

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“DESARROLLO DE UNA APLICACIÓN WEB UTILIZANDO LA NORMA NTE INEN-ISO/IEC 9126-1 PARA AUTOMATIZAR EL COBRO DEL SERVICIO DE AGUA POTABLE EN LA PROVINCIA DE MORONA SANTIAGO”**

Trabajo de titulación presentado para optar al grado de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**AUTORES:** EDUARDO JOSÉ GUADALUPE RODRIGUEZ

SAÚL MESÍAS MOROCHO AHOÑA

**TUTOR:** ING.JUAN CARLOS YÚNGAN

Macas, Ecuador

2016

**©2016,** Eduardo José Guadalupe Rodríguez, Saúl Mesías Morocho Ahoña.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Eduardo José Guadalupe Rodríguez Saúl Mesías Morocho Ahoña

**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

El Tribunal del trabajo de titulación certifica que: DESARROLLO DE UNA APLICACIÓN WEB UTILIZANDO LA NORMA NTE INEN-ISO/IEC 9126-1 PARA AUTOMATIZAR EL COBRO DEL SERVICIO DE AGUA POTABLE EN LA PROVINCIA DE MORONA SANTIAGO*,* de responsabilidad de los señores Eduardo José Guadalupe Rodríguez y Saúl Mesías Morocho Ahoña, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, quedando autorizada su presentación.

DR. MIGUEL TASAMBAY

**DECANO DE LA FACULTAD DE ------------------------- --------------------------**

**INFORMÁTICA Y ELECTRÓNICA**

DR. JULIO SANTILLAN

**DIRECTOR DE ESCUELA DE ------------------------- --------------------------**

**INGENIERÍA EN SISTEMAS**

ING. JUAN YÚNGAN

**DIRECTOR DE TESIS --------------------------- ---------------------------**

ING. ROBERTO INSUASTI

**MIEMBRO DE TRIBUNAL -------------------------- ---------------------------**

“Nosotros EDUARDO JOSÉ GUALAUPE RODRÍEGUEZ Y SAÚL MESÍAS MOROCHO AHOÑA, somos responsables de las ideas y resultados expuestos en este trabajo de titulación; y, el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Eduardo José Guadalupe Rodríguez Saúl Mesías Morocho Ahoña

# **DEDICATORIA**

A mis padres Rosa y Hugo quienes me han apoyado desde el momento en que decidí emprender este camino, a mi hermano q me ha alentado durante los momentos difíciles, a mis abuelos Piedad y José quienes cuidaron de mi durante los días de cansancio, a mi esposa Gabriela que en los momentos de duda supo alentarme para seguir sin desmayar y a mí hijo Yurem quien ha sido el motor que me da la fuerza para seguir luchando cada día y permitirme ser un ejemplo para él.

**Eduardo G.**

Dedico este Trabajo de Titulación a mi Dios por cuidar de mí en cada paso que doy en este camino largo de la vida que recién empieza, a mi Familia por darme el apoyo incondicional y por proporcionar ese aliento de fortaleza que he necesitado cuando decaía pero en especial dedico este trabajo a mi hijo Dylan por ser la inspiración para salir adelante en busca de un mejor futuro.

**Saúl M.**

# **AGRADECIMIENTO**

Quiero agradecer a mis profesores ya que ellos me enseñaron que ser un profesional es mucho más que tener un título, me enseñaron que nunca debemos olvidar de dónde venimos y que las metas planteadas siempre se pueden lograr, a mis compañeros que me han acompañado durante todo este camino apoyándome y muchas veces guiándome. A mi familia por brindarme las facilidades y el amor necesario para ser constante y decidido en este largo andar. Por último quiero agradecer a Dios por cuidar de mí y de mi familia.

**Eduardo G.**

Agradezco a Dios por guiarme y darme la fuerza cada día para seguir adelante en mis estudios y lo profesional, a mi Familia por el apoyo en las decisiones tomadas y a mis profesores quienes me impartieron sus conocimientos en las aulas día tras día para formar el profesional que he llegado a ser.

**Saúl M.**

# **TABLA DE CONTENIDO**

**CONTENIDO**

PORTADA…………………………………………………………………………………….…i

DERECHOS DE AUTOR………………………………………………………………..……...ii

CERTIFICACION…..……………………………………………………………………..……iii

DECLARACION DE RESPONSABILIDAD…………,…………………………………….....iv

[**DEDICATORIA** iv](#_Toc446506639)

[**AGRADECIMIENTO** v](#_Toc446506640)

[**TABLA DE CONTENIDO** vi](#_Toc446506641)

[**ÍNDICE DE TABLAS** ix](#_Toc446506642)

[**ÍNDICE DE FIGURAS** xi](#_Toc446506643)

[**ÍNDICE DE ANEXOS** xiv](#_Toc446506644)

[**SUMMARY** xvii](#_Toc446506645)

[**INTRODUCCIÓN** 1](#_Toc446506646)

[**CAPITULO I** 8](#_Toc446506647)

[**1.** **MARCO TEORICO** 8](#_Toc446506648)

[**1.1.** **NTE INEN-ISO/IEC 9126-1** 8](#_Toc446506649)

[**1.2.** **JSP** 12](#_Toc446506650)

[***1.2.1.*** ***Concepto*** 12](#_Toc446506651)

[***1.2.2.*** ***Ventajas*** 12](#_Toc446506652)

[***1.2.3.*** ***Como funciona JSP*** 13](#_Toc446506653)

[**1.3.** **Servlet** 14](#_Toc446506654)

[**1.4.** **Bootstrap** 14](#_Toc446506655)

[***1.4.1.*** ***Concepto*** 14](#_Toc446506656)

[***1.4.2.*** ***Ventajas*** 15](#_Toc446506657)

[***1.4.3.*** ***Desventajas*** 15](#_Toc446506658)

[**1.5.** **Java Script** 16](#_Toc446506659)

[***1.5.1.*** ***Concepto*** 16](#_Toc446506660)

[***1.5.2.*** ***Ventajas*** 16](#_Toc446506661)

[***1.5.3.*** ***Desventajas*** 17](#_Toc446506662)

[***1.5.4.*** ***Formas de Usar JavaScript*** 17](#_Toc446506663)

[**1.6.** **Arquitectura 3 Capas** 18](#_Toc446506664)

[***1.6.1.*** ***Concepto*** 18](#_Toc446506665)

[***1.6.2.*** ***Capas*** 19](#_Toc446506666)

[**1.7.** **iReport** 20](#_Toc446506667)

[***1.7.1.*** ***Concepto*** 20](#_Toc446506668)

[***1.7.2.*** ***Características*** 20](#_Toc446506669)

[**1.8.** **PostgreSQL** 21](#_Toc446506670)

[***1.8.1.*** ***Concepto*** 21](#_Toc446506671)

[***1.8.2.*** ***Ventajas*** 21](#_Toc446506672)

[***1.8.3.*** ***Desventajas*** 22](#_Toc446506673)

[**1.9.** **PgAdmin 3** 22](#_Toc446506674)

[***1.9.1.*** ***Concepto*** 22](#_Toc446506675)

[***1.9.2.*** ***Características*** 22](#_Toc446506676)

[**1.10.** **Aplicación WEB** 24](#_Toc446506677)

[***1.10.1.*** ***Concepto*** 24](#_Toc446506678)

[***1.10.2.*** ***Ventajas*** 24](#_Toc446506679)

[***1.10.3.*** ***Desventajas*** 24](#_Toc446506680)

[**1.11.** **Ubuntu Server** 25](#_Toc446506681)

[***1.11.1.*** ***¿Qué es?*** 25](#_Toc446506682)

[***1.11.2.*** ***Requisitos para instalar Ubuntu Server*** 26](#_Toc446506683)

[**CAPITULO II** 27](#_Toc446506684)

[2. Desarrollo de una Aplicación Web utilizando la norma NTE INEN-ISO/IEC 9126-1 para automatizar el cobro del servicio de Agua Potable en la Provincia de Morona Santiago 27](#_Toc446506685)

[**2.1.** **Visión y alcance.** 27](#_Toc446506686)

[***2.1.1.*** ***Definición del problema.*** 27](#_Toc446506687)

[***2.1.2.*** ***Visión del proyecto.*** 27](#_Toc446506688)

[***2.1.3.*** ***Perfiles de Usuario*** 28](#_Toc446506689)

[***2.1.4.*** ***Ámbito del proyecto.*** 29](#_Toc446506690)

[***2.1.5.*** ***Herramientas a utilizar.*** 30](#_Toc446506691)

[***2.1.6.*** ***Objetivos del proyecto*** 36](#_Toc446506692)

[***2.1.7.*** ***Análisis y gestión del riesgo*** 37](#_Toc446506693)

[***2.1.8.*** ***Planificación inicial*** 44](#_Toc446506694)

[***2.1.9.*** ***Estudio de Factibilidad*** 56](#_Toc446506695)

[**2.2.** **Planificación** 58](#_Toc446506696)

[***2.2.1.*** ***Definición y descripción de requerimientos*** 58](#_Toc446506697)

[***2.2.2.*** ***Diseño lógico*** 84](#_Toc446506698)

[***3.1.1.*** ***Diseño físico*** 95](#_Toc446506699)

[**2.3.** **Desarrollo** 96](#_Toc446506700)

[***2.3.1.*** ***Estándar de diseño de base de datos*** 96](#_Toc446506701)

[***2.3.2.*** ***Estándar De Programación De Aplicaciones.*** 97](#_Toc446506702)

[***2.3.3.*** ***SIFACOAG*** 99](#_Toc446506703)

[**2.4.** **Estabilización** 103](#_Toc446506704)

[***2.4.1.*** ***Revisión del sistema*** 103](#_Toc446506705)

[***2.4.2.*** ***Plan de pruebas.*** 112](#_Toc446506706)

[**2.5.** **Instalación** 114](#_Toc446506707)

[***2.5.1.*** ***Para instalar SIFACOAG se debe usar un servidor con las siguientes características.*** 114](#_Toc446506708)

[**CAPITULO III** 124](#_Toc446506709)

[3. Evaluación del sistema de Facturación y Control del Servicio de Agua Potable bajo la norma NTE INEN-ISO/IEC 9126-1. 124](#_Toc446506710)

[**3.1.** **NORMA NTE INEN-ISO/IEC 9126-1** 124](#_Toc446506711)

[***3.1.1.*** ***Introducción*** 124](#_Toc446506712)

[***3.1.2.*** ***Objeto y campo de aplicación*** 124](#_Toc446506713)

[***3.1.3.*** ***Calidad del producto y ciclo de vida*** 125](#_Toc446506714)

[***3.1.4.*** ***Modelo de calidad para calidad interna y externa*** 127](#_Toc446506715)

[***3.1.5.*** ***Modelo de calidad para calidad en uso*** 132](#_Toc446506716)

[***3.1.6.*** ***Las seis características ISO de calidad del software*** 133](#_Toc446506717)

[3.2. **EVALUACIÓN** 134](#_Toc446506718)

[*3.2.1.* *Modelo de calidad para calidad externa e interna* 134](#_Toc446506719)

[**ANALISIS DE RESULTADOS** 141](#_Toc446506720)

[**CONCLUSIONES** 146](#_Toc446506721)

[**RECOMENDACIONES** 147](#_Toc446506722)

# **ÍNDICE DE TABLAS**

[Tabla 1 – 2 Herramientas a utilizar en SIFACOAG 30](#_Toc446506723)

[Tabla 2 – 2 Riesgos 37](#_Toc446506724)

[Tabla 3 – 2 Determinación del impacto 38](#_Toc446506725)

[Tabla 4 – 2 Probabilidad que ocurra 38](#_Toc446506726)

[Tabla 5 – 2 Determinación de la exposición del riesgo 38](#_Toc446506727)

[Tabla 6 – 2 Código de colores de impacto 39](#_Toc446506728)

[Tabla 7 – 2 Análisis de Riesgo 39](#_Toc446506729)

[Tabla 8 – 2 Línea de corte 40](#_Toc446506730)

[Tabla 9 – 2 Riesgo 9 41](#_Toc446506731)

[Tabla 10 – 2 Riesgo 4 42](#_Toc446506732)

[Tabla 11 – 2 Riesgo 1 43](#_Toc446506733)

[Tabla 12 – 2 Roles de los miembros 44](#_Toc446506734)

[Tabla 13 – 2 DEFINICION DE PIECES 47](#_Toc446506735)

[Tabla 14 – 2 ESTRCTURA PIECES 47](#_Toc446506736)

[Tabla 15 – 2 Requerimiento 1 53](#_Toc446506737)

[Tabla 16 – 2 Requerimiento 2 53](#_Toc446506738)

[Tabla 17 – 2 Requerimiento 3 53](#_Toc446506739)

[Tabla 18 – 2 Requerimiento 4 53](#_Toc446506740)

[Tabla 19 – 2 Requerimiento 5 54](#_Toc446506741)

[Tabla 20 – 2 Requerimiento 6 54](#_Toc446506742)

[Tabla 21 – 2 Requerimiento 7 54](#_Toc446506743)

[Tabla 22 – 2 Requerimiento 8 54](#_Toc446506744)

[Tabla 23 – 2 Requerimiento 9 55](#_Toc446506745)

[Tabla 24 – 2 Requerimiento 10 55](#_Toc446506746)

[Tabla 25 – 2 Factibilidad económica 58](#_Toc446506747)

[Tabla 26 – 2 Características del servidor 62](#_Toc446506748)

[Tabla 27 – 2 Equipo cliente 62](#_Toc446506749)

[Tabla 28 – 2 Software requerido 63](#_Toc446506750)

[Tabla 29 – 2 Caso de uso. Requerimiento # 1: 74](#_Toc446506751)

[Tabla 30 – 2 Caso de uso. Requerimiento # 2: 75](#_Toc446506752)

[Tabla 31 - 2 Caso de uso. Requerimiento # 3: 76](#_Toc446506753)

[Tabla 32 – 2 Caso de uso. Requerimiento # 4: 77](#_Toc446506754)

[Tabla 33 – 2 Caso de uso. Requerimiento # 5: 78](#_Toc446506755)

[Tabla 34 – 2 Caso de uso. Requerimiento # 6: 79](#_Toc446506756)

[Tabla 35 – 2 Caso de uso. Requerimiento # 7: 80](#_Toc446506757)

[Tabla 36 – 2 Caso de uso. Requerimiento # 8: 81](#_Toc446506758)

[Tabla 37 – 2 Caso de uso. Requerimiento # 9: 82](#_Toc446506759)

[Tabla 38 – 2 Caso de uso requerimiento 10 83](#_Toc446506760)

[Tabla 39 – 2 extensiones 98](#_Toc446506761)

[Tabla 40 – 2 aspectos sujetos a evaluación 112](#_Toc446506762)

[Tabla 41 – 3 Métrica Funcionalidad 135](#_Toc446506763)

[Tabla 42 – 3 Métrica Fiabilidad 136](#_Toc446506764)

[Tabla 43 – 3 Métrica Usabilidad 137](#_Toc446506765)

[Tabla 44 – 3 Métrica Eficiencia 138](#_Toc446506766)

[Tabla 45 – 3 Métrica Mantenibilidad 139](#_Toc446506767)

[Tabla 46 – 3 Métrica Portabilidad 140](#_Toc446506768)

[Tabla 47 – 3 Resultados de métricas 145](#_Toc446506769)

# **ÍNDICE DE FIGURAS**

[Figura 1 – 1 Relación entre las normas ISO/IEC 9162 e ISO/IEC 14598 8](#_Toc446506770)

[Figura 2 – 1 Influencia de la Norma en el ciclo de vida del software 10](#_Toc446506771)

[Figura 3 – 1 Calidad del producto y métricas asociadas. 10](#_Toc446506772)

[Figura 4 – 1 Petición de página JSP y obtención de respuesta. 13](#_Toc446506773)

[Figura 5 – 1 Arquitectura de la aplicación 19](#_Toc446506774)

[Figura 6 – 2 Orgánico Estructural 45](#_Toc446506775)

[Figura 7 – 2 Requerimiento 1 48](#_Toc446506776)

[Figura 8 – 2 Requerimiento 2 48](#_Toc446506777)

[Figura 9 – 2 Requerimiento 3 49](#_Toc446506778)

[Figura 10 – 2 Requerimiento 4 49](#_Toc446506779)

[Figura 11 - 2 Requerimiento 5 50](#_Toc446506780)

[Figura 12 – 2 Requerimiento 6 50](#_Toc446506781)

[Figura 13 – 2 Requerimiento 7 51](#_Toc446506782)

[Figura 14 – 2 Requerimiento 8 51](#_Toc446506783)

[Figura 15– 2 Requerimiento 9 52](#_Toc446506784)

[Figura 16– 2 Requerimiento 10 52](#_Toc446506785)

[Figura 17 – 2 Cronograma de Actividades, diagrama de GRANT 55](#_Toc446506786)

[Figura18 – 2 Diagrama de GRANT de seguimiento 56](#_Toc446506787)

[Figura 19 – 2 Caso de uso 1 74](#_Toc446506788)

[Figura 20 – 2 Caso de uso 2 75](#_Toc446506789)

[Figura 21 – 2 Caso de uso 3 76](#_Toc446506790)

[Figura 22 – 2 Caso de uso 4 77](#_Toc446506791)

[Figura 23 – 2 Caso de uso 5 78](#_Toc446506792)

[Figura 24 – 2 Caso de uso 6 79](#_Toc446506793)

[Figura 25 – 2 Caso de uso 7 80](#_Toc446506794)

[Figura 26 – 2 Caso de uso 8 81](#_Toc446506795)

[Figura 27 – 2 Caso de uso 9 82](#_Toc446506796)

[Figura 28 – 2 Caso de uso 10 83](#_Toc446506797)

[Figura 29- 2 Arquitectura del sistema 84](#_Toc446506798)

[Figura 30 – 2 Requerimiento 1 85](#_Toc446506799)

[Figura 31 – 2 Requerimiento 2 85](#_Toc446506800)

[Figura 32 – 2 Requerimiento 3 86](#_Toc446506801)

[Figura 33 – 2 Requerimiento 4 86](#_Toc446506802)

[Figura 34 – 2 Requerimiento 5 87](#_Toc446506803)

[Figura 35 – 2 Requerimiento 6 87](#_Toc446506804)

[Figura 36 – 2 Requerimiento 7 88](#_Toc446506805)

[Figura 37 – 2 Requerimiento 8 88](#_Toc446506806)

[Figura 38 – 2 Requerimiento 9 89](#_Toc446506807)

[Figura 39 – 2 Requerimiento 10 89](#_Toc446506808)

[Figura 40 – 2 Diagrama de clases 90](#_Toc446506809)

[Figura 41 – 2 Página principal 91](#_Toc446506810)

[Figura 42 – 2 Registro de clientes 91](#_Toc446506811)

[Figura 43 – 2 Registro tipo usuarios 92](#_Toc446506812)

[Figura 44 – 2 Registro usuarios 92](#_Toc446506813)

[Figura 45 – 2 Asignación de Roles 92](#_Toc446506814)

[Figura 46 – 2 Registro zonas 93](#_Toc446506815)

[Figura 47 – 2 Registro tipo tarifas 93](#_Toc446506816)

[Figura 48 – 2 Registro tarifas 93](#_Toc446506817)

[Figura 49 – 2 Registro cuenta medidor 94](#_Toc446506818)

[Figura 50 – 2 Consulta historial del cliente 94](#_Toc446506819)

[Figura 51 – 2 Generar reportes 95](#_Toc446506820)

[Figura 52 – 2 Diagrama de implementación 95](#_Toc446506821)

[Figura 53 – 2 Modelo físico de base de datos 96](#_Toc446506822)

[Figura 54 – 2 Código correspondiente a la función cliente 100](#_Toc446506823)

[Figura 55 – 2 Capa acceso a datos 101](#_Toc446506824)

[Figura 56 – 2 Capa de negocio 101](#_Toc446506825)

[Figura 57 – 2 Servlets 102](#_Toc446506826)

[Figura 58 – 2 Capa de aplicación 103](#_Toc446506827)

[Figura 59 – 2 PgAdmin 117](#_Toc446506828)

[Figura 60 – 2 Creación base de datos 118](#_Toc446506829)

[Figura 61 – 2 Ingresando al formulario para la creación de la base de datos. 118](#_Toc446506830)

[Figura 62 – 2 Creando base de datos 119](#_Toc446506831)

[Figura 63 – 2 Restauración 119](#_Toc446506832)

[Figura 64 – 2 Escogiendo backup 119](#_Toc446506833)

[Figura 65 – 2 Restaurando la base de datos 120](#_Toc446506834)

[Figura 66 – 2 Formulario para el ingreso al servidor web 121](#_Toc446506835)

[Figura 67 – 2 opciones del servidor 121](#_Toc446506836)

[Figura 68 – 2 Opciones para subir el archivo 122](#_Toc446506837)

[Figura 69 – 2 Opción para buscar el archivo .war 122](#_Toc446506838)

[Figura 70 – 2 buscar el archivo .war 122](#_Toc446506839)

[Figura 71 – 2 Subir el archivo .war 123](#_Toc446506840)

[Figura 72 – 2 Entrar al sistema 123](#_Toc446506841)

[Figura 73 – 3 Métricas de calidad externa e interna 127](#_Toc446506842)

[Figura 74 – 3 Atributos de calidad en uso 132](#_Toc446506843)

[Figura 75 – 3 Métrica Funcionalidad 135](#_Toc446506844)

[Figura 76 – 3 Métrica Fiabilidad 136](#_Toc446506845)

[Figura 77 – 3 Métrica Usabilidad 137](#_Toc446506846)

[Figura 78 – 3 Métrica Eficiencia 138](#_Toc446506847)

[Figura 79 – 3 Métrica Mantenibilidad 139](#_Toc446506848)

[Figura 80 – 3 Métrica Portabilidad 140](#_Toc446506849)

# **ÍNDICE DE ANEXOS**

**Anexo A:** Norma NTE INEN-ISO/IEC 9126-1.

**Anexo B:** Ordenanza que reglamenta la prestación de los servicios de agua potable del Cantón Logroño.

**Anexo C:** Encuestas.

**Anexo D:** Manual de Usuario.

**ÍNDICE DE ABREVIATURAS**

|  |  |
| --- | --- |
| **JSP**  **BD**  **SICACOAG**  **INEN**  **ISO**  **IEC**  **SO**  **IP**  **HTTP**  **JDK**  **SRS**  **UML**  **SQL**  **MSF**  **RP**  **RT**  **RN**  **PHP**  **HTML**  **CSS**  **PC**  **PDF**  **AMD** | SERVIDOR DE PÁGINAS JAVA  BASE DE DATOS  SISTEMA DE FACTURACIÓN Y CONTROL DEL AGUA POTABLE  INSTITUTO NACIONAL ECUATORIANO DE NORMALIZACIÓN  ORGANIZACIÓN INTERNACIONAL DE ESTANDARIZACIÓN  COMISIÓN INTERNACIONAL DE ELECTRÓNICA  SISTEMA OPERATIVO  PROTOCOLO DE INTERNET  PROTOCOLO DE TRANSFERENCIA DE HIPERTEXTO  KIT DE DESARROLLO JAVA  ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE  LENGUAJE UNIFICADO DE MODELADO  LENGUAJE DE CONSULTA ESTRUCTURADO  MICROSOFT SOLUTION FRAMEWORK  RIESGO DEL PROYECTO  RIESGO TÉCNICO  RIESGO DE NEGOCIOS  PROCESADOR DE HIPERTEXTO  LENGUAJE DE MARCAS DE HIPERTEXTO  HOJAS DE ESTILO EN CASCADA  COMPUTADOR PERSONAL  FORMATO DE DOCUMENTO PORTÁTIL  MICRO DISPOSITIVOS AVANZADOS |

**RESUMEN**

El presente trabajo de titulación muestra la investigación y evaluación de la norma NTE INEN-ISO/IEC 9126-1 que nos permite medir la calidad del producto de software final en base a los requisitos proporcionados por el cliente, esta norma está divida en 4 partes como son *Modelo de Calidad, Métricas Externas, Métricas Internas, Métricas de Calidad en Uso.*

La evaluación de la norma NTE INEN-ISO/IEC 9126-1 se enfocó en las métricas internas y externas para la calidad del producto de software que categoriza los atributos de calidad del software en seis características: *funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.*

Con los resultados obtenidos de la investigación de la norma NTE INEN-ISO/IEC 9126-1 se procede al desarrollo de la Aplicación Web denominada **SIFACOAG** *(Sistema de Facturación y Control del Agua Potable),* bajo las herramientas de desarrollo: **Netbeans 8.0.2** *(usado como entorno de desarrollo de la aplicación web),* **Java JSP** *(usado para agregar contenido dinámico a las páginas web),* **PostgreSql** *(utilizado como motor de base de datos para almacenamiento de la información),* **Ireport Java** *(usado para visualizar los diferentes reportes en formato PDF),* **Framework Bootstrap** *(usado para dar estilo a las páginas web y que sean adaptables al tamaño de dispositivo que lo visualice).*

Con la aplicación culminada se procede a realizar las pruebas de calidad del software con las métricas que establece la norma NTE INEN-ISO/IEC 9126-1 dando como resultado que **SIFACOAG** cumple a cabalidad con las necesidades requeridas por los clientes.

# **SUMMARY**

This paper shows the research titration and evaluation of the NTE INEN ISO / IEC 9126-1 standard that allows us to measure the quality of the final software product based on the requirements provided by the customer, this standard is divided into 4 parts such as Quality Model, Metrics External, Internal Metrics, Quality Metrics in Use.

The evaluation of the NTE INEN ISO / IEC 9126-1 standard focused on internal and external quality metrics for software product that categorizes software quality attributes into six characteristics: functionality, reliability, usability, efficiency, maintainability and portability.

 With the results of the investigation of the NTE INEN ISO / IEC 9126-1 is applicable to the development of the so-called Web Application SIFACOAG (Billing System and Control of Water), under development tools: Netbeans 8.0.2 (used as a development environment of the web application), Java JSP (used to add dynamic content to web pages), PostgreSql (used as engine database for storing information), Java Ireport (used to display the different reports in PDF format), Framework Bootstrap (used to style web pages and are adaptable to the size of the display device).

With the implementation culminated proceed to testing software quality metrics established by the NTE INEN ISO / IEC 9126-1 standard SIFACOAG result that fully meets the needs required by customers.

# **INTRODUCCIÓN**

En la actualidad existen gran demanda por los productos de software ya sea por grandes o pequeñas empresas, convirtiéndose el software en uno de los objetivos estratégicos de dichas empresas y su correcto funcionamiento es crítico para el éxito del negocio y seguridad de las personas, pero en el mercado competitivo no basta solo con producir y distribuir estos productos si no también brindar calidad en cada uno de los productos software que se ofertan.

La calidad en un producto de software ya sea en el proceso de desarrollo al momento de adquirir el producto final es muy importante, ya que de esta forma mediante varios procesos de evaluación, basado en las métricas más adecuadas para un software específico se pude garantizar la calidad de este producto.

La Norma NTE INEN-ISO/IEC 9126-1 es una de varias normas que ayudan en el control de calidad de software, este estándar está divido en cuatro partes las cuales son:

* *Parte 1: Modelo de Calidad.*
* *Parte 2: Métricas Externas.*
* *Parte 3: Métricas Internas.*
* *Parte 4: Métricas de Calidad en Uso.*

Esta norma está diseñada para los desarrolladores, adquirientes, personal que asegure la calidad y evaluadores independientes, quienes son los encargados de especificar y evaluar la calidad de software. Por lo que se puede decir que la Norma NTE INEN-ISO/IEC 9126-1 nos permite validar la completitud de los requisitos, identificar requisitos de calidad de software, objetivos de diseño, pruebas y criterios de aseguramiento de calidad.

El presente trabajo de titulación consta de 3 capítulos que se describen a continuación:

**Capítulo I** Marco Teórico, comprende conceptos, definiciones y características principales de las herramientas utilizadas en el desarrollo del ciclo de vida de software.

**Capitulo II** Desarrollo de la Aplicación **SIFACOAG** (*Sistema de Facturación y Control del Agua Potable*), se procede al desarrollo de la aplicación **SIFACOAG** y también está construido por la documentación de la aplicación desarrollada con la metodología ágil Microsoft Solution Framework (MSF).

**Capitulo III** Evaluación de la Norma NTE INEN-ISO/IEC 9126-1, está compuesta por la investigación, evaluación y aplicación de los resultados de la investigación en el producto final de software (*SIFACOAG*).

**ANTECEDENTES**

El desarrollo de aplicaciones ha evolucionado en el mundo tanto en los negocios, comunicación, educación, etc. Por tal motivo las empresas públicas o privadas se ven en la necesidad de mejorar sus servicios incluyéndose en este mundo tecnológico. Por lo que buscan automatizar todos los procesos que han venido realizando de una manera manual.

La principal ventaja que tienen las Aplicaciones Web es su independencia de ser instaladas en la computadora, ya que solo necesitamos una computadora, Internet y un buen navegador web, como Mozilla Firefox o Google Chrome, etc. Esta es la ventaja más significativa en cuanto a las Aplicaciones Web se refiere.

Existe otra ventaja la cual es la independencia del sistema operativo que se maneja, ya que eso no importa, lo único que le interesa a la aplicación es que tengas un navegador web y conexión a Internet, ya sea que estés en Windows, MacOS o Linux.

Al darnos cuenta que los sistemas web ofrecen valiosas ventajas tanto para las empresas como para sus clientes, creemos que es importante enfocarse en desarrollar este tipo de aplicaciones de forma ágil y eficiente, apuntando a la calidad del software.

En la actualidad para determinar la calidad del software se han establecido normas que ayudan al desarrollador a entregar un software de calidad a sus clientes. Por ejemplo la norma NTE INEN-ISO/IEC 9126-1 permite especificar y evaluar la calidad del software desde diferentes perspectivas ya sea con los requisitos o durante desarrollo.

Actualmente el manejo correcto de los recursos del estado, así como el libre acceso a la información hace que principalmente las instituciones públicas manejen estos recursos de forma transparente. Esto lleva a que hagan uso de nuevas tecnologías e internet como medio de comunicación y negocio, entre estos y la población.

Por medio de la propuesta presentada con este proyecto, se pretende automatizar el proceso de COBRO DEL SERVICIO DE AGUA POTABLE EN EL CANTÓN LOGROÑO, PROVINCIA DE MORONA SANTIAGO, logrando que este se lleve de forma ágil y clara.

**FORMULACIÓN DEL PROBLEMA**

El municipio del Cantón Logroño, Provincia de Morona Santiago desde su inicio ha brindado de manera exitosa servicios a los ciudadanos de su localidad, teniendo a cargo el mantenimiento y mejora de los sistemas de servicios públicos.

Por lo que tiene la obligación de realizar el cobro del servicio de agua potable. El cual se viene realizando de forma manual.

* Teniendo como consecuencia:
* El ingreso indirecto de la información a los registros.
* El cliente no tiene acceso al monto de su deuda hasta que se acerca a cancelar la misma.
* El cliente no cuenta con un historial de todas sus deudas y pagos.
* Perdida de información.
* Lentitud en los procesos de registro.
* Entre otros.

Al no permitir que el cliente pueda acceder a la información de su pago hace que el proceso sea lento al momento del cobro, así mismo la persona que realiza el registro al no realizarlo directamente da lugar a demoras y tergiversación de la información, lo cual afecta tanto a clientes como a los encargados del departamento de cobro de agua en el municipio.

Por este motivo es recomendable desarrollarlo una aplicación web que permita mejorar la ejecución de los procesos dentro de la institución, siendo estos: el registro de los medidores de forma directa, presentar la información a los clientes y facturación.

**SISTEMATIZACIÓN DEL PROBLEMA**

¿El registro por parte de las personas que recorren las calles es automatizado?

¿El sistema permitirá realizar registros de nuevos usuarios, Gestionar y emitir reportes?

¿Existe un sistema Web que automatice el proceso de facturación y registro?

¿Los clientes pueden acceder al sistema para consultar sus deudas?

**JUSTIFICACIÓN DEL TRABAJO DE GRADO**

**JUSTIFICACIÓN TEÓRICA**

Con la llegada de la era tecnológica y la adaptación de la sociedad al uso de sistemas web, ya sea para usarlos como medio de comunicación, negocios o placer. Las empresas e instituciones se han visto en la necesidad y/u obligación de realizar cambios en la forma en como ofrecen y brindan sus servicios, automatizando estos y logrando que se realicen de forma ágil y eficaz.

El uso de un sistema de facturación e información en el cobro de agua potable es importante dado que beneficia a la institución recibiendo la información de forma directa de las personas que leen los medidores al sistema, y para sus clientes ya que estos pueden tener información anticipada de cuanto deben cancelar por el servicio.

La aplicación web para la facturación de agua potable es una solución clara, ya que permite a la institución agilizar los procesos, con lo que mejora la atención a sus clientes, tiene como objetivo manejar información clara y precisa, presentar a tiempo la información a sus clientes (usuarios).

Para desarrollar la aplicación web que se propone, evaluaremos las normas NTE INEN-ISO/IEC 9126-1 las cuales nos permitirán entregar un software de calidad.

Para la gestión del proyecto de software se plantea usar la metodología ágil MSF.

MSF es un proceso en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto.

**JUSTIFICACIÓN APLICATIVA**

Nuestra propuesta va enfocada a automatizar el proceso de “Cobro y Facturación del Servicio de Agua Potable” del Cantón Logroño, Provincia de Morona Santiago mediante el desarrollo de una aplicación web que tendrá los siguientes módulos:

A continuación se describe las funciones que realizara cada usuario que maneje la Aplicación propuesta:

Administrador.- Se encargara de:

* Gestionar:
* Cuentas de cajero/a
* Cuentas clientes
* Cuentas registrador
* Generar reportes

Cajero/a.- Se encargara de realizar los cobros de las planillas, registrar la cantidad cancelada y emitir la factura correspondiente. Generar reportes de cobros realizados con anterioridad por parte de sus clientes (si estos lo solicitaran), reportes del trabajo realizado por los registradores ya sea diario, mensual o anual.

Registrador.- Se encargara de registrar vía web (móvil) los valores indicados en cada uno de los medidores.

Clientes.- Podrán ingresar a la página de consultas de la aplicación y realizar consultas correspondientes ya sea a sus medidores (numero) o con su número de cedula, emitir reportes sobre su consumo o imprimir respaldos de sus facturar.

Parámetros de usabilidad de la aplicación:

* Navegabilidad.
* Interfaz amigable para el usuario.
* Guía de ayuda en cada cuadro de introducción de texto

Interactiva con el usuario.

Con la aplicación web optimizaremos los siguientes procesos:

* Registro manual de los valores de cada medidor.
* Las consultas de los montos a pagar se los puedan realizar desde los hogares.
* Generación de reportes.

Ventajas que se obtendrán con su implementación:

* Control y seguridad en el cobro
* Seguimiento realizado a cada usuario registrador
* Integridad de la información
* Obtención de reportes que ayudaran a los clientes a determinar cuánto es el consumo que realizan.
* Poder acceder al sistema desde cualquier lugar, siempre y cuando se tenga acceso a internet.

**OBJETIVOS**

*OBJETIVO GENERAL*

“Desarrollar una aplicación web utilizando la norma NTE INEN-ISO/IEC 9126-1 para automatizar el servicio de cobro de agua.

*OBJETIVOS ESPECÍFICOS*

* Evaluar la norma NTE INEN-ISO/IEC 9126-1.
* Identificar los procesos que formen parte del cobro del servicio de agua potable.
* Desarrollar los módulos de registros, gestión y reportes de la aplicación web.
* Tomar como caso práctico el Municipio del Cantón Logroño.

# **CAPITULO I**

## **MARCO TEORICO**

## **NTE INEN-ISO/IEC 9126-1**

*INGENIERIA DEL SOFTWARE, CALIDAD DEL PRODUCTO SOFTWARE. MODELO DE CALIDAD.*

La norma 9136-1 es una traducción idéntica de la Norma Internacional ISO/IEC 9126-1:2001, IDT.

La Norma ISO/IEC 9126 consta de las siguientes partes que comparten el mismo título general: Ingeniería del Software. Calidad del producto:

* + *Parte 1: Modelo de calidad*
  + *Parte 2: Métricas externas.*
  + *Parte 3: Métricas Internas.*
  + *Parte 4: Métricas de calidad en uso.*

La Norma ISO/IEC 9126 (Calidad del producto de software) junto con la Norma ISO/IEC 14598 (Evaluación del producto de software). Surgen para reemplazar la norma ISO/IEC (1991). Las características de calidad del producto de software que se definen en esta parte la Norma ISO/IEC 9126 se pueden usar para especificar los requisitos funcionales y no funcionales tanto del cliente como del usuario.

**La figura 1 - 1,** muestra la relación entre las normas ISO/IEC 9162 e ISO/IEC 14598.

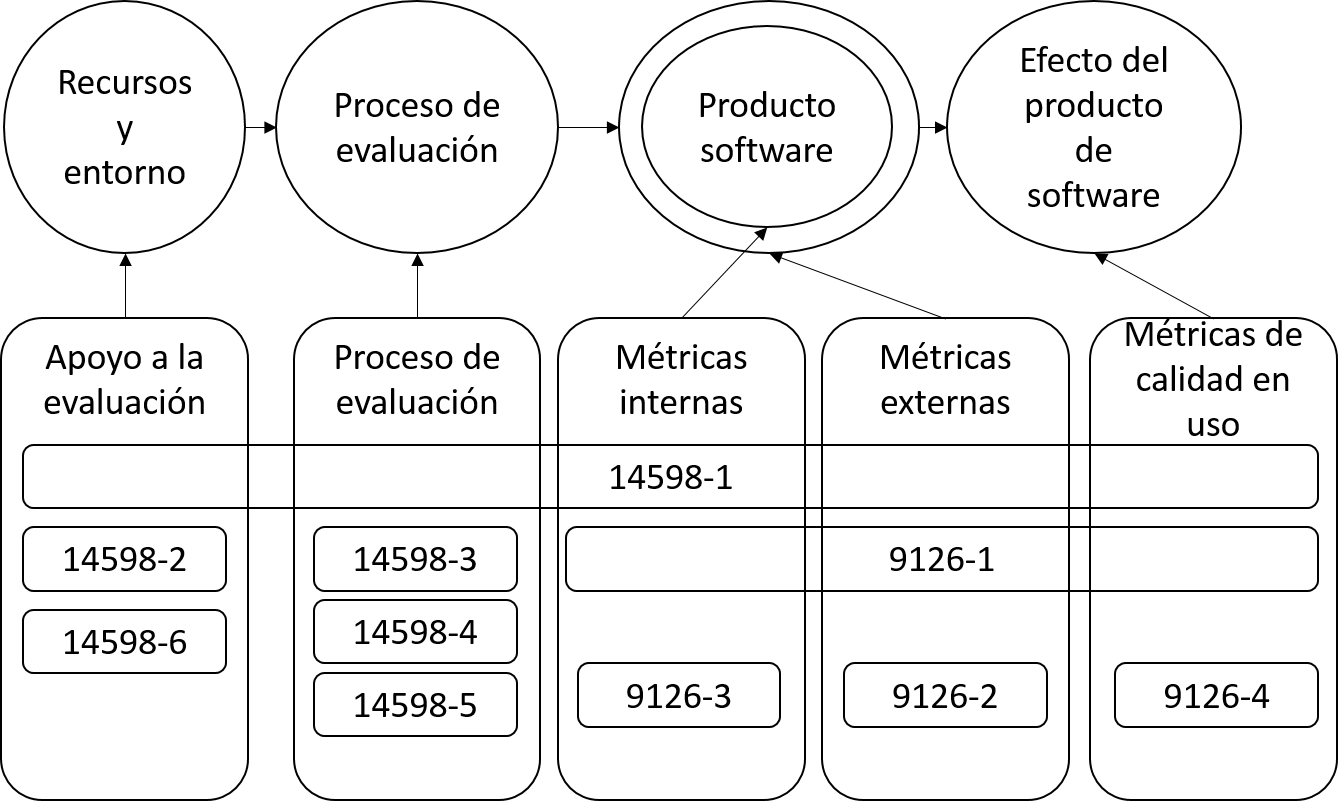


Figura 1 – 1 Relación entre las normas ISO/IEC 9162 e ISO/IEC 14598

**Fuente:** Norma NTE INE\_ISO/IEC 9126-1

Esta parte de la Norma ISO/IEC 9126 describe un modelo en dos partes para la calidad del producto software: a) calidad interna y externa, y b) calidad en uso. La primera parte del modelo especifica seis características para la calidad interna y externa, que se subdividen posteriormente en sub-características. Estas sub-características se manifiestan externamente cuando el software se usa como parte de un sistema informático, y son el resultado de los atributos internos del software.

La segunda parte del modelo especifica cuatro características de calidad en uso, pero no elabora el modelo de calidad en uso más allá del nivel de característica. La calidad en uso es el efecto combinado para el usuario de las seis características de calidad del producto software.

Esta parte de la Norma ISO/IEC 9126 permite especificar y evaluar la calidad de los productos software desde diferentes perspectivas por parte de aquellos agentes involucrados con la adquisición, los requisitos, el desarrollo, el uso, la evaluación, el soporte, el mantenimiento, el aseguramiento de la calidad y la auditoria del software. Por ejemplo, se puede usar por los desarrolladores, compradores, personal de aseguramiento de la calidad, y evaluadores independientes, especialmente aquellos responsables de especificar y evaluar la calidad del producto software. Ejemplos de usos del modelo de calidad definido en esta parte de la Norma ISO/IEC 9126 son:

* *Validar la completitud de una definición de requisitos;*
* *Identificar los requisitos del software;*
* *Identificar objetivos para el diseño del software;*
* *Identificar objetivos para las pruebas del software;*
* *Identificar requisitos para el aseguramiento de la calidad;*
* *Identificar los criterios de aceptación para un producto software completado.*

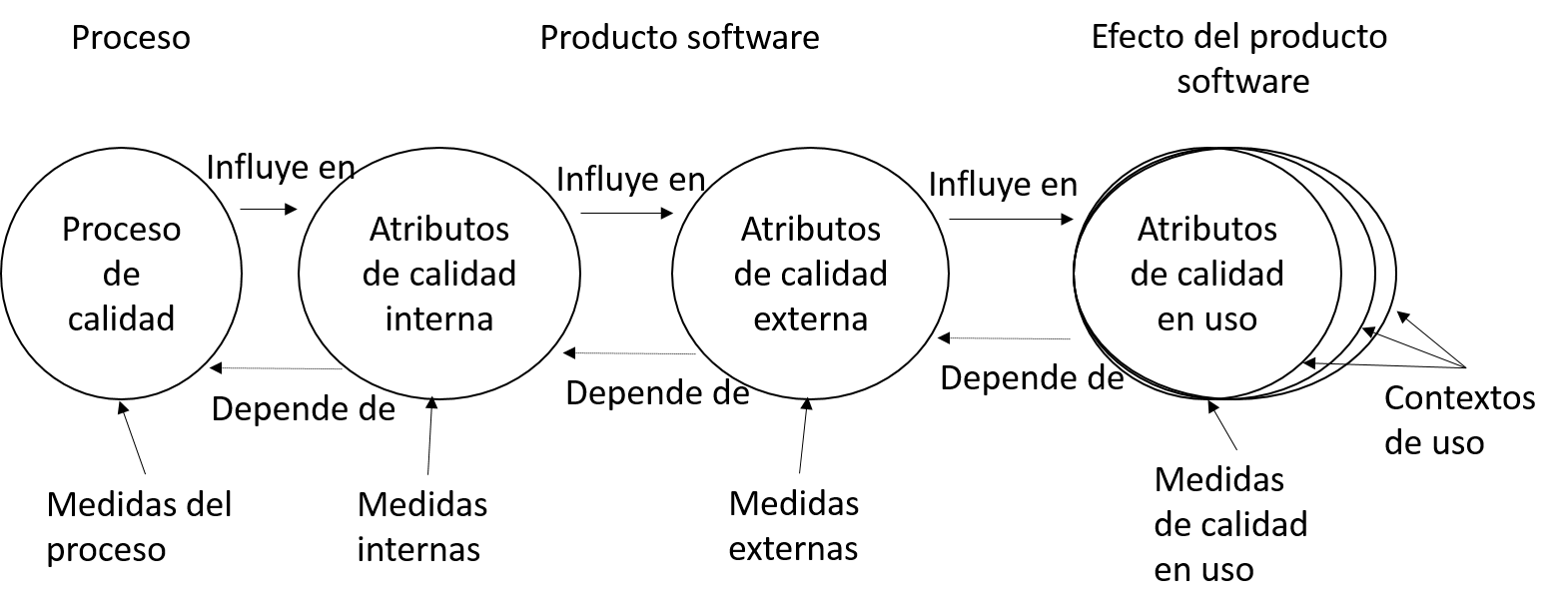


Figura 2 – 1 Influencia de la Norma en el ciclo de vida del software

**Fuente:** Norma NTE INE\_ISO/IEC 9126-1

***Calidad en el ciclo de vida***

La evaluación de los productos software para que satisfagan las necesidades de calidad software es uno de los procesos del ciclo de vida del desarrollo de software. La calidad del producto software se puede evaluar midiendo los atributos internos (normalmente medidas estáticas de productos intermedios), o midiendo los atributos externos (normalmente midiendo el comportamiento del código en ejecución), o midiendo los atributos de la calidad en uso. El objetivo es que el producto tenga el efecto requerido en un contexto de uso particular (figura 2 - 1).

Hay distintas perspectivas de la calidad del producto y de las métricas asociadas en las diferentes etapas del ciclo de vida del software (véase la figura 3 - 1).

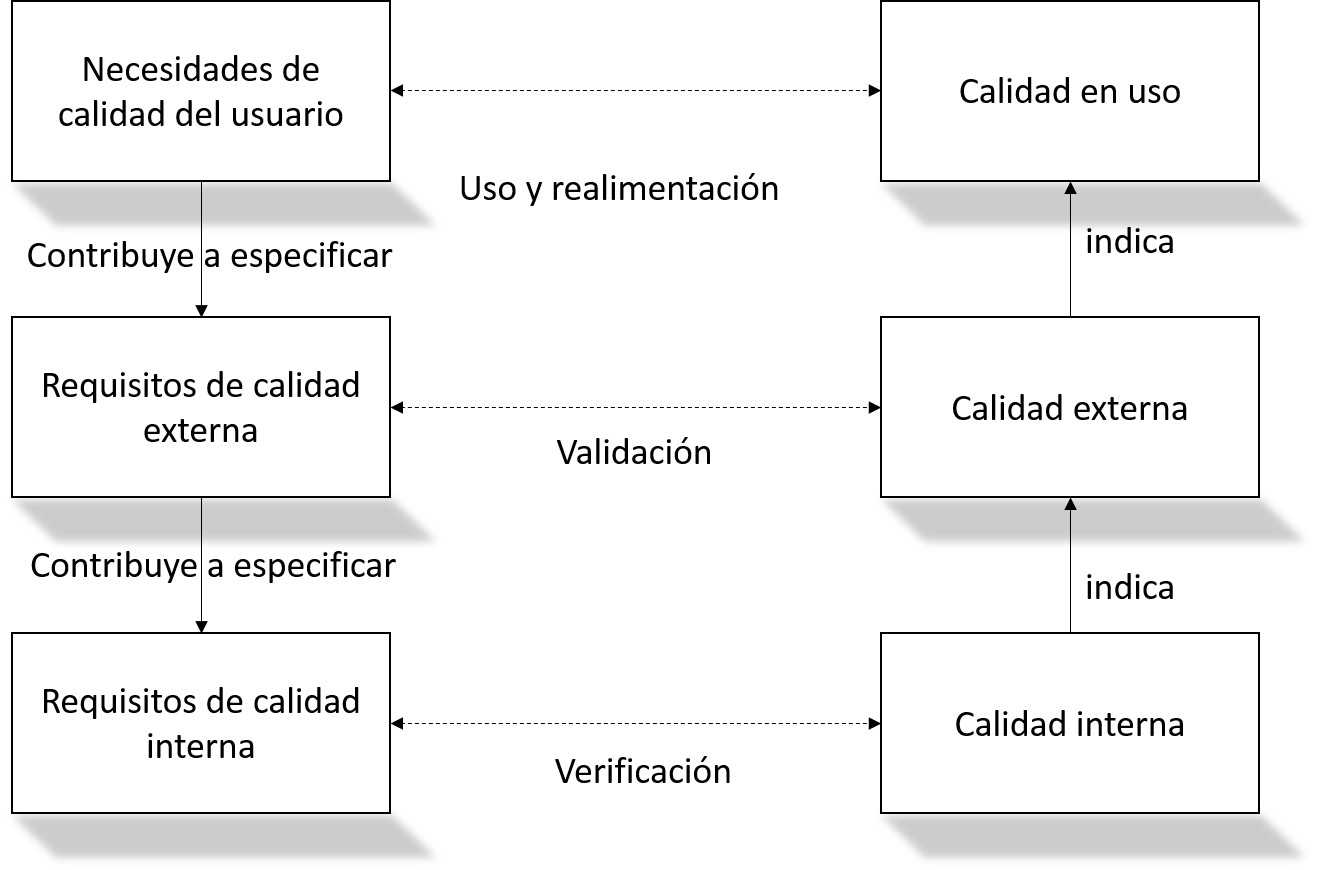


Figura 3 – 1 Calidad del producto y métricas asociadas.

**Fuente:** Norma NTE INE\_ISO/IEC 9126-1

***Necesidades de calidad del usuario*** Estos requisitos especificados por métricas se deberían usar como criterios cuando se valida el producto. El conseguir un producto que satisfaga las necesidades de los usuarios requiere normalmente un enfoque iterativo del desarrollo del software, con retroalimentación continua desde la perspectiva del usuario. (En la Norma ISO 13407 se dan guías sobre procesos de diseño para sistemas interactivos.)

***Requisitos de calidad externa*** Los requisitos de calidad externa para todas las características de calidad definidas en esta parte de la Norma ISO/IEC 9126 se deberían definir en la especificación de requisitos de calidad usando métricas externas, se deberían transformar en requisitos de calidad interna, y se deberían usar como criterios cuando se evalúe el producto.

***Requisitos de calidad interna*** Los requisitos de calidad interna se pueden usar como objetivos para la validación en varias etapas del desarrollo. Se pueden usar también para definir estrategias de desarrollo y criterios para la evaluación y verificación durante el desarrollo. Esto podría incluir el uso de métricas adicionales (por ejemplo para reusabilidad).

***Calidad Interna*** La calidad interna se mide y evalúa frente a los requisitos de calidad interna. Ciertos detalles de la calidad interna se pueden mejorar durante la implantación, revisión y pruebas de código, pero la naturaleza fundamental de la calidad del producto software representada por la calidad interna, permanece sin cambios a no ser que se diseñe de nuevo.

***Calidad externa estimada (o pronosticada)*** es la calidad que se estima o predice para el producto software final en cada etapa del desarrollo para cada característica de calidad, basada en el conocimiento de la calidad interna.

***Calidad externa*** Esta es la calidad cuando el software se ejecuta, y se mide y evalúa normalmente durante las pruebas en un entorno simulado con datos simulados y usando métricas externas. Durante las pruebas, se deberían encontrar y corregir muchos fallos. Sin embargo, algunos fallos pueden permanecer aun después de las pruebas. Dado que la arquitectura u otros aspectos fundamentales del diseño son difíciles de corregir, estos aspectos fundamentales del diseño normalmente permanecen sin cambios a lo largo de las pruebas.

***Calidad en uso estimada (o pronosticada)*** es la calidad que se estima o predice para el producto software final, en cada etapa del desarrollo para cada característica de calidad en uso, y basado en el conocimiento de la calidad interna y externa.

***Calidad en uso*** es la visión de calidad que tiene el usuario del producto software cuando lo usa en un entorno y en un contexto de uso específico. Mide más el grado en que los usuarios pueden alcanzar sus objetivos en un entorno en particular, que las propiedades del software en sí. ***(NTE INEN-ISO/IEC 9126-1. INGENIERIA DEL SOFTWARE. CALIDAD DEL PRODUCTO SOFTWARE. MODELO DE CALIDAD)***

## **JSP**

### ***Concepto***

Una página Java en servidor (JSP, acrónimo en inglés JavaServer Pages) es una plantilla para una página web que emplea código java para generar un documento HTML dinámicamente. Las paginas JSP se ejecutan en un componente del servidor conocido como contendor de JSP, que las traduce a JAVA servlets equivalentemente. Las paginas JSP tienen todas las ventajas de los servlets.

### ***Ventajas***

* Se vuelven a compilar automáticamente cuando es necesario.
* Como están en el espacio común de documentos del servidor web, dirigirse a ellas es más fácil que dirigirse a los servlets.
* Como las páginas JSP son similares al HTML, tienen mayor compatibilidad con las herramientas de desarrollo web.

### ***Como funciona JSP***

La página JSP pasa por 3 etapas en la evolución de su código:

* **Código fuente JSP**.- Este código es el que realmente escribe el desarrollador, se encuentra en un archivo de texto con extensión .jsp y consiste en una mezcla de código de plantilla HTML instrucciones en lenguaje JAVA, directivas JSP y acciones que describen como generar una página WEB para dar servicio a una petición concreta.
* **Código fuente JAVA**.- El contenedor de JSP traduce el código fuente JSP al código fuente de un servlet Java equivalente. Este código fuente se guarda en un área de trabajo y suele ser útil en el proceso de depuración de errores.
* **Clase JAVA compilada**.- Como cualquier otra clase Java, el código del servlet generado se compila en código de bytes en un archivo .class, preparado para ser cargado ejecutado. (**McGRAW, H.** JSP Manual de referencia. Madrid-España: MMI Companies, 2002, 47-48.)

**La figura 4 – 1** Muestra el flujo completo desde la petición de la página JSP hasta la obtención de la respuesta.

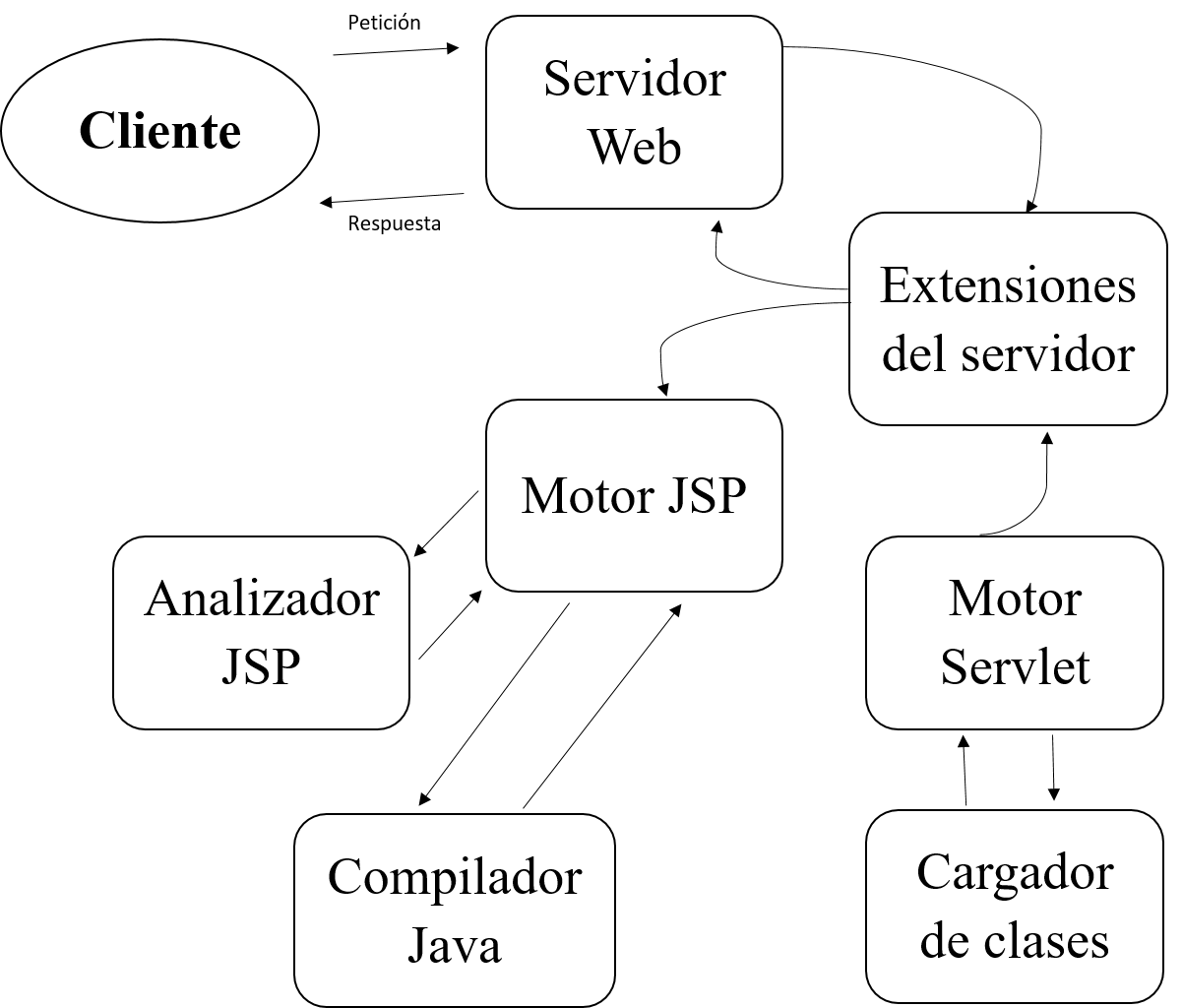


Figura 4 – 1 Petición de página JSP y obtención de respuesta.

**Fuente: Juan, O. C.** Java Server PAges. Madrid-España: ALFAOMEGA, 2002.

## **Servlet**

Son clases Java que amplían la funcionalidad de un servidor Web mediante la generación dinámica de páginas Web. Un entorno de ejecución denominado motor de servlets administra la carga y descargar del servlet, y trabaja con el servidor web para dirigir peticiones a los servlets y enviar la respuesta a los clientes.

***Ventajas clave:***

* **Rendimiento:** Las servlets se cargan cuando los solicitamos por primera vez y permanecen indefinidamente en la memoria, el motor de servlets carga un solo ejemplar o instancia de la clase servlet y le lanza peticiones empleando un conjunto de subprocesos disponibles.
* **Simplicidad:** Los servlets se ejecutan en una máquina virtual en un entorno de servidor controlado y solo necesitan el HTTP basicbásico comunicarse con sus clientes.
* **Sesiones HTTP:** Aunque los servidores HTTP no tienen capacidad para recordar detalles de una petición previa del mismo cliente, la interfaz API servlet proporciona una clase HttpSesion que permite superar esta limitación.
* **Acceso a la tecnología JAVA:** Al ser aplicaciones JAVA, los servlets tienen acceso directo a toda la gama de características JAVA, como el uso de subprocesos, acceso a redes y conectividad a base de datos.

(**McGRAW, H.** JSP Manual de referencia. Madrid-España: MMI Companies, 2002, 47-48.)

## **Bootstrap**

### ***Concepto***

Es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “diseño responsivo” o diseño adaptativo.

El beneficio de usar diseño responsivo en un sitio web, es principalmente que el sitio web se adapta automáticamente al dispositivo desde donde se acceda al sitio web.

(Jorge,L., 2013, <http://jorgelessin.com>)

### ***Ventajas***

* Utiliza componentes y servicios creados por la comunidad web, tales como: HTML5 shim, Normalize.css, OOCSS (CSS orientado a objetos), jQuery UI, LESS y GitHub.
* Existe una enorme comunidad que soporta este desarrollo y cuenta con implementaciones externas como WordPress, Drupal, SASS o jQuery UI.
* Es una herramienta sencilla y ágil para construir sitios web e interfaces.
* Existe mucha variedad de plantillas y temas.

### ***Desventajas***

* *Aprendizaje*: Es necesario adaptarse a su forma de trabajo, si bien su curva de aprendizaje es liviana, deberás comprender y familiarizarte con su estructura y nomenclatura.
* *Adaptación*: Debes adaptar tu diseño a un grid de 12 columnas, que se modifican según el dispositivo. Aquí empiezan los problemas, Bootstrap por defecto te trae anchos, márgenes y altos de línea, y realizar cambios específicos es por decir, un poco tedioso.
* *Mantenimiento*: Es complicado, cambiar de versión si has realizado modificaciones profundas sobre el core.
* *Ampliar* *componentes*: Si necesitas añadir componentes que no existen, debes hacerlos tú mismo en CSS y cuidar de que mantenga coherencia con tu diseño y cuidando el responsive.
* *Pesado*: No es ligero, y además, para algunas funcionalidades, será necesario tener que usar JavaScript y jQuery.

(José,A., 2015, <http://programacion.jias.es>)

## **Java Script**

### ***Concepto***

Javascript es un lenguaje de programación utilizado para crear programas pequeños encargados de realizar acciones dentro del ámbito de una página web.

No hay que confundir Java con Javascript, Java es un lenguaje completo que permite crear aplicaciones independientes, mientras que JavaScript es un lenguaje que funciona como extensiones del HTML.

Tiene el objetivo de integrarse a HTML y facilitar la creación de páginas interactivas sin necesidad de utilizar scripts de CGI o Java. Con Javascript podemos crear efectos especiales en las páginas y definir interacciones con el usuario.

El navegador del cliente es el encargado de interpretar las instrucciones Javascript y ejecutarlas para realizar estos efectos e interacciones, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador.

No hay un software específico para programar en javascript, se puede utilizar cualquier editor de texto. Si lo deseas puedes utilizar inclusive el block de notas de Windows u otros programas pensados en crear páginas web.

* Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez.

(**Juan, O. C.** Diseño de páginas web interactivas con JavaScript. México: Alfaomega, 2000)

### ***Ventajas***

* Lenguaje seguro y fiable.
* El código javascript se ejecuta en el cliente, usando su navegador, por ello no requiere realizar solicitudes al servidor.

### ***Desventajas***

* Los script tienen capacidades limitadas, no todo es posible de realizar ya que para ciertas necesidades se debe usar en conjunto con otros lenguajes de mayor amplitud como Java.
* El código script es visible y puede ser leído por cualquiera.
* El código script debe descargarse completamente antes de ejecutarse (Angel,E., 2011, (<http://computacionudla.blogspot.com>)

### ***Formas de Usar JavaScript***

* La primera forma de usar JavaScript dentro de una página web es embebiendo directamente el código JavaScript dentro del código HTML. Vamos a poner un ejemplo de código JavaScript pero no vamos a entrar en detalle a comentarlo ni explicarlo ya que en este curso no vamos a explicar JavaScript. Simplemente queremos saber interpretar por qué dentro de un documento HTML pueden aparecer fragmentos de código escrito en otro lenguaje.

Ejemplo:

<html>

<head>

<title>Embeber JavaScript – aprenderaprogramar.com</title>

</head>

<body>

<script type="text/javascript">

document.write('Hola Mundo');

</script>

</body>

</html>

* La segunda forma para usar JavaScript es definir JavaScript en un archivo externo. Los archivos de JavaScript son archivos de texto guardados con la extensión .js. Veamos un ejemplo: imaginemos que tenemos un archivo denominado mensaje.js con código JavaScript:

alert('Mensaje JavaScript');

Ahora, para incluir dicho archivo en la una página web utilizaríamos el siguiente código:

<script type="text/javascript" src="archivo.js"></script>

Este código se escribiría dentro de las etiquetas <head> y </head> del documento HTML en el que vayamos a insertarlo.

<html>

<head>

<meta charset="utf-8">

<title>Embeber JavaScript – aprenderaprogramar.com</title>

<script type="text/javascript" src="archivo.js"></script>

</head>

<body>

Página web de prueba – aprenderaprogramar.com

</body>

</html>

NOTA: Tal y como hemos pues la ruta, el archivo.js debe estar en el mismo directorio que el fichero HTML. Si estuviera en otro directorio, habría que especificar la ruta. (Enrique, G., 2006, http://aprenderaprogramar.com)

## **Arquitectura 3 Capas**

### ***Concepto***

Es un estilo de programación, su objetivo primordial es la separación de la capa de presentación, capa de negocio y la capa de datos.

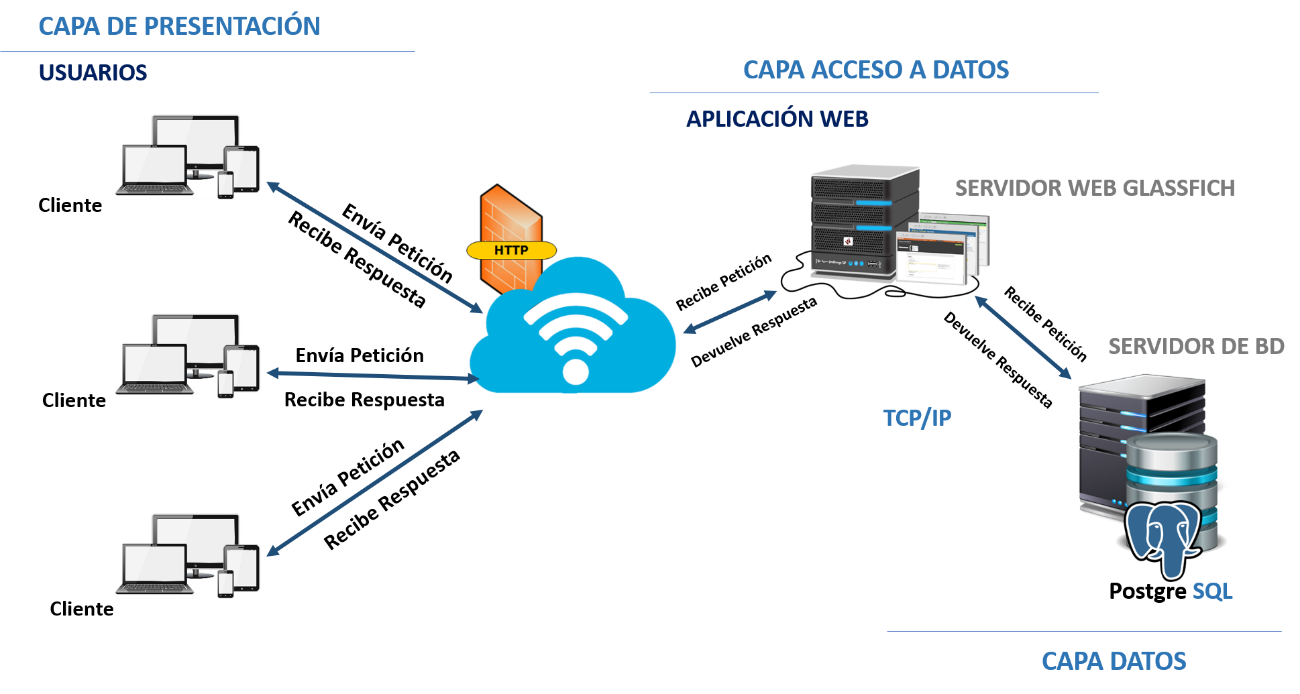


Figura 5 – 1 Arquitectura de la aplicación

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

### ***Capas***

#### **Capa de presentación**

Esta capa es la que ve el usuario, presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso.

Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser amigable; para el usuario generalmente se presentan como formularios.

#### **Capa de negocio**

Aquí es donde, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. S e denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

* Toda aplicación tiene código para implementar reglas de negocios.
* Se puede seleccionar almacenar la lógica de negocios sobre cada estación de cliente, u optar por ejecutar la lógica de negocios sobre un servidor de aplicaciones.
* No toda la lógica de negocio es la misma algunas no requieren un frecuente acceso a los datos, pero una interface de usuario robusta necesitara de la lógica de negocios para la validación en la entrada de campos, cálculos en tiempo real u otras interacciones de usuarios.

#### **Capa de datos**

Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. (Fani, C., 2008, <http://es.slideshare.net>)

## **iReport**

### ***Concepto***

Es un constructor / diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, sub-informes, etc.

### ***Características***

* 100% escrito en JAVA y además OPENSOURCE y gratuito.
* Maneja el 98% de las etiquetas de JasperReports
* Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, sub-reportes.
* Soporta internacionalización nativamente.
* Recopilador y exportador integrados.
* Soporta JDBC.
* Facilidad de instalación. (Cristhian, H. 2007, <http://www.adictosaltrabajo.com>)

## **PostgreSQL**

### ***Concepto***

Es un sistema avanzado de administración de base de datos objeto-relacionales de código abierto, funciona bajo los distintos sistemas operáticos.

Características:

* Está basado en lenguaje C
* Almacenamiento confiable y robusto
* Manipulación potente flexible y eficiente.
* Considerado el competidor de Oracle
* Debe ser instalado en un sistema de archivos NTFS

### ***Ventajas***

* Menor coste de mantenimiento
* Mayor seguridad
* Menores tiempos de desarrollo
* Bajo costo
* Instalación ilimitada
* Multiplataforma (**Maribel, S, M.** PHP 5 con PostgreSQL8. Perú: Megabyte, 2006)

### ***Desventajas***

* En comparación con MySQL es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tiene MySQL.
* Existen foros oficiales pero la ayuda no es obligatorio.

<http://postgresql-dbms.blogspot.com>)

## **PgAdmin 3**

### ***Concepto***

Es una aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL. La aplicación se puede utilizar para manejar postgreSQL 7.3 y superiores y funciona sobre casi todas las plataformas. Este software fue diseñado para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL a la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita la administración.

### ***Características***

En pgAdmin3 se puede ver y trabajar con casi todos los objetos de la base de datos, examinar sus propiedades y realizar tareas administrativas.

* Agregados.
* Columnas.
* Constraints.
* Bases de datos.
* Funciones.
* Grupos.
* Índices.
* Lenguajes (PLpgsql, PLpython, PLperl, etc.).
* Clases de operadores.
* Operadores.
* Servidores PostgreSQL.
* Reglas.
* Esquemas.
* Secuencias.
* Tablas.
* Triggers.
* Tipos de datos.
* Usuarios.
* Vistas. (Ecured, 2016, <http://www.ecured.cu>)

## **Aplicación WEB**

### ***Concepto***

Es cualquier aplicación que es accedida vía web por una red como internet o una intranet. En general, el término también se utiliza para designar aquellos programas informáticos que son ejecutados en el entorno del navegador o codificado con algún lenguaje soportado por el navegador (como JavaScript, combinado con HTML); confiándose en el navegador web para que reproduzca (renderice) la aplicación.

### ***Ventajas***

* Las aplicaciones web requieren poco o nada de espacio en disco. Además suelen ser livianas.
* No requieren que los usuarios las actualicen, eso es implementado del lado del servidor.
* Proveen gran compatibilidad entre plataformas (portabilidad), dado que operan en un navegador web.

### ***Desventajas***

* Las aplicaciones web requieren navegadores web totalmente compatibles para funcionar. Incluso muchas veces requieren las extensiones apropiadas y actualizadas para operar.
* Muchas veces requieren una conexión a internet para funcionar, si la misma se interrumpe, no es posible utilizarla más. De todas maneras, en ocasiones, pueden ser descargadas e instaladas localmente para su uso offline.
* Muchas no son de código abierto, perdiendo flexibilidad.
* La aplicación web desaparece si así lo requiere el desarrollador o si el mismo se extingue. Las aplicaciones tradicionales, en general, pueden seguir usándose en esos casos.
* El usuario, en general, no tiene libertad de elegir la versión de la aplicación web que quiere usar. Un usuario podría preferir usar una versión más antigua, hasta que la nueva sea probada.
* En teoría, el desarrollador de la aplicación web puede rastrear cualquier actividad que el usuario haga. Esto puede traer problemas de privacidad.

(Alegsa, 2010, <http://www.alegsa.com.ar>)

## **Ubuntu Server**

### ***¿Qué es?***

Ubuntu Server es una variante de Ubuntu que se genera con cada versión y está dedicada especialmente para su uso en servidores. El uso de Ubuntu como servidor se ha extendido mucho en los últimos años, sobre todo para el uso de servidores web, de un modo tanto particular como profesional.

Un servidor es una máquina que nos proporciona algún servicio. Pueden ser de diferentes tipos, servidor web, servidor de base de datos, servidor de archivos, u otras diferentes funciones, incluso varias a la vez. No tienen por qué ser grandes y potentes máquinas, podemos tener montado un servidor en casa en un ordenador antiguo, que nos sirva para tener guardados todos nuestros datos importantes y acceder a ellos desde cualquier otro ordenador o dispositivo desde nuestra casa, o incluso desde cualquier lugar.

Ubuntu Server es un Sistema Operativo sin entorno gráfico (aunque podemos instalarlo) lo que quiere decir que todas las acciones se realizan mediante consola, y normalmente ni si quiera a través del propio servidor, sino desde una conexión remota. El manejo de Ubuntu Server es muy similar al de cualquier otro Sistema Linux, pero con las particularidades de Ubuntu (como el sudo).

(Ubuntu Fácil, 2013, <http://www.ubuntufacil.com>)

### ***Requisitos para instalar Ubuntu Server***

Los requisitos para una versión server Linux son mínimos debido a que no utiliza el entorno gráfico, pero para que actúe como servidor dependiendo del tráfico que tengamos puede requerir más.

*Mínimo (Consola)*

* 256 MB de memoria
* 2 Gb de espacio en HDD (Incluido swap)
* AMD o Intel Procesador de 64-32bits
* Incluido AMD Optaron e Intel EM64T Xeon, para versiones de 64.

*Mínimo (Gráfico)*

* 512 MB de memoria
* 4 Gb de espacio en HDD (Incluido swap)
* AMD o Intel Procesador de 64-32bits
* Tarjeta Gráfica VGA, monitor con resolución de 800x600

(Javier, 2011, http://isft179-ubuntuserver.blogspot.com)

# **CAPITULO II**

## Desarrollo de una Aplicación Web utilizando la norma NTE INEN-ISO/IEC 9126-1 para automatizar el cobro del servicio de Agua Potable en la Provincia de Morona Santiago

## **Visión y alcance.**

### ***Definición del problema.***

El Municipio del Cantón Logroño es una institución pública que se encuentra ubicada en el Cantón Logroño, provincia Morona Santiago, teniendo como objetivo fundamental el brindar los servicios básicos a los ciudadanos de su localidad.

Actualmente el manejo correcto de los recursos del estado, así como el libre acceso a la información hace que principalmente las instituciones públicas manejen estos recursos de forma transparente. Esto lleva a que hagan uso de nuevas tecnologías e internet como medio de comunicación y negocio, entre estos y la población.

Al momento la institución ofrece su servicio de facturación de agua potable haciendo uso de un sistema de escritorio que es limitado en cuanto a los servicios que ofrece.

Tomando en cuenta el antecedente anterior se propone el desarrollo de una aplicación WEB que automatice y mejore los procesos que se realizan durante el proceso de facturación y cobro del servicio de agua potable.

### ***Visión del proyecto.***

Nuestro objetivo principal es entregar una aplicación WEB que funcione adecuadamente para cubrir las necesidades durante el proceso de cobro del servicio de agua potable, además de ofrecer a la población un método rápido y cómodo para que estos puedan conocer y enterarse sobre cuál es su deuda con la institución, cuál es su historial de consumo y los pagos realizados a lo largo del uso del servicio.

### ***Perfiles de Usuario***

*Administrador*

Profesional que posee un título de educación superior, conocimientos básicos de computación e informática y conocimiento en el desarrollo de aplicación bajo el lenguaje de programación JAVA.

Funciones:

* Revisar los datos ingresados por todos los usuarios.
* Modificar los datos que requieran modificación.
* Eliminar información.
* Tener acceso a todas las funciones de la aplicación: Crear, actualizar.
* Creación de perfiles de usuario.
* Creación de usuarios.

*Ejecutivo*

Persona que posea conocimiento en el uso de herramientas informáticas.

Funciones:

* Crear cuentas a nuevos clientes que deseen acceder al servicio de agua.
* Emisión de reportes específicos
* Actualizar información de los clientes
* Actualizar información de los medidores

*Cajero*/*a*

Persona que posea conocimiento sobre el uso de herramientas informáticas y de contabilidad.

Funciones:

* Realizar el cobro del servicio.
* Facturar
* Visualización y emisión de reportes específicos.

*Operador de agua*

Persona con conocimientos en el uso de aplicaciones informáticas y aplicaciones móviles.

Funciones:

* Ingreso de información a la aplicación (consumo del servicio).

### ***Ámbito del proyecto.***

La aplicación a desarrollarse para el Municipio del Cantón Logroño tiene como nombre SIFACOAG (Sistema de Facturación y Control del Agua Potable) la cual es una aplicación de apoyo durante el proceso de control y facturación del servicio de agua potable. Para cubrir todas las funciones que se realizan durante este proceso en la institución SIFACOAG brinda una plataforma para que los usuarios operadores de agua pueden ingresar de forma directa información al sistema evitan tergiversación por pare de terceros y para que los clientes del servicio puedan acceder al historial de su consumo así como a las deudas pendientes que tiñen con la institución.

Al hacer uso de SIFACOAG la primera imagen que observaremos será la de bienvenida a la aplicación junto a esto tendremos las opciones de autenticación para el ingreso al sistema, en el cual dependiendo de qué usuario sea el que se logue se presentara un menú con las opciones pertinentes a cada perfil, así como la opción para los clientes en donde estos pueden realizar las consultas que crean pertinentes y necesarias para ellos.

### ***Herramientas a utilizar.***

Para el desarrollo de la aplicación se hará uso de las herramientas:

Tabla 1 – 2 Herramientas a utilizar en SIFACOAG

|  |  |
| --- | --- |
| **HERRAMIENTA** | **CARACTERÍSTICAS** |
| Netbeans 8.0.2 | IDE de desarrollo |
| Java | Lenguaje de programación |
| JSP | Tecnología que permite mezclar html estático con html dinámico |
| PostgreSQL | Motor de base de datos |
| Microsoft Windows | Sistema Operativo sobre el cual correrá la aplicación |
| iReport | Plugin para el diseño de reportes |
| Jasperreport | Librerías para el funcionamiento de los reportes |
| Lenguaje Unificado de Modelado UML | Lenguaje de modelado |
| Microsoft Solutions Framework MSF | Metodología para el desarrollo de software |
| DeZign for Databases | Herramienta para el diseño y modelado de la base de datos |
| Bootstrap | Framework CSS |
| Java Script | Crea efectos dinámicos en las páginas web |
| NTE INEN-ISO/IEC 9126-1 | Norma para determinar la calidad del software |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**Netbeans**

Es una Herramienta que se utiliza para desarrollar aplicaciones Web,

Móvil y de Escritorio para diferentes lenguajes de programación como son Java, C++, Ruby y PHP entre otros. Es de código abierto, es multiplataforma, multilenguaje, contiene servidores web y es fácil de instalarlo e utilizarlo. (Jiménez, 2014)

**Java**

Es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

**JSP**

Java Server Pages (JSP) es una tecnología que nos permite mezclar HTML estático con HTML generado dinámicamente. Muchas páginas Web que están construidas con programas CGI son casi estáticas, con la parte dinámica limitada a muy pocas localizaciones. Pero muchas variaciones CGI, incluyendo los servlets, hacen que generemos la página completa mediante nuestro programa, incluso aunque la mayoría de ella sea siempre lo mismo.

**PostgreSQL**

Es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de datos.

Se ejecuta en los principales sistemas operativos que existen en la actualidad como:

* Linux
* UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64)
* Windows

**Microsoft Windows**

Es un sistema operativo, es decir, un conjunto de programas que posibilita la administración de los recursos de una computadora. Este tipo de sistemas empieza a trabajar cuando se enciende el equipo para gestionar el hardware a partir desde los niveles más básicos.

**iReport**

Es una herramienta visual que sirve para generar ficheros XML (plantillas de informes) utilizando la herramienta de generación de informes JasperReport.

Características de IReport:

La lista siguiente describe algunas de las características importantes de IReport:

* 100% escrito en Java y además OpenSource y gratuito.
* Maneja el 98% de las etiquetas de JasperReport.
* Permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, subreports (subreportes).
* Soporta internacionalización nativamente.
* Browser de la estructura del documento.
* Recopilador y exportador integrados.
* Soporta JDBC.
* Soporta JavaBeans como orígenes de datos (éstos deben implementar la interface JRDataSource).
* Incluye Wizard’s (asistentes) para crear automáticamente informes.
* Tiene asistentes para generar los subreportes.
* Tiene asistentes para las plantillas.
* Facilidad de instalación.

**JasperReport**

Es una herramienta de creación de informes que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML.

Su principal propósito es crear documentos de tipo páginas, preparados para imprimir en una forma simple y flexible. Se usa comúnmente con IReport, un front-end gráfico de código abierto para la edición de informes. Está bajo GNU Lesser General Public License, por lo que es Software libre.

**UML (Lenguaje Unificado de Modelado)**

Es una herramienta propia de personas que tienen conocimientos relativamente avanzados de programación y es frecuentemente usada por analistas funcionales (aquellos que definen qué debe hacer un programa sin entrar a escribir el código) y analistas-programadores (aquellos que dado un problema, lo estudian y escriben el código informático para resolverlo en un lenguaje como Java, C#, Python o cualquier otro).

**MSF Microsoft Solutions Framework**

Es un enfoque personalizable para entregar con éxito soluciones tecnológicas de manera más rápida, con menos recursos humanos y menos riesgos, pero con resultados de más calidad. MSF ayuda a los equipos a enfrentarse directamente a las causas más habituales de fracaso de los proyectos tecnológicos y mejorar así las tasas de éxito, la calidad de las soluciones y el impacto comercial.

MSF se centra en:

* Alinear los objetivos de negocio y de tecnología
* Establecer de manera clara los objetivos, los roles y las responsabilidades
* Implementar un proceso iterativo controlado por hitos o puntos de control
* Gestionar los riesgos de manera proactiva
* Responder con eficacia ante los cambios

**DeZign for Databases**

DeZign for Databases es una herramienta de diseño intuitivo para bases de datos para desarrolladores y DBA's que puede ayudarlo a modelar, crear y mantener bases de datos. El software usa diagramas de entidades de relacionamiento (ERD) para bases de datos gráficamente y genera automáticamente las más populares bases de datos y PC's en SQL.

**Bootstrap**

Un entorno de desarrollo con una serie de recursos que simplifican el desarrollo de un proyecto web con html5, css3 y Jquery, de manera que simplifica mucho el trabajo a la hora de diseñar, ya que el framework bootstrap ya tiene una buena parte del trabajo hecho lo cual simplifica mucho la tarea del desarrollo.

Ventajas:

* Diseño responsivo y Mobile First lo cuál es muy importante.
* Una gran comunidad de desarrolladores en Git hub para dar soporte, por lo que hacen a Bootstrap un entorno de trabajo más robusto que otros frameworks.
* La integración de Html5 y css3 lo cuál lo hace muy poderoso y por tanto mucho más ligero de cara a los navegadores, esto se traduce en una ventaja respecto al SEO de las páginas realizadas con bootstrap.
* LESS es un procesador para css que lo hace mucho más potente y eficaz, ayudando a aligerar el código.

**Java Script**

Es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente (es decir, se ejecuta en nuestro ordenador, no en el servidor) permitiendo crear efectos atractivos y dinámicos en las páginas web. Los navegadores modernos interpretan el código JavaScript integrado en las páginas web.

**PgAdmin3**

Es una aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL. La aplicación se puede utilizar para manejar postgreSQL 7.3 y superiores y funciona sobre casi todas las plataformas. Este software fue diseñado para responder a las necesidades de todos los usuarios, desde la escritura de simples consultas SQL a la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de PostgreSQL y facilita la administración.

**NORMA NTE INEN-ISO/IEC 9126-1**

Los ordenadores están siendo usados cada vez más para la ejecución de diversos tipos de aplicaciones y el correcto funcionamiento de esta aplicaciones determinan el éxito de un negocio así como la seguridad de la información de las personas que usan estas aplicaciones. La evaluación de una aplicación software termina siendo un factor clave para determinar la calidad del software.

Para poder definir esto se puede determinar las características de calidad apropiadas tomando en cuenta cual será el propósito del software. Para determinar la calidad del software se hace uso de métricas que hayan sido validadas y aceptadas.

Por ello que el uso de la norma NTE INEN-ISO/IEC 9126-1 beneficia al desarrollo de aplicaciones ya que esta norma nos brinda las métricas para determinar si un software está o no cumpliendo con un grado de calidad adecuado para el uso del mismo.

### ***Objetivos del proyecto***

*Objetivo General*

“Desarrollar un sistema web utilizando la norma NTE INEN-ISO/IEC 9126-1 para automatizar el servicio de cobro de agua.

*Objetivos específicos*

* Evaluar la norma NTE INEN-ISO/IEC 9126-1.
* Identificar los procesos que formen parte del cobro del servicio de agua potable.
* Desarrollar los módulos de registros, gestión y reportes de la aplicación web.
* Tomar como caso práctico el Municipio del Cantón Logroño.

### ***Análisis y gestión del riesgo***

El factor crítico en el desarrollo de una aplicación es el surgimiento de un problema que afecte la etapa de desarrollo de la aplicación, por ello es necesario tener un plan que facilite la solución de estos problemas y que estos no lleguen a afectar.

**NOMENCLATURA UTILIZADA:**

**RP:** Riesgo del proyecto

**RT:** Riesgo técnico

**RN:** Riesgo del negocio

**La tabla 2 – 2** muestra los riegos que pueden afectar el desarrollo del proyecto, retrasar este o hacer que este se cancele.

Tabla 2 – 2 Riesgos

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **DETALLE** | **TIPO RIESGO** | **CONSECUENCIA** |
| **R1** | Los usuarios cambian continuamente los requisitos | RP | Aumento de costos |
| **R2** | La base de datos está mal diseñada | RP | Retraso en la entrega del proyecto |
| **R3** | No existe disponibilidad de la tecnología | RT | Desarrollo en tecnología inadecuada |
| **R4** | Los equipos donde se desarrolla el sistema se dañan y no existe acceso a la información | RP | Cancelación del proyecto |
| **R5** | Falta de conocimiento en las herramientas | RT | Mala aplicación del proyecto |
| **R6** | Rotación de personal de desarrollo | RN | Retraso en la entrega del proyecto |
| **R7** | Menos utilización del software | RN | Desarrollo en vano |
| **R8** | Cambio de responsable del Proyecto | RN | Retraso en la entrega del proyecto |
| **R9** | Incumplimiento con el diagrama de actividades y sus tiempos. | RP | Retraso en la entrega del proyecto. |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Los siguientes parámetros nos permitirán determinar el impacto de los riesgos, esto se indica en la **tabla 3 – 2.**

Tabla 3 – 2 Determinación del impacto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **IMPACTO** | **RETRASO** | **IMPACTO TÉCNICO** | **IMPACTO DEL COSTO** | **VALOR** |
| Bajo | 5 días | Ligero efecto en el desarrollo del proyecto. | <1% | 1 |
| Moderado | 10 días | Moderado efecto en el desarrollo del proyecto. | <5% | 2 |
| Alto | 15 días | Severo efecto en el desarrollo del proyecto. | <10% | 3 |
| Critico | 20 días | Proyecto no puede ser culminado. | >=10% | 4 |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Para determinar cuál es la probabilidad de ocurrencia de un riesgo se determinado los parámetros mostrados en la **tabla 4 – 2.**

Tabla 4 – 2 Probabilidad que ocurra

|  |  |  |
| --- | --- | --- |
| **RANGO DE PROBABILIDAD** | **DESCRIPCIÓN** | **VALOR** |
| 1% - 33% | Baja | 1 |
| 33% - 66% | Media | 2 |
| 67% - 99% | Alta | 3 |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**Las tablas 5 – 2 y 6 - 2** Determinan y categorizan la exposición de cada uno de los riesgos.

Tabla 5 – 2 Determinación de la exposición del riesgo

|  |  |  |
| --- | --- | --- |
| **EXPOSICIÓN** | **VALOR** | **COLOR** |
| Baja | 1 o 2 | Verde |
| Media | 3 o 4 | Amarillo |
| Alta | Mayor a 6 | Rojo |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 6 – 2 Código de colores de impacto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Probabilidad** | **Bajo = 1** | **Moderado = 2** | **Alto = 3** | **Crítico = 4** |
| Baja | 3 | 6 | 9 | 12 |
| Media | 2 | 4 | 6 | 8 |
| Alta | 1 | 2 | 3 | 4 |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### **Desarrollo del análisis de riesgos**

El analista define cual será la probabilidad de ocurrencia e impacto de cada uno de los riesgos, tomando en cuenta los parámetros indicados en las tablas 4 – 2 y 5 - 2. Luego se determina la exposición y la prioridad tomando en cuenta los parámetros de las tablas 6 - 2, Teniendo como resultado la **tabla 7 - 2.**

Tabla 7 – 2 Análisis de Riesgo

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **PROBABILIDAD** | | | **IMPACTO** | | **EXPOSICIÓN** | | **PRIORIDAD** |
| % | Valor | Probabilidad | Impacto | Valor | Valor | Exposición |
| R9 | 90% | Alta | 3 | Alto | 3 | 9 | Alta | 1 |
| R4 | 50% | Media | 2 | Crítico | 4 | 8 | Alta | 2 |
| R1 | 70% | Alta | 3 | Moderado | 2 | 6 | Alta | 3 |
| R6 | 10% | Baja | 1 | Crítico | 4 | 4 | Media | 4 |
| R5 | 10% | Baja | 1 | Alto | 3 | 3 | Media | 5 |
| R3 | 20% | Baja | 1 | Alto | 3 | 3 | Media | 6 |
| R8 | 30% | Baja | 1 | Moderado | 2 | 2 | Baja | 7 |
| R7 | 40% | Media | 2 | Bajo | 1 | 2 | Baja | 8 |
| R2 | 65% | Media | 2 | Bajo | 1 | 2 | Baja | 9 |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### **Línea de corte**

**En la tabla 8 - 2** podemos observar los riesgos con mayor exposición entendiendo que estos requieren una pronta gestión.

Tabla 8 – 2 Línea de corte

|  |  |
| --- | --- |
| **IDENTIFICACIÓN** | **EXPOSICIÓN** |
| R9 | 9 |
| R4 | 8 |
| R1 | 6 |
| R6 | 4 |
| R3 | 3 |
| R5 | 3 |
| R2 | 2 |
| R7 | 2 |
| R8 | 2 |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**La tabla 9 - 2 hasta la tabla 11 - 2** nos permiten observar los riesgos que se han considerado como prioritarios, permitiendo conocer cuáles serían las causas y consecuencias de cada uno de estos así como la manera de gestionarlos.

Tabla 9 – 2 Riesgo 9

|  |  |  |  |
| --- | --- | --- | --- |
| **HOJA DE GESTION DEL RIESGO** | | | |
| **ID DEL RIESGO:** R9 | | **FECHA:** | |
| **Probabilidad:** Alta  **Valor:** 3 | **Impacto:** Alto  **Valor:** 3 | **Exposición:** Alta  **Valor:** 9 | **Prioridad:** 1 |
| **DESCRIPCION:** Incumplimiento con el diagrama de actividades y sus tiempos. | | | |
| **REFINAMIENTO:**  **Causas:** Los integrantes no tomaron el interés necesario o carecen de conocimiento en el uso de las herramientas necesarias para el desarrollo.  **Consecuencias:** Retraso del proyecto o cancelación del mismo. | | | |
| **REDUCCIÓN:** Establecer tareas a realizarse durante el desarrollo, dialogar con los integrantes para determinar el grado de conocimiento que tienen en el uso de las herramientas. | | | |
| **SUPERVISIÓN:** El gestor del proyecto debe realizar controles constantes sobre el cumplimiento de las tareas. | | | |
| **GESTIÓN:** Incentivar al grupo de desarrollo para que estos sientan motivación durante sus labores y cumplan con cabalidad sus tareas. | | | |
| **ESTADO ACTUAL:**   * Fase de reducción iniciada   Fase de Supervisión iniciada  Gestionando el Riesgo | | | |
| **RESPONSABLES:** Eduardo Guadalupe  Saúl Morocho | | | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 10 – 2 Riesgo 4

|  |  |  |  |
| --- | --- | --- | --- |
| **HOJA DE GESTION DEL RIESGO** | | | |
| **ID DEL RIESGO:** R4 | | **FECHA:** | |
| **Probabilidad:** Media  **Valor:** 2 | **Impacto:** Critico  **Valor:** 4 | **Exposición:** Alta  **Valor:** 8 | **Prioridad:** 1 |
| **DESCRIPCION:** Los equipos donde se desarrolla el sistema se dañan y no existe acceso a la información | | | |
| **REFINAMIENTO:**  **Causas:** Debido al mal manejo de los equipos o a que estos son antiguos.  **Consecuencias:** Cancelación del proyecto. | | | |
| **REDUCCIÓN:** Establecer buenas prácticas de uso de los equipos, evitando que estos sean usados en zonas donde sean propensos a sufrir daños. | | | |
| **SUPERVISIÓN:** Realizar mantenimientos mensuales de los equipos tanto en software como en hardware. | | | |
| **GESTIÓN:** Dar un día al mes al grupo de trabajo para que estos se encarguen de realizar el mantenimiento de sus equipos. | | | |
| **ESTADO ACTUAL:**   * Fase de reducción iniciada   Fase de Supervisión iniciada  Gestionando el Riesgo | | | |
| **RESPONSABLES:** Eduardo Guadalupe  Saúl Morocho | | | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 11 – 2 Riesgo 1

|  |  |  |  |
| --- | --- | --- | --- |
| **HOJA DE GESTION DEL RIESGO** | | | |
| **ID DEL RIESGO:** R1 | | **FECHA:** | |
| **Probabilidad:** Alta  **Valor:** 3 | **Impacto:** Moderado  **Valor: 2** | **Exposición:** Alta  **Valor:** 6 | **Prioridad:** 1 |
| **DESCRIPCION:** Los usuarios cambian continuamente los requisitos | | | |
| **REFINAMIENTO:**  **Causas:** No se realizó el correcto análisis de los requerimientos por parte del equipo de desarrollo.  **Consecuencias:** Retraso en el desarrollo y entrega del proyecto. | | | |
| **REDUCCIÓN:** Realizar varias reuniones con los usuarios finales del proyecto, en las cuales se refinaran los requisitos. | | | |
| **SUPERVISIÓN:** El gestor revisara junto con el equipo de desarrollo todos los requerimientos obtenidos y refinaran cada uno. | | | |
| **GESTIÓN:** El gestor revisara durante la etapa de desarrollo si se está cumpliendo con los requerimientos. | | | |
| **ESTADO ACTUAL:**   * Fase de reducción iniciada   Fase de Supervisión iniciada  Gestionando el Riesgo | | | |
| **RESPONSABLES:** Eduardo Guadalupe  Saúl Morocho | | | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

### ***Planificación inicial***

**DEFINICIÓN DEL PLAN BORRADOR**

Sistema de facturación y control de agua potable “SIFACOAG”

Sector solicitante: Municipio del Cantón Logroño – Morona Santiago.

Equipo de trabajo:

Para el correcto desarrollo del proyecto es importante definir los roles de cada uno de los miembros del equipo de desarrollo como se establece en la tabla 12 - 2.

Tabla 12 – 2 Roles de los miembros

|  |  |
| --- | --- |
| **INTEGRANTES** | **CARGOS** |
| Eduardo Guadalupe | Gestor de proyecto, Desarrollador, documentador |
| Saúl Morocho | Gestor de proyecto, Desarrollador, documentador |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Aspectos generales de la empresa.

Nombre de la empresa: Municipio del Cantón Logroño – Morona Santiago

Dirección: Santiago La febre S/N y Av. 10 de Agosto Logroño – Ecuador

Teléfono: 07-3911158

*Misión*:

Al término de los cinco años de mi administración del Gobierno Autónomo Descentralizado el Cantón Logroño será reconocido por ser un paraíso natural y cultural con la mayor cobertura de servicios básicos, la mejor vialidad urbana y rural, promoviendo una administración transparente, optimizando los recursos y mejorando la calidad del gasto y la inversión pública, con una armónica convivencia intercultural, una efectiva coordinación interinstitucional, una amplia participación ciudadana y un convivir armónico con la naturaleza que resultarán en altos estándares de bienestar y Buen Vivir para toda la población del Cantón Logroño.

*Visión*:

Una Administración Descentralizada Municipal, cumpliendo los principios, regulaciones y estándares de calidad para una Administración pública eficiente, enmarcados dentro de la Planificación, Organización, Dirección; Evaluación y Retroalimentación (control) (PODER) comprometidos a la construcción de una sociedad Intercultural más equitativa e incluyente.

*Orgánico estructural*

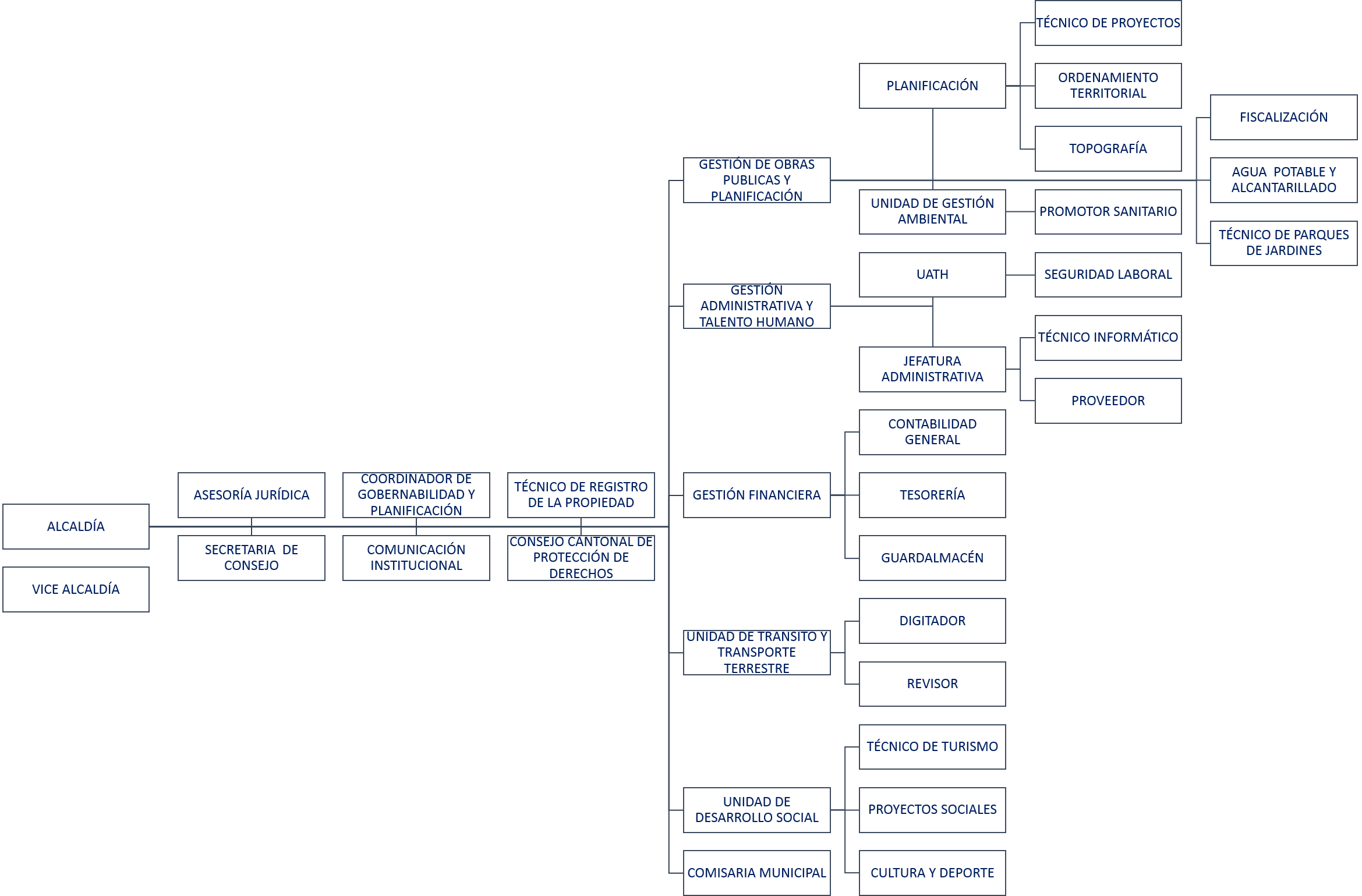


Figura 6 – 2 Orgánico Estructural

**Fuente:** JAIME, N., MUNICIPIO CANTON LOGROÑO, MS., 2016

*Generalidades*

Fuente de recopilación de la información

* Persona que labora en los distintos departamentos del municipio del Cantón Logroño
  + Jaime Naranza – Técnico de Avalúos y Catastros
  + Rosa Lucero - Tesorera
* Internet, libros, artículos facilitaran la comprensión y el uso de las herramientas.

*Técnicas a utilizar*

Técnicas para la recopilación de la información

Entrevistas

* Se realizaran entrevistas con cada una de las personas involucradas en el proyecto, con la finalidad de determinar de forma correcta cada uno de los requerimientos.

Lluvia de ideas

* El equipo de desarrollo realizara reuniones luego de cada una de las entrevistas esto con la finalidad de que cada miembro contribuya con ideas que sean de ayuda para satisfacer las necesidades que el personal del Municipio del Cantón Logroño desea que cumpla la aplicación.

*Técnicas para la representación de actividades del sistema.*

Técnica de estructura PIECES

Esta técnica nos permitirá determinar todos los posibles problemas y necesidades que surgen en la empresa y establecer posibles soluciones para cada uno de estos, de esta manera desarrollar un producto software que cumpla con las expectativas del usuario.

Tabla 13 – 2 DEFINICION DE PIECES

|  |  |
| --- | --- |
| **P** | Necesidad de mejorar las prestaciones (productividad, tiempo de espera) |
| **I** | Necesidad de mejorar respecto a la información (entradas, salidas, datos guardados) |
| **E** | Necesidad de mejorar aspectos económicos |
| **C** | Necesidad de mejorar controles y seguridad |
| **E** | Necesidad de mejorar Eficacia con los equipos y personas |
| **S** | Necesidad de mejorar los servicios |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 14 – 2 ESTRCTURA PIECES

|  |  |  |
| --- | --- | --- |
| **PROBLEMAS** | | **OPRTUNIDADES Y MEJORAS** |
| **P** | Falta en el tiempo de desarrollo | Calendarizar las actividades |
| Eventos/Actividades que afecten el desarrollo. | Priorizar las actividades y objetivos |
| Presiones adversas al desarrollo | Uso de metodologías para el equilibrio mental |
| Falta de experiencia del desarrollador | * Solicitar asesoría * Investigación |
| **I** | Información física inconsistente | Introducir la información al sistema con el usuario |
| Datos de entrega ambiguos | Implementar un buen diseño de base de datos |
| Difícil gestión de la información | * Mostrar la información de manera clasificada y con interfaces visuales amigables * Ingreso de información de manera correcta y coherente |
| **E** | Costos elevado de Licencias de Software | Usar software Libre |
| Recursos económicos insuficientes | Economizar en lo más posible |
| **C** | La información debe estará bien protegida | Crear controles para el flujo del sistema |
| Todos no pueden acceder a la misma información | Crear perfiles y permisos de usuarios en el sistema |
| **S** | Los servicios prestados a los clientes son ineficientes | * Generar información dedicada a los clientes * Facilitar herramientas y recursos para la atención de los clientes |
| Colas largas en la ventanilla de pagos | * Sistema automatizado a prueba de fallos * Herramientas fáciles de usar |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Diagrama de actividades*

Se creara un diagrama de actividades mediante el cual podremos cumplir cada una de las tareas que nos proponemos para de esta forma poder cumplirlos siguiendo procesos y tiempos establecidos para cada uno.

E:\Downloads\Requerimiento1 (1).png

Figura 7 – 2 Requerimiento 1

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

E:\Downloads\Requerimiento2.png

Figura 8 – 2 Requerimiento 2

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

E:\Downloads\Requerimiento3.png

Figura 9 – 2 Requerimiento 3

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

E:\Downloads\Requerimiento4.png

Figura 10 – 2 Requerimiento 4

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

E:\Downloads\Requerimiento5.png

Figura 11 - 2 Requerimiento 5

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

E:\Downloads\Requerimiento6.png

Figura 12 – 2 Requerimiento 6

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

E:\Downloads\Requerimiento7.png

Figura 13 – 2 Requerimiento 7

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

E:\Downloads\Requerimiento8.png

Figura 14 – 2 Requerimiento 8

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

E:\Downloads\Requerimiento9.png

Figura 15– 2 Requerimiento 9

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

E:\Downloads\Requerimiento10.png

Figura 16– 2 Requerimiento 10

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Determinación de los diagramas de actividades*

Tabla 15 – 2 Requerimiento 1

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Administración de usuarios | Administrador |
| Almacena el registro en la Base de Datos | Sistema |
| Confirma Registro | Administrador |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 16 – 2 Requerimiento 2

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Administración de clientes | Administrador , ejecutivo |
| Almacena el registro en la Base de Datos | Sistema |
| Confirma Registro | Administrador , ejecutivo |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 17 – 2 Requerimiento 3

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Administración de zonas | Administrador, ejecutivo |
| Almacena el registro en la Base de Datos | Sistema |
| Confirma Registro | Administrador, ejecutivo |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 18 – 2 Requerimiento 4

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Administración de tarifas | Administrador, ejecutivo |
| Almacena el registro en la Base de Datos | Sistema |
| Confirma Registro | Administrador, ejecutivo |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 19 – 2 Requerimiento 5

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Administración de medidores | Administrador, ejecutivo |
| Almacena el registro en la Base de Datos | Sistema |
| Confirma Registro | Administrador, ejecutivo |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 20 – 2 Requerimiento 6

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Cobro y emisión de facturas | Cajera (o) |
| Almacena el registro en la Base de Datos | Sistema |
| Confirma Registro | Cajera (o) |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 21 – 2 Requerimiento 7

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Registro lectura actual | Operador |
| Almacena el registro en la Base de Datos | Sistema |
| Confirma Registro | Operador |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 22 – 2 Requerimiento 8

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Consulta del historial de pagos | Cliente |
| Realiza consulta | Sistema |
| Visualiza lista de adeudos | Cliente |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 23 – 2 Requerimiento 9

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Generación de reportes | Administrador, operador |
| Solicitud de emisión de información | Sistema |
| Visualización del reporte | Administrador, operador |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 24 – 2 Requerimiento 10

|  |  |
| --- | --- |
| **ACTIVIDADES** | **AUTOR** |
| Asignación de Roles | Administrador |
| Guardado de los registros | Sistema |
| Mensaje de confirmación | Administrador |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Diagrama Gantt*

**Las figuras 17 – 2 y 18 - 2** muestran cómo se desarrollara el proyecto de manera cronológica en cada una de sus etapas según la metodología MSF.

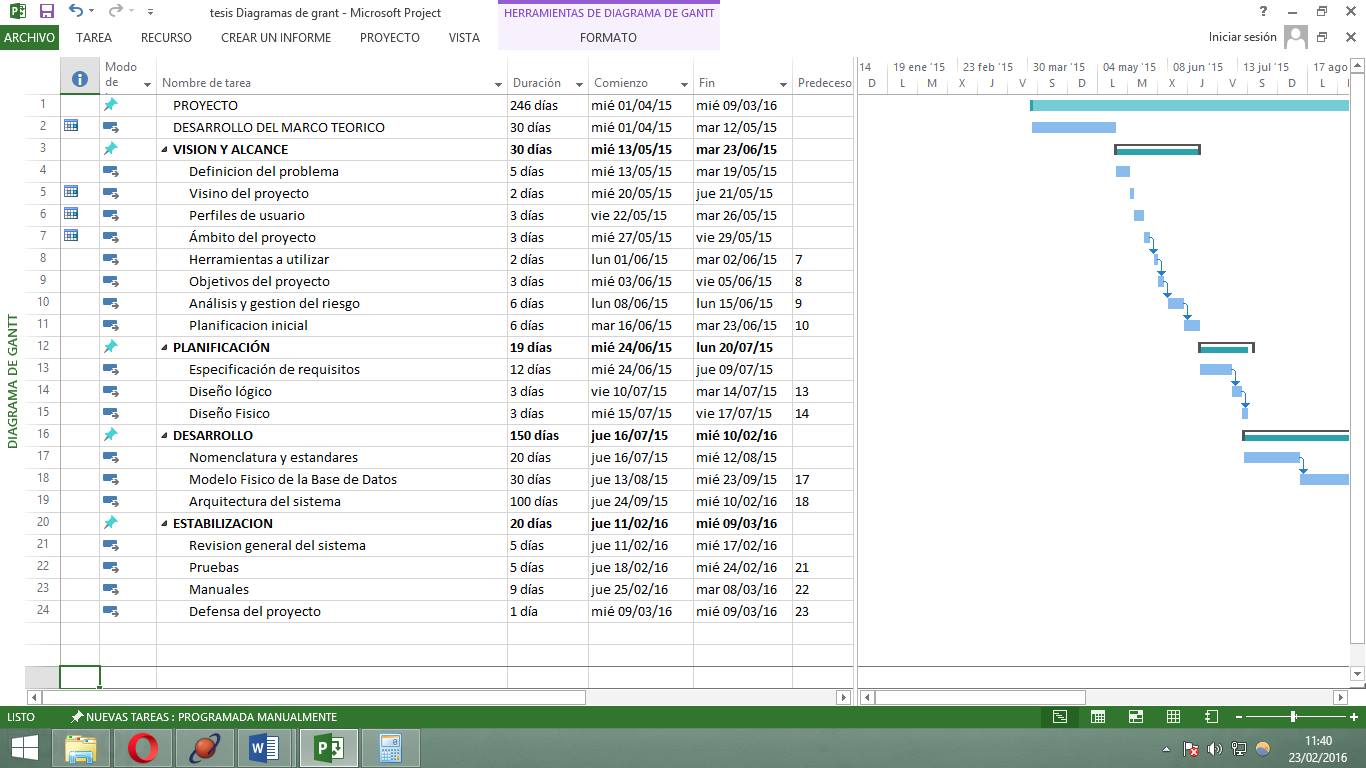


Figura 17 – 2 Cronograma de Actividades, diagrama de GRANT

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

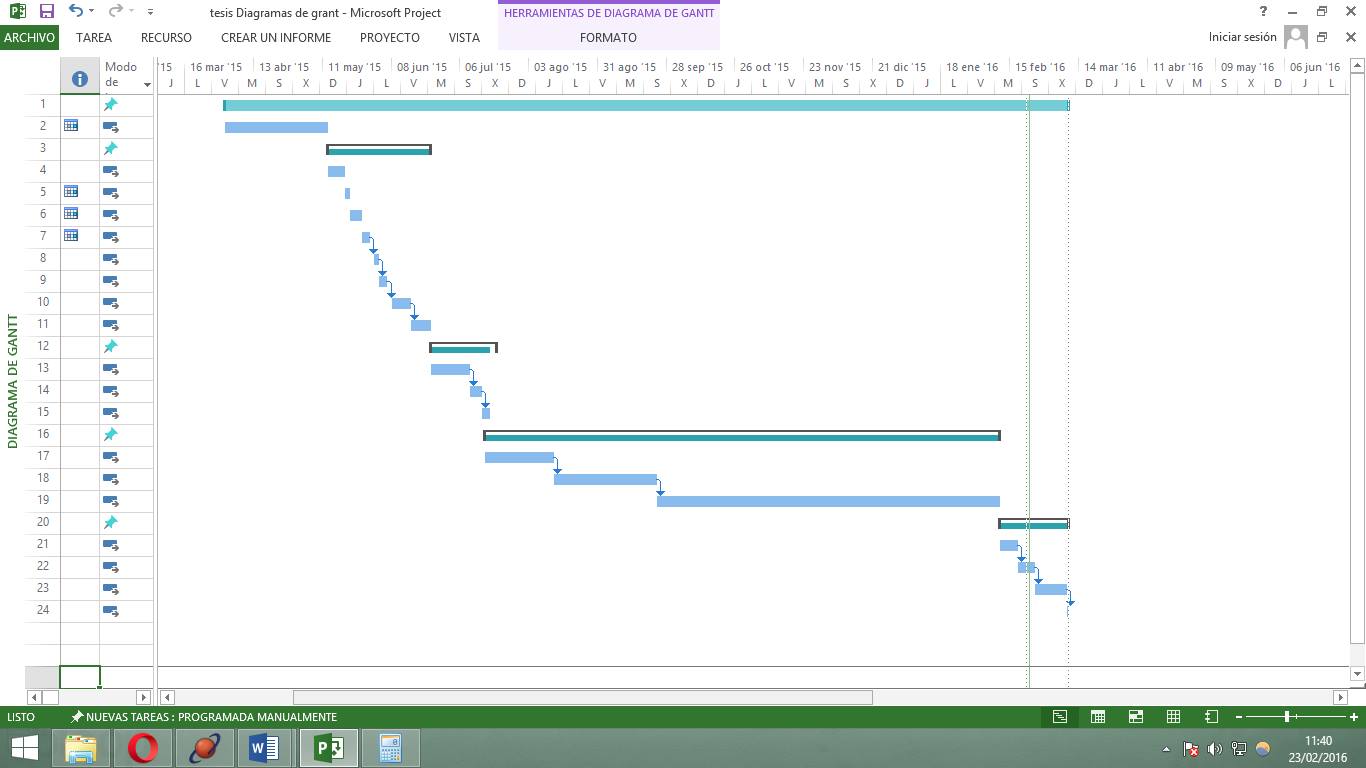


Figura18 – 2 Diagrama de GRANT de seguimiento

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

### ***Estudio de Factibilidad***

#### **Factibilidad Técnica**

Software mínimo requerido

Montar un servidor en Linux, para esto se usara Ubuntu Server

En el cual se instalara:

* + Glassfish
  + PostgrSQL

Hardware Requerido

1 PC servidor

Procesador: Intel Centrino

Memoria RAM: 2 GB

Disco Duro 500 GB

Herramientas utilizadas

Netbeans

PostgreSQL

Office 2013

iReport

#### **Factibilidad Operativa**

1. Para que el personal o los usuario que harán uso dela aplicación no sientan confusión al manejarlo se crearan interfaces amigables e intuitivas las cuales ayudaran en gran medida a que estos se acoplen de mejor manera al uso de la aplicación.
2. Para la administración de la aplicación se recomienda que los usuarios tengan conocimientos básicos de computación y en lo posible de programación.

#### **Factibilidad legal.**

Para la realización de la aplicación se hará uso de la ordenanza que reglamenta la prestación de los servicios de agua potable y de alcantarillado del Cantón Logroño, este documento nos ha sido facilitado por el departamento de catastros del Municipio de Logroño.

#### **Factibilidad económica**

Tabla 25 – 2 Factibilidad económica

|  |  |  |  |
| --- | --- | --- | --- |
| **HARDWARE** | | | |
| **CANTIDAD** | **DESCRIPCIÓN** | **PRECIO U.** | **PRECIO TOTAL** |
| 2 | PC portátiles | 600,00 | 1200,00 |
| 1 | Impresora | 200,00 | 200,00 |
| **RECURSO HUMANO** | | | |
| **CANTIDAD** |  |  |  |
| 2 | Desarrolladores por 6 meses | 5000,00 | 10000,00 |
| **OTROS GASTOS** | | | |
| **CANTIDAD** | **DESCRIPCIÓN** | **PRECIO U.** | **PRECIO TOTAL** |
| 1 | Internet | 150,00 | 150,00 |
| 1 | Transporte | 300,00 | 300,00 |
| 1 | Teléfono | 50,00 | 50,00 |
| 1 | Insumos de oficina | 150,00 | 150,00 |
| **COSTO DEL PROYECTO** | | | **12050,00** |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

## **Planificación**

### ***Definición y descripción de requerimientos***

*Objetivo del SRS*

La especificación de requerimientos de software (SRS) tiene como objetivo primordial brindar una referencia sobre las funciones que la Aplicación va a desempeñar, teniendo en cuenta las necesidades, objetivos de las funciones y procesos de los usuarios y requerimientos de los mismos.

*Ámbito*

El Municipio del Cantón Logroño en vista de contribuir con la sociedad de una mejor manera y la forma de prestar sus servicios emprende el proyecto de un sistema de cobro del Servicio del Agua Potable para agilizar el proceso de cobranza del dicho servicio a los moradores de su localidad.

*Sistema de Facturación y Control del Agua Potable “SIFACOAC”*

**SIFACOAC** tiene la función principal de crear una cuenta al a un medidor en determinada dirección y asignarla al cliente que solicite la instalación del mismo y realizar la facturación mensual de cada cuenta de medidor.

*SRS Visión General*

El SRS es un documento estrictamente detallado que nos permite especificar todos los requerimientos de la Aplicación, las funcionalidades que debe cumplir el producto final, describir la arquitectura, organización y flujo de datos. Luego de la recopilación de toda la información necesaria se emite este documento para su análisis y aprobación por el usuario.

*SRS Descripción General*

Para agilizar los procesos que se realizan dentro del Cobro del Servicio de Agua Potable se desarrolla SIFACOAC (Sistema de Facturación y Control del Agua Potable) para solucionar de forma eficaz la ejecución de dichos procesos en un ambiente de interfaz amigable para los usuarios.

*Razones que motiva la realización del Proyecto*

* Mejorar eficientemente los procesos que intervienen el cobro del Servicio del Agua Potable.
* La satisfacción de los usuarios en el ahorro de tiempo al ejecutar las actividades.
* Satisfacción en los cliente ya que pueden acceder a información de sus pagos.
* Obtención de reportes que ayuden en la toma de decisiones a las autoridades y personal de la institución.

*Perspectiva del Producto*

La aplicación puede ser utilizada por muchos Municipios de la Provincia de Morona Santiago, ya que la aplicación cubrirá todos los objetivos para maximizar la efectividad de los procesos, datos, requerimientos y también brindar todas las facilidades a los usuarios en la ejecución de dichas actividades.

* Agilizar los procesos de cada actividad que interviene en el cobro del servicio de agua potable y que sirva de apoyo en la toma de decisiones cuando requiera la ocasión.
* El sistema podrá ser manejado desde un entorno web y dispositivos móviles.

*Funciones del Producto*

Las funciones principales con las que cuenta la aplicación son:

* Realizar el cobro del servicio de agua potable.
* Emisión de la factura al realizar el cobro.
* Gestión de información de usuarios, clientes, medidores etc.
* Visualización e impresión de reportes para la entrega y verificación de resultados en el cobro del agua potable.

*Características del Usuario*

Las características de los usuarios de la aplicación SIFACOAG se muestran en la siguiente jerarquía.

* Administrador
* Ejecutivos
* Cajeros
* Operadores
* Clientes

*Limitaciones Generales*

* El sistema no acepta el ingreso documentos, imágenes o cualquier otro tipo de archivos.
* El sistema no registra ni emite ningún tipo de retenciones.

*Supuestos y Dependencias*

Algunos factores que podrían afectar en los requerimientos de la aplicación son:

* Que se cambien o agreguen funcionalidades a las ya determinadas, cuando el proyecto esté en desarrollo.
* La información ingresada por parte de los encargados sea incorrecta.

#### Interfaces externas

*Interfaz de usuarios*

Todos los usuarios (Administrador, Ejecutivos, Cajeros y Operadores) podrán tener acceso a las diferentes interfaces de la aplicación según los permisos que le otorgue su perfil de usuario, estas páginas se encuentran implementadas en interfaces gráficas, que podrán observarse mediante un navegador web.

Se hará uso de mensajes de advertencias y de confirmación para realizar el proceso de verificación antes de almacenar la información a la base de datos.

**Interfaz de hardware**

A continuación se muestra en las tablas los requerimientos mínimos de hardware para el servidor y los usuarios respectivamente.

Para el correcto funcionamiento de la aplicación se podría hacer uso de un servidor con las características presentadas en la **tabla 14 - 2:**

Tabla 26 – 2 Características del servidor

|  |  |
| --- | --- |
| **CARACTERÍSTICAS** | **REQUERIMIENTO MÍNIMO** |
| Procesador | Intel Xeon 2.0 Ghz |
| Memoria RAM | 16 Gb |
| Disco duro | 2 Tb |
| Interfaz de Red | 100 Mbps |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Para que el usuario pueda acceder y hacer uso del sistema necesita un equipo con las características presentadas en la **tabla 15 - 2:**

Tabla 27 – 2 Equipo cliente

|  |  |
| --- | --- |
| **CARACTERÍSTICAS** | **REQUERIMIENTO MÍNIMO** |
| Procesador | Celeron 2.6 GHz |
| Memoria RAM | 2 Gb |
| Disco duro | 500 Gb |
| Interfaz de Red | 100 Mbps |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Impresora*

Para la presentación e impresión de reportes será necesario una impresora para que el usuario pueda presentar dicha información impresa donde corresponda.

#### Interfaces de software

Para el correcto funcionamiento de **SIFACOG** es necesario disponer de los requerimientos en software presentados en la **tabla 16 - 2:**

Tabla 28 – 2 Software requerido

|  |  |
| --- | --- |
| **TIPO** | **PLATAFORMA SOFTWARE** |
| Sistema operativo | Ubuntu Server |
| DBMS | PostgreSQL |
| Plataforma de Programación | Java NetBeans |
| Plataforma de Aplicaciones | Mozilla Firefox, Google Chrome, Internet Explorer, Opera. |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

En el caso de los usuarios y clientes.

*Navegador*

Para que los usuarios puedan hacer uso de la aplicación solo es necesario que cuenten con un navegador web (Mozilla Firefox, Google Chrome, Internet Explorer) ya que el sistema esta implementado en un Servidor Web.

#### Interfaz de comunicación

* La plataforma donde se pondrá en funcionamiento la Aplicación Web es mediante un servidor Web (Ubuntu Server) el cual permite desplegar las páginas dinámicas de la aplicación.
* La información estará alojada en el servidor en el mismo servidor por lo que no hay problema ya que el Municipio cuneta con equipos informáticos para la implementación de dicho servidor.

#### Requerimientos funcionales de SIFACOAG.

A continuación se detalla cada uno de los requerimientos del sistema SIFACOAG para el cual se definirá los datos de entradas, procesos y salidas.

1. *Administración de usuarios:*SIFACOAG debe permitir el registro y edición de usuarios (Administrador, Ejecutivo, Cajeros, Operadores) y otorgarle los permisos correspondientes a cada perfil de usuario es decir para que los usuarios puedan acceder al sistema tienen que ser registrados previamente en la base de datos y de acuerdo a su perfil se le mostraran diferentes opciones que pueda realizar en el menú de la aplicación.
2. *Administración de clientes:* La aplicación debe permitir el registro y edición de los clientes que realizaran los pagos de las cuentas asignadas por medidor a su nombre.
3. *Gestión de zonas:* La aplicación debe permitir crear y editar información acerca de las zonas en las que se maneja la cobranza del servicio de agua potable en el cantón Logroño, estas zonas serán asignadas a los usuarios operadores que tiene a cargo una zona respectiva.
4. *Gestión de tarifas:* SIFACOAG permitirá a asignación del precio a cobrar a la cuenta del medidor, el precio varía de acuerdo al tipo de tarifa que se haya asignado a dicha cuenta siendo esta residencial, comercial o industrial.
5. *Gestión de medidores:* La aplicación debe permitir crear y actualizar la información de cuentas de medidores según sea solicitada por el cliente.
6. *Cobro y emisión de facturas:* Sabiendo que el cobro del servicio del agua potable se lo realiza mensualmente la aplicación actualizara automáticamente la deuda a cobrar según se haya registrado el consumo actual del cliente y a su vez registrara y emitirá el cobro/pago de esa factura por parte del cliente.
7. *Registro de lectura actual del medidor:* La aplicación debe permitir el registro de consumo actual del medidor, este proceso se lo llevara a cabo cada mes y por parte del usuario operador.
8. *Consultas del historial de pagos del cliente:* SIFACOAG debe permitir las consultas por parte del cliente, estas consultas reflejaran un historial de los pagos realizados por dicho cliente de los medidores a su cargo.
9. *Generar de reportes:* La aplicación debe permitir a los responsables de cada departamento generar reportes de acuerdo a sus necesidades requeridas que indique las actividades realizadas dentro y fuera de la institución.
10. *Asignar Permisos al Tipo de Usuarios:* SIFACOAC permitirá crear diferentes tipos de usuario a los cuales se les asignara distintos permisos que puedan realizar dentro de la aplicación que se verán reflejadas en las opciones en el menú acordes con su perfil de usuario asignado cada usuario.

*Requerimiento 1: Administración de usuarios.*

*Descripción:* SIFACOAG debe permitir el registro y edición de usuarios (Administrador, Ejecutivo, Cajeros, Operadores) y otorgarle los permisos correspondientes a cada perfil de usuario es decir para que los usuarios puedan acceder al sistema tienen que ser registrados previamente en la base de datos y de acuerdo a su perfil se le mostraran diferentes opciones que pueda realizar en el menú de la aplicación.

*Entradas*:

* Número de cédula.
* Nombres.
* Apellidos.
* Nick.
* Contraseña.
* Dirección.
* Teléfono.
* Correo.
* Tipo de usuario.

*Procesos*:

* El administrador ingresa los datos requeridos para crear la cuenta de usuarios.
* El administrador asigna el rol que desempeñara cada usuario registrado.
* El administrador registra los datos en el sistema.

*Salidas:*

* Mensaje de error o de registro exitoso.

*Requerimiento 2:* *Administración de clientes.*

*Descripción:* La aplicación debe permitir el registro y edición de los clientes que realizaran los pagos de las cuentas asignadas por medidor a su nombre.

*Entradas*:

* Número de cédula.
* Nombres.
* Apellidos.
* Teléfono.
* Fecha de nacimiento.
* Dirección.

Procesos:

* El administrador ingresa los datos requeridos para crear la cuenta de clientes.
* El administrador registra los datos en el sistema.

*Salidas:*

* Mensaje de error o de registro exitoso.

*Requerimiento 3: Gestión de zonas.*

*Descripción:* La aplicación debe permitir crear y editar información acerca de las zonas en las que se maneja la cobranza del servicio de agua potable en el cantón Logroño, estas zonas serán asignadas a los usuarios operadores que tiene a cargo una zona respectiva.

*Entradas*:

* Descripción.
* Numero de zona.
* Rango que comprende la zona.
* Región.
* Operador.

*Procesos*:

* El usuario responsable ingresa los datos para crear y actualizar datos de una zona y los registra en el sistema.
* La aplicación verifica los datos y los almacena.

*Salidas:*

* Mensaje de error o de registro exitoso.

*Requerimiento 4:* *Gestión de tarifas.*

*Descripción:* SIFACOAG permitirá a asignación del precio a cobrar a la cuenta del medidor, el precio varía de acuerdo al tipo de tarifa que se haya asignado a dicha cuenta siendo esta residencial, comercial o industrial.

*Entradas:*

* Rango.
* Volumen.
* Excedente.
* Tipo de tarifa.

*Procesos*:

* El usuario responsable ingresa los datos.
* El usuario registra los datos en la base de datos.

*Salidas:*

* Mensaje de error o de registro exitoso.

*Requerimiento 5: Gestión de Medidores.*

*Descripción: L*a aplicación debe permitir crear y actualizar la información de cuentas de medidores según sea solicitada por el cliente.

*Entradas*:

* Numero de medidor.
* Marca.
* Estado.
* Zona.
* Dirección: Calle 1, Calle 2, Calle 3.
* Putos GPS X, Y.
* Cliente.
* Tarifa.

*Procesos:*

* El usuario responsable ingresa los datos para crear o actualizar una cuenta para el medidor de acuerdo a la solicitud del cliente.
* El sistema verifica los datos y los almacena.

*Salidas:*

Mensaje de error o de registro exitoso.

*Requerimiento 6: Cobro y emisión de facturas.*

*Descripción:* Sabiendo que el cobro del servicio del agua potable se lo realiza mensualmente la aplicación actualizara automáticamente la deuda a cobrar según se haya registrado el consumo actual del cliente y a su vez registrara y emitirá el cobro/pago de esa factura por parte del cliente.

*Entradas:*

* Fecha.
* Subtotal.
* IVA.
* Descuento.
* Total.
* Cuenta medidor.

*Procesos:*

* El sistema muestra el saldo, deuda o estado a cancelar de los medidores a cargo del cliente.
* El usuario responsable realiza el cobro y registro de la factura.
* El sistema registra y emite la factura al cliente.

*Salidas:*

* Mensaje de error o de registro exitoso y emisión de factura.

*Requerimiento 7: Registro de lectura actual del medidor.*

*Descripción:* La aplicación debe permitir el registro de consumo actual del medidor, este proceso se lo llevara a cabo cada mes y por parte del usuario operador.

*Entradas:*

* Lectura actual.

*Procesos:*

* El usuario operador realiza la consulta de los medidores a su cargo.
* Verifica los datos que muestra la aplicación.
* Ingresa los datos actuales del consumo de agua potable.
* Registra los datos en la aplicación.

*Salidas:*

* Mensaje de error o de registro exitoso.

*Requerimiento 8:* *Consultas del historial de pagos del cliente.*

*Descripción:* SIFACOAG debe permitir las consultas por parte del cliente, estas consultas reflejaran un historial de los pagos realizados por dicho cliente de los medidores a su cargo.

*Entradas*:

* Cuenta de medidor/ número de cedula.

*Procesos*:

* El cliente ingresa el número de cuenta del medidor o su número de cédula para la consulta respectiva.
* El sistema verifica los datos.

*Salidas*:

* Mensaje de error o muestra los resultados de la consulta.

*Requerimiento 9: Generar reportes.*

*Descripción:* La aplicación debe permitir a los responsables de cada departamento generar reportes de acuerdo a sus necesidades requeridas que indique las actividades realizadas dentro y fuera de la institución.

*Entradas:*

* Solicitud de vista de reportes.

*Procesos:*

* El usuario solicita la generación de un reporte.
* El usuario elige el tipo de reporte a visualizar.
* El sistema despliega el reporte solicitado.

*Salidas:*

* Despliegue del reporte solicitado.

*Requerimiento 10: Asignar Permisos al Tipo de Usuarios.*

*Descripción:* SIFACOAC permitirá crear diferentes tipos de usuario a los cuales se les asignara distintos permisos que puedan realizar dentro de la aplicación que se verán reflejadas en las opciones en el menú acordes con su perfil de usuario asignado cada usuario.

*Entradas:*

* Roles.

*Proceso*:

* El súper usuario crea el tipo de usuario.
* Solicita asignación de roles.
* Asigna roles.
* Registra asignación.

*Salidas*:

* Mensaje de confirmación.

#### Requerimientos no funcionales.

La siguiente lista detalla los requerimientos no funcionales que son de vital importancia para que la aplicación funcione de la mejor manera.

1. Seguridad.
2. Matenibilidad.
3. Fiabilidad.
4. Disponibilidad.
5. Escalabilidad.
6. Flexibilidad.
7. Facilidad de uso.

#### Actores.

Estos son los actores que intervienen en la funcionalidad de la aplicación.

* Administrador.
* Ejecutivo.
* Cajeros.
* Operadores.
* Clientes.

#### Casos de uso.

A continuación se muestra el caso de uso del requerimiento # 1:

*Administración de usuarios.*

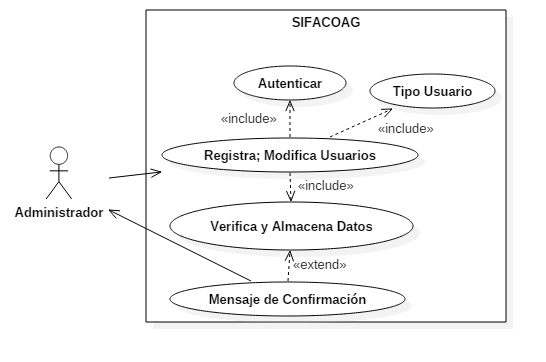


Figura 19 – 2 Caso de uso 1

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 29 – 2 Caso de uso. Requerimiento # 1:

|  |  |
| --- | --- |
| **Numero:** Requerimiento 1 | **Usuarios:** Administrador |
| **Nombre Historia:** Administración de usuarios | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:** SIFACOAG debe permitir el registro y edición de usuarios (Administrador, Ejecutivo, Cajeros, Operadores) y otorgarle los permisos correspondientes a cada perfil de usuario es decir para que los usuarios puedan acceder al sistema tienen que ser registrados previamente en la base de datos y de acuerdo a su perfil se le mostraran diferentes opciones que pueda realizar en el menú de la aplicación. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El administrador ingresa a la aplicación mediante su usuario y contraseña.  2. El administrador solicita la operación de registro o edición de usuarios.  4. El administrador registra o modifica la información de los usuarios. | 3. Despliega la interfaz gráfica adecuada para la operación solicitada por el usuario.  5. Valida los datos de entrada y ejecuta la acción solicitada. |
| **Observaciones:**  La información que la aplicación solicita para el registro/modificación debe ser ingresada en su totalidad y correctamente caso contrario el sistema no almacena los datos y regresa al usuario al paso 4. | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

A continuación se muestra el caso de uso del requerimiento # 2:

*Administración de clientes.*

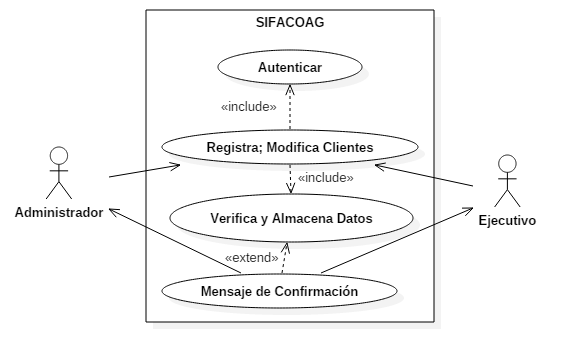


Figura 20 – 2 Caso de uso 2

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 30 – 2 Caso de uso. Requerimiento # 2:

|  |  |
| --- | --- |
| **Numero:** Requerimiento 2 | **Usuarios:** Administrador, Ejecutivo |
| **Nombre Historia:** Administración de clientes | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:** La aplicación debe permitir el registro y edición de los clientes que realizaran los pagos de las cuentas asignadas por medidor a su nombre. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El usuario (Administrador, Ejecutivo) ingresa a la aplicación mediante su usuario y contraseña.  2. El administrador o ejecutivo solicita la operación de registro o edición de clientes.  4. El administrador o ejecutivo registra o modifica la información de los clientes. | 3. Despliega la interfaz gráfica adecuada para la operación solicitada por el usuario.  5. Valida los datos de entrada y ejecuta la acción solicitada. |
| **Observaciones:**  La información que la aplicación solicita para el registro/modificación debe ser ingresada en su totalidad y correctamente caso contrario el sistema no almacena los datos y regresa al usuario al paso 4. | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

A continuación se muestra el caso de uso del requerimiento # 3:

*Gestión de zonas.*

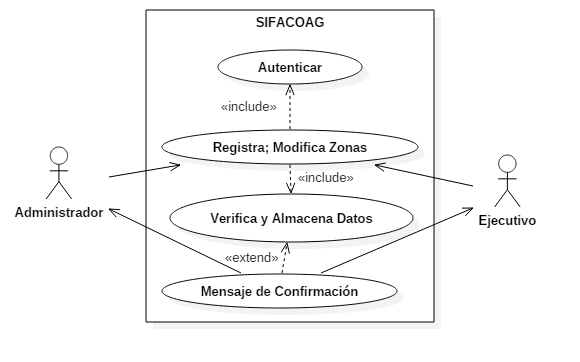
**

Figura 21 – 2 Caso de uso 3

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 31 - 2 Caso de uso. Requerimiento # 3:

|  |  |
| --- | --- |
| **Numero:** Requerimiento 3 | **Usuarios:** Administrador, Ejecutivo |
| **Nombre Historia:** Gestión de zonas | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:** La aplicación debe permitir crear y editar información acerca de las zonas en las que se maneja la cobranza del servicio de agua potable en el cantón Logroño, estas zonas serán asignadas a los usuarios operadores que tiene a cargo una zona respectiva. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El usuario (Administrador, Ejecutivo) ingresa a la aplicación mediante su usuario y contraseña.  2. El administrador o ejecutivo solicita la operación de registro o edición de zonas.  4. El administrador o ejecutivo registra o modifica la información de las zonas. | 3. Despliega la interfaz gráfica adecuada para la operación solicitada por el usuario.  5. Valida los datos de entrada y ejecuta la acción solicitada. |
| **Observaciones:**  La información que la aplicación solicita para el registro/modificación debe ser ingresada en su totalidad y correctamente caso contrario el sistema no almacena los datos y regresa al usuario al paso 4. | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

A continuación se muestra el caso de uso del requerimiento # 4:

*Gestión de tarifas.*

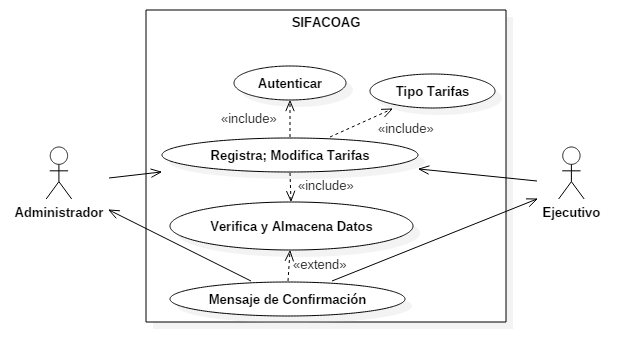
**

Figura 22 – 2 Caso de uso 4

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 32 – 2 Caso de uso. Requerimiento # 4:

|  |  |
| --- | --- |
| **Numero:** Requerimiento 4 | **Usuarios:** Administrador, Ejecutivo |
| **Nombre Historia:** Gestión de tarifas | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:** SIFACOAG permitirá la asignación del precio a cobrar a la cuenta del medidor, el precio varía de acuerdo al tipo de tarifa que se haya asignado a dicha cuenta siendo esta residencial, comercial o industrial. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El usuario (Administrador, Ejecutivo) ingresa a la aplicación mediante su usuario y contraseña.  2. El usuario solicita la operación de registro o edición de tarifas.  4. El usuario registra o modifica la información de las tarifas de cobro. | 3. Despliega la interfaz gráfica adecuada para la operación solicitada por el usuario.  5. Valida los datos de entrada y ejecuta la acción solicitada. |
| **Observaciones:**  La información que la aplicación solicita para el registro/modificación debe ser ingresada en su totalidad y correctamente caso contrario el sistema no almacena los datos y regresa al usuario al paso 4. | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

A continuación se muestra el caso de uso del requerimiento # 5:

*Gestión de medidores.*

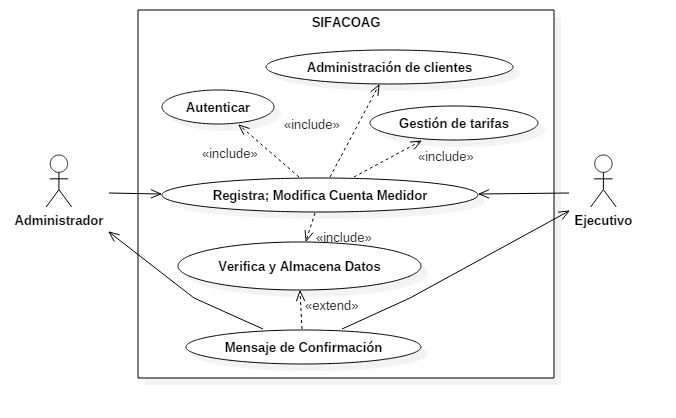
**

Figura 23 – 2 Caso de uso 5

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 33 – 2 Caso de uso. Requerimiento # 5:

|  |  |
| --- | --- |
| **Numero:** Requerimiento 5 | **Usuarios:** Administrador, Ejecutivo |
| **Nombre Historia:** Gestión de medidores | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:** La aplicación debe permitir crear y actualizar la información de cuentas de medidores según sea solicitada por el cliente. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El usuario (Administrador, Ejecutivo) ingresa a la aplicación mediante su usuario y contraseña.  2. El usuario solicita la operación de registro o edición de cuentas de medidores.  4. El usuario registra o modifica la información de las cuentas del medidor. | 3. Despliega la interfaz gráfica adecuada para la operación solicitada por el usuario.  5. Valida los datos de entrada y ejecuta la acción solicitada. |
| **Observaciones:**  La información que la aplicación solicita para el registro/modificación debe ser ingresada en su totalidad y correctamente caso contrario el sistema no almacena los datos y regresa al usuario al paso 4. | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

A continuación se muestra el caso de uso del requerimiento # 6:

*Cobro y emisión de facturas.*

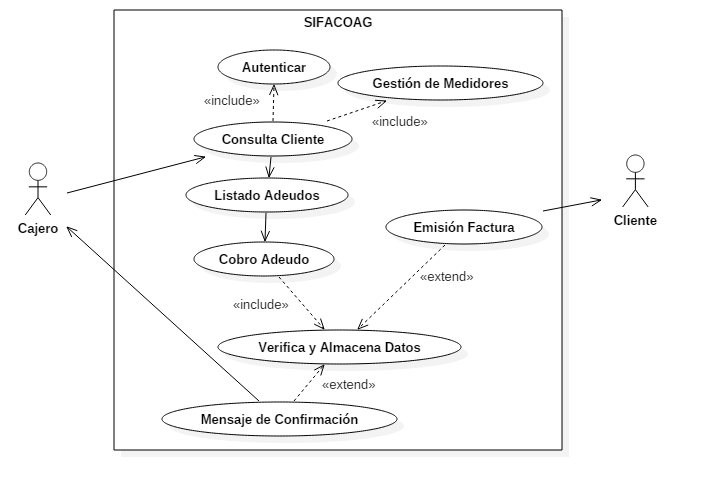
**

Figura 24 – 2 Caso de uso 6

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 34 – 2 Caso de uso. Requerimiento # 6:

|  |  |
| --- | --- |
| **Numero:** Requerimiento 6 | **Usuarios:** Cajero |
| **Nombre Historia:** Cobro y emisión de facturas | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:** Sabiendo que el cobro del servicio del agua potable se lo realiza mensualmente la aplicación actualizara automáticamente la deuda a cobrar según se haya registrado el consumo actual del cliente y a su vez registrara y emitirá el cobro/pago de esa factura por parte del cliente. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El cajero ingresa a la aplicación mediante su usuario y contraseña.  2. El cajero solicita la operación para el cobro del adeudo del medidor.  4. El cajero ingresa datos del cliente a buscar para el cobro.  7. Realiza el cobro al cliente y solicita registro de factura. | 3. Despliega la interfaz gráfica adecuada para la operación solicitada por el usuario.  5. Valida los datos.  6. Despliega un listado con adeudos por cobrar de dicho cliente.  8. Valida datos de entrada registra y actualiza datos de la cuenta.  9. Emite la factura. |
| **Observaciones:**  Si el cliente no presenta adeudos la aplicación no desplegara la lista de adeudos por cobrar para dicho cliente. | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

A continuación se muestra el caso de uso del requerimiento # 7:

*Registro de lectura actual del medidor.*

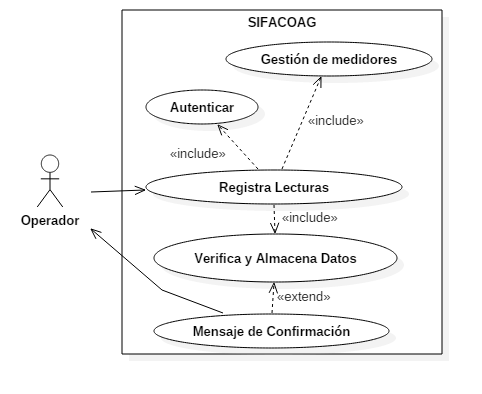
**

Figura 25 – 2 Caso de uso 7

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 35 – 2 Caso de uso. Requerimiento # 7:

|  |  |
| --- | --- |
| **Numero:** Requerimiento 7 | **Usuarios:** Operador |
| **Nombre Historia:** Gestión de medidores | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:** La aplicación debe permitir el registro de consumo actual del medidor, este proceso se lo llevara a cabo cada mes y por parte del usuario operador. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El operador ingresa a la aplicación mediante su usuario y contraseña.  2. El operador solicita datos del medidor que tiene a cargo.  4. El operador registra la información actual de lecturas de consumo de agua potable. | 3. Despliega la interfaz gráfica adecuada para la operación solicitada por el usuario.  5. Valida los datos de entrada y registra los datos ingresados. |
| **Observaciones:** | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

A continuación se muestra el caso de uso del requerimiento # 8:

*Consultas del historial de pagos del cliente.*

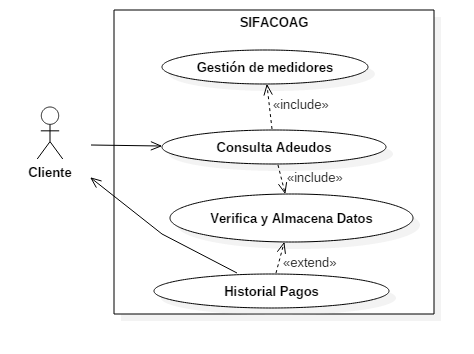
**

Figura 26 – 2 Caso de uso 8

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 36 – 2 Caso de uso. Requerimiento # 8:

|  |  |
| --- | --- |
| **Numero:** Requerimiento 8 | **Usuarios:** Cliente |
| **Nombre Historia:** Consultas del historial de pagos del cliente | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:** SIFACOAG debe permitir las consultas por parte del cliente, estas consultas reflejaran un historial de los pagos realizados por dicho cliente de los medidores a su cargo. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El cliente ingresa al módulo de consultas.  3. El cliente ingresa el número de cuenta con número de cédula. | 2. Muestra la interfaz gráfica adecuada para la consulta.  4. Valida los datos.  5. Despliega historial de pagos y adeudos. |
| **Observaciones:**  Si el cliente cuenta con varios medidores a su nombre es recomendable que realice la consulta por el número de cedula para que pueda observar todos su medidores a cargo. | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

A continuación se muestra el caso de uso del requerimiento # 9:

*Generar de reportes.*

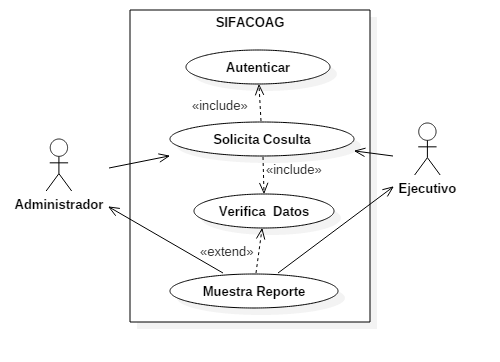
**

Figura 27 – 2 Caso de uso 9

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 37 – 2 Caso de uso. Requerimiento # 9:

|  |  |
| --- | --- |
| **Numero:** Requerimiento 9 | **Usuarios:** Administrador, Ejecutivo |
| **Nombre Historia:** Generar de reportes | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:** La aplicación debe permitir a los responsables de cada departamento generar reportes de acuerdo a sus necesidades requeridas que indique las actividades realizadas dentro y fuera de la institución. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El usuario (Administrador, Ejecutivo) ingresa a la aplicación mediante su usuario y contraseña.  2. El usuario solicita la información para el reporte. | 3. Despliega la interfaz gráfica adecuada para la operación solicitada por el usuario.  4. Valida los datos la solicitud y ejecuta la acción solicitada.  5. Muestra el reporte solicitado. |
| **Observaciones:**  El usuario puede realizar la consulta de reportes de información ya antes ingresada en la base de datos de la aplicación caso contrario no se mostrara algún resultado. | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

A continuación se muestra el caso de uso del requerimiento # 10:

*Asignar Permisos al Tipo de Usuarios.*

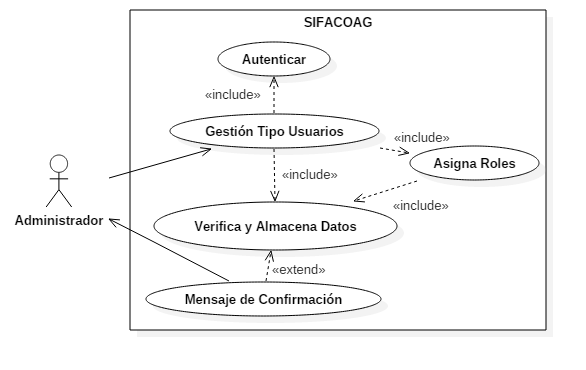
**

Figura 28 – 2 Caso de uso 10

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Tabla 38 – 2 Caso de uso requerimiento 10

|  |  |
| --- | --- |
| **Numero:** Requerimiento 10 | **Usuarios:** Administrador |
| **Nombre Historia:** Generar de reportes | |
| **Programador responsable:** Eduardo Guadalupe, Saúl Morocho | |
| **Descripción:**SIFACOAC permitirá crear diferentes tipos de usuario a los cuales se les asignara distintos permisos que puedan realizar dentro de la aplicación que se verán reflejadas en las opciones en el menú acordes con su perfil de usuario asignado cada usuario. | |
| **ACCIONES DE ACTORES** | **REPUESTA DEL SISTEMA** |
| 1. El usuario (Administrador) ingresa a la aplicación mediante su usuario y contraseña.  2. El usuario solicita el ingreso de tipo de usuarios.  5. Ingresa los datos solicitados. | 3. Despliega la interfaz gráfica adecuada para la operación solicitada por el usuario.  4. Valida los datos y solicita datos a ingresar.  6. Valida y almacena los datos. |
| **Observaciones:**  El usuario puede la asignación de los roles a cada tipo de usuario una vez ya ingresados los tipos de usuarios anteriormente. | |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### Arquitectura apropiada para la solución.

*Arquitectura interna de SIFACOAG*

La figura muestra la arquitectura de la aplicación SIFACOAG

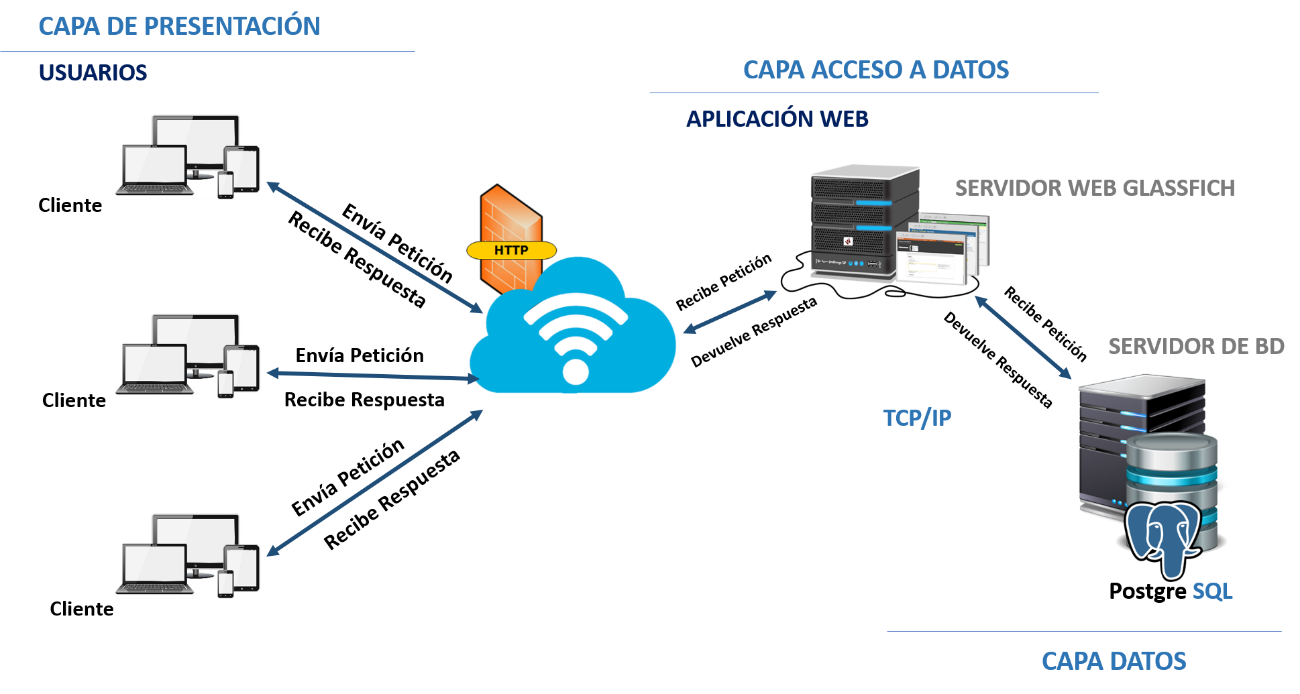


Figura 29- 2 Arquitectura del sistema

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

### ***Diseño lógico***

#### Diagramas de secuencia

**En la figura 30 – 2** del diagrama de secuencia se muestra cómo funciona el requerimiento # 1: *Administración de usuarios*.

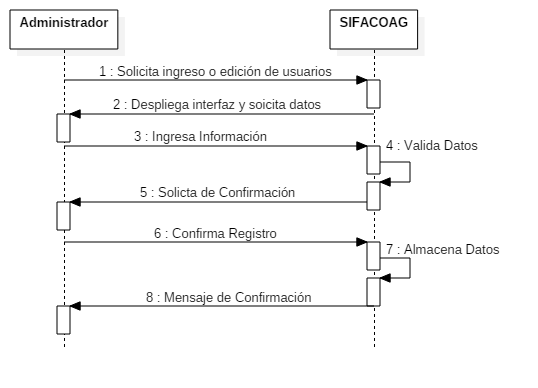


Figura 30 – 2 Requerimiento 1

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**En la figura 31 - 2** del diagrama de secuencia se muestra cómo funciona el requerimiento # 2:

*Administración de clientes.*

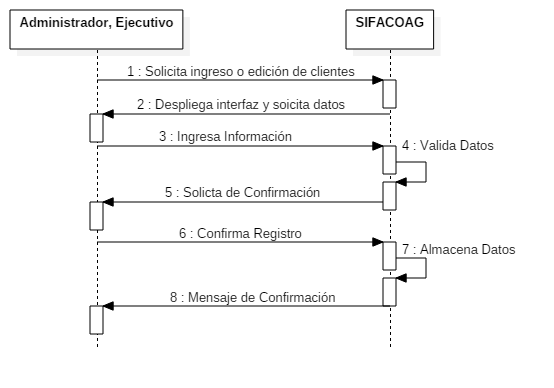
**

Figura 31 – 2 Requerimiento 2

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**En la figura 32 – 2** del diagrama de secuencia se muestra cómo funciona el requerimiento # 3:

*Gestión de zonas.*

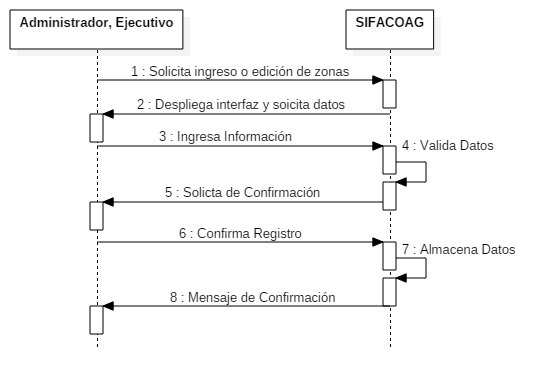


Figura 32 – 2 Requerimiento 3

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**En la figura 33 – 2** del diagrama de secuencia se muestra cómo funciona el requerimiento # 4:

*Gestión de tarifas.*

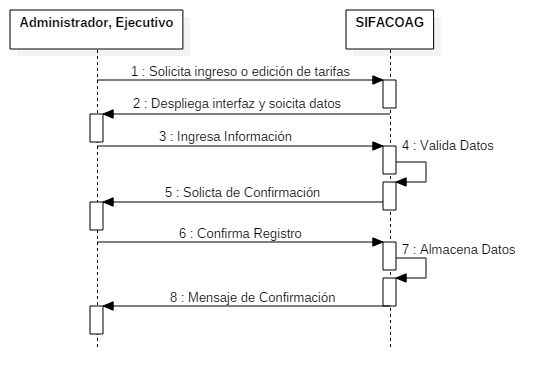
**

Figura 33 – 2 Requerimiento 4

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**En la figura 34 – 2** del diagrama de secuencia se muestra cómo funciona el requerimiento # 5:

*Gestión de medidores.*

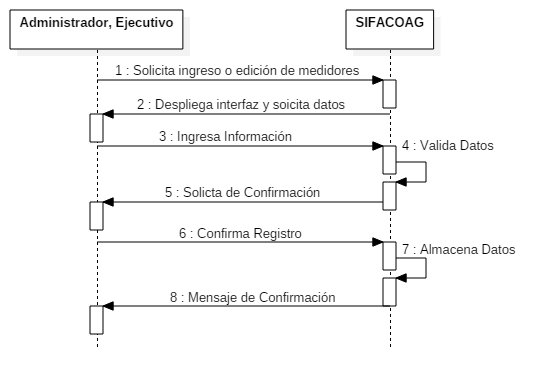
**

Figura 34 – 2 Requerimiento 5

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**En la figura 35 – 2** del diagrama de secuencia se muestra cómo funciona el requerimiento # 6:

*Cobro y emisión de facturas.*

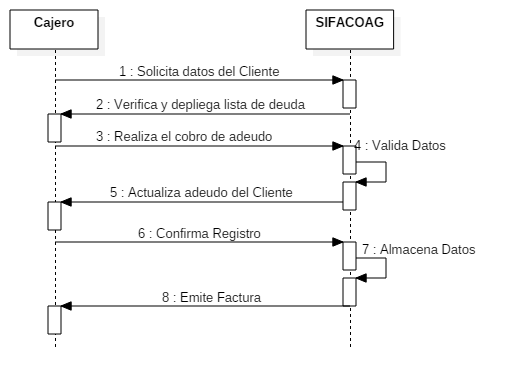


Figura 35 – 2 Requerimiento 6

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**En la figura 36 – 2** del diagrama de secuencia se muestra cómo funciona el requerimiento # 7:

*Registro de lectura actual del medidor.*

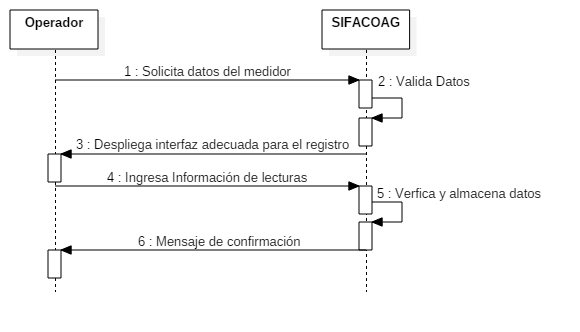
**

Figura 36 – 2 Requerimiento 7

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**En la figura 37 – 2** del diagrama de secuencia se muestra cómo funciona el requerimiento # 8:

*Consultas del historial de pagos del cliente.*

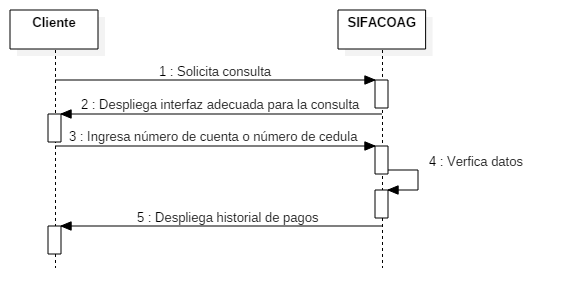


Figura 37 – 2 Requerimiento 8

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

**En la figura 38 – 2** del diagrama de secuencia se muestra cómo funciona el requerimiento # 9:

*Generar de reportes.*

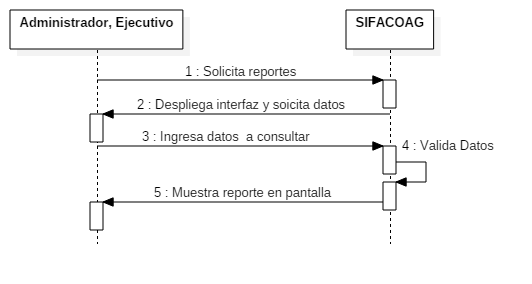


Figura 38 – 2 Requerimiento 9

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

En la figura 39 – 2 del diagrama de secuencia se muestra cómo funciona el requerimiento # 10:

*Asignar Permisos al Tipo de Usuarios.*

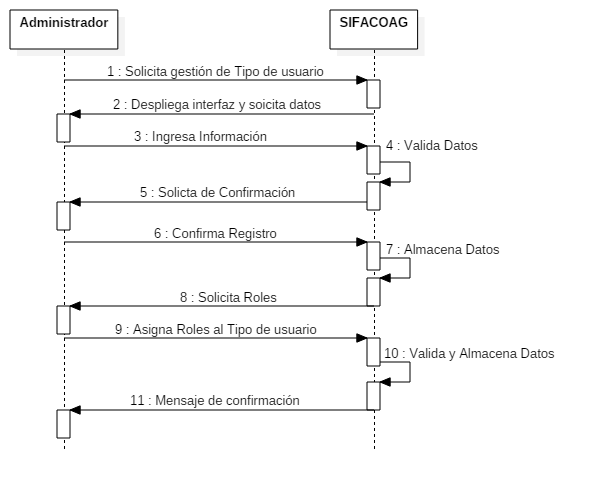


Figura 39 – 2 Requerimiento 10

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### Diagrama de clases

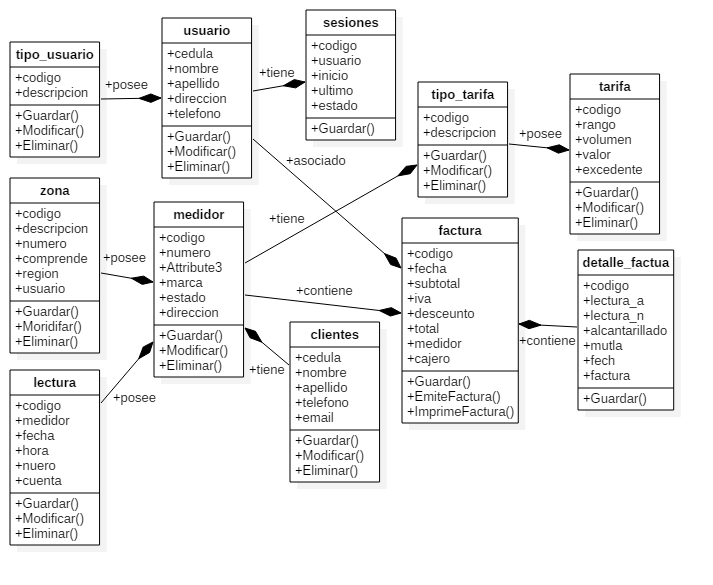
**

Figura 40 – 2 Diagrama de clases

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### Diseño de interfaces de usuario

A continuación se muestra la forma de interactuar el usuario con el sistema mediante interfaces que se presenta en el navegador y a través de internet.

Para el proceso de registro de información el usuario podrá observar masajes por los cuales tendrá en cuenta las acciones realizadas en el sistema antes y después de almacenar la información.

A continuación se muestra un prototipo de las interfaces gráficas de las cuales hará uso el usuario dentro de la aplicación.

*Página Principal del* ***SIFACOAG***



Figura 41 – 2 Página principal

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Registro de clientes*

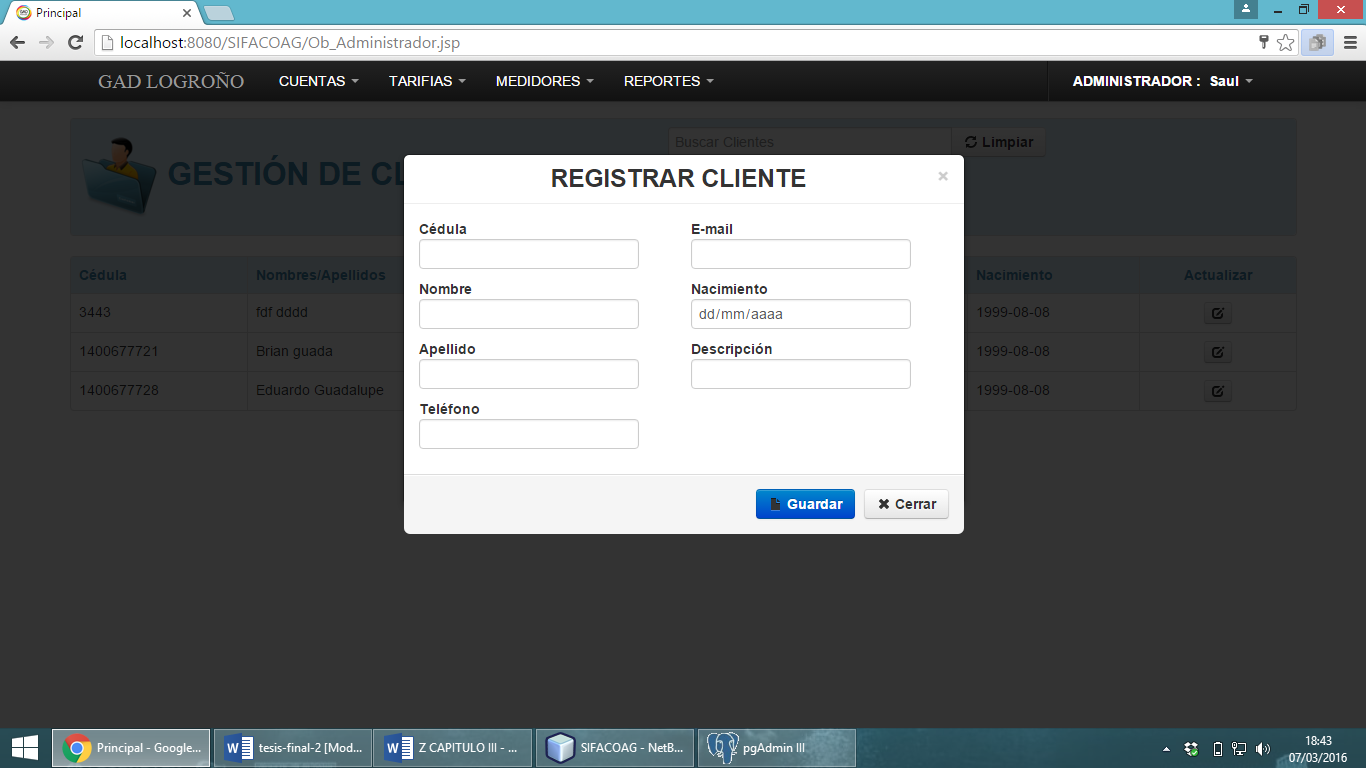


Figura 42 – 2 Registro de clientes

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Registro Tipo Usuarios*

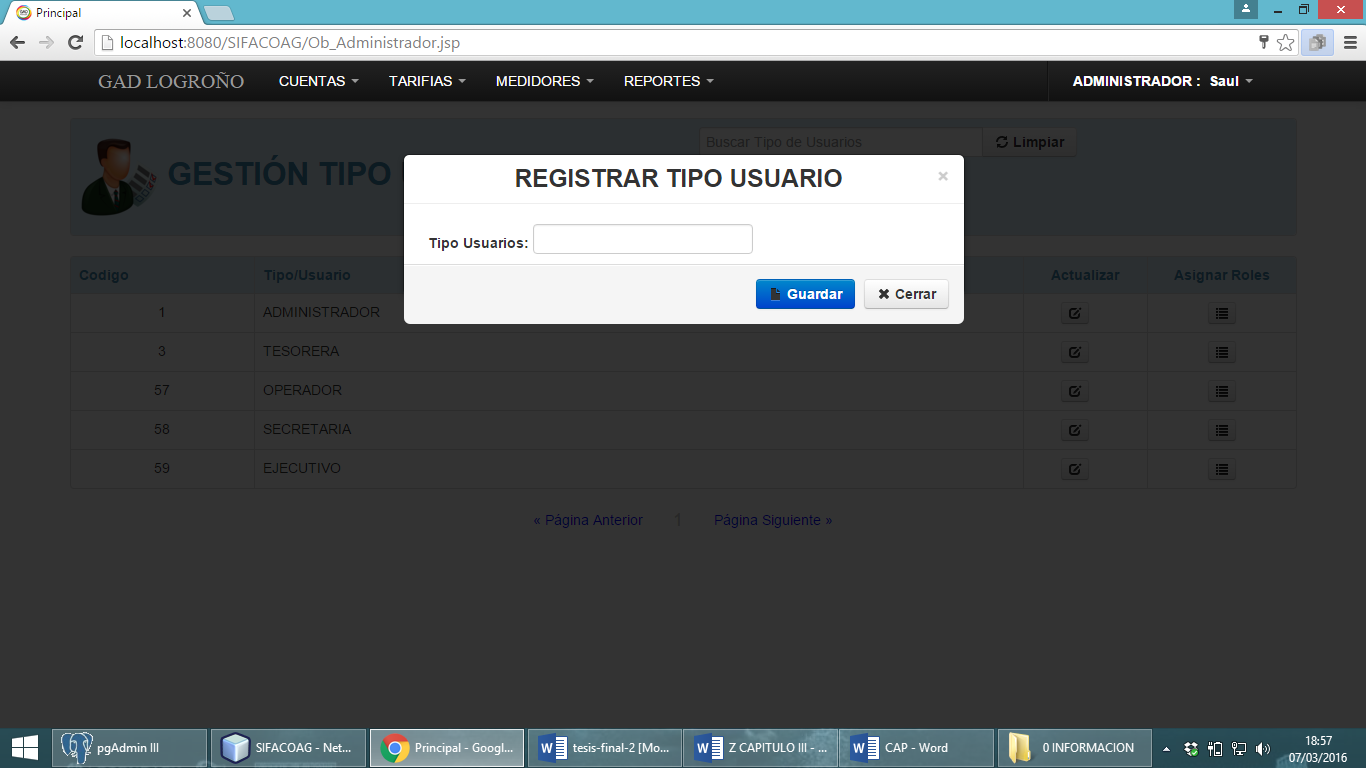


Figura 43 – 2 Registro tipo usuarios

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Registro de Usuarios*

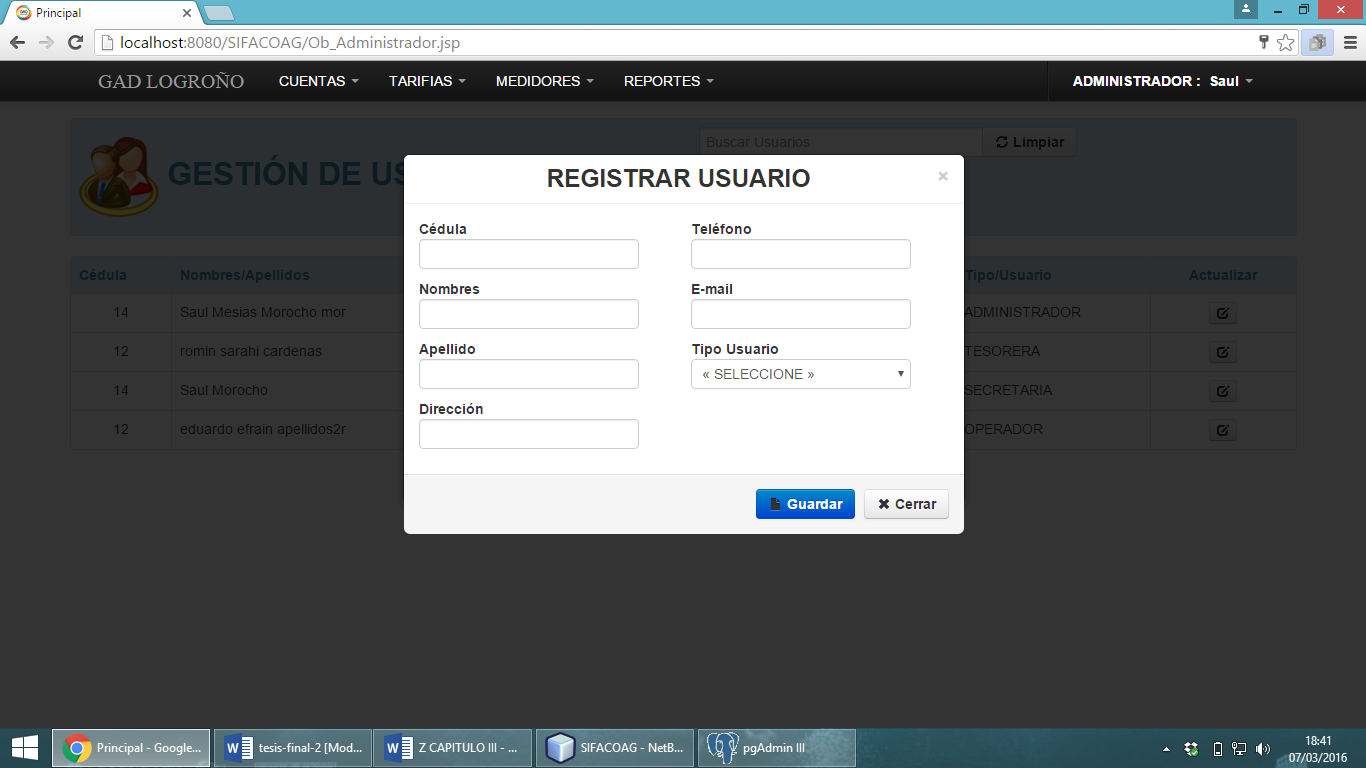


Figura 44 – 2 Registro usuarios

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Asignar Roles de Usuarios*



Figura 45 – 2 Asignación de Roles

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Registro de Zonas*

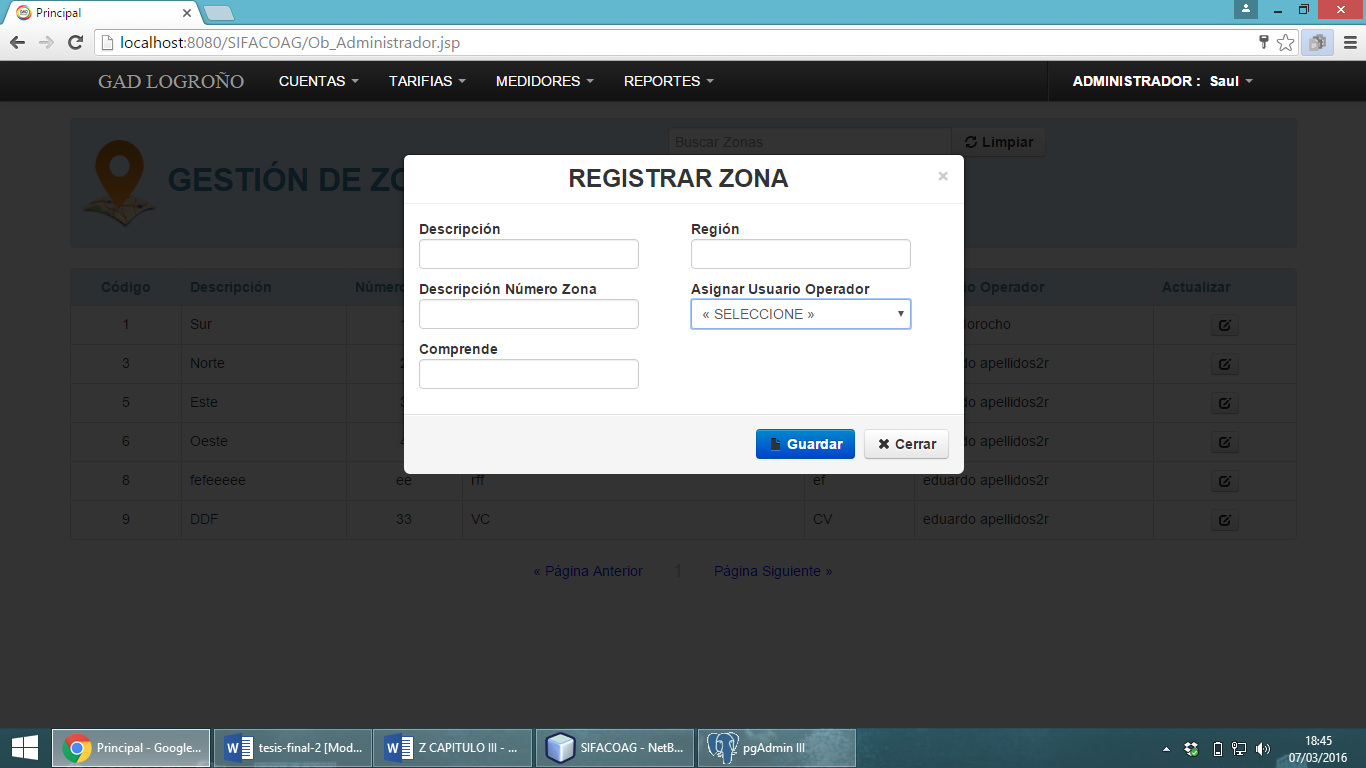


Figura 46 – 2 Registro zonas

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Registro Tipo Tarifas*

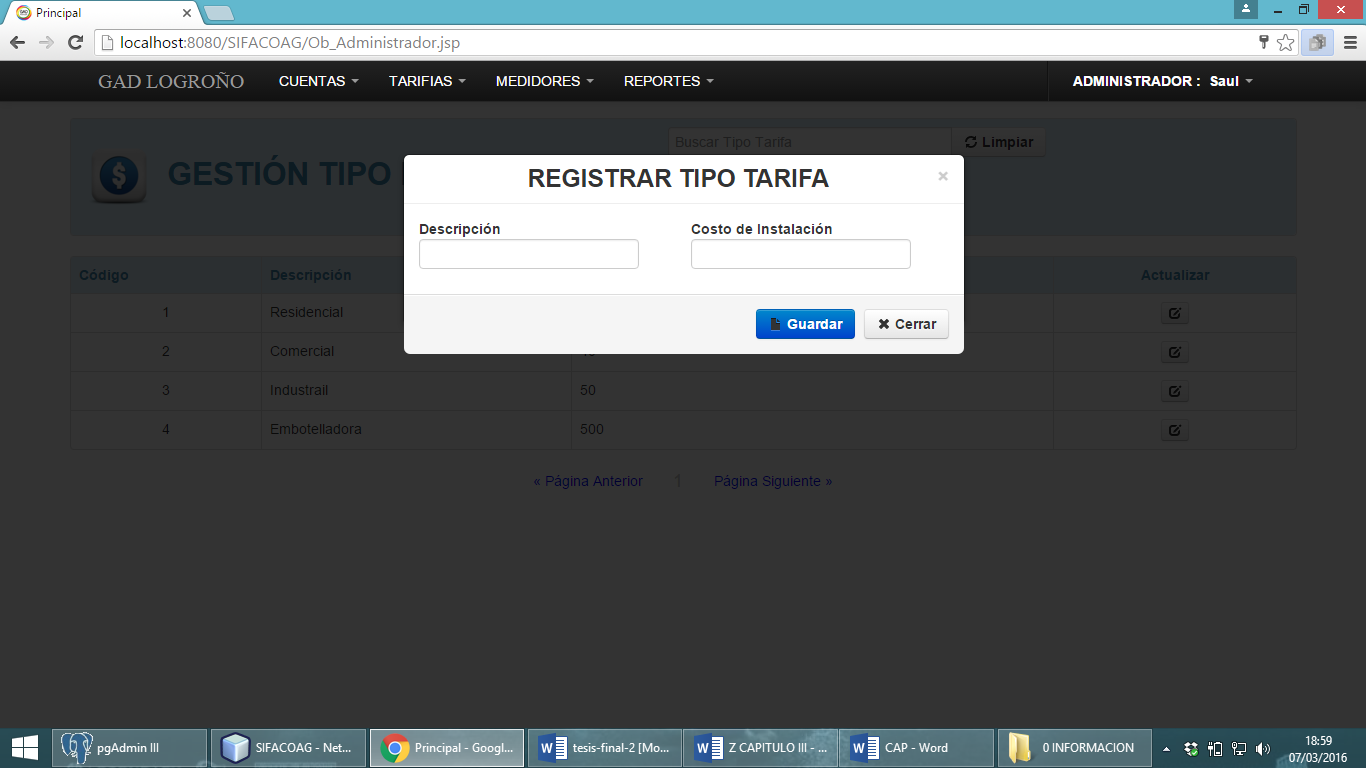


Figura 47 – 2 Registro tipo tarifas

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Registro de Tarifas*

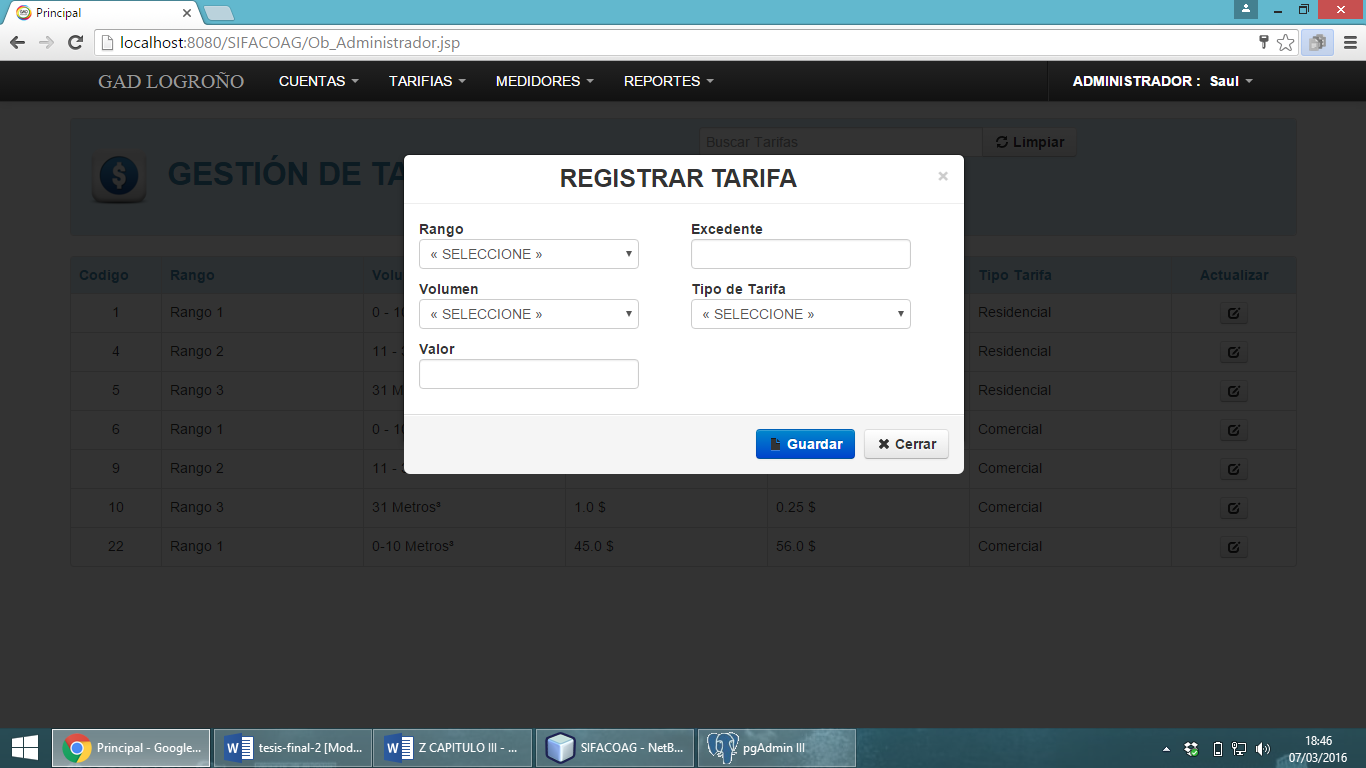


Figura 48 – 2 Registro tarifas

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Registrar Cuenta de Medidor*

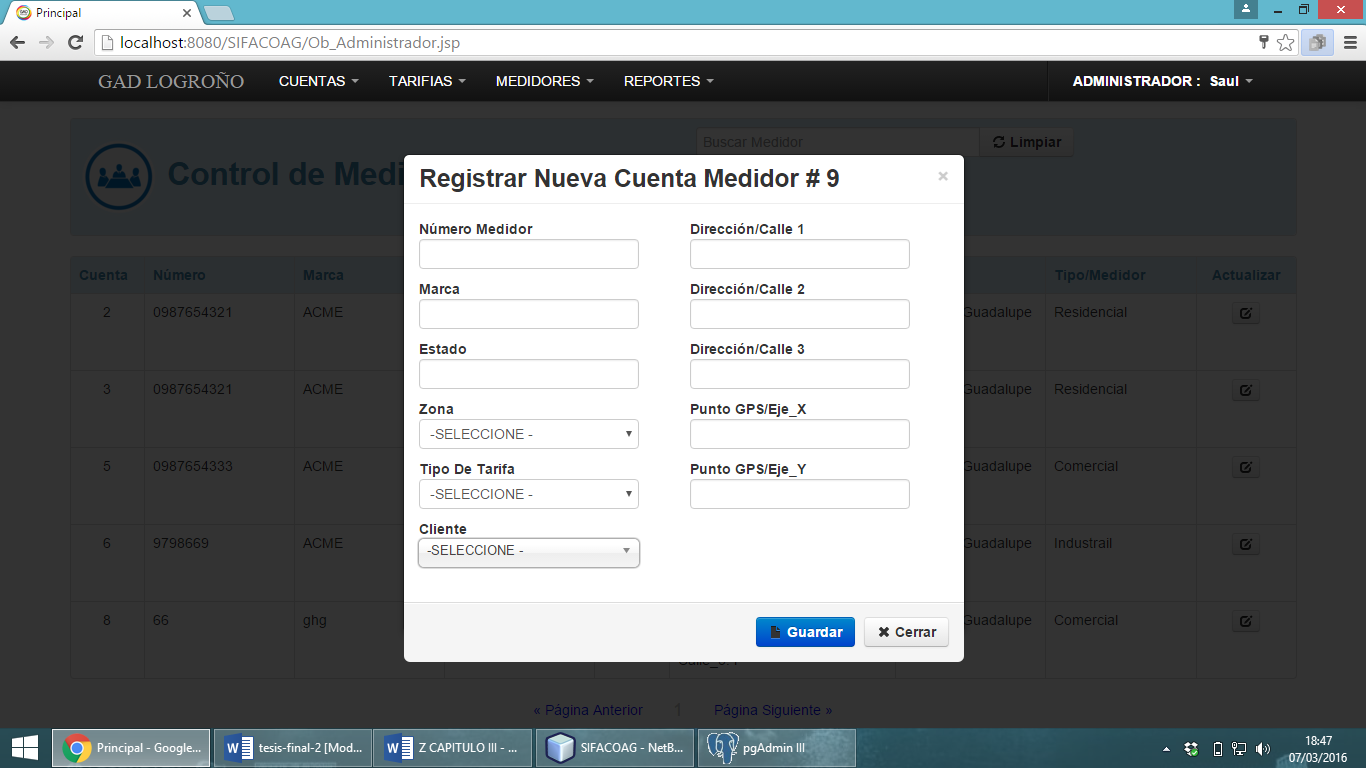


Figura 49 – 2 Registro cuenta medidor

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Consulta de historial del cliente*

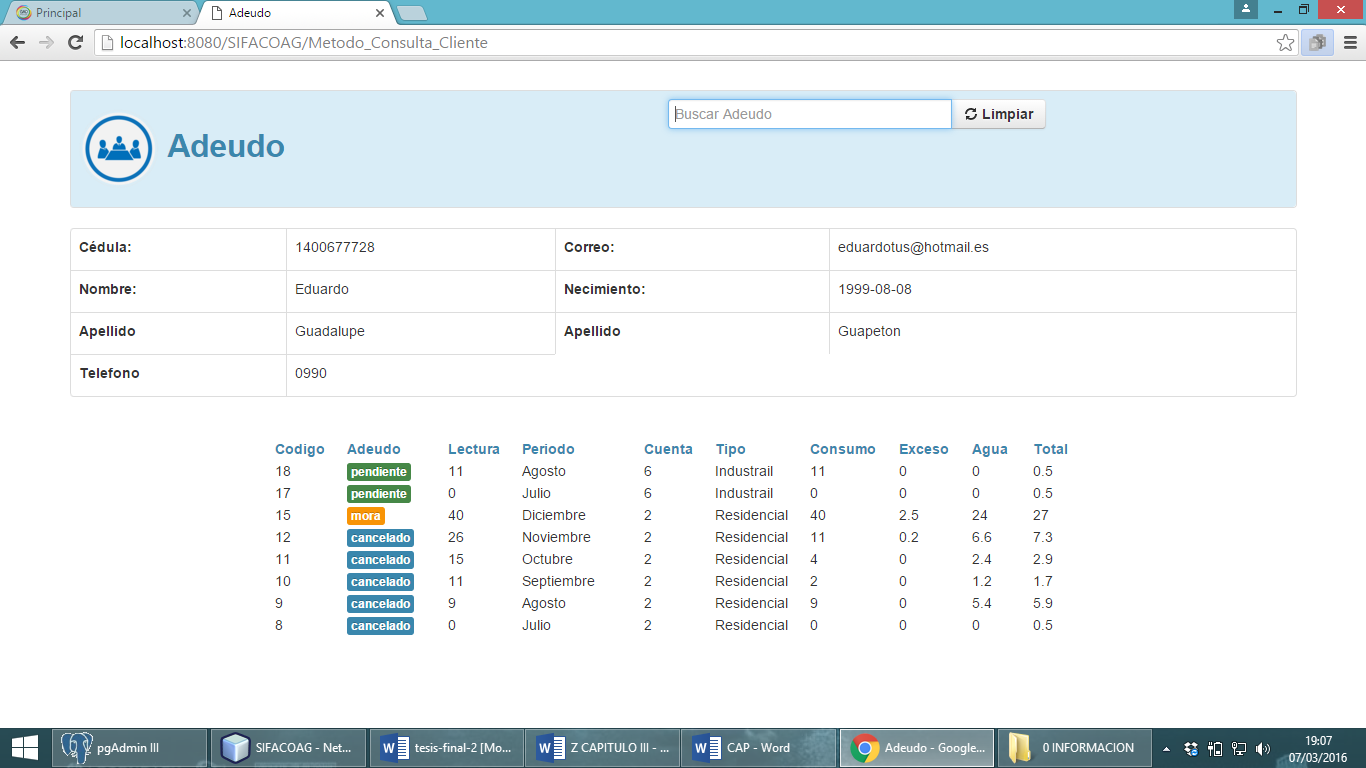


Figura – 2 Consulta historial del cliente

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Generar Reportes*



Figura 51 – 2 Generar reportes

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

### ***Diseño físico***

#### Diagrama de implementación



Figura 52 – 2 Diagrama de implementación

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### Modelo físico de base de datos

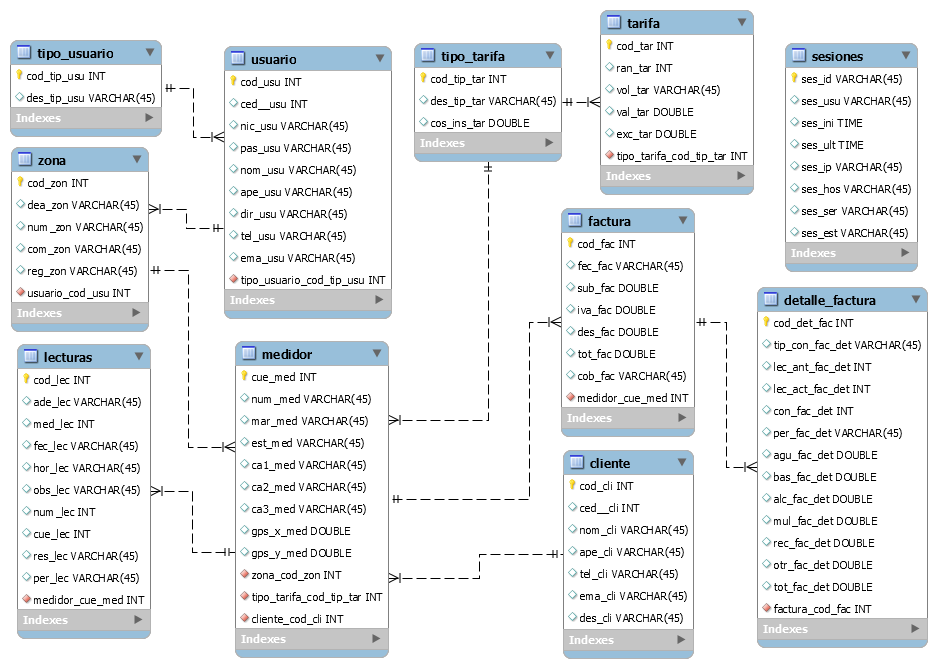
**

Figura 53 – 2 Modelo físico de base de datos

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

## **Desarrollo**

### ***Estándar de diseño de base de datos***

#### **Nombre de los objetos de la base de datos**

Para la identificación de los objetos dentro de la base de datos se manejar la siguiente nomenclatura:

* Nombre

El nombre de los objetos se escribirá en minúsculas, en caso de ser tablas compuestas se escribirá todo en minúsculas y separado por un guion bajo “\_”

Ejemplo:

Factura

detalle\_factura

* nom\_atr

El nombre de los atributos contenidos en cada una de las tablas se escribirá en minúsculas, escribiendo las 3 primeras letras del atributo seguido de un guion bajo “\_” y luego las 3 primeras letras de la tabla. Ejemplo:

cod\_fac

cod\_fac\_det

#### **Sentencias SQL**

Debido a que se usan funciones para realizar consultas desde la aplicación a la base de datos, estas funciones se escribirán en minúsculas con el siguiente formato. La letra “f” que hace referencia a la función seguida del nombre de la tabla donde se realizara dicha función.

Ejemplo:

f\_factura

### ***Estándar De Programación De Aplicaciones.***

#### **Extensiones de archivos**

**La tabla 26 – 2** muestra la extensiones de los formatos que se utilizan en la aplicación, permitiendo conocer cómo deben ser nombrados y facilitando su utilización.

Tabla 39 – 2 extensiones

|  |  |  |
| --- | --- | --- |
| **TIPO DE ARCHIVO** | **EXTENSIÓN** | **NOMENCLATURA** |
| Java | .java | Conexión.java |
| Java Server Pages | .jsp | Consulta.jsp |
| Java Script | .js | Factura.js |
| Hojas de estilo | .css | Archivo.css |
| html | .html | Index.html |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### **Convenciones De Nombres**

La convención de nombres hace que los programas sean fáciles de entender y de leer. Estos nos facilitan información sobre la función que cumple una entidad con estos nombres.

*Clases*

Los nombres de las clases serán sustantivos, y deben basarse en los nombres de las tablas que existen en la base de datos en ser posible. Cuando estos nombres sean compuestos deberán ser separados por un guion bajo “\_”, la primera letra de los sustantivos será mayúscula. Se debe mantener los nombres de las clases lo más simples y descriptivos posible, se usaran siempre palabras completas y evitar el uso de acrónimos y abreviaturas.

Ejemplo:

class Acceso\_Cliente

class Acceso\_Empresa

*Métodos*

Los métodos deberán ser verbos y en el caso de ser métodos compuestos se usara guion bajo “\_” para separarlos.

Ejemplo:

insertarEmpresa()

editarCliente()

*Variables*

Las variables dentro de las clases o de los métodos se escribirán en minúsculas en el caso de ser compuestas usara un guion bajo “\_” para separarlas, las constantes podrán ser escritas sin el guion bajo. Los nombres que se asignan a las variables tendrán un significado correspondiente a la tares o función que se realizara con esa variable, esto para que el observador sepa que función cumple cada una.

### ***SIFACOAG***

EL desarrollo tecnológico ha facilitado a las empresas brindar información rápida a sus clientes, las aplicaciones web facilitan a los estamentos públicos y privados realizar sus labores de forma transparente y estas facilitan a los clientes el acceso oportuno a las necesidades que se generen al hacer uso de los servicios de estas empresas.

#### **Capa de datos**

En la capa de datos se encuentra la información o datos que son necesarios para el correcto funcionamiento de la aplicación, para nuestra aplicación se hace uso de PostgreSQL es un motor de base de datos que permite realizar todo tipo de consultas, para este caso se manejaran funciones creadas dentro del motor de base de datos las cuales ya contienen las consultas que deseemos realizar.

**En la figura 54 – 2** se presenta las funciones usadas con el código correspondiente a la función f\_cliente.

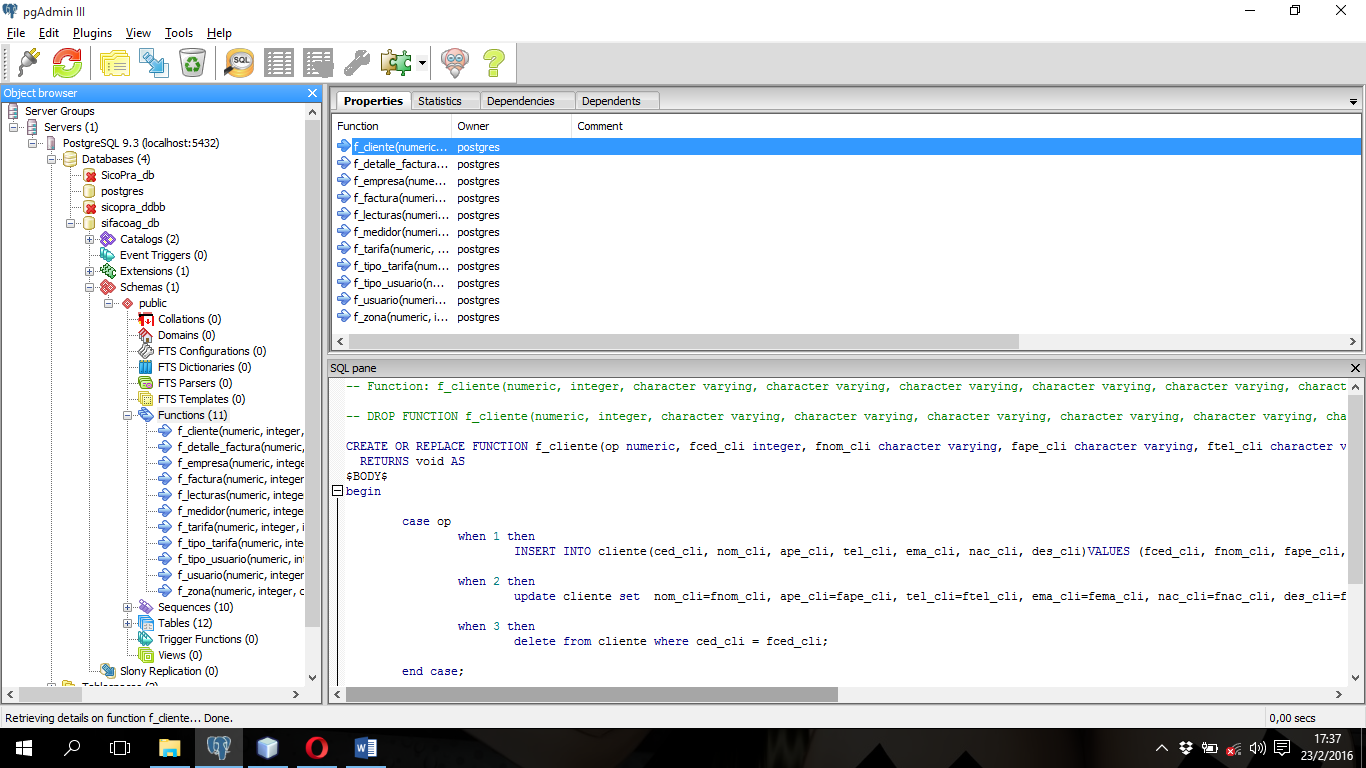


Figura 54 – 2 Código correspondiente a la función cliente

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### **Capa de acceso datos**

La capa de acceso a datos nos sirve de puente entre la capa de negocio de la aplicación y la capa de base datos aquí se encapsula todos los datos o información que va a ser almacenada o modificada en la Basa de Datos, al dividir los procesos por capas logramos que en el caso de que se cambien el motor de base de datos, solo se cambien una instrucción en la capa de acceso a datos en la conexión y que la aplicación no sufra daños.

**En la imagen 55 – 2** se muestra los diferentes archivos .java que cumplen la función de acceso a datos, el IDE usado es Netbeans.

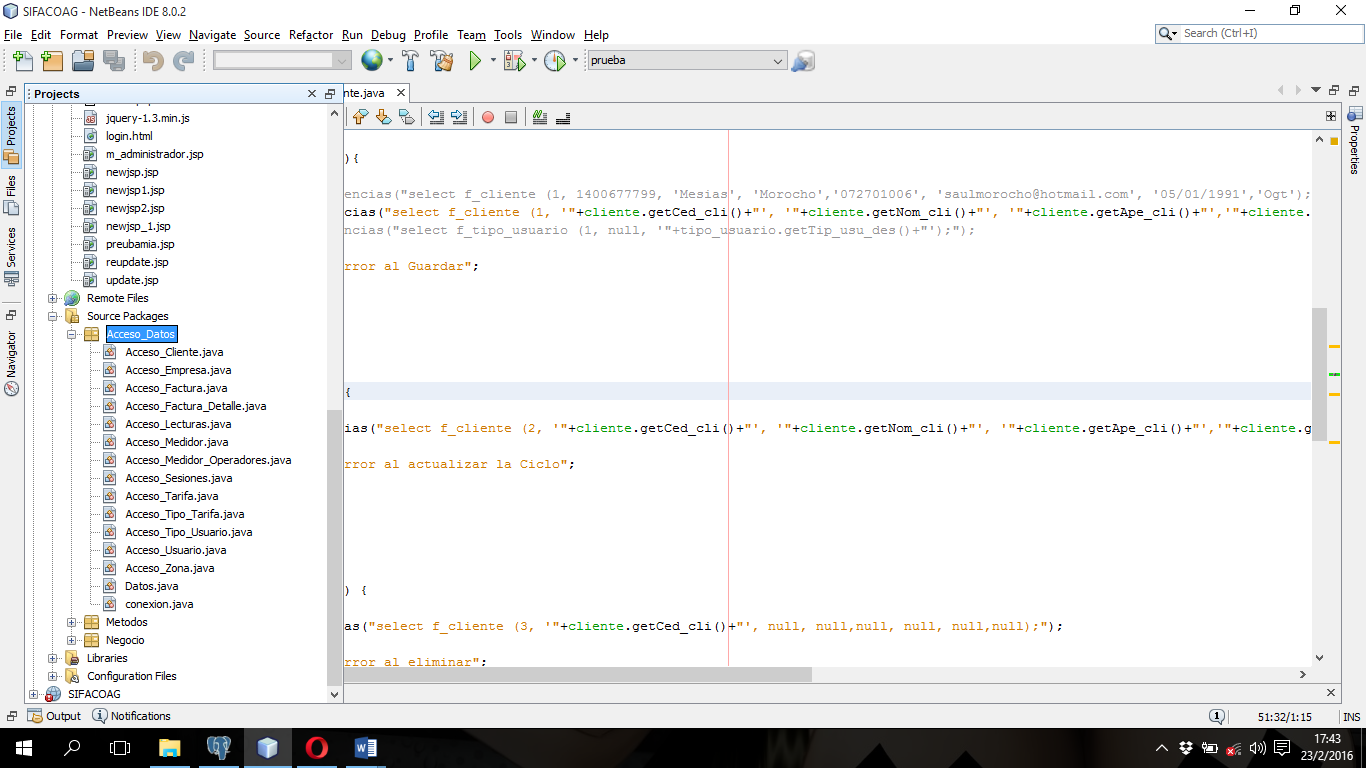


Figura 55 – 2 Capa acceso a datos

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### **Capa de negocios.**

La capa de negocio tiene el acceso a la capa de acceso a datos, esta capa permite que desde la aplicación se pueda mantener la información en memoria hasta que esta pueda ser usada por la capa de acceso a datos y ser enviada al motor de base de datos.

En la imagen 56 – 2 se muestra los distintos archivos java usados en la capa de negocio.

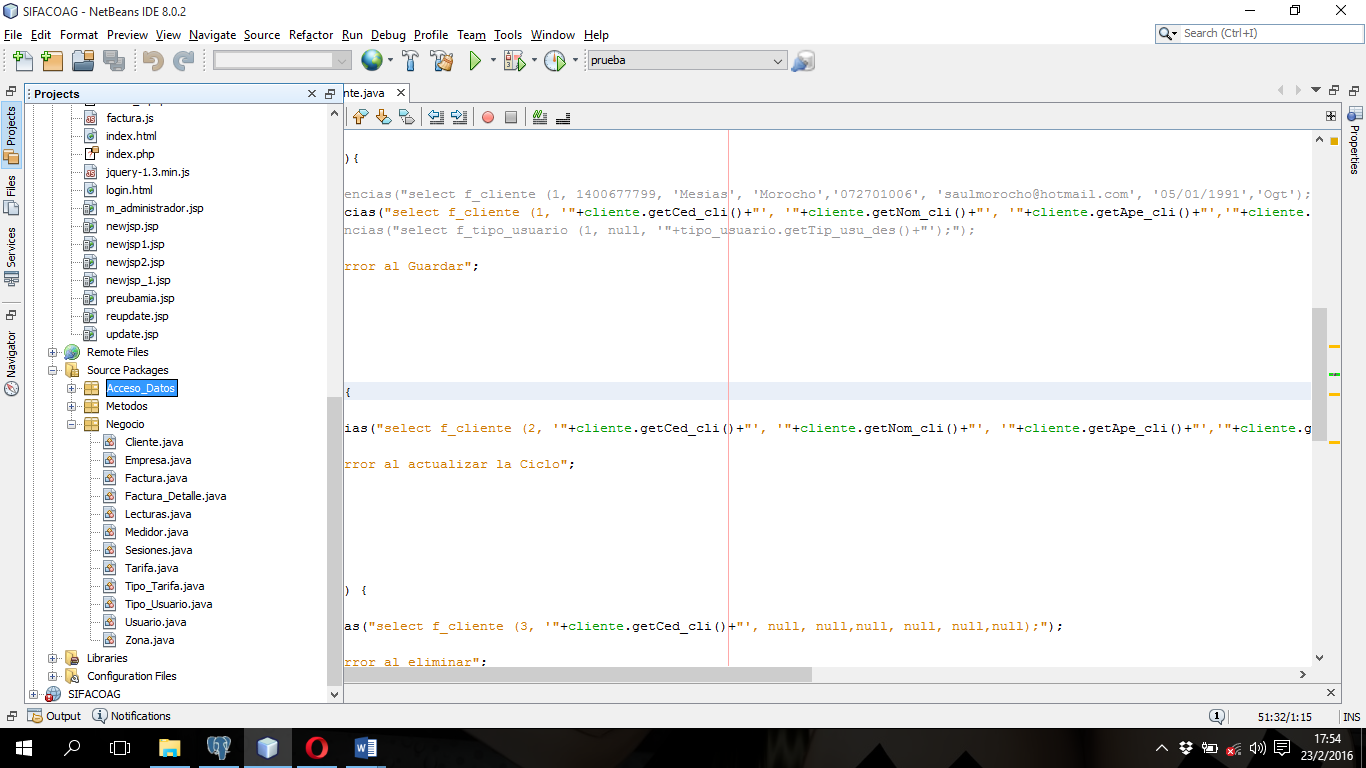


Figura 56 – 2 Capa de negocio

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### **Capa de métodos**

Esta capa contiene los servlets que son usados para permitir a la capa de aplicación insertar, modificar, eliminar o realizar cualquier consulta que requiera, claro esto pasando por las capas antes mencionadas.

**La imagen 57 – 2** muestra la capa de métodos la cual contiene los servlets usados en la aplicación.

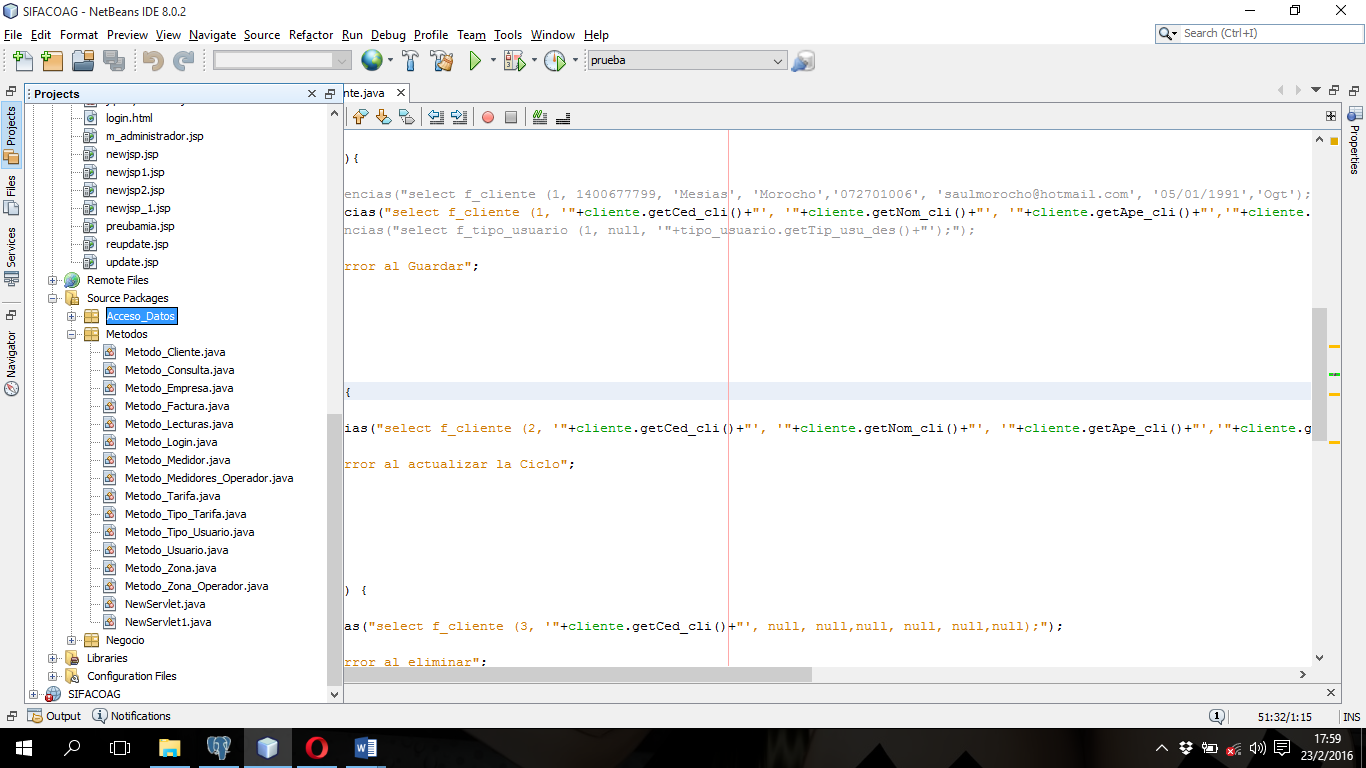


Figura 57 – 2 Servlets

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

#### **Capa de aplicación**

En esta capa se encuentran todos los archivos jsp que sirven como interfaz con el usuario, para aclarar es lo que el usuario observa en su ordenador (Computador personal). Estos archivos se muestran en la figura 58 – 2.

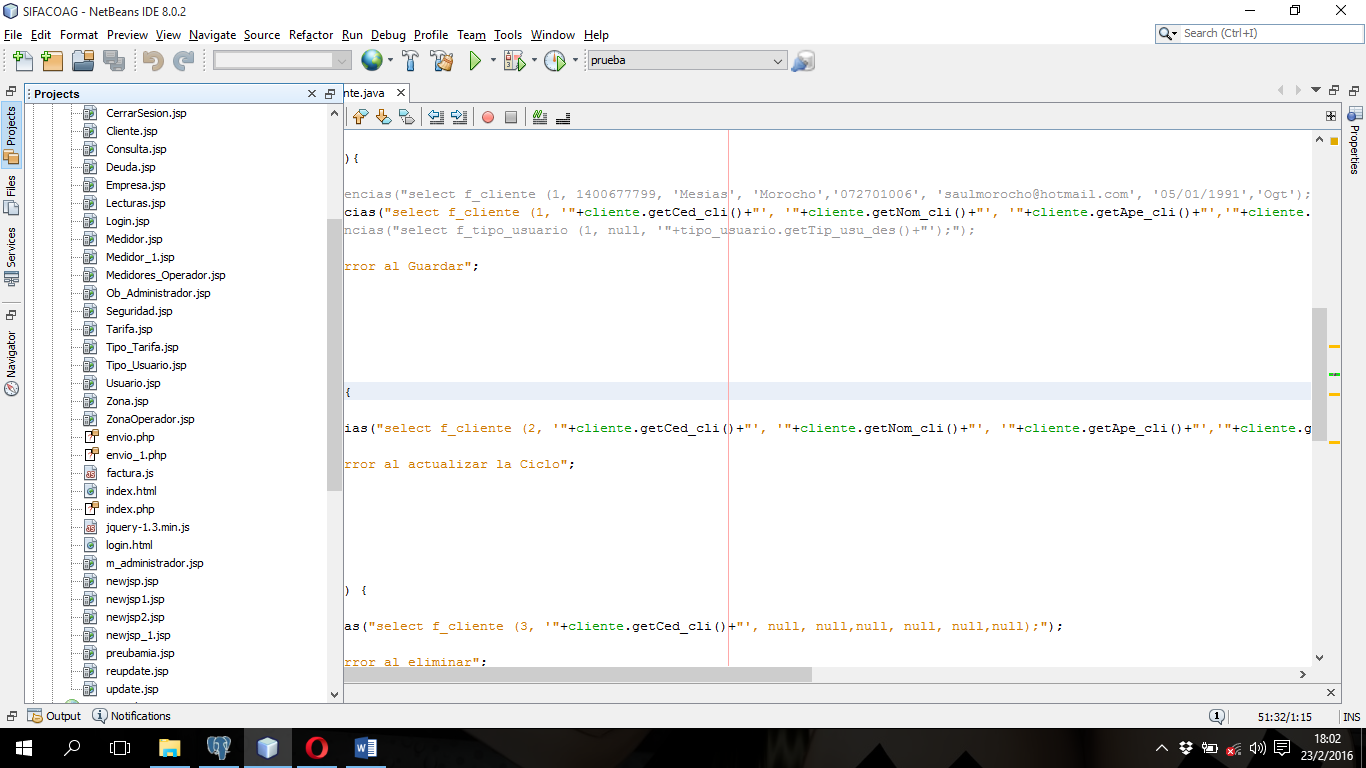


Figura 58 – 2 Capa de aplicación

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

## **Estabilización**

### ***Revisión del sistema***

**SIFACOG** se desarrolló en base a una arquitectura de 3 capas, las cuales son la capa de presentación que es lo que el usuario usa en su PC, la capa de negocio es donde se reciben las peticiones del usuario y se envían las respuestas tras el proceso, la capa de datos es donde residen los datos y es la encargada de acceder a los mismos.

**Capa de presentación – Capa de negocio – Capa de Datos**

Ventajas:

* El desarrollo se puede llevar a cabo en varios niveles
* Permite distribuir el trabajo de desarrollo de una aplicación en niveles de modo que el grupo de trabajo no encuentre inconvenientes al momento de desarrollar.

Código fuente:

A continuación se muestra extractos del código fuente usado en cada una de las capas.

* **Negocio.**

package Negocio;

import static Acceso\_Datos.conexion.conectalistar;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

public class Cliente {

static private int ced\_cli;

private String nom\_cli ;

private String ape\_cli ;

private String tel\_cli ;

private String ema\_cli ;

private String nac\_cli ;

private String des\_cli ;

public int getCed\_cli() {

return ced\_cli;

}

public void setCed\_cli(int ced\_cli) {

this.ced\_cli = ced\_cli;

}

public String getNom\_cli() {

return nom\_cli;

}

public void setNom\_cli(String nom\_cli) {

this.nom\_cli = nom\_cli;

}

public String getApe\_cli() {

return ape\_cli;

}

public void setApe\_cli(String ape\_cli) {

this.ape\_cli = ape\_cli;

}

public String getTel\_cli() {

return tel\_cli;

}

public void setTel\_cli(String tel\_cli) {

this.tel\_cli = tel\_cli;

}

public String getEma\_cli() {

return ema\_cli;

}

public void setEma\_cli(String ema\_cli) {

this.ema\_cli = ema\_cli;

}

public String getNac\_cli() {

return nac\_cli;

}

public void setNac\_cli(String nac\_cli) {

this.nac\_cli = nac\_cli;

}

public String getDes\_cli() {

return des\_cli;

}

public void setDes\_cli(String des\_cli) {

this.des\_cli = des\_cli;

}

static private PreparedStatement pst1 =null;

public static ResultSet BuscarCliente(){

ArrayList<Negocio.Cliente> lista = new ArrayList<Negocio.Cliente>();

ResultSet rs = null;

try{

// String sql=("Select \* FROM ARTICULO where COD\_CATEGORIA ='"+codcartegoria+"' ");

String sql=("Select \* FROM cliente where ced\_cli ='"+ced\_cli+"' ");

pst1 = conectalistar().prepareStatement(sql);

rs=pst1.executeQuery();

}catch (Exception e){

System.out.print(e.getMessage());

}

return rs;

}

}

* **Acceso a datos**

package Acceso\_Datos;

import static Acceso\_Datos.conexion.conectalistar;

import Negocio.Cliente;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

public class Acceso\_Cliente {

private String textoerror;

private Cliente cliente;

Datos dato = new Datos();

public String getTextoerror() {

return textoerror;

}

public void setTextoerror(String textoerror) {

this.textoerror = textoerror;

}

public Cliente getCliente() {

return cliente;

}

public void setCliente(Cliente cliente) {

this.cliente = cliente;

}

public byte insertarCliente(){

try {

//dato.ejecutarSentencias("select f\_cliente (1, 1400677799, 'Mesias', 'Morocho','072701006', 'saulmorocho@hotmail.com', '05/01/1991','Ogt');");

dato.ejecutarSentencias("select f\_cliente (1, '"+cliente.getCed\_cli()+"', '"+cliente.getNom\_cli()+"', '"+cliente.getApe\_cli()+"','"+cliente.getTel\_cli()+"', '"+cliente.getEma\_cli()+"', '"+cliente.getNac\_cli()+"','"+cliente.getDes\_cli()+"');");

// dato.ejecutarSentencias("select f\_tipo\_usuario (1, null, '"+tipo\_usuario.getTip\_usu\_des()+"');");

} catch (Exception e) {

this.textoerror = "Error al Guardar";

e.printStackTrace();

return 0;

}

return 1;

}

public byte editarCliente() {

try {

dato.ejecutarSentencias("select f\_cliente (2, '"+cliente.getCed\_cli()+"', '"+cliente.getNom\_cli()+"', '"+cliente.getApe\_cli()+"','"+cliente.getTel\_cli()+"', '"+cliente.getEma\_cli()+"', '"+cliente.getNac\_cli()+"','"+cliente.getDes\_cli()+"');");

} catch (Exception e) {

this.textoerror = "Error al actualizar la Ciclo";

e.printStackTrace();

return 0;

}

return 1;

}

public byte eliminarCliente() {

try {

dato.ejecutarSentencias("select f\_cliente (3, '"+cliente.getCed\_cli()+"', null, null,null, null, null,null);");

} catch (Exception e) {

this.textoerror = "Error al eliminar";

e.printStackTrace();

return 0;

}

return 1;

}

public static ResultSet ListarClientes(){

ResultSet rs = null;

try{

String sql=("select \* from cliente order by \"ced\_cli\" ASC");

// ResultSet rs=stm.executeQuery("select \* from tipo\_usuario order by \"cod\_tip\_usu\" ASC ");

PreparedStatement pst = conectalistar().prepareStatement(sql);

rs=pst.executeQuery();

}catch (Exception e){

System.out.print(e.getMessage());

}

return rs;

}

public static ResultSet DescripcionCliente(int consultaCliente) {

ResultSet rs = null;

try{

String sql="SELECT \* FROM cliente where ced\_cli = "+consultaCliente+";";

PreparedStatement pst = conectalistar().prepareStatement(sql);

rs=pst.executeQuery();

}catch (Exception e){

System.out.print(e.getMessage());

}

return rs;

}

}

* **Métodos**

package Metodos;

import Acceso\_Datos.Acceso\_Cliente;

import Negocio.Cliente;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet(name = "Metodo\_Cliente", urlPatterns = {"/Metodo\_Cliente"})

public class Metodo\_Cliente extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

response.setContentType("text/html;charset=UTF-8");

try (PrintWriter out = response.getWriter()) {

/\* TODO output your page here. You may use following sample code. \*/

Cliente cliente = new Cliente();

Acceso\_Cliente Acccliente = new Acceso\_Cliente();

if (request.getParameter("insertar") != null) {

int ced = Integer.parseInt(request.getParameter("cedula"));

String nombre = request.getParameter("nombre");

String apellido = request.getParameter("apellido");

String telefono = request.getParameter("telefono");

String email = request.getParameter("email");

String nacimiento = request.getParameter("nacimiento");

String descripcion = request.getParameter("descripcion");

cliente.setCed\_cli(ced);

cliente.setNom\_cli(nombre);

cliente.setApe\_cli(apellido);

cliente.setTel\_cli(telefono);

cliente.setEma\_cli(email);

cliente.setNac\_cli(nacimiento);

cliente.setDes\_cli(descripcion);

Acccliente.setCliente(cliente);

if (Acccliente.insertarCliente()== 1) {

System.out.println("Registrado Correctamente");

} else {

System.out.println("Registrado Incorrectamente"+Acccliente.getTextoerror());

}

}

else if (request.getParameter("modificar") != null) {

int ced = Integer.parseInt(request.getParameter("cedula"));

String nombre = request.getParameter("nombre");

String apellido = request.getParameter("apellido");

String telefono = request.getParameter("telefono");

String email = request.getParameter("email");

String nacimiento = request.getParameter("nacimiento");

String descripcion = request.getParameter("descripcion");

cliente.setCed\_cli(ced);

cliente.setNom\_cli(nombre);

cliente.setApe\_cli(apellido);

cliente.setTel\_cli(telefono);

cliente.setEma\_cli(email);

cliente.setNac\_cli(nacimiento);

cliente.setDes\_cli(descripcion);

Acccliente.setCliente(cliente);

if (Acccliente.editarCliente()== 1) {

System.out.println("Registrado Correctamente");

} else {

System.out.println("Registrado Incorrectamente"+Acccliente.getTextoerror());

}

}

else if (request.getParameter("eliminar") != null) {

int ced = Integer.parseInt(request.getParameter("cedula"));

cliente.setCed\_cli(ced);

Acccliente.setCliente(cliente);

if (Acccliente.eliminarCliente()== 1) {

System.out.println("Registrado Correctamente");

} else {

System.out.println("Registrado Incorrectamente"+Acccliente.getTextoerror());

}

}

response.sendRedirect("Cliente.jsp");

}

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

* **Datos**

CREATE OR REPLACE FUNCTION f\_cliente(op numeric, fced\_cli integer, fnom\_cli character varying, fape\_cli character varying, ftel\_cli character varying, fema\_cli character varying, fnac\_cli character varying, fdes\_cli character varying)

RETURNS void AS

$BODY$

begin

case op

when 1 then

INSERT INTO cliente(ced\_cli, nom\_cli, ape\_cli, tel\_cli, ema\_cli, nac\_cli, des\_cli)VALUES (fced\_cli, fnom\_cli, fape\_cli, ftel\_cli, fema\_cli, fnac\_cli, fdes\_cli);

when 2 then

update cliente set nom\_cli=fnom\_cli, ape\_cli=fape\_cli, tel\_cli=ftel\_cli, ema\_cli=fema\_cli, nac\_cli=fnac\_cli, des\_cli=fdes\_cli where ced\_cli = fced\_cli;

when 3 then

delete from cliente where ced\_cli = fced\_cli;

end case;

end;

$BODY$

LANGUAGE plpgsql VOLATILE

COST 100;

ALTER FUNCTION f\_cliente(numeric, integer, character varying, character varying, character varying, character varying, character varying, character varying)

OWNER TO postgres;

### ***Plan de pruebas.***

Las pruebas que se realizan a una aplicación informática nos permitirán determinar si está a cumplido con las expectativas de los usuarios y de esta forma corregir cualquier fallo que pueda surgir durante las pruebas.

Para realizar las pruebas se ha solicitado la participación del personal del municipio.

Aspectos a evaluar

**La tabla 27 - 2** muestra los aspectos que estarán sujetos a evaluación así como los usuarios que harán las pruebas.

Tabla 40 – 2 aspectos sujetos a evaluación

|  |  |  |
| --- | --- | --- |
| **ASPECTO** | **ACCIONES** | **USUARIO** |
| Lecturas | Registrar las lecturas | Operador |
| Cobro | Cobro y emisión de facturas | Cajero/a |
| Registro | Registrar, editar o eliminar medidores, usuarios | Administrador |
| Consulta | Realizar consultas de deudas | Cliente |
| Reportes | Emitir reportes | Administrador |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

*Entorno de Pruebas*

Para llevar a cabo de forma correcta las distintas pruebas se definirá un entorno de pruebas.

Servidor

* Servidor Físico corriendo sobre un laptop marca DELL
* Procesador Intel Centrino
* Ram 2 GB
* Disco duro 500 GB
* Sistema Operativo Linux Ubuntu Server
* Servidor Glassfish

*Cliente*

* Procesador Intel Core i5
* Ram 4GB
* Disco Duro 500 GB
* Sistema Operativo Windows
* Navegadores web Chrome, Opera, Mozilla.

Las tareas que se mencionan a continuación se deben realizar sin inconvenientes, esto nos permitirá determinar que la aplicación ha superado las pruebas a realizarse.

* Lecturas: El usuario operador de la lista de medidores que se le presenten seleccionara uno en el cual se le presentara un formulario donde tendrá que llenar con la lectura de ese día del medidor.
* Cobro: El usuario cajero se encargara de realizar la consulta con el número de cedula de un cliente, esta consulta le presentara una lista con los valores que este adeude, luego de esto si el usuario desea cancelar se procederá a realizar el cobro y emitir la factura.
* Registro: El usuario administrador deberá registrar en el sistema nuevos usuarios, medidores, así como editar o eliminar estos.
* Consulta: El usuario cliente podrá realizar consultas de la deuda que este tiene con la institución, tan solo con ingresar su número de cedula. También podrá imprimir facturas de valores ya cancelados.
* Reportes: El usuario administrador podrá emitir reportes que este crea necesarios.

## **Instalación**

### ***Para instalar SIFACOAG se debe usar un servidor con las siguientes características.***

* Procesador Intel Centrino
* Memoria Ram 4 GB
* Disco duro 500 GB
* Conexión a Internet
* Sistema Operativo Ubuntu Server

Instalación de dependencias

**Servidor Glassfish 4.1**

* Instalar unzip

Sudo apt-get install unzip

* Instalar el JDK

Sudo apt-get install openjdk-7-jdk

* Descargar Galssfish

cd/tmp

wget <http://download.java.net/glassfish/4.1.1/release/glassfish-4.1.1.zip>

* Descomprimir el servidor en el directorio opt

Cd /opt

Sudo unzip /tmp/glassfish-4.1.1.zip

* Crear el usuario glassfish

Sudo useradd - -system glassfish –d /opt/glassfish4

* Cambiar el grupo del directorio de Glassfish

Sudo chgrp –R sudo /opt/glassfish4

* Cambiar el dueño de los archivos

Sudo chown –R glassfish glassfish4

* Cambiar los permisos de ejecución para los comandos

Sudo chmod –R +x glassfish4/bin/

Sudo chmod –R +x glassfish4/glassfish/bin/

* Iniciar un dominio

Cd glassfish4

Sudo –u glassfish bin/asadmin start-domain domain1

* Cambiar la clave del usuario para permitir la conexión a la consola web

Sudo –u glassfish bin/asadmin change-admin-password

* Habilitar el sitio seguro de la consola web

Sudo –u glassfish bin/asadmin enable –secure-admin

* Reiniciar el glassfish

Sudo –u glassfish bin/asadmin stop-domain domain1

Sudo –u glassfish bin/asadmin start-domain domain1

* Luego en un navegador web escribir

<https://IPSERVIDOR:4848>

**Instalacion PostgreSQL y PgAdmin**

* Instalación de postgresql

Sudo apt-get install postgresql postgresql-contrib

* Crear usuarios

Sudo –i –u postgres

Psql –d postgres –U postgres

* Cambiaremos la contraseña

Alter user postgres with password ‘12345’;

* Saldremos con \q e ingresaremos como el usuario postgresql

Sudo su – postgres

* Crearemos un nuevo usuario

Createuser usuarioroot

Psql

ALTER USER usarioroot WITH PASSWORD ‘12345’;

* Saldremos /q
* Instalaremos PgAdmin3

Sudo apt-get install pgadmin3

* Luego de que termine la instalación del PgAdmin procederemos a abrir nuestra herramienta en un entorno Grafico y crearemos nuestra base de datos.
* Se nos presenta la pantalla de la figura en la cual daremos clic en el icono del enchufe

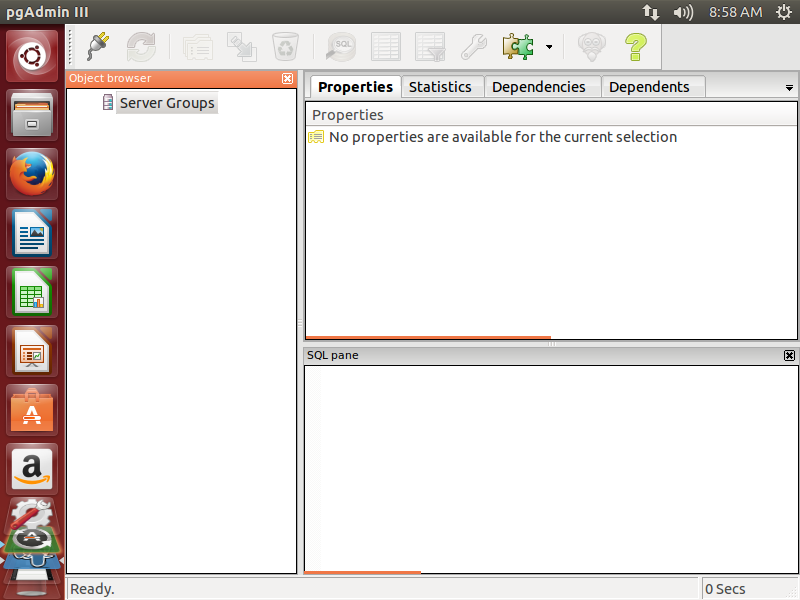


Figura 59 – 2 PgAdmin

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Una vez ahí llenaremos el campo name con “prueba” el campo host con “localhost” y en password pondremos la contraseña ates configurada en nuestro caso es “12345” luego daremos clic a ok.

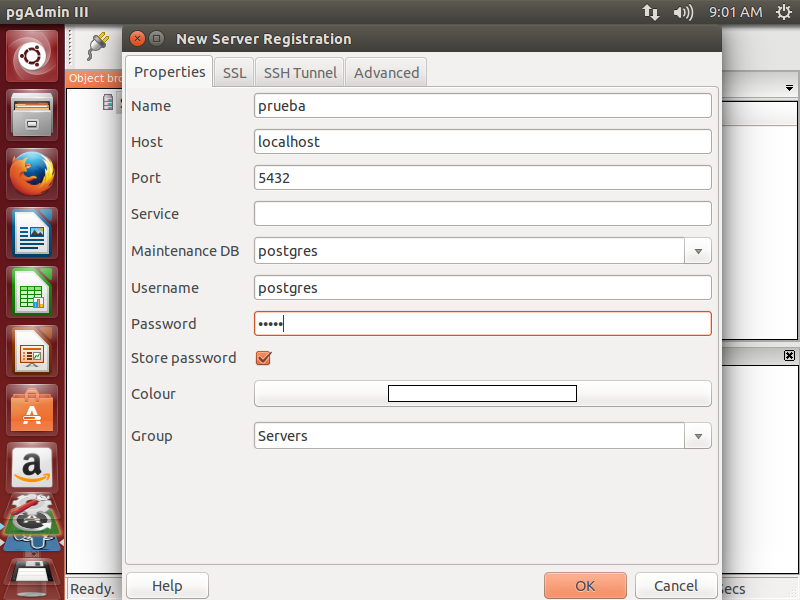


Figura 60 – 2 Creación base de datos

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Se nos presentara lo siguiente en donde iremos hasta donde dice “Databases” y daremos clic derecho luego de esto daremos clic izquierdo en donde dice “créate nex database”

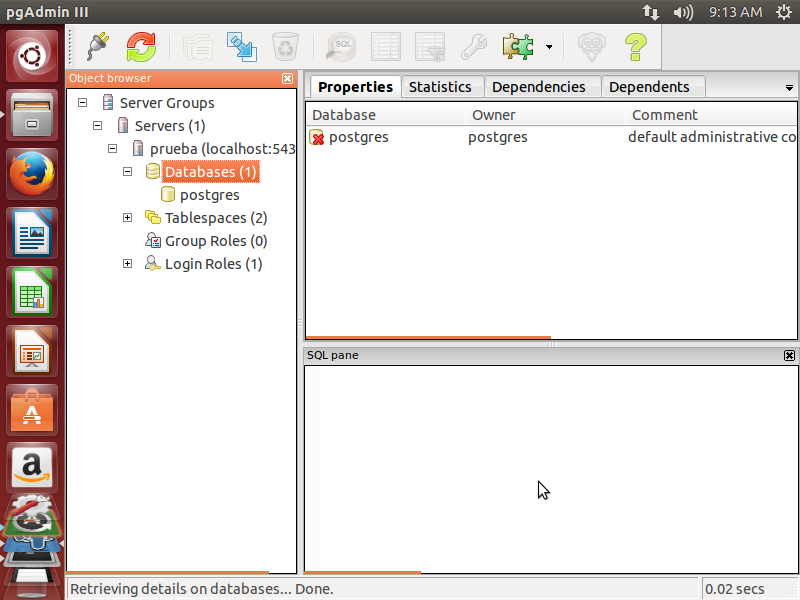


Figura 61 – 2 Ingresando al formulario para la creación de la base de datos.

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Una vez ahí pondremos el nombre de nuestra base de datos donde dice name en nuestro caso “sifacoag\_db” y daremos clic a ok.

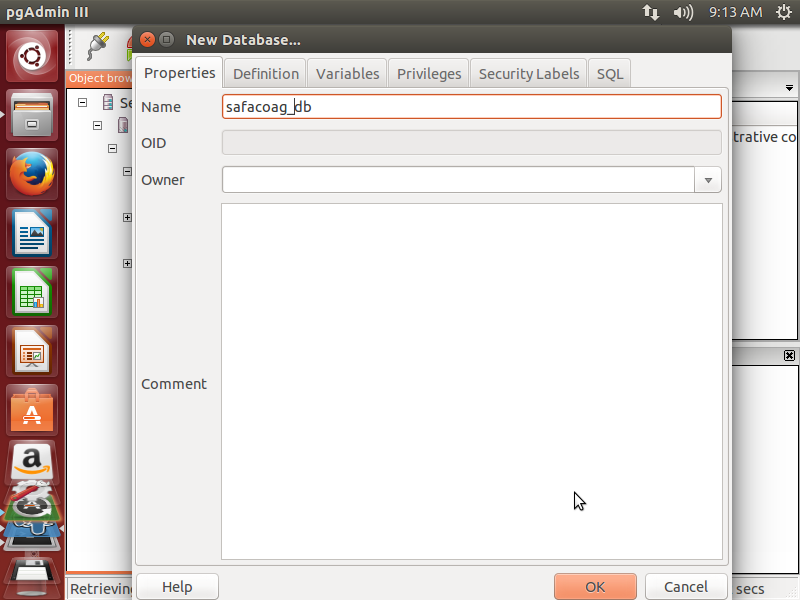


Figura 62 – 2 Creando base de datos

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Una vez hecho eso se nos presentara lo siguiente, en la base de datos que hemos creado daremos clic derecho y luego clic izquierdo en Restore.

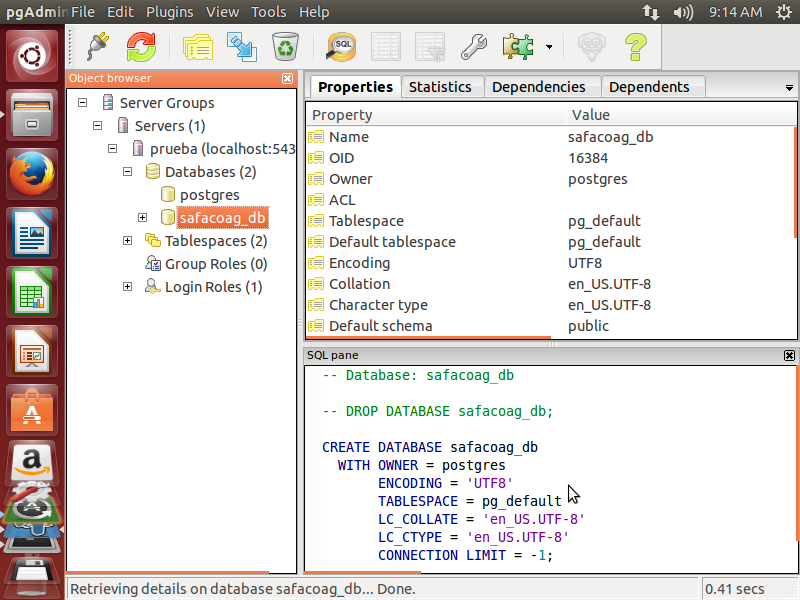


Figura 63 – 2 Restauración

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* En la siguiente pantalla haremos clic en el icono con los 3 puntos “…” ahí iremos a la dirección de nuestro backup de la base de datos lo seleccionaremos y luego daremos clic en abrir.

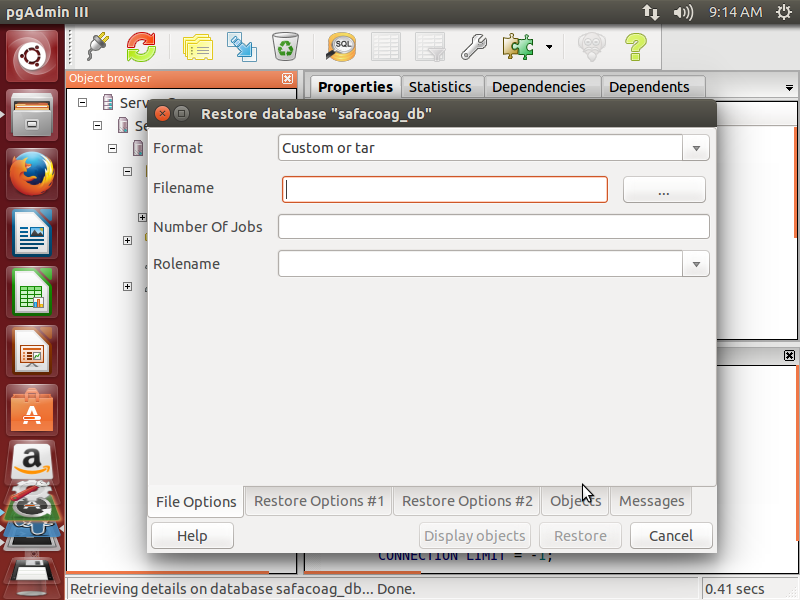


Figura 64 – 2 Escogiendo backup

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Luego daremos clic en Restore y tendremos nuestra base de datos lista en el servidor.

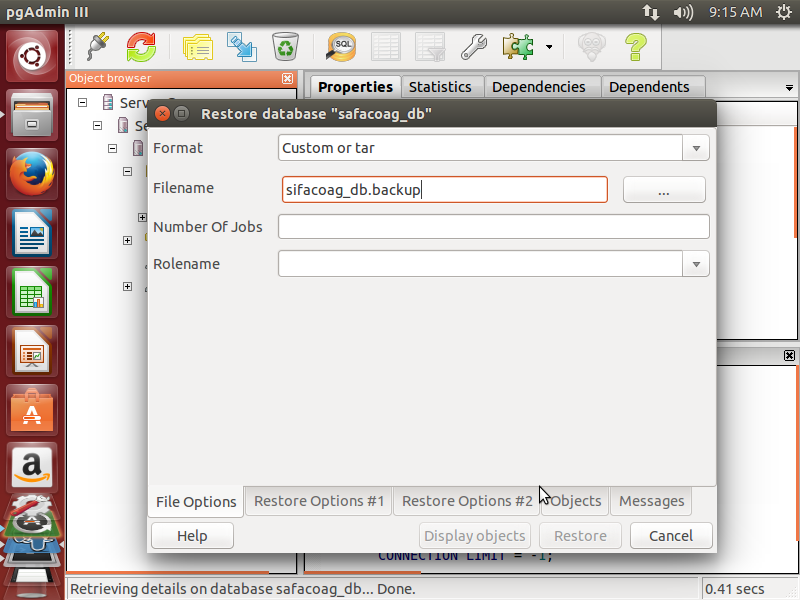


Figura 65 – 2 Restaurando la base de datos

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Subir el archivo .war al servidor

* Desde un equipo que se encuentre en la red del servidor, ingresar en un navegador web la ip del servidor de la siguiente forma

Ipservidor:4848

Mi caso es

192.168.1.15:4848

Les presentara el siguiente formulario en el cual deberán escribir como user “admin” y en password la contraseña que hayan configura para glassfish

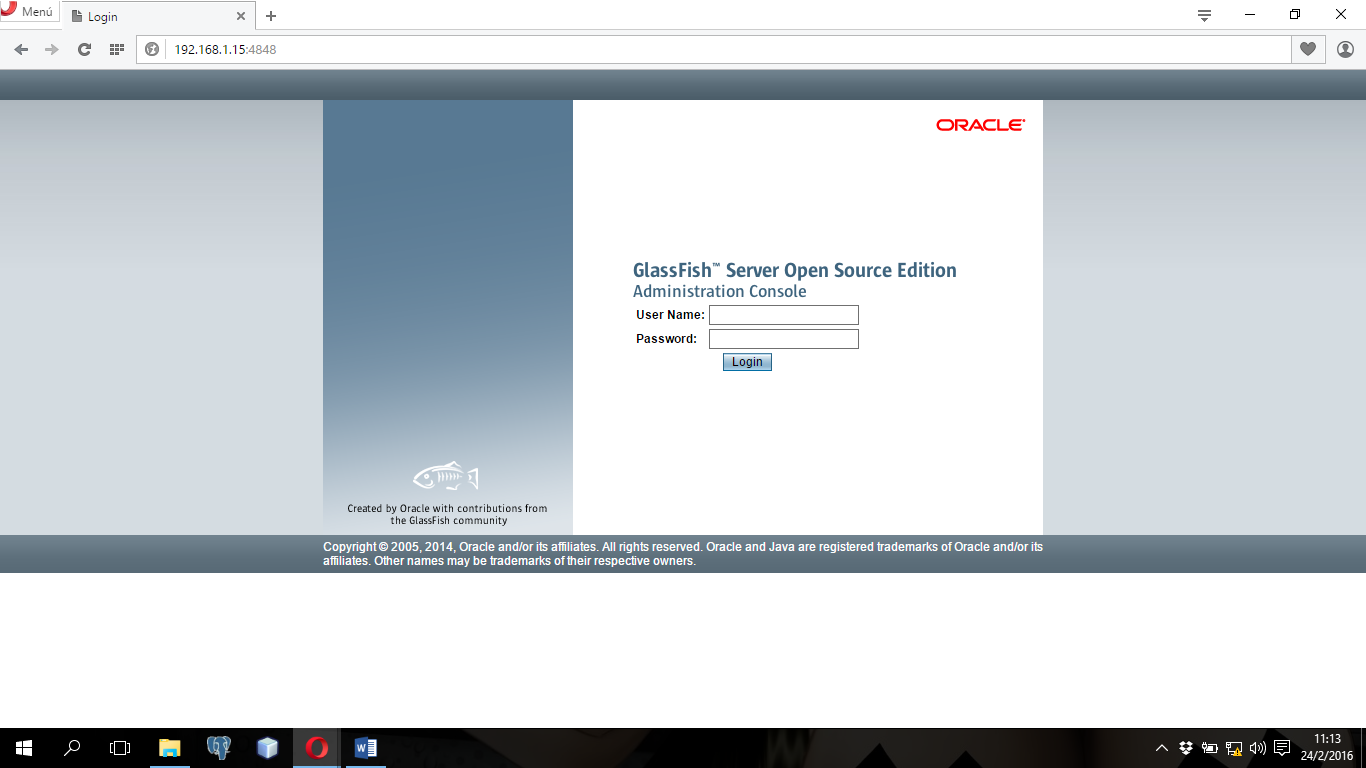


Figura 66 – 2 Formulario para el ingreso al servidor web

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Una vez dentro iremos a la pestaña applications

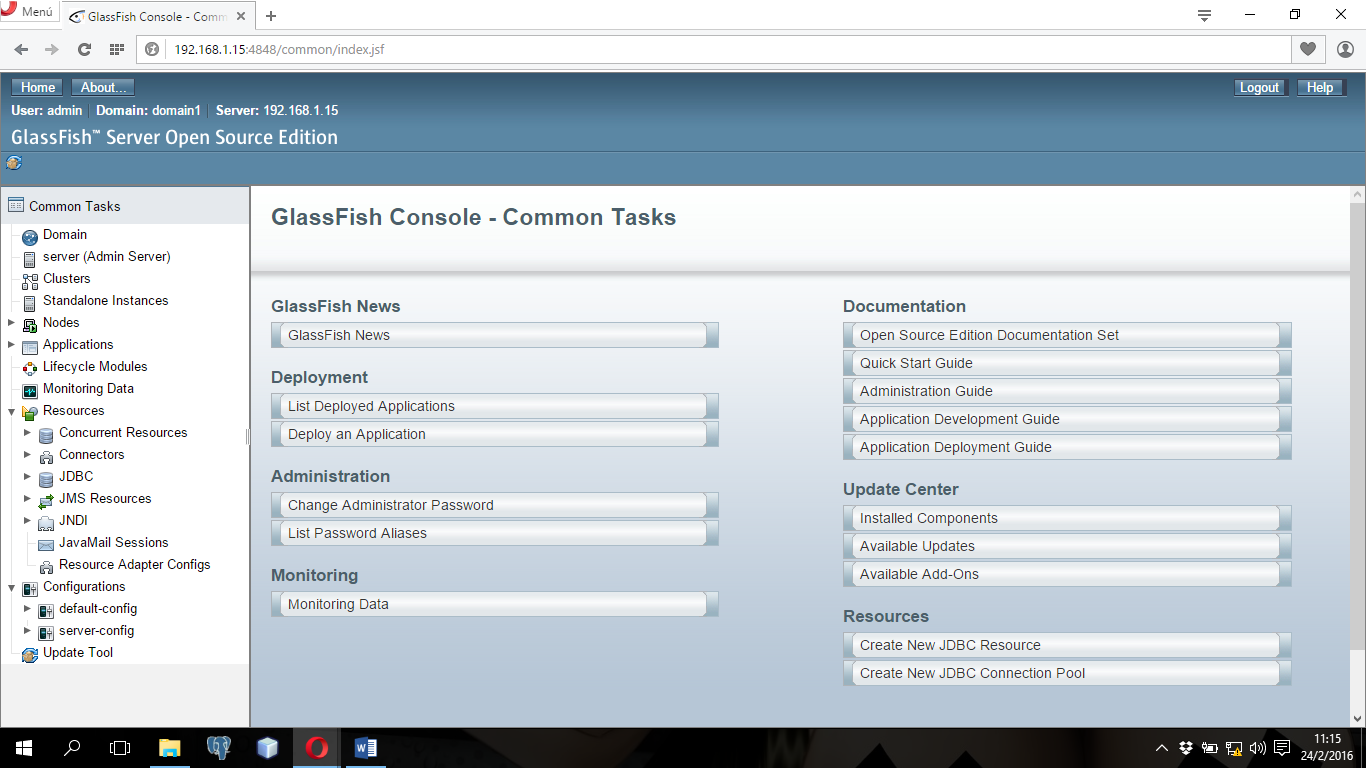


Figura 67 – 2 opciones del servidor

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Luego nos dirigiremos al botón deploy

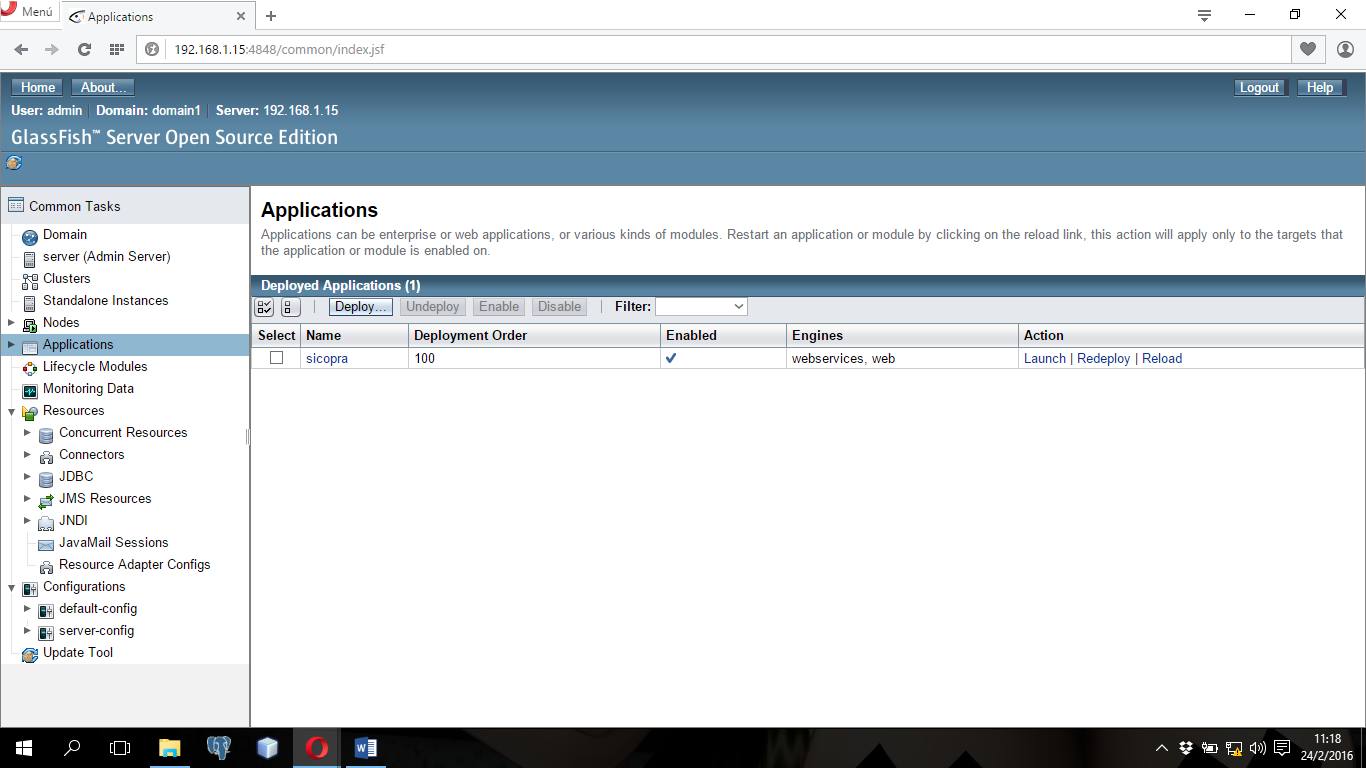


Figura 68 – 2 Opciones para subir el archivo

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Daremos clic donde dice seleccionar archivo

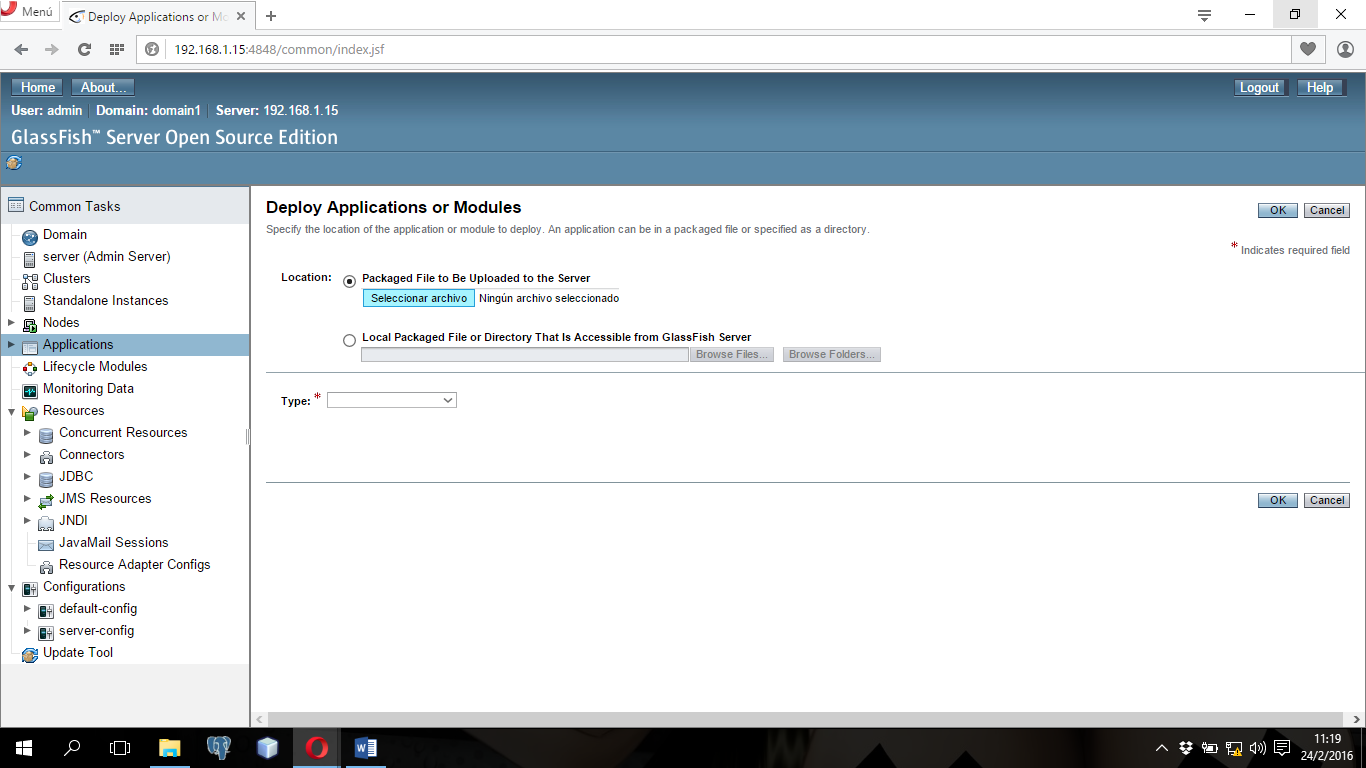


Figura 69 – 2 Opción para buscar el archivo .war

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Luego nos dirigiremos donde se encuentra nuestro archivo .war que por lo general es en la carpeta dist dentro de nuestro proyecto, lo seleccionaremos y haremos clic en abrir.

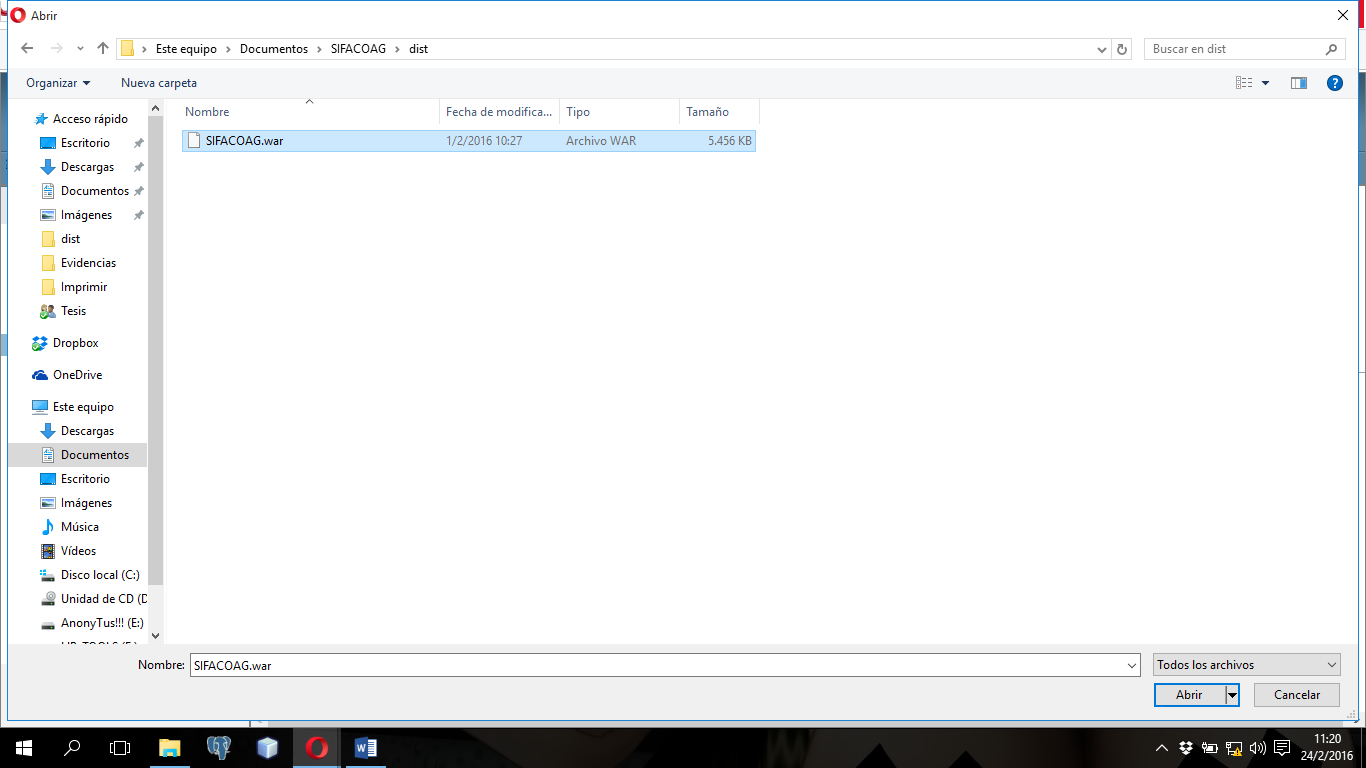


Figura 70 – 2 buscar el archivo .war

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Una vez hecho esto podemos dejar tal cual los casilleros restantes ya que toman el nombre de nuestro archivo o en otro caso podríamos cambiar por el nombre con el que deseemos que se abra nuestra página, esto en el casillero que dice Application name.
* En mi caso en puesto el nombre en minúsculas, luego de esto le damos clic a ok

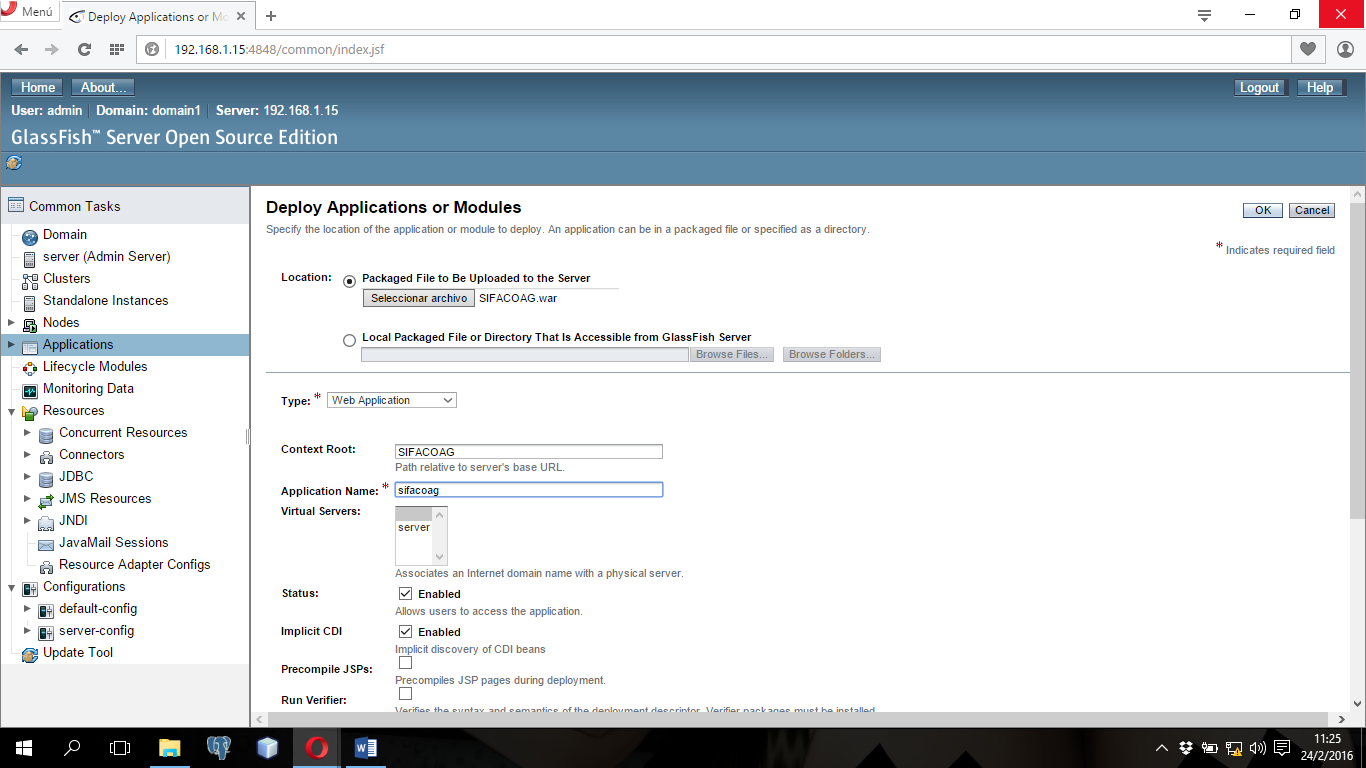


Figura 71 – 2 Subir el archivo .war

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

* Una vez hecho esto nos dirigiremos a nuestro navegador y pondremos lo siguiente

Ipdelservidor:8080/sifacoag/

En mi caso

192.168.1.15:8080/sifacoag/

* En mi caso me muestra la opción de logueo

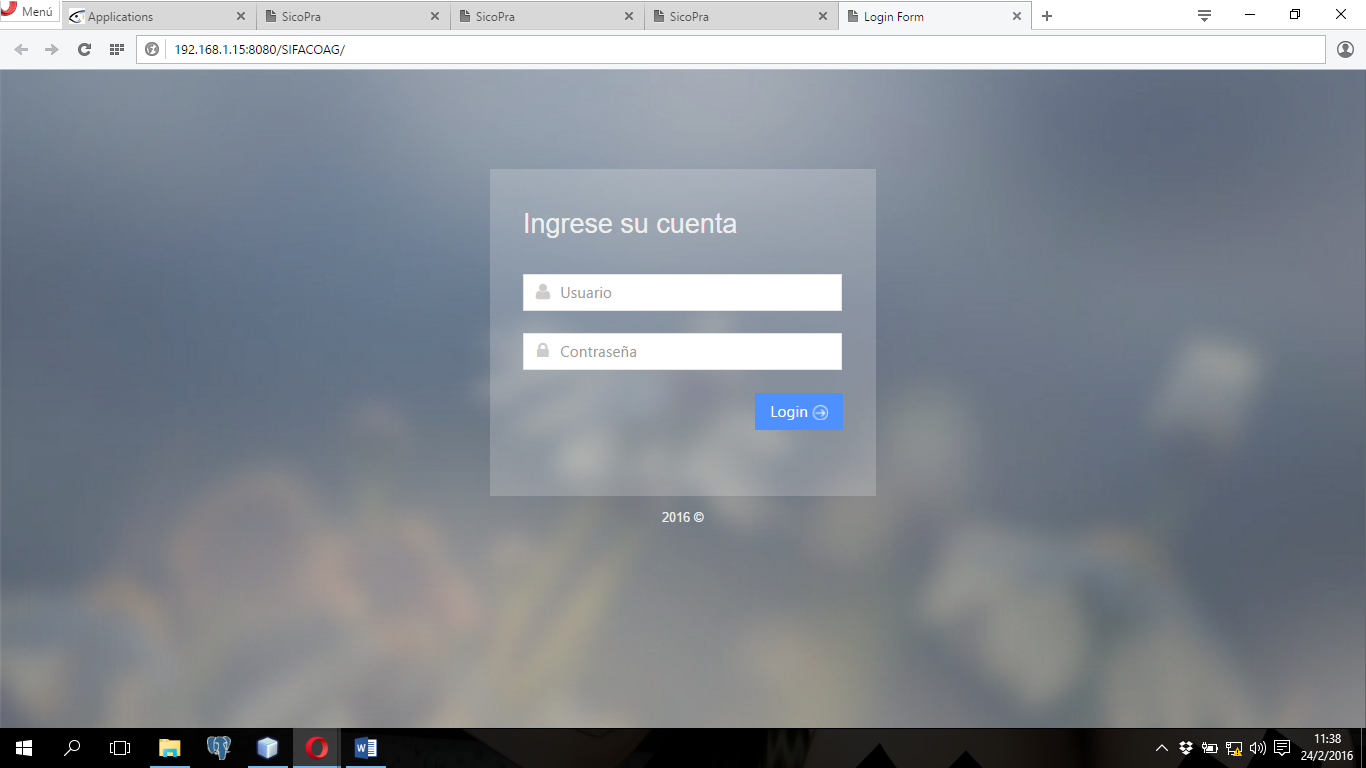


Figura 72 – 2 Entrar al sistema

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

# **CAPITULO III**

## Evaluación del sistema de Facturación y Control del Servicio de Agua Potable bajo la norma NTE INEN-ISO/IEC 9126-1.

## **NORMA NTE INEN-ISO/IEC 9126-1**

### ***Introducción***

Las aplicaciones informáticas están siendo usadas a diario para cubrir las necesidades de los usuarios y mejorar su estilo de vida, para que estas necesidades sean cubiertas de la mejor manera y sin problemas se busca que estas aplicaciones tengan un correcto funcionamiento. Del correcto funcionamiento de estas aplicaciones muchas veces depende el éxito de un negocio, de una empresa y la seguridad de la información de las personas, por ello que se busca que un producto de software sea de calidad y cumpla en lo mayor posible con las necesidades de los usuarios.

Para lograr que un producto de software sea de calidad se definen ciertos parámetros que son definidos dependiendo la finalidad del software. Se debe poner énfasis en usar parámetros que ya hayan sido evalúanos y usados antes, de preferencia que estos sean ampliamente aceptados.

La Norma NTE INEN-ISO/IEC 9126-1 define un grupo de métricas para determinar la calidad de un producto de software, estas métricas están enfocadas en el propósito del software, la facilidad de su uso, la seguridad, etc.

### ***Objeto y campo de aplicación***

La Norma ISO/IEC 9126 nos ayuda a evaluar la calidad de software desde las diferentes perspectivas en las que puede involucrarse nuestro software, siendo estas todo el personal involucrado con el desarrollo, adquisición de requisitos, el uso del software y aquellos que den mantenimiento a este.

Se puede hacer uso del personal que manejara el software así como de aquellos que facilitaron los requisitos, para que evalúen si nuestro software es de calidad esto bajo las métricas que brinda la Norma ISO/IEC 9126-1.

Ejemplo de usos del modelo de calidad definido en esta norma:

* Validar la completitud de una definición de requisitos.
* Identificar los requisitos del software.
* Identificar objetivos para el diseño del software.
* Identificar objetivos para las pruebas del software.
* Identificar requisitos para el seguimiento de la calidad.
* Identificar los criterios de aceptación para un producto de software completado. (NTE INEN-ISO/IEC 9126-1. INGENIERIA DEL SOFTWARE. CALIDAD DEL PRODUCTO SOFTWARE. MODELO DE CALIDAD)

### ***Calidad del producto y ciclo de vida***

La Norma ISO/IEC 9126-1 busca definir la calidad en términos de habilidad para satisfacer necesidades de los usuarios. Muchas veces las necesidades implícitas por los usuarios no reflejan sus necesidades reales, esto debido a que el usuario no es consciente de su realidad, las necesidades pueden cambiar luego de haber sido especificadas, un usuario no define el entorno de trabajo de todos, es difícil consultar sobre las necesidades a todos los usuarios.

Así pues el objetivo de la Norma ISO/IEC 9126-1 es alcanzar una calidad necesaria y suficiente para cubrir las necesidades del usuario a la hora de la entrega el proyecto, esto sin olvidar cumplir en lo posible las necesidades reales de estos.

Las escalas para medir la calidad de los requisitos pueden ser especificadas determinando los diferentes grados en los que los usuarios se ven satisfechos con el cumplimiento de los requisitos. Estas escalas pueden ser: 2 categorías (Satisfactorio o insatisfactorio), 4 categorías (Excede los requisitos, cumple los requisitos, mínimamente aceptable, no aceptable.).

#### **Perspectivas de calidad del producto.**

* Necesidades de calidad del usuario

Un producto de software cumple la calidad que el usuario requiere, solo cuando las necesidades del usuario son retroalimentadas continuamente.

* Requisitos de calidad externa

Estos requisitos son usados como objetivos de validación durante varias etapas del desarrollo del software

* Requisitos de calidad interna

Estos requisitos se usan para medir la calidad de un producto intermedio, estos productos pueden ser: documentos, código fuente, modelos estéticos o dinámicos. También pueden ser usados para definir qué estrategia es la mejor para llevar a cabo el desarrollo.

* Calidad interna

Esta calidad se evalúa frente a los requisitos de calidad interna, algunos detalles de calidad interna se pueden mejorar durante las etapas de implantación, revisión y pruebas.

* Calidad externa estimada

Es la calidad que se estima para el software final o al final de cada etapa de desarrollo.

* Calidad externa

Esta calidad se determina cuando el software es ejecutado, medido y evaluado en ambientes controlados.

* Calidad en uso estimada

Es la calidad que se estima para el software final

* Calidad en uso

Esta calidad es la que define el usuario final al ver que sus necesidades y requerimientos han sido cubiertos de la mejor manera.

### ***Modelo de calidad para calidad interna y externa***

**La figura 73 - 3** muestra los atributos de calidad de software según la norma ISO/IEC 9126-1 y sus sub-características.

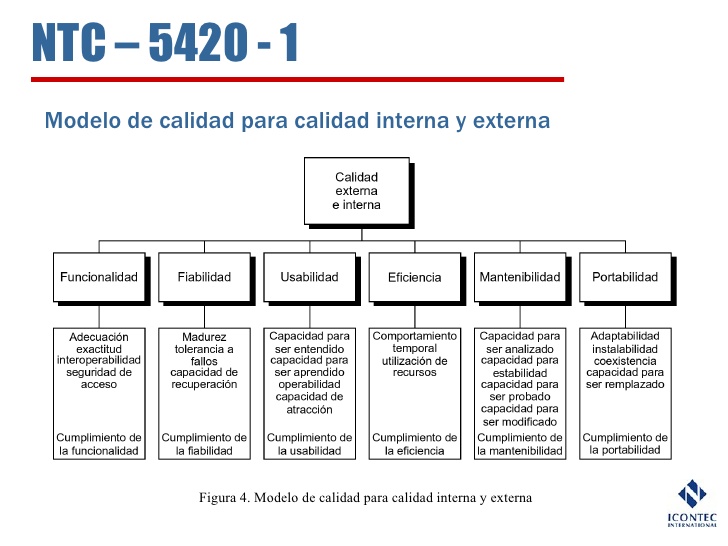


Figura 73 – 3 Métricas de calidad externa e interna

Fuente: (NTE INEN-ISO/IEC 9126-1. INGENIERIA DEL SOFTWARE. CALIDAD DEL PRODUCTO SOFTWARE. MODELO DE CALIDAD)

#### **Funcionalidad**

La capacidad del producto software para proporcionar funciones que satisfacen necesidades declaradas e implícitas cuando se usan bajo condiciones especificadas.

##### *Adecuación*

La capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas y objetivos de usuarios especificados.

##### *Exactitud*

La capacidad del producto software para proporcionar los resultados o efectos correctos o acordados, con el grado necesario de precisión.

##### *Interoperabilidad*

La capacidad del producto software para interactuar con uno o más sistemas especificados.

##### *Seguridad de acceso*

La capacidad del producto software para proteger información y datos de manera que las personas o sistemas no autorizados no puedan leerlos o modificarlos, al tiempo que no se deniegue el acceso a las personas o sistemas autorizados.

##### *Cumplimiento funcional*

La capacidad del producto software para adherirse a normas, convenciones o legislación y prescripciones similares relacionadas con la funcionalidad.

#### **Fiabilidad**

La capacidad del producto software para mantener un nivel especificado de prestaciones cuando se usa bajo condiciones especificadas.

##### *Madurez*

La capacidad del producto software para evitar fallar como resultado de fallos en el software.

##### *Tolerancia a fallos*

La capacidad del producto software para mantener un nivel especificado de prestaciones en caso de fallos software a de infringir sus interfaces especificados.

##### *Capacidad de recuperación*

La capacidad del producto software para reestablecer un nivel de prestaciones especificado y recuperar los datos directamente afectados en caso de fallo.

##### *Cumplimiento de fiabilidad*

La capacidad del producto software para adherirse a normas, convenciones o legislación relacionadas con la fiabilidad.

#### **Usabilidad**

La capacidad del producto software para ser entendido, aprendido, usado y ser atractivo para el usuario, cuando se usa bajo condiciones especificadas.

##### *Capacidad para ser entendido*

La capacidad del producto software que permite al usuario entender si el software es adecuado y cómo puede ser usado para unas tareas o condiciones de uso particulares.

##### *Capacidad para ser aprendido*

La capacidad del producto software que permite al usuario aprender sobre su aplicación.

##### *Capacidad para ser operado*

La capacidad del producto software que permite al usuario operarlo y controlarlo.

##### *Capacidad de atracción*

La capacidad del producto software para ser atractivo al usuario.

##### *Cumplimiento de la usabilidad*

La capacidad del producto software para adherirse a normas, convenciones, guías de estilo o legislación relacionadas con la usabilidad.

#### **Eficiencia**

La capacidad del producto software para proporcionar prestaciones apropiadas, relativas a la cantidad de recursos usados, bajo condiciones determinadas.

##### *Comportamiento de temporal*

La capacidad del producto software para proporcionar tiempos de respuesta, tiempos de proceso y potencia apropiados, bajo condiciones determinadas.

##### *Utilización de recursos*

La capacidad del producto software para usar las cantidades y tipos de recursos adecuados cuando el software lleva a cabo su función bajo condiciones determinadas.

##### *Cumplimiento de la eficiencia*

La capacidad del producto software para adherirse a normas o convenciones relacionadas con la eficiencia

#### Mantenibilidad

La capacidad del producto software para ser modificado. Las modificaciones podrían incluir correcciones, mejoraras o adaptación del software a cambios en el entorno, requisitos y especificaciones funcionales.

##### *Capacidad para ser analizado*

La capacidad del producto software para serle diagnosticadas diferencias o causas de los fallos en el software o para identificar las partes que han de ser modificadas.

##### *Capacidad para ser cambiado*

La capacidad del producto software que permite que una determinada modificación sea implantada.

##### ***Estabilidad***

La capacidad del producto software para evitar efectos inesperados debido a modificaciones del software.

##### *Capacidad para ser probado*

La capacidad del producto software que permite que el software modificado sea validado.

##### *Cumplimiento de la mantenibilidad*

La capacidad del producto software para adherirse a normas o convenciones relacionadas con la mantenibilidad.

#### Portabilidad

La capacidad del producto software para ser transferido de un entorno a otro.

##### *Adaptabilidad*

La capacidad del producto software para ser adaptado a diferentes entornos especificados, sin aplicar acciones o mecanismos distintos de aquellos proporcionados para este propósito por el propio software considerado.

##### *Instalabilidad*

La capacidad del producto software para ser instalado en un entorno especificado.

##### *Coexistencia*

La capacidad del producto software para coexistir con otro software independiente, en un entorno común compartiendo recursos comunes.

##### *Capacidad para reemplazar*

La capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito, en el mismo entorno.

##### *Cumplimiento de portabilidad*

La capacidad del producto software para adherirse a normas o convenciones relacionadas con la portabilidad.

### ***Modelo de calidad para calidad en uso***

**La figura 74 – 3** muestra los atributos de calidad en uso.

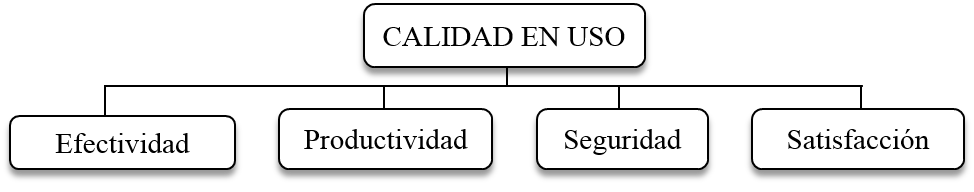


Figura 74 – 3 Atributos de calidad en uso

Fuente: (NTE INEN-ISO/IEC 9126-1. INGENIERIA DEL SOFTWARE. CALIDAD DEL PRODUCTO SOFTWARE. MODELO DE CALIDAD)

La calidad en uso es la calidad desde el punto de vista del usuario. El alcanzar la calidad en uso depende de alcanzar la calidad externa necesaria, que a su vez depende de alcanzar la calidad interna necesaria.

#### Calidad en uso

La capacidad del producto de software para permitir a determinados usuarios alcanzar objetivos especificados con efectividad, productividad, seguridad y satisfacción, en contextos de uso especificados.

##### *Efectividad*

La capacidad del producto software para permitir a los usuarios alcanzar los objetivos especificados con exactitud y completitud, en un contexto de uso especificado.

##### *Productividad*

La capacidad del producto de software para permitir a los usuarios gastar una cantidad adecuada de recursos con relación a la efectividad alcanzada, en un contexto de uso especificado.

##### *Seguridad física*

La capacidad del producto software para alcanzar niveles aceptables del riesgo de hacer daño a personas, al negocio, al software, a las propiedades o al medio ambiente en un contexto de uso especificado.

##### *Satisfacción*

La capacidad del producto software para satisfacer a los usuarios en un contexto de uso especificado.

### ***Las seis características ISO de calidad del software***

Los requisitos para la selección de las características descritos en la Norma ISO/IEC 9126 fueron:

* Cubrir conjuntamente todos los aspectos de la calidad que surjan de la definición de calidad ISO.
* Describir la calidad del producto con un mínimo solape.
* Ser lo más cercano posible a la terminología establecida.
* Formar un conjunto de seis a ocho características como máximo, por razones de claridad y facilidad de uso.
* Identificara áreas o atributos de los productos software que serán refinados posteriormente. (NTE INEN-ISO/IEC 9126-1. INGENIERIA DEL SOFTWARE. CALIDAD DEL PRODUCTO SOFTWARE. MODELO DE CALIDAD)

## **EVALUACIÓN**

### *Modelo de calidad para calidad externa e interna*

El Sistema de Facturación y Control de Cobro de Servicio de Agua Potable “SIFACOAG” será evaluado bajo las métricas y sub-métricas que establece la **Norma ISO/IEC 9126-1** tomando en cuenta los siguientes indicadores de evaluación en cada métrica:

* *Excede los requisitos.*

Cuando las funciones del sistema cumplen los requisitos del usuario y exceden sus expectativas positivamente.

* *Cumple los requisitos.*

Cuando se cumple al pie de la letra con la funcionalidad definida por el cliente en los requisitos.

* *Mínimamente aceptable.*

Cuando las funcionalidades cumplen los requisitos del cliente pero no las necesidades reales.

* *No aceptable.*

Cuando la funcionalidad del software no cumple con ninguno de los requisitos establecidos por el cliente.

**De la tabla 41 - 3 hasta la tabla 46 – 3** se detalla la tabulación de las encuestas realizadas al personal del Área de Catastros y Avalúos (Área encargada del Servicio de Agua Potable) del Municipio del Cantón Logroño, cabe recalcar que dichas encuesta se realizaron a un total de 10 personas.

Para las tabulaciones se toma en cuenta el número de personas encuestadas y se asigna un punto respectivamente a cada encuesta (10 puntos en total según las encuestas realizadas) y se multiplica por el número total de sub-métricas que posee cada Métrica, teniendo así la población total a considerar para realizar el porcentaje de aceptación para la aplicación.

**La tabla 41 - 3** muestra la tabulación de las encuestas pertenecientes a la métrica Funcionalidad, teniendo **50 puntos** como total de población (10 puntos \* 5 sub-métricas).

Tabla 41 – 3 Métrica Funcionalidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MÉTRICA FUNCIONALIDAD | | | | |
| SUB-MÉTRICAS | INDICADORES | | | |
| Excede los requisitos | Cumple los requisitos | Mínimamente  aceptable | No aceptable |
| Adecuación |  | 10 |  |  |
| Exactitud |  | 10 |  |  |
| Interoperabilidad |  | 6 | 4 |  |
| Seguridad de acceso |  | 10 |  |  |
| Cumplimiento Funcional |  | 10 |  |  |
| TOTAL |  | **46** | **4** |  |
| PORCENTAJES |  | 92% | 8% |  |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Figura 75 – 3 Métrica Funcionalidad

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

En la **Figura 75 – 3**, muestra datos estadísticos en los cuales la aplicación **cumple los requisitos**  en un 92% en cuanto a la *Métrica Funcionalidad* según los encuestados, y en un 8% que esta métrica es **mínimamente aceptable** debido a la sub-métrica **Interoperabilidad** con otros sistemas.

**La tabla 42 – 3** muestra la tabulación de las encuestas pertenecientes a la métrica Fiabilidad, teniendo **40 puntos** como total de población (10 puntos \* 4 sub-métricas).

Tabla 42 – 3 Métrica Fiabilidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MÉTRICA FIABILIDAD | | | | |
| SUB-MÉTRICAS | INDICADORES | | | |
| Excede los requisitos | Cumple los requisitos | Mínimamente  aceptable | No aceptable |
| Madurez |  | 10 |  |  |
| Tolerancia a fallos |  | 10 |  |  |
| Capacidad de recuperación |  | 6 | 4 |  |
| Cumplimiento de fiabilidad |  | 10 |  |  |
| TOTAL |  | **36** | **4** |  |
| PORCENTAJES |  | 90% | 10% |  |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Figura 76 – 3 Métrica Fiabilidad

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

En la **Figura 76 – 3**, muestra datos estadísticos en los cuales la aplicación **cumple los requisitos**  en un 90% en cuanto a la *Métrica Fiabilidad*  según los encuestados, y en un 10% que esta métrica es **mínimamente aceptable** debido a la sub-métrica **capacidad de recuperación** de información eliminada en la base de datos.

**La tabla 43 - 3** muestra la tabulación de las encuestas pertenecientes a la métrica Usabilidad, teniendo **50 puntos** como total de población (10 puntos \* 5 sub-métricas).

Tabla 43 – 3 Métrica Usabilidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MÉTRICA USABILIDAD | | | | |
| SUB-MÉTRICAS | INDICADORES | | | |
| Excede los requisitos | Cumple los requisitos | Mínimamente  aceptable | No aceptable |
| Capacidad para ser entendido |  | 10 |  |  |
| Capacidad para ser aprendido |  | 10 |  |  |
| Capacidad para ser operado |  | 10 |  |  |
| Capacidad de atracción |  | 10 |  |  |
| Cumplimiento de usabilidad |  | 10 |  |  |
| TOTAL |  | **50** |  |  |
| PORCENTAJES |  | 100% |  |  |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Figura 77 – 3 Métrica Usabilidad

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

En la **Figura 77 – 3**, muestra datos estadísticos en los cuales la aplicación **cumple los requisitos**  en un 100% en cuanto a la *Métrica Usabilidad*  según los encuestados, ya que la aplicación es amigable para los usuarios y de fácil compresión para los mismos.

**La tabla 44 - 3** muestra la tabulación de las encuestas pertenecientes a la métrica Eficiencia, teniendo **30 puntos** como total de población (10 puntos \* 3 sub-métricas).

Tabla 44 – 3 Métrica Eficiencia

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MÉTRICA EFICIENCIA | | | | |
| SUB-MÉTRICAS | INDICADORES | | | |
| Excede los requisitos | Cumple los requisitos | Mínimamente  aceptable | No aceptable |
| Comportamiento temporal |  | 10 |  |  |
| Utilización de recursos |  | 10 |  |  |
| Cumplimiento de la eficiencia |  | 10 |  |  |
| TOTAL |  | **30** |  |  |
| PORCENTAJES |  | 100% |  |  |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Figura 78 – 3 Métrica Eficiencia

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

En la **Figura 78 – 3**, muestra datos estadísticos en los cuales la aplicación **cumple los requisitos**  en un 100% en cuanto a la *Métrica Eficiencia*  según los encuestados, ya que la aplicación cumple con las funciones realizadas por cada usuario.

**La tabla 45 – 3** muestra la tabulación de las encuestas pertenecientes a la métrica Mantenibilidad, teniendo **50 puntos** como total de población (10 puntos \* 5 sub-métricas).

Tabla 45 – 3 Métrica Mantenibilidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MÉTRICA MANTENIBILIDAD | | | | |
| SUB-MÉTRICAS | INDICADORES | | | |
| Excede los requisitos | Cumple los requisitos | Mínimamente  aceptable | No aceptable |
| Capacidad para ser analizado |  | 10 |  |  |
| Capacidad para ser cambiado |  | 10 |  |  |
| Estabilidad |  | 10 |  |  |
| Capacidad para ser probado |  | 10 |  |  |
| Cumplimiento de la Mantenibilidad |  | 10 |  |  |
| TOTAL |  | **50** |  |  |
| PORCENTAJES |  | 100% |  |  |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Figura 79 – 3 Métrica Mantenibilidad

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

En la **Figura 79 – 3**, muestra datos estadísticos en los cuales la aplicación **cumple los requisitos**  en un 100% en cuanto a la *Métrica Mantenibilidad*  según los encuestados, ya que a la aplicación se le puede agregar o quitar módulos que interactúen entre sí, sin que estos afecten la funcionalidad de la misma.

**La tabla 46 - 3** muestra la tabulación de las encuestas pertenecientes a la métrica Portabilidad, teniendo **50 puntos** como total de población (10 puntos \* 5 sub-métricas).

Tabla 46 – 3 Métrica Portabilidad

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MÉTRICA PORTABILIDAD | | | | |
| SUB-MÉTRICAS | INDICADORES | | | |
| Excede los requisitos | Cumple los requisitos | Mínimamente  aceptable | No aceptable |
| Adaptabilidad |  | 10 |  |  |
| Instalabilidad |  | 10 |  |  |
| Coexistencia |  | 10 |  |  |
| Capacidad para reemplazar |  | 10 |  |  |
| Cumplimiento de la Portabilidad |  | 10 |  |  |
| TOTAL |  | **50** |  |  |
| PORCENTAJES |  | 100% |  |  |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

Figura 80 – 3 Métrica Portabilidad

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

En la **Figura 80 – 3**, muestra datos estadísticos en los cuales la aplicación **cumple los requisitos**  en un 100% en cuanto a la *Métrica Portabilidad*  según los encuestados, ya que a la aplicación se le puede instalar y evaluar en cualquier entorno de trabajo sin que afecte su rendimiento.

# **ANALISIS DE RESULTADOS**

Luego de la culminación del proyecto se pueden tomar las siguientes conclusiones sobre cada una de las etapas del desarrollo así como de sus sub-etapas para así determinar cuáles fueron los logros y experiencias alcanzadas en cada una de estas.

* Una vez investigado cuales son las mejores opciones para realizar el desarrollo de la aplicación hemos tomado a consideración las que contienen mayor soporte, son gratuitas y facilitan a los desarrolladores el uso de las mismas, a continuación describimos brevemente cada una de estas:
  + *JSP.-* Tecnología de java que nos permite crear páginas dinámicas, mezclando HTML y java.
  + *Bootstrap.-* Un framework CSS que facilita la maquetación de sitios WEB en minutos ya que trae consigo archivos CSS lo cual evita al desarrollador la labor de escribir código para su web. También nos ayuda a que nuestra aplicación se despliegue y adapte en cualquier dispositivo donde vaya a ser ejecutada.
  + *Java Script.-* Es un lenguaje de programación que se ejecuta del lado del cliente, se integra dentro del HTML interactúa con este para mejorar las interfaces que se presentan al cliente.
  + *Arquitectura 3 Capas.-* Es una arquitectura que facilita a los desarrolladores la etapa de programación ya que al estar estos programando en diferentes capas las distintas funciones, hace que el desarrollo se agilice y que los cambios que se den no afecte en gran medida a el resto del equipo de desarrollo.
  + *iReport.-* Aplicación que facilita la creación de reportes con parámetros especificados por los clientes.
  + *PostgreSQL.-* Motor de base de datos que permite gestionar y almacenar la información manejada por la aplicación, es gratuito y tiene amplio soporte lo que facilita su uso.
  + *PgAdmin3.-* Aplicación de escritorio que permite y facilita la gestión de la base de datos POstgreSQL.
* En base a las encuestas y reuniones realizadas con el personal de la institución se obtuvo la información que permitió establecer los requisitos y la aprobación de estos por parte de los clientes, por lo cual se logra definir cuáles son las funcionalidades clave con las que debe contar la aplicación, siendo estas:
  + *Administración de usuarios.-* Permitirá crear, modificar o eliminar cuentas a los usuarios que intervienen directamente con la aplicación.
  + *Administración de clientes.-* Permite crear, modificar o eliminar los clientes que hacen uso del servicio de agua potable en el cantón.
  + *Gestión de zonas.-* Permite administrar las zonas en las cuales se encuentran ubicados los distintos medidores de los clientes.
  + *Gestión de tarifas.-* Permite establecer el precio de las tarifas a los distintos servicios en el cobro del agua potable que presta el municipio.
  + *Gestión de medidores.-* Crea y establece una cuenta al cliente por medidor que tiene a su cargo.
  + *Cobro y emisión de facturas.-* Permite cobrar almacenar y emitir una factura al cliente por el cobro mensual del servicio.
  + *Registro de lecturas del medidor.-* Facilita al operador el registro del consumo mensual de cada uno de los medidores.
  + *Consulta del historial de pagos.-* Permite al cliente revisar cual ha sido su consumo y pagos realizados durante el tiempo que ha hecho uso del servicio de agua potable.
  + *Generar reportes.-* Permite a los distintos usuarios de la aplicación crear reportes que ayudaran en la toma de decisiones en la institución.
  + *Asignar permisos a los tipos de usuarios.-* Permite crear perfiles de usuarios ya asignar las funciones que estos pueden realizar dentro de la aplicación.
* Las herramientas usadas facilitaron el desarrollo de la aplicación, ya que java era una de las tecnologías con las que el personal se encontraba familiarizado, javascript es una tecnología fácil de usar y que al contar con mucha información facilita la creación de funciones, postgreSQL es un motor de base de datos con el cual los desarrolladores ya habían realizado varios proyectos, por lo cual al hacer uso de estas herramientas el personal se vio con la confianza necesaria para logra finalizar este proyecto.
* El servidor web bajo el cual se realizaron las pruebas de la aplicación fue instalado en un sistema Linux (Ubuntu Server), este fue glassfish al ser instalado bajo Linux hizo que el personal de desarrollo tuvo que familiarizarse con la forma como se hace uso de Linux, sea esta sus comandos y la falta de interfaces. Luego de instalar el servidor Glassfish y postgreSQL, se procedió a instalar una interfaz gráfica que facilite el uso y administración del servidor, con esto se pudo instalar PgAdmin que facilito el manejo de la base de datos, logrando así implementar el servidor que se utilizaría para la etapa de pruebas.
* La evaluación de la norma se lo realizo, adquiriendo esta y revisando de que constaba, logrando determinar que esta facilita métricas que ayudan a los desarrolladores a determinar si han desarrollado un software que cumpla parámetros de calidad y que cubra en todo con respecto a las necesidades de los clientes, estas métricas y sub-métricas son:
  + Funcionalidad
    - Adecuación
    - Exactitud
    - Interoperabilidad
    - Seguridad de acceso
    - Cumplimiento funcional
  + Fiabilidad
    - Madurez
    - Tolerancia a fallos
    - Capacidad de recuperación
    - Cumplimiento de fiabilidad
  + Usabilidad
    - Capacidad para ser entendido
    - Capacidad para ser aprendido
    - Capacidad para ser operado
    - Capacidad de atracción
    - Cumplimiento de la usabilidad
  + Eficiencia
    - Comportamiento de temporal
    - Utilización de recursos
    - Cumplimiento de la eficiencia
  + Mantenibilidad
    - Capacidad para ser analizado
    - Capacidad para ser cambiado
    - Estabilidad
    - Capacidad para ser probado
    - Cumplimiento de la mantenibilidad
  + Portabilidad
    - Adaptabilidad
    - Instalabilidad
    - Coexistencia
    - Capacidad para reemplazar
    - Cumplimiento de portabilidad
* Después de haber tabulado y evaluado la métricas con l personal que interactúa directamente con la aplicación se obtuvieron los siguientes resultados:

Tabla 47 – 3 Resultados de métricas

|  |  |
| --- | --- |
| MÉTRICA | OBSERVACIÓN |
| Funcionalidad | Aceptación del 92%, el 4% desearía que SIFACOAG interactúe con otros sistemas. |
| Fiabilidad | Aceptación del 90%, el 10% de los encuestados desean que la información eliminada de la BD se pueda recuperar. |
| Usabilidad | El 100% conformes con el cumplimento de los requerimientos. |
| Eficiencia | El 100% conformes con el cumplimento de los requerimientos. |
| Mantenibilidad | El 100% conformes con el cumplimento de los requerimientos. |
| Portabilidad | El 100% conformes con el cumplimento de los requerimientos. |

**Fuente:** GUADALUPE, E., MOROCHO, S., 2016

# **CONCLUSIONES**

1. Mediante los resultados obtenidos de la evaluación de la aplicación (SIFACOAG) bajo la norma NTE INEN-ISO/IEC 9126-1 podemos determinar que la aplicación cumple con los estándares de calidad que dicta dicha Norma, basándonos en los resultados de las encuestas realizadas al personal de la institución.
2. Después de evaluar la Norma NTE INEN-ISO/IEC 9126-1 se logró aplicar con facilidad las métricas de calidad de software que dispone la misma, cumpliendo y garantizando estos estándares en la aplicación SIFACOAC.
3. Luego de culminar con el desarrollo de la aplicación, se realizó las pruebas pertinentes con el personal del Municipio del Cantón Logroño, Provincia de Morona Santiago, determinando que la aplicación cumple satisfactoriamente con los requerimientos especificados por los usuarios.
4. Una fase primordial del uso de la metodología MSF para un buen diseño y desarrollo, es el Análisis de Requerimientos ya que en base de esto se puede modelar la correcta funcionalidad que debe cumplir y satisfacer la aplicación.

# **RECOMENDACIONES**

1. Hacer uso de la norma NTE INEN-ISO/IEC 9126-1 y otros estándares que faciliten y ayuden al desarrollador en la construcción de software de calidad.

1. Para el desarrollo de software es importante que el desarrollador esté actualizando diariamente su conocimiento en nuevas mejoras o modificaciones que se le añadan a las herramientas de su elección.
2. Contratar personal que tenga conocimientos sobre las herramientas utilizadas en el desarrollo para prevenir cualquier circunstancia adversa al correcto funcionamiento de la aplicación.
3. Al personal de la institución hacer uso máximo del potencial de la aplicación del cobro del servicio de agua potable, mantener informada a la población sobre el módulo de consultas.
4. Automatizar los procesos de otros servicios que preste la institución a la ciudadanía y que estos procesos interactúen directamente con la aplicación SIFACOAG.

**BIBLIOGRAFÍA**

Pérez, D. (3 de julio de 2007). *Qué es JavaScript*. Obtenido de: http://www.maestrosdelweb.com/que-es-javascript/

Alegsa, L. (15 de Junio de 2015). *Definición de Aplicación Web*. Obtenido de http://www.alegsa.com.ar/Dic/aplicacion%20web.php

Jorge. L. (2013*) ¿Que es Bootstrap?* Obtenido de http://jorgelessin.com/que-es-bootstrap-y-como-funciona-en-el-diseno-web/

Jose, A. (2015) *Ventajas y desventajas de Bootstrap* Obtenido de http://programacion.jias.es/2015/05/web-¿que-es-el-framework-bootstrap-ventajas-desventajas/

Angel, E. (2011) *Java Script Ventajas y desventajas* Obtenido de http://computacionudla.blogspot.com/2011/07/resena-de-javascript.html

Enrique, G., (2006) *Fomras de usar Java Script* Obtenido de http://aprenderaprogramar.com/index.php?option=com\_content&view=article&id=590%3Aique-es-y-para-que-sirve-javascript-embeber-javascript-en-html-ejercicio-ejemplo-basico-cu00731b&catid=69%3Atutorial-basico-programador-web-html-desde-cero&Itemid=192

Eliseo, P. B. (08 de Diciembre de 2011) *JSP Caracteristicas* Obtenido de http://aplicaciones-web-lenguajes-programaci.blogspot.com/2011/12/jsp.html

José, L. P. (16 de Septiembre de 2015) *Ventajas desventajs de JSP* Obtenido de http://www.registrodominiosinternet.es/2013/08/lenguajes-programacion-web-ventajas.html

Fani, C. (24 de Junio de 2008) *Arquitectura 3 capas* Obtenido de http://es.slideshare.net/Decimo/arquitectura-3-capas

Cristhian, H. (29 de abril del 2005) *Introducción a iReport* Obtenido de http://www.adictosaltrabajo.com/tutoriales/ireport/

Moycas (04 de Mayo de 2011) *¿Que es PostgreSQL?* Obtenido de https://microbuffer.wordpress.com/2011/05/04/que-es-postgresql/

( McGRAW, H. JSP Manual de referencia. Madrid-España: MMI Companies, 2002, 47-48 Obtenido de “Biblioteca ESPOCH”

Juan, O. C. Java Server PAges. Madrid-España: ALFAOMEGA, 2002 Obtenido de “Biblioteca ESPOCH”

Juan, O. C. Diseño de páginas web interactivas con JavaScript. México: Alfaomega, 2000 Obtenido de “Biblioteca ESPOCH”

***GLOSARIO***

**B**

*BOOTSTRAP*: es un framework que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo.

**C**

*COMPILADOR:* Un compilador es un pequeño programa informático, que se encarga de traducir (compilar) el código fuente de cualquier aplicación que se esté desarrollando.

**F**

FRAMEWORKS: Conjunto de herramientas que proporcionan es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, puede servir de base para la organización y desarrollo de software.

**G**

*GITHUB:* es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.

**I**

*IREPOR*T: Es una herramienta visual que sirve para generar ficheros [XML](http://www.ecured.cu/XML) (plantillas de informes) utilizando la herramienta de generación de informes [JasperReport](http://www.ecured.cu/JasperReport).

**J**

*JAVA:* es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios.

**L**

*LINUX:* es un sistema operativo, una gran pieza de software que controla un computador. Es parecido a Microsoft Windows, pero completamente libre.

**M**

*MAQUINAR VIRTUAL:* Una máquina virtual es un software que **emula un ordenador justo como si fuese uno real.**

**N**

*NETBEANS:* es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java.

**O**

*OPEN SOURCE:* Código Abierto es un término que se aplica al Software distribuido bajo una licencia que le permita al usuario acceso al código fuente del Software, y además le permita estudiar y modificarlo con toda libertad.

**P**

*POSTGRE SQL:* es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente.

*PGAMIN:* es una aplicación gráfica para gestionar el gestor de bases de datos [PostgreSQL](http://www.guia-ubuntu.com/index.php/PostgreSQL), siendo la más completa y popular con licencia Open Source.

**S**

*SERVLETS:* son módulos escritos en Java que se utilizan en un servidor, que puede ser o no ser servidor web, para extender sus capacidades de respuesta a los clientes al utilizar las potencialidades de Java.

*SERVIDOR:*Es un ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras.

*SOFTWARE:* Son aplicaciones que ayudan a crear otras aplicaciones y/o son aplicaciones finales para ser usadas.

**W**

*WINDOWS:* es un [**sistema operativo**](http://definicion.de/sistema-operativo/), es decir, un conjunto de programas que posibilita la administración de los recursos de una [**computadora**](http://definicion.de/computadora).

**Web:** consiste en un documento electrónico que contiene información, cuyo formato se adapta para estar insertado en la World Wide Web, de manera que los usuarios a nivel mundial puedan entrar a la misma por medio del uso de un navegador, visualizándola con un dispositivo móvil como un smartphone o un monitor de computadora.