Módulo de facturación, Caso 1
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo observar los resultados obtenidos con detalles específicos.

Caso 2. Orientado para un promedio de 1000 usuarios en un promedio de 10 segundos.

Módulo de autenticación

Árbol de resultados

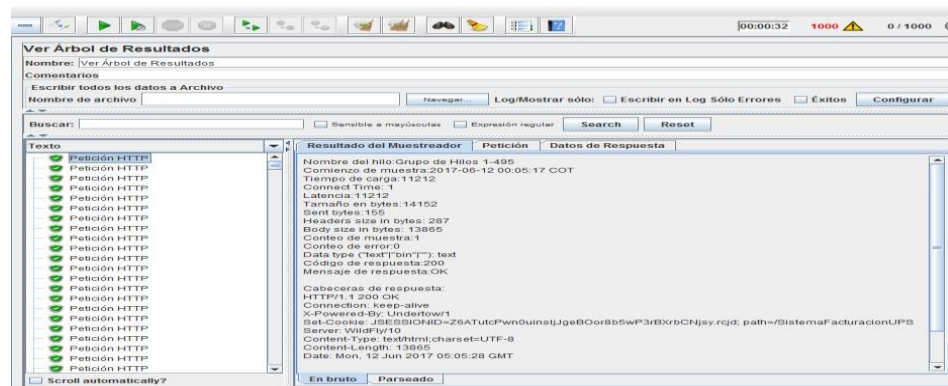


Figura 32.- Resultados árbol de resultados, Módulo de autenticación, Caso 2
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo observar que ninguna de las peticiones tuvo inconvenientes

Gráfico de resultados

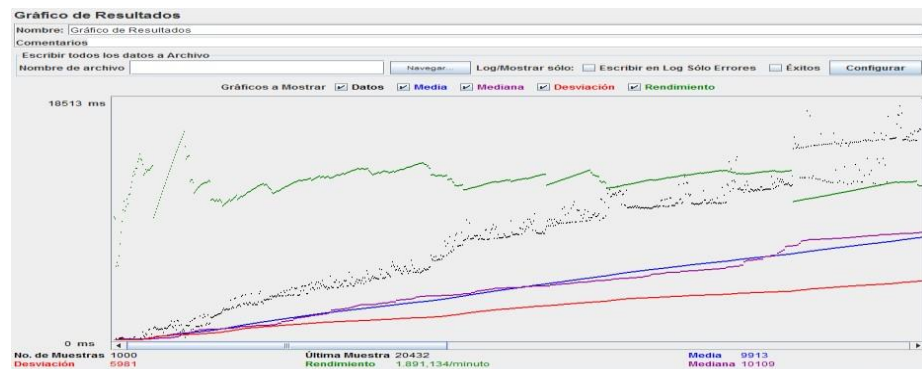


Figura 33.- Resultados gráfico de resultados, Módulo de autenticación, Caso 2
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo ver el comportamiento del sistema frente a la situación propuesta.

Informe agregado

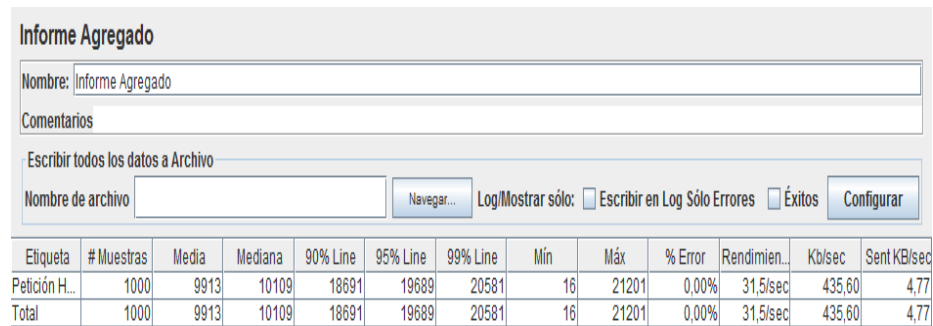


Figura 34.- Resultados informe agregado, Módulo de autenticación, Caso 2
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo observar los resultados obtenidos con detalles específicos.

Módulo de facturación

Árbol de resultados

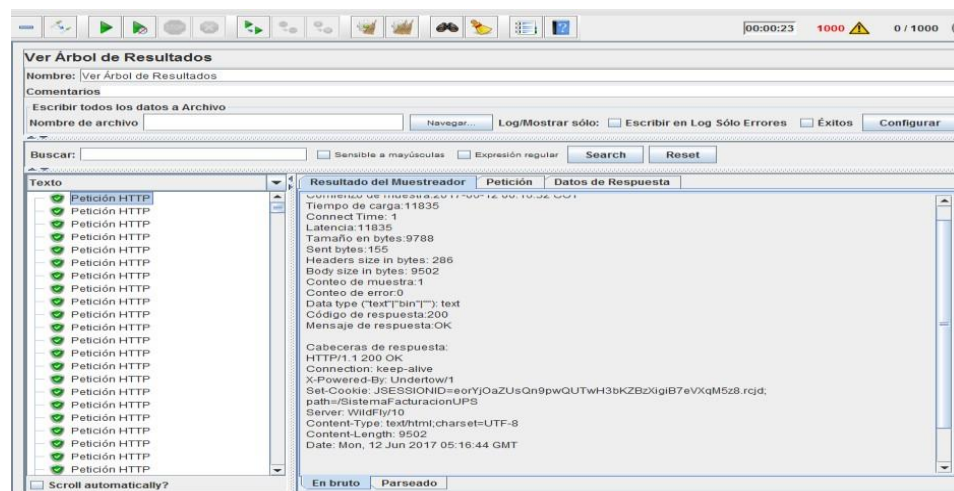


Figura 35.- Resultados árbol de resultados, Módulo de facturación, Caso 1
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo observar que ninguna de las peticiones tuvo inconvenientes.

Gráfico de resultados

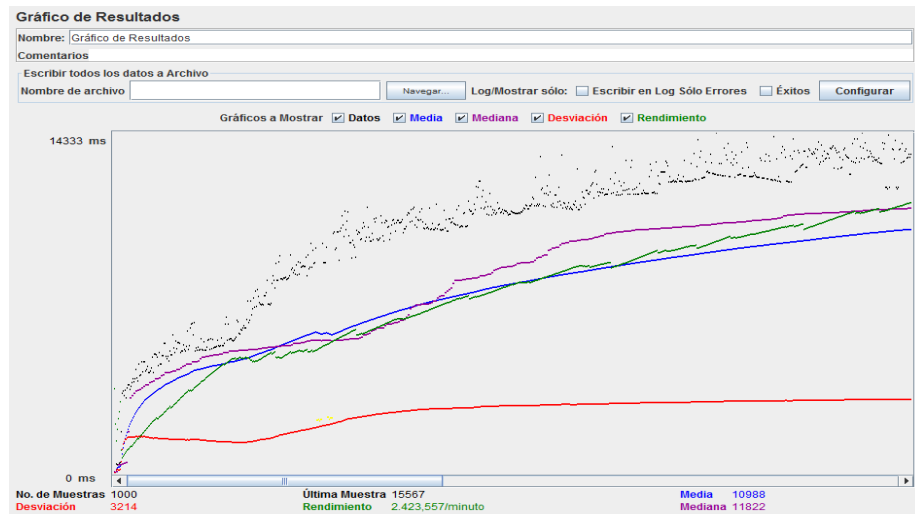


Figura 36.- Resultados gráfico de resultados, Módulo de facturación, Caso 2
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo ver el comportamiento del sistema frente a la situación propuesta.

Informe agregado

Informe Agregado

Nombre: Informe Agregado

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo: Navegar...

Log/Mostrar sólo: ☐ Escribir en Log Sólo Errores ☐ Éxitos

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimien...	Kb/sec	Sent KB/sec
Petición H...	1000	10988	11822	14333	14477	15596	42	16601	1,20%	40,4/sec	382,46	6,04
Total	1000	10988	11822	14333	14477	15596	42	16601	1,20%	40,4/sec	382,46	6,04

Figura 37.- Resultados informe agregado, Módulo de facturación, Caso 2
Elaborado por: Carrión Richard & Ladino Stalin (2017)

En la gráfica se pudo observar los resultados obtenidos con detalles específicos.

Luego de haber observado los resultados y tomando en cuenta los valores promedio de los informes agregados se pudo decir que para el primer caso con el módulo de autenticación se tuvo un rendimiento de 15,9 s y un error del 0% y para el módulo de facturación 19,9 s con un error del 0%. Mientras que para el segundo caso para el módulo de autenticación se tuvo un rendimiento de 31,5 % con un error del 0 % y para el módulo de facturación del 40,4% con un error del 1,20 %.

Con estos valores se pudo llegar a la conclusión que el rendimiento esta entre unos muy buenos parámetros, estos valores concuerdan con la proyección estimada y se dan gracias a la tecnología utilizada en el desarrollo de nuestra aplicación.

Como otro punto a favor se pudo tener en cuenta que en el plan de Amazon que se tuvo para este servidor es muy bueno, el resultado y si se mejora las características del mismo se puede llegar a la conclusión que el sistema tendría un rendimiento excelente.

Como punto final de estas pruebas se pudo tomar en cuenta que todas las solicitudes fueron atendidas con éxito y esto lo se pudo visualizar en el árbol de resultados de cada una de los casos evaluados.

3.3.2.3 Pruebas de aceptación.

Las pruebas de aceptación se realizaron para observar la satisfacción de los clientes en el momento de recibir el software, para ver el grado de confianza en las características no funcionales.

Las pruebas se realizaron para observar si todos los requerimientos solicitados en el alcance del presente proyecto se cumplieron (Requerimientos del usuario).

Las pruebas de aceptación se las realizó con los representantes de las regionales que asistieron a las reuniones de capacitación. En total se tuvo una concurrencia de 20 representantes a los cuales se efectuó las preguntas que se detallan a continuación.

¿Qué tal le pareció la interfaz del sistema?



Figura 38.- Resultados obtenidos prueba de aceptación, Pregunta 1
Elaborado por: Carrión Richard & Ladino Stalin (2017)

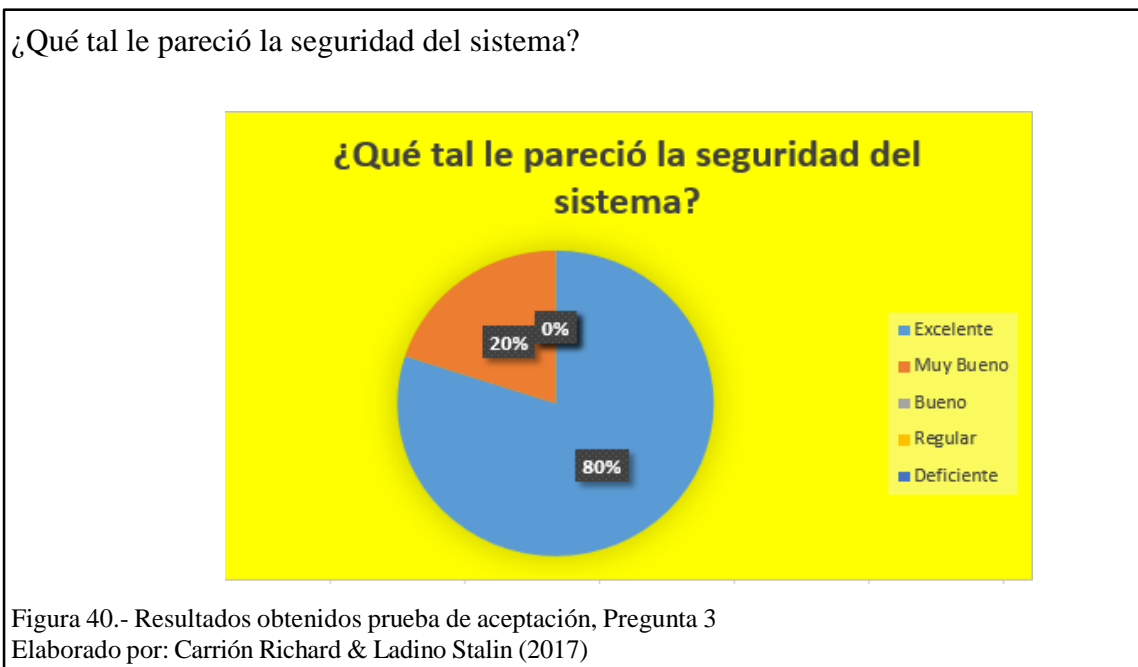
Como se pudo observar en cuanto a la interfaz del sistema se obtuvo un 85% excelente y un 15% de Muy bueno, por lo que se pudo concluir que tuvo un alto grado de aceptación por los usuarios.

¿Qué tal le pareció la usabilidad del sistema?



Figura 39.- Resultados obtenidos prueba de aceptación, Pregunta 2
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Como se pudo observar en cuanto a la usabilidad del sistema se obtuvo un 70% excelente y un 30 % de Muy bueno, por lo que se pudo concluir que tuvo un alto grado de aceptación por los usuarios, esto se debió a que la interfaz es muy intuitiva y facilita el uso de la misma.



Como se pudo observar en cuanto a la seguridad del sistema se obtuvo un 80% excelente y un 20 % de Muy bueno, por lo que se pudo concluir que tuvo un alto grado de aceptación por los usuarios, esta pregunta se realizó debido a que los usuarios tenían cierto nivel de inseguridad en cuanto a esta funcionalidad del sistema, esto debido a que estaban acostumbrados a hacer el proceso de manera manual o desconocían de los procesos que se llevan actualmente para controlar la información.

¿Qué tal le pareció los tiempos de respuesta del sistema?



Figura 41.- Resultados obtenidos prueba de aceptación, Pregunta 4
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Como se pudo observar en cuanto a los tiempos de respuesta y rendimiento del sistema se obtuvo un 75% excelente, un 20 % de Muy bueno y un 5% Bueno, por lo que se pudo concluir que tuvo un alto grado de aceptación por los usuarios, esta pregunta se la realizó para observar el nivel de aceptación de los usuarios en cuanto a la respuesta que dio el sistema a las peticiones realizadas por los mismos.

¿Qué tal le pareció la experiencia al usar el sistema?



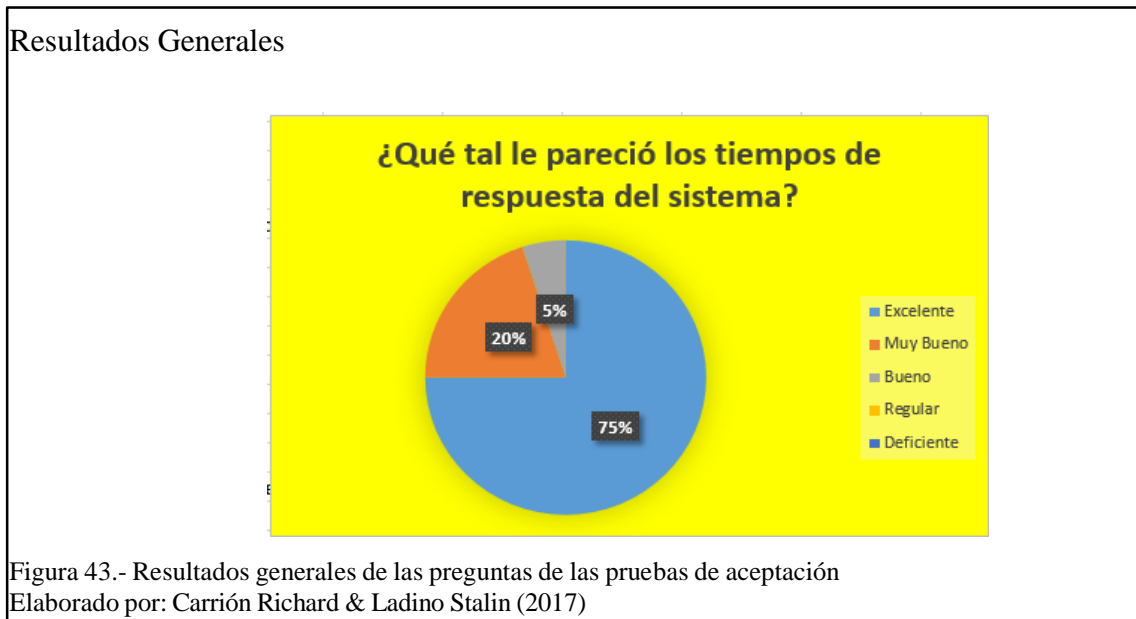
Figura 42.- Resultados obtenidos prueba de aceptación, Pregunta 5
Elaborado por: Carrión Richard & Ladino Stalin (2017)

Como se pudo observar en cuanto a la experiencia en forma general con los sistemas se obtuvo un 80% excelente, un 15 % de Muy bueno y un 5% Bueno, por lo que se pudo concluir que tuvo un alto grado de aceptación por los usuarios al momento de realizar las

pruebas respectivas.

En forma general se pudo obtener un 78% excelente, un 20 % de Muy bueno y un 2% Bueno, con lo cual se puede llegar a la conclusión que el sistema tuvo una buena aceptación por parte de los usuarios, hay un pequeño porcentaje de “Bueno” el cual se pudo observar que se dio porque hay usuarios que no tienen mucha experiencia en el uso de estos sistemas y que tienen ciertos inconvenientes los cuales no les permiten tener una percepción del 100% en cuanto a las características evaluadas en estas pruebas.

Los resultados finales los se pudo observar en la figura que se presenta a continuación.



CONCLUSIONES

- El proceso del desarrollo de una aplicación comienza con la etapa de la recolección de los datos y de la información, que ayude a entender cada uno de los procesos a automatizar. Es por esta razón que se llevó un gran trabajo con los miembros de las juntas comunitarias de Pichincha e Imbabura para entender cada una de las realidades que llevan al momento de realizar el proceso de la facturación y al final poder construir una aplicación que se acople a cada una de ellas y de esta forma lograr la total satisfacción de todas las partes interesadas.
- La metodología utilizada como es “Scrum” permitió definir cada uno los roles necesarios y artefactos para poder desarrollar este software con las mejores prácticas posibles. Cada una de las reuniones realizadas como recomienda la metodología permitió una retroalimentación continua en cuanto a cambios y para solventar las dudas que se presentaban a lo largo del ciclo. Esta metodología fue de gran ayuda puesto que en cada reunión permitió mostrar resultados tangibles, con lo cual se pudo tener al cliente y al grupo de trabajo motivado y por ende mejorar la productividad.
- La arquitectura utilizada como fue “MVC” permitió tener una aplicación bien estructurada, lo cual facilitó al equipo de trabajo en la construcción del software, disminuyó tiempos de trabajo gracias a la reutilización de ciertas funcionalidades en las diferentes interfaces. Al utilizar esta arquitectura permitió mejorar el proceso de cambios y en un futuro mantenimiento facilitará el trabajo.

- La tecnología utilizada “JSF” beneficio de manera relevante, puesto que es una tecnología que tiene gran robustez y que permite mejorar de manera significativa el rendimiento de nuestra aplicación como se pudo constatar en las diferentes pruebas practicadas a la aplicación. Además de otros beneficios que dispone como son el conjunto de componentes que dispone para el manejo de la interfaz de usuario.
- En temas de seguridad y de funcionamiento se optó por un servidor ofertado por la empresa “Amazon”, el cual brinda muchos beneficios en cuanto a seguridad ya que esta empresa se encarga de estos servicios. Otra de las razones por las cuales se eligió este servidor es por la escalabilidad de planes que dispone si se llega a la necesidad de mejorar el rendimiento de nuestra aplicación o del espacio de almacenamiento, además que los precios son muy flexibles.
- El software utilizado en el desarrollo del proyecto es de licencia gratuita, por lo cual los usuarios no tendrán que pagar licencia, accediendo a un sistema de bajo costo. Durante el desarrollo del proyecto se tomaron en cuenta todos los aspectos para entregar un producto de excelente calidad y que cumpla con todos los requerimientos solicitados en este proyecto técnico de titulación.

RECOMENDACIONES

- Se debe llevar a cabo pruebas continuas y con la mayor participación del cliente esto permite corroborar que el sistema cumpla con los requerimientos establecidos, permite encontrar errores y corregirlos oportunamente.
- Se debe tener la mayor interacción del usuario con el sistema, de esta manera el usuario puede relacionarse con el mismo y por tanto se reduce el riesgo de resistencia al uso cuando el software entre en un proceso de producción.
- Se debe hacer uso de herramientas como el prototipo, puesto que herramientas como estas facilitan aterrizar los requerimientos que muchas de las veces no tienen muy claro los clientes y que permiten tener más claro el alcance del proyecto.
- Se debe saber elegir la mejor metodología y sus artefactos esto permitirá entregar un producto de calidad en el menor tiempo y con un costo reducido.
- Se debe tomar muy en cuenta la arquitectura del sistema, esta es la parte más importante del mismo. Se debe para esto tomar en cuenta aspecto como el número de usuarios, perfiles, frecuencia de uso y que valla orientada a la necesidad a resolver.
- Se debe escoger muy bien la tecnología a usar dependiendo de la necesidad a resolver, puesto que se debe tomar en cuenta aspectos importantes como la persistencia, que luego se observara en el rendimiento de nuestra aplicación.

REFERENCIAS BIBLIOGRÁFICAS

- Báez, M., Borrego, Á., Cordero, J., Cruz, L., González, M., Hernández, F., . . . Zapata, Á. (2012). *Intrducción a Android*. (V. López, & G. T. UCM, Edits.) Madrid, España, España: E.M.E. & Universidad Complutense de Madrid.
- Bernal Bermúdez, J., & Fernández Muñoz, L. (2012). *Programación Orientada a objetos con Java (POOJ)*. Madrid: Universidad Politécnica de Madrid.
- Cédric, S. (02 de 04 de 2011). *Curso de Java Server Faces 2*. Obtenido de www.solucionjava.com:
http://www.solucionjava.com/pdf/Curso_JSF2_Hibernate3.pdf
- Civici, C. (2014 de 05 de 05). *Primefaces User Guide*. Obtenido de Primefaces:
http://www.primefaces.org/docs/guide/primefaces_user_guide_5_0.pdf
- Gómez, R. (11 de 11 de 2015). *Blog de Rodrigo Gómez*. Obtenido de <http://rodrigogr.com/blog/modelo-vista-controlador/>
- Mateu, C. (2004). *Desarrollo de aplicaciones web*. Catalunya: Universidad Oberta de Catalunya.
- Perez Mora, O., & Gibert Ginestà, M. (2009). *Bases de datos en PostgreSQL*. Catalunya: Editorial de la Universitat Oberta de Catalunya.
- Sáenz Morras, J. (25 de 06 de 2008). *Universitat Oberta de Catalunya*. Obtenido de <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/669/1/00848tfc.pdf>
- Sánchez López, E. A., & Letelier Torres, P. (12 de 11 de 2003). Metodologías Ágiles en el Desarrollo de Software. *VIII Jornadas de Ingeniería del Software y Bases de Datos*,. Alicante, España. Obtenido de <http://issi.dsic.upv.es/archives/f-1069167248521/actas.pdf>
- Sheldon, R., & Oppel, A. (2010). *Fundamentos de SQL "Tercera edición"*. México: McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Sierra Rivadeneira, E. D., & Salgado Flores, L. P. (2014). *ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE FACTURACIÓN PARA LAS COMUNIDADES DEL PROYECTO PESILLO IMBABURA*. Quito: Universidad Politécnica Salesiana.
- Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). *FUNDAMENTOS DE BASES DE DATOS. Cuarta edición*. Madrid: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.