

# UNIVERSIDAD PÚBLICA DE EL ALTO

## CARRERA INGENIERÍA DE SISTEMAS



### PROYECTO DE GRADO

#### SISTEMAS DE INFORMACIÓN GEOGRÁFICAS APLICADO AL MONITOREO DE RIESGOS DE QUEMA CON IMÁGENES DE SATÉLITE

CASO: (AUTORIDAD DE FISCALIZACIÓN Y CONTROL SOCIAL DE BOSQUES Y  
TIERRA - ABT)

Para optar el título de Licenciatura en Ingeniería de Sistemas

**MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

**Postulante:** Katteryn Flavia Huanca Villanueva

**Tutor Metodológico:** Ing. Marisol Arguedas Balladares

**Tutor Especialista:** Ing. Elias Carlos Hidalgo Mamani

**Tutor Revisor:** Lic. Norman Gudy Cardeña Pinto

EL ALTO – BOLIVIA

2020

## **DEDICATORIA**

Dedicado especialmente con todo cariño a:

Mis padres Saúl Domingo Huanca Huanca y Flavia Villanueva de Huanca.

Por su apoyo permanente, atenciones, aliento moral,

Material durante toda mi formación profesional,

Gracias Papá y Mamá.

## **AGRADECIMIENTOS A:**

Deseo expresar mis sinceros agradecimientos a:

Dios que me dio la oportunidad de llegar a este punto. Porque es la fuerza que me da la vida. Porque no me abandonó en las pruebas duras de mi vida y que espero no me abandone en el porvenir.

Mis padres Por el sacrificio que hicieron por mí, no sólo económicamente, sino por todo el apoyo moral y la confianza que depositaron en mí. Quiero decirles que no tengo palabras para agradecerles, simplemente los quiero mucho, siempre serán los padres que amo.

A la Universidad Pública de El Alto, Carrera Ingeniería de Sistemas por acogerme como universitaria de esta prestigiosa casa superior de estudios.

A mis tutores M.Sc. Ing. Enrique Flores Baltazar, Ing. Elias Carlos Hidalgo Mamani, Lic. Norman Gudy Cardeña Pinto por su asesoramiento, observaciones y consejos durante el transcurso de la elaboración del documento.

## **RESUMEN**

El presente proyecto de grado se inicia con la recopilación de toda la información posible sobre los Sistemas de Información Geográfica “SIG”, la cual detalla todos los conceptos básicos necesarios para la interpretación del sistema y poder aplicarlos en muchos campos de la vida real. Se profundiza más en los sistemas de información geográfica, pero aplicado al monitoreo de riesgos de quema con imágenes de satélite. Se obtuvo información sobre las áreas de riesgos de quema en Bolivia, en la institución Autoridad y fiscalización social de control de bosques y tierras – ABT. Los objetivos fueron: Realizar el análisis y requerimiento de la institución; Módulo de análisis sobre el grado de riesgo de quema; Implementar módulo de reportes; Implementar un visor geográfico para el monitoreo de áreas de quema y riesgo de quema; Generar estadísticas de riesgos de quema; Analizar y diseñar el sistema en base a la información (plana, espacial).

Para lograr los objetivos nombrados, se utilizó software libre, base de datos espacial y herramientas SIG, para el desarrollo de este proyecto se basa en la Metodología UWE, para la evaluación de calidad del software se utilizó WEB - QEM y para la estimación de costos se utilizó COCOMO.

**Palabras Clave: SIG, Riesgo, Geovisor, Monitoreo, QGIS, UWE.**

## **ABSTRACT**

The present project of degree begins with the compilation of all the possible information on the Geographic Information Systems "GIS", which details all the basic concepts necessary for the interpretation of the system and to be able to apply them in many fields of real life. Geographical information systems are further explored, but applied to monitoring burning risks with satellite images. Information was obtained on the areas of risk of burning in Bolivia, at the institution Authority and social control of forest and land control - ABT. The objectives were: Carry out the analysis and requirement of the institution; Analysis module on the degree of risk of burning; Implement reporting module; Implement a geographic viewer for monitoring burning areas and risk of burning; Generate burn risk statistics; Analyze and design the system based on information (flat, spatial).

To achieve the named objectives, free software, spatial database and GIS tools were used, for the development of this project it is based on the UWE Methodology, for the quality evaluation of the software WEB - QEM was used and for the cost estimation it was used COCOMO.

**Key Words: GIS, Risk, Geovisor, Monitoring, QGIS, UWE.**

# INDICE GENERAL

## CAPÍTULO

### MARCO PRELIMINAR

1.1.INTRODUCCIÓN.....	1
1.2.ANTECEDENTES.....	2
1.2.1.ANTECEDENTES DE LA INSTITUCIÓN.....	2
1.3.PLANTEAMIENTO DEL PROBLEMA.....	4
1.3.1. <i>Problema principal</i> .....	4
1.3.2. <i>Problemas secundarios</i> .....	4
1.4.OBJETIVOS.....	5
1.4.1. <i>Objetivo general</i> .....	5
1.4.2. <i>Objetivos específicos</i> .....	5
1.5.JUSTIFICACIONES.....	5
1.5.1. <i>Justificación Técnica</i> .....	5
1.5.2. <i>Justificación Económica</i> .....	6
1.5.3. <i>Justificación Social</i> .....	6
1.6.METODOLOGÍAS.....	7
1.6.1. <i>Metodología UWE</i> .....	7
1.6.2. <i>Métrica de calidad</i> .....	9
1.6.3. <i>Metodología WEB - QEM</i> .....	10
1.6.4. <i>Modelo de Costos COCOMO II</i> .....	12
1.7.....	HERRAMIENTAS
.....	16
1.7.1. <i>Lenguajes de Programación</i> .....	16

1.7.2.	<i>Sistemas de información geográficas</i> .....	17
1.7.3.	<i>Librería de desarrollo espacial</i> .....	18
1.7.4.	<i>Imagen de satélite</i> .....	20
1.7.5.	<i>Base de Datos</i> .....	21
1.7.6.	<i>Servidores</i> .....	22
1.8.	LÍMITES Y ALCANCES .....	22
1.8.1.	<i>Limites</i> .....	22
1.8.2.	<i>Alcances</i> .....	22
1.9.	APORTES .....	23

## **CAPÍTULO II**

### **MARCO TEÓRICO**

2.1.	INTRODUCCIÓN.....	24
2.2.	DATO .....	24
2.3.	SISTEMA .....	25
2.4.	INFORMACIÓN.....	26
2.5.	GEOGRAFÍA.....	28
2.6.	ANÁLISIS .....	28
2.7.	MONITOREO .....	29
2.7.1.	<i>Tipos de monitoreo</i> .....	30
2.8.	RIESGO DE QUEMA .....	32
2.8.1.	<i>Tipos de riesgos de quema</i> .....	32
2.9.	IMAGEN SATELITAL.....	32
2.9.1.	<i>Tipos de imágenes de satélite</i> .....	33
2.10.	SISTEMAS DE INFORMACIÓN GEOGRÁFICA .....	35

2.10.1. Representaciones de sistemas de información geográfica .....	36
2.11.MÓDULO .....	41
2.11.1. Tipos de módulos .....	42
2.12.VISOR GEOGRÁFICO .....	43
2.13.INFRAESTRUCTURA DE DATOS ESPACIALES .....	43
2.14.INGENIERÍA DE SOFTWARE .....	45
2.14.1. Módulos del desarrollo de software .....	46
2.14.2. Arquitectura del software .....	50
2.14.2.1. Descomposición Modular .....	51
2.14.2.2. Cliente – Servidor .....	52
2.14.2.3.PROGRAMACIÓN POR CAPAS.....	53
2.14.2.4.SEGURIDAD .....	55
2.14.2.5.CALIDAD.....	56
2.14.3. Metodología UWE .....	57
2.14.4. Métricas de calidad al software ISO 9126 .....	60
2.14.4.1.METODOLOGÍA WEB - QEM .....	61
2.14.4.2.Fases de WEB - QEM.....	61
2.14.4.2.1.ANÁLISIS DE RESULTADOS Y RECOMENDACIONES.....	64
2.14.4.3.CARACTERÍSTICAS DE WEB - QEM.....	65
2.14.4.3.1.FUNCIONALIDAD .....	65
2.14.4.3.2.CONFIABILIDAD.....	66
2.14.4.3.3.USABILIDAD.....	67
2.14.4.3.4.MANTENIBILIDAD.....	68

2.14.4.3.5.PORTABILIDAD .....	69
2.15.MODELO DE ESTIMACIÓN DE COSTES.....	69
2.15.1.  COCOMO II .....	69
2.16.HERRAMIENTAS.....	74
2.16.1.  PHP.....	74
2.16.2.LARAVEL .....	76
2.16.3.  QGIS .....	76
2.16.4.  Open Layer .....	77
2.16.5.  Leaflet .....	78
2.16.6.  Google Earth Engine .....	79
2.16.7.  PostgreSQL.....	80
2.16.7.1. Ideas Básicas Acerca del Funcionamiento.....	80
2.16.8.  PostGIS.....	81
2.16.9.  Laragon.....	82

## **CAPÍTULO III**

### **MARCO APLICATIVO**

3.1.INTRODUCCIÓN.....	83
3.2.ESTRUCTURA O ESQUEMA DEL SISTEMA .....	83
3.2.1. <i>Funcionalidad del sistema</i> .....	84
3.3.DESARROLLO DEL SISTEMA EN BASE A LA METODOLOGÍA UWE.....	84
3.3.1. <i>Análisis de requerimiento</i> .....	85
3.3.2. <i>Modelo de Casos de Uso</i> .....	86
3.3.2.1. Descripción de los actores .....	86
3.3.3. <i>Modelo Conceptual (Base de Datos)</i> .....	96

3.3.4.	<i>Modelo de Navegación</i> .....	98
3.3.5.	<i>Modelo de Presentación</i> .....	98
3.3.6.	CODIFICACIÓN DEL SOFTWARE .....	99
3.4.	IMPLEMENTACIÓN DEL SISTEMA .....	103
3.4.1.	<i>Implementación de los Geoservicios</i> .....	103
3.4.2.	<i>Recolección de datos</i> .....	106
3.4.3.	<i>Calcular el área de riesgo de quema</i> .....	107
3.5.2.	<i>Conexión a la Base de Datos Espacial en QGIS</i> .....	115
3.4.4.	<i>Importar archivos Shapes del cálculo de nivel de riesgo de quema</i> .....	117
3.4.5.	<i>Automatización de la Estadística Zonal</i> .....	120
3.5.	.....APLICACIÓN DE LA MÉTRICA DE CALIDAD WEB - QEM ISO 9126	
	.....	122
3.5.1.	<i>Funcionalidad</i> .....	122
3.5.2.	<i>Confiabilidad</i> .....	125
3.5.3.	<i>Usabilidad</i> .....	126
3.5.4.	<i>Mantenibilidad</i> .....	128
3.5.5.	<i>Portabilidad</i> .....	129
3.5.6.	<i>Calidad Global</i> .....	130
3.6.	APLICACIÓN DEL MODELO DE ESTIMACIÓN DE COSTOS COCOMO II .....	131

## **CAPÍTULO IV**

### **PRUEBAS Y RESULTADOS**

4.1.	PRUEBAS Y RESULTADOS .....	132
4.1.1.	<i>Actualización de datos</i> .....	132
4.1.2.	<i>Duplicidad de información</i> .....	132

4.1.3. <i>Visor de mapas</i> .....	132
4.1.4. <i>Estadísticas y Reportes</i> .....	132
4.2. PRUEBAS FUNCIONALES DEL SISTEMA.....	132

## **CAPÍTULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

5.1. CONCLUSIONES.....	138
5.2. RECOMENDACIONES.....	138
<b>Bibliografía</b> .....	<b>141</b>

## **ANEXOS**

ANEXO A. ÁRBOL DE PROBLEMAS

ANEXO B. ÁRBOL DE OBJETIVOS

ANEXO C. MANUAL DE USUARIO

ANEXO D. AVALES

## INDICE DE FIGURAS

Figura 2.1. Entidades de punto.....	37
Figura 2.2. Entidades de líneas .....	38
Figura 2.3. Entidades de polígonos .....	38
Figura 2.4. Tablas relacionadas.....	40
Figura 2.5. Tres presentaciones fundamentales de capas de información geográfica .....	41
Figura 2.6. Infraestructura de datos espaciales .....	44
Figura 2.7. Modelo Cascada .....	47
Figura 2.8. Modelo Evolutivo .....	47
Figura 2.9. Modelo Espiral.....	49
Figura 2.10. Modelo Concurrente .....	50
Figura 2.11. Cliente – Servidor .....	53
Figura 2.12. Programación por capas.....	54
Figura 2.13. Características WEB - QEM Usabilidad .....	67
Figura 3.1. Contenido del sistema.....	83
Figura 3.2. Esquema del sistema.....	84
Figura 3.3. Análisis sobre el grado de riesgo de quema.....	87
Figura 3.4. Análisis sobre el grado de riesgo de quema.....	88
Figura 3.5. Importación de la información geográfica a la Base de Datos Espacial.....	89

Figura 3.6. Registro de Tablas en el Sistema .....	90
Figura 3.7. Visualización del visor geográfico con imágenes de satélite .....	92
Figura 3.8. Generar reportes estadísticos sobre el riesgo de quema en el visor de mapa .....	93
Figura 3.9. Generar reportes estadísticos sobre el riesgo de quema .....	95
Figura 3.10. Modelo conceptual .....	97
Figura 3.11. Modelo de navegación.....	98
Figura 3.12. Modelo de presentación.....	99
Figura 3.13. Ventana Principal.....	100
Figura 3.14. Box de conceptos.....	101
Figura 3.15. Ventana del Geovisor .....	101
Figura 3.16. Ventana de Estadísticas y Reportes .....	102
Figura 3.17. Ventana de Servicios .....	102
Figura 3.18. Ventana de Ayuda .....	103
Figura 3.19. Laragon instalado .....	104
Figura 3.20. Composer.....	105
Figura 3.21. Creación de Base de Datos Espacial en PostgreSQL .....	106
Figura 3.22. Conexión a la Base de Datos .....	106
Figura 3.23. Calculo de número de pixeles.....	108
Figura 3.24. Resultado de Histograma Zonal.....	108

Figura 3.25. Sumatoria de campos.....	109
Figura 3.26. Resultado de sumatoria de campos.....	109
Figura 3.27. Calculo del nivel de riesgo muy bajo .....	110
Figura 3.28. Resultado del cálculo de nivel de riesgo muy bajo.....	110
Figura 3.29. Cálculo de nivel de riesgo bajo.....	111
Figura 3.30. Resultado del cálculo de nivel de riesgo bajo.....	111
Figura 3.31. Calculo de nivel de riesgo medio .....	112
Figura 3.32. Resultado del cálculo de nivel de riesgo medio.....	112
Figura 3.33. Calculo de nivel de riesgo alto .....	113
Figura 3.34. Resultado del cálculo de riesgo alto .....	113
Figura 3.35. Calculo de nivel de riesgo muy alto .....	114
Figura 3.36. Resultado del cálculo de nivel de riesgo muy alto .....	114
Figura 3.37. Cálculo de nivel de riesgo extremo .....	115
Figura 3.38. Resultado del cálculo de nivel de riesgo extremo .....	115
Figura 3.39. Conexión de la Base de Datos Espacial en.....	117
Figura 3.40. Administrador de Base de Datos .....	118
Figura 3.41. Importación de archivos Shapes a la Base de datos .....	119
Figura 3.42. Tablas importadas de QGIS.....	119
Figura 3.43. Tablas importadas con datos correctos .....	120

Figura 3.44. Unión de tablas .....	120
Figura 3.45. Script Fecha .....	121
Figura 3.46. Script Conteo .....	121
Figura 3.47. Valores de coste COCOMO II.....	<b>¡Error! Marcador no definido.</b>

## INDICE DE TABLAS

Tabla 1.1.....	14
<i>Ecuaciones del Modelo Básico</i> .....	14
Tabla 2.1.....	32
<i>Tipos de riesgos de quema</i> .....	32
Tabla 2.2.....	64
<i>Características de calidad</i> .....	64
Tabla 2.3.....	71
<i>Calcular distintos aspectos de costes</i> .....	71
Tabla 2.4.....	72
<i>Modelo intermedio</i> .....	72
Tabla 3.1.....	85
<i>Requerimientos Funcionales</i> .....	85
Tabla 3.2.....	87
<i>Descripción de Actores del sistema</i> .....	87
Tabla 3.3.....	88
<i>Descripción de Caso de uso: Análisis sobre el grado de riesgo de quema</i> .....	88
Tabla 3.4.....	90
<i>Descripción de Caso de Uso: Importación de la información geográfica a la Base de Datos Espacial</i> .....	90

Tabla 3.5.....	91
<i>Descripción de Caso de Uso: Inicio de Sesión .....</i>	91
Tabla 3.6.....	92
<i>Descripción de Caso de Uso: Visualización del visor geográfico con imágenes de satélite ..</i>	92
Tabla 3.7.....	94
Tabla 3.8.....	96
<i>Descripción de Caso de uso: Ingresar consulta .....</i>	96
Tabla 3.9.....	107
<i>Clasificación de información geográfica de tipo vectorial.....</i>	107
Tabla 3.10.....	123
<i>Calculo de Funcionalidad.....</i>	123
Tabla 3.11.....	124
<i>Tabla de ajuste de complejidad.....</i>	124
Tabla 3.12.....	128
<i>Encuesta sobre la usabilidad del sistema .....</i>	128
Tabla 3.13.....	130
<i>Calidad Global.....</i>	130
Tabla 3.14.....	132
<i>Conversión de puntos de función KLDC.....</i>	132

Tabla 3.15.....	133
<i>Coefficientes de a, b, c, d de COCOMO</i> .....	133
Tabla 3.16.....	135
<i>Costo de elaboración del proyecto</i> .....	135
Tabla 3.17.....	135
<i>Costo total del proyecto</i> .....	135
Tabla 4.1.....	133
<i>Prueba de Navegación</i> .....	133
Tabla 4.2.....	134
<i>Prueba de Visualización de visor de mapas</i> .....	134
Tabla 4.3.....	135
<i>Prueba Estadísticas y Reportes</i> .....	135
Tabla 4.5.....	136
<i>Prueba de Servicios</i> .....	136



**CAPÍTULO I**  
**MARCO PRELIMINAR**



## **1.1. Introducción**

Los sistemas de información geográfica desde sus inicios hasta ahora, son sistemas que ayudan a la población a resolver diferentes tipos de problemas, en la actualidad según la ABT 38% del área de toda Bolivia son propensos a incendios.

La mayor parte de la información que manejamos en cualquier tipo de disciplina está georeferenciada. Es decir, que se trata de información a la cual puede asignarse una posición geográfica, y es por tanto información que viene acompañada de otra información adicional relativa a su localización. SIG es tanto un sistema de base de datos con capacidades específicas para datos georreferenciados, como un conjunto de operaciones para trabajar con esos datos. En cierto modo, un SIG es un mapa de orden superior. (Olaya, ¿Que es un SIG?, 2014).

La propuesta de aplicar sistemas de información geográficos para el monitoreo de riesgos de quema con imágenes de satélite, se fundamenta en la importancia a el monitoreo que describe a un proceso mediante el cual se reúne, observa, estudia y emplea información para luego realizar un seguimiento de un programa o hecho particular, las imágenes de satélite se definen como la representación visual de la información capturada por un sensor montado en un satélite artificial. Estos sensores recogen la información reflejada por la superficie de la Tierra que luego es enviada de regreso a ésta y es procesada convenientemente.

Con la metodología UWE que es un proceso del desarrollo para aplicaciones web enfocadas sobre el diseño sistémico, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación web y mediante el uso de herramientas como ser: base de datos espaciales, servidor y lenguajes de programación ayudaran al desarrollo del sistema.

Por lo tanto, implementando el siguiente proyecto nos permite entregar una herramienta para el monitoreo de riesgos de quema con imágenes de satélite con la finalidad de poder realizar un mejor análisis de riesgo de incendio automatizado y tener información disponible para las autoridades y los usuarios.

## **1.2. Antecedentes**

### **1.2.1. Antecedentes de la Institución**

La Autoridad de Fiscalización y Control Social de Bosques y Tierra – ABT, se encarga de tener datos y registros de las áreas de quema y da licencias sobre las áreas de quema. La institución se creó por el supremo No. 071 de 09 de abril del 2009, (Artículo 3 Parágrafo I literal c) en sustitución de la extinta Superintendencia Forestal y Agraria, como institución pública técnica y operativa, con personalidad jurídica y patrimonio propio, independencia administrativa, financiera, legal y técnica, supeditadas al Ministerio cabeza de sector. De acuerdo al Artículo 27 asume el control, supervisión y regulación de los sectores Forestales y Agrario. El Decreto Supremo N° 0429 de 10 de febrero de 2010, determina que la Autoridad de Fiscalización y Control Social de Bosques y Tierra – ABT pase a tuición del Ministerio de Medio Ambiente y Agua.

### **Antecedentes Internacionales**

- [Amalia Vaneska Palacio Buendía, 2017] “IMPLEMENTACIÓN DE SISTEMAS DE INFORMACIÓN GEOGRÁFICA EN LA GESTIÓN DE ESPACIOS PROTEGIDOS”, Objetivo General: Analizar la planificación y gestión de los ENPs con énfasis en la gestión del paisaje, el uso público y actividades recreativas, a partir de Sistemas de Información Geográfica de Participación Pública (PPGIS), Metodología: PPSIG, Herramientas: Google Maps, Google

Forms, Editor Notepad++, un servicio de almacenamiento de información Dropbox.  
UNIVERSIDAD ROVIRA I VIRGILI, Xarxa.

- [Liliana Lisbeth Calixto Ramón, 2017] “DETERMINACIÓN DE ZONAS SUSCEPTIBLES DE INUNDACIÓN MEDIANTE EL USO DE HERRAMIENTAS SIG EN EL ÁREA DE INFLUENCIA DEL RIO CAVRO SUR, UBICADO EN EL MUNICIPIO DE YOPAL, DEPARTAMENTO DE CASANARE”, Objetivo General: Determinar las zonas susceptibles de inundación mediante el uso de herramientas SIG en el área de influencia del Río Cravo Sur, ubicado en el Municipio de Yopal, Departamento de Casanare, Metodología: Investigación Cuantitativa, Herramientas: Qgis, SIG, UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS, Bogotá.

#### **Antecedentes Nacionales**

- [Alejandro Pascual Quispe Capquique, 2014] “EVALUACIÓN MULTITEMPORAL DEL CAMBIO DE LA COBERTURA VEGETAL MEDIANTE EL USO DE TÉCNICAS DE TELEDETECCIÓN Y SIG EN LA COLONIA SIEMPRE UNIDOS DEL MUNICIPIO DE CARANAVI, LA PAZ”, Objetivo General: Evaluar los cambios de la Cobertura Vegetal producidas en la Colonia Siempre Unidos del municipio de Caranavi y la influencia del mismo en su población, Metodología: investigación, obtención de imágenes, digitalización del área de estudio, Herramientas: imágenes Landsat, imágenes satelitales, ArcGIS,, IMÁGENES RASTER, UNIVERSIDAD MAYOR DE SAN ANDRES, Ingeniería Agronómica, La Paz-Bolivia.

- [Eslid Ana Guerra Cerezo, 2006] “ANÁLISIS MULTITEMPORAL DE LA COBERTURA Y USO DE LA TIERRA A TRAVÉS DE SIG EN LA CUENCA BAJA DEL RÍO GRANDE-SANTA CRUZ”, Objetivo General: Realizas un análisis multitemporal de la cobertura de la tierra en el periodo 1986-2005 utilizando el sistema LCCS en la cuenca del Rio

Grande Zona Norte (área de expansión agrícola), Metodología: Manejo y preparación de la geo-información Herramientas: Imagen Landsat, ArcGIS, ArcView, GeoVis, Ilwis Academic, Erdas y Mapas topográficos, UNIVERSIDAD MAYOR DE SAN SIMON, Cochabamba-Bolivia.

### **Antecedentes Locales**

- [Raquel Cristina Medina Sosa, Erick Aguilar Ramirez,2016] “SISTEMA DE INFORMACION TERRITORIAL – GEOVIACHA, MUNICIPIO DE VIACHA”, Objetivo General: Diseñar, desarrollar e implementar una IDE que permita facilitar el acceso y la integración de la información espacial, tanto a nivel institucional y empresarial como de los propios ciudadanos, lo que permitirá el uso de la información Geográfica y coadyuvara en la toma de decisiones, Herramientas: Web Servlet Tomcat, PostGIS, MapFish, Ubustu, PostgreSQL, Geoserver, Qgis, IDEE, IDERA, UNIVERSIDAD PÚBLICA DE EL ALTO, La Paz-Bolivia.

## **1.3. Planteamiento del Problema**

### **1.3.1. Problema principal**

Actualmente la Autoridad de Fiscalización y Control Social de Bosques y Tierra – ABT, hace un control manual de las áreas de quema, lugares que son propensos a incendiarse, por la cual todos los reportes, registros, análisis del riesgo de quema y la actualización de datos es lenta y esto ocasiona un mal manejo de información.

### **1.3.2. Problemas secundarios**

- No se cuenta con el análisis de riesgo de quema automatizado.
- No se tiene actualizaciones en tiempo real sobre las áreas de quema.
- Existe duplicidad de información.
- No se cuenta con un visor de mapas.

- No existen reportes de las áreas de quema.

## **Formulación del problema**

¿Cómo permitiría el sistema de información geográfica coadyuvar al monitoreo con imágenes de satélite en temas de quema?

## **1.4. Objetivos**

### **1.4.1. Objetivo general**

Desarrollar un Sistema de Información Geográfica para el monitoreo sobre riesgos de quema mediante imágenes satelitales.

### **1.4.2. Objetivos específicos**

- Realizar el análisis de quema y requerimientos consecuentes para la institución de Autoridad y Fiscalización de Control Social de Bosques y Tierras - ABT.
- Diseñar el sistema de información de prevención en base a los datos proporcionados, de naturalezas plana y espacial.
- Plantear un módulo de análisis sobre el grado: riesgo de quema.
- Implementar módulo de información referida a la prevención de fenómenos de quema.
- Generar información de riesgos de quema, mediante la estadística zonal.
- Efectuar un visor geográfico para el monitoreo de áreas de riesgo de quema.

## **1.5. Justificaciones**

### **1.5.1. Justificación Técnica**

La realización del presente proyecto se sustenta en el continuo avance tecnológico, que mejorara los métodos de acceso, manipulación y seguridad de los datos de información, utilizando ingeniería de software, metodología UWE, tecnologías de última generación y

software libre como es el caso de la institución Autoridad de Fiscalización y Control Social de Bosques y Tierra – ABT, el cual necesita de este sistema para el logro de objetivos.

Además, la institución cuenta con los equipos necesarios para manipular el sistema; dentro y fuera de la institución.

### **1.5.2. Justificación Económica**

El presente proyecto se justifica económicamente por el ahorro de material de escritorio, ya que la manipulación de información es manual, se verá reducida gracias a la automatización de la misma. Además, de que contara con un software libre, la cual no tiene un costo alguno, también contará con información relevante, organizada, estructurada y confiable para la institución y los usuarios.

El diseño e implementación del SIG y la actualización de la información permitirán el ahorro de esfuerzos tanto económicos como humanos, ya que al contar con esta herramienta la dependencia invertirá menos tiempo en la ejecución de sus procesos, ahorrando así parte del presupuesto destinado para este tipo de proyectos.

### **1.5.3. Justificación Social**

La implementación del presente proyecto ofrece grandes beneficios a la institución Autoridad de Fiscalización y Control Social de Bosques y Tierra – ABT, como es el de brindar un mejor control del área de riesgos de quema, además tendrá un beneficio con información actualizada y centralizada para los usuarios.

Se evita la gran acumulación de papeles de registros de datos, con la ayuda de un módulo de registro de entrada, coadyuvara de gran manera al registro y el tiempo que llevaba hacerlo de forma manual.

Los beneficiados del sistema serán:

- La institución, ya que tendrán un mejor control y organización de las áreas de riesgo de quema.
- Los usuarios, la cual tendrán acceso a visualizar y obtener información sobre las áreas de riesgo de quema.

## **1.6. Metodologías**

### **1.6.1. Metodología UWE**

UWE (Koch & Kraus, 2003) es una metodología con un modelado de proceso espiral / interactivo – incremental y tomando como lenguaje de notación UML, esta metodología está enfocada en el diseño sistémico, la personalización y la generación semiautomática de escenarios. UWE utiliza vistas especiales soportadas por los diagramas gráficos de UML, como el modelo de navegación y el modelo de presentación. UWE no limita el número de vistas de una aplicación, ya que los diseñadores también pueden hacer uso de otra técnica de modelado UML para agregar otras vistas a la aplicación.

Es una herramienta que nos permite modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización (sistemas adaptativos). UWE es una propuesta basada en el proceso unificado y UML, pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

UWE propone los siguientes pasos para el desarrollo de una aplicación web:

- Especificación de requerimientos
- Modelo de Lógico-Conceptual
- Modelo de Navegación
- Modelo de presentación

## **Fases de la UWE.**

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

Las fases o etapas a utilizar son:

### **1) Captura, análisis y especificación de requisitos**

En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales.

Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

### **2) Diseño del sistema**

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define el cumplimiento de los requisitos, la estructura que debe darse a la aplicación web.

### **3) Codificación del software**

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

### **4) Pruebas**

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

## **5) La Instalación o Fase de Implementación**

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

## **6) El Mantenimiento**

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control. (Vilariño, 2010)

### **1.6.2. Métrica de calidad**

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software, a los requerimientos implícitos y explícitos del cliente.

### **Modelo de Calidad Establecido por el estándar ISO 9126**

La ISO, bajo la norma ISO-9126, ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 con el nombre de “Information technology –Software product evaluation: Quality characteristics and guidelines for their use”, en el cual se establecen las características de calidad para productos de software.

El estándar ISO-9126 [Sanders, Joc & Eugene Curran. Software Quality. A Framework for Success in Software Development and Support, Addison Wesley] establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia,

mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de subcaracterísticas que permiten profundizar en la evaluación de la calidad de productos de software. (Calidad en la Industria del Software La Norma ISO-9126)

### **1.6.3. Metodología WEB - QEM**

Se utiliza esta metodología para medir la calidad del sistema.

#### **Fases de WEB - QEM**

A continuación, se describe las fases de la metodología WEB - QEM que se utilizará para realizar los cálculos que ayudara a comprobar que el sistema es confiable, eficiente y de calidad.

#### **A. Definición de Las Metas de Evaluación y Selección del Perfil de Usuario**

En esta fase se consideran dos pasos primordiales y comunes en toda evaluación de calidad siguiendo la metodología WEB – QEM.

##### **i) Metas de Evaluación**

En esta fase se define las metas de evaluación y selección del perfil de usuario, los evaluadores deben definir las metas y establecer el alcance del proyecto.

Las metas a llegar a cumplir con la evaluación del sistema son:

- Conocer la percepción de la calidad que tienen los usuarios finales sobre el sistema.
- Lograr que la calidad del sistema esté por encima de las expectativas del usuario final.

##### **ii) Selección de Perfil de Usuario**

Para el diseño e implementación de un caso de estudio de evaluación de calidad en uso, una meta muy utilizada es determinar el cumplimiento de requerimientos elementales, parciales y globales de calidad para una aplicación Web operativa, considerando el perfil de encargado de requerimiento de pedidos y compras.

Los encargados de realizar los pedidos y compras son los usuarios que interactuaran con el sistema de forma recurrente y a diario estos usuarios conocen la lógica del negocio y tuvieron capacitación sobre el manejo del sistema.

### **B. Definición de los Requerimientos de calidad y/o Costo**

En esta fase, teniendo en cuenta los aspectos definidos en la fase anterior respecto a metas de evaluación, selección de perfil de usuario, se deben establecer atributos y subconceptos (características) de calidad cuantificables que, agrupados jerárquicamente, representen un modelo de calidad apropiado para el perfil de usuario seleccionado.

### **C. Definición de Criterios Elementales y Procedimientos de Medición**

En esta etapa se define una base de criterios para evaluación elemental, y realizar el proceso de medición y puntuación de los valores que se calcule para medir la calidad del software.

### **D. Definición de Estructuras de Agregación e Implementación de la Evaluación Global**

Para esta fase se realiza un promedio de todos los parámetros encontrados para así poder sacar un resultado global y así el usuario pueda comprender de mejor manera la calidad del sistema.

### **Análisis de Resultados y Recomendaciones**

En esta fase una vez calculado las características de calidad se darán las recomendaciones respectivas además de saber los resultados y ver si el sistema cumple con lo pronosticado.

### **Características de WEB - QEM**

La metodología WEB - QEM toma las métricas del modelo de calidad ISO 9126-1 la cual da referencia a las siguientes características:

- **Funcionalidad**
- **Confiabilidad**
- **Usabilidad**

- **Mantenibilidad**
- **Portabilidad.**

#### **1.6.4. Modelo de Costos COCOMO II**

El Modelo Constructivo de Costos (o COCOMO, por su acrónimo del inglés COConstructive COst MOdel) es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981).

#### **Características generales**

Pertenece a la categoría de modelos estimadores basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en función de la cantidad de líneas de código, principalmente.

Se presentan tres niveles: básico, intermedio y detallado.

#### **Modelos de estimación**

Las ecuaciones que se utilizan en los tres modelos son:

En Persona – Mes:

$$\mathbf{A. \quad E = a(KI^{(b)*m(X)})}$$

En Meses:

$$\mathbf{B. \quad Tdev = c(E^d)}$$

En Personas:

$$C. P = \frac{E}{T_{dev}}$$

Dónde:

**E:** es el esfuerzo requerido por el proyecto, en persona-mes

**T<sub>dev</sub>:** es el tiempo requerido por el proyecto, en meses

**P:** es el número de personas requerido por el proyecto

**a, b, c y d** son constantes con valores definidos en una tabla, según cada submodelo

**Kl:** es la cantidad de líneas de código, en miles.

**m(X):** Es un multiplicador que depende de 15 atributos.

A la vez, cada submodelo también se divide en modos que representan el tipo de proyecto, y puede ser:

- a. **Modo orgánico:** un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).
- b. **Modo semilibre o semiencajado:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- c. **Modo rígido o empotrado:** el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

### **Modelo básico**

Se utiliza para obtener una primera aproximación rápida del esfuerzo, y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

**Tabla 1.1**

***Ecuaciones del Modelo Básico***

<b>MODO</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>Orgánico</b>	2.40	1.05	2.50	0.38
<b>Semi – Orgánico</b>	3.00	1.12	2.50	0.35
<b>Empotrado</b>	3.60	1.20	2.50	0.33

**Nota.** Ecuaciones del Modelo Básico. (Boehm 1981)

Estos valores son para las fórmulas:

Personas necesarias por mes para llevar adelante el proyecto:

$$MM = a * (KI^b)$$

Tiempo de desarrollo del proyecto:

$$TDEV = c * (MM^d)$$

Personas necesarias para realizar el proyecto:

$$CosteH = \frac{MM}{TDEV}$$

Costo total del proyecto:

$$CosteM = CosteH * \text{Salario medio entre los programadores y analistas}$$

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno

**Modelo intermedio**

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.

### **Atributos**

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy bajo - bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula (por ejemplo, si para un proyecto el atributo DATA es calificado como muy alto, el resultado de la fórmula debe ser multiplicado por 1000).

El significado de los atributos es el siguiente, según su tipo:

- **De software**

**RELY:** garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (muy bajo) hasta la posible pérdida de vidas humanas (extremadamente alto, software de alta criticidad).

**DATA:** tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación:  $D/K$ , donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.

**CPLX:** representa la complejidad del producto.

- **De hardware**

**TIME:** limitaciones en el porcentaje del uso de la CPU.

**STOR:** limitaciones en el porcentaje del uso de la memoria.

**VIRT:** volatilidad de la máquina virtual.

**TURN:** tiempo de respuesta requerido.

- **De personal**

**ACAP:** calificación de los analistas.

**AEXP:** experiencia del personal en aplicaciones similares.

**PCAP:** calificación de los programadores.

**VEXP:** experiencia del personal en la máquina virtual.

**LEXP:** experiencia en el lenguaje de programación a usar.

- **De proyecto**

**MODP:** uso de prácticas modernas de programación.

**TOOL:** uso de herramientas de desarrollo de software.

**SCED:** limitaciones en el cumplimiento de la planificación.

## **1.7. Herramientas**

### **1.7.1. Lenguajes de Programación**

- **PHP:** acrónimo recursivo en inglés de PHP, Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el preprocesado de texto plano en UTF-8. Posteriormente se aplicó al desarrollo web de contenido dinámico, dando un paso evolutivo en el concepto de aplicación en línea, por su carácter de servicio.

Su implementación en los documentos HTML era aparentemente muy sencilla. Hay que decir, que, PHP no genera HTML, sino que ofrece una salida de texto con codificación UTF-8 compatible con los documentos HTML. El programador puede dotar a la salida de los tag's propios del HTML y los exploradores más comunes para navegar por internet, reconocerán muy rápidamente el formato UTF-8 y lo adaptarán ofreciendo una salida entendible.

Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el texto plano en formato UTF-8, ampliamente reconocido por el estándar HTML, dando como resultado, en los exploradores, una salida al usuario perfectamente entendible.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP.<sup>2</sup> Este lenguaje forma parte del software libre publicado bajo la licencia PHPv3\_01, una licencia Open Source validada por Open Source Initiative. La licencia de PHP es del estilo de licencias BSD, sin la condición de copyleft asociada con la Licencia Pública General de GNU. (PHP, 2020)

### **1.7.2. Sistemas de información geográficas**

- **QGIS:** es un Sistema de Información Geográfica (SIG) de software libre para plataformas GNU/Linux, Unix, Mac OS, Microsoft Windows y Android.<sup>2</sup> Era uno de los primeros ocho proyectos de la Fundación OSGeo y en 2008 oficialmente graduó de la fase de incubación. Permite manejar formatos Ráster y Vectoriales a través de la biblioteca GDAL (GADL/OGR), así como bases de datos. Algunas de sus características son:
  - ✓ Soporte para la extensión espacial de PostgreSQL, PostGIS.
  - ✓ Manejo de archivos vectoriales Shapefile, ArcInfo coverages, MapInfo, GRASS GIS, DXF, DWG, etc.

- ✓ Soporte para un importante número de tipos de archivos Ráster (GRASS GIS, GeoTIFF, TIFF, JPG, etc.)

Una de sus mayores ventajas es la posibilidad de usar Quantum GIS como GUI del SIG GRASS, utilizando toda la potencia de análisis de este último en un entorno de trabajo más amigable. QGIS está desarrollado en C++, usando la biblioteca Qt para su Interfaz gráfica de usuario.

Una de las grandes fortalezas de QGIS es que trabaja en cualquiera de los sistemas operativos:

- GNU/Linux
- BSD
- Unix
- MacOS
- Windows
- Android (en fase experimental)
- Funcionando de manera similar en todos ellos.

## **Licencia**

QGIS es un software libre y opera bajo la licencia GNU GPL. El software QGIS puede ser modificado libremente de tal manera que pueda realizar diferentes y más especializadas funcionalidades. Ya existen dos nuevos productos denominados: QGIS Browser y QGIS Server. Estos productos poseen diferentes interfaces del usuario (front-end). (QGIS, 2020)

### **1.7.3. Librería de desarrollo espacial**

- **Open Layer:** es una biblioteca de JavaScript de código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores web. OpenLayers ofrece un API para acceder a diferentes fuentes de información cartográfica en la red: Web Map

Services, Mapas comerciales (tipo Google Maps, Bing, Yahoo), Web Features Services, distintos formatos vectoriales, mapas de OpenStreetMap, etc.

Inicialmente fue desarrollado por MetaCarta en junio de 2006. Desde el noviembre del 2007 este proyecto forma parte de los proyectos de Open Source Geospatial Foundation. Actualmente el desarrollo y el soporte corren a cargo de la comunidad de colaboradores.

OpenLayers facilita poner un mapa dinámico en cualquier página web. Puede mostrar mosaicos de mapas, datos vectoriales y marcadores cargados desde cualquier fuente. OpenLayers ha sido desarrollado para promover el uso de información geográfica de todo tipo. Es completamente gratuito, Open Source JavaScript, lanzado bajo la Licencia BSD de 2 cláusulas (también conocida como FreeBSD). (Open Layers, 2018)

- **Leaflet:** Es una biblioteca JavaScript de código abierto ampliamente utilizada para crear aplicaciones de mapeo web. Lanzado por primera vez en 2011, es compatible con la mayoría de las plataformas móviles y de escritorio, compatible con HTML5 y CSS3. Junto con OpenLayers y la API de Google Maps, es una de las bibliotecas de mapeo de JavaScript más populares y es utilizada por los principales sitios web como FourSquare , Pinterest y Flickr . Leaflet permite a los desarrolladores sin experiencia en SIG visualizar mapas web en mosaico alojados en un servidor público, con superposiciones en mosaico opcionales. Puede cargar datos de características de archivos GeoJSON , diseñarlos y crear capas interactivas, como marcadores con ventanas emergentes cuando se hace clic. Está desarrollado por Vladimir Agafonkin, quien se unió a Mapbox en 2013. (Agafonkin, 2019)

## Características

Leaflet admite capas del Servicio de mapas web (WMS), capas GeoJSON , capas vectoriales y capas de mosaico de forma nativa. Muchos otros tipos de capas son compatibles a través de complementos.

Al igual que otras bibliotecas de mapas web, el modelo de visualización básico implementado por Leaflet es un mapa base, más cero o más superposiciones translúcidas, con cero o más objetos vectoriales mostrados en la parte superior.

## Elementos

Los principales tipos de objetos de Leaflet son:

- Tipos de Ráster (TileLayer e ImageOverlay)
- Tipos de Vectores (ruta, polígono y tipos específicos como círculo)
- Tipos agrupados (LayerGroup, FeatureGroup y GeoJSON)
- Controles (Zoom, Capas, etc.)

También hay una variedad de clases de utilidad, como interfaces para gestionar proyecciones, transformaciones e interactuar con el DOM. (Morales, 2016)

### 1.7.4. Imagen de satélite

- **Google Earth Engine:** Una de las últimas aplicaciones lanzadas por la empresa Google es la denominada Google Earth Engine, la cual es una plataforma para el análisis científico a escala petabyte (PB) y la visualización de conjuntos de datos geoespaciales, tanto para el beneficio público como para los usuarios comerciales y de la Administración. La principal diferencia con la aplicación Google Earth es la capacidad de análisis de los datos.

Earth Engine almacena imágenes satelitales, las organiza y las pone a disposición por primera vez para la extracción de datos a escala global. El archivo público de datos incluye imágenes

históricas de la tierra que se remontan a más de cuarenta años, y se recopilan nuevas imágenes todos los días. Earth Engine también proporciona APIs en JavaScript y Python, así como otras herramientas, para permitir el análisis de grandes conjuntos de datos. (López, 2018)

### 1.7.5. Base de Datos

- **PostgreSQL y PostGIS:** PostGIS convierte al sistema de administración de bases de datos PostgreSQL en una base de datos espacial mediante la adición de tres características: tipos de datos espaciales, índices espaciales y funciones que operan sobre ellos. Debido a que está construido sobre PostgreSQL, PostGIS hereda automáticamente las características de las bases de datos empresariales, así como los estándares abiertos que implementan un Sistema de Información Geográfica dentro del motor de base de datos. (**PostgreSQL, 2020**)
- **PostGIS:** es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en Sistema de Información Geográfica. Se publica bajo la Licencia Pública General de GNU.

Postgis ha sido desarrollado por la empresa canadiense Refraction Research, especializada en productos "Open Source" entre los que habría que citar a Udig. PostGIS es hoy en día un producto veterano que ha demostrado versión a versión su eficiencia. En relación con otros productos, PostGIS ha demostrado ser muy superior a la extensión geográfica de la nueva versión de MySQL, y a juicio de muchos, es muy similar a la versión geográfica de la base de datos Oracle.

Un aspecto que debemos tener en cuenta es que PostGIS ha sido certificado en 2006 por el Open Geospatial Consortium (OGC) lo que garantiza la interoperabilidad con otros sistemas también interoperables. PostGIS almacena la información geográfica en una columna del tipo GEOMETRY, que es diferente del homónimo "GEOMETRY" utilizado por PostgreSQL, donde

se pueden almacenar la geometría en formato WKB (Well-Known Binary), aunque hasta la versión 1.0 se utilizaba la forma WKT (Well-Known Text). (PostGIS, 2020)

### **1.7.6. Servidores**

- **Laragon:** Laragon es una opción relativamente nueva para la creación de lo que llamamos el entorno de desarrollo, es decir, todo un conjunto de programas necesarios para desarrollar aplicaciones. Sirve para trabajar con PHP, pero también con otros lenguajes del lado del servidor, como Node, Python o Ruby.

Laragon posee una utilidad realmente interesante que permite crear automáticamente virtual hosts para cada proyecto, así por ejemplo en lugar de ingresar desde el navegador a un proyecto con localhost/tu-proyecto, puedes usar una url más legible como tu-proyecto.dev. (Jeff, 2019)

## **1.8. Límites y Alcances**

### **1.8.1. Límites**

El proyecto se implementa en la institución de Autoridad de Fiscalización y Control Social de Bosques y Tierras - ABT, como es un sistema web la información se visualiza por los usuarios externos. Muestra la información de todas las áreas de Bolivia que tienen riesgo o son propensos a quemarse.

### **1.8.2. Alcances**

El sistema propuesto incluirá los siguientes módulos:

- Módulo de registro
- Módulo de visor de datos
- Módulo de análisis
- Módulo de reportes

## **1.9. Aportes**

Generar una herramienta que nos sirva para monitorear los riesgos de áreas de quema, para obtener una información actualizada de los datos, las características de estas áreas podrían facilitar la gestión de los incendios para prevenirlos y llevar a cabo programas orientados a evitar mayores daños al medio ambiente.



**CAPÍTULO II**  
**MARCO TEÓRICO**



## **2.1. Introducción**

El presente proyecto de grado se inicia con la recopilación de toda la información posible sobre los Sistemas de Información Geográfica SIG, la cual detalla todos los conceptos básicos necesarios para la interpretación del sistema y poder aplicarlos en muchos campos de la vida real. (Calixto, 2017) afirma: “Los SIG son generalmente un conjunto de herramientas que nos permiten obtener, almacenar, visualizar, y procesar datos espacialmente del mundo real”. (pág. 14). Se introducen conceptos geográficos y de computación utilizados en este tipo de sistemas.

Con el avance de la tecnología, han surgido herramientas SIG que se pueden utilizar como ayuda para el desarrollo de estrategias tanto de prevención como de mitigación en las pérdidas materiales y humanas frente a una amenaza natural.

En el caso de la prevención frente a la probabilidad de un evento sísmico ha surgido una tecnología relativamente nueva, que puede y de hecho es utilizado en muchos países para la prevención y/o mitigación de daños frente a un evento sísmico. (España, 2007, pág. 23)

## **2.2. Dato**

Dato se refiere a números o palabras, que permiten estudiarlos, analizarlos y conocerlos, también es una representación simbólica de una variable. En el ámbito informático un dato es un valor que recibe un computador por distintos medios, la cual representan la información que el programador manipula.

El dato no tiene valor semántico se refiere a que no tiene sentido en sí mismo, pero convenientemente procesado, se puede utilizar en la realización de cálculos o toma de decisiones. Un dato es una representación simbólica numérica, alfabética, etc. de una característica de una entidad. Es de empleo muy común en el ámbito informático.

Los datos son la materia prima de la cual se deriva la información. Cualquier cantidad o hecho, sin analizar, que por sí solos no tienen significado alguno y deben ser presentados en forma utilizable y colocados en un contexto que le de valor. Ejemplos: Edad, número de artículos vendidos, sueldo, etc. (Patricia, 2019)

En el ámbito de las ciencias de la información y la bibliotecología, ejemplos de datos son la altura de una montaña, la fecha de nacimiento de un personaje histórico, el peso específico de una sustancia, el número de habitantes de un país, etc.

### **2.3. Sistema**

Es un conjunto de normas y procedimientos que ayudan al funcionamiento de un grupo. En el ámbito de informática, sistema es un conjunto de elementos relacionados entre sí, que funciona como un todo. Este término es, tal vez, el más trabajado y conceptualizado en todos los tiempos y contextos de la ciencia. Por ello se relacionan a continuación las definiciones de Sistema propuestas por varios autores:

- Ludwig von Bertalanffy (1968) afirma: “Un sistema es un conjunto de unidades en interrelación”.
- Ferdinand de Saussure (1931) afirma: “Sistema es una totalidad organizada, hecha de elementos solidarios que no pueden ser definidos más que los unos con relación a los otros en función de su lugar en esa totalidad”.
- Mario Bunge (1979) afirma: Sistema  $\Sigma$  es una terna ordenada [  $C(\Sigma)$ ,  $E(\Sigma)$ ,  $S(\Sigma)$  ] en la que:  $C(\Sigma)$  (composición de  $\Sigma$ ) representa el conjunto de partes de  $\Sigma$ .  $E(\Sigma)$  (entorno o medio ambiente de  $\Sigma$  es el conjunto de aquellos elementos que, sin pertenecer a  $C(\Sigma)$ , actúan sobre sus componentes o están sometidos a su influencia.  $S(\Sigma)$  (estructura de  $\Sigma$ ) es el conjunto de

relaciones y vínculos de los elementos de  $C(\Sigma)$  entre sí o bien con los miembros del entorno  $E(\Sigma)$ .

- IEEE Standard Dictionary of Electrical and Electronic Terms afirma: Sistema es un todo integrado, aunque compuesto de estructuras diversas, interactuantes y especializadas. Cualquier sistema tiene un número de objetivos, y los pesos asignados a cada uno de ellos puede variar ampliamente de un sistema a otro. Un sistema ejecuta una función imposible de realizar por una cualquiera de las partes individuales. La complejidad de la combinación está implícita.
- Estándar X3.12-1970 (ANSI), Estándar 2382/V, VI (ISO) Vocabulary for Information Processing afirma: “Sistema es una colección organizada de hombres, máquinas y métodos necesaria para cumplir un objetivo específico”. (Flechas Becerra & Almanza, 2012, pág. 13)

#### **2.4. Información**

Es un conjunto de datos procesados y ordenados que proporcionan nuevos conocimientos a un individuo o sistema sobre un determinado ente o fenómeno. Por la cual Luhmann(1998) afirma: “Se denomina información a un acontecimiento que selecciona estados del sistema, para ello se requiere estructuras que limitan y preseleccionan las posibilidades, de tal modo que la información presupone estructura, pero no es en sí misma ninguna estructura”. Más bien, se trata de un acontecimiento que actualiza el uso de las estructuras, ya que los acontecimientos son elementos que se fijan puntualmente en el tiempo, acontecen una sola vez y sólo en el lapso mínimo necesario para su aparición.

Este suceder temporal los identifica y, por lo tanto, son irrepetibles. Precisamente por esto sirven como elementos de unidad del proceso. Esto se comprueba, justamente, por medio de la

información. Una información cuyo carácter de sorpresa se repita ya no es información, conserva su sentido en la repetición, pero pierde el valor de información. Por otro lado, no se pierde la información, aunque haya desaparecido como acontecimiento. Cambió el estado del sistema y dejó con ello un efecto de estructura, el sistema reacciona ante estas estructuras cambiadas y cambia con ellas. La información es, por lo tanto, siempre información de un sistema.

Las siguientes características están asociadas a la información:

a) Para que la información tenga el valor de la función articulada a los estados de sistema, debe tratarse de sistemas autopoieticos<sup>1</sup>, es decir, de sistemas que siempre actúan en la transformación de su propio estado.

b) La información reduce complejidad en la medida en que da a conocer una selección y, en consecuencia, excluye posibilidades. Puede aumentar, no obstante, la complejidad.

c) La información posee también dos rasgos. Por una parte, en cuanto al carácter de sorpresa y, por la otra, si la sorpresa está presupuesta en el sistema de expectativas.

Luhmann(1998) afirma: La información no es la exteriorización de una unidad, sino la selección de una diferencia que lleva a que el sistema cambie de estado y que, por consiguiente, se opere en él otra diferencia. Tomado todo esto en conjunto conduce a la consideración de que la información sólo es posible en el sistema. Cada sistema produce su información ya que cada sistema construye sus propias expectativas y esquemas de ordenamiento.

En resumen, “Luhmann considera que la información es un estado que surge desde dentro del sistema y, por las razones antes expuestas, no fuera de él”. (Rios Ortega, 2013, pág. 27)

---

<sup>1</sup> **Autopoiesis.** Es la capacidad que tiene un sistema, a pesar de no estar en equilibrio, mantener una estabilidad estructural absorbiendo energía del entorno o autorregulándose continuamente.

## **2.5. Geografía**

Es una ciencia que estudia, representa gráficamente y describe la superficie de la tierra en su aspecto físico, actual y natural, también es la que estudia la distribución en la superficie terrestre de los hechos y fenómenos geográficos: físicos, biológicos y humanos, explica sus causas y las relaciones recíprocas entre ellos. “Geografía es la ciencia que estudia el escenario físico en el cual se desenvuelve la actividad humana y las recíprocas relaciones que existen entre la Tierra y el hombre” (Marttone, 2016, pág. 14). En síntesis, la geografía es una ciencia que se encarga de las dimensiones espacio-temporales de la relación naturaleza-cultura-sociedad.

El espacio geográfico es su objeto de estudio primordial y la dimensión temporal le permite explicar cómo se ha construido ese espacio a lo largo de la historia.

La geografía es una ciencia de síntesis, pues requiere la utilización de datos sobre la naturaleza (relieve, clima, suelo, vegetación, hidrología, etc.) y sobre aspectos sociales (población, economía, cultura, política, etc.), con el fin de explicar su espacialidad y temporalidad en conjunto. Es una ciencia de localización porque ubica los hechos y fenómenos socioecológicos en el espacio y en el tiempo, privilegiando al mapa como medio de representación de sus resultados. La geografía de hoy tiene un carácter interdisciplinario, pues asume que las relaciones hombre-naturaleza son inseparables y, por lo tanto, requiere de la información que proveen tanto las ciencias naturales como las ciencias sociales para explicar sus dimensiones espacio-temporales. (Barrera Bassols & Palma Ruiz, 2012, pág. 14 y 15)

## **2.6. Análisis**

En estudios estadísticos análisis se refiere al estudio de objetos o situaciones que se interpretan en cifras, estas pueden ser, volumen, peso y otros. El análisis formal se refiere al estudio

detallado de su origen y contexto, en si el análisis estudia un objeto para conocer su naturaleza, sus características, su estado para determinar conclusiones o factores que intervienen.

El análisis es el proceso de dividir un tema complejo o sustancia en partes más pequeñas para obtener una mejor comprensión de él. “Como concepto formal, el método se ha atribuido a Alhazen, René Descartes (Discurso sobre el método) y Galileo Galilei. También se le ha atribuido a Isaac Newton, en la forma de un método práctico de descubrimiento físico” (Lexico, 2020).

Esta técnica se aplicó en el estudio de las matemáticas y la lógica desde antes de Aristóteles (384–322 a.C.), aunque el análisis como concepto formal es un desarrollo reciente.

## **2.7. Monitoreo**

Es el control de una acción o suceso mediante varios monitores, la cual ayuda a describir un proceso el cual se estudia, observa para realizar un seguimiento respectivo. El monitoreo nos permite inspeccionar y registrar una situación o circunstancia, que se refiere a como, cuando y donde el individuo o las actividades dan lugar.

Monitoreo ayuda a controlar o supervisar una situación. Su origen se encuentra en monitor, un aparato que toma imágenes de instalaciones filmadoras o sensores y que permite visualizar algo en una pantalla. “Esto significa seguimiento y evaluación de una determinada actividad o proceso sin intervenir en el o influenciarlo” (Chiavenato, 2006, pág. 166).

Esto nos permite inferir que monitoreo es la acción y efecto de monitorear, el verbo que se utiliza para nombrar a la supervisión o el control realizado a través de un monitor. Por extensión, el monitoreo es cualquier acción de este tipo, más allá de la utilización de un monitor. (Pérez Porto & Gardey, 2013)

El monitoreo, a rasgos generales, consiste en la observación del curso de uno o más parámetros para detectar eventuales anomalías.

### **2.7.1. Tipos de monitoreo**

El monitoreo manual es más frecuente de lo que podría pensarse. De hecho, es el mecanismo más habitual. En este, uno o varios operadores, revisan parámetros que consideran relevantes de la estructura IT, e intentan detectar condiciones no deseadas para luego poder proceder a la toma de acción.

Habitualmente este modelo es extremadamente costoso en términos de esfuerzo (horas-hombre), ya que el tiempo insumido se repite ad-infinitum, en la medida que no se tomen acciones para automatizarlo.

En el monitoreo automático, se elaboran instrumentos informáticos para efectuar el control, que deben considerar básicamente 3 aspectos:

- Como ejecutar el control (puede ser en forma manual, o automática mediante schedulers)
- Como obtendrá el control la información deseada (una vez disparado)
- Como reportará los resultados

El segundo de estos puntos es el que habitualmente recibe una mayor atención. Si bien es seguramente el más importante de los tres, sin él los otros dos no tienen sentido, es de suma importancia conseguir respuestas efectivas para los otros aspectos.

En el monitoreo automático, por estar estrechamente ligado a la noción de repetición del control, es fundamental la revisión y ajuste de las herramientas utilizadas para asegurar la efectividad a lo largo del tiempo.

## **Local vs Remoto**

El monitoreo remoto ocurre cuando para el control de un determinado servicio IT, toda la ejecución del software de control ocurre sin utilizar recursos del equipamiento monitoreado.

Como ejemplo puede pensarse en el control de un web server. Si desde un equipo "cliente" se realiza un requerimiento http hacia el servidor web, y según su respuesta (o falta de ella) se llega a una cierta conclusión sobre su estado, podemos decir que todo el software de control se ha ejecutado fuera del web server, y por lo tanto estamos en presencia de monitoreo remoto.

En el monitoreo local, se involucran recursos del equipo a monitorear para la ejecución de software de control.

## **Activos vs pasivos**

Este aspecto refiere básicamente a la respuesta a la pregunta: quien "dispara" la ejecución del control. No ya quien la ejecuta, sino quien la dispara.

## **Invasivos vs no-invasivos**

Esta distinción es más sutil, y busca encontrar un punto de equilibrio entre proveer al control de todo lo necesario para que la información que este releva sea suficiente y significativa; y diseñarlo de tal modo que su propia ejecución no resulte disruptiva del servicio que pretende controlar. Puede tomarse como ejemplo una medición sobre la performance de un servidor de base de datos. Si deseamos conocer que tan bien está desempeñándose el servidor frente a las consultas que recibe, y para ello medimos la ejecución de una consulta ad-hoc, incluida en el propio control, de naturaleza particularmente compleja y de costoso procesamiento, seguramente estaremos afectando significativamente con la propia medición, la performance que deseamos controlar. (Georgina, 2012).

Es por esto que se considera deseable balancear estos aspectos de cualquier control, para acercarse lo más posible al óptimo.

## 2.8. Riesgo de quema

Se refiere a que un lugar o algo es propenso a incendiarse, la cual puede destruir lo que no está destinado a arder, como por ejemplo, cosas almacenadas, parajes naturales o edificios.

“Es el peligro relativo de que un incendio se pueda iniciar y expandir, que se puedan generar humos y gases. También se puede producir una explosión poniendo en peligro la vida y seguridad todo a su alrededor” (Parro, 2020). Y de esta manera ocurre un riesgo de quema.

### 2.8.1. Tipos de riesgos de quema

Los tipos de riesgo de quema se muestran en la siguiente tabla:

**Tabla 2.1**

*Tipos de riesgos de quema*

CLASIFICACIÓN	
A	<b>SÓLIDOS:</b> Madera, papel, carbón tejidos, plásticos.
B	<b>LÍQUIDOS:</b> Gasolina, gasóleo, alcohol, petróleo, asfalto.
C	<b>GASES:</b> Butano, propano, gas ciudad, metano, acetileno.
D	<b>METALES:</b> Aluminio, sodio, titanio, productos químicos inorgánicos.

**Nota.** Esta clasificación es importante tenerla en cuenta a la hora de utilizar los sistemas de extinción, ya que no todos son apropiados para todos los tipos de fuegos. Recuperado de (Riesgo de incendio, 2015).

Copyright 2015 por Portal de los riesgos de laborales de los trabajadores de la enseñanza.

## 2.9. Imagen satelital

Es una representación visual de algo, obtenido y registrada por un satélite artificial. Estos satélites cuentan con sensores que permiten recoger información que reflejan a la superficie terrestre, cuando se reciben los datos, se envían de nuevo a la Tierra, donde serán procesados.

Una imagen satelital o imagen de satélite se define como la representación visual de la información capturada por un sensor montado en un satélite artificial. Estos sensores recogen la información reflejada por la superficie de la Tierra que luego es enviada de regreso a ésta y es procesada convenientemente para su respectiva revisión.

Primera imagen satelital de la Tierra transmitida por TV, en 1960, tomada por el satélite meteorológico TIROS-1. La primera imagen satelital de la Tierra fue tomada el 14 de agosto de 1959 por el satélite estadounidense Explorer 6. La primera fotografía satelital de la luna fue tomada por el satélite soviético Luna 3 el 6 de octubre de 1959, en una misión para fotografiar el lado oculto de la Luna. La canica azul, fue tomada en el espacio en 1972, esta fotografía se volvió muy popular en los medios de comunicación y entre la gente. (EcuRed, 2012)

También en 1972 los Estados Unidos comenzaron con el programa Landsat, el mayor programa para la captura de imágenes de la tierra desde el espacio.

### **2.9.1. Tipos de imágenes de satélite**

Los tipos de imágenes satelitales son las siguientes:

- **Imágenes Pancromáticas**

La imagen Pancromática se realiza cuando el sensor de imagen se vuelve sensible a una gran cantidad de longitudes de onda de luz, normalmente a ambos lados de una parte visible del espectro.

El sensor es un detector de canal que es sensible a la radiación dentro del rango de la longitud de onda amplia. Si hay una coincidencia del rango visible con el rango de longitud de onda, entonces la imagen resultante se ve como una foto en "blanco y negro" desde el espacio. La

cantidad física medida es el brillo aparente del objetivo y, por lo tanto, se pierde el "color" de los resultados o la información espectral. (Earth Observing System, 2020)

Esto muestra que permite todos los colores, lo que significa que la banda contiene una señal de amplio rango.

La producción de las imágenes pancromáticas se realiza mediante satélites como SPOT6/7 y el tipo de satélites de Landsat. Esta banda única de imágenes "mezcla" la información de las bandas azules, verdes y rojas visibles utilizando la totalidad de energía de la luz que ilumina el espectro visible en lugar de dividirla en un espectro diferente.

- **Imágenes Multiespectrales**

Una imagen Multiespectral es la que captura datos de imágenes dentro de rangos de longitud de onda específicos a través del espectro electromagnético. Las longitudes de onda pueden estar separadas por filtros o mediante el uso de instrumentos sensibles a longitudes de onda particulares, incluida la luz de frecuencias más allá del rango de luz visible, como infrarrojo y ultravioleta. “La imagen Multispectral divide la luz en un pequeño número (normalmente de 3 a 15) de bandas espectrales. La obtención de imágenes Hiperespectrales es un caso especial de imágenes espectrales donde a menudo hay cientos de bandas espectrales contiguas” (Fernando, 2016). La obtención de imágenes espectrales puede permitir la extracción de información adicional que el ojo humano no captura con sus receptores rojo, verde y azul. Fue desarrollada originalmente para imágenes obtenidas desde el espacio, y también ha encontrado aplicación en el análisis de documentos y la pintura.

- **Imágenes Hiperespectrales**

Una imagen Hiperespectral es una imagen que tiene varias bandas espectrales de información a través de todo el espectro electromagnético.

La formación de imágenes Hiperespectrales consiste en recopilar y procesar información a lo largo de todo el espectro electromagnético. La formación de imágenes espectrales divide el espectro en muchas bandas. Esta técnica de dividir las imágenes en bandas puede extenderse más allá de lo visible. Y así surge la técnica de formación de imágenes Hiperespectrales.

Los Ingenieros construyen sensores y sistemas de procesamiento para la aplicación de este método de visión en la agricultura, la mineralogía, la física, y la vigilancia. Los sensores hiperespectrales miran a los objetos usando una vasta porción del espectro electromagnético. Ciertos objetos dejan unas huellas únicas a lo largo de todo el espectro electromagnético catalogándolos como si de huellas dactilares se tratara. Estas 'huellas dactilares' que se conocen como firmas espectrales permiten la identificación de los materiales que componen un objeto analizado. (Gonzalez, 2013)

Por ejemplo, una firma espectral para el petróleo ayuda a mineralogistas a encontrar nuevos campos de petróleo.

## **2.10. Sistemas de información geográfica**

Es un sistema de información que está diseñado para trabajar con base de datos espaciales también está diseñado para manejar datos cartográficos que son datos referenciados coordenadas de mapas para una ubicación espaciales.

SIG es un conjunto de software y hardware diseñado específicamente para la adquisición, mantenimiento y uso de datos cartográficos. “Un SIG es un elemento que permite analizar, presentar interpretar hechos relativos a la superficie terrestre” (Tomli, 2015, pág. 9). Esta es una definición muy amplia, y habitual mente se emplea otra más concreta.

Un SIG se define como un sistema de información diseñado para trabajar con datos referenciados coordenadas espaciales o geográficas. En otras palabras, un SIG es tanto un

sistema de base de datos con capacidades especializadas para datos georeferenciados, como un conjunto de operaciones para trabajar con esos datos. En cierto modo, un SIG es un mapa de orden superior. (Jeffrey & John, 2006, pág. 10).

La definición actual de un SIG debe fundamentarse sobre todo en el concepto de sistema como elemento integrador que engloba a un conjunto de componentes interrelacionados.

Como se refiere: “software y hardware son dos elementos primordiales del SIG, pero no son sin embargo los únicos. En el contexto actual, otros componentes juegan un papel igual de importante en la ideal global de un SIG”. (Tomli, 2015, pág. 10) De igual modo, un SIG puede considerarse como un mapa de orden superior, entendiendo que se trata de una forma más potente y avanzada de hacer todo aquello que, previamente a la aparición de los SIG, se llevaba a cabo mediante el uso de mapas y cartografía en sentido clásico.

Es decir, los SIG representan un paso más allá de los mapas. No obstante, esta definición resulta en exceso simplista, pues mapas y SIG no son conceptos equiparables en el contexto actual de estos últimos. (Olaya, Sistemas de Información Geográfica, 2014, pág. 7)

### **2.10.1. Representaciones de sistemas de información geográfica**

El comportamiento de un SIG para representar y administrar la información geográfica se basa en tres representaciones o expresiones de información geográfica fundamentales las cuales son:

- **Entidades (colecciones de puntos, líneas y polígonos)**
- **Atributos**
- **Imágenes**

#### **Entidades: puntos, líneas y polígonos**

Las entidades geográficas son representaciones de cosas ubicadas en la superficie de la Tierra o cercanas a ella. Las entidades geográficas pueden ocurrir de forma natural (por ejemplo, ríos y

vegetación) y pueden ser construcciones (como carreteras, canalizaciones, pozos y edificios) o subdivisiones de tierra (como condados, divisiones políticas y parcelas de terreno).

Aunque existen diversos tipos de entidades adicionales, las entidades geográficas se representan más comúnmente como puntos, líneas o polígonos.

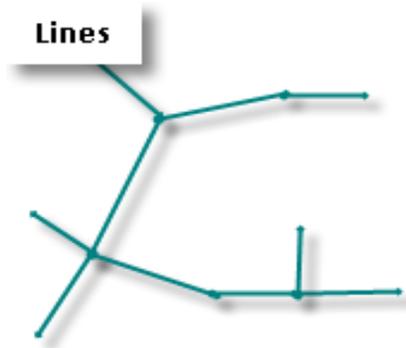
- **Puntos:** definen ubicaciones discretas de entidades geográficas demasiado pequeñas para mostrarse como líneas o áreas, por ejemplo, ubicaciones de pozos, postes de teléfono y estaciones hidrométricas. Los puntos también pueden representar ubicaciones de dirección, coordenadas GPS o picos de montañas.



*Figura 2.1. Entidades de punto*

**Fuente: Esri, 2020.**

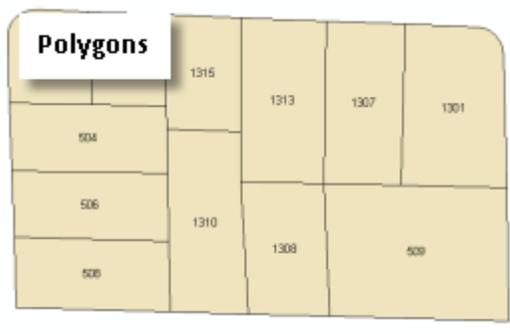
- **Líneas:** representan la forma y la ubicación de objetos geográficos demasiado estrechos para mostrarse como áreas (tales como líneas de centro de calle y arroyos). Las líneas también se utilizan para representar las entidades que tienen longitud, pero no área, como líneas de curvas de nivel y límites administrativos. (Las curvas de nivel son interesantes, como se podrá leer más adelante, porque proporcionan diversas alternativas para representar superficies continuas.)



**Figura 2.2. Entidades de líneas.**

Fuente: Esri, 2020

- **Polígonos:** son áreas cerradas (figuras de muchos lados) que representan la forma y la ubicación de entidades homogéneas como estados, condados, parcelas, tipos de suelo y zonas de uso del suelo. En el ejemplo siguiente, los polígonos representan parcelas de terreno.



**Figura 2.3. Entidades de polígonos.**

Fuente: Esri, 2020.

### Atributos

Los mapas transmiten información descriptiva a través de símbolos de mapa, colores y etiquetas.

Por ejemplo:

- Las carreteras se muestran en función de su clase (como símbolos de línea que representan autopistas divididas, calles principales, vías residenciales, carreteras sin asfaltar y pistas).
- Los arroyos y las masas de agua se dibujan en color azul para indicar el agua.
- Las calles de ciudades se etiquetan con sus nombres y a menudo contienen alguna información de rango de direcciones.
- Símbolos de línea y punto especiales denotan entidades específicas como vías férreas, aeropuertos, escuelas, hospitales e incidentes de diversos tipos.

En un SIG, los atributos descriptivos se administran en tablas, que se basan en una serie de conceptos de base de datos relacional esenciales. Las tablas de atributos proporcionan un modelo de datos sencillo y universal para almacenar y trabajar con la información de atributos. Están inherentemente abiertas porque su simplicidad y su flexibilidad permiten la compatibilidad con una amplia variedad de aplicaciones. Entre los conceptos clave se incluyen los siguientes:

- Los datos descriptivos se organizan en tablas.
- Las tablas contienen filas.
- Todas las filas de una tabla tienen las mismas columnas.
- Cada columna tiene un tipo, como un entero, un número decimal, un carácter y una fecha.
- Dentro de las bases de datos relacionales, estos conceptos se amplían para incluir una serie de operadores y funciones relacionales que pueden utilizarse para operar sobre las tablas y sus elementos de datos. Esto se conoce como Lenguaje estructurado de consultas o SQL.

Tabular View Feature class table				Related ownership table					
PIN	Area	Addr	Code	PIN	Owner	Relat.	Acq.Date	Assessed	TaxStat
334-1626-001	7,342	341 Cherry Ct.	SFR	334-1626-001	G. Hall	SO	1995/10/20	\$115,500.00	02
334-1626-002	8,020	343 Cherry Ct.	UND	334-1626-002	H. L. Holmes	UK	1993/10/06	\$24,375.00	01
334-1626-003	10,031	345 Cherry Ct.	SFR	334-1626-003	W. Rodgers	HW	1980/09/24	\$175,500.00	02
334-1626-004	9,254	347 Cherry Ct.	SFR	334-1626-004	J. Williamson	HW	1974/09/20	\$135,750.00	02
334-1626-005	8,856	348 Cherry Ct.	UND	334-1626-005	P. Goodman	SO	1966/06/06	\$30,350.00	02
334-1626-006	9,975	346 Cherry Ct.	SFR	334-1626-006	K. Staley	HW	1942/10/24	\$120,750.00	02
334-1626-007	8,230	344 Cherry Ct.	SFR	334-1626-007	J. Dormandy	UK	1996/01/27	\$110,650.00	01
334-1626-008	8,645	342 Cherry Ct.	SFR	334-1626-008	S. Gooley	HW	2000/05/31	\$145,750.00	02

*Figura 2.4. Tablas relacionadas.*

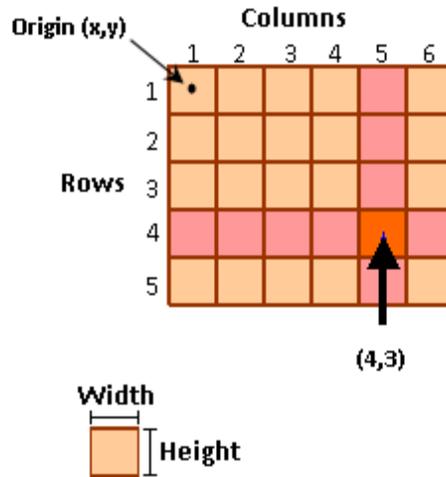
Fuente: Esri, 2020.

## Imágenes

Con el término imágenes, en SIG a menudo se hace referencia a una serie de tipos fuentes de datos basadas en celdas o en píxeles para satélites, fotografía aérea, modelos digitales de elevación Dataset Ráster etc.

Las imágenes se administran como un tipo de datos Ráster compuesto por celdas organizadas en una cuadrícula de filas y columnas. Además de la proyección cartográfica, el sistema de coordenadas para un Dataset Ráster incluye su tamaño de celda y una coordenada de referencia generalmente la esquina superior o inferior izquierda de la cuadrícula.

Estas propiedades permiten que se describa un Dataset Ráster por medio de una serie de valores de celda comenzando por la fila superior izquierda. (ArcGIS Resources, 2020)



**Figura 2.5. Tres presentaciones fundamentales de capas de información geográfica.**

Fuente: Esri, 2020. Cada ubicación de celda se puede localizar automáticamente mediante una coordenada de referencia para el origen, el tamaño de celda y el número de filas y columnas.

## 2.11. Módulo

Es una estructura de piezas o partes, que se ubican en cantidad y que ayuda a realizarse de una manera más sencilla, regular y eficiente una tarea, por lo tanto, todo modulo forma parte de un sistema y está conectado de alguna manera a el resto de los componentes.

En programación, un módulo es una porción de un programa de ordenador. De las varias tareas que debe realizar un programa para cumplir con su función u objetivos, un módulo realizará, comúnmente, una de dichas tareas o varias, en algún caso. En sí, un módulo es un componente de un sistema más grande y opera dentro del sistema independientemente de las operaciones de otros componentes.

En general un módulo recibe como entrada la salida que haya proporcionado otro módulo o los datos de entrada al sistema si se trata del módulo principal de éste; y proporcionará una salida que, a su vez, podrá ser utilizada como entrada de otro módulo o bien contribuirá directamente a la salida final del sistema o programa, si se retorna al módulo principal.

Particularmente, en el caso de la programación, los módulos suelen estar no necesariamente organizados jerárquicamente en niveles, de forma que hay un módulo principal que realiza las llamadas oportunas a los módulos de nivel inferior.

Cuando un módulo es convocado, recibe como entrada los datos proporcionados por otro del mismo o superior nivel, el que ha hecho la llamada; luego realiza su tarea. A su vez este módulo convocado puede llamar a otro u otros módulos de nivel inferior si fuera necesario; cuando ellos finalizan sus tareas, devuelven la salida pertinente al módulo inmediato llamador, en secuencia reversa. Finalmente se continúa con la ejecución del módulo principal. (Infomática, 2017)

### **2.11.1. Tipos de módulos**

Cada uno de los módulos de un programa idealmente debería cumplir las siguientes características:

**Tamaño relativamente pequeño.-** Esto facilita aislar el impacto que pueda tener la realización de un cambio en el programa, bien para corregir un error, o bien por rediseño del algoritmo correspondiente.

**Independencia modular.-** Cuanto más independientes son los módulos entre sí, más fácil y flexiblemente se trabajará con ellos. Esto implica que para desarrollar un módulo no es necesario conocer detalles internos de otros módulos. Como consecuencia de la independencia modular, un módulo cumplirá:

- **Características de caja negra:** es decir, abstracción, una caja negra es un elemento que se estudia desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una caja negra nos interesará su forma de interactuar con el medio que le rodea (en ocasiones, otros

elementos que también podrían ser cajas negras) entendiendo qué es lo que hace, pero sin dar importancia a cómo lo hace. Por tanto, de una caja negra deben estar muy bien definidas sus entradas y salidas, es decir, su interfaz; en cambio, no se precisa definir ni conocer los detalles internos de su funcionamiento.

- **Aislamiento de los detalles mediante encapsulamiento**, se denomina encapsulamiento al ocultamiento del estado, es decir, del dato miembro de un objeto de manera que solo se pueda cambiar mediante las operaciones definidas para ese objeto.
- **Cada objeto está aislado del exterior**, es un módulo natural, y la aplicación entera se reduce a un agregado o rompecabezas de objetos. El aislamiento protege a los datos asociados de un objeto contra su modificación por quien no tenga derecho a acceder a ellos, eliminando efectos secundarios e interacciones. (Fing, 2017)

La independencia modular mejora el rendimiento humano, pudiendo realizarse programación en equipo y desarrollar módulos paralelamente. También contribuye a la reutilización de software.

## **2.12. Visor geográfico**

Es una herramienta donde se muestra datos en forma de mapa, donde permite graficar valores que se representan, también nos permite acceder a información espacial.

Visor geográfico es una herramienta que ofrece una solución completa para consultas geográficas. Visualiza los elementos de la red eléctrica, la cartografía base y cualquier elemento de una base de datos conectada. Clasifica estos elementos a través de mapas temáticos que crea el usuario en forma fácil y rápida. (Energy computer systems, 2020)

## **2.13. Infraestructura de datos espaciales**

Una Infraestructura de Datos Espaciales (IDE) integra datos, metadatos, servicios web (WMS, WFS, WCS) e información de tipo geográfico para promover su uso.

Una IDE es el conjunto de tecnologías, políticas, estándares y recursos humanos para adquirir, procesar, almacenar, distribuir y mejorar la difusión de la información geográfica. Al igual que las carreteras y autopistas facilitan el transporte de vehículos, las IDE facilitan el transporte de información geoespacial. Las IDE promueven el desarrollo social, económico y ambiental del territorio.



**Figura 2.6. Infraestructura de datos espaciales.**

Fuente: TYC GIS Soluciones Integrales SL, 2019

Los factores que han contribuido a la necesidad de crear las diferentes IDEs han sido:

- **El aumento de la producción de datos espaciales**, no sólo por parte de organismos públicos sino también, por empresas privadas e incluso particulares. Este aumento de producción de calidad de los datos, ha sido posible gracias a grandes inversiones, tanto de las administraciones como de las grandes empresas, para el desarrollo de sistemas tecnológicos avanzados como son los sensores teletransportados, la tecnología de teledetección, GPS y otros sistemas de captura de datos terrestres.
- **El alza en la distribución de dispositivos móviles con aplicaciones e información espacial**, y la posibilidad de conectar estos dispositivos a redes inalámbricas que

posibilitan el acceso a la información en cualquier momento y en cualquier lugar, han facilitado al usuario el contacto con la información geográfica y con una frecuencia de uso inimaginable hace unos años.

- **La accesibilidad del usuario a la información de forma gratuita**, no sólo mediante la visualización sino también a partir de servicios de descarga, ha significado un aumento de la demanda que necesita ir parejo de un sistema de organización y comparación de datos.

**(Bermejo, 2019)**

## **2.14. Ingeniería de software**

Es diseñar, construir aplicaciones informáticas, estimar costes de un proyecto, hacer seguimiento de costes y plazos, organizar la realización de pruebas las cuales verifiquen el correcto funcionamiento de los programas, incorporar procedimientos de calidad en los sistemas y verificar que el sistema se ajuste a las necesidades de las organizaciones.

La Ingeniería de Software es una de las ramas de las ciencias de la computación que estudia la creación de software confiable y de calidad, basándose en métodos y técnicas de ingeniería. Brindando soporte operacional y de mantenimiento, el estudio de las aplicaciones de la ingeniería de software. Integra ciencias de la computación, ciencias aplicadas y las ciencias básicas en las cuales se encuentra apoyada la ingeniería.

Se citan las definiciones más reconocidas, formuladas por prestigiosos autores:

Ingeniería de software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas software (Zelkovitz, 1978).

Ingeniería de software es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para

desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software (Bohem, 1976).

La propuesta por Fritz Bauer (1972) es: “La ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales”. (Pressman, Ingeniería del Software, Un enfoque práctico, 2010, pág. 11)

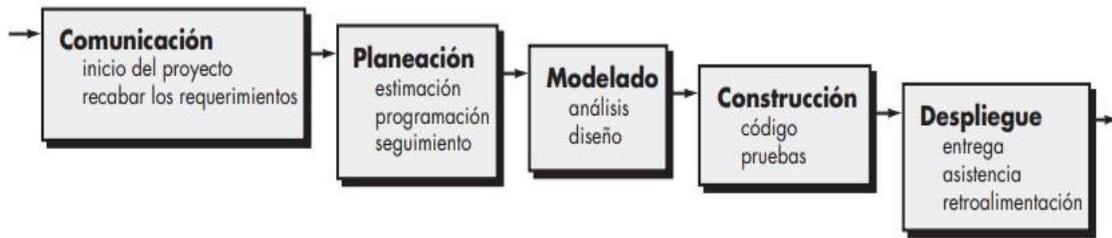
### **2.14.1. Módulos del desarrollo de software**

Los modelos de proceso prescriptivo fueron propuestos originalmente para poner orden en el caos del desarrollo de software. La historia indica que estos modelos tradicionales han dado cierta estructura útil al trabajo de ingeniería de software y que constituyen un mapa razonablemente eficaz para los equipos de software. Sin embargo, algunos modelos son más apropiados para producir ciertos tipos de sistemas, de forma que si no se utiliza un modelo adecuado puede ocurrir que el sistema software resultante sea de menor calidad.

El autor los llama “prescriptivos” porque prescriben un conjunto de elementos del proceso: actividades estructurales, acciones de ingeniería de software, tareas, productos del trabajo, aseguramiento de la calidad y mecanismos de control del cambio para cada proyecto. Cada modelo del proceso también prescribe un flujo del proceso (también llamado flujo de trabajo), es decir, la manera en la que los elementos del proceso se relacionan entre sí.

- **Modelo en cascada**

El modelo de la cascada, a veces llamado ciclo de vida clásico, sugiere un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue, para concluir con el apoyo del software terminado.

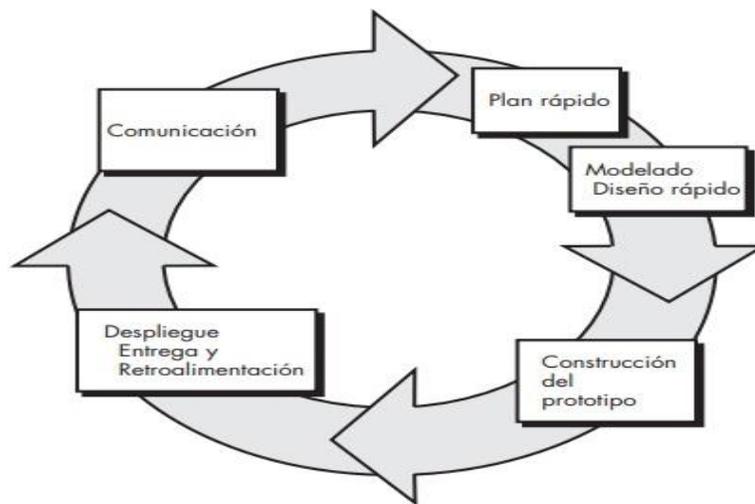


**Figura 2.7. Modelo Cascada**

Fuente: Pressman, 2010

- **Modelo evolutivo**

El software, como todos los sistemas complejos, evoluciona en el tiempo. Es frecuente que los requerimientos del negocio y del producto cambien conforme avanza el desarrollo, lo que hace que no sea realista trazar una trayectoria rectilínea hacia el producto final. En estas situaciones y otras parecidas se necesita un modelo de proceso diseñado explícitamente para adaptarse a un producto que evoluciona con el tiempo.



**Figura 2.8. Modelo Evolutivo**

Fuente: Pressman, 2010

Existen dos tipos de procesos de desarrollo evolutivos:

**Exploratorio:** Su objetivo es trabajar con el cliente para identificar y construir el sistema final a partir de una especificación informal. El resultado del proceso es el sistema final.

**Prototipado desechable:** Su objetivo es entender los requisitos del cliente. El resultado del proceso es la especificación del sistema el prototipo se deshecha.

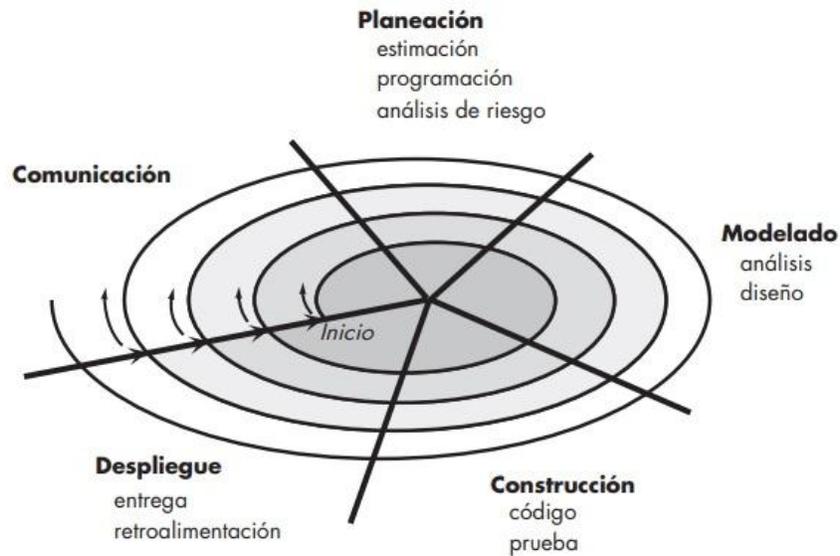
Los principales problemas de este modelo son: escasa visibilidad; los continuos cambios que hacen que los sistemas desarrollados estén deficientemente estructurados; y la necesidad de disponer, en muchos casos, de un equipo de desarrollo altamente calificado. Estos problemas hacen que la aplicación de este modelo se suela limitar a sistemas interactivos de tamaño pequeño o mediano. La deficiente estructura dificulta las tareas de mantenimiento de ahí que se suela aplicar a sistemas con una vida corta y a partes de grandes sistemas, especialmente a sistemas de inteligencia artificial y a interfaces de usuario.

- **Modelo Espiral**

Propuesto en primer lugar por Barry Boehm [Boe88], el modelo espiral es un modelo evolutivo del proceso del software y se acopla con la naturaleza iterativa de hacer prototipos con los aspectos controlados y sistémicos del modelo de cascada. Tiene el potencial para hacer un desarrollo rápido de versiones cada vez más completas. Boehm [Boe01a] describe el modelo del modo siguiente:

El modelo de desarrollo espiral es un generador de modelo de proceso impulsado por el riesgo, que se usa para guiar la ingeniería concurrente con participantes múltiples de sistemas intensivos en software. Tiene dos características distintivas principales. La primera es el enfoque cíclico para el crecimiento incremental del grado de definición de un sistema y su implementación, mientras que disminuye su grado de riesgo. La otra es un conjunto de puntos

de referencia de anclaje puntual para asegurar el compromiso del participante con soluciones factibles y mutuamente satisfactorias.

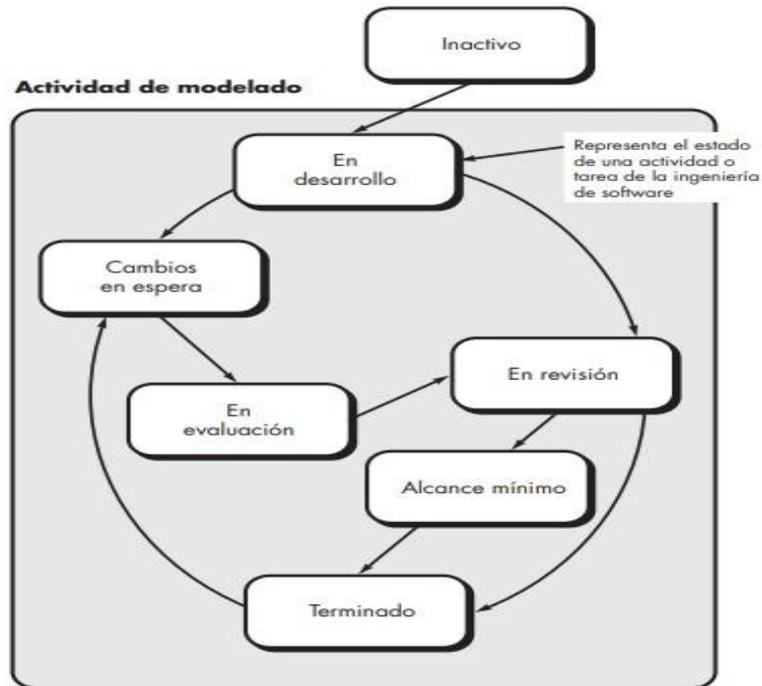


**Figura 2.9. Modelo Espiral**

Fuente: Pressman, 2010

- **Modelo Concurrente**

El modelo de desarrollo concurrente, en ocasiones llamado ingeniería concurrente, permite que un equipo de software represente elementos iterativos y concurrentes de cualquiera de los modelos de proceso descritos.



**Figura 2.10. Modelo Concurrente.** Se muestra la representación esquemática de una actividad de ingeniería de software dentro de la actividad de modelado con el uso del enfoque de modelado concurrente.

Fuente: Pressman, 2010

### 2.14.2. Arquitectura del software

Es el conjunto de estructuras para razonar acerca del sistema, la arquitectura se refiere a como organizar las partes del sistema y como conectarlas. La arquitectura de software también es un conjunto de decisiones de diseño para organizar el software y producir atributos de calidad deseados.

En los inicios de la informática, la programación se consideraba un arte y se desarrollaba como tal debido a la dificultad para la mayoría de las personas, pero con el tiempo se han ido descubriendo y desarrollando formas y guías generales, con base a las cuales se puedan resolver los problemas. “La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellos” (Pressman, Ingeniería del Software, un

enfoque práctico, 2010, pág. 207). A estas, se les ha denominado arquitectura de software, porque, a semejanza de los planos de un edificio o construcción, estas indican la estructura, funcionamiento e interacción entre las partes del software. “Definen que la arquitectura es un nivel de diseño que hace foco en aspectos más allá de los algoritmos y estructuras de datos de la computación; el diseño y especificación de la estructura global del sistema es un nuevo tipo de problema”. ( Garlan & Shaw, 2016, pág. 4)

#### **2.14.2.1. Descomposición Modular**

Es un elemento que tiene la función de agrupar de distintas maneras instrucciones de programación que cumplen una función específica. Esto quiere decir que modularizar significa dividir un problema en partes, que tengan operaciones y datos lógicos y datos de comunicación especificados.

Descomposición Modular o Modularización es el proceso de descomposición de un sistema en un conjunto de elementos con un índice bajo acoplamiento independiente y alto índice de coherencia con significado propio.

Consiste en descomponer el problema a resolver en módulos o tareas más simples. Cada tarea representa una actividad completa y se codifica de manera independiente. Facilita el diseño descendente del problema, centrándonos cada vez en la resolución de subproblemas de magnitud inferior.

A la resolución de cada uno de estos subproblemas de complejidad inferior se denomina refinamiento por pasos. Los módulos pueden ser planificados, codificados, comprobados y depurados independientemente, y a continuación se combinan uno a uno con otros módulos.

El diseño modular propone dividir el sistema en partes diferenciadas y definir sus interfaces.  
(ittgweb, 2016)

## **Ventajas**

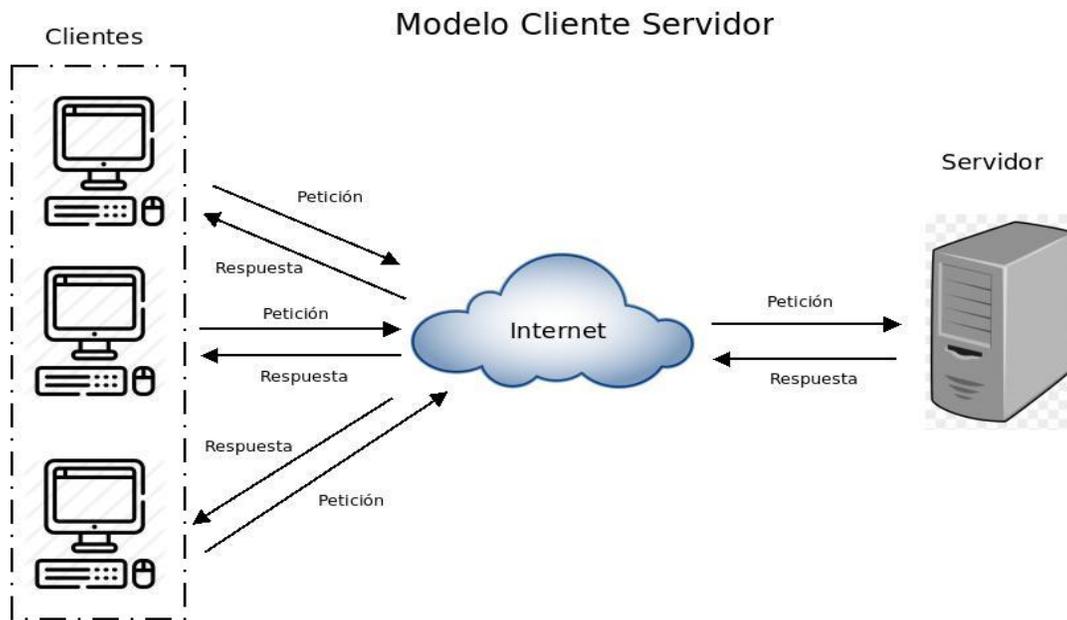
- Claridad
- Reducción de costos
- Reutilización

Se siguen los siguientes pasos para poder realizar la descomposición modular:

- Identificar los módulos
- Describir cada módulo
- Describir las relaciones entre módulos.

### **2.14.2.2. Cliente – Servidor**

La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta. Esta idea también se puede aplicar a programas que se ejecutan sobre una sola computadora, aunque es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.



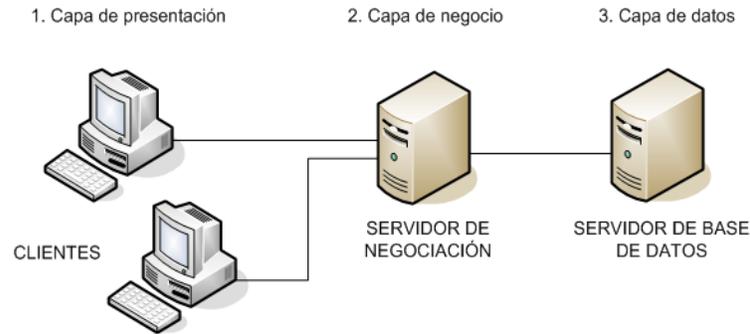
**Figura 2.11. Cliente – Servidor.**

Fuente: Schiaffarino, 2019.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un solo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. (Schiaffarino, 2019)

### **2.14.2.3. Programación por capas**

La programación por capas es un modelo de desarrollo software en el que el objetivo primordial es la separación (desacoplamiento) de las partes que componen un sistema software o también una arquitectura cliente-servidor: lógica de negocios, capa de presentación y capa de datos. De esta forma, por ejemplo, es sencillo y mantenible crear diferentes interfaces sobre un mismo sistema sin requerirse cambio alguno en la capa de datos o lógica.



**Figura 2.12. Programación por capas.**

Fuente: EcuRed, 2019.

La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, solo afectará al nivel requerido sin tener que revisar entre el código fuente de otros módulos, dado que se habrá reducido el Acoplamiento informático hasta una interfaz de paso de mensajes.

Además, permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

### Capas y niveles

- **Capa de presentación:** la que ve el usuario (también se la denomina «capa de usuario»), presenta el sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). También es conocida como interfaz gráfica y debe tener la característica de ser «amigable» (entendible y fácil de usar) para el usuario. Esta capa se comunica únicamente con la capa de negocio.
- **Capa de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se 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 almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

- **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. (Virtuniversidad, 2016)

#### 2.14.2.4. Seguridad

Para ello existen una serie de estándares, protocolos, métodos, reglas, herramientas y leyes concebidas para minimizar los posibles riesgos a la infraestructura o a la información. La ciberseguridad comprende software (bases de datos, metadatos, archivos), hardware, redes de computadoras y todo lo que la organización valore y signifique un riesgo si esta información confidencial llega a manos de otras personas, convirtiéndose, por ejemplo, en información privilegiada.

La seguridad siempre busca la gestión de riesgos, esto quiere decir que se tenga siempre una forma de evitarlo o prevenirlo y que se pueda realizar ciertas acciones para evitar esas situaciones de la mejor forma. Se definió que la seguridad podría ser catalogada como la ausencia de riesgo, la definición de este término involucra cuatro acciones que siempre están inmersas en cualquier asunto de seguridad como son:

- Prevención del riesgo
- Transferir el riesgo
- Mitigar el riesgo

- Aceptar el riesgo

Así que, cuando se está buscando hacer algo más seguro, estas acciones son algo que se debe de considerar sin importar el área, se aplica a cualquier intento de tener mejor o mayor seguridad en cualquier tema que se requiere. (Romero Castro, 2018, pág. 13)

La seguridad informática está concebida para proteger los activos informáticos, entre los que se encuentran los siguientes:

**La infraestructura computacional:** es una parte fundamental para el almacenamiento y gestión de la información, así como para el funcionamiento mismo de la organización. La función de la seguridad informática en esta área es velar por que los equipos funcionen adecuadamente y anticiparse en caso de fallos, robos, incendios, sabotajes, desastres naturales, fallos en el suministro eléctrico y cualquier otro factor que atente contra la infraestructura informática.

**Los usuarios:** son las personas que utilizan la estructura tecnológica, zona de comunicaciones y que gestionan la información. Debe protegerse el sistema en general para que el uso por parte de ellos no pueda poner en entredicho la seguridad de la información y tampoco que la información que manejan o almacenan sea vulnerable.

**La información:** esta es el principal activo. Utiliza y reside en la infraestructura computacional y es utilizada por los usuarios.

#### **2.14.2.5. Calidad**

Es un proceso eficaz de software que se aplica de manera que crea un producto útil que proporciona valor medible a quienes lo producen y a quienes lo utilizan.

Hay pocas dudas acerca de que la definición anterior podría modificarse o ampliarse en un debate sin fin. Se toma en cuenta tres puntos importantes:

- Un proceso eficaz de software establece la infraestructura que da apoyo a cualquier esfuerzo de elaboración de un producto de software de alta calidad. Los aspectos de administración del proceso generan las verificaciones y equilibrios que ayudan a evitar que el proyecto caiga en el caos, contribuyente clave de la mala calidad.
- Un producto útil siempre satisface los requerimientos establecidos en forma explícita por los participantes. Además, satisface el conjunto de requerimientos, por ejemplo, la facilidad de uso, con los que se espera que cuente el software de alta calidad.
- Al agregar valor para el productor y para el usuario de un producto, el software de alta calidad proporciona beneficios a la organización que lo produce y a la comunidad de usuarios finales. La organización que elabora el software obtiene valor agregado porque el software de alta calidad requiere un menor esfuerzo de mantenimiento, menos errores que corregir y poca asistencia al cliente. **(Pressman, Ingeniería del Software, un enfoque práctico, 2010, pág. 340)**

### 2.14.3. Metodología UWE

Es una herramienta que permite modelar aplicaciones web, utilizada en la ingeniería web, centrado en la sistematización y personalización de sistemas adaptativos, esta metodología no limita el número de vistas de una aplicación, también nos ayuda a realizar correctamente los sistemas con ayuda de sus fases.

UWE (Koch y Kraus, 2003) es una metodología con un modelado de proceso espiral / interactivo – incremental y tomando como lenguaje de notación UML, esta metodología está enfocada en el diseño sistémico, la personalización y la generación semiautomática de escenarios. UWE utiliza vistas especiales soportadas por los diagramas gráficos de UML, como el modelo de navegación y el modelo de presentación. UWE no limita el número de vistas de una aplicación, ya que los

diseñadores también pueden hacer uso de otra técnica de modelado UML para agregar otras vistas a la aplicación.

UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

UWE propone los siguientes pasos para el desarrollo de una aplicación web:

- **Especificación de requerimientos:** es donde se describen los requisitos funcionales de la aplicación a desarrollar. UWE propone el modelado de casos de uso de UML para el levantamiento de los requerimientos, ya que a través de esta herramienta se puede describir una parte del comportamiento de la aplicación sin revelar la estructura interna, así como la identificación de los distintos usuarios que interactúan con la aplicación.
- **Modelo de Lógico-Conceptual:** en este paso se especifican los elementos del dominio de la aplicación. UWE propone para este paso la utilización de un diagrama de clases de UML.
- **Modelo de Navegación:** en este paso se genera la especificación de los objetos que pueden ser visitados mediante la navegación dentro de la aplicación Web y las asociaciones entre ellos. En UWE los modelos de la navegación son representados por los diagramas de clases estereotipos de UML.
- **Modelo de presentación:** en este paso define como los usuarios visualizaran los objetos de navegación y las primitivas de acceso. Para este modelo UWE propone una forma particular de un diagrama de clase, el cual representa de forma gráfica, las clases definidas anteriormente.

## **Fases de la UWE.**

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrando además su atención en aplicaciones personalizadas o adaptativas.

Las fases o etapas a utilizar son:

- 1. Captura, análisis y especificación de requisitos:** En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.

Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales.

Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

- 2. Diseño del sistema:** Se basa en la especificación de requisitos producido por el análisis de los requerimientos, fase de análisis, el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.

- 3. Codificación del software:** Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

- 4. Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.

- 5. La Instalación o Fase de Implementación:** es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados, y, eventualmente, configurados; todo ello con el propósito de ser ya utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

**6. El Mantenimiento:** es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control. (Vilariño, 2010)

#### **2.14.4. Métricas de calidad al software ISO 9126**

Nos ayuda a medir el proceso de control calidad del trabajo o del producto, esto se refiere a que la métrica de calidad puede ser el tiempo de respuesta de un sistema para realizar un reporte de datos específicos.

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software, a los requerimientos implícitos y explícitos del cliente.

#### **Modelo de Calidad Establecido por el estándar ISO 9126**

La ISO, bajo la norma ISO-9126, ha establecido un estándar internacional para la evaluación de la calidad de productos de software el cual fue publicado en 1992 con el nombre de “Information technology –Software product evaluation: Quality characteristics and guidelines for their use”, en el cual se establecen las características de calidad para productos de software.

El estándar ISO-9126[Sanders, Joc & Eugene Curran. Software Quality. A Framework for Success in Software Development and Support, Addison Wesley] establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de seis características básicas, las cuales son: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad; cada una de las cuales se detalla a través de un conjunto de subcaracterísticas. (González Pinzón & González Sanabria, 2013)

Estas permiten profundizar en la evaluación de la calidad de productos de software.

#### **2.14.4.1. Metodología WEB - QEM**

La metodología WEB – QEM nos sirve para medir la calidad del sistema que proporciona un enfoque sistémico y cuantitativo para la evaluación y comparación de la calidad de sitios y aplicaciones web. (M. F., M. E., G., & L., 2017)

#### **2.14.4.2. Fases de WEB - QEM**

A continuación se describe las fases de la metodología WEB - QEM que se utiliza para realizar los cálculos que ayudara a comprobar que el sistema es confiable, eficiente y de calidad.

##### **A. Definición de Las Metas de Evaluación y Selección del Perfil de Usuario**

En esta fase se consideran dos pasos primordiales y comunes en toda evaluación de calidad siguiendo la metodología WEB – QEM.

##### **i) Metas de Evaluación**

En esta fase se define las metas de evaluación y selección del perfil de usuario, los evaluadores deben definir las metas y establecer el alcance del proyecto.

Las metas a llegar a cumplir con la evaluación del sistema son:

- Conocer la percepción de la calidad que tienen los usuarios finales sobre el sistema.
- Lograr que la calidad del sistema esté por encima de las expectativas del usuario final.

##### **ii) Selección de Perfil de Usuario**

Para el diseño e implementación de un caso de estudio de evaluación de calidad en uso, una meta muy utilizada es determinar el cumplimiento de requerimientos elementales, parciales y globales de calidad para una aplicación Web operativa, considerando el perfil de encargado de requerimiento de pedidos y compras.

Los encargados de realizar los pedidos y compras son los usuarios que interactuaran con el sistema de forma recurrente y a diario estos usuarios conocen la lógica del negocio y tuvieron capacitación sobre el manejo del sistema.

**B. Definición de los Requerimientos de calidad y/o Costo:** En esta fase, teniendo en cuenta los aspectos definidos en la fase anterior respecto a metas de evaluación, selección de perfil de usuario, se deben establecer atributos y subconceptos (características) de calidad cuantificables que, agrupados jerárquicamente, representen un modelo de calidad apropiado para el perfil de usuario seleccionado.

Para esto se realiza un árbol de requisitos:

» **1. Usabilidad**

- Comprensibilidad del Sistema
- El sistema debe ser amigable para el usuario
- Aspectos de Interfaces y Estéticos
- Los datos deben estar ordenados de la forma correcta para una mejor comprensión
- Las interfaces deben ser claras

» **2. Funcionalidad**

- Aspectos de Búsqueda
- Mecanismo de Búsqueda
- Búsqueda parcial bajo diversos parámetros
- Mostrar todos los resultados coincidentes con los parámetros ingresados
- Validar todos los parámetros de ingreso la búsqueda.
- Aspectos de Navegación y Exploración
- Navegabilidad Local

- Navegación solo por interfaces asignados al rol del usuario

#### » **2.1 Funciones de registro de compras y pedidos.**

- Formulario comprensible
- Indicar campos obligatorios
- Validar campos del formulario
- Ver estado actual de la compra o pedido.
- Funciones de registro de movimientos de madera.
- Formulario de registro comprensible
- Validar campos del formulario

#### » **3. Confiabilidad**

- No Deficiencia
- Errores de Enlaces
- Enlaces Rotos
- Enlaces Inválidos
- Enlaces no Implementados

#### » **4. Eficiencia**

- Accesibilidad de Información
- Visualización de la información solicitada de forma clara
- Eficiencia a la hora de realizar un registro
- Performance
- Páginas Rápidas

### C. Definición de Criterios Elementales y Procedimientos de Medición

En esta etapa se definirá una base de criterios para evaluación elemental, y realizar el proceso de medición y puntuación de los valores que se calcule para medir la calidad del software en tal caso se definirá los siguientes criterios:

**Tabla 2.2**

#### *Características de calidad*

<b>CARACTERISTICAS DE CALIDAD</b>	<b>MALA</b>	<b>REGULAR</b>	<b>BUENO</b>	<b>MUY BUENO</b>
<b>Funcionalidad</b>	0 – 30%	31 – 50%	51 – 90%	91 – 100%
<b>Confiabilidad</b>	0 – 30%	31 – 50%	51 – 90%	91 – 100%
<b>Usabilidad</b>	0 – 30%	31 – 50%	51 – 90%	91 – 100%
<b>Mantenibilidad</b>	0 – 30%	31 – 50%	51 – 90%	91 – 100%
<b>Portabilidad</b>	0 – 30%	31 – 50%	51 – 90%	91 – 100%
<b>TOTAL</b>	0 – 30%	31 – 50%	51 – 90%	91 – 100%

**Nota.** Esta tabla nos indica que nuestras características de calidad serán evaluados según al rango al que se encuentre.

### D. Definición de Estructuras de Agregación e Implementación de la Evaluación Global

Para esta fase se realiza un promedio de todos los parámetros encontrados para así poder sacar un resultado global y así el usuario pueda comprender de mejor manera la calidad del sistema.

#### **2.14.4.2.1. Análisis de Resultados y Recomendaciones**

En esta fase una vez calculado las características de calidad se darán las recomendaciones respectivas además de saber los resultados y ver si el sistema cumple con lo pronosticado.

### 2.14.4.3. Características de WEB - QEM

La metodología WEB - QEM toma las métricas del modelo de calidad ISO 9126-1 la cual da referencia a las siguientes características, funcionalidad, Confiabilidad, Usabilidad, Mantenibilidad y Portabilidad.

#### 2.14.4.3.1. Funcionalidad

Funcionalidad es una métrica orientada a la función del sistema y al proceso por el cual se desarrolla. Se centra en la funcionalidad o utilidad del programa.

- **Entradas del usuario**, se toma en cuenta cada entrada del usuario que el sistema proporciona a medida que ingresa al sistema.
- **Salidas del usuario**, se refleja las salidas que tiene el sistema tanto reportes como estadísticas que tiene el sistema.
- **Número de peticiones del usuario**, una petición se define como una entrada interactiva que resulta de la generación de algún tipo de respuesta en forma de salida.
- **Número de archivos**, se define cada archivo lógico.
- **Número de interfaces externas**, se definen todas aquellas interfaces legibles por el ordenador que solicitan transmitir información a otro sistema.

Para calcular los puntos función se usó las siguientes formulas:

$$PF = Cuenta\ Total * (confiabilidad\ proyecto + error\ min * \sum Fi) \text{ Ecuación 2.1}$$

$$Funcionalidad = \left( \frac{PF}{PF_{m\acute{a}ximo}} \right) \text{ Ecuación 2.2}$$

Dónde:

**PF:** Medida de funcionalidad.

**Máximo:** Medida de funcionalidad con su valor máximo.

**Cuenta Total:** Es la suma de los siguientes datos: Número de entradas, número de salidas, número de peticiones, número de archivos y número de interfaces externas.

**Confiabilidad proyectó:** Confiabilidad del proyecto, varía de 1% al 100% (0 a 1).

Error min: Error mínimo aceptable de complejidad.  $\Sigma Fi$ : Son los valores de ajuste de complejidad, donde ( $1 \leq i \leq 14$ ).

#### 2.14.4.3.2. Confiabilidad

Es la probabilidad de operación libre de fallos en un programa en un entorno determinado y durante un tiempo específico se toma en cuenta:

##### A) Enlaces

- Enlaces Rotos
- Enlaces Inválidos
- Enlaces no Implementados

##### B) Paginas

- Páginas Muertas
- Páginas bajo Construcción
- Errores de Ortografía

Para calcular los puntos función se usa la siguiente formula:

$$\text{Porcentaje de enlaces rotos} = \frac{CERI + CERE}{CTE} * 100 \text{ Ecuación 2.3}$$

Dónde:

**CERI:** Cantidad de enlaces rotos internos

**CERE:** Cantidad de enlaces rotos externos

**CTE:** Cantidad total de enlaces

Porcentaje de presencia de propiedad:

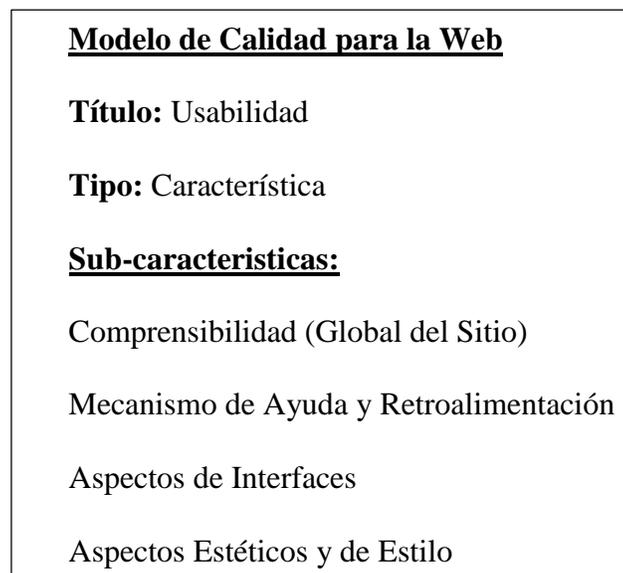
$$\text{Porcentaje presencia ALT} = \frac{\text{CantidadImágenesALT}}{\text{CantidadTotalImágenes}} * 100 \text{ Ecuación 2.4}$$

Y la confiabilidad se calcularía con la siguiente fórmula:

$$\text{Confiabilidad} = \text{PorcentajeEnlaces} + \text{PorcentajePresencialALT} \text{ Ecuación 2.5}$$

### 2.14.4.3.3. Usabilidad

Toma en cuenta la capacidad del software para ser comprendido, utilizado y atractivo para el usuario en determinadas condiciones.



**Figura 2.13. Características WEB - QEM Usabilidad**

Fuente: Elaboración propia

Se realiza una tabla que toma en cuenta los siguientes puntos:

#### A) Comprensión Global del Sitio

##### i. Esquema de Organización Global

- Tabla de Contenidos
- Mapa del Sitio

- Índices (Alfabéticos, Temáticos, Híbridos...)

ii. Visita Guiada (convencional y/o virtual)

iii. Mapa de Imagen

## **B) Aspectos de Interfaces y Estéticos**

i. Permanencia y Estabilidad en la Presentación de los Controles Principales

- Controles Directos
- Controles Indirectos
- Estabilidad

## **C) Mantenimiento del Color de Los Enlaces**

Se toma en cuenta el diseño que tiene el sistema.

### **2.14.4.3.4. Mantenibilidad**

La mantenibilidad es la facilidad con que una modificación puede ser realizada. Está indicada por los siguientes sub atributos.

- Facilidad de análisis
- Facilidad de cambio
- Estabilidad
- Facilidad de prueba

El índice de madurez del software se calcula con la siguiente formula:

$$IMS = \frac{[Mt - (Fa + Fb + Fc)]}{Mt} \quad \text{Ecuación 2.6}$$

Dónde:

**Mt:** Número de módulos en la versión actual

**Fa:** Número de módulos en la versión actual que se han cambiado

**Fb:** Número de módulos en la versión actual que se han añadido

**Fc:** Número de módulos en la versión anterior que se han borrado en la versión actual.

#### 2.14.4.3.5. Portabilidad

La portabilidad es la capacidad que tiene el sistema para ser trasladado de un entorno a otro.

Para poder medir la portabilidad del sistema usaremos la siguiente formula que indica el grado de portabilidad que tiene un software.

$$GP = 1 - \frac{ET}{ER} \text{ Ecuación 2.7}$$

Dónde:

**ET:** Es la medida de los recursos necesarios para llevar el sistema a otro entorno.

**ER:** Es la medida de los recursos necesarios para crear el sistema en el entorno residente.

Si **GP > 0**, la portabilidad es más rentable que el re-desarrollo

Si **GP = 1**, la portabilidad es perfecta

Si **GP < 0**, el re-desarrollo es más rentable que la portabilidad.

### 2.15. Modelo de Estimación de Costes

#### 2.15.1. COCOMO II

El Modelo Constructivo de Costos (o COCOMO, por su acrónimo del inglés COConstructive COst MOdel) (Boehm, 1981) afirma:

Es un modelo matemático de base empírica utilizado para estimación de costos de software.

Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Y tiene las siguientes características generales:

## Características generales

Pertenece a la categoría de modelos estimadores basados en estimaciones matemáticas. Está orientado a la magnitud del producto final, midiendo el "tamaño" del proyecto, en función de la cantidad de líneas de código, principalmente.

Se presentan tres niveles: básico, intermedio y detallado.

## Modelos de estimación

Las ecuaciones que se utilizan en los tres modelos son:

En Persona – Mes:

$$E = a(KI^{(b)*m(X)}) \text{ Ecuación 2.8}$$

En Meses:

$$Tdev = c(E^d) \text{ Ecuación 2.9}$$

En Personas:

$$P = \frac{E}{Tdev} \text{ Ecuación 2.10}$$

Donde:

**E:** es el esfuerzo requerido por el proyecto, en persona-mes

**Tdev:** es el tiempo requerido por el proyecto, en meses

**P:** es el número de personas requerido por el proyecto

**a, b, c y d** son constantes con valores definidos en una tabla, según cada submodelo

**KI:** es la cantidad de líneas de código, en miles.

**m(X):** Es un multiplicador que depende de 15 atributos.

A la vez, cada submodelo también se divide en modos que representan el tipo de proyecto, y puede ser:

- A. Modo orgánico:** un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).
- B. Modo semilibre o semiencajado:** corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- C. Modo rígido o empotrado:** el proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

### Modelo básico

Se utiliza para obtener una primera aproximación rápida del esfuerzo,  $2$  y hace uso de la siguiente tabla de constantes para calcular distintos aspectos de costes:

**Tabla 2.3**

*Calcular distintos aspectos de costes*

<b>MODO</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>Orgánico</b>	2.40	1.05	2.50	0.38
<b>Semi – Orgánico</b>	3.00	1.12	2.50	0.35
<b>Empotrado</b>	3.60	1.20	2.50	0.33

**Nota.** Distintos aspectos de coste de COCOMO. (Boehm 1981)

Estos valores son para las fórmulas:

Personas necesarias por mes para llevar adelante el proyecto:

$$MM = a * (KI^b) \text{ Ecuación 2.11}$$

Tiempo de desarrollo del proyecto:

$$TDEV = c * (MM^d) \text{ Ecuación 2.12}$$

Personas necesarias para realizar el proyecto:

$$CosteH = \frac{MM}{TDEV} \text{ Ecuación 2.13}$$

Costo total del proyecto:

$$CosteM = CosteH * \text{Salario medio entre los programadores y analistas}$$

#### **Ecuación 2.14**

Se puede observar que a medida que aumenta la complejidad del proyecto (modo), las constantes aumentan de 2.4 a 3.6, que corresponde a un incremento del esfuerzo del personal. Hay que utilizar con mucho cuidado el modelo básico puesto que se obvian muchas características del entorno

#### **Modelo intermedio**

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

Para este ajuste, al resultado de la fórmula general se lo multiplica por el coeficiente surgido de aplicar los atributos que se decidan utilizar.

Los valores de las constantes a reemplazar en la fórmula son:

**Tabla 2.4**

#### ***Modelo intermedio***

<b>MODO</b>	<b>A</b>	<b>B</b>
<b>Orgánico</b>	3.20	1.05
<b>Semi – Orgánico</b>	3.00	1.12
<b>Empotrado</b>	2.80	1.20

**Nota.** Modelo interno de COCOMO. (Boehm 1981)

Se puede observar que los exponentes son los mismos que los del modelo básico, confirmando el papel que representa el tamaño; mientras que los coeficientes de los modos orgánico y rígido han cambiado, para mantener el equilibrio alrededor del semilibre con respecto al efecto multiplicador de los atributos de coste.

### **Atributos**

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy baja - bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula El significado de los atributos es el siguiente, según su tipo:

- **De software**

**RELY:** garantía de funcionamiento requerida al software. Indica las posibles consecuencias para el usuario en el caso que existan defectos en el producto. Va desde la sola inconveniencia de corregir un fallo (muy bajo) hasta la posible pérdida de vidas humanas, extremadamente alto, software de alta criticidad.

**DATA:** tamaño de la base de datos en relación con el tamaño del programa. El valor del modificador se define por la relación:  $D/K$ , donde D corresponde al tamaño de la base de datos en bytes y K es el tamaño del programa en cantidad de líneas de código.

**CPLX:** representa la complejidad del producto.

- **De hardware**

**TIME:** limitaciones en el porcentaje del uso de la CPU.

**STOR:** limitaciones en el porcentaje del uso de la memoria.

**VIRT:** volatilidad de la máquina virtual.

**TURN:** tiempo de respuesta requerido.

- **De personal**

**ACAP:** calificación de los analistas.

**AEXP:** experiencia del personal en aplicaciones similares.

**PCAP:** calificación de los programadores.

**VEXP:** experiencia del personal en la máquina virtual.

**LEXP:** experiencia en el lenguaje de programación a usar.

- **De proyecto**

**MODP:** uso de prácticas modernas de programación.

**TOOL:** uso de herramientas de desarrollo de software.

**SCED:** limitaciones en el cumplimiento de la planificación.

## **Modelo Detallado**

Presenta principalmente dos mejoras respecto al anterior:

Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, etc., tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra. (Gómez, López, Migani, & Otazú, 2020)

Establece una jerarquía de tres niveles de productos, de forma que los aspectos que representan gran variación a bajo nivel, se consideran a nivel módulo, los que representan pocas variaciones, a nivel de subsistema; y los restantes son considerados a nivel sistema.

## **2.16. Herramientas**

### **2.16.1. PHP**

Es un lenguaje de programación de texto plano en UTF-8 de código abierto, adecuado para el desarrollo web y que se puede introducir HTML, esto quiere decir que podemos combinar

código PHP con código HTML, PHP nos ayuda a generar páginas web dinámicas, se refiere que el contenido no es el mismo.

PHP es un preprocesador de hipertexto, es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el preprocesado de texto plano en UTF-8. Posteriormente se aplicó al desarrollo web de contenido dinámico, dando un paso evolutivo en el concepto de aplicación en línea, por su carácter de servicio.

Su implementación en los documentos HTML era aparentemente muy sencilla. Hay que decir, que, PHP no genera HTML, sino que ofrece una salida de texto con codificación UTF-8 compatible con los documentos HTML. El programador puede dotar a la salida de los tag's propios del HTML y los exploradores más comunes para navegar por internet, reconocerán muy rápidamente el formato UTF-8 y lo adaptarán ofreciendo una salida entendible.

Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el texto plano en formato UTF-8, ampliamente reconocido por el estándar HTML, dando como resultado, en los exploradores, una salida al usuario perfectamente entendible.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

(Rees, 2016)

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del

software libre publicado bajo la licencia PHPv3\_01, una licencia Open Source validada por Open Source Initiative. La licencia de PHP es del estilo de licencias BSD, sin la condición de copyleft asociada con la Licencia Pública General de GNU.

### **2.16.2. Laravel**

Es un framework de código abierto que asimila fácilmente PHP, nos ayuda a desarrollar aplicaciones y servicios web de forma elegante y simple, nos permite desarrollar aplicaciones pequeñas, grandes y muy grandes también facilita la parte de mantenimiento y escalabilidad de nuestros proyectos.

Laravel es un marco de aplicación web con una sintaxis elegante y expresiva. Ya hemos sentado las bases, permitiéndole crear sin preocuparse por las pequeñas cosas. Creemos que el desarrollo debe ser una experiencia divertida y creativa para ser verdaderamente satisfactorio. Laravel intenta aliviar el dolor del desarrollo al facilitar las tareas comunes que se utilizan en la mayoría de los proyectos web. (Laravel, 2020)

### **2.16.3. QGIS**

Es un Sistema de Información Geográfica (SIG) de software libre para plataformas GNU/Linux, Unix, Mac OS, Microsoft Windows y Android. Era uno de los primeros ocho proyectos de la Fundación OSGeo y en 2008 oficialmente graduó de la fase de incubación. Permite manejar formatos Ráster y vectoriales a través de la biblioteca GDAL (GADL/OGR), así como bases de datos. Algunas de sus características son:

- Soporte para la extensión espacial de PostgreSQL, PostGIS.
- Manejo de archivos vectoriales Shapefile, ArcInfo coverages, MapInfo, GRASS GIS, DXF, DWG, etc.

- Soporte para un importante número de tipos de archivos Ráster (GRASS GIS, GeoTIFF, TIFF, JPG, etc.)

Una de sus mayores ventajas es la posibilidad de usar Quantum GIS como GUI del SIG GRASS, utilizando toda la potencia de análisis de este último en un entorno de trabajo más amigable. QGIS está desarrollado en C++, usando la biblioteca Qt para su Interfaz gráfica de usuario.

Una de las grandes fortalezas de QGIS es que trabaja en cualquiera de los sistemas operativos:

- GNU/Linux
- BSD
- Unix
- MacOS
- Windows
- Android
- funcionando de manera similar en todos ellos.

“QGIS es un software libre y opera bajo la licencia GNU GPL. El software QGIS puede ser modificado libremente de tal manera que pueda realizar diferentes y más especializadas funcionalidades” (QGIS, 2020). Ya existen dos nuevos productos denominados: QGIS Browser y QGIS Server. Estos productos poseen diferentes interfaces del usuario (front-end).

#### **2.16.4. Open Layer**

Es una biblioteca de JavaScript de código abierto bajo una derivación de la licencia BSD para mostrar mapas interactivos en los navegadores web. OpenLayers ofrece un API para acceder a diferentes fuentes de información cartográfica en la red: Web Map Services, Mapas comerciales (tipo Google Maps, Bing, Yahoo), Web Features Services, distintos formatos vectoriales, mapas de OpenStreetMap, etc.

Inicialmente fue desarrollado por MetaCarta en junio de 2006. Desde el noviembre del 2007 este proyecto forma parte de los proyectos de Open Source Geospatial Foundation. Actualmente el desarrollo y el soporte corren a cargo de la comunidad de colaboradores. “OpenLayers facilita poner un mapa dinámico en cualquier página web. Puede mostrar mosaicos de mapas, datos vectoriales y marcadores cargados desde cualquier fuente. OpenLayers ha sido desarrollado para promover el uso de información geográfica de todo tipo” (Open Layers, 2018). Es completamente gratuito, Open Source JavaScript, lanzado bajo la Licencia BSD de 2 cláusulas (también conocida como FreeBSD).

#### **2.16.5. Leaflet**

Es una biblioteca JavaScript de código abierto ampliamente utilizada para crear aplicaciones de mapeo web. Lanzado por primera vez en 2011, es compatible con la mayoría de las plataformas móviles y de escritorio, compatible con HTML5 y CSS3. Junto con OpenLayers y la API de Google Maps, es una de las bibliotecas de mapeo de JavaScript más populares y es utilizada por los principales sitios web como FourSquare, Pinterest y Flickr.

Leaflet permite a los desarrolladores sin experiencia en SIG visualizar mapas web en mosaico alojados en un servidor público, con superposiciones en mosaico opcionales. Puede cargar datos de características de archivos GeoJSON , diseñarlos y crear capas interactivas, como marcadores con ventanas emergentes cuando se hace clic.

### **Características**

Leaflet admite capas del Servicio de mapas web (WMS), capas GeoJSON, capas vectoriales y capas de mosaico de forma nativa. Muchos otros tipos de capas son compatibles a través de complementos.

Al igual que otras bibliotecas de mapas web, el modelo de visualización básico implementado por Leaflet es un mapa base, más cero o más superposiciones translúcidas, con cero o más objetos vectoriales mostrados en la parte superior.

## **Elementos**

Los principales tipos de objetos de Leaflet son:

- Tipos de Ráster (TileLayer e ImageOverlay)
- Tipos de Vectores (ruta, polígono y tipos específicos como círculo)
- Tipos agrupados (LayerGroup, FeatureGroup y GeoJSON)
- Controles (Zoom, Capas, etc.)

También hay una variedad de clases de utilidad, como interfaces para gestionar proyecciones, transformaciones e interactuar con el DOM. (Geosysnet, 2017)

### **2.16.6. Google Earth Engine**

Una de las últimas aplicaciones lanzadas por la empresa Google es la denominada Google Earth Engine, la cual es una plataforma para el análisis científico a escala petabyte (PB) y la visualización de conjuntos de datos geospaciales, tanto para el beneficio público como para los usuarios comerciales y de la Administración. La principal diferencia con la aplicación Google Earth es la capacidad de análisis de los datos.

El archivo público de datos incluye imágenes históricas de la tierra que se remontan a más de cuarenta años, y se recopilan nuevas imágenes todos los días. Earth Engine también proporciona

APIs en JavaScript y Python, así como otras herramientas, para permitir el análisis de grandes conjuntos de datos. (López, 2018)

Earth Engine almacena imágenes satelitales, las organiza y las pone a disposición por primera vez para la extracción de datos a escala global.

### **2.16.7. PostgreSQL**

Es un sistema de gestión de base de datos relacionales, es orientada a objetos, multisistema que se refiere a su funcionamiento en sistemas operativos como, Windows, Linux, Mac Os, también es extensible, escalable y funciona bajo licencia libre y tiene la capacidad de manejar base de datos de hasta 90 Mgbytes.

PostgreSQL es un avanzado sistema de bases de datos relacionales basado en Open Source. Esto quiere decir que el código fuente del programa está disponible a cualquier persona libre de cargos directos, permitiendo a cualquiera colaborar con el desarrollo del proyecto o modificar el sistema para ajustarlo a sus necesidades. PostgreSQL está bajo licencia BSD.

Un sistema de base de datos relacionales es un sistema que permite la manipulación de acuerdo con las reglas del algebra relacional. Los datos se almacenan en tablas de columnas y renglones. Con el uso de llaves, esas tablas se pueden relacionar unas con otras.

#### **2.16.7.1. Ideas Básicas Acerca del Funcionamiento**

En la jerga de bases de datos, PostgreSQL usa el modelo cliente/servidor. Una sesión en PostgreSQL consiste en ejecución de los siguientes procesos.

- El servidor, que maneja archivos de bases de datos, acepta conexiones a las aplicaciones cliente, y realiza acciones en la base de datos. El programa servidor de bases de datos se conoce como postmaster.

- La aplicación cliente, que necesita realizar operaciones en la base de datos. Las aplicaciones cliente pueden ser de la más diversa naturaleza: pueden ser aplicaciones de texto en una consola, aplicaciones gráficas, un servidor web que accede a la base de datos para mostrar una página, o herramientas especializadas de mantenimiento de bases de datos. (Denzer, 2002, pág. 2)

Como es habitual en las aplicaciones cliente/servidor, el cliente y el servidor pueden estar en diferentes máquinas. En este caso, estos se comunican sobre una conexión de red TCP/IP.

#### **2.16.8. PostGIS**

Es el complemento y solución de almacenamiento de datos espaciales de PostgreSQL, potente en el ámbito de GIS Open Source, nos permite trabajar con datos de tipo geométricos, soporta archivos de tipo Ráster y Vectorial, tiene funciones de análisis, transformaciones y consultas espaciales.

Es un módulo que añade soporte de objetos geográficos a la base de datos objeto-relacional PostgreSQL, convirtiéndola en una base de datos espacial para su utilización en Sistema de Información Geográfica. Se publica bajo la Licencia Pública General de GNU.

Durante muchos años, éstos han sido la forma de almacenar y compartir información geográfica, sin embargo, la ausencia de topología, la falta de integridad referencial en los campos y la carencia de mecanismos que aseguren la calidad en la propia información, entre otros, ha provocado que las Bases de Datos Espaciales se conviertan en la opción más acertada a la hora de trabajar con un modelo de datos cartográfico. (Martinez Llario, 2013, pág. 25)

Las Bases de Datos Espaciales permiten gestionar y almacenar la cartografía de una manera mucho más eficiente que los modelos tradicionales como por ejemplo los shapes.

### **2.16.9. Laragon**

Laragon es una opción relativamente nueva para la creación de lo que llamamos el entorno de desarrollo, es decir, todo un conjunto de programas necesarios para desarrollar aplicaciones. Sirve para trabajar con PHP, pero también con otros lenguajes del lado del servidor, como Node, Python o Ruby. (Laragon, 2018)

Es una herramienta de desarrollo local, la cual nos facilita mucho la actualización y es flexible al momento de cambiar sus dependencias, también usa menos de 4MB de ram y es un servidor gratuito, nos permite la creación de Virtualhost, laragon ofrece un stack de programas necesarios para trabajar con: Apache, PHP, MySQL y MariaDB.



**CAPÍTULO III**  
**MARCO APLICATIVO**



### 3.1. Introducción

En este capítulo se describe la utilización de los Sistemas de Información Geográfica, como una herramienta para el monitoreo de riesgos de quema con imágenes de satélite. Se mostrara el diseño y creación de la base de datos espacial con una interface sencilla, para la actualización y administración de la información recopilada. Se desarrollara y aplicara la metodología UWE para este proyecto, también se mostrara la utilización del mapa Ráster y los mapas Vectoriales y la generación de dichos mapas geográficos. También se define el programa QGIS que actualizará y administrará la información geográfica y la base de datos de las evaluaciones.

### 3.2. Estructura o esquema del sistema

Se presenta un esquema que muestra como está estructurada la funcionalidad del SISTEMA DE INFORMACION GEOGRAFICA APLICADO AL MONITOREO DE RIESGOS DE QUEMA CON IMÁGENES DE SATELITE, se muestra como el sistema interactúa e intercambia información, a continuación se observa la estructura o esquema del sistema.



**Figura 3.1. Contenido del sistema.**

Fuente: Elaboración propia.



UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito.

### 3.3.1. Análisis de requerimiento

Uno de los procedimientos más importantes, en el proceso de desarrollo del software es determinar que necesidades del cliente debe cubrir el sistema. El análisis de requerimiento de desarrollo de software implica adquirir un entendimiento de las necesidades del cliente, en el contexto de ingeniería de software existen técnicas que facilitan la captura de los requisitos de un sistema. Para esto se fijara los requisitos funcionales del sistema, para reflejarlos en un modelo de casos de uso. Y con UWE se utiliza notaciones y diagramas UML para el análisis y diseño de aplicaciones web.

A continuación se muestra la lista del análisis de requerimiento para el Sistema de información Geográfica aplicada al monitoreo de riesgo de quema con imágenes de satélite.

**Tabla 3.1**

***Requerimientos Funcionales***

<b>COD.</b>	<b>REQUERIMIENTOS</b>	<b>CATEGORIA</b>
R1	Análisis sobre el grado de riesgo de quema.	OCULTO
R2	Importación de la información geográfica en la base de datos espacial (PostgreSQL - PostGis).	OCULTO
R3	Visualización del visor geográfico con imágenes de satélite.	EVIDENTE
R4	Consultas dentro de la Base de Datos Espaciales.	OCULTO
R5	Generar reportes sobre el riesgo de quema	EVIDENTE

**Nota.** Elaboración propia

### **3.3.2. Modelo de Casos de Uso**

El modelo de casos uso está conformado por dos elementos de modelado principales, llamados casos de uso y actores (Koch, 2000). Este modelo describe un trozo de comportamiento de la aplicación sin revelar su estructura interna. A continuación se describe las partes del modelo de casos de uso:

- Un caso de uso es una unidad coherente de funcionalidad provista de aplicaciones que interactúan con uno o más actores externos de la aplicación.
- Un actor, es el rol que un usuario puede desempeñar con respecto a un sistema o una entidad, tales como otro sistema o una base de datos.
- Relaciones entre actores, existen relaciones entre estos dos elementos, tales como las asociaciones entre actores y casos de uso, y las dependencias «includes» y «extends» entre casos de uso.

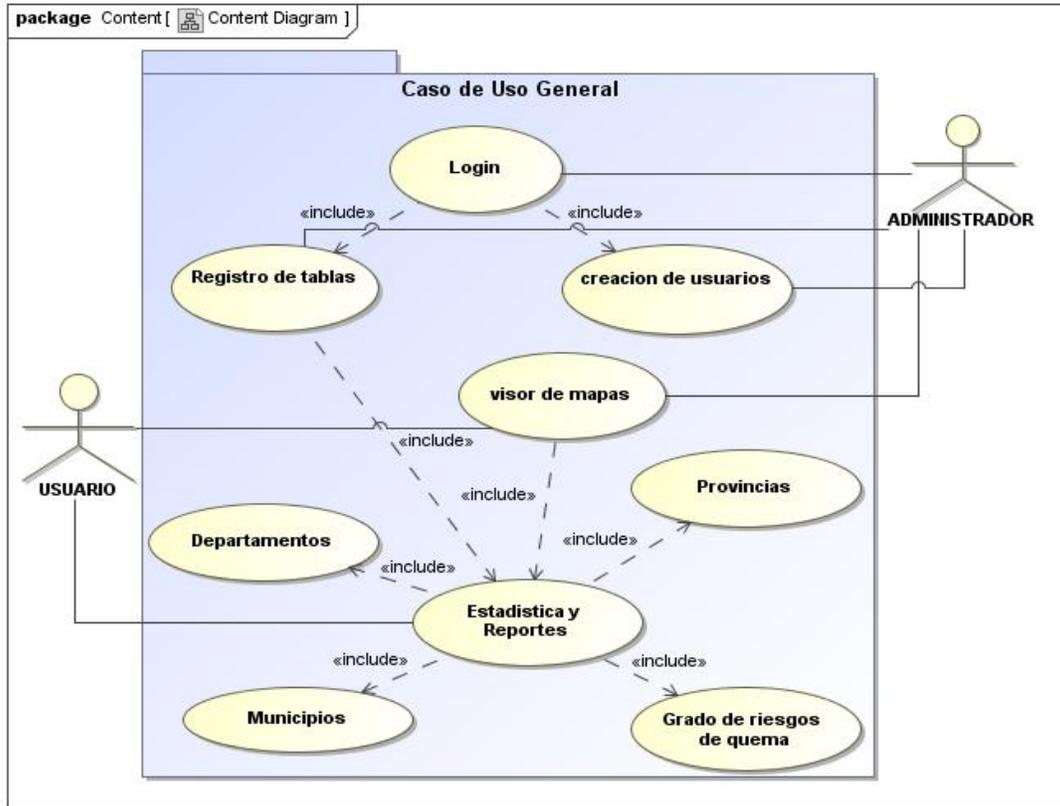
Todos estos elementos así como paquetes y vista se representan gráficamente con la notación de UML.

Para aplicar la metodología UWE, se utilizó la herramienta MagicDraw, que permite representar visualmente y fácil los casos de uso, diagrama conceptual, diagrama de navegación y diagrama de presentación.

#### **3.3.2.1. Descripción de los actores**

Los actores representan a los usuarios que presenta el sistema. Se comprende como usuario cualquier persona que llegue a interactuar con el sistema.

A continuación se describe a los actores que interactúan con el sistema que se desarrolla.



**Figura 3.3. Análisis sobre el grado de riesgo de quema**

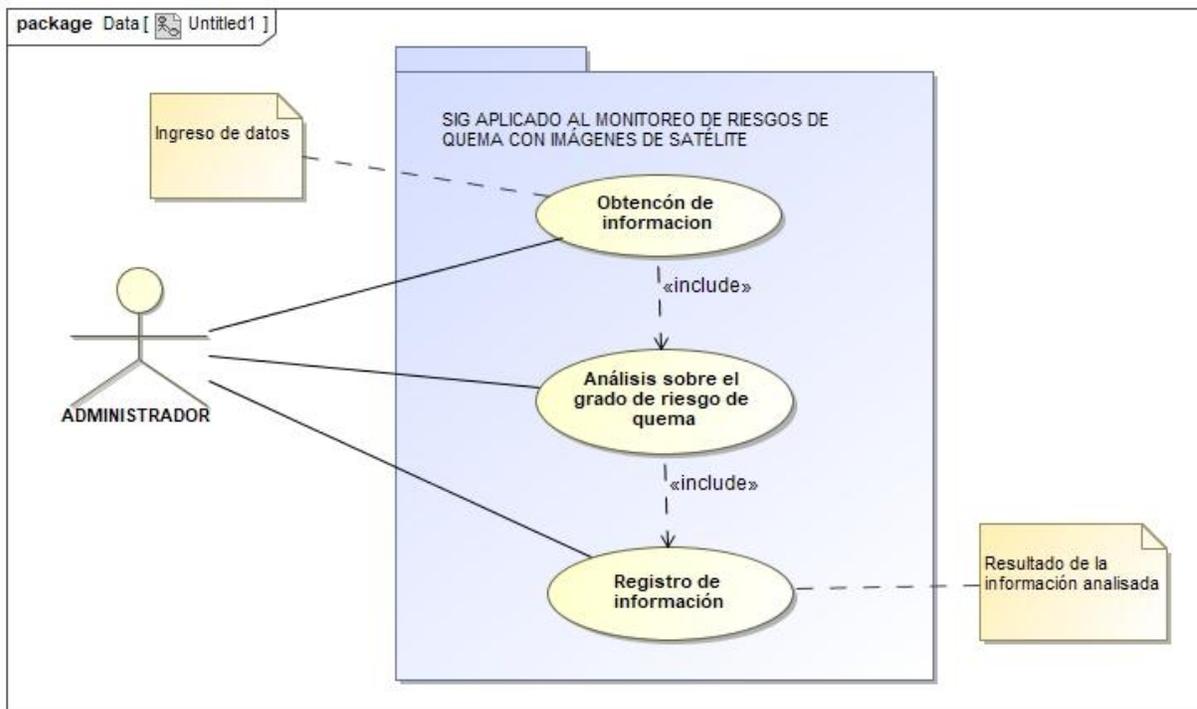
Fuente: Elaboración propia.

**Tabla 3.2**

*Descripción de Actores del sistema*

ACTORES	DESCRIPCIÓN
Administrador	Es el actor con más privilegios, es el usuario que registra las tablas en el sistema las cuales contienen información georeferenciada.
Usuario	Es la persona singular o jurídica que ingresa al sistema para obtener información, también obtiene datos sobre las áreas de riesgo de quema, el sistema tiene registrado los datos de las áreas de riesgo de quema para poder tener un mejor control.

**Nota.** Elaboración propia



**Figura 3.4. Análisis sobre el grado de riesgo de quema**

Fuente: Elaboración propia.

A continuación se describe detalladamente los casos de uso:

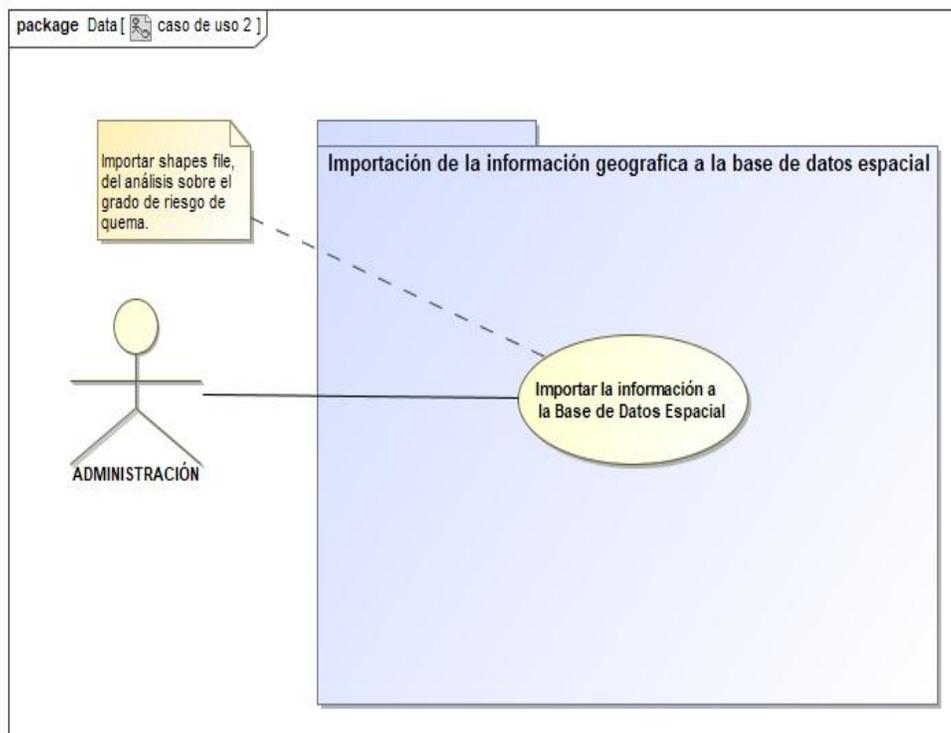
**Tabla 3.3**

**Descripción de Caso de uso: Análisis sobre el grado de riesgo de quema**

CASO DE USO ANÁLISIS SOBRE EL GRADO DE RIESGO DE QUEMA		
Nombre:	Análisis sobre el grado de riesgo de quema	
Actor:	Administrador	
Descripción:	Describe el análisis sobre el grado de riesgo de quema	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Realiza el conteo de pixeles de la imagen Shape.	1. Almacena la información obtenida.
	2. Realiza el cálculo del valor de riesgo de quema.	

CASO DE USO REGISTRO DE LA INFORMACIÓN		
Nombre:	Registro de la información	
Actor:	Administrador	
Descripción:	Describe el registro de la información	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Verifica la información obtenida.	1. Muestra la información almacenada.
	2. Realiza el registro.	

**Nota.** Elaboración propia



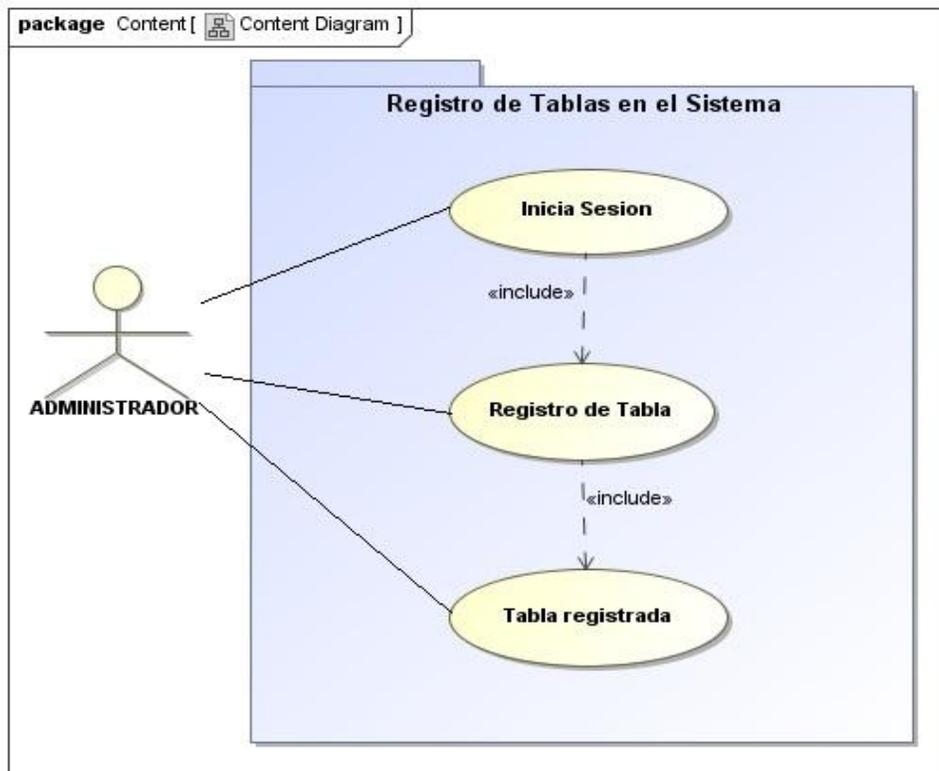
**Figura 3.5. Importación de la información geográfica a la Base de Datos Espacial.**  
Fuente: Elaboración propia.

**Tabla 3.4**

*Descripción de Caso de Uso: Importación de la información geográfica a la Base de Datos Espacial*

CASO DE USO IMPORTACIÓN DE INFORMACIÓN GEOGRÁFICA A LA BASE DE DATOS ESPACIAL		
Nombre:	Importación de la información geográfica a la base de datos espacial	
Actor:	Administrador	
Descripción:	Describe la importación y almacenamiento de un archivo shape a la Base de Datos Espacial.	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Verificar la información de los archivos Shapes.	1. Muestra la información almacenada de los archivos Shapes.
	2. Almacena los archivos Shapes.	

**Nota.** Elaboración propia.



**Figura 3.6. Registro de Tablas en el Sistema.**

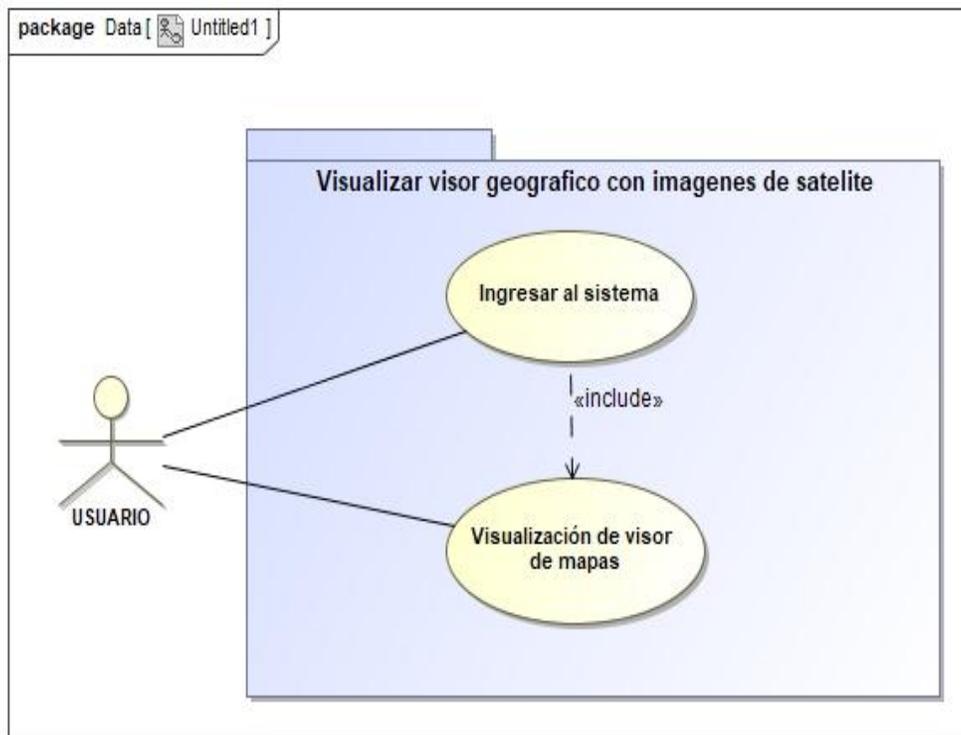
Fuente: Elaboración propia.

**Tabla 3.5****Descripción de Caso de Uso: Inicio de Sesión**

<b>CASO DE USO INICIO DE SESIÓN</b>		
Nombre:	Inicio de Sesión	
Actor:	Administrador	
Descripción:	Describe el inicio de sesión en el sistema.	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Introduce el correo electrónico del usuario.	1. Muestra la pantalla de registro de tablas.
	2. Introduce la contraseña.	
	3. Acceder	
<b>CASO DE USO REGISTRO DE TABLAS</b>		
Nombre:	Registro de tablas	
Actor:	Administrador	
Descripción:	Describe el registro de tablas en el sistema.	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Llena el campo de nombre de tabla.	1. Registra el nombre de la tabla.
	2. Presiona registrar.	2. Muestra una tabla donde se aprecia la tabla registrada.
<b>CASO DE USO TABLA REGISTRADA</b>		
Nombre::	Tabla registrada	
Actor	Administrador	
Descripción:	Describe la tabla registrada.	

Flujo principal:	Evento Actor	Evento Sistema
	1. Aprecia el registro de la tabla.	1. Muestra una tabla con el registro, donde existe las columnas de (nombre de tabla, fecha de registro).

**Nota.** Elaboración propia.



**Figura 3.7.** Visualización del visor geográfico con imágenes de satélite.

Fuente: Elaboración propia.

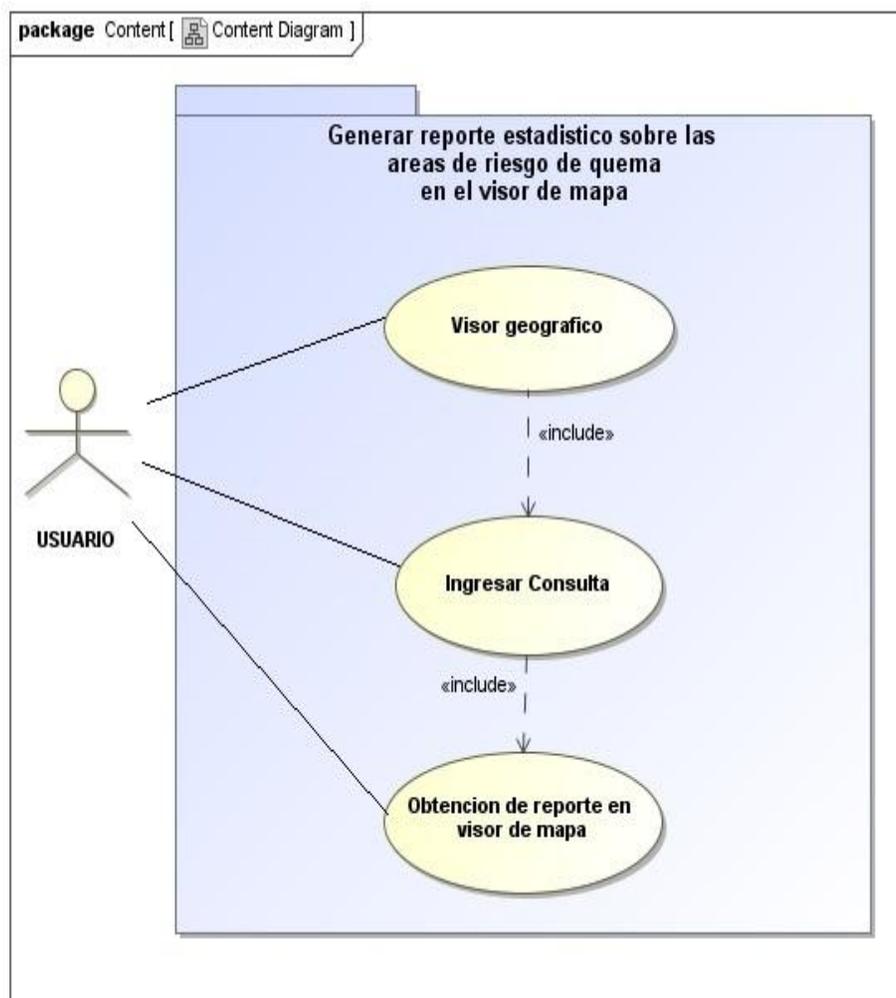
**Tabla 3.6**

*Descripción de Caso de Uso: Visualización del visor geográfico con imágenes de satélite*

CASO DE USO VISUALIZACIÓN DE VISOR DE MAPAS	
Nombre:	Visualización de visor de mapas
Actor:	Usuario

Descripción:	Describe la visualización de visor de mapas.	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Ingresa al sistema y se va al menú de “Geovisor”	1. Muestra el visor de mapas.
	2. Hace un click en el menú “Geovisor”.	

**Nota.** Elaboración propia.



**Figura 3.8.** Generar reportes estadísticos sobre el riesgo de quema en el visor de mapa.

Fuente: Elaboración propia.

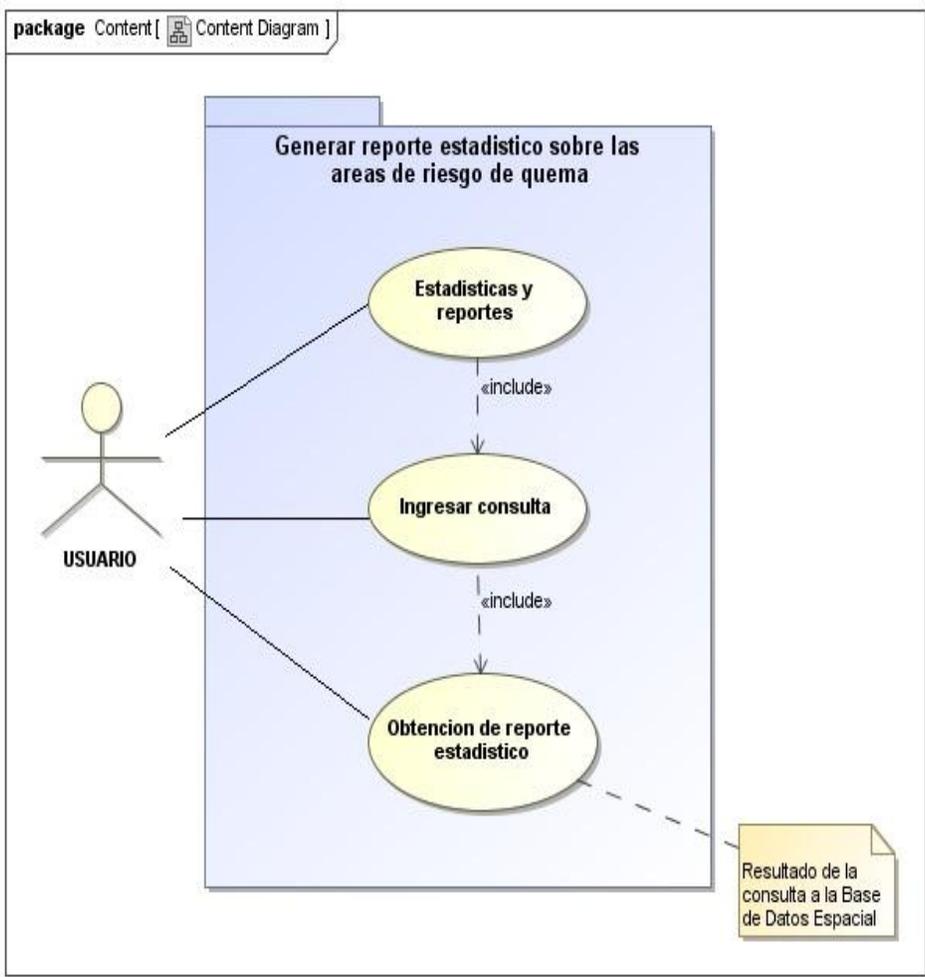
**Tabla 3.7**

*Descripción de Caso de Uso: Generar reportes estadísticos sobre el riesgo de quema*

CASO DE USO VISOR GEOGRÁFICO		
Nombre:	Visor geográfico	
Actor:	Usuario	
Descripción:	Describe el visor geográfico.	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Ingresar al sistema y se va al menú de “Geovisor”	1. Muestra el visor de mapas con la imagen satelital.
	2. Hace un click en el menú “Geovisor”.	
CASO DE USO INGRESAR CONSULTA		
Nombre:	Ingresar consulta	
Actor:	Usuario	
Descripción:	Describe el ingreso de consulta.	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Realiza una búsqueda de un departamento, provincia o municipio.	1. Muestra el nombre del departamento, provincia o municipio elegido.
	2. Pone en el filtro el dato elegido.	2. Muestra el grado de riesgo de quema.
CASO DE USO OBTENCION DE REPORTE EN VISOR DE MAPAS		
Nombre:	Obtención de reporte en visor de mapas	
Actor:	Usuario	

Descripción:	Describe la obtención de reporte de riesgo de quema de departamento, provincia y municipio en el visor de mapa.	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Elige en el buscador o mapa el departamento, provincia o municipio que desea saber la información sobre el riesgo de quema.	1. Muestra el reporte estadístico sobre el riesgo de quema del lugar geográfico elegido.
	2. Hace click o buscar.	

**Nota.** Elaboración propia.



**Figura 3.9. Generar reportes estadísticos sobre el riesgo de quema.**  
Fuente: Elaboración propia.

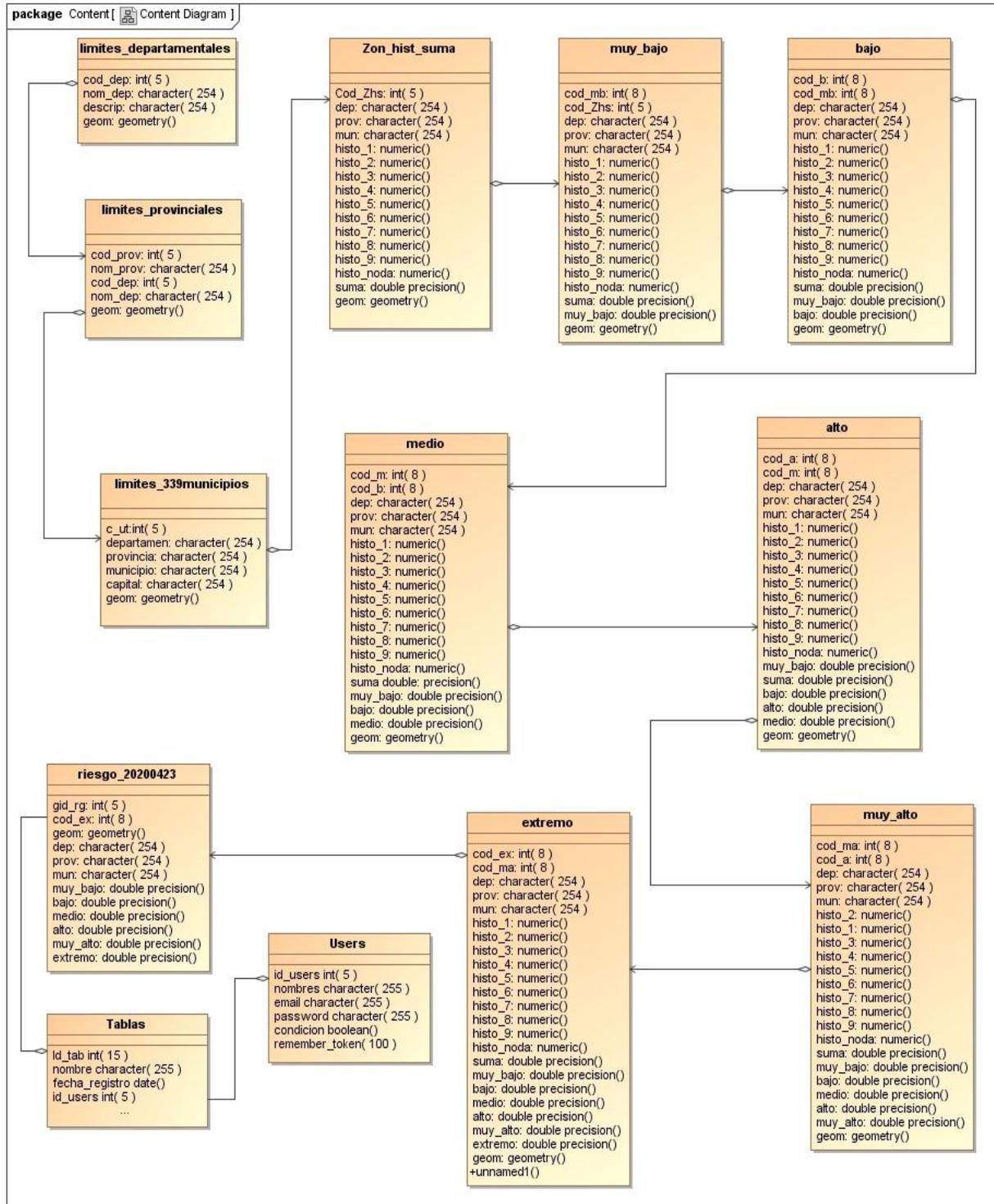
**Tabla 3.8****Descripción de Caso de uso: Ingresar consulta**

<b>CASO DE USO INGRESAR CONSULTA</b>		
Nombre:	Ingresar consulta	
Actor:	Usuario	
Descripción:	Describe el ingreso de consulta.	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Realiza la consulta en los campos.	1. Realiza el pedido de consulta a la Base de Datos Espacial.
	2. Hace click en mostrar.	
<b>CASO DE USO OBTENCION DE REPORTE ESTADISTICO</b>		
Nombre:	Obtención de reporte estadístico	
Actor:	Usuario	
Descripción:	Describe la obtención de reporte estadístico.	
Flujo principal:	<b>Eventos Actor</b>	<b>Eventos Sistema</b>
	1. Realiza la consulta.	1. Muestra en pantalla la petición de los datos de riesgo de quema.

**Nota.** Elaboración propia

**3.3.3. Modelo Conceptual (Base de Datos)**

Un Modelo Conceptual es una descripción de alto nivel de la estructura de la base de datos, en la cual la información es representada a través de Objetos, Relaciones y Atributos. El objetivo de este modelo es proporcionar una especificación visual de la información por medio de diagrama de clases.

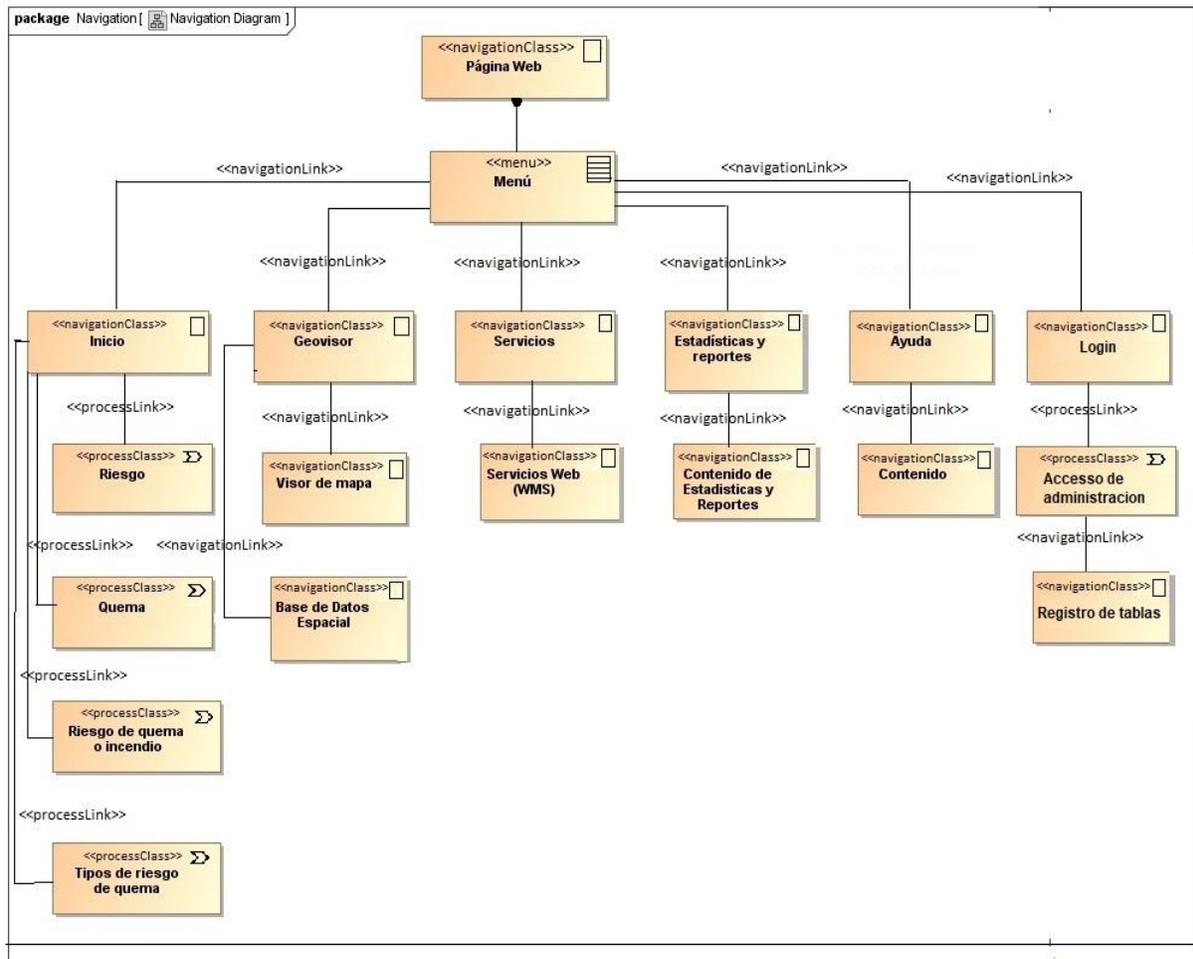


**Figura 3.10. Modelo conceptual.**

Fuente: Elaboración propia.

### 3.3.4. Modelo de Navegación

Un modelo de navegación, es el encargado de definir cómo se le brindará a cada usuario del sistema el acceso a la información y la funcionalidad que le es relevante para llevar a cabo su labor y qué secuencias de caminos deberán seguir para conseguirlo.

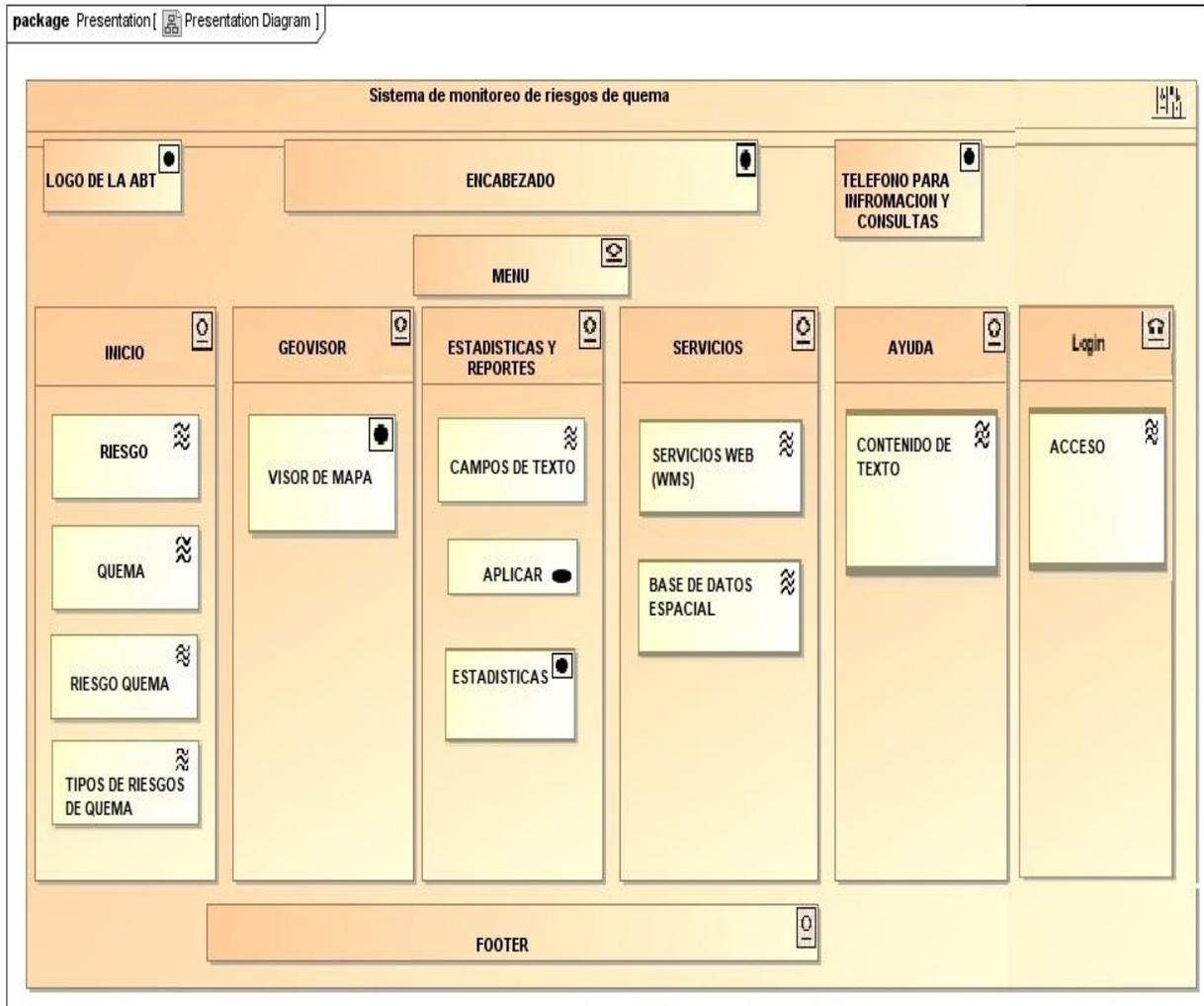


**Figura 3.11. Modelo de navegación.**

Fuente: Elaboración propia.

### 3.3.5. Modelo de Presentación

El modelo de presentación proporciona una vista abstracta de la interfaz de usuario (UI) de la aplicación web. Se basa en el modelo de navegación y describe qué elementos (por ejemplo texto, elementos, links, formularios) se utilizarán para presentar los nodos de navegación.



**Figura 3.12. Modelo de presentación.**

Fuente: Elaboración propia.

### 3.3.6. Codificación del Software

Esta fase consiste en la transformación en código fuente, en el lenguaje de programación escogido, que es Php y Html, también tomando en cuenta los requerimientos obtenidos y el diseño funcional planteado anteriormente.

A continuación se muestra pantallas del sistema web, y las pantallas de operaciones que se realizan en el sistema.

- **Ventana Principal:** Un usuario al ingresar al link de la página web, tiene esta pantalla, donde se muestra la estructura de la página.



**Figura 3.13. Ventana Principal**  
Fuente: Elaboración Propia

- **Box de Conceptos**

En la siguiente imagen se muestra, conceptos que al usuario le pueden ser útiles, para un mejor entendimiento de la página.

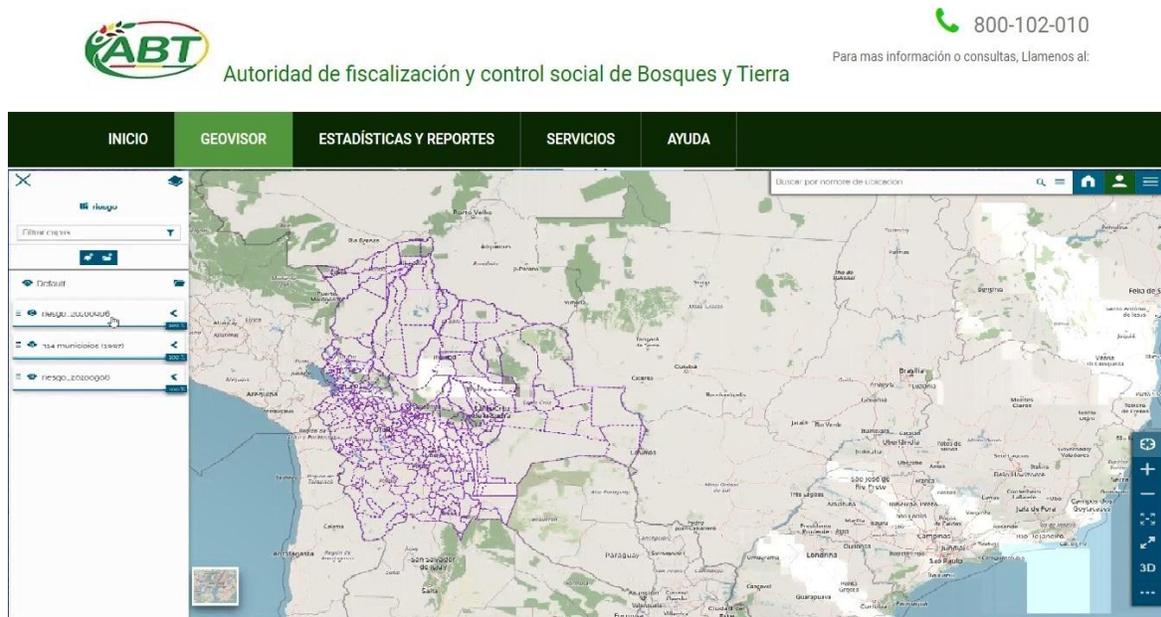
- Riesgo
- Quema
- Riesgo de quema
- Tipos de riesgo de quema



**Figura 3.14. Box de conceptos**  
Fuente: Elaboración Propia

- **Ventana del Geovisor**

En la siguiente imagen se puede observar el visor de mapa, donde el usuario tiene acceso a los datos obtenidos sobre las áreas de riesgos de quema.



**Figura 3.15. Ventana del Geovisor**  
Fuente: Elaboración Propia

- **Ventana de Estadísticas y Reportes**

En la siguiente imagen se aprecia los campos donde el usuario puede consultar reportes de las áreas de riesgos de quema, las cuales son por: Departamento, Provincia y Municipio.

ESTADÍSTICAS Y REPORTES

Seleccione Fecha de Riesgo de Quema  
risgo\_20200423

Seleccione la tabla de riesgo  
risgo

Seleccione el Departamento  
BENI

Seleccione el Departamento riesgo

Por favor seleccione un Departamento

MOSTRAR

**Figura 3.16. Ventana de Estadísticas y Reportes**

Fuente: Elaboración Propia

- **Ventana de Servicios**

El usuario podrá tener acceso a los servicios utilizados, para la obtención de datos de las áreas de riesgos de quema, también podrá descargar los mismos.

ABT  
Autoridad de fiscalización y control social de Bosques y Tierra

INICIO GEOVISOR ESTADÍSTICAS Y REPORTES **SERVICIOS** AYUDA

**SERVICIOS WEB(WMS)**

- ✓ Imagen raster o satelital de calor
- ✓ Limite Provincial
- ✓ Limite Nacional
- ✓ Limite Municipal

BASE DE DATOS ESPACIAL

PARA MAS INFORMACIÓN O CONSULTAS, LLAMAR AL:  
800-102-010

**Figura 3.17. Ventana de Servicios**

Fuente: Elaboración Propia

- **Ventana de Ayuda**

En la siguiente imagen se aprecia una guía y ayuda de la página, para que el usuario no tenga problemas al momento de navegar por la página.



**Figura 3.18. Ventana de Ayuda**

Fuente: Elaboración Propia

### **3.4. Implementación del sistema**

A lo largo de este capítulo se explica detenidamente cada uno de los servicios creados. El flujo de trabajo es el que se muestra en la figura 3.10, de elaboración propia.

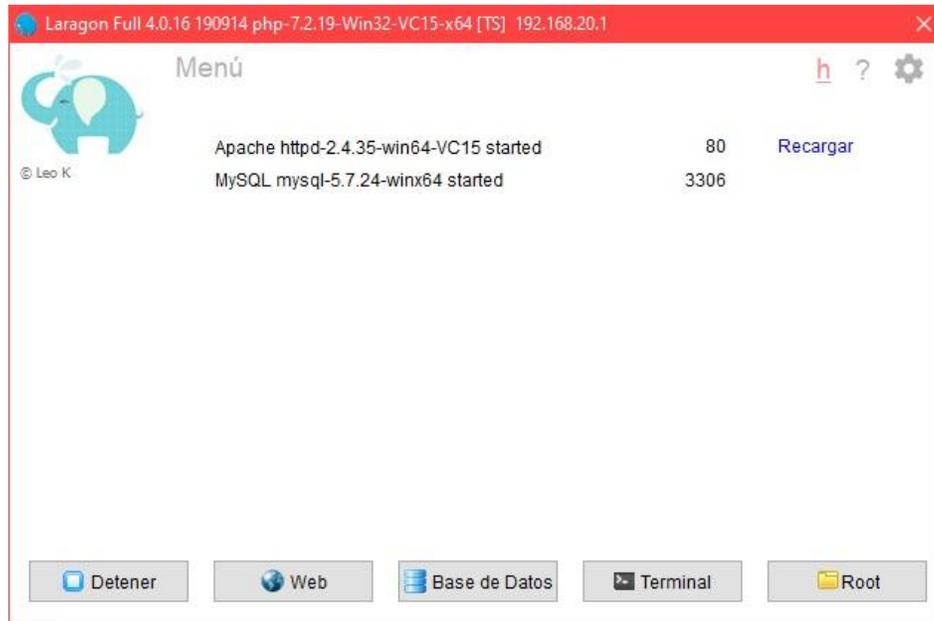
Lo primero es la información, a partir de ella se desarrollan los servicios y finalmente se documentan y catalogan. Se considera un flujo cerrado y continuo ya que podemos incorporar en cualquier momento información adicional con lo que tendríamos que entrar en los siguientes procesos.

#### **3.4.1. Implementación de los Geoservicios**

A continuación se describe la implementación de todos los geoservicios del sistema, datos espaciales, información georeferenciada. Para la implementación se hizo por pasos que son:

### **A1. Tener instalado Laragon.**

Ir a la página oficial de Laragon (<https://laragon.org/download/>) para la respectiva descarga del servidor, y se procede a la instalación.



**Figura 3.19. Laragon instalado.**

Fuente: Elaboración propia.

### **A2. Tener instalado PostgreSQL con su complemento PostGis.**

Se ingresa a la página oficial de Postgresql (<https://www.postgresql.org>) y se descarga PostgreSQL con su complemento PostGis y luego se procede a la instalación.

### **A3. Tener instalado Composer.**

Se ingresa a la página oficial de Composer (<https://getcomposer.org>) y se realiza la descarga y después la instalación.



**Figura 3.20. Composer.**

Fuente: Elaboración propia.

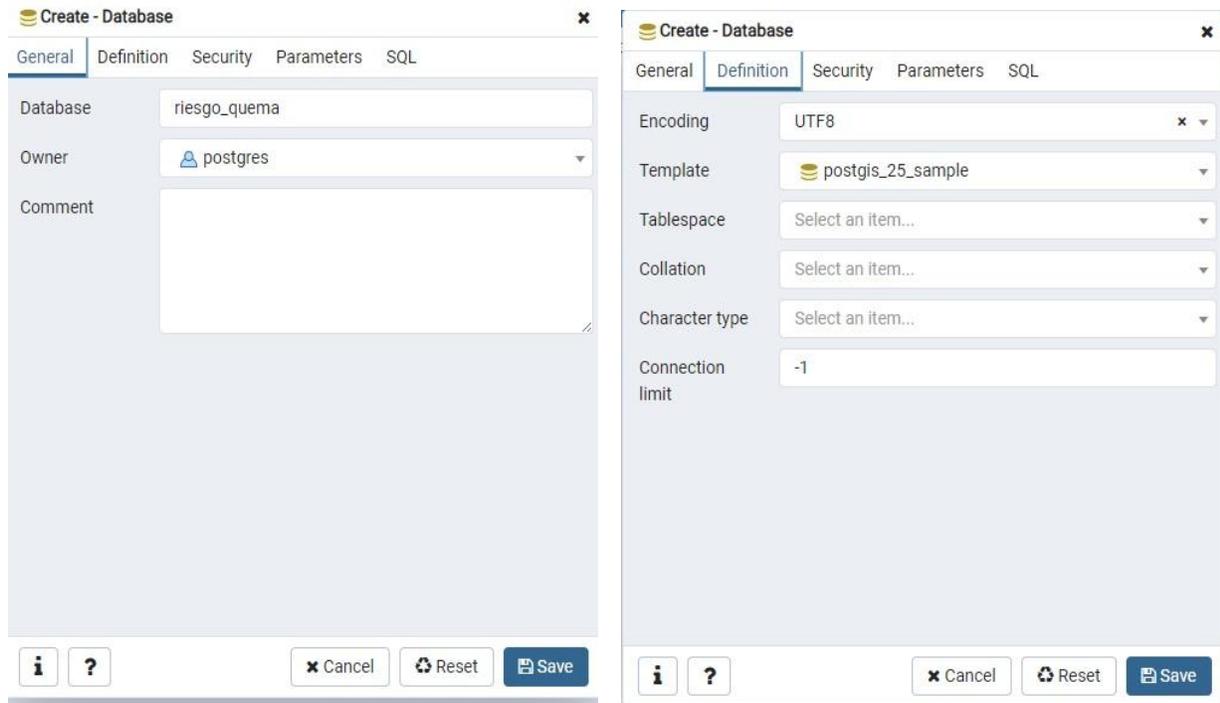
#### **A4. Tener instalado el Framework Laravel.**

Lo primero que se hace, es ir a la página oficial de Laravel, para sacar información sobre como instalar y crear un proyecto de laravel. La versión más estable es Laravel 6 y para la instalación y creación de un nuevo proyecto se usa el siguiente comando:

```
C:\Bitnami\wappstack-7.4.7-0\apache2\htdocs>composer create-project laravel/laravel ABTrq 6.*
```

#### **A5. Base de Datos**

El sistema gestor de base de datos es PostGreSQL con su extensión PostGis. En esta base de datos espacial se almacena toda la geoinformación que se desarrolló en QGIS. Se crea una base de datos espacial llamada riesgo\_quema en la cual se tiene la geoinformación de las imágenes Shapes en cada tabla, como se muestra a continuación:



**Figura 3.21. Creación de Base de Datos Espacial en PostgreSQL.**

Fuente: Elaboración propia

#### **A6. Configuración de Laravel para conexión con la Base de Datos**

Se realiza la configuración de Laravel para la conexión a la Base de Datos, y así poder acceder a los datos espaciales de nuestra Base de Datos riesgo\_quema.

```

9 DB_CONNECTION=localhost
10 DB_HOST=127.0.0.1
11 DB_PORT=5432
12 DB_DATABASE=riesgo_quema
13 DB_USERNAME=postgres
14 DB_PASSWORD=

```

**Figura 3.22. Conexión a la Base de Datos**

Fuente: Elaboracion Propia

#### **3.4.2. Recolección de datos**

La institución Autoridad de Fiscalización y Control Social de Bosques y tierra – ABT, no cuenta con un visor de mapas, reportes y graficas de las áreas de riesgo de quema, motivo por la cual se realizó una recolección de datos:

**Tabla 3.9**

*Clasificación de información geográfica de tipo vectorial*

<b>Nº</b>	<b>ENTIDAD O INFRAESTRUCTURA</b>	<b>TIPO</b>
1	Limite Nacional	Polígono
2	Limite Departamental	Polígono
3	Límite Provincial	Polígono
4	Limite Municipal	Polígono

**Nota.** Elaboración propia.

**3.4.3. Calcular el área de riesgo de quema**

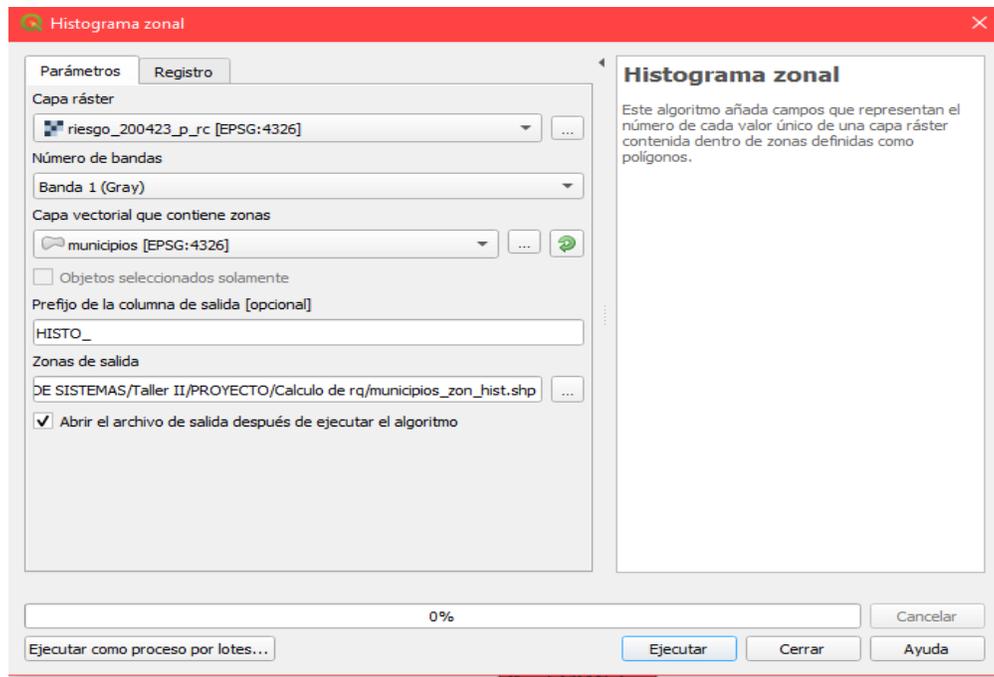
A continuación se detallan los procesos que se siguió para realizar el cálculo, sobre las áreas de riesgo de quema en Bolivia.

- Primero, descargo e instalo QGIS.
- Se realizó la descarga de la imagen de satélite (imagen de calor) de Bolivia.
- Se realizó la descarga de los mapas vectoriales (Departamentos, provincias y municipios).

Teniendo todo lo necesario para realizar el cálculo de riesgo de quema en toda Bolivia, se procede a hacer lo siguiente:

**B1. Calculo de número de pixeles por categoría y municipio**

Para realizar el conteo de pixeles por categoría y municipio se utiliza la herramienta de Histograma Zonal de QGIS:



**Figura 3.23. Calculo de número de pixeles.**

Fuente: Elaboración propia

Resultado de Histograma Zonal:

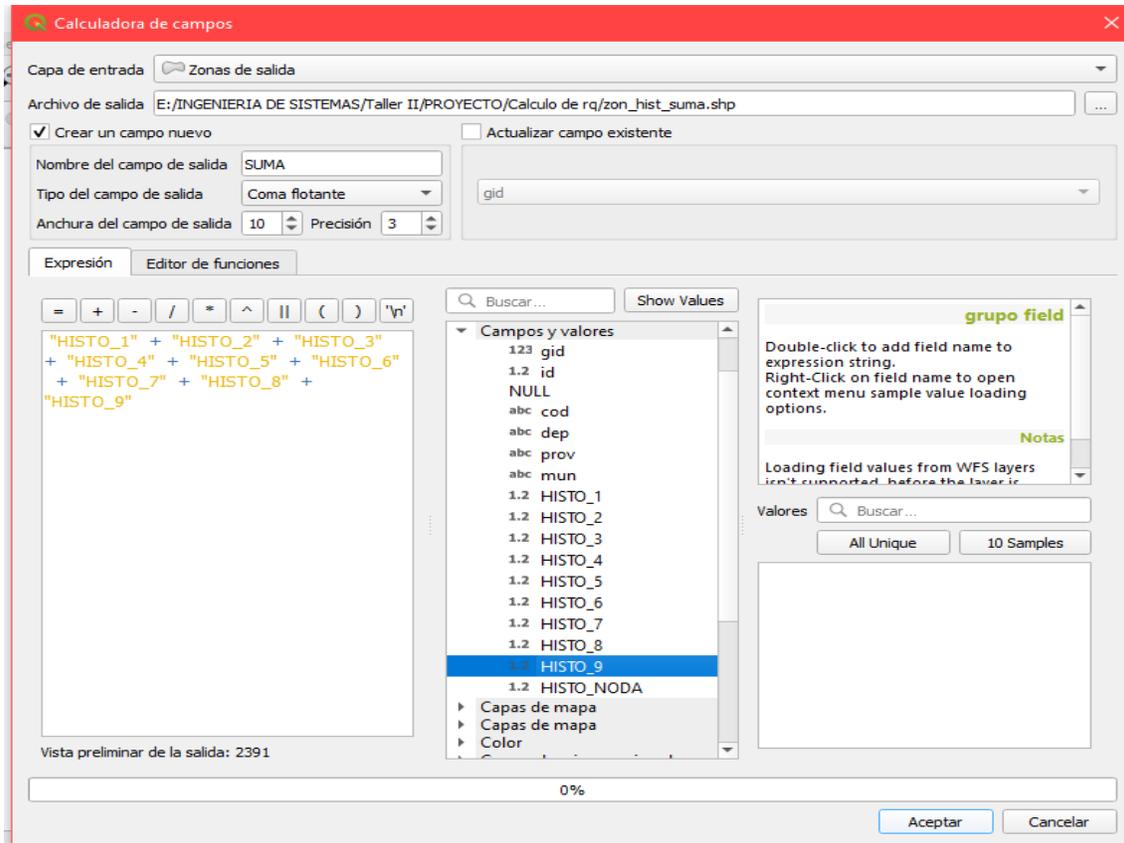
prov	mun	HISTO_1	HISTO_2	HISTO_3	HISTO_4	HISTO_5	HISTO_6	HISTO_7	HISTO_8	HISTO_9	HISTO_NODA	
1	Madre de Dios	San Lorenzo	5100	4073	2095	857	340	168	59	23	6	338
2	Madre de Dios	Sena	17904	6818	3332	1499	643	255	119	31	3	350
3	Loayza	Malla	1440	0	0	1	0	0	0	0	0	35
4	General Jos	B... Santa Rosa	33616	2119	983	521	296	165	100	44	2	1428
5	Aroma	Patacamaya	2345	0	0	0	0	0	0	0	0	3
6	Punata	Punata	360	0	0	0	0	0	0	0	0	0

**Figura 3.24. Resultado de Histograma Zonal.** Como se aprecia en la figura se realizó el conteo de pixeles (HISTO\_1, HISTO\_2, HISTO\_3, etc)

Fuente: Elaboración Propia

## B2. Sumatoria de los campos

Realizamos la sumatoria de los campos (HISTO\_1, HISTO\_2, HISTO\_3, etc), esto será para calcular el nivel de riesgo y para esto se utiliza la herramienta calculadora de campos de Qgis.



**Figura 3.25. Sumatoria de campos.**

Fuente: Elaboración propia

Resultado de la sumatoria de los nueve campos:

	mun	HISTO_1	HISTO_2	HISTO_3	HISTO_4	HISTO_5	HISTO_6	HISTO_7	HISTO_8	HISTO_9	HISTO_NODA	SUMA
1	San Lorenzo	5100	4073	2095	857	340	168	59	23	6	338	12721.000
2	Sena	17904	6818	3332	1499	643	255	119	31	3	350	30604.000
3	Malla	1440	0	0	1	0	0	0	0	0	35	1441.000
4	Santa Rosa	33616	2119	983	521	296	165	100	44	2	1428	37846.000
5	Patacamaya	2345	0	0	0	0	0	0	0	0	3	2345.000

**Figura 3.26. Resultado de sumatoria de campos.**

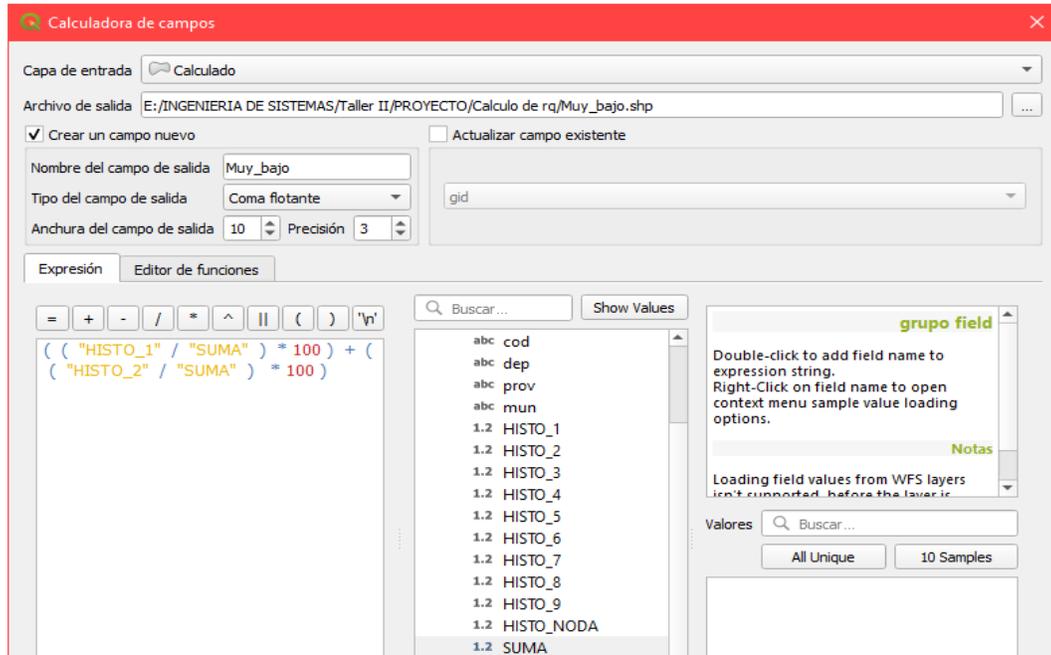
Fuente: Elaboración propia.

### B3. Calculo de Niveles de riesgo

Se realiza el cálculo de niveles de riesgo en valores de porcentaje (muy bajo, bajo, medio, alto, muy alto y extremo), para esto se hizo lo siguiente:

- **Calculo de nivel de riesgo muy bajo**

La ecuación utilizada en la calculadora de campos es:  $((\text{HIST}_1/\text{SUMA}) * 100) + ((\text{HIST}_2/\text{SUMA}) * 100)$ .



**Figura 3.27. Calculo del nivel de riesgo muy bajo.**

Fuente: Elaboración propia.

Resultado de cálculo realizado para el nivel de riesgo muy bajo:

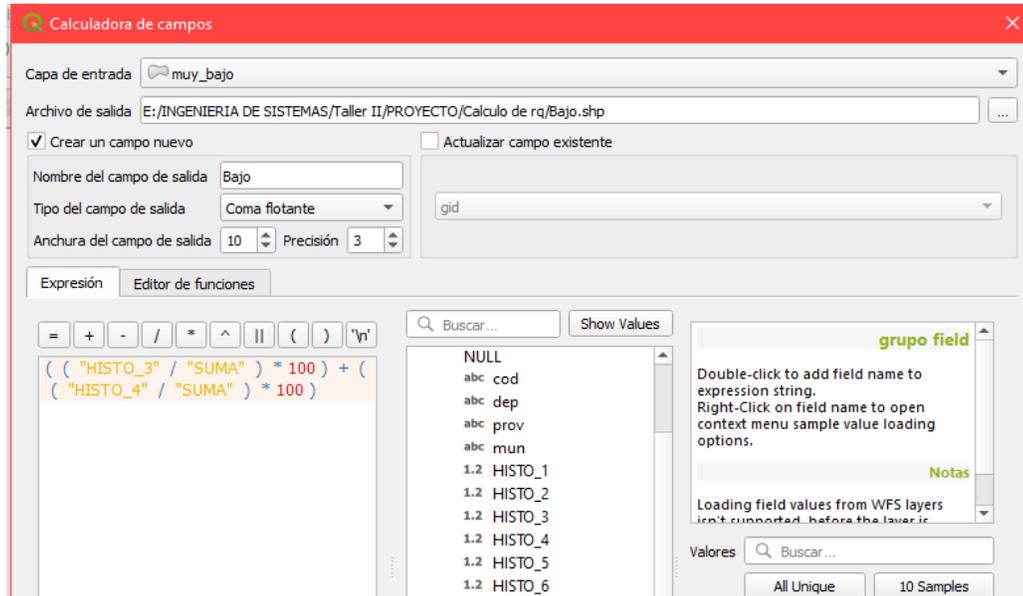
	HISTO_1	HISTO_2	HISTO_3	HISTO_4	HISTO_5	HISTO_6	HISTO_7	HISTO_8	HISTO_9	HISTO_NODA	SUMA	Muy_bajo
1	5100	4073	2095	857	340	168	59	23	6	338	12721.000	72.109
2	17904	6818	3332	1499	643	255	119	31	3	350	30604.000	80.780
3	1440	0	0	1	0	0	0	0	0	35	1441.000	99.931
4	33616	2119	983	521	296	165	100	44	2	1428	37846.000	94.422

**Figura 3.28. Resultado del cálculo de nivel de riesgo muy bajo.**

Fuente: Elaboración propia.

- **Calculo de nivel de riesgo bajo**

La ecuación utilizada en la calculadora de campos es:  $((\text{HIST}_3/\text{SUMA})*100) + ((\text{HIST}_4/\text{SUMA})*100)$ .



**Figura 3.29. Cálculo de nivel de riesgo bajo.**

Fuente: Elaboración propia.

Resultado del cálculo de riesgo bajo:

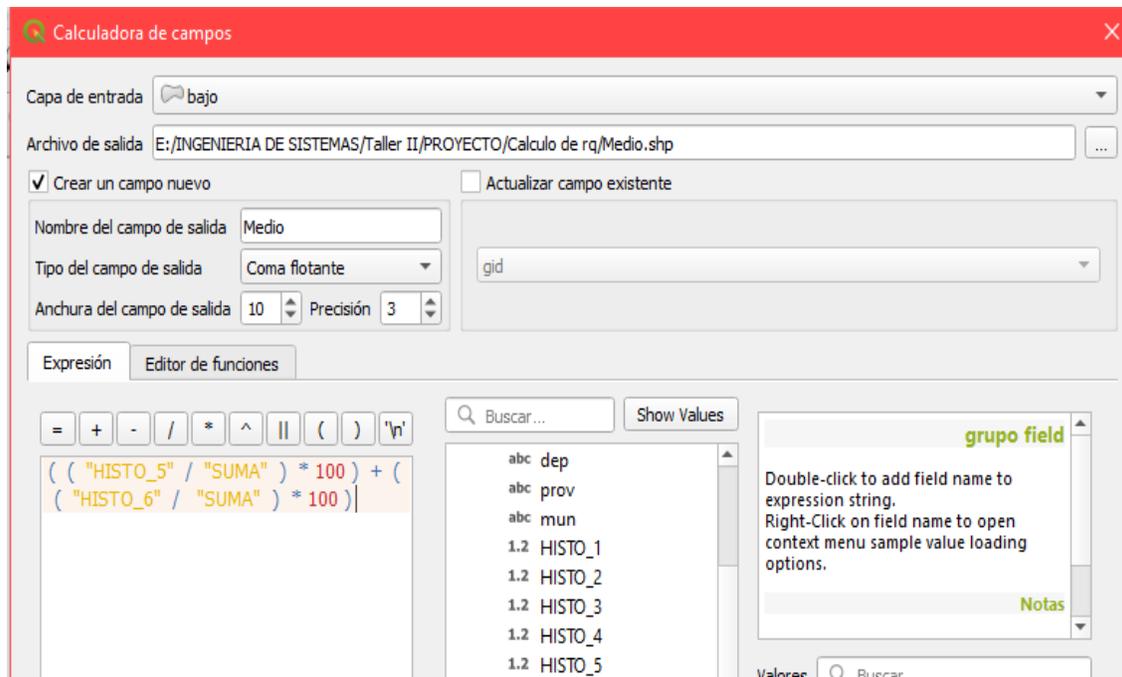
	HISTO_2	HISTO_3	HISTO_4	HISTO_5	HISTO_6	HISTO_7	HISTO_8	HISTO_9	HISTO_NODA	SUMA	Muy_bajo	Bajo
1	4073	2095	857	340	168	59	23	6	338	12721.000	72.109	23.206
2	6818	3332	1499	643	255	119	31	3	350	30604.000	80.780	15.786
3	0	0	1	0	0	0	0	0	35	1441.000	99.931	0.069
4	2119	983	521	296	165	100	44	2	1428	37846.000	94.422	3.974
5	0	0	0	0	0	0	0	0	3	2345.000	100.000	0.000

**Figura 3.30. Resultado del cálculo de nivel de riesgo bajo.**

Fuente: Elaboración propia.

- **Cálculo de nivel de riesgo de medio**

La ecuación utilizada en la calculadora de campos es:  $((\text{HIST}_5/\text{SUMA})*100) + ((\text{HIST}_6/\text{SUMA})*100)$ .



**Figura 3.31. Cálculo de nivel de riesgo medio.**

Fuente: Elaboración propia.

Resultado del cálculo de nivel de riesgo medio:

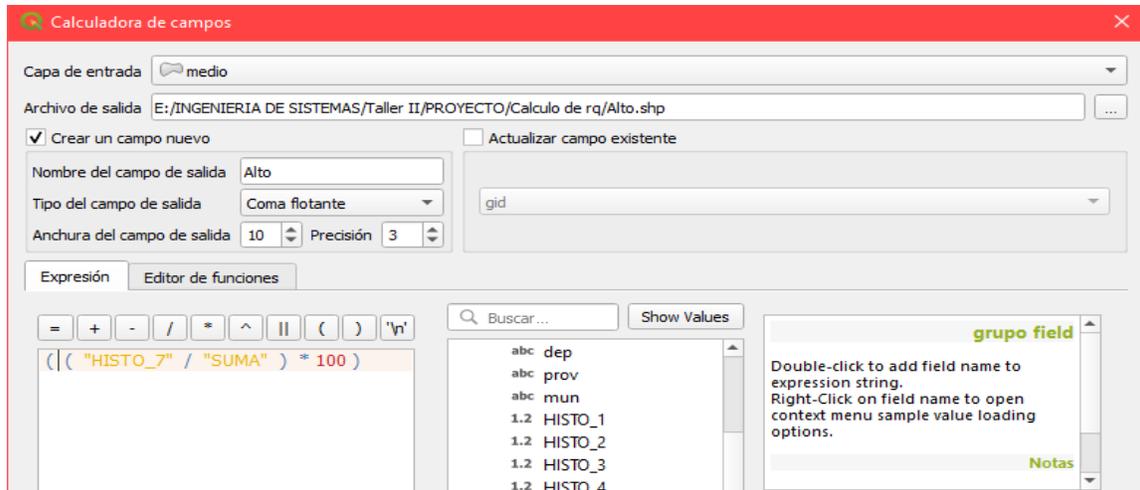
	HISTO_3	HISTO_4	HISTO_5	HISTO_6	HISTO_7	HISTO_8	HISTO_9	HISTO_NODA	SUMA	Muy_bajo	Bajo	Medio
1	2095	857	340	168	59	23	6	338	12721.000	72.109	23.206	3.993
2	3332	1499	643	255	119	31	3	350	30604.000	80.780	15.786	2.934
3	0	1	0	0	0	0	0	35	1441.000	99.931	0.069	0.000

**Figura 3.32. Resultado del cálculo de nivel de riesgo medio.**

Fuente: Elaboración propia.

- **Cálculo de nivel de riesgo alto**

La ecuación utilizada en la calculadora de campos es:  $((\text{HIST}_7/\text{SUMA}) * 100)$ .



**Figura 3.33. Calculo de nivel de riesgo alto.**

Fuente: Elaboración propia.

Resultado del cálculo de nivel de riesgo alto:

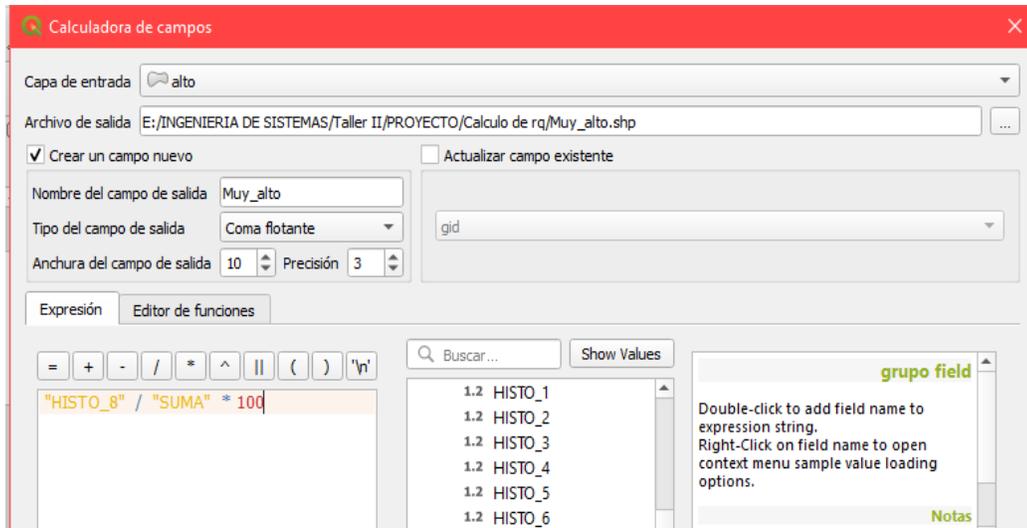
	HISTO_4	HISTO_5	HISTO_6	HISTO_7	HISTO_8	HISTO_9	HISTO_NODA	SUMA	Muy_bajo	Bajo	Medio	Alto
1	857	340	168	59	23	6	338	12721.000	72.109	23.206	3.993	0.464
2	1499	643	255	119	31	3	350	30604.000	80.780	15.786	2.934	0.389
3	1	0	0	0	0	0	35	1441.000	99.931	0.069	0.000	0.000

**Figura 3.34. Resultado del cálculo de riesgo alto.**

Fuente: Elaboración propia.

- **Calculo de nivel de riesgo muy alto**

La ecuación utilizada en la calculadora de campos es: ((HIST\_8/SUMA)\*100).



**Figura 3.35. Cálculo de nivel de riesgo muy alto.**

Fuente: Elaboración propia.

Resultado del cálculo de nivel de riesgo muy alto:

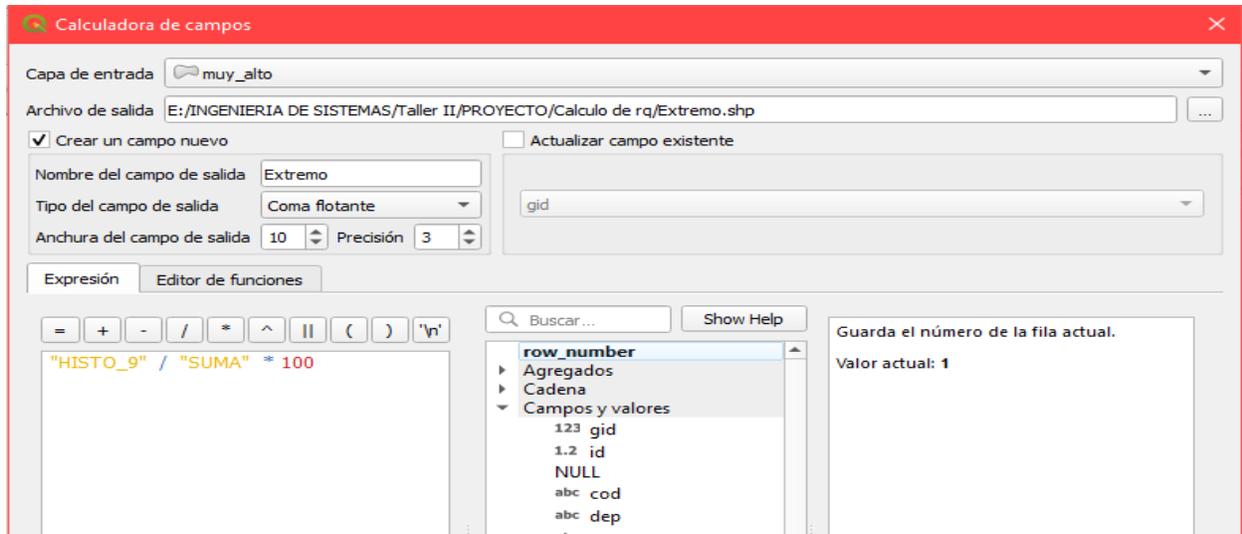
	HISTO_5	HISTO_6	HISTO_7	HISTO_8	HISTO_9	HISTO_NODA	SUMA	Muy_bajo	Bajo	Medio	Alto	Muy_alto
153	9656	4243	1871	137	0	14	80605.000	50.903	29.363	17.243	2.321	0.170
154	0	0	0	0	0	2	2883.000	100.000	0.000	0.000	0.000	0.000
155	0	0	0	0	0	1	6300.000	100.000	0.000	0.000	0.000	0.000

**Figura 3.36. Resultado del cálculo de nivel de riesgo muy alto.**

Fuente: Elaboración propia.

- **Cálculo de nivel de riesgo extremo**

La ecuación utilizada en la calculadora de campos es:  $((\text{HIST}_9/\text{SUMA}) * 100)$ .



**Figura 3.37. Cálculo de nivel de riesgo extremo.**

Fuente: Elaboración propia.

Resultado del cálculo de riesgo extremo:

	HISTO_6	HISTO_7	HISTO_8	HISTO_9	HISTO_NODA	SUMA	Muy_bajo	Bajo	Medio	Alto	Muy_alto	Extremo
273	0	0	0	0	11	5004.000	100.000	0.000	0.000	0.000	0.000	0.000
274	0	0	0	0	7	8899.000	100.000	0.000	0.000	0.000	0.000	0.000
275	0	0	0	0	117	3007.000	100.000	0.000	0.000	0.000	0.000	0.000
276	0	0	0	0	80	12272.000	00.067	0.023	0.000	0.000	0.000	0.000

**Figura 3.38. Resultado del cálculo de nivel de riesgo extremo.**

Fuente: Elaboración propia.

### 3.5.2. Conexión a la Base de Datos Espacial en QGIS

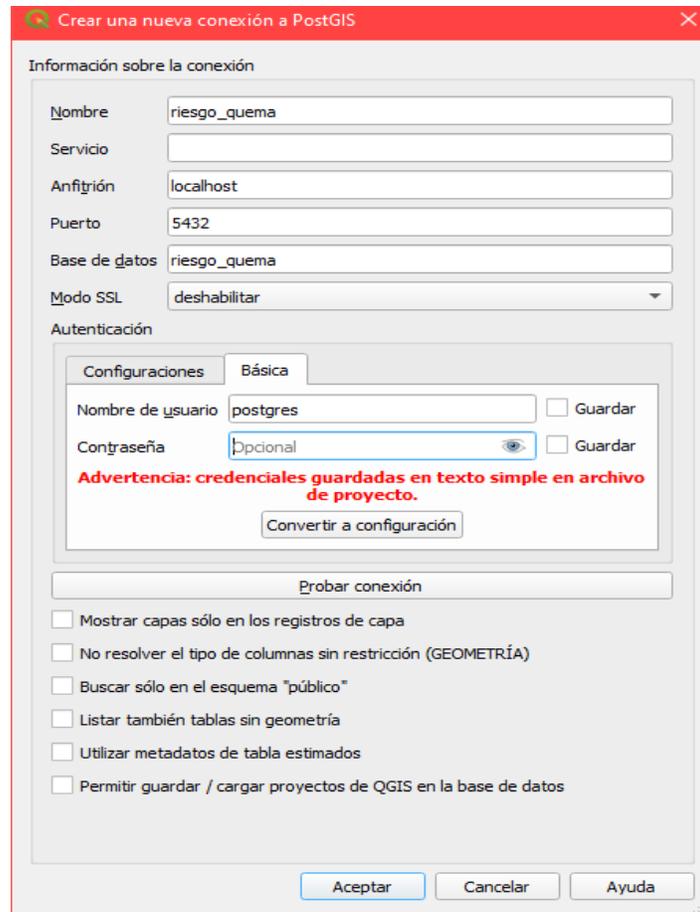
A continuación se detalla cómo se realiza la conexión a la base de datos espacial en QGIS, mediante PostGis.

Lo siguiente que se realizó fue: al tener una base de datos espacial ya creada en PostgreSQL con su complemento PostGis y también al tener la información geográfica en los archivos Shapes, se

procedió a realizar la conexión de la base de datos en QGIS mediante PostGis, la cual se hizo de la siguiente manera.

Nos vamos a la opción de PostGis y hacer click en nueva conexión en la cual nos saldrán los siguientes campos que se deben de llenar para realizar la conexión:

- **Nombre:** Es el nombre que se pondrá a la nueva conexión, en este caso sería (riesgo\_quema).
- **Anfitrión:** Se refiere al servidor que estamos usando puede ser externo o local, en nuestro caso será (localhost).
- **Puerto:** Es el puerto que usamos de nuestro servidor, por defecto es (5432).
- **Base de datos:** Colocamos el nombre de nuestra base de datos que ya creamos en PostgreSQL, (riesgo\_quema).
- Por ultimo damos click en probar conexión, y si no sale ningún error, le damos click en aceptar, y de esta manera ya tendríamos nuestra base de datos espacial conectada a Qgis.

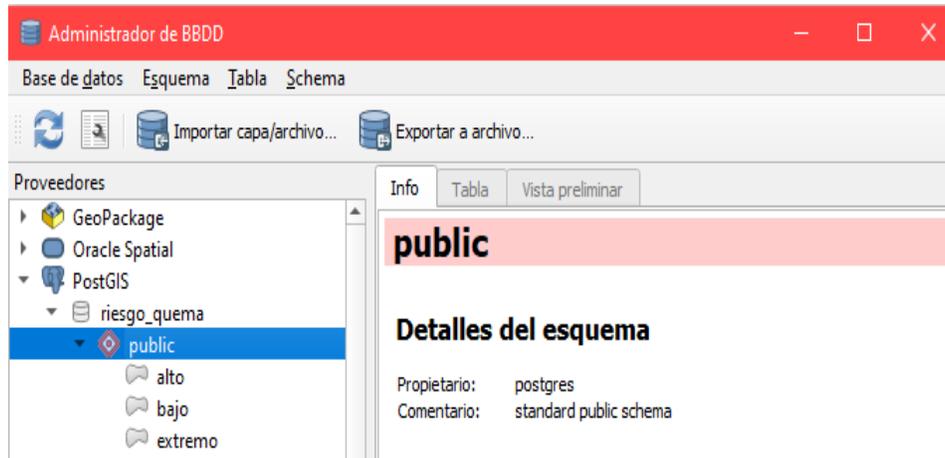


**Figura 3.39. Conexión de la Base de Datos Espacial en QGIS.**  
Fuente: Elaboración propia.

#### **3.4.4. Importar archivos Shapes del cálculo de nivel de riesgo de quema**

A continuación se detalla cómo se realizó la importación de los archivos shapes que contienen la información geográfica almacenada en la base de datos espacial, para eso se realizó lo siguiente:

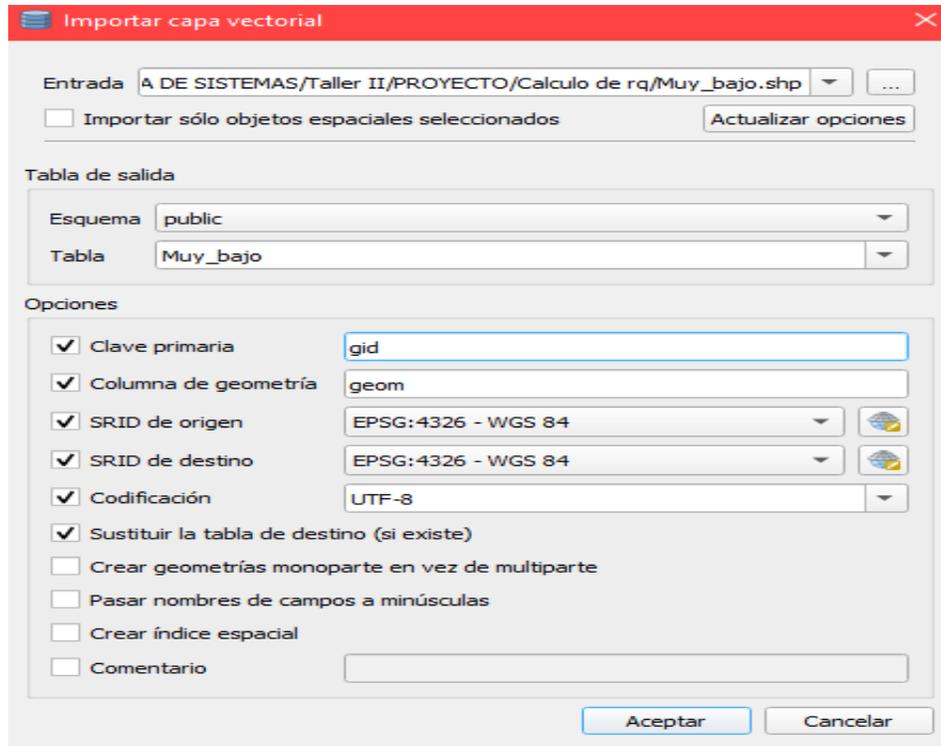
- Nos vamos a la pestaña de QGIS, Base de Datos y seleccionamos Administrador de Base de Datos.
- Buscamos en la opción PosGis nuestra base de datos que conectamos y nos vamos al esquema Public.



**Figura 3.40. Administrador de Base de Datos.**

Fuente: Elaboración propia.

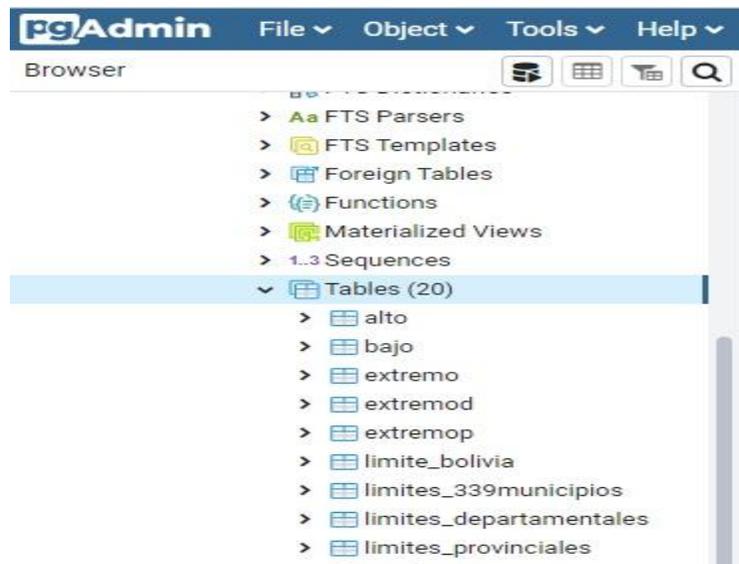
- Hacemos click en Importar capa/archivo, y seleccionamos los archivos Shapes, que creamos (muy\_bajo, bajo, medio, alto, muy\_alto y extremo).
- Después de seleccionar el archivo Shape debemos marcar los siguientes campos: (clave primaria, columna de geometría, SRID de origen, SRID de destino, codificación y sustituir la tabla de destino (si existe)), ya que son necesarios para una importación de información geográfica correcta.
- Si todo está correcto le damos en aceptar, de esta manera se realiza la importación de nuestros archivos Shapes a nuestra base de datos espacial, esto se debe hacer con cada archivo Shapes (muy\_bajo, bajo, medio, alto, muy\_alto y extremo).



**Figura 3.41. Importación de archivos Shapes a la Base de datos.**

Fuente: Elaboración propia.

En la siguiente figura se visualiza en PostgreSQL las tablas que se importaron desde Qgis con PostGis a la base de datos espacial.



**Figura 3.42. Tablas importadas de QGIS.**

Fuente: Elaboración Propia

Aquí podemos apreciar que las tablas se importaron correctamente, ya que tienen la información geográfica correcta.

	muy_bajo double precision	bajo double precision	medio double precision	alto double precision	muy_alto double precision	geom geometry
391	97.407	2.593	0	0	0	0106000020E61000...
518	100	0	0	0	0	0106000020E61000...
941	100	0	0	0	0	0106000020E61000...
764	99.738	0.262	0	0	0	0106000020E61000...
198	99.682	0.318	0	0	0	0106000020E61000...
506	100	0	0	0	0	0106000020E61000...

**Figura 3.43. Tablas importadas con datos correctos.**

Fuente: Elaboración propia.

Ya que nuestra información geográfica se encontraba en distintas tablas, se procedió a unir las con la siguiente consulta en PostgreSQL.

```

1 create table riesgo_20200423 as select gid, geom, cod, dep,
2 prov, mun, muy_bajo, bajo, medio, alto, muy_alto, extremo from extremo;

```

**Figura 3.44. Unión de tablas**

Fuente: Elaboración Propia

### 3.4.5. Automatización de la Estadística Zonal

Debido a que se realiza de manera manual el cálculo de las áreas de riesgos de quema, se hizo dos scripts, donde se automatiza el cálculo de las áreas de riesgos de quema.

A continuación se aprecia el primer Script, donde se realiza la descarga automática de las imágenes que se utilizan para el cálculo de Estadística Zonal.

```

GNU nano 2.7.4                               File: manual_fecha.sh

#!/bin/bash
mes="07"
dia="0701"
HORA=`date +%H:%M`

export PGPASSWORD="raster"

cd /descarga/
wget -c http://incendios.fan-bo.org/Satrifo/geotiff/ri/$mes/riesgo_20"$dia"_rc$
wget -c http://incendios.fan-bo.org/Satrifo/geotiff/ri/$mes/riesgo_20"$dia"_p$

rm -rf /temporal/*.*
cp /descarga/riesgo_20"$dia"_rc.zip /temporal/
cp /descarga/riesgo_20"$dia"_p_rc.zip /temporal/

cd /temporal/
unzip *.zip

[ Read 53 lines ]
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace    ^U Uncut Text ^T To Linter  ^_ Go To Line

```

**Figura 3.45. Script Fecha**

Fuente: Elaboración Propia

En la siguiente imagen se parecía el segundo Script, el cual realizada de forma automática el conteo de píxeles, seguido del cálculo de riesgos de quema

```

GNU nano 2.7.4                               File: conteo.sql

-----conteo de pixeles----- creacion de la capa sh contiene el conteo por las 9 categorias-----
create table rshp as (SELECT cod, dep, prov, mun, (pvc).VALUE, SUM((pvc).COUNT) AS hist_1
FROM r
INNER join municipios
ON ST_Intersects(rast, geom),
ST_ValueCount(ST_Clip(rast,geom),1, TRUE, ARRAY[1, 2, 3, 4, 5, 6, 7,8,9]) AS pvc)
GROUP BY (pvc).VALUE, cod, dep, prov, mun
ORDER BY mun, (pvc).VALUE);

-----copia de datos de rshp a rshp_backup
create table rshp_backup as select * from rshp;

-----clasificacion de histogramas-----
----- hist_1-----
create table h1 as (select cod, dep, prov, mun, "hist_1" from rshp where value='1');
----- hist_2-----
ALTER TABLE rshp RENAME COLUMN hist_1 TO hist_2;
ALTER TABLE rshp RENAME COLUMN mun TO mun2;
ALTER TABLE rshp RENAME COLUMN cod TO cod2;
create table h2 as (select cod2, mun2, "hist_2" from rshp where value='2');
----- hist_3-----
ALTER TABLE rshp RENAME COLUMN hist_2 TO hist_3;
ALTER TABLE rshp RENAME COLUMN mun2 TO mun3;
ALTER TABLE rshp RENAME COLUMN cod2 TO cod3;
create table h3 as (select cod3, mun3, "hist_3" from rshp where value='3');
----- hist_4-----
ALTER TABLE rshp RENAME COLUMN hist_3 TO hist_4;
ALTER TABLE rshp RENAME COLUMN mun3 TO mun4;
ALTER TABLE rshp RENAME COLUMN cod3 TO cod4;
create table h4 as (select cod4, mun4, "hist_4" from rshp where value='4');
----- hist_5-----

```

**Figura 3.46. Script Conteo**

Fuente: Elaboración Propia

### **3.5. Aplicación de la Métrica de Calidad WEB - QEM ISO 9126**

El objetivo de la métrica de calidad es alcanzar la calidad necesaria para satisfacer las necesidades del cliente. Esta métrica de calidad tiene la característica de evaluar dos ámbitos; el producto final y los procesos.

Los criterios de evaluación de calidad que tiene esta norma WEB - QEM ISO 9126 son: Funcionalidad, Confiabilidad, Usabilidad, Mantenibilidad y Portabilidad.

#### **3.5.1. Funcionalidad**

La funcionalidad examina si el sistema satisface los requisitos funcionales esperados. El objetivo es revelar problemas y errores en lo que concierne a la funcionalidad del sistema y su conformidad al comportamiento, expresado o deseado por el usuario.

En la siguiente tabla se calcula el punto función, los cuales miden el software desde una perspectiva del usuario, dejando de lado los detalles de codificación.

**a) Técnica Punto Función:** Esta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación y la metodología utilizada.

Para el cálculo de Punto Función se toma en cuenta 5 características de dominio de información.

- **Número de entradas de usuario:** Se refiere a cada entrada que proporciona datos al sistema.
- **Número de salidas de usuario:** Se refiere a cada salida que proporciona el sistema al usuario, entre estos están: informes, pantallas, mensajes de errores, etc.
- **Número de peticiones de usuario:** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta de software en forma de salidas interactivas.

- **Número de Archivos:** Se cuenta archivos maestro lógico, estos pueden ser: grupo lógico de datos, o un archivo independiente.
- **Número de interfaces externas:** se cuenta las interfaces legibles por la máquina que se utilizan para transmitir información a otro sistema.

En la siguiente tabla se realiza el cálculo punto función hallando la suma de estas características y el factor de complejidad, el resultado es el punto función no ajustado.

**Tabla 3.10**

*Calculo de Funcionalidad*

<b>PARÁMETROS DE ENTRADA</b>	<b>CUENTA</b>	<b>FACTOR DE COMPLEJIDAD</b>	<b>TOTALES</b>
Número de entradas de usuario	50	*3	150
Número de salidas de usuario	50	*2	100
Consultas	20	*3	60
Numero de archivos	18	*3	54
Numero de interfaces externas	2	*5	10
<b>Cuenta Total</b>			<b>374</b>

**Nota.** Elaboración propia.

Calculo del punto función ajustada. Los valores de  $F_i$ , se obtiene de los resultados de la siguiente tabla, bajo las ponderaciones descritos en la escala.

**Tabla 3.11**

*Tabla de ajuste de complejidad*

<b>ESCALAS</b>		<b>Sin importancia</b>	<b>Incremental</b>	<b>Moderado</b>	<b>Medio</b>	<b>Significativo</b>	<b>Esencial</b>
<b>Nº</b>	<b>Importancia</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
1.	¿Requiere el sistema copias de seguridad y recuperación fiable?						X
2.	¿Se requiere comunicación de datos?						X
3.	¿Existe funciones de procesos distribuidos?				X		
4.	¿Es crítico el rendimiento?					X	
5.	¿El sistema web será ejecutado el SO. Actual?						X
6.	¿Se requiere una entrada interactiva para el sistema?						X
7.	¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?			X			
8.	¿Se actualiza los archivos de forma interactiva?			X			
9.	¿Son complejas las entradas, salidas, os archivos o las peticiones?				X		
10.	¿Es complejo el procesamiento interno del sistema?					X	
11.	¿Se ha diseñado el código para ser Reutilizado?						X
12.	¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?						X
<b>Cuenta Total</b>		$\sum(F_i)=48$					

**Nota.** Elaboración propia.

- **Para el ajuste se utiliza la ecuación 2.1:**

$$PF = Cuenta\ Total * (confiabilidad\ proyecto + error\ min * \sum F_i)$$

$$PF = 374 * (0.65 + 0.01 * 48)$$

$$PF_{obtenida} = 422.62$$

Para el ajuste se utiliza la ecuación para hallar el punto función ideal al 100% de los factores que sería 60:

$$PF = 374 * (0.65 + 0.01 * 60)$$

$$PF_{ideal} = 467.5$$

- **Calculando del % de funcionalidad real:**

$$PF_{real} = PF_{obtenida} / PF_{ideal}$$

$$\text{Funcionalidad} = \frac{422.62}{467.5} * 100$$

$$\text{Funcionalidad} = 90\%$$

Interpretando, el sistema tiene una funcionalidad o utilidad del 90% para la institución, lo que indica que el sistema cumple con los requisitos funcionales de forma satisfactoria

### 3.5.2. Confiabilidad

Es la probabilidad de operación libre de fallos en un programa en un entorno determinado y durante un tiempo específico se toma en cuenta:

#### Enlaces

- Enlaces Rotos
- Enlaces Inválidos
- Enlaces no Implementados

#### Páginas

- Páginas Muertas
- Páginas bajo Construcción
- Errores de Ortografía

Para calcular los puntos función se usa la siguiente formula:

$$\text{Porcentaje de enlaces rotos} = \frac{CERI+CERE}{CTE} * 100 \text{ Ecuación 2.3}$$

Dónde:

**CERI:** Cantidad de enlaces rotos internos

**CERE:** Cantidad de enlaces rotos externos

**CTE:** Cantidad total de enlaces

Aplicamos la ecuación 2.3:

$$\text{Porcentaje de enlaces rotos} = \frac{35+60}{200} * 100 \Rightarrow 47.5$$

Porcentaje de presencia de propiedad:

$$\text{Porcentaje presencia ALT} = \frac{\text{CantidadImágenesALT}}{\text{CantidadTotalImágenes}} * 100 \text{ Ecuación 2.4}$$

Aplicamos la ecuación 2.4:

$$\text{Porcentaje presencia ALT} = \frac{3}{6} * 100 \Rightarrow 50$$

Y la confiabilidad se calcularía con la siguiente formula:

$$\text{Confiabilidad} = \text{PorcentajeEnlaces} + \text{PorcentajePresencialALT} \text{ Ecuación 2.5}$$

$$\text{Confiabilidad} = 47.5 + 50$$

$$\text{Confiabilidad} = 97.5\%$$

El sistema tiene un grado de confiabilidad del 98%, esto quiere decir si tomamos una muestra de 100 usuarios el 98 funciono correctamente y 2 usuarios tuvieron errores.

### 3.5.3. Usabilidad

Toma en cuenta la capacidad del software para ser comprendido, utilizado y atractivo para el usuario en determinadas condiciones.

**Modelo de Calidad para la Web**

**Título:** Usabilidad

**Tipo:** Característica

**Sub-características:**

Comprensibilidad (Global del Sitio)

Mecanismo de Ayuda y Retroalimentación

Aspectos de Interfaces

Aspectos Estéticos y de Estilo

**Características WEB - QEM Usabilidad**

Se realiza una tabla que toma en cuenta los siguientes puntos:

**A) Comprensión Global del Sitio**

**i. Esquema de Organización Global**

- Tabla de Contenidos
- Mapa del Sitio
- Índices (Alfabéticos, Temáticos, Híbridos...)

**ii. Visita Guiada (convencional y/o virtual)**

**iii. Mapa de Imagen**

**B) Aspectos de Interfaces y Estéticos**

**i. Permanencia y Estabilidad en la Presentación de los Controles Principales**

- Controles Directos
- Controles Indirectos
- Estabilidad

### C) Mantenimiento del Color de Los Enlaces

Se toma en cuenta el diseño que tiene el sistema.

**Tabla 3.12**

***Encuesta sobre la usabilidad del sistema***

N°	PREGUNTA	RESPUESTAS		PORCENTAJE
		SI	NO	DE SI
1.	¿Aprendió fácilmente a usar el sistema?	47	3	85
2.	¿En el momento que ingreso al sistema, le pareció amigable?	45	5	80
3.	¿Las pantallas que vio en el sistema fueron fáciles de comprender?	47	3	85
4.	¿El sistema respondió rápidamente a sus solicitudes?	47	3	85
5.	¿El sistema le facilito el trabajo?	50	0	100
6.	¿El sistema reduce el tiempo de trabajo?	50	0	100
7.	¿Es fácil navegar por las distintas opciones?	50	0	100
8.	¿El sistema le proporciono las respuestas solicitadas?	48	2	90
9.	¿Le fue de ayuda el sistema?	48	2	90
10.	¿El sistema no presento errores?	47	3	80
<b>PROMEDIO</b>				<b>90%</b>

**Nota.** Elaboración propia.

Se concluye que el sistema tiene una Usabilidad del 90%, es decir que cada 100 personas que lleguen a usar el sistema 90 personas indican que el sistema es fácil de manejar y comprender.

#### **3.5.4. Mantenibilidad**

La Mantenibilidad es la facilidad con que una modificación puede ser realizada. Está indicada por los siguientes sub atributos.

- Facilidad de análisis
- Facilidad de cambio
- Estabilidad

- Facilidad de prueba

El índice de madurez del software se calcula con la siguiente formula:

$$IMS = \frac{[Mt - (Fa + Fb + Fc)]}{Mt} \quad \text{Ecuación 2.6}$$

Dónde:

**Mt:** Número de módulos en la versión actual

**Fa:** Número de módulos en la versión actual que se han cambiado

**Fb:** Número de módulos en la versión actual que se han añadido

**Fc:** Número de módulos en la versión anterior que se han borrado en la versión actual.

Aplicando la ecuación 2.6:

$$IMS = \frac{[4 - (0 + 0 + 0)]}{4}$$

$$IMS = 4/4 \Rightarrow 100\%$$

La interpretación a este resultado establece un 100%, a lo que indica que no requiere de Mantenibilidad inmediatamente.

### 3.5.5. Portabilidad

La portabilidad es la capacidad que tiene el sistema para ser trasladado de un entorno a otro.

Para poder medir la portabilidad del sistema usaremos la siguiente formula que indica el grado de portabilidad que tiene un software.

$$GP = 1 - \frac{ET}{ER} \quad \text{Ecuación 2.7}$$

Dónde:

**ET:** Es la medida de los recursos necesarios para llevar el sistema a otro entorno.

**ER:** Es la medida de los recursos necesarios para crear el sistema en el entorno residente.

Si **GP > 0**, la portabilidad es más rentable que el re-desarrollo

Si  $GP = 1$ , la portabilidad es perfecta

Si  $GP < 0$ , el re-desarrollo es más rentable que la portabilidad.

Se procede a calcular la portabilidad con la ecuación 2.7:

$$GP = 1 - \frac{1}{5}$$

$$GP=75\%$$

### 3.5.6. Calidad Global

Con los cinco criterios que se desarrollaron anteriormente, nos ayudara a entender en que porcentaje se está manejando la calidad del sistema.

A continuación se muestra los resultados de la calidad del sistema en la siguiente tabla:

**Tabla 3.13**

*Calidad Global*

CARACTERÍSTICAS	PORCENTAJES
FUNCIONALIDAD	90%
CONFIABILIDAD	98%
USABILIDAD	90%
MANTENIBILIDAD	100%
PORTABILIDAD	75%
<b>TOTAL</b>	<b>91%</b>

**Nota.** Elaboración propia.

Se obtuvo un total de 91%, esto quiere decir que la calidad del sistema es altamente efectiva, por lo tanto se considera aceptable el producto.

### 3.6. Aplicación del Modelo de Estimación de Costos COCOMO II

El Modelo Constructivo de Costos, es un modelo matemático de base empírica utilizado para estimación de costos de software. Incluye tres submodelos, cada uno ofrece un nivel de detalle y aproximación, cada vez mayor, a medida que avanza el proceso de desarrollo del software: básico, intermedio y detallado.

Este modelo fue desarrollado por Barry W. Boehm a finales de los años 70 y comienzos de los 80, exponiéndolo detalladamente en su libro "Software Engineering Economics" (Prentice-Hall, 1981).

A continuación se describe los componentes relacionales con su complejidad asignada a cada uno de los factores que se deben considerar para la estimación del proyecto.

Las ecuaciones que se utilizaron son las siguientes:

En Persona – Mes:

$$E = a(KI^{(b)*m(X)}) \text{ Ecuación 2.8}$$

En Meses:

$$Tdev = c(E^d) \text{ Ecuación 2.9}$$

En Personas:

$$P = \frac{E}{Tdev} \text{ Ecuación 2.10}$$

Dónde:

**E:** es el esfuerzo requerido por el proyecto, en persona-mes

**Tdev:** es el tiempo requerido por el proyecto, en meses

**P:** es el número de personas requerido por el proyecto

**a, b, c y d** son constantes con valores definidos en una tabla, según cada submodelo

**KI:** es la cantidad de líneas de código, en miles.

**m(X):** Es un multiplicador que depende de 15 atributos.

Para calcular el esfuerzo, es necesario hallar la variable **KI o KLDC** (cantidad de líneas de código en miles), donde los PF es 422.62 (dato que se calculó en el capítulo anterior), y las líneas por cada PF equivalen a 29 según vemos en la tabla siguiente:

**Tabla 3.14**

*Conversión de puntos de función KLDC*

LENGUAJE	NIVEL	FACTOR LDC/PF
C	2.5	128
ANSI BASIC	5	6464
JAVA	6	53
PL/I	4	80
ANSI COBOL.74	3	107
VISUAL BASIC	7.00	46
ASP	9.00	36
<u>PHP</u>	<u>11.00</u>	<u>29</u>
VISUAL C++	9.50	34

**Nota.** T-Chambi, 2007.

Tras saber que 29 por cada LDC por cada PF, por ser PHP, el resultado será el siguiente:

$$\text{LDC} = \text{PFobtenido} * \text{FactorLDC/PF}$$

$$\text{LDC} = 422.62 * 29 / 1000 \Rightarrow 12.26$$

El tipo orgánico será el apropiado para nuestro caso, ya que el número de líneas de código no supera los 50 KLDC, los coeficientes que utilizaremos se la siguiente tabla, será de la fila de orgánico.

**Tabla 3.15***Coefficientes de a, b, c, d de COCOMO*

MODO	A	B	C	D
<b>Orgánico</b>	2.40	1.05	2.50	0.38
<b>Semi – Orgánico</b>	3.00	1.12	2.50	0.35
<b>Empotrado</b>	3.60	1.20	2.50	0.33

**Fuente:** Recuperado de Wikipedia

A partir de las tablas hechas publicadas por oehm se asigna un valor numérico a cada factor y se aplica la ecuación; el resultado es el factor de ajuste del esfuerzo.

**Tabla 3.16***Valores de coste COCOMO II*

Atributos	Valor					
	Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
<b>Atributos de Software</b>						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de Datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
<b>Atributos de Hardware</b>						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la maquina virtual		0,87	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
<b>Atributos de Personal</b>						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	0,86	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90		
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
<b>Atributos del Proyecto</b>						
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91	0,82	
Utilización de herramientas de software	1,24	1,10	1,00	0,91	0,83	
Restricciones de tiempo de desarrollo	1,24	1,10	1,00	1,04	1,10	

**Fuente:** Boehm 1989.

$$FAE=1.15*1.00*0.85*1.00*1.00*1.00*0.87*0.86*1.00*0.86*1.00*1.00*0.91*0.91*1.08$$

FAE=0.562522

Cálculos:

Cálculo de esfuerzo del desarrollo, En Persona – Mes:

$$E = a(KI^{(b)*m(X)}) \text{ Ecuación 2.8}$$

$$E = 2.40(12.26^{(1.05)*0.562522}) \Rightarrow 18.762429 \text{ persona/mes}$$

Calculo de tiempo de desarrollo, En Meses:

$$Tdev = c(E^d) \text{ Ecuación 2.9}$$

$$Tdev = 2.50(18.762429^{0.38}) \Rightarrow 6.8712 \text{ meses}$$

Calculo de personal promedio, En Personas:

$$P = \frac{E}{Tdev} \text{ Ecuación 2.10}$$

$$P = \frac{18.76.2429}{7.8712} \Rightarrow 2.3837 \text{ personas}$$

- **Costo en el Personal**

Según las cifras será necesario un equipo de 2 personas trabajando alrededor de 6 meses, suponiendo que el desarrollo del proyecto debe realizarse en un plazo de 3 meses, la cual incrementaremos a 3 el número de personas del equipo de proyecto. Supongamos que un programador gana alrededor de \$us 600:

$$\text{Costo en recursos de personal} = 3*3*600 = 5400 \text{ $us}$$

- **Costo en la elaboración del proyecto**

Se refieren a los gastos realizados a lo largo de las diferentes fases de AUP. Estas las podemos ver en la siguiente tabla:

**Tabla 3.16****Costo de elaboración del proyecto**

<b>DETALLE</b>	<b>IMPORTE \$US</b>
<b>ANALISIS Y DISEÑO DEL PROYECTO</b>	230
<b>MATERIAL DE ESCRITORIO</b>	65
<b>INTERNET</b>	80
<b>TOTAL</b>	<b>375</b>

**Fuente:** Elaboración propia.

- **Costo total del software**

El costo total del software se la obtiene de la sumatoria de los costos vistos hasta el momento, costo de desarrollo, implementación y elaboración del proyecto. Lo veremos en la siguiente tabla con una cantidad aproximada.

**Tabla 3.17****Costo total del proyecto**

<b>DETALLE</b>	<b>IMPORTE \$US</b>
<b>COSTO DE DESARROLLO</b>	5400
<b>COSTO DE ELABORACIÓN DEL PROYECTO</b>	375
<b>TOTAL</b>	<b>5775</b>

**Fuente:** Elaboración propia.



## **CAPÍTULO IV**

### **PRUEBAS Y RESULTADOS**



## **4.1. Pruebas y Resultados**

En este capítulo se comentan las pruebas hechas al sistema que validan a este proyecto como resultado al problema definido en el capítulo I, y finalmente se comenta los resultados.

### **4.1.1. Actualización de datos**

Se realizaba de forma manual el registro de datos e información por la cual se demoraba la actualización de datos, ahora con el sistema, se tiene una actualización de datos en menor tiempo.

### **4.1.2. Duplicidad de información**

Ya que los registros se realizaban de forma manual existía duplicidad de información, pero ahora con el sistema, se elimina la duplicidad de información que existía.

### **4.1.3. Visor de mapas**

No existía un visor de mapas, donde se visualice las áreas de riesgo de quema, pero ahora existe un módulo donde se visualiza el visor de mapas, en la cual se muestra las áreas de riesgo de quema de Bolivia.

### **4.1.4. Estadísticas y Reportes**

No se contaba con reportes y graficas sobre las áreas de quema. En este módulo se podrá realizar consultas para obtener información y datos sobre las áreas de riesgos de quema.

## **4.2. Pruebas Funcionales del Sistema**

Se realiza las pruebas necesarias para garantizar el funcionamiento del sistema, es también para asegurar el correcto trabajo de entrada de datos, la navegación en el sistema, procedimientos y la obtención de resultados.

**Tabla 4.1**

**Prueba de Navegación**

<b>PROCEDIMIENTO</b>		<b>DESCRIPCIÓN</b>	<b>VALOR</b>	
<b>Prueba previa requerida</b>		Navegación en los menús de la página web.	Si	
<b>Usuario</b>		Persona que ingresa al sistema para obtener información.		
<b>SECUENCIA DE PRUEBA</b>				
<b>PROCEDIMIENTOS</b>		<b>RESULTADOS ESPERADOS</b>	<b>CALIFICACIÓN DE FUNCIONALIDAD</b>	
Ingresa a los menús existentes en la página web		Acceso a todos los menús que tiene la página web.	Si	
<b>FALLAS ENCONTRADAS</b>		<b>DESCRIPCION</b>	<b>GRAVEDAD</b>	
Ninguna		Ninguna	-	
<b>Nº</b>	<b>Pasos de Prueba</b>	<b>Resultados Esperados</b>	<b>Positivo</b>	<b>Negativo</b>
1	Se prueba la entrada a los Box (Cajas de conceptos pequeños) de la página principal.	Se muestra los conceptos respectivos que contienen Box.	<b>X</b>	
2	Se prueba el acceso al menú de Geovisor.	Se muestra el visor de mapas	<b>X</b>	
3	Se prueba el acceso al menú de Estadísticas y Reportes.	Se muestra los campos de correspondientes de la selección de departamento, provincia y municipio.	<b>X</b>	
4	Se prueba el acceso al menú de Servicios.	Se muestra los servicios utilizados.	<b>X</b>	
5	Se prueba el acceso al menú de Ayuda.	Se muestra una guía de la página web.	<b>X</b>	
<b>COMENTARIO DE LA PRUEBA REALIZADA</b>				
Las pruebas de ingreso y navegación en la página web, se efectuaron con absoluta normalidad. Se obtuvo el resultado esperado en cuanto al ingreso de cada menú, mostrando la información correspondiente.				

**Fuente:** Elaboración propia

**Tabla 4.2**

*Prueba de Visualización de visor de mapas*

<b>PROCEDIMIENTO</b>		<b>DESCRIPCIÓN</b>	<b>VALOR</b>	
<b>Prueba previa requerida</b>		Visualización de visor de mapas.	Si	
<b>Usuario</b>		Persona que ingresa al sistema para obtener información.		
<b>SECUENCIA DE PRUEBA</b>				
<b>PROCEDIMIENTOS</b>		<b>RESULTADOS ESPERADOS</b>	<b>CALIFICACIÓN DE FUNCIONALIDAD</b>	
Ingresa al menú de Geovisor, que contiene el visor de mapas.		Se muestra el visor de mapas.	Si	
<b>FALLAS ENCONTRADAS</b>		<b>DESCRIPCION</b>	<b>GRAVEDAD</b>	
Ninguna		Ninguna	-	
<b>Nº</b>	<b>Pasos de Prueba</b>	<b>Resultados Esperados</b>	<b>Positivo</b>	<b>Negativo</b>
1	Se prueba el ingreso al menú de Geovisor.	Se muestra el visor de mapas donde se visualiza las áreas de riesgo de quema.	X	
2	Se prueba el zoom en el visor de mapas	Posteriormente se agranda el visor de mapa, realizando un zoom.	X	
3	Se prueba las herramientas del visor de mapas.	Se obtiene el arranque de las herramientas.	X	
<b>COMENTARIO DE LA PRUEBA REALIZADA</b>				
Las pruebas de visualización del visor de mapas, se efectuaron con absoluta normalidad. Se obtuvo el resultado esperado en cuanto a la obtención del visor de mapas.				

**Fuente:** Elaboración propia

**Tabla 4.3**

*Prueba Estadísticas y Reportes*

<b>PROCEDIMIENTO</b>		<b>DESCRIPCIÓN</b>	<b>VALOR</b>	
<b>Prueba previa requerida</b>		Estadísticas y Reportes	Si	
<b>Usuario</b>		Persona que ingresa al sistema para obtener información.		
<b>SECUENCIA DE PRUEBA</b>				
<b>PROCEDIMIENTOS</b>		<b>RESULTADOS ESPERADOS</b>	<b>CALIFICACIÓN DE FUNCIONALIDAD</b>	
Ingresa al menú de Estadísticas y Reportes.		Se muestra los campos de selección de departamento, provincia y municipio, también se visualiza una tabla y una gráfica estadística.	Si	
<b>FALLAS ENCONTRADAS</b>		<b>DESCRIPCION</b>	<b>GRAVEDAD</b>	
Ninguna		Ninguna	-	
<b>Nº</b>	<b>Pasos de Prueba</b>	<b>Resultados Esperados</b>	<b>Positivo</b>	<b>Negativo</b>
1	Se prueba la selección de los campos de Departamento, Provincia y Municipio.	El usuario selecciona los campos correspondientes.	<b>X</b>	
2	Se prueba botón de Aplicar, para mostrar reportes.	El usuario obtendrá una tabla con los datos obtenidos de la selección.	<b>X</b>	
3	Se prueba la visualización de gráficos estadísticos.	Se muestra en un gráfico los datos obtenidos de la selección.	<b>X</b>	
<b>COMENTARIO DE LA PRUEBA REALIZADA</b>				
Las pruebas de Estadísticas y Reportes, se efectuaron con absoluta normalidad. Se obtuvo el resultado esperado en cuanto a la obtención de datos sobre las áreas de riesgo de quema.				

**Fuente:** Elaboración propia

**Tabla 4.5**

**Prueba de Servicios**

<b>PROCEDIMIENTO</b>		<b>DESCRIPCIÓN</b>	<b>VALOR</b>	
<b>Prueba previa requerida</b>		Ingreso a los servicios usados en el sistema.	Si	
<b>Usuario</b>		Persona que ingresa al sistema para obtener información.		
<b>SECUENCIA DE PRUEBA</b>				
<b>PROCEDIMIENTOS</b>		<b>RESULTADOS ESPERADOS</b>	<b>CALIFICACIÓN DE FUNCIONALIDAD</b>	
Ingresa al menú de Servicios, donde contiene los servicios utilizados en el sistema.		Se muestran links de los servicios utilizados.	Si	
<b>FALLAS ENCONTRADAS</b>		<b>DESCRIPCION</b>	<b>GRAVEDAD</b>	
Ninguna		Ninguna	-	
<b>Nº</b>	<b>Pasos de Prueba</b>	<b>Resultados Esperados</b>	<b>Positivo</b>	<b>Negativo</b>
1	Se prueba el ingreso al servicio web (WMS) de Imagen Satelital de Calor.	Posteriormente se dirige a la página de origen de donde se obtuvo el servicio de Imagen Satelital de Calor.	X	
2	Se prueba el ingreso al servicio web (WMS) de Limite Nacional.	Posteriormente se dirige a la página de origen de donde se obtuvo el servicio de Limite Nacional.	X	
3	Se prueba el ingreso al servicio web (WMS) de Limite Departamental.	Posteriormente se dirige a la página de origen de donde se obtuvo el servicio de Limite Departamental.	X	
4	Se prueba el ingreso al servicio web (WMS) de Limite Provincial.	Posteriormente se dirige a la página de origen de donde se obtuvo el servicio de Limite Provincial.	X	
5	Se prueba el ingreso al servicio web (WMS) de Limite Municipal.	Posteriormente se dirige a la página de origen de donde se obtuvo el servicio de Limite Municipal.	X	

**COMENTARIO DE LA PRUEBA REALIZADA**

Las pruebas de ingreso a los servicios usados en el sistema se efectuaron con absoluta normalidad. Se obtuvo el resultado esperado en cuanto a la direccionalidad a las páginas de origen de los servicios web (WMS).

**Fuente:** Elaboración propia



**CAPÍTULO V**

**CONCLUSIONES Y RECOMENDACIONES**

## **5.1. Conclusiones**

- ✓ Los Sistemas de Información Geográfica (SIG), comprenden un conjunto de equipo informático, que manipulado por medio de programas de computadora, hace más eficiente la administración de información digital georeferenciada.
- ✓ Es importante dejar bien estructurada una base de datos que almacene grandes volúmenes de información, porque de esto depende la eficiencia y funcionabilidad que tenga, para poder trabajar conjuntamente con el programa SIG.
- ✓ El proyecto de grado fue desarrollado satisfactoriamente y cumplió con todos y cada uno de los objetivos propuestos, pero principalmente se logró proporcionar una herramienta eficiente que dispone información geográfica importante sobre las áreas de riesgo de quema, en la cual ayuda en la toma de decisiones.

## **5.2. Recomendaciones**

- ✓ Se sugiere que todo el personal humano involucrado en el uso de sistemas de información geográfica, tenga la debida capacitación, ya que mientras mayor es la capacitación, es mejor la perspectiva global del funcionamiento del sistema. Por lo tanto pueda llegar, además de administrar el sistema, aportar ideas para mejorar su funcionamiento.
- ✓ Es recomendable que antes de implementar un SIG, se realice una investigación de las instituciones públicas y privadas, que se dedican en la generación y comercialización de información geográfica. Si se desea obtener algún tipo de información para el desarrollo de sistemas de información geográfica, se debe investigar cual es la fuente acorde a las necesidades específicas.
- ✓ Se propone que instituciones como la ABT, divulguen o den a conocer la información que desarrollan.

- ✓ Es aconsejable que instituciones enfocadas al medio ambiente, desarrollen programas a través de los cuales, se haga ver la realidad del medio ambiente del país y cómo reducir la vulnerabilidad de las áreas con riesgo de quema.



# **BIBLIOGRAFÍA**



## Bibliografía

- Garlan , D., & Shaw, M. (2016). *An introduction to Software Architecture*. New Jersey: Publishing Company.
- Agafonkin, V. (17 de Noviembre de 2019). *Leaflet*. Obtenido de Leaflet: <https://leafletjs.com/>
- ArcGIS Resources. (4 de Junio de 2020). *esri*. Recuperado el 20 de marzo de 2020, de Tres representaciones fundamentales de capas de información geográfica: <https://resources.arcgis.com/es/help/getting-started/articles/026n0000000n000000.htm>
- Barrera Bassols, N., & Palma Ruiz, A. (2012). *Geografía*. Mexico: ISBN.
- Bermejo, E. (13 de Mayo de 2019). *GeoInnova*. Recuperado el 17 de marzo de 2020, de Infraestructura de Datos Espaciales: [https://es.wikipedia.org/wiki/Infraestructura\\_de\\_Datos\\_Espaciales](https://es.wikipedia.org/wiki/Infraestructura_de_Datos_Espaciales)
- Bitnami. (2020). *Bitnami*. Recuperado el 26 de abril de 2020, de Wapp: <https://bitnami.com/stack/wapp>
- Boehm, B. W. (1981). *Software Engineering Economics*. EEUU: Prentice-Hall.
- Chiavenato, I. (2006). *Introducción a la teoría general de la administración*. Portugal: Elsevier.
- Denzer, P. (2002). *PostgreSQL*. Mexico: U.T.F.S.M.
- Earth Observing System*. (2020). Recuperado el 18 de marzo de 2020, de Pancromatico: <https://eos.com/panchromatic/es/>
- EcuRed. (4 de Mayo de 2012). *EcuRed*. Obtenido de Imagen Satelital: [https://www.ecured.cu/Imagen\\_satelital](https://www.ecured.cu/Imagen_satelital)
- Energy computer systems*. (13 de Marzo de 2020). Recuperado el 17 de marzo de 2020, de Visor geográfico: <https://energyco.com/soluciones/visor-geografico/?x=0&y=0>

- España, J. (2007). Aplicando SIG. En J. España, *Prevención de Sismos* (pág. 23). Morelia: Séneca.
- Fernando, S. (12 de Enero de 2016). *Portal de Comunicación*. Obtenido de Imágenes Multiespectrales: <https://comunicacion.us.es/canal-ciencia/imagenes-multiespectrales-partir-de-fotografias-rgb>
- Fing. (30 de Marzo de 2017). *Tecnólogo Informático Estructuras de Datos y Algoritmos* . Obtenido de Módulos: <https://www.fing.edu.uy/tecnoinf/mvd/cursos/eda/material/teo/anterior/6-EDA-teo.pdf>
- Flechas Becerra, H. Y., & Almanza, W. (2012). *Introduccion a la ingenieria de sistemas*. Duitama: ISBN.
- Georgina. (2012). *Definiciones de auditoria infromatica*. Obtenido de Que es el monitoreo: <http://actividadesauditoriainformatica382.blogspot.com/2012/03/que-es-el-monitoreo-y-escribe-los-4.html>
- Geosysnet. (21 de Febrero de 2017). *Leaflet*. Recuperado el 02 de marzo de 2020, de Leaflet: [http://geosysnet.de/custom/downloads/MaptimeBerlin\\_Leaflet\\_21February2017.pdf](http://geosysnet.de/custom/downloads/MaptimeBerlin_Leaflet_21February2017.pdf)
- Gómez, A., López, M., Migani, S., & Otazú, A. (24 de marzo de 2020). *COCOMO Un modelo de estimación de proyectos de software*. Obtenido de COCOMO Un modelo de estimación de proyectos de software: [https://www.academia.edu/4853589/UN\\_MODELO\\_DE\\_ESTIMACION\\_DE\\_PROYECTOS\\_DE\\_SOFTWARE](https://www.academia.edu/4853589/UN_MODELO_DE_ESTIMACION_DE_PROYECTOS_DE_SOFTWARE)
- González Pinzón, M. F., & González Sanabria, J. S. (2013). *Aplicación del estándar ISO/IEC 9126-3 en el modelo de datos conceptual entidad-relación* . Colombia: ISSN 0121-1129.

Gonzalez, A. R. (Septiembre de 2013). *ResearchGet*. Obtenido de Análisis de Imágenes Hiperespectrales:  
[https://www.researchgate.net/publication/259840849\\_Analisis\\_de\\_Imagenes\\_Hiperespectrales](https://www.researchgate.net/publication/259840849_Analisis_de_Imagenes_Hiperespectrales)

Infomática. (12 de Abril de 2017). *Infomática*. Obtenido de Módulo:  
<https://sites.google.com/site/informaticae14/iv-desarrollo-de-modularidad/unidad-iv-desarrollo-de-modularidad>

ittgweb. (29 de Mayo de 2016). *Ingeniería de Software*. Obtenido de Descomposición Modular:  
<https://ittgweb.wordpress.com/2016/05/29/descomposicion-modular/>

Jeff. (19 de Abril de 2019). *Styde*. Recuperado el 02 de Agosto de 2020, de Laragon:  
<https://styde.net/laragon-un-entorno-de-desarrollo-para-laravel-en-windows/>

Jeffrey , S., & John, E. (2006). Definition Geographic Information Systems. En S. Jeffrey, & E. John, *Geographic Information Systems: An Introduction* (pág. 10). Prentice-Hall.

Koch, N., & Kraus, A. (2003). A metamodel for UWE. En *A metamodel for UWE* (pág. 2). Munchen: Oettingest.

Laragon. (24 de Octubre de 2018). *Laragon*. Recuperado el 25 de septiembre de 2020, de Laragon: <https://laragon.org/>

Laravel. (2020). *Laravel*. Recuperado el 12 de abril de 2020, de Laravel: <https://laravel.com/>

Lexico. (16 de marzo de 2020). *Analisis*. Obtenido de Analisis:  
<https://www.lexico.com/es/Análisis>

López, B. R. (9 de Enero de 2018). *TYC GIS*. Recuperado el 21 de marzo de 2020, de ¿Que es Google Earth Engine?: <https://www.cursosgis.com/que-es-google-earth-engine/>

M. F., P., M. E., S., G., R., & L., O. (17 de Enero de 2017). *Soporte Automatizado a la Metodología Web QEM*. Obtenido de <http://sedici.unlp.edu.ar>

Mardi. (13 de Abril de 2020). *Observatorio Tecnológico*. Recuperado el 16 de Abril de 2020, de BitNami: <http://recursostic.educacion.es/observatorio/web/fr/software/software-general/767-bitnami-instalador-de-aplicaciones-web>

Martinez Llario, J. (2013). *PostGIS 2 Análisis Espacial Avanzado*. ISBN-13 978-84-615-8833-6.

Marttone, E. d. (2016). Geografía. En E. d. Marttone, *Tratado de geografía física* (pág. 14). Mexico: Juventud.

Morales, A. (23 de Mayo de 2016). *Leaflet*. Recuperado el 04 de octubre de 2019, de Leaflet: [https://en.wikipedia.org/wiki/Leaflet\\_\(software\)](https://en.wikipedia.org/wiki/Leaflet_(software))

Olaya, V. (2014). ¿Que es un SIG? En V. Olaya, *Sistemas de Información Geográficas* (pág. 7). Creative Common Atribución.

Olaya, V. (2014). *Sistemas de Información Geográfica*.

Open Layers. (6 de Noviembre de 2018). *Open Layers*. Obtenido de Open Layers: <https://openlayers.org/>

Parro. (8 de Agosto de 2020). *Definición de riesgo de incendio*. Obtenido de Definiciones y traducciones: <https://www.parro.com.ar/definicion-de-riesgo+de+incendio>

Patricia. (1 de Agosto de 2019). *Tecnología e Informática*. Obtenido de Datos: <http://latecnologiavirtual.blogspot.com/2009/08/datos.html>

Pérez Porto, J., & Gardey, A. (2013). *definicion.de*. Obtenido de monitoreo: <https://definicion.de/monitoreo/>

PHP. (9 de Julio de 2020). *PHP*. Obtenido de PHP: <https://www.php.net>

PostGIS. (23 de Junio de 2020). *PostGIS*. Obtenido de PostGIS: <https://postgis.net>

PostgreSQL. (25 de Junio de 2020). *PostgreSQL*. Obtenido de PostgreSQL:  
<https://www.postgresql.org>

Pressman, R. S. (2010). *Ingeniería del Software, un enfoque práctico*. México: McGraw Hill .

Pressman, R. S. (2010). *Ingeniería del Software, Un enfoque práctico*. México: McGraw-Hill.

QGIS. (21 de Junio de 2020). *QGIS*. Obtenido de QGIS - El SIG Líder de Código Abierto para Escritorio: <https://www.qgis.org/es/site/about/index.html>

Rees, D. (2016). *PHP Panda The Php programming language for everyone*. canada.

*Riesgo de incendio*. (2015). Obtenido de Portal de los riesgos de labores de los trabajadores de la enseñanza: <https://riesgoslaborales.saludlaboral.org/portal-preventivo/riesgos-laborales/riesgos-relacionados-con-la-seguridad-en-el-trabajo/riesgos-de-incendios/>

Rios Ortega, J. (2013). *El concepto de informacion: dimensiones bibliotecologica sociologica y cognostiva*. Mexico: ISSN.

Romero Castro, M. I. (2018). *INTRODUCCIÓN A LA SEGURIDAD INFORMÁTICA Y EL ANÁLISIS DE VULNERABILIDADES*. Manabi: Área de Innovación y Desarrollo,S.L.

Schiaffarino, A. (12 de Marzo de 2019). *Infranetworking*. Obtenido de Modelo de Cliente Servidor: <https://blog.infranetworking.com/modelo-cliente-servidor/>

Tomli, C. D. (2015). Geographic information systems . En C. D. Tomli, *Geographic information systems and cartographic modelling* (pág. 9). Portland: PrenticeHall.

Vilariño, J. (2010). *Modelo para la selección de la metodología de desarrollo web de una aplicación según sus características funcionales*. Caracas.

Virtuniversidad. (16 de Marzo de 2016). *Programación por capas*. Recuperado el 19 de marzo de 2020, de Programación por capas:

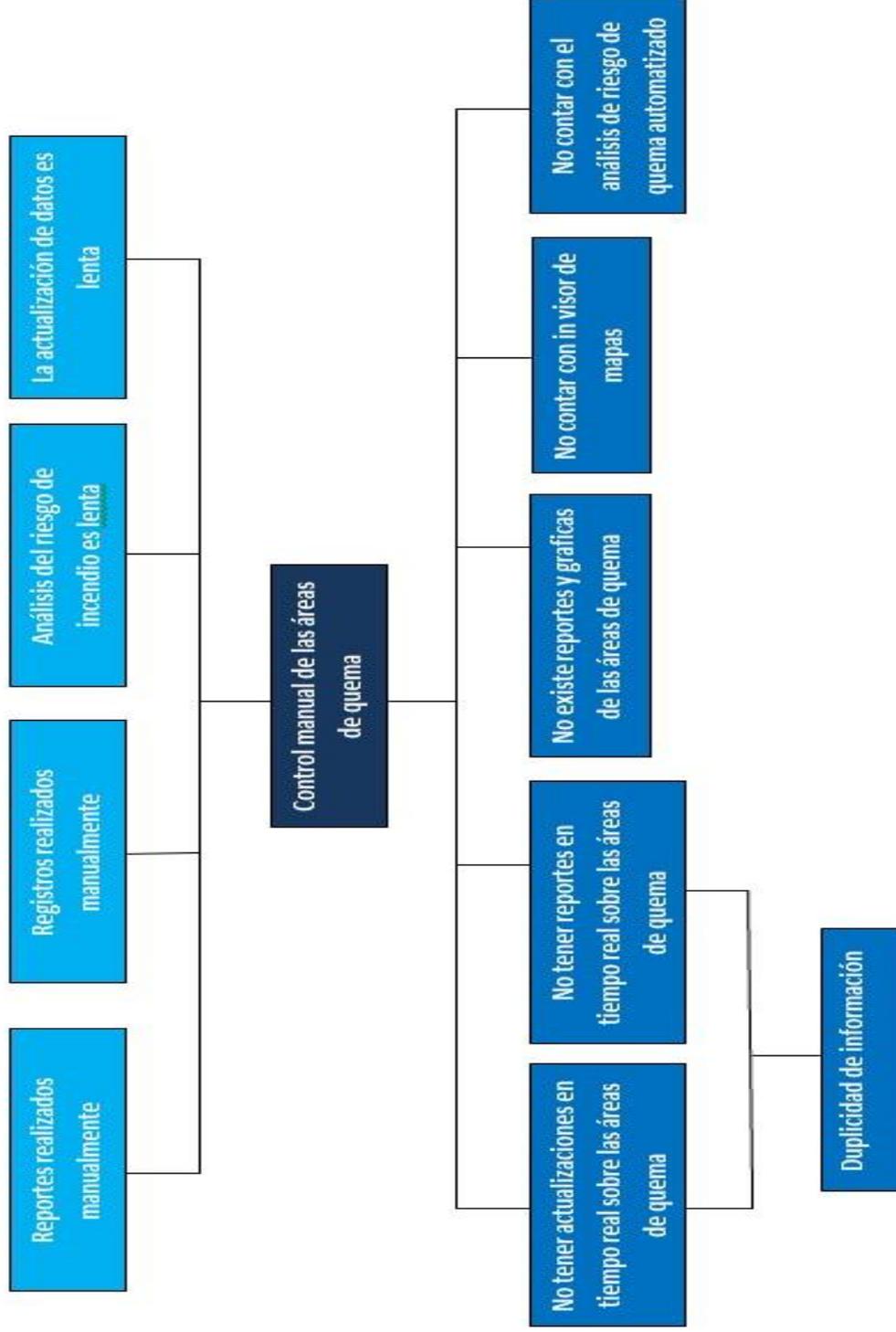
[https://www.virtuniversidad.com/greenstone/collect/informatica/archives/HASH0195.dir/  
doc.pdf](https://www.virtuniversidad.com/greenstone/collect/informatica/archives/HASH0195.dir/doc.pdf)



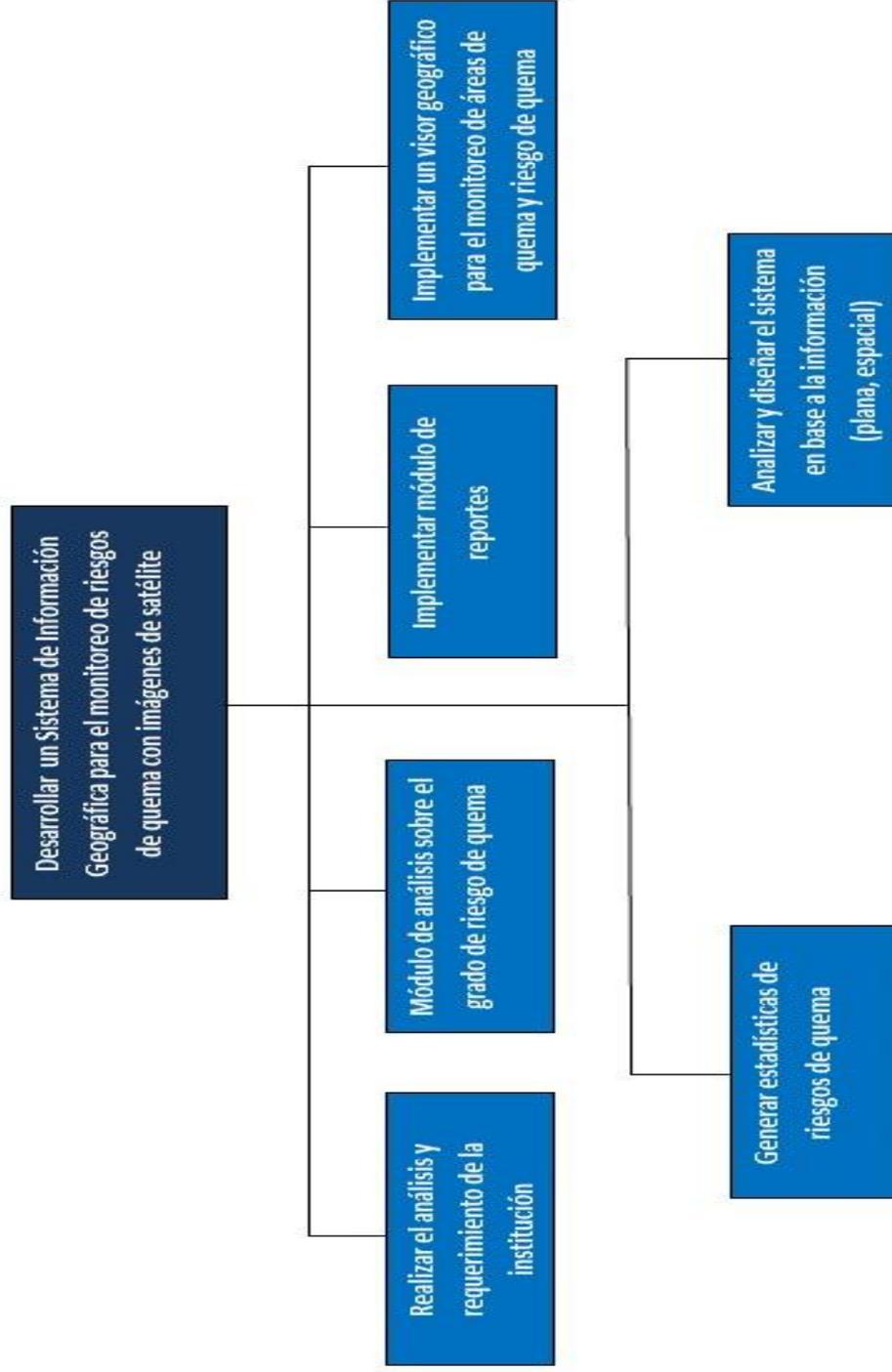
# **ANEXOS**



## Anexo A. Árbol de Problema



## Anexo B. Árbol de Objetivos





Autoridad de Fiscalización y  
Control Social de Bosques y Tierra

**MANUAL DE USUARIO DEL SISTEMA WEB DE  
MONITOREO DE RIESGO DE QUEMA - ABT**



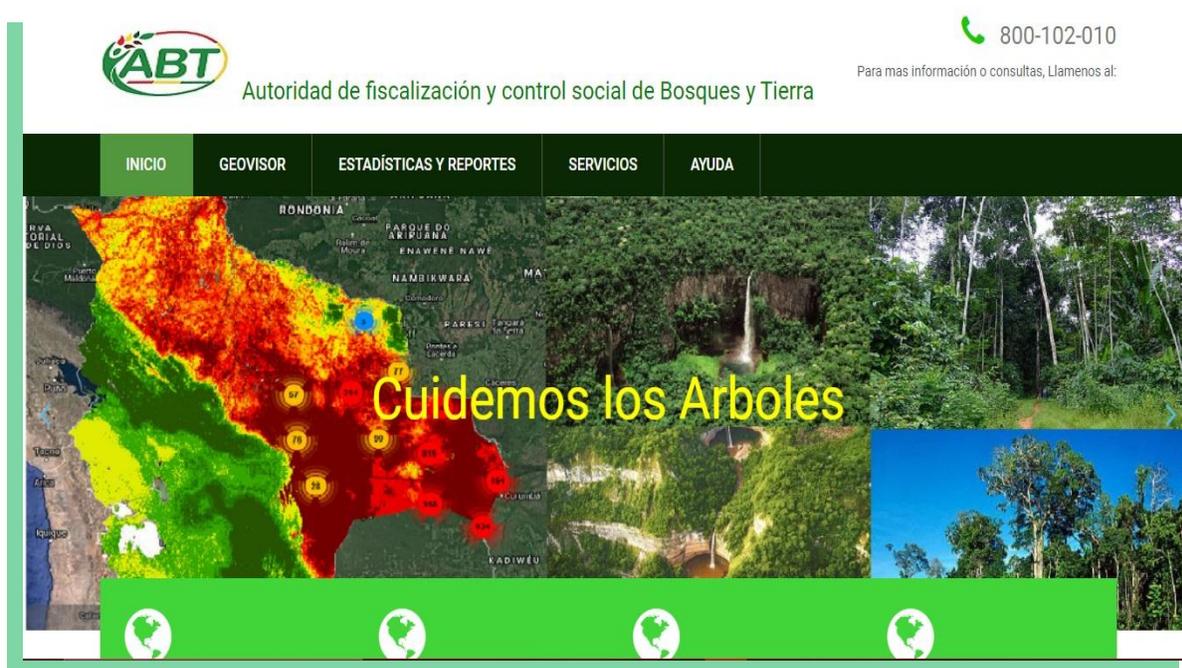
Provincia	Municipio	Muy Bajo	Bajo	Medio	Alto	Muy Alto	Extremo
Cercado	San Javier	93.076	5.77	0.868	0.21	0.073	0.003
Cercado	Trinidad	94.226	5.478	0.27	0.013	0.013	0
General Jos? Ballivi? n	Puerto Menor de Rurrenabaque	88.046	8.905	2.406	0.441	0.163	0.038
General Jos? Ballivi? n	San Borja	95.185	4.519	0.277	0.019	0	0
General Jos? Ballivi? n	Reyes	93.987	4.323	1.347	0.246	0.092	0.004
General Jos? Ballivi? n	Santa Rosa	94.422	3.974	1.218	0.264	0.116	0.005

## 1. ¿QUÉ ES EL SISTEMA WEB DE MONITOREO DE RIESGO DE QUEMA?

Es una herramienta, en la cual nos permite obtener información, reportes y estadísticas sobre las áreas de quema o incendio en Bolivia.

## 2. MODO DE ACCESO AL SISTEMA WEB

Primeramente, ingresamos a la página o sistema web de monitoreo de riesgo de quema, en la cual nos mostrara la página principal.



## 3. VISIÓN GENERAL DEL SISTEMA WEB

La página principal se divide en 4 zonas las cuales son:

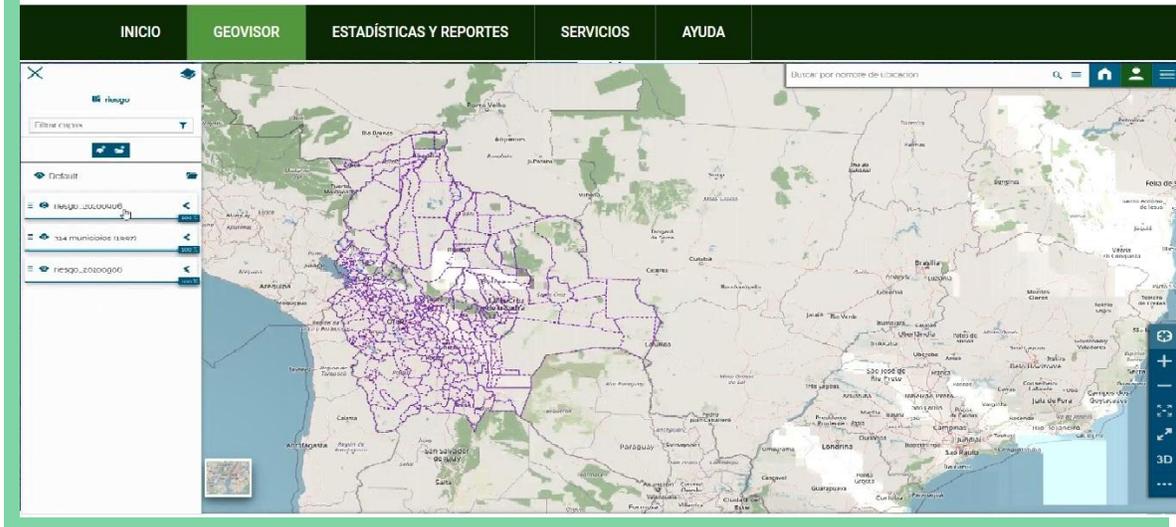
- a. **Encabezado:** Aquí se observa en el lado izquierdo el logo de la ABT y en el lado derecho el número telefónico para consultas.



b. **Menú:** Se observa 5 secciones: Inicio, Geovisor, Estadísticas y Reportes, Servicios y Ayuda.



- **Geovisor:** Se aprecia el visor de mapas, en la cual nos muestra las áreas de riesgo de quema en Bolivia.



En el visor de mapas nos muestra herramientas las cuales son:

**A1. Signos Más y Menos:** nos sirven para alejar y acercarnos a la imagen satelital.

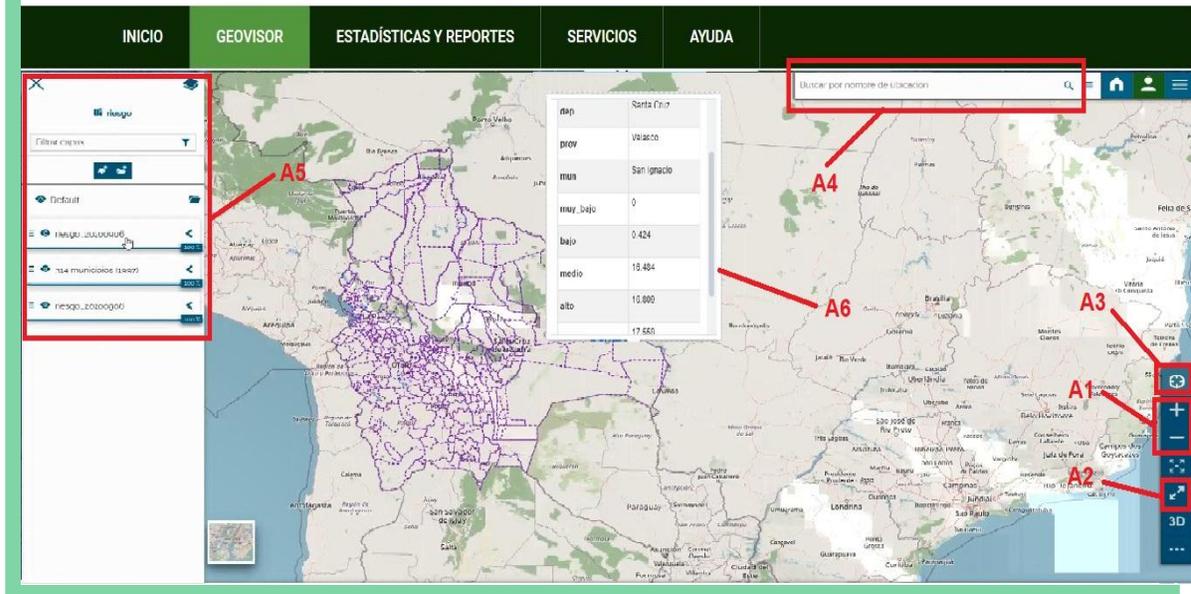
**A2. Medición de longitud:** ayuda a medir la distancia entre los puntos marcados dentro de la imagen satelital, mostrando la medida de forma acumulativa.

**A3. Geo localización:** nos ayuda a que el usuario pueda ver su posición sobre el mapa.

**A4. Búsqueda:** ayuda a realizar una búsqueda de forma eficaz para obtener resultados correctos, esto quiere decir que podemos realizar una búsqueda de departamento, provincia o municipio, para obtener datos sobre el grado de riesgo de quema.

**A5. Capas:** se apreciara las capas que tiene el visor de mapa.

**A6. Tabla de grado de riesgo de quema:** nos mostrara en una tabla los datos de riesgo de quema, sobre el lugar elegido.



- **Estadísticas y Reportes:** Se muestra los reportes y graficas estadísticas sobre las áreas de riesgo de quema en Bolivia.

ESTADÍSTICAS Y REPORTES

Seleccione Fecha de Riesgo de Quema

riesgo\_20200423

Seleccione la tabla de riesgo

Por favor seleccione un Departamento

Seleccione el Departamento

BENI

Seleccione el Departamento riesgo

MOSTRAR

- **Seleccionar Fecha de Riesgo de Quema:** Debe elegir en los siguientes cuadros, primeramente la **Fecha de riesgo de quema** que desea saber.

Seleccione Fecha de Riesgo de Quema

riesgo\_20200423

Seleccione la tabla de **riesgo**

Seleccione el Departamento

BENI

Seleccione el Departamento **riesgo**

**MOSTRAR**

- **Seleccionar Departamento:** Debe seleccionar el Departamento, de la cual desea saber los datos estadísticos de riesgo de quema.

Seleccione Fecha de Riesgo de Quema

riesgo\_20200423

Seleccione la tabla de **riesgo**

Seleccione el Departamento

BENI

Seleccione el Departamento **riesgo**

**MOSTRAR**

- **Mostrar:** Luego de realizar la selección de fecha y departamento de riesgo de quema, presionar el botón **Mostrar**.

Seleccione Fecha de Riesgo de Quema

riesgo\_20200423

Seleccione la tabla de **riesgo**

Seleccione el Departamento

BENI

Seleccione el Departamento **riesgo**

**MOSTRAR**

- **Datos Estadísticos de los Riesgos de quema:** Cuando se presiona el botón de **Mostrar**, se muestra los datos consultados en pantalla, como se aprecia en la siguiente imagen.



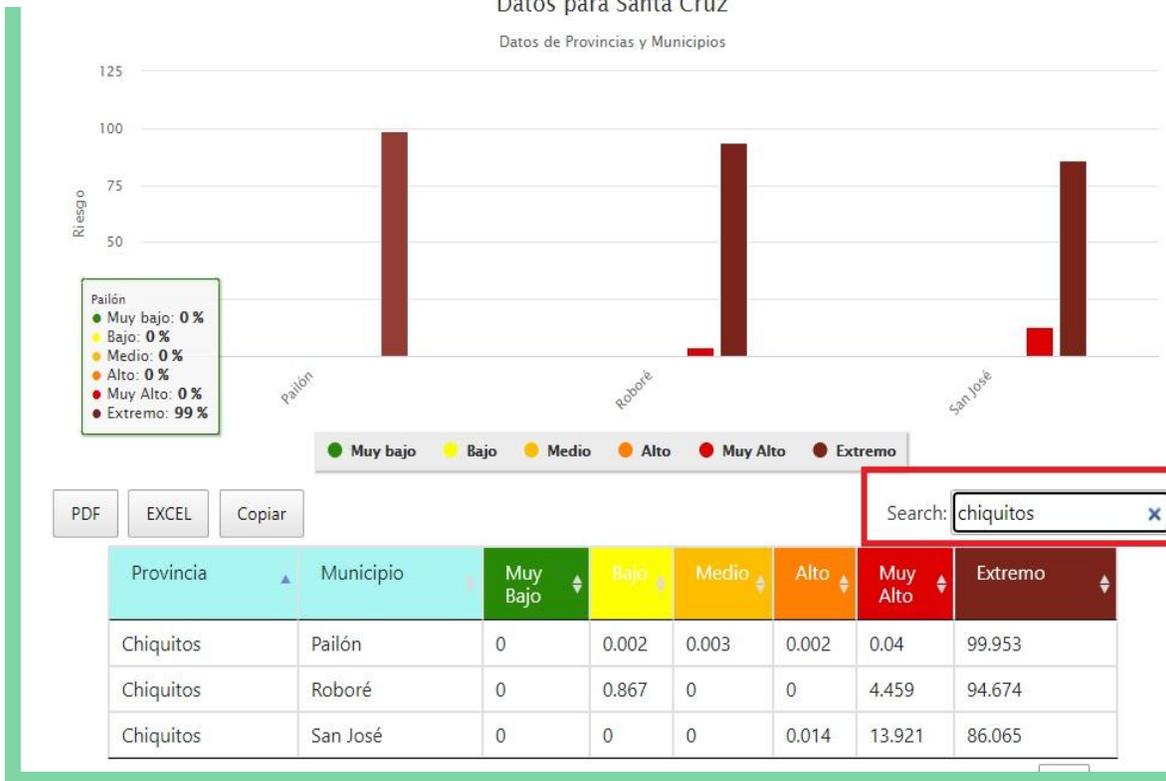
En la parte inferior vera la siguiente imagen.

PDF	Excel	Copiar	Search: <input type="text"/>				
Provincia	Municipio	Muy Bajo	Bajo	Medio	Alto	Muy Alto	Extremo
Cercado	San Javier	93.076	5.77	0.868	0.21	0.073	0.003
Cercado	Trinidad	94.226	5.478	0.27	0.013	0.013	0
General Jos? Ballivi? n	Puerto Menor de Rurrenabaque	88.046	8.905	2.406	0.441	0.163	0.038
General Jos? Ballivi? n	San Borja	95.185	4.519	0.277	0.019	0	0
General Jos? Ballivi? n	Reyes	93.987	4.323	1.347	0.246	0.092	0.004
General Jos? Ballivi? n	Santa Rosa	94.422	3.974	1.218	0.264	0.116	0.005

- Buscador (Search):** Para realizar la búsqueda de Riesgo de Quema de una Provincia o Municipio, se debe dirigir al siguiente campo llamado **Search**, ubicado en la parte esquina superior de la tabla.

PDF	EXCEL	Copiar	Search: <input type="text" value="chiquitos"/>				
Provincia	Municipio	Muy Bajo	Bajo	Medio	Alto	Muy Alto	Extremo
Chiquitos	Pailón	0	0.002	0.003	0.002	0.04	99.953
Chiquitos	Roboré	0	0.867	0	0	4.459	94.674
Chiquitos	San José	0	0	0	0.014	13.921	86.065

La búsqueda se realiza en la gráfica como en la tabla, como se aprecia en la siguiente imagen.



- **Realizar descarga de reporte:** Ya elegido el Departamento, la Provincia o Municipio, para realizar una descarga de lo que se aprecia en pantalla, la cual existe tres tipos de archivos para su respectiva descarga:
- **PDF:** **1.** Muestra los datos del Departamento seleccionado, **2.** Muestra Fecha de reporte y Fecha para que día se está sacando el reporte, **3.** Se muestra la tabla de riesgo de quema.

1 **Reporte de Riesgo de Quema de Santa Cruz**

2 **Reporte de Riesgo de Quema de la fecha: 2020-09-09 para la fecha: 2020-11-17**

3

Provincia	Municipio	Muy Bajo	Bajo	Medio	Alto	Muy Alto	Extremo
Chiquitos	Pailón	0	0.002	0.003	0.002	0.04	99.953
Chiquitos	Roboré	0	0.867	0	0	4.459	94.674
Chiquitos	San José	0	0	0	0.014	13.921	86.065

De un total de registros: 56

- **EXCEL:** 1. Muestra los datos del Departamento seleccionado, 2. Muestra Fecha de reporte y Fecha para que día se está sacando el reporte, 3. Se muestra la tabla de riesgo de quema.

	A	B	C	D	E	F	G	H	I
1		Reporte de Riesgo de Quema de Santa Cruz							1
2		Reporte de Riesgo de Quema de la fecha: 2020-09-09 para la fecha: 2020-11-17							2
3	Provincia	Municipio	Muy Bajo	Bajo	Medio	Alto	Muy Alto	Extremo	
4	Chiquitos	Pailón	0	0.002	0.003	0.002	0.04	99.953	
5	Chiquitos	Roboré	0	0.867	0	0	4.459	94.674	3
6	Chiquitos	San José	0	0	0	0.014	13.921	86.065	
7		De un total de registros: 56							

- **COPIA:** Se realiza una copia de los datos de la tabla en papelera, la cual puede ser, pegado en un Bloc de Notas o Word.

Luego debe ir a presionar el botón, **PDF, EXCEL o COPIAR**, puede elegir el tipo de archivo que desea obtener.

● Muy bajo ● Bajo ● Medio ● Alto ● Muy Alto ● Extremo

PDF EXCEL Copiar Search: chiquitos

Provincia	Municipio	Muy Bajo	Bajo	Medio	Alto	Muy Alto	Extremo
Chiquitos	Pailón	0	0.002	0.003	0.002	0.04	99.953
Chiquitos	Roboré	0	0.867	0	0	4.459	94.674
Chiquitos	San José	0	0	0	0.014	13.921	86.065

Para la descarga de los reportes, se puede realizar la

- **Servicios:** Se muestra la geoinformación que se utilizó para la base de datos, también se muestra los servicios web que se utilizaron para la realización del sistema.



- **Ayuda:** Se muestra una guía corta sobre la página.



## Ayuda de Geovisor

Geovisor es un sistema de información geográfica gratuito que ofrece la institución de Autoridad de fiscalización y control social de Bosques y Tierra - ABT. La página está diseñada para ser útil a los usuarios que necesitan una herramienta intuitiva y fácil de utilizar para tratar el riesgo de quema en Bolivia.

### Estructura de la página web

- En el menú de "Inicio", tenemos la presentación de la página web, luego mas abajo tenemos algunos conceptos que nos pueden ayudar a comprender mejor el riesgo de quema, en los conceptos esta: ¿Que es riesgo?, ¿Que es quema?, ¿Que es riesgo de quema o incendio? y Tipos o clasificación de riesgo de quema o incendio.
- En el menú de "Geovisor", tenemos el visor de mapas, donde podemos apreciar el riesgo de quema toda Bolivia, departamento, provincia y municipio.
- En el menú de "Servicios", se muestra "Servicios Web (WMS)" y "Base de datos espacial", en ambas divisiones podemos ver link, la cual nos llevan a los materiales que se usaron, para la realización del geovisor.

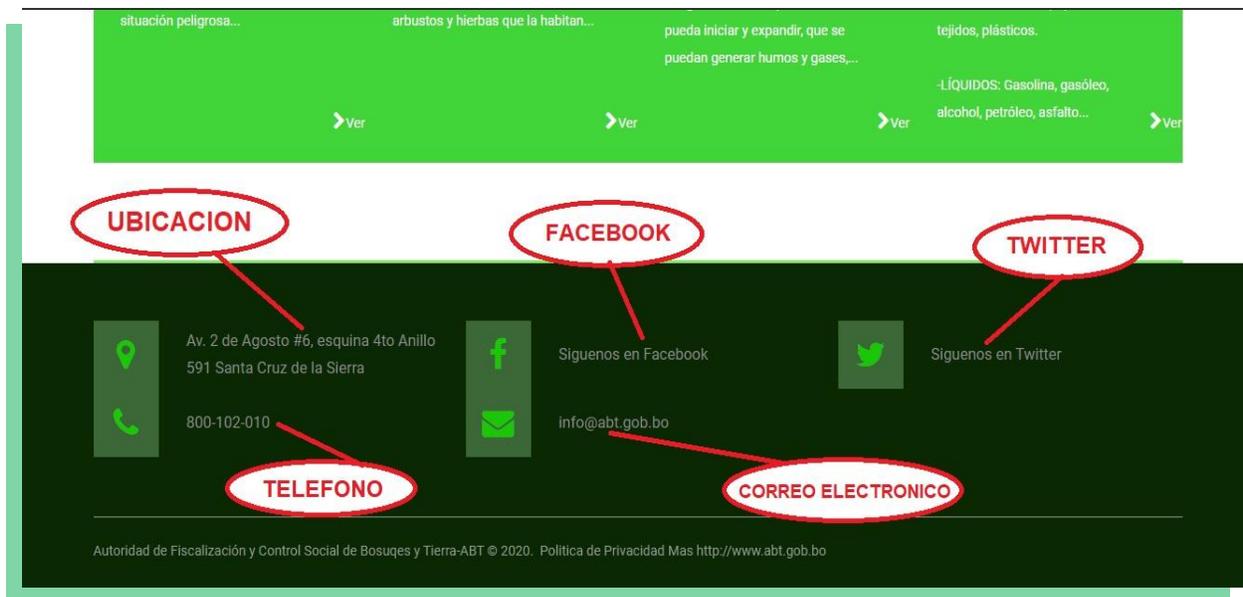


c. **Zona central:** Muestra el contenido de la página web, y también muestra conceptos pequeños referidos a las áreas de riesgo de quema.

### Cuidemos los Arboles

 <b>Riesgo</b> Es una proximidad de una medida de la magnitud de los daños frente a una situación peligrosa... <a href="#">Ver</a>	 <b>Quema</b> Es la deforestación de un pedazo de tierra donde se queman los árboles, arbustos y hierbas que la habitan... <a href="#">Ver</a>	 <b>Riesgo de quema o incendio</b> Peligro relativo de que un incendio se pueda iniciar y expandir, que se puedan generar humos y gases... <a href="#">Ver</a>	 <b>Tipos de riesgos de quema o incendio</b> -SÓLIDOS: Madera, papel, carbón tejidos, plásticos. -LÍQUIDOS: Gasolina, gasóleo, alcohol, petróleo, asfalto... <a href="#">Ver</a>
---	---	---	--

d. **Barra inferior:** Aquí se observa todas las redes sociales de la institución de Autorización y Fiscalización de Control Social de Tierra y Bosque: Facebook, Twitter, Correo Electrónico y Dirección.



#### 4. AUTENTICACIÓN COMO USUARIO ADMINISTRADOR

Para el acceso a Usuario Administrador, se va al menu de **Login**, en la cual nos mostrara la siguiente pantalla.

**ABT** Autoridad de fiscalización y control social de Bosques y Tierra

INICIO    GEOVISOR    ESTADÍSTICAS Y REPORTE    SERVICIOS    AYUDA    **LOGIN**

Acceder

Correo electrónico    admin@gmail.com

Contraseña    .....

Recuérdame

**ACCEDER**

Donde tendremos el formulario de inicio de sesion, en la cual se llena los siguientes campos: correo electronico y contraseña, luego precionamos el boton **Acceder**.

- Correo Electronico
- Contraseña

### REGISTRO DE TABLAS

Nombre de la tabla

Escriba el nombre de la tabla de riesgo

**REGISTRAR**

Search:

#	Nombre Tabla	Fecha de registro	Acciones
1	riesgo_20200423	2020-11-04	<input type="button" value="EDITAR"/>
2	riesgo_20200906	2020-11-07	<input type="button" value="EDITAR"/>
3	riesgo_20200907	2020-11-07	<input type="button" value="EDITAR"/>
4	riesgo_20200908	2020-11-07	<input type="button" value="EDITAR"/>

**Nombre de tabla:** En este campo se realiza el registro de tablas, la cual colocamos el nombre de la tabla y presionamos el botón de **Registrar**, donde nos mostrara la siguiente pantalla.

### REGISTRO DE TABLAS

Nombre de la tabla

Escriba el nombre de la tabla de riesgo

**REGISTRAR**

Search:

#	Nombre Tabla	Fecha de registro	Acciones
1	riesgo_20200423	2020-11-04	<input type="button" value="EDITAR"/>
2	riesgo_20200906	2020-11-07	<input type="button" value="EDITAR"/>
3	riesgo_20200907	2020-11-07	<input type="button" value="EDITAR"/>
4	riesgo_20200908	2020-11-07	<input type="button" value="EDITAR"/>

Se muestra en pantalla las tablas que fueron registradas por el usuario Administrador, donde puede editar los nombres de las tablas, si en algún registro se equivocó al momento de registrar.

Para tal efecto presiona en el botón de **Editar**, la cual le mostrara la siguiente pantalla.

Copy	Excel	PDF	Search: <input type="text"/>
#	Nombre Tabla	Fecha de registro	Acciones
1	riesgo_20200423	2020-11-04	<div style="border: 2px solid red; padding: 5px; display: inline-block;">           EDITAR         </div>

Donde debe llenar un campo, que es el nombre de la tabla a editar, luego de presionar el botón editar se mostrara la siguiente pantalla, donde se realiza la edición del nombre de la tabla registrada.

### Editar Tabla

Nombre de la tabla

riesgo\_20200906

Escriba el nombre de la tabla de **riesgo**

CERRAR

ACTUALIZAR

Una vez editado el nombre de la tabla registrada debe ir al botón de **Actualizar**, Y la tabla se actualizara exitosamente.

INICIO
GEOVISOR
ESTADÍSTICAS Y REPORTES
SERVICIOS
AYUDA
ADMINISTRADOR ▾

La tabla se actualizo exitosamente! ×

## REGISTRO DE TABLAS

Nombre de la tabla

Nombre de la tabla

Escriba el nombre de la tabla de **riesgo**

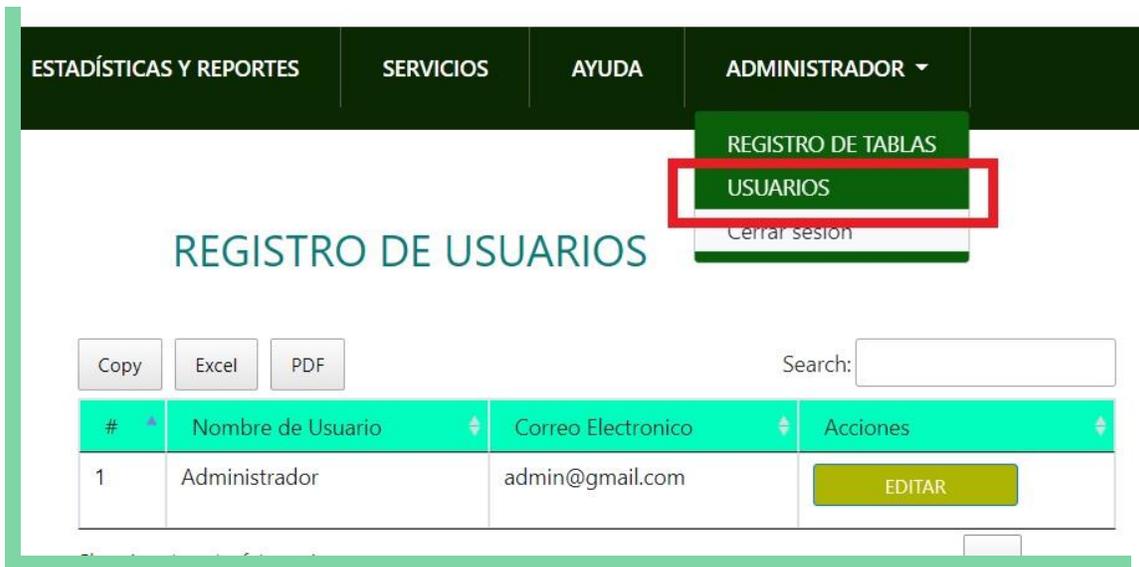
Copy Excel PDF

Search:

#	Nombre Tabla	Fecha de registro	Acciones
1	riesgo_20200423	2020-11-04	<div style="background-color: #4CAF50; color: white; padding: 5px; border: 1px solid gray;">EDITAR</div>

- **Sección de usuarios:** Es donde el administrador podrá registrar más usuarios, también es donde el Usuario administrador podrá editar: **Nombre de usuario, Correo Electrónico y Contraseña.**

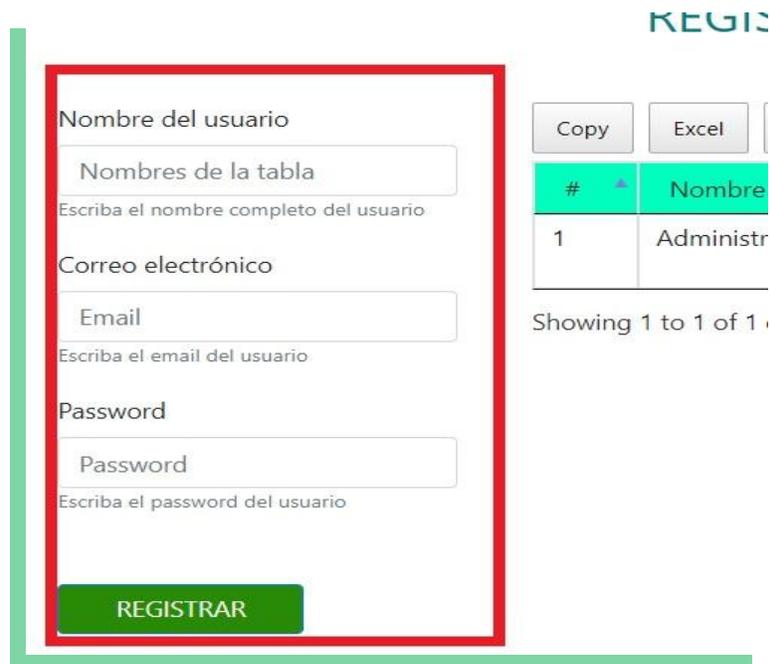
Para tal efecto debe ir al meno de **Administrador**, donde se desplegara dos submenús, la cual elegirá el menú de **Usuarios**.



Una vez que se ingrese al menú de **Usuarios**, se mostrara la siguiente pantalla.



- En el lado **izquierdo** observaremos los campos donde se realiza el registro de los nuevos usuarios. Se debe realizar el llenado de los siguientes campos: **Nombre de usuario**, **Correo electrónico y contraseña**. Una vez llenado los campos presionar el botón de **Registrar**, y se registrara exitosamente el nuevo usuario si todos los campos fueron llenados correctamente.



The image shows a user registration interface. On the left, a form is enclosed in a red border. It contains three input fields: 'Nombre del usuario' (with placeholder 'Nombres de la tabla' and instruction 'Escriba el nombre completo del usuario'), 'Correo electrónico' (with placeholder 'Email' and instruction 'Escriba el email del usuario'), and 'Password' (with placeholder 'Password' and instruction 'Escriba el password del usuario'). A green 'REGISTRAR' button is at the bottom of the form. To the right, there is a table with columns '#', 'Nombre', and 'Acciones'. The first row shows '1' and 'Administrador'. Above the table are 'Copy' and 'Excel' buttons. Below the table, it says 'Showing 1 to 1 of 1'.

#	Nombre	Acciones
1	Administrador	

- En el lado derecho se observa la tabla de los registros de los usuarios, donde se puede realizar la edición y eliminación de un usuario, el usuario con más privilegios es el **Usuario Administrador**, el cual puede realizar la edición y eliminación de algún usuario, en cambio los usuarios nuevos no pueden eliminar al usuario administrador.

## REGISTRO DE USUARIOS

Copy Excel PDF Search:

#	Nombre de Usuario	Correo Electronico	Acciones
1	Administrador	admin@gmail.com	<button>EDITAR</button>
2	Usuario tercero	userter@gmail.com	<button>EDITAR</button> <button>ELIMINAR</button>

Showing 1 to 2 of 2 entries Previous 1 Next

- Para realizar la edición de algún usuario se debe presionar el botón de **Editar**, la cual nos mostrara la siguiente pantalla.

### Editar Usuario

Nombre del usuario

Escriba el nombre completo del usuario

Correo electronico

Escriba el email del usuario

Password

Escriba el password del usuario

CERRAR ACTUALIZAR

Donde se debe realizar el llenado de los campos de: **Nombre de usuario**, **correo electrónico** y **contraseña**, para finalizar con la edición, presionar en el botón **Actualizar**.

- **Cerrar Sesión:** para cerrar sesión nos dirigimos al menú de Administrador donde se despliega los submenús, y es en esa parte donde se encuentra la opción de **Cerrar Sesión**, presionamos esa opción y listo, se finalizará la sesión iniciada.



La Paz – El Alto 17 de Noviembre de 2020

Señor  
Ing. David Carlos Mamani Quispe  
**DIRECTOR DE CARRERA INGENIERIA DE SISTEMAS**

Presente.

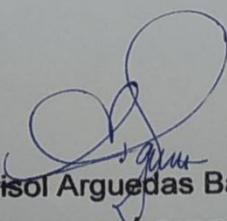
**REF: AVAL DE CONFORMIDAD**

Distinguido Ingeniero.

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado **“SISTEMAS DE INFORMACION GEOGRAFICAS APLICADO AL MONITOREO DE RIESGOS DE QUEMA CON IMÁGENES DE SATELITE”** Caso de Estudio: **Autoridad de Fiscalización y Control Social de Bosques y Tierra – ABT.** Que propone el estudiante postulante Katteryn Flavia Huanca Villanueva, con cedula de identidad 9945227 LP., para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.

  
Ing. Marisol Arguedas Balladares  
**TUTOR METODOLOGICO TALLER II**

## AVAL DE CONFORMIDAD

La Paz – El Alto Noviembre de 2020

Señor  
Ing. David Carlos Mamani Quispe  
**DIRECTOR DE CARRERA INGENIERIA DE SISTEMAS**

Presente.

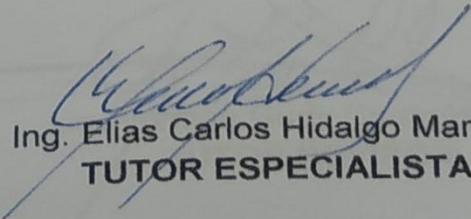
REF: AVAL DE CONFORMIDAD

Distinguido Ingeniero.

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado **"SISTEMAS DE INFORMACION GEOGRAFICAS APLICADO AL MONITOREO DE RIESGOS DE QUEMA CON IMÁGENES DE SATÉLITE"** Caso de Estudio: **Autoridad de Fiscalización y Control Social de Bosques y Tierra – ABT**. Que propone el estudiante postulante Katteryn Flavia Huanca Villanueva, con cedula de identidad 9945227 LP., para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.

  
Ing. Elías Carlos Hidalgo Mamani  
**TUTOR ESPECIALISTA**

## AVAL DE CONFORMIDAD

La Paz – El Alto 18 de Noviembre de 2020

Señor  
Ing. David Carlos Mamani Quispe  
DIRECTOR DE CARRERA INGENIERIA DE SISTEMAS

Presente.

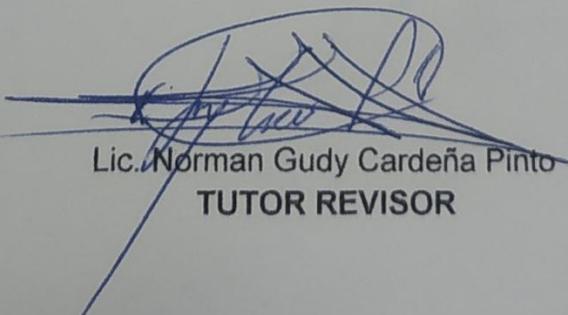
REF: AVAL DE CONFORMIDAD

Distinguido Ingeniero.

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado "SISTEMAS DE INFORMACION GEOGRAFICAS APLICADO AL MONITOREO DE RIESGOS DE QUEMA CON IMÁGENES DE SATÉLITE" Caso de Estudio: Autoridad de Fiscalización y Control Social de Bosques y Tierra – ABT. Que propone el estudiante postulante Katteryn Flavia Huanca Villanueva, con cedula de identidad 9945227 LP., para su defensa pública, evaluación correspondiente a la materia de Taller de Licenciatura II, de acuerdo al reglamento vigente de la carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.



Lic. Norman Gudy Cardeña Pinto  
TUTOR REVISOR



Estado Plurinacional de Bolivia



Santa Cruz, 19 de noviembre del 2020

**Señor(a)**

**A QUIEN CORRESPONDA**

Presente.-

**Ref. : AVAL DE CONFORMIDAD**

De mi mayor consideración:

Tengo a bien dirigirme a su persona para comunicarle mi conformidad a la implementación y puesto en marcha con las pruebas necesarias, en el departamento de sistemas del PROYECTO DE GRADO, titulado "**SISTEMA DE INFORMACIÓN GEOGRÁFICA APLICADA AL MONITOREO DE QUEMA CON IMÁGENES DE SATELITE**" caso: **AUTORIDAD DE FISCALIZACIÓN Y CONTROL SOCIAL DE BOSQUES Y TIERRAS** elaborado por el universitario **Katteryn Flavia Huanca Villanueva**, con R.U. **15000117**, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Sin otro particular, me despido.

Atentamente,

**Tomas Tapia Suño**  
AREA INFORMATICA  
UNJP/045/BOL