

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

MÓDULO FINANCIERO PARA EL SISTEMA DE REFORESTACIÓN Y FORESTACIÓN

CASO: PROYECTO UNJP/BOL/045/UNJ

Para optar al título de Licenciatura en Ingeniería de Sistemas
MENCIÓN: MENCIÓN DE INFORMÁTICA Y COMUNICACIONES

Postulante: Calixto Torrez Uruña
Tutor Metodológico: Ing. Enrique Flores Baltazar
Tutor Especialista: Ing. Roberto Camilo Choque Apaza
Tutor Revisor: Lic. Rosa Patricia Nina Chura

EL ALTO – BOLIVIA
2020

Dedicatoria:

Este proyecto fue realizado con mucha dedicación y esfuerzo, es dedicado a mi familia, más que todo a mis padres Cirilo Torrez Quispe y Francisca Uruña Callisaya, por darme la vida, por quererme tanto y confiar en mí, quienes son la principal inspiración de superación y fortaleza en mi vida.

Agradecimiento

A dios por permitirme llegar hasta estos momentos de mi vida, por los triunfos y los momentos más difíciles que me han enseñado a valorar cada día.

Agradecer a mi familia, a mis padres por el apoyo incondicional que me dieron dándome salud y educación.

A mi tutor Ing. Roberto Choque Camilo Apaza quien me ha orientado en realización en todo momento.

A mi asesor Ing. Enrique Flores Baltazar por su dedicación de tiempo, por sus correcciones, aportes y paciencia que ha tenido, fueron importantes para la finalización del proyecto.

Habiendo una mención agradecer a la Lic. Rosa Patricia Nina Chura, por el apoyo y la colaboración para la conclusión del proyecto

A la Universidad Pública de El Alto de la carrera de Ingeniería de Sistemas por acogerme y ver culminar mis estudios superiores, ha sido como la segunda casa.

RESUMEN

El desarrollo de este proyecto se estructura a partir del marco referencial, donde se da a conocer los antecedentes, las problemáticas que existen en la institución, las necesidades de desarrollar un módulo financiero para el sistema de forestaciones y reforestaciones caso, proyecto unjp/bol/045/unj, además de los objetivos, justificaciones y entre otros que están descritos en el acápite.

En el desarrollo del capítulo II, se hace referencia al marco teórico el cual es una base principal que sustenta la investigación por lo cual fueron seleccionados los aspectos más importantes, estos son: metodología de desarrollo UWE juntamente con el lenguaje modelado UML y las herramientas de desarrollo de software.

Por las características del proyecto en el marco aplicativo, el desarrollo del proyecto está basado bajo la metodología de desarrollo software UWE, que propone sus diferentes fases para el ciclo de vida del proyecto.

Se desarrolló los controles de calidad de la siguiente manera, en calidad del software se empleó la norma internacional ISO 9126, en cuanto para los cálculos de costos el modelo de costos cocomo II y la seguridad del sistema está basado en la ISO 27000 (Sistema de Gestión de Seguridad de la Información).

Se implementó el módulo financiero para el sistema de forestaciones y reforestaciones con éxito, con la realización de las pruebas necesarias en el funcionamiento, se verifico el cumplimiento de los objetivos propuestos.

Palabras claves: módulo, financiero, UWE, forestaciones y reforestaciones.

ABSTRACT

The development of this project is structured based on the referential framework, where the background, the problems that exist in the institution, the needs to develop a financial module for the afforestation and reforestation system, case, project unjp/bol/045/unj, in addition to the objectives, justifications and among others that are described in the section.

In the development of chapter II, reference is made to the theoretical framework which is a main basis that supports the research, for which the most important aspects were selected, these are: UWE development methodology together with the UML modeling language and the tools of software development.

Due to the characteristics of the project in the application framework, the development of the project is based on the UWE software development methodology, which proposes its different phases for the project life cycle.

The quality controls were developed as follows, as the software quality the international standard ISO 9126 was used, as for the cost calculations the cocomo II cost model and the security of the system is based on ISO 27000 (Management System Information Security).

The financial module for the afforestation and reforestation system was successfully implemented, with the performance of the necessary tests in operation, the fulfillment of the proposed objectives was verified.

Keywords: module, financial, UWE, afforestation and reforestation.

INDICE

CAPITULO I	1
1. MARCO PRELIMINAR	1
1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES	2
1.3 PLANTEAMIENTO DEL PROBLEMA	7
1.3.1 PROBLEMA PRINCIPAL	7
1.3.2 PREGUNTA DE LA INVESTIGACIÓN	8
1.4 OBJETIVOS	8
1.4.1 OBJETIVO GENERAL	8
1.4.2 OBJETIVOS ESPECÍFICOS	8
1.5 JUSTIFICACIÓN	8
1.5.1 JUSTIFICACIÓN TÉCNICA	8
1.5.2 JUSTIFICACIÓN ECONÓMICA	9
1.5.3 JUSTIFICACIÓN SOCIAL	9
1.6 METODOLOGIA	9
1.6.1 MÉTODO CIENTÍFICO	9
1.6.2 METODOLOGIA DE DESARROLLO	10
1.6.3 METRICA DE CALIDAD	11
1.6.4 METRICA DE COSTOS	11
1.7 HERRAMIENTAS DE DESARROLLO	13
1.8 LIMITES Y ALCANCES	14
1.8.1 LIMITES	14
1.8.2 ALCANCES	14
1.9 APORTES	14
CAPITULO II	15
2. MARCO TEORICO	15
2.1 INTRODUCCIÓN	15
2.2 CONCEPTOS BÁSICOS	15
2.3 MÓDULO FINANCIERO	23
2.3.1 REPORTES FINANCIEROS	24
2.3.2 TIPOS DE REPORTES	24

2.4	INGENIERÍA DE SOFTWARE	25
2.4.1	EL PROCESO DE SOFTWARE	26
2.4.2	INGENIERIA WEB	28
2.5	METODOLOGÍA DE DESARROLLO DE SOFTWARE	29
2.5.1	METODOLOGÍA DE DESARROLLO TRADICIONAL	31
2.5.1.1	MODELO ENCASCADA	31
2.5.1.2	MODELO ITERATIVO	34
2.5.1.3	MODELO EVOLUTIVO	35
2.5.2	METODOLOGÍAS ÁGILES	36
2.6	METODOLOGÍA UWE	37
2.6.1	UWE Y SU RELACION CON UML	37
2.6.2	ACTIVIDADES DEL MODELADO UWE	38
2.6.3	FASES DE LA UWE	44
2.7	MÉTRICAS DE CALIDAD	46
2.7.1	NORMA ISO 9126	48
2.8	MÉTRICAS DE COSTOS	57
2.8.1	COCOMO II	57
2.9	SISTEMA DE INFORMACIÓN GEOGRÁFICA	61
2.10	TECNOLOGÍAS WEB	63
2.10.1	PATRONES DE DISEÑO	63
2.10.2	LENGUAJE DE PROGRAMACION	65
2.10.3	FRAMEWORK	66
2.10.4	BASE DE DATOS	67
CAPITULO III		68
3.	MARCO APLICATIVO	68
3.1	INTRODUCCIÓN	68
3.2	ESQUEMA DEL SISTEMA	68
3.3	METODOLOGÍA UWE	69
3.3.1	MODELO DE CASOS DE USO	70
3.3.1.1	DESCRIPCION DE CASOS DE USO	70
3.3.1.2	CASO DE USO GENERAL	71
3.3.1.3	ESPECIFICACIÓN DE CASOS DE USO	72
3.3.2	DIAGRAMA DE SECUENCIAS	79

3.3.3	DIAGRAMA DE ACTIVIDADES	82
3.3.4	MODELO DE CONTENIDO.....	83
3.3.4.1	DISEÑO CONCEPTUAL.....	83
3.3.5	MODELO DE NAVEGACIÓN	87
3.3.6	MODELO DE PRESENTACIÓN	88
3.4	DESARROLLO DEL SISTEMA	90
3.4.1	ESTRUCTURA MVC.....	91
3.4.2	FICHEROS.....	92
3.4.3	PANTALLAS DE SISTEMA.....	95
3.4.4	MODELO DE BASE DE DATOS	98
3.4.5	DICCIONARIO DE DATOS.....	102
4.	CONTROL DE CALIDAD IV.....	112
4.1	MÉTRICAS DE CALIDAD DE SOFTWARE.....	112
4.1.1	NORMA ISO 9126.....	112
4.1.2	FUNCIONALIDAD.....	112
4.1.3	FIABILIDAD	116
4.1.4	USABILIDAD	117
4.1.5	MANTENIBILIDAD.....	118
4.1.6	PORTABILIDAD	119
4.1.7	CALIDAD GLOBAL	119
4.2	EVALUACIÓN DE COSTOS	120
4.2.1	MODELO COCOMO II.....	120
4.2.2	COSTO DEL SISTEMA.....	121
4.2.2.1	COSTO DE DESARROLLO.....	121
4.2.2.2	COSTO ELABORACIÓN DEL PROYECTO.....	123
4.2.3	COSTO TOTAL DEL SISTEMA.....	123
CAPITULO V		125
5.	PRUEBAS Y RESULTADOS	125
5.1	PRUEBAS.....	125
5.1.1	SEGURIDAD DEL SISTEMA.....	127
5.1.2	ENCRIPCIÓN	127
5.2	RESULTADOS.....	128
CAPITULO VI.....		130

6. CONCLUSIONES Y RECOMENDACIONES	130
6.1 CONCLUSIONES	130
6.2 RECOMENDACIONES	131
BIBLIOGRAFÍA	132
WEBGRAFÍA	133
ANEXOS	136
ÁRBOL DE PROBLEMAS	136
ÁRBOL DE OBJETIVOS	137
ORGANIGRAMA DE SISTEMA DEL PROYECTO UNJP/BOL/045/UNJ	138
ENCUESTA SOBRE LA USABILIDAD DEL SISTEMA	139
MANUAL DE USUARIO	140

INDICE DE FIGURAS

	Pág.
<i>Figura 1. 1: Propuesta de esquema conceptual del sistema</i>	4
<i>Figura 2.1: Capas de la ingeniería de software.....</i>	26
<i>Figura 2. 2: Fases del modelo encascada</i>	33
<i>Figura 2. 3: Fases del modelo iterativo</i>	34
<i>Figura 2. 4: Fases del modelo evolutivo</i>	36
<i>Figura 2. 5: Modelos de UWE.....</i>	39
<i>Figura 2. 6: Modelado de casos de uso</i>	40
<i>Figura 2. 7: Modelo de contenido</i>	41
<i>Figura 2. 8: Modelo de navegacion.....</i>	42
<i>Figura 2. 9: Modelo de presentación.....</i>	43
<i>Figura 2. 10: Modelo de proceso</i>	44
<i>Figura 2. 11: Fases de UWE</i>	46
<i>Figura 2. 12: Características y sub-características del modelo ISO 9126.....</i>	49
<i>Figura 2. 13: Características del modelo de calidad en uso</i>	56
<i>Figura 2. 14: Concepto operacional cocomo II</i>	59
<i>Figura 2. 15 Dimensión espacial de los datos en un SIG.....</i>	63
<i>Figura 2. 16: Modelo de patrones de diseño.....</i>	64
<i>Figura 3. 1: Esquema del sistema.....</i>	69
<i>Figura 3. 2: Diagrama de casos de uso general.....</i>	71
<i>Figura 3. 3: Casos de uso datos de plantaciones.....</i>	73
<i>Figura 3. 4: Casos de uso visualización de proyectos con y sin ubicación.</i>	74
<i>Figura 3. 5: Casos de uso reportes estadísticos</i>	76
<i>Figura 3. 6: Casos de uso reportes</i>	77
<i>Figura 3. 7: Diagrama de secuencias autenticación de usuarios</i>	79
<i>Figura 3. 8: Diagrama de secuencias de plantaciones y proyectos.....</i>	80
<i>Figura 3. 9: Diagrama de secuencias de gráficos estadísticos y reportes.....</i>	81
<i>Figura 3. 10: Diagrama de actividades.....</i>	82
<i>Figura 3. 11: Ilustración del diseño conceptual parte 1.....</i>	83
<i>Figura 3. 12: Ilustración del diseño conceptual parte 2.....</i>	84
<i>Figura 3. 13: Ilustración del diseño conceptual parte 3.....</i>	85
<i>Figura 3. 14: Ilustración del diseño conceptual general.....</i>	86
<i>Figura 3. 15: Modelo de navegación</i>	87
<i>Figura 3. 16: Diagrama de presentación, pagina madre del módulo.....</i>	88
<i>Figura 3. 17: Diagrama de presentación, modulo principal.....</i>	89
<i>Figura 3. 18: Actividad realizada en el desarrollo de sistema</i>	90
<i>Figura 3. 19: Arquitectura MVC.....</i>	92
<i>Figura 3. 20: Ficheros de Modelo y controlador</i>	93
<i>Figura 3. 21: Archivos de vista.....</i>	94
<i>Figura 3. 22: Pantalla principal del módulo</i>	95
<i>Figura 3 23: Menú de autenticación del usuario.....</i>	97
<i>Figura 3 24. Modelo de base de datos parte 1.....</i>	98
<i>Figura 3. 25: Modelo de base de datos parte 2</i>	99
<i>Figura 3. 26: Modelo de base de datos parte 3</i>	100

<i>Figura 3. 27: Modelo general de la base de datos</i>	<i>101</i>
<i>Figura 5. 1: Pruebas de Caja negra</i>	<i>125</i>
<i>Figura 5. 2: Prueba de autenticación de usuario.....</i>	<i>126</i>
<i>Figura 5. 3: Pruebas de validación de formulario.....</i>	<i>126</i>
<i>Figura 5. 4: Función hash.....</i>	<i>128</i>
<i>Figura 5. 5: Esquema funcional del módulo financiero</i>	<i>129</i>

ÍNDICE DE TABLAS

	Pág.
Tabla 2.1: Comparación de metodologías de desarrollo	30
Tabla 2. 2: Sub-características de normas ISO 9126	51
Tabla 2. 3: Modelo de calidad del producto software para la calidad en uso.	56
Tabla 2. 4: Distribución del mercado de software actual y futuro	58
Tabla 3. 1: Tabla de descripción de casos de uso.....	70
Tabla 3. 2: Tabla descripción de caso de uso autenticación	72
Tabla 3. 3: Tabla generar datos de plantaciones	73
Tabla 3. 4: Tabla obtener datos de proyectos sin y con ubicación geográfica.....	75
Tabla 3. 5: Tabla caso de uso reportes estadísticos	76
Tabla 3. 6: Tabla de casos de uso reportes	78
Tabla 3. 7: Tabla de caso de uso generar datos sin autenticación.	78
Tabla 3. 8: Tabla de usuarios.....	102
Tabla 3. 9: Tabla tipo de plantaciones	103
Tabla 3. 10: Tipo de empleo	103
Tabla 3. 11: Departamentos	103
Tabla 3. 12: Tabla de roles para usuarios	104
Tabla 3. 13: Responsable.....	104
Tabla 3. 14: Proyecto ubicación.....	104
Tabla 3. 15: Proyecto nueva comunidad	105
Tabla 3. 16: Proyecto empleo	105
Tabla 3. 17: Proyectos	105
Tabla 3. 18: Procedencia plantas estacas.....	107
Tabla 3. 19: Planificaciones de ubicación	107
Tabla 3. 20: Planificaciones.....	107
Tabla 3. 21: Personas.....	108
Tabla 3. 22: Periodo de plantaciones	109
Tabla 3. 23: Municipios	109
Tabla 3. 24: Instituciones	109
Tabla 3. 25: Entidad solicitante	110
Tabla 3. 26: Cuencas.....	110
Tabla 3. 27: Comunidades	110
Tabla 3. 28: Provincias.....	111
Tabla 4. 1: Factores de ponderación.....	113
Tabla 4. 2: Ajuste de complejidad.....	114
Tabla 4. 3: Encuesta sobre la usabilidad del sistema	117
Tabla 4. 4: Información requerida por el IMS	118
Tabla 4. 5: Calidad global del sistema.....	120
Tabla 4. 6: Constantes a,b,c,d COCOMO II.....	121
Tabla 4. 7: Muestra de relación para convertir el valor de PF a KLDC.....	122
Tabla 4. 8: Costos de recursos empleados para la elaboración del sistema.....	123
Tabla 4. 9: Costos total del sistema.....	124



CAPITULO I

MARCO PRELIMINAR

1.1 INTRODUCCIÓN

Con el avance de la tecnología a pasos agigantados en la actualidad, los sistemas financieros, sistemas de información, académicos entre otras, han sido de vital importancia para la sistematización de distintas áreas, uno de los cuales es en manejo de recursos económicos, los sistemas contables o financieros son de gran ayudada para instituciones o empresas, donde se les facilita la generación de informes automáticos, reportes necesarios, datos de inventarios y otros que se requieren. (Kit, 2019)

Módulo financiero, recibe como entrada, la salida que haya proporcionado otro módulo o los datos de entrada al sistema, en el proceso de gestión de presupuestos, gestión de cuentas, gestión de tesorería entre otras, son los módulos que facilitan las distintas actividades de planear, administrar, controlar y gestionar presupuestos de recursos económicos, permitiendo que la información financiera fluya rápidamente. (Julian, 2019)

Los planes de mitigación al cambio climático en los bosques del país, Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) a través del proyecto “Apoyo al Mecanismo Conjunto en Acciones de Mitigación al Cambio Climático en los Bosques de Bolivia” UNJP/BOL/045/UNJ, implementa el sistema de monitoreo holístico e integral de bosques de Bolivia, brinda apoyo técnico para normalización y estandarización de la información. (FAO, 2015, págs. 1,2)

En la actualidad resultaría inviable realizar manualmente la mayoría de las tareas contables o financieras, de manera que con la realización de este proyecto se obtendrá mejor calidad de información que se requiera de recursos económicos de plantaciones de forestación y reforestación.

1.2 ANTECEDENTES

ANTECEDENTES INSTITUCIONALES

Con el fin de contribuir a la mitigación del cambio climático, Bolivia desarrolló el Mecanismo Conjunto de Mitigación y Adaptación para el Manejo Sostenible de Bosques y de la Madre Tierra (el "Mecanismo Conjunto", JMA), en el marco del cual se está implementando el PN, trabajando con la Dirección General de Gestión y Desarrollo Forestal, la Autoridad Plurinacional de la Madre Tierra y la Autoridad de Bosques y Tierra, instancias bajo tuición del Ministerio de Medio Ambiente y Agua.

Consecuentemente en coordinación con las instancias mencionadas, se redomina y replanifica al PN, como Proyecto "Apoyo al Mecanismo Conjunto en Acciones de Mitigación al Cambio Climático en los Bosques de Bolivia", el cual tiene previsto realizar un registro (inventario) de iniciativas de manejo integral y sustentable del bosque, y la implementación del Sistema de Monitoreo Integral y Holístico de Bosques (con cuatro actividades principales: i) diseño del sistema e implementación de la unidad técnica; ii) diseño y desarrollo de una plataforma de diseminación web; iii) actualización del mapa oficial de cambio de uso de suelo, tasa de deforestación, y deforestación histórica; iv) inventario nacional forestal).

El PN apoyó la conceptualización, desarrollo e implementación del SMHIB, en sus diferentes módulos y subsistemas (Subsistema 1: fiscalización y control para el manejo forestal integral, 2: desarrollo forestal, 3: cambio climático y bosques), Identificando a los actores técnicos con mandato por ley de monitoreo forestal e integrando la información de los sistemas de información forestal a nivel estatal.

La plataforma de diseminación web del sistema, se encuentra en funcionamiento, habiendo considerado las necesidades tecnológicas básicas en la incorporación de los centros de datos.

El portal web del sistema de monitoreo integral y holístico de bosques (SMHIB) ha publicado las siguientes secciones/datos:

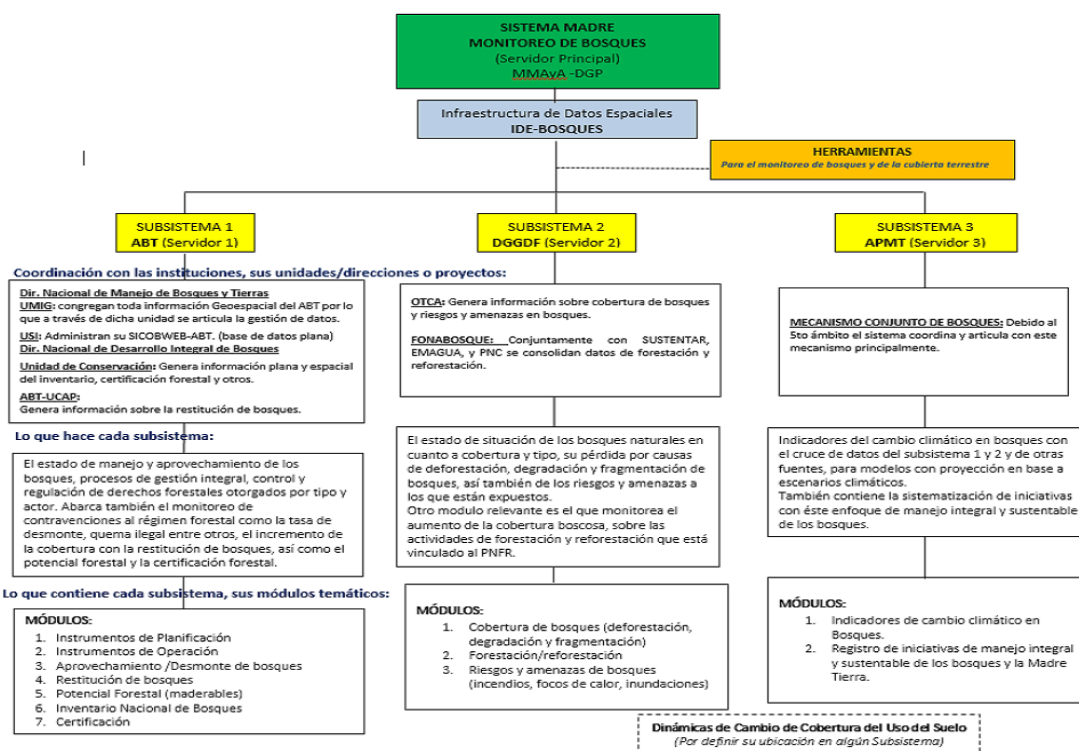
- i) Geobosque (plataforma web geoespacial de código abierto);
- ii) Herramienta de monitoreo de los bosques y la cubierta terrestre;

- iii) Fiscalización y control; conservación forestal vinculada a la información del inventario forestal;
- iv) Seguimiento de datos sobre forestación / reforestación;
- v) Cambio climático y bosques (vinculado al registro de iniciativas).

En ese marco se fue diseñando, desarrollando e implementado el SISTEMA DE MONITOREO HOLÍSTICO E INTEGRAL DE BOSQUES que incorpora una Infraestructura de datos espaciales GEOBOSQUES, con tres servicios GEO DGGDF, GEO ABT y GEO APMT, principalmente para compartir información geoespacial sobre regulación del uso y acceso al bosque, temáticas relativas a deforestación, amenazas que sufren los bosques, como focos de calor, incendios, y coberturas relativas a bosques y cambio climático; también otra parte importante son las herramientas de monitoreo que pone a disponibilidad información satelital desde 1984 hasta el 2016; además del módulo de Forestación y Reforestación, así mismo presenta información del piloto del Inventario Nacional Forestal con una propuesta metodológica para el Inventario Nacional de Bosques que incluye una herramienta para la elaboración de los Planes de Manejo Integral de Bosques PMIB; y por ultimo una herramienta que recopila información de iniciativas de manejo integral y sustentable del bosque.

Propuesta de esquema conceptual del sistema.

Figura 1. 1: Propuesta de esquema conceptual del sistema



Fuente:(Proyecto/unjp/045/bol/njp)

El Estado Plurinacional de Bolivia, a través de sus autoridades nacionales, departamentales e institucionales no tiene acceso rápido y oportuno en la consulta de información y la toma de decisiones geoespacial en torno a la gestión de los bosques.

La información geoespacial de monitoreo de los bosques tiene una dinámica espacio temporal diferente y es realizado por distintas instancias según tuición, estando en muchos casos dispersa, por lo cual, el acceso a información, su visualización y consultas de datos requiere tiempos según la coordinación institucional.

Existen sistemas o procesos de generación y tratamiento de la información que se generan exclusivamente de acuerdo a competencias y atributos específicos de las instituciones, unidades, jefaturas, direcciones, por lo que se torna una problemática al momento de recopilar y homogenizar la información que involucra el monitoreo de bosques, dificultándose contar con una información oportuna.

Uno de los requerimientos más importantes que se genera en el SISTEMA DE MONITOREO HOLÍSTICO E INTEGRAL DE BOSQUES; es el de disponer de una

plataforma de disseminación Web con software libre, que sea capaz de visualizar, descargar y publicar información geográfica, bajo normativas ISO 19100 y lineamientos Open GeoSpatial Consortium (OGC), como son: los mapas en web (WMS), objetos vectoriales en web (WFS) y coberturas raster (WCS), además del uso de herramientas con software gratuito, que proporcione otras ventajas de procesamiento de imágenes de satélite como la plataforma de Google Engine y otros.

El SMHIB a través de la infraestructura de datos espaciales GEOBOSQUE actual se encuentra enfocado en trabajar bajo la estructura de los Geoservicios, estando implementados los servicios: WMS y WFS, pero no cuenta con el servicio WCS, es una innovación a nivel Nacional dentro de lo que es la Infraestructura de Datos Espaciales "IDE", incurriendo siempre en un constante análisis de proyectos geográficos, con relevancia en el campo de la investigación, innovación y desarrollo; permitiendo el acceso público a la información, a la capacitación y al uso de herramientas necesarias para el uso de la información.

INTERNACIONAL

- Plataforma Web para la Programación y Control del Recurso Financiero, POA (2016). La presente tesis implementa una aplicación web que apoya los procesos administrativos de una institución de educación superior, de la administración pública federal, el propósito de dicho sistema informático es la gestión de su recurso financiero, permitiendo que el jefe de cada departamento hacer su solicitud de recursos para que el administrador pueda asignarles los recursos solicitados. Así también cada jefe de área podrá saber cuánto ha gastado y cuanto le falta para llegar a su límite. Con esto se aprovechan ampliamente las tecnologías de las Tics, siendo un baluarte que puede ser difundido para que otras aplicaciones de la misma naturaleza la puedan utilizar y adecuar a sus necesidades. (Obando, 2016)
- Propuesta de diseño e implementación de un sistema financiero contable, aplicado a "la comunidad terapéutica San José Marina", dedicado al tratamiento de pacientes con alteraciones mentales, (2016). La tesis es una Propuesta de Diseño e Implementación de un Sistema Financiero/Contable, el cual se basa

en que partiendo de la documentación legal base, se pueda realizar el ingreso de datos y valores en el libro diario y a su vez éstos valores se actualicen automáticamente en el Mayor General, Balance de Comprobación y los Respectivo Estados Financieros, para poder realizar el análisis financiero en tiempo real y en forma oportuna y confiable. (Armendáriz, 2016)

NACIONAL

- Sistema Web de Generación de Reportes Gerenciales de Indicadores Financieros de Responsabilidad Social, UMSA, (2016). surge como una necesidad de Bolivia Tech Hub para automatizar la generación de reportes con gráficos de esta manera poder abrir datos y poder visualizarlos, estos reportes son diseñados de acuerdo a la estrategia institucional para después ser evaluados y finalmente ser presentados por las instituciones financieras al público en general adjunto a los balances anuales. Para satisfacer las necesidades de la problemática del proyecto se desarrolló el sistema de información vía web para la institución que permite generar reportes de indicadores de responsabilidad social financieros incluyendo gráficos con los resultados de las evaluaciones, comparaciones entre indicadores y comparaciones entre reportes, así como la gestión de estos reportes para su publicación.

Para el desarrollo del sistema se determinó el uso de la metodología Ágil SCRUM, que utiliza un modelo de desarrollo incremental, y este se complementó con la herramienta UML para las etapas de desarrollo en cada iteración.

Realizada en Universidad Mayor de San Andrés, La paz. (Gonzales, 2016)

- Sistema de Administración de proyectos mediante aplicaciones Web y Móvil. Caso: Ministerio de Obras Publica, Servicios y Vivienda, UMSA, (2016). El presente Proyecto de Grado consiste en implementar una Aplicación Móvil integrada con la Aplicación Web que permita administrar información consolidada (Físico, Financiero y Gerencial) de Proyectos del Ministerio de

Obras Públicas, Servicios y Vivienda (MOPSV), de forma que puedan ser obtenidos en cualquier momento y desde cualquier lugar.

La Aplicación Móvil fue desarrollada bajo la plataforma Android SDK – API 23, soporta dispositivos con versiones Android superiores o iguales a 4.0.3, con uso del Entorno de Desarrollo “Android Studio”. La Aplicación Web fue desarrollada bajo la plataforma “.NET” con uso del Entorno de Desarrollo “Visual Studio”. Aplicación Web que funciona tanto en Firefox y Google Chrome. Para la realización de este Proyecto de Grado me apoyé en la metodología de desarrollo de software “Programación Extrema (XP)”, la cual se basa en la realimentación continua entre cliente y equipo de desarrollo. La XP tiene cuatro fases estructurales: Planificación, Diseño, Desarrollo y Pruebas.

Realizada en Universidad Mayor de San Andrés, La paz. (Rojas, 2016)

1.3 PLANTEAMIENTO DEL PROBLEMA

Las informaciones financieras de forestaciones y reforestaciones que estén generalizadas por diferentes departamentos, por niveles de usuario y en distintos formatos para su exportación, son sumamente importante para las instituciones gubernamentales, privadas y usuarios (cualquier persona natural).

Actualmente el sistema de forestaciones y reforestaciones existe en la institución que consta de varios módulos, cada uno cumple con el fin que fue elaborado las cuales se encuentran implantadas¹ y en funcionamiento.

1.3.1 PROBLEMA PRINCIPAL

La problemática que existe es la generación de reportes financieros dinámicos visualizados en “gráficos, histogramas, líneas, mapas”, por lo cual se desconoce la información por niveles como departamentos, provincias, municipios, comunidades, a nivel de usuarios y por años en el proyecto UNJP/BOL/045/UNJ.

Problemas secundarios

- No cuenta con reportes dinámicos.
- Ausencia de generación de gráficos basado en la base de datos.

¹ Establecer y poner en ejecución doctrinas nuevas

- Falencia de exportación de datos generados en formato (xls y pdf) porque no existe un módulo financiero.
- Falta de un módulo de administración de usuario para su acceso correspondiente.

1.3.2 PREGUNTA DE LA INVESTIGACIÓN

Después de todo lo mencionado se llegó a la siguiente pregunta de investigación.

¿De qué manera coadyuvaría el módulo financiero en el sistema de reforestación y forestación del proyecto UNJP/BOL/045/UNJ?

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Desarrollar un módulo de reportes financiero para el sistema de reforestación y forestación del proyecto UNJP/BOL/045/UNJ.

1.4.2 OBJETIVOS ESPECÍFICOS

- Realizar el análisis de los requerimientos de la institución y recabar la información existente.
- Diseñar el módulo de reportes financieros para el sistema de forestación y reforestación.
- Implementar la generación de reportes dinámicos, en “gráficos, histogramas, líneas, mapas” por diferentes niveles de usuarios, niveles departamentales y la exportación en formatos (xls y pdf).
- Evaluar pruebas de calidad² en el funcionamiento del módulo financiero.

1.5 JUSTIFICACIÓN

1.5.1 JUSTIFICACIÓN TÉCNICA

El proyecto se justifica técnicamente en la construcción del módulo financiero para el sistema de reforestación y forestación caso, proyecto UNJP/BOL/045/UNJ, se ha de elaborar con herramientas de software libre, en la programación del módulo se desarrollará basado en programación orientada a objetos.

² La calidad del software se encuentra casi a la par de la calidad tradicional

1.5.2 JUSTIFICACIÓN ECONÓMICA

El módulo financiero que se realizara, será netamente con software libre, no tendrá ningún costo económico en el desarrollo, en la implementación y en la licencia del mismo para la institución.

1.5.3 JUSTIFICACIÓN SOCIAL

Se justifica socialmente que el módulo, será de gran aporte con la obtención de información eficiente a través de reportes, los beneficiados serán las Instituciones Gubernamentales (Ministerios), Instituciones y usuarios (cualquier persona natural) en general que necesiten información de recursos económicos de plantaciones referentes a su comunidad o institución perteneciente al sistema de reforestación y forestación.

1.6 METODOLOGIA

1.6.1 MÉTODO CIENTÍFICO

La investigación científica³ es una investigación sistemática, controlada, empírica y crítica de proposiciones hipotéticas sobre las propuestas relaciones que existen entre fenómenos naturales. (Kirlenger, 2002)

Otra definición es el procedimiento racional e inteligente de dar respuesta a una serie de incógnitas, entendiendo su origen, su esencia y su relación con uno o varios efectos. (Sosa de Martínez, 1990, pág. 45)

En términos generales, el método científico es inherente a la ciencia, tanto a la pura como a la aplicada. Sin método científico no puede haber ciencia. El método científico, como ya lo mencionamos, no es infalible, tampoco es autosuficiente, es decir, debe partir de algún conocimiento previo que se requiera concretar o bien ampliar, para posteriormente adaptarse a las especificaciones de cada tema, materia y/o especialidad. Integra una serie de procedimientos lógicos sistemáticos, racionales e intelectuales que permite resolver interrogantes.

¿Qué es una técnica de investigación?

³ Se ajusta a los principios y métodos de la ciencia o está relacionado con ella.

La técnica propone las normas para ordenar las etapas del proceso de investigación, proporciona instrumentos de recolección, clasificación, medición, correlación y análisis de datos, y aporta a la ciencia todos los medios para aplicar el método.

Las técnicas de investigación que se utilizaran bajo el método empírico que permiten las diversas posibilidades de recolección de información son las siguientes en el proyecto. (Maya, 2014)

- Encuesta
- Entrevista
- Cuestionario

1.6.2 METODOLOGIA DE DESARROLLO

UWE (UML Web Engineering), es un proceso de desarrollo para aplicaciones web enfocado sobre el diseño, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación web. Describe una metodología de diseño sistemática, basada en las técnicas de UML (lenguaje modelado unificado), la notación de UML y los mecanismos de extensión de UML.

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización y personalización. UWE es una propuesta basada en el proceso unificado y UML pero adaptados a la web.

Además UWE no limita el número de vistas posibles de una aplicación por lo que UML proporciona mecanismos de extensión, estos mecanismos de extensión son los que UWE utiliza para definir estereotipos, que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones web. (Daniel & Jadisha, 2019)

Fases a utilizar:

- Fase de captura, análisis y especificaciones de requisitos
- Fase de diseño del sistema
- Fase de codificación del software
- Fase de pruebas
- Fase de instalación o implementación

- Fase de mantenimiento

1.6.3 METRICA DE CALIDAD

NORMA ISO 9126

Un estándar internacional para evaluación del software que surge debido a la necesidad de un modelo único para expresar la calidad de un software, supervisado por el proyecto SQuaRE (System and Software Quality Requirements and Evaluation), ISO 25000:2005, el cual sigue los mismos conceptos.

Este estándar propone un modelo de calidad que se divide en tres vistas, exterior, interior y en uso.

La normativa define diez características de los cuales seis son comunes estos son divididos en un número de sub-características, las cuales representan un modelo detallado para la evaluación de cualquier sistema informático.

(Roger S. Pressman, 2005, pág. 343)

- Funcionalidad
- Fiabilidad
- Usabilidad
- Eficiencia
- Mantenibilidad
- Portabilidad

1.6.4 METRICA DE COSTOS

COCOMO II

El Modelo Constructivo de Costos (Constructive Cost Model) fue desarrollado por B. W. Boehm a finales de los 70 y comienzos de los 80 y Cocomo II, presentado por Danny Yhoel Cuno Coila a finales de 90.

Modelo cocomo II, un modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software, una herramienta basada en las líneas de código la cual la hace muy poderosa para la estimación de costos. Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y

así incrementar la precisión en la estimación de costos e implementar una herramienta de software que soportara el modelo.

Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información.

(Gómez, López, & Migani, Cocomo II, 2019)

Estos tres modelos son:

- Modelo de composición de aplicaciones, donde el prototipo de las interfaces de usuario, la interacción del sistema y del software, la evaluación del rendimiento, y la evaluación de la madurez de la tecnología son de suma importancia.
- Modelo de fase de diseño previo, utilizado una vez que se han estabilizado los requisitos y que se ha establecido la arquitectura básica del software.
- Modelo de fase posterior a la arquitectura, utilizado durante la construcción del software.

Las normas ISO son normas o estándares de seguridad establecidas por la Organización Internacional para la Estandarización (ISO) y la Comisión Electrotécnica Internacional (IEC) que se encargan de establecer estándares y guías relacionados con sistemas de gestión y aplicables a cualquier tipo de organización internacionales y mundiales, con el propósito de facilitar el comercio, facilitar el intercambio de información y contribuir a la transferencia de tecnologías. (ISO-27000, 2018)

En concreto la familia de normas ISO/IEC 27000 son un conjunto de estándares de seguridad (desarrollados o en fase de desarrollo) que proporciona un marco para la gestión de la seguridad.

Contiene las mejores prácticas recomendadas en seguridad de la información para desarrollar, implementar y mantener especificaciones para los Sistemas de Gestión de la Seguridad de la Información (SGSI) utilizable por cualquier tipo de organización, pública o privada, grande o pequeña.

La seguridad de la información, según la ISO 27001, se basa en la preservación de su confidencialidad, integridad y disponibilidad, así como la de los sistemas aplicados para su tratamiento. (Vargas & Alonso Castro, 2017, págs. 11-13)

- Preservar la confidencialidad de sus datos.
- Conservar la integridad de sus datos.
- Disponibilidad de la información protegida.

1.7 HERRAMIENTAS DE DESARROLLO

- **Debian**, es un Sistema Operativo de software libre.
- **PostgreSQL**, es una Base de Datos relacional orientada a objeto de código abierto.
- **Apache**, es un servidor web HTTP de código abierto.
- **PHP**, es un lenguaje de programación.
- **Laravel**, es uno de los frameworks de código abierto.
- **pgAdmin**, una herramienta de código abierto para la administración de bases de datos.
- **VueJS**, es un framework progresivo, que sirve para consumir interfaz del usuario.
- **Bootstrap**, es un framework que permite crear interfaces web con css y JavaScript.
- **CSS**, lenguaje de diseño gráfico.
- **JavaScript**, es un lenguaje de programación.
- **Highcharts**, gráficos de código abierto modernos e interactivos.
- **Amcharts 4**, generador de gráficos JavaScript flexibles para diseñadores y desarrolladores.
- **QGis**, Sistema de Información Geográfica de código abierto.
(totalpc, 2019)

1.8 LIMITES Y ALCANCES

1.8.1 LIMITES

Las limitaciones del proyecto es el desarrollo del módulo financiero para el sistema de reforestación y forestación, caso proyecto UNJP/BOL/045/UNJ, y no para otros sistemas.

1.8.2 ALCANCES

El modulo financiero se integrará al sistema que actualmente existe además se desenvolverá, a través de:

- Módulo de reportes
- Módulo de autenticación de usuario
- Generación de reportes en gráficos
- Reportes que se generaran en forma general, por niveles y según usuario

1.9 APORTES

El desarrollo del presente proyecto se realizará bajo las nuevas tecnologías de desarrollo con software libre, por lo cual el módulo financiero mejorara la obtención de información según los niveles de usuario para el sistema de reforestación y forestación caso proyecto UNJP/BOL/045/UNJ.



CAPITULO II

MARCO TEORICO

2.1 INTRODUCCIÓN

En este capítulo se describen, se brindan los conocimientos y conceptos para la realización del proyecto, se va a detallar completamente de forma extensa las teorías de metodologías, técnicas, herramientas entre otras conceptualizaciones convenientes que se utilizaran en el presente proyecto.

La base para el módulo de reportes financieros de forestaciones y reforestaciones es obtener una información generalizada, esta información es bastante importante para los que necesitan dicha información en diferentes reportes.

A continuación se desglosan las teorías relacionadas a base de lo mencionado.

2.2 CONCEPTOS BÁSICOS

Sistemas: La palabra sistema procede del latín system, conjunto de elementos relacionados entre sí que funciona como un todo. (Graus, 2020)

Se entiende por un sistema a un conjunto ordenado de componentes relacionados entre sí, ya se trate de elementos materiales o conceptuales, dotado de una estructura, una composición y un entorno particulares. Se trata de un término que aplica a diversas áreas del saber, como la física, la biología y la informática o computación. (Raffino, 2019)

Un conjunto de elementos relacionados entre sí que tienen un desenvolvimiento común entre todos.

Módulo: Módulo recibe como entrada la salida que haya proporcionado otro módulo o los datos de entrada al sistema (programa) 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.

(Ortíz, 2017, pág. 34)

Un módulo es un software que agrupa un conjunto de subprogramas y estructuras de datos, los módulos son unidades que pueden ser compiladas por separado y los hace reusables y permite que múltiples programadores trabajen en diferentes módulos en forma simultánea, produciendo ahorro en los tiempos de desarrollo. (Alegsa, 2018)

Parte de un sistema y suele estar conectado de alguna manera con el resto de los componentes.

Metodología: Una metodología de software es un enfoque, una manera de interpretar la realidad o la disciplina en cuestión, que en este caso particular correspondería a la ingeniería de software. De hecho, la metodología destinada al desarrollo de software se considera como una estructura utilizada para planificar y controlar el procedimiento de creación de un sistema de información especializada. (Megalnbound, 2017)

Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. (Jacobson & Ivar, s.f.)

Una metodología de desarrollo de software se refiere: entorno o marco de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo

Herramientas: El ciclo del software se compone por diversas etapas que dependen precisamente de qué es lo que se está llevando a cabo, cada una de esas etapas cuenta con distintas herramientas de desarrollo de software. (Hosting, s.f.)

Hay muchas opciones entre las que escoger a la hora de decidir qué herramientas de desarrollo de software vas a utilizar en tus tareas diarias en el trabajo. Para construir software que funcione con eficiencia necesitas las mejores herramientas de software disponibles, que te harán la vida mucho más fácil. (Ubiquim, s.f.)

Las herramientas que proporciona ingeniería de software cubren las distintas actividades del ciclo de desarrollo.

Modelo: El modelo de proceso de desarrollo de software es quizás la pieza más importante de este engranaje conocido como ingeniería de software, existen varios modelos para el proceso de desarrollo software. Los modelos están conformados por etapas que son generales a todos los enfoques. (Tarradelles, 2019)

Un modelo es una representación de un objeto, sistema o idea, de forma diferente al de la entidad misma. El propósito de los modelos es ayudarnos a explicar, entender o

mejorar un sistema. Un modelo de un objeto puede ser una réplica exacta de éste o una abstracción de las propiedades dominantes del objeto. (Perez, 2016)

Es una representación de un objeto, sistema o idea, de forma diferente al de la entidad misma que están conformados por modelos.

Financiero: El análisis financiero nos ayuda a estudiar todo y cada uno de los resultados de la empresa separadas en sus partes para después generara un diagnóstico integral del desempeño financiero de la misma. (Bogdanski, 2016, pág. 25)

Objetivos y políticas generales de la empresa, a fin de lograr que la asignación de recursos sea para aquellos proyectos o segmentos de la misma que ofrezcan la mejor mezcla del rendimiento y liquidez, promoviendo el uso eficiente de estos recursos para mantener su productividad. (Levy, 2017, pág. 24)

Gestionar de todos los procesos de recursos financieros de una organización para diferentes proyectos y generar un diagnóstico.

Software: Composición entre elementos o puntos de reusó en la que dividimos el sistema en procesos. (Arias A. , 2016, pág. 150)

El software está compuesto por un conjunto de programas que son diseñados para cumplir una determinada función dentro de un sistema, ya sean estos realizados por parte de los usuarios o por las mismas corporaciones dedicadas a la informática. (Raffino, 2019)

Conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en una computadora.

Datos: Un dato es una representación simbólica (numérica, alfabética, etc.) de un atributo de una entidad. Un dato no tiene valor semántico (sentido) en sí mismo, pero al ser procesado puede servir para realizar cálculos o tomar decisiones, los datos son susceptibles de ser comprimidos, encriptados, transmitidos y almacenados.

(Leandro, 2018)

Esencialmente mantienen estado y por lo tanto es posible que en algún punto de la arquitectura los diversos elementos necesiten de datos ausentes en los parámetros de peticiones porque estos estados no son capaces de responder a esta carga de nuevas peticiones. (Daniel, Ramo, 2016, pág. 326)

Representaciones simbólicas de un determinado atributo o variable.

Plantaciones de forestación: Se designa con el término de forestación a aquella actividad que se ocupa de estudiar y de gestionar la práctica de las plantaciones, especialmente de los bosques, como lo que son, recursos naturales renovables. El procedimiento que encara la forestación de plantar árboles de manera extendida tiene la misión de equilibrar la masiva e indiscriminada tala que de árboles se ha venido haciendo desde hace mucho tiempo. (Ucha, 2018)

Se denomina forestación al proceso mediante el cual se plantan arboles de forma extensa a efectos de contrabalancear la tala que en el pasado se haya efectuado. La forestación se inscribe así como una metodología de cuidar el hábitat y evitar que especies vegetales o animales se pierdan por el accionar del hombre. (Forestación, 2014)

Creación de plantaciones donde no haya evidencia de vegetación forestal previa, en distintos lugares.

Plantaciones de reforestación: La reforestación es la acción por la cual se vuelve a poblar de árboles un territorio. Su finalidad es de tipo medioambiental, ya que no hay que olvidar que la masa forestal es esencial para el oxígeno que respiramos, regula el clima y es el hábitat natural de especies vegetales y animales. (Navarro, 2015)

En general se acepta que la reforestación se desarrolla en terrenos que, en algún momento de las últimas cinco décadas, hayan estado cubiertos por plantas pero que, por alguna razón, hayan perdido su vegetación. (Pérez & Ana, 2015)

Recuperar plantaciones que se hayan perdido por diversas causas en determinados lugares del planeta.

Reportes: Los reportes relacionados son dos o más reportes que se han asociado entre sí en función de campos o filtros comunes. Puede crear reportes relacionados para proporcionar diferentes capas de interacción y análisis para sus lectores de reportes.

Un único reporte puede contener una tabla y varios gráficos, para proporcionar diferentes capas de interacción y análisis, puede definir relaciones entre las tablas y los gráficos en varios reportes, basado en campos o filtros comunes.

(Galvanize., 2019)

Un reporte es un documento o un escrito que se emplea para dar a conocer algo. A través de los reportes se transmiten distintos tipos de informaciones, con fines muy diversos.

Se conoce como reporte de lectura al informe que elabora una persona después de leer un determinado texto. En dicho reporte se deben incluir ciertos datos que permiten demostrar al individuo que, efectivamente, ha leído el texto y lo ha comprendido.

(Porto & Merino, 2016)

Son una visualización de la información existente en la base de datos del sistema.

Proyecto: Un proyecto es un conjunto de acciones interrelacionadas y dirigidas a lograr resultados para transformar o mejorar una situación, en un plazo limitado y con recursos presupuestados. (Palomar, 2016)

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. (Velasco, 2019)

Es una memoria o escrito donde se detalla el modo y conjunto de medios necesarios para llevar a cabo esa idea; especialmente el que recoge el diseño de una obra de ingeniería o arquitectura.

Usuarios: Usuario se refiere a la persona que utiliza un producto o servicio de forma habitual. Dependiendo del área que se utiliza la palabra, podemos diferenciar algunos aspectos de la persona y del producto o servicio. (Graus, 2019)

El concepto de perfil de usuario se emplea en el ámbito de la informática. Así se denomina a un entorno personalizado para un individuo que se desarrolla de acuerdo a sus preferencias de configuración. (Porto Pérez, 2019)

Aquella persona que utiliza un dispositivo o un ordenador y realiza múltiples operaciones con distintos propósitos.

Sistema contable: Los datos económicos y fiscales reflejan la actividad de la empresa, se realiza el proceso contable correspondiente a un ciclo económico, se calculan los indicadores adecuados para el análisis económico financiero requerido y se elaboran las declaraciones de las liquidaciones. (Mendez, 2015, pág. 9)

Un sistema contable de información está formado por un conjunto de elementos que se interrelacionan para transformar los datos proporcionados por las transacciones y otros eventos económicos que afectan a una organización y producir información de carácter financiero dirigida a todas las personas que tienen interés en la organización, se encuentra fuera o dentro de ella. (Niño, 2015, pág. 26)

Utilizar un sistema contable te ayuda a registrar, organizar o sistematizar la información financiera de tu negocio, y esto te permitirá tomar decisiones de negocios que impulsen una buena gestión empresarial.

Modular: Facilitar la comprensión del diseño modular para lograr el desarrollo de programas complejos de manera sencilla permitiendo la reutilización de fragmentos de un programa. (Guaman, 2015, págs. 4-5)

La programación modular permite dividir un problema en módulos o sub problemas más simples que permitan la solución del mismo, si estos no llegan a ser suficientes, tiene que seguir dividiendo en sub problemas hasta que se halle el resultado y poder ser manipulados con cualquier lenguaje de programación. (Dimas, 2019)

Un sistema es modular cuando está compuesto de varias partes que tienen entre sí alguna conexión, un enfoque donde se subdivide un sistema en partes más pequeñas llamadas módulos, que pueden ser creadas independientemente y luego utilizadas en diferentes sistemas.

Método: Los métodos son los comportamientos o conductas de un objeto y permite identificar la forma en que actúa respecto a su entorno o respecto a otros objetos. Además, representa una operación o función que un objeto realiza, el conjunto de los métodos de un objeto determinan el comportamiento general del objeto.

(Álvarez, Helmer , & Yeison , 2017)

La palabra método hace referencia a ese conjunto de estrategias y herramientas que se utilizan para llegar a un objetivo preciso, el método por lo general representa un medio instrumental por el cual se realizan las obras que cotidianamente se hacen. Cualquier proceso de la vida requiere de un procedimiento para funcionar. El uso de esta palabra es casi coloquial, su uso en cualquier oración indica que existe un procedimiento que seguir si se desea llegar al final de la operación.

El método es una forma de hacer algo de manera sistemática, organizada y/o estructurada. Se refiere a una técnica o conjunto de actividades para desarrollar una tarea. En algunos casos, también se entiende lo que es método como la forma habitual de hacer algo para una persona basada en la experiencia y las preferencias personales. (Eliannys, 2019)

Es un procedimiento que elegimos para obtener un fin predeterminado por otro lado en consecuencia, los métodos de investigación son los distintos modelos de procedimientos que se pueden emplear en una investigación específica, atendiendo a las necesidades de la misma, o sea, a la naturaleza del fenómeno que deseamos.

Costos: Tener el control de los costos de una empresa es primordial para el estado financiero de la misma, ya que permite tomar mejores resultados financieros del negocio. De igual forma, otro de los objetivos de controlar los costos en contabilidad, es poder identificar la cantidad invertida para la producción de un bien y de esta forma establecer el precio de venta del producto final, el margen de rentabilidad y la utilidad real. (Ruiz, 2018)

Se llama costo al desembolso económico que se realiza con el objetivo de mantener o adquirir un producto o un servicio. La idea de total, por otra parte, alude a aquello que abarca todo lo de su especie o que resulta general.

El concepto de costo total, en este marco, refiere a la totalidad de los costos de una empresa. Se trata de la suma de los costos variables (que se modifican cuando cambia el volumen de producción) y los costos fijos (que se mantienen estables más allá del nivel productivo). (Pérez & María, 2019)

Los costos tienden a realizar objetivos primordiales para empresas, instituciones, personas u otros en mantener o adquirir servicios o productos.

Métricas: Las ganancias producidas por los factores anteriormente señalados, debido a su gran diversidad, su ubicuidad y penetración en la economía de productos tangibles y el internet de las cosas, su impacto en la seguridad, la calidad e inclusive en la salud y la vida de las personas y la sociedad en general. Un pobre manejo de ese crecimiento de la complejidad del software se expresa en las estadísticas de productividad, costo y calidad de los proyectos. (Gonzalez & Renato R, 2016, pág. 2)

Las métricas es el de obtener un indicador, el cual permite obtener conocimiento acerca del sistema en estudio y a partir del cual se puedan tomar las decisiones correspondientes. Este indicador apunta a la medición de la calidad de un sistema software, calidad que puede ser definida a través de atributos externos del software como la funcionalidad, la reusabilidad, la usabilidad, la eficiencia, el mantenimiento y la portabilidad o a través de atributos internos como la complejidad, el acoplamiento y la cohesión, entre otros. (Pagani, 2019, pág. 10)

En la evaluación para obtener la calidad en función a: funcionalidad, rentabilidad, eficiencia, portabilidad entre otras de un sistema de software a través de una verificación, como ejemplo son de calidad y de costos.

Registros: Anotar cosas para incluirlas en una lista o relación o para recordarlas si no escribir en un registro el nombre de alguien o de algo, en especial cuando es de carácter oficial. (Lexico, 2020)

El término registro puede referirse a un gran número de circunstancias que tienen en común el hecho de dejar establecido un determinado fenómeno con sus características específicas para que haya conocimiento al respecto por parte de terceros o por un control en la Tecnología de información.

Existen diferentes tipos de registros, pero en todos los casos se hace referencia al concepto de almacenamiento de datos o información sobre el estado, proceso o uso de la computadora. Un registro de sistema se convierte en una base de datos para almacenar la configuración, las opciones y los comandos del sistema operativo. (Redaccion, 2019)

La finalidad de una redacción es almacenar la información en un determinado lugar para su posterior manipulación de los mismos en base a su formato predefinido o definido.

Información: Un conocimiento explícito extraído por seres vivos o sistemas expertos como resultado de interacción con el entorno o percepciones sensibles del mismo entorno. En principio la información, a diferencia de los datos o las percepciones sensibles, tienen estructura útil que modificará las sucesivas interacciones del que posee dicha información con su entorno. (Informacion, 2020)

Como información denominamos al conjunto de datos, ya procesados y ordenados para su comprensión, que aportan nuevos conocimientos a un individuo o sistema sobre un asunto, materia, fenómeno o ente determinado. (Graus, Informacion, 2017)

La información es conjunto de datos recopilados de diferentes formas están son procesados para generar otra información para nuevos conocimientos a adquirirse posterior mente sobre determinados temas.

2.3 MÓDULO FINANCIERO

Los módulos de área financiera permiten formular una serie de estimaciones creíbles y comprensibles que reflejen las proyecciones de resultados económicos, recopilación de datos financieros, genera informes de forma rápida y eficientes como de contabilidad, los activos, los cobros, la tesorería, permitiendo automatizar tareas reducir costes y aumentar el rendimiento de instituciones o empras.

(Tarradelles, 2019)

Este tipo de sistema reciben en resúmenes de las operaciones originadas de otros módulos a través de las entradas automáticas o manuales de dichos módulos que componen de esta forma se dispone de toda la información contable de la compañía.

Dando un ejemplo, algunas operaciones que se realizan repiten en un mes tras otro esto pueden ser definidas en el sistema como entradas fijas o recurrentes y afectar la contabilidad general todos los meses sin previa digitación de las informaciones que la componen. (Lufero, 2015)

Permite formular una serie de estimaciones creíbles y comprensibles que reflejen las proyecciones de resultados económicos.

2.3.1 REPORTES FINANCIEROS

Son los informes sobre la situación económica de una institución o empresa, para que el contador pueda realizar recomendaciones en función del plan de negocio y pronosticar el desempeño futuro. Los reportes financieros básicos son el balance general, estado de resultados, el estado de capital y estado de flujo de efectivo, que es la herramienta básica para realizar el presupuesto.

Los reportes financieros son considerados para muchas compañías como una carga administrativa en vez de una herramienta financiera poderosa, cuando su uso no está correctamente definido. Estos informes funcionan para hacer cumplir los aspectos regulatorios y las declaraciones de impuestos, además permiten generar información financiera con fines indistintos. Pero, sobre todo, sirven para alinear las metas que cada empresa se plantea al inicio del año y crear ajustes en la estrategia del negocio, por lo cual su importancia.

2.3.2 TIPOS DE REPORTES

No cabe duda de que existen muchos tipos de reportes en función de los fines concretos para los cuales se realizan, y no cabe duda de que, cuanto mayor sea la flexibilidad del sistema de gobernanza de datos e informaciones que se adopte, mayores serán las posibilidades de sacar el máximo beneficio a los sistemas de información implementados en la institución.

Sin embargo a la variedad de reportes posibles, se define identificar 5 grandes grupos según la tipología de los fines que persiguen:

Reportes históricos: Se basan en el análisis de los datos y las informaciones históricas relevantes. Sirven para saber qué sucedió en áreas, departamentos o con

el desempeño de determinadas operaciones, identificando patrones e irregularidades y visibilizando las consecuencias de las decisiones tomadas en el pasado.

Reportes operativos: Remiten al presente de la organización, al estado actual de las operaciones que se llevan a cabo y permiten saber qué está sucediendo, en tiempo real, en el sí de la corporación.

Reportes analíticos: Demuestran hasta qué punto es relevante el análisis para una correcta gestión y gobernanza de la información corporativa. Nos permiten descubrir los porqués de la información mostrada en los reportes anteriores, traduciendo esta información en conocimiento relevante para guiar la toma de decisiones.

Reportes predictivos: Fundamentales para el aprovechamiento de las nuevas oportunidades de negocio. Prevén las consecuencias que acarreará la toma de decisiones en un sentido u otro mediante el diseño de escenarios posibles, mostrando qué podría suceder en el futuro y el modo más adecuado para evitar riesgos innecesarios. (Muñiz, 2014, págs. 7-50)

2.4 INGENIERÍA DE SOFTWARE

Según la definición de la IEEE, software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo. Según el mismo autor, un producto de software es un producto diseñado para un usuario. En este contexto, la Ingeniería de Software (SE del inglés Software Engineering) es un enfoque sistemático del desarrollo, operación y mantenimiento. (Lewis, 1994, págs. 1-10)

Se considera que la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software, es decir, permite elaborar consistentemente productos correctos, utilizables y costo-efectivos. (Cotas, 1994, págs. 5-13)

Por otra parte se aclara que cuando se ejecuta el software proporciona las características, función y desempeño buscados que manipulen adecuadamente la información.

Actualmente, hay siete grandes categorías de software de computadora que plantean retos continuos a los ingenieros de software según son: software de sistemas, software de aplicaciones, software de ingeniería y ciencias, software incrustado, software de línea de productos, aplicaciones web llamadas (webapps) y software de inteligencia artificial. (Pressman, 2010, págs. 5-6)

La ingeniería de software es una disciplina de la ingeniería que se interesa por todos los aspectos de la producción de software.

El buen software debe entregar al usuario la funcionalidad y el desempeño requeridos, y debe ser sustentable, confiable y utilizable. (Sommerville, 2011, pág. 7)

La ingeniería de software es una tecnología con varias capas. Como se aprecia en la figura 2.1, cualquier enfoque de ingeniería (incluso la de software) debe basarse en un compromiso organizacional con la calidad.

Figura 2.1: Capas de la ingeniería de software



Fuente: (Pressman, 2010, pág. 12)

2.4.1 EL PROCESO DE SOFTWARE

Un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. Una actividad busca lograr un objetivo amplio (por ejemplo, comunicación con los participantes) y se desarrolla sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de software.

Una acción (diseño de la arquitectura) es un conjunto de tareas que producen un producto importante del trabajo (por ejemplo, un modelo del diseño de la arquitectura)

una tarea que se centra en un objetivo pequeño pero bien definido (por ejemplo, realizar una prueba unitaria que produce un resultado tangible).

En el contexto de la ingeniería de software, un proceso no es una prescripción rígida de cómo elaborar software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán.

Además, la estructura del proceso incluye un conjunto de actividades que son aplicables a través de todo el proceso del software una estructura de proceso general para la ingeniería de software consta de cinco actividades:

- **Comunicación:** Antes de que comience cualquier trabajo técnico, tiene importancia crítica comunicarse y colaborar con el cliente (y con otros participantes). Se busca entender los objetivos de los participantes respecto del proyecto y reunir los requerimientos que ayuden a definir las características y funciones del software.
- **Planeación:** Un proyecto de software es un viaje difícil, y la actividad de planeación crea un “mapa” que guía al equipo mientras viaja. El mapa llamado plan del proyecto de software define el trabajo de ingeniería de software al describir las tareas técnicas por realizar los riesgos probables, los recursos que se requieren, los productos del trabajo que se obtendrán y una programación de las actividades.
- **Modelado:** Ya sea usted diseñador de paisaje, constructor de puentes, ingeniero aeronáutico, carpintero o arquitecto, a diario trabaja con modelos. Crea un bosquejo del objeto por hacer a fin de entender el panorama general cómo se verá arquitectónicamente, cómo ajustan entre sí las partes constituyentes y muchas características más. Si se requiere refinar el bosquejo con más y más detalles en un esfuerzo por comprender mejor el problema y cómo resolverlo. Un ingeniero de software hace lo mismo al crear modelos a fin de entender mejor los requerimientos del software y el diseño que se optimo.

- **Construcción:** Esta actividad combina la generación de código (ya sea manual o automatizada) y las pruebas que se requieren para descubrir errores en éste.
- **Despliegue:** El software (como entidad completa o como un incremento parcialmente terminado) se entrega al consumidor que lo evalúa y que le da retroalimentación, misma que se basa en dicha evaluación.

(Pressman, 2010, págs. 12-13)

2.4.2 INGENIERIA WEB

La ingeniería web es la aplicación de metodologías sistemáticas, disciplinadas cuantificables al desarrollo eficiente, operación y evolución de aplicaciones de alta calidad en la World Wide Web.

Los expertos en la web necesitan utilizar herramientas y técnicas basadas en la ingeniería del software, para poder garantizar el buen funcionamiento y administración de los sitios web. (Belshaw, 2019, págs. 2-4)

La ingeniería web toma prestado muchos de los conceptos y principios básicos de la ingeniería del software, dando importancia a las mismas actividades y técnicas de gestión. Existen diferencias sutiles en la forma en que se llevan a cabo estas actividades, pero la filosofía primordial es idéntica dado que dicta un enfoque disciplinado para el desarrollo de un sistema basado en computadora.

(Bolaños, Vanessa Urrea, & Anyeli Gomez, 2017)

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la web que está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta, ha hecho de que las personas tiendan a realizar todas sus actividades por esta vía.

¿Qué particularidades tiene el desarrollo web?

Se reconoce que las aplicaciones web tienen sus particularidades, y por ello deben recibir especial atención en algunos puntos, pero esto no significa que deban ignorar por completo la ingeniería. (Huerta, Ingeniería web, 2014)

Entre las particularidades más significativas podemos listar:

- **Residente en red:** Una aplicación web reside en una red, y debe dar servicio a una comunidad diversa de clientes.
- **Inmediatez:** Se refiere al corto tiempo que normalmente tienen los proyectos web para terminar, o por lo menos, lanzar una versión inicial.
- **Evolución continúa:** A diferencia del software de aplicaciones convencional, que evoluciona a través de versiones planeadas y cronológicamente espaciadas, las aplicaciones web están en constante evolución, y se actualizan gradualmente.
- **Seguridad:** Dado que no controlamos con certeza quién puede acceder a nuestra aplicación; la seguridad y confidencialidad de la información requieren un énfasis especial.
- **Estética.** Es bien sabido que la primera impresión jamás se olvida, por lo que nuestro sitio debe ser atractivo, ergonómico y usable.
- **Medible:** Mediante la cuantificación de resultados, podemos conocer la cantidad de usuarios que tenemos, así como sus patrones de comportamiento.

2.5 METODOLOGÍA DE DESARROLLO DE SOFTWARE

Las metodologías de desarrollo de software tienen como objetivo presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas que permitan desarrollar software de calidad, incluyendo construcción y criterios de comparación de modelos de sistemas. (Senn, 2020)

Los métodos definen las reglas para las transformaciones internas de las actividades, mientras que las metodologías definen el conjunto de métodos. Un método es un procedimiento que define tareas o acciones a realizar, donde cada tarea incluye condiciones de entrada y de salida que se deben satisfacer antes de realizarse y después de completarse.

Las diferentes metodologías varían en el alcance del apoyo que proporcionan al desarrollo de software pero los métodos deben apoyar conceptos básicos que se consideren significativos para resolver el problema. Se deben poder utilizar los métodos en diferentes dominios de aplicación para aplicar a sistemas basados en

diferentes arquitecturas, incluyendo secuencial, concurrente, distribuida e incluso en tiempo real.

Además los métodos deben ajustarse al ciclo de vida del proceso, apoyando las distintas actividades, incluyendo la documentación.

Deben explicar las suposiciones, metas y objetivos que llevaron hacia un resultado particular no deben contradecir el orden establecido para las actividades del modelo de proceso, sino proveer guías para llevarlas a cabo incluyendo el mantenimiento de un sistema que también debe estar apoyado por los métodos. (Martinez, 2015)

Se puede mencionar los métodos que están dentro de las metodologías de desarrollo de software como ser: iterativos, evolutivos y ágiles.

Para tener éxito en el proyecto y obtener un producto final para nuestros clientes, depende en gran parte de la metodología que escoge el equipo de desarrollo, estas pueden ser tradicional o ágil, para que los equipos maximicen su potencial, aumenten la calidad del producto con los recursos y tiempo establecido. La metodología elegida tiene que adaptarse a las necesidades definidas al inicio del proyecto.

Tabla 2.1: Comparación de metodologías de desarrollo

Metodologías Tradicionales	Metodologías Ágiles
Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo	Basadas en heurísticas provenientes de prácticas de producción de código
Cierta resistencia a los cambios	Especialmente preparados para cambios durante el proyecto
Impuestas externamente	Impuestas internamente (por el equipo)
Proceso mucho más controlado, con numerosas políticas/normas	Proceso menos controlado, con pocos principios.
El cliente interactúa con el equipo de desarrollo mediante reuniones	El cliente es parte del equipo de desarrollo
Más artefactos	Pocos artefactos

Más roles	Pocos roles
Grupos grandes y posiblemente distribuidos	Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio
La arquitectura del software es esencial y se expresa mediante modelos	Menos énfasis en la arquitectura del software
Existe un contrato prefijado	No existe contrato tradicional o al menos es bastante flexible

Fuente: (Solis, Figueroa-Diaz, & Cabrera Silva, 2018, pág. 9)

2.5.1 METODOLOGÍA DE DESARROLLO TRADICIONAL

Al inicio el desarrollo de software era artesanal en su totalidad la fuerte necesidad de mejorar el proceso y llevar los proyectos a la meta deseada tuvieron que importarse la concepción los fundamentos de metodologías existentes en otras áreas para adaptarlas al desarrollo de software. Esta nueva etapa de adaptación contenía el desarrollo dividido en etapas de manera secuencial para que de algo mejorara la necesidad latente en el campo del software.

Entre las principales metodologías tradicionales tenemos los ya tan conocidos RUP y MSF entre otros que centran su atención en llevar una documentación exhaustiva de todo el proyecto y centran su atención en cumplir con un plan de proyecto definido todo esto en la fase inicial del desarrollo del proyecto.

Otra de las características importantes dentro de este enfoque tenemos los altos costos al implementar un cambio y al no ofrecer una buena solución para proyectos donde el entorno es volátil. (Solis, Figueroa-Diaz, & Cabrera Silva, 2018, págs. 2-9)

2.5.1.1 MODELO ENCASCADA

El modelo en cascada es un proceso de desarrollo secuencial en el que el desarrollo de software se concibe como un conjunto de etapas que se ejecutan una tras otra. Se le denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una encima de otra, y siguiendo un flujo de ejecución de arriba hacia abajo, como una cascada.

LAS FASES DEL MODELO

El modelo de desarrollo en cascada sigue una serie de etapas de forma sucesiva, la etapa siguiente empieza cuando termina la etapa anterior.

Las fases que componen el modelo son las siguientes:

REQUISITOS DEL SOFTWARE

En esta fase se hace un análisis de las necesidades del cliente para determinar las características del software a desarrollar y se especifica todo lo que debe hacer el sistema sin entrar en detalles técnicos. Hay que ser especialmente cuidadoso en esta primera fase, ya que en este modelo no se pueden añadir nuevos requisitos en mitad del proceso de desarrollo.

Por lo tanto, esta es la etapa en la que se lleva a cabo una descripción de los requisitos del software, acuerda entre el cliente y la empresa desarrolladora lo que el producto deberá hacer. Disponer de una especificación de los requisitos que permita estimar de forma rigurosa las necesidades del software antes de su diseño. Además, permitir tener una base a partir de la cual estimar el coste del producto los riesgos y los plazos.

Diseño: En esta etapa se describe la estructura interna del software las relaciones entre las entidades que lo componen.

Descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo, como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Implementación: En esta fase se programan los requisitos especificados haciendo uso de las estructuras de datos diseñadas en la fase anterior. La programación es el proceso que lleva de la formulación de un problema de computación, a un programa que se ejecute produciendo los pasos necesarios para resolver dicho problema.

Al programar, tenemos que realizar actividades como el análisis de las condiciones, la creación de algoritmos y la implementación de éstos en un lenguaje de programación específico.

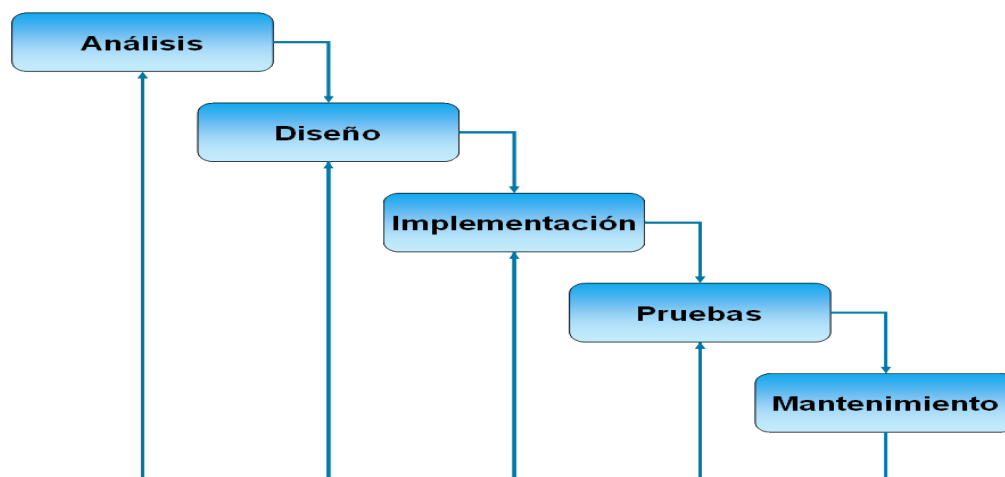
Verificación: Como su propio nombre indica, una vez se termina la fase de implementación se verifica que todos los componentes del sistema funcionen correctamente y cumplen con los requisitos.

El objetivo de las pruebas es el de obtener información de la calidad del software, que sirvan para: encontrar defectos o bugs, aumentar la calidad del software, refinar el código previamente escrito sin miedo a romperlo o introducir nuevos bugs.

Instalación y mantenimiento: Una vez se han desarrollado todas las funcionalidades del software y se ha comprobado que funcionan correctamente, se inicia la fase de instalación y mantenimiento. Se instala la aplicación en el sistema y se comprueba que funcione correctamente en el entorno en que se va a utilizar.

A partir de ahora hay que asegurarse de que el software funcione y hay que destinar recursos a mantenerlo. El mantenimiento del software consiste en la modificación del producto después de haber sido entregado al cliente, ya sea para corregir errores o para mejorar el rendimiento o las características. (Pressman, 2010, págs. 33-34)

Figura 2. 2: Fases del modelo encascada



Fuente: (Pressman, 2010, pág. 34)

2.5.1.2 MODELO ITERATIVO

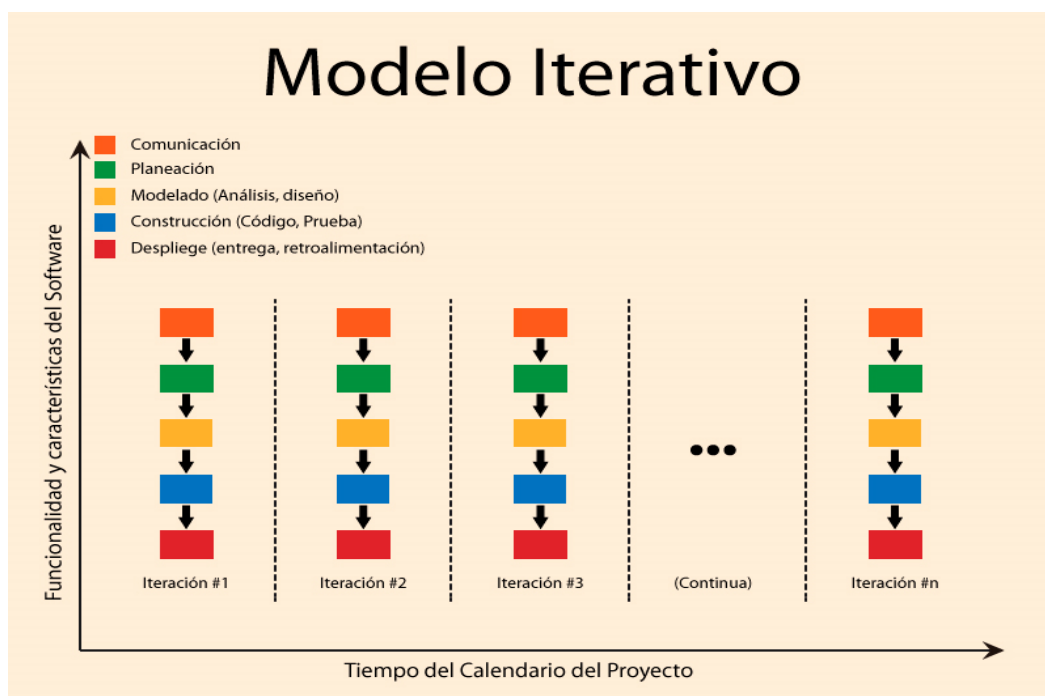
Es un modelo derivado del ciclo de vida en cascada. Este modelo busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de recogida de requisitos.

Consiste en la iteración de varios ciclos de vida en cascada. Al final de cada iteración se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto. El cliente es quien después de cada iteración evalúa el producto y lo corrige o propone mejoras, estas iteraciones se repetirán hasta obtener un producto que satisfaga las necesidades del cliente.

La primera de las ventajas que ofrece este modelo, el no ser necesario tener los requisitos definidos desde el principio, puede verse también como un inconveniente ya que pueden surgir problemas relacionados con la arquitectura.

(Pressman, 2010, págs. 35-36)

Figura 2. 3: Fases del modelo iterativo



Fuente: (Pressman, 2010, pág. 36)

2.5.1.3 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. Los plazos apretados del mercado hacen que sea imposible la terminación de un software perfecto, pero debe lanzarse una versión limitada a fin de aliviar la presión de la competencia o del negocio, se comprende bien el conjunto de requerimientos o el producto básico pero los detalles del producto o extensiones del sistema aún están por definirse. En estas situaciones y otras parecidas se necesita un modelo de proceso diseñado explícitamente para adaptarse a un producto que evoluciona con el tiempo.

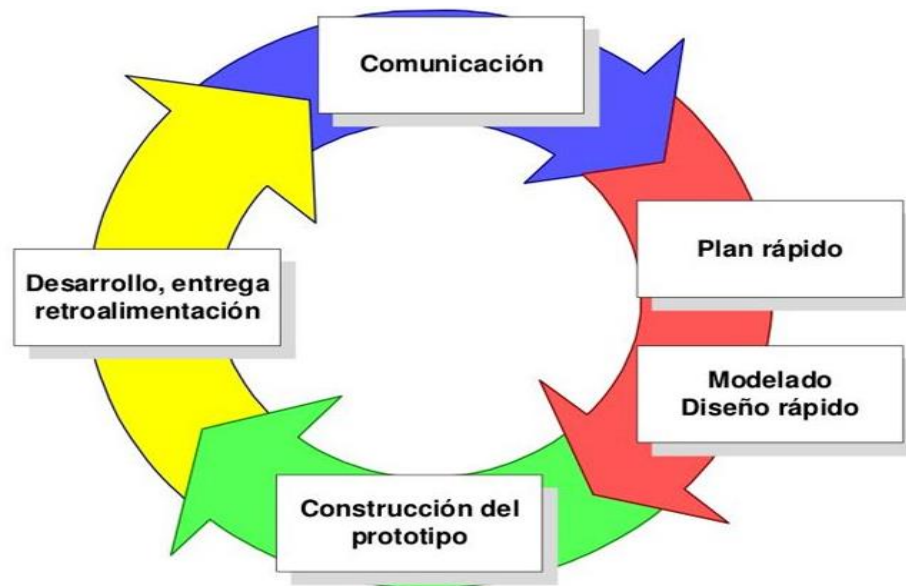
Los modelos evolutivos son iterativos. Se caracterizan por la manera en la que permiten desarrollar versiones cada vez más completas del software. En los párrafos que siguen se presentan dos modelos comunes de proceso evolutivo.

(Pressman, 2010, págs. 36-37)

Al usar prototipos, las etapas del ciclo de vida clásico quedan modificadas de la siguiente manera:

- Análisis de requisitos del sistema
- Diseño, desarrollo e implementación del prototipo
- Prueba del prototipo.
- Refinamiento iterativo del prototipo
- Refinamiento de las especificaciones del prototipo
- Diseño e implementación del sistema final
- Explotación (u operación) o mantenimiento

Figura 2. 4: Fases del modelo evolutivo



Fuente: (Pressman, 2010, pág. 37)

2.5.2 METODOLOGÍAS ÁGILES

Las metodologías ágiles son en realidad una familia de modelos o técnicas, todas ellas compartiendo la característica de interpretar el desarrollo de software como una actividad en la que siempre hay un cierto grado de incertidumbre.

Incertidumbre que hace necesario poner el énfasis en las personas, dejar que se auto organicen y que interactúen buscando siempre satisfacer los requisitos del cliente que sigan planificando iteración a iteración, adaptándose con flexibilidad a los cambios que se producirán de seguro durante la vida del proyecto. (Garza, 2019)

Los principios básicos de modelado ágil se describen a continuación.

- Satisfacer al cliente por medio de la entrega de software funcional.
- Adoptar el cambio, incluso si se introduce en las últimas etapas del desarrollo.
- Seguir entregando software funcional en incrementos y con frecuencia.
- Fomentar a los clientes y analistas a que trabajen juntos a diario.
- Confiar en los individuos motivados para que realicen su trabajo.
- Promover la conversación cara a cara.
- Concentrarse en hacer que el software funcione.

- Fomentar el desarrollo continuo, regular y sostenible.
- Adoptar la agilidad con especial atención en un diseño lúcido.
- Apoyar a los equipos auto organizado.
- Proveer retroalimentación rápida.
- Fomentar la calidad.
- Revisar y ajustar el comportamiento de vez en cuando.
- Adoptar la simpleza.

2.6 METODOLOGÍA UWE

UWE es un proceso del desarrollo para aplicaciones web enfocadas sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación web. UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML. (Luis, 2017)

Es una herramienta que nos permitirá modelar aplicaciones web, utilizada en la ingeniería web, prestando especial atención en sistematización, personalización además de estar considerado como una extensión del estándar UML, también se basa en otros estándares como por ejemplo: XMI (modelo de intercambio de formato), MOF para la meta-modelado, los principios de modelado MDA, el modelo de transformación del lenguaje QVT y XML. (Felipe, 2017)

2.6.1 UWE Y SU RELACION CON UML

La ingeniería Web basada en UML (UWE) fue presentada por Nora Koch en el 2000. Esta metodología utiliza un paradigma orientado a objetos, orientada al usuario y está basada en los estándares UML y UP (Proceso Unificado), que cubre todo el ciclo de vida de este tipo de aplicaciones centrando su atención en aplicaciones personalizadas.

En el marco de UWE es necesario la definición de un perfil UML (extensión) basado en estereotipos con este perfil se logra la asociación de una semántica distinta a los diagramas del UML puro, con el propósito de acoplar el UML a un dominio específico, en este caso, las aplicaciones web.

UWE define vistas especiales representadas gráficamente por diagramas en UML, con esto UWE no limita el número de vistas posibles de una aplicación, UML proporciona mecanismos de extensión basados en estereotipos, estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones web. Los principales aspectos en los que se fundamenta UWE son los siguientes:

- Notación estándar: el uso de la metodología UML para todos los modelos.
- Métodos definidos: pasos definidos para la construcción de cada modelo.
- Especificación de restricciones: recomendables de manera escrita, para que la exactitud en cada modelo aumente.

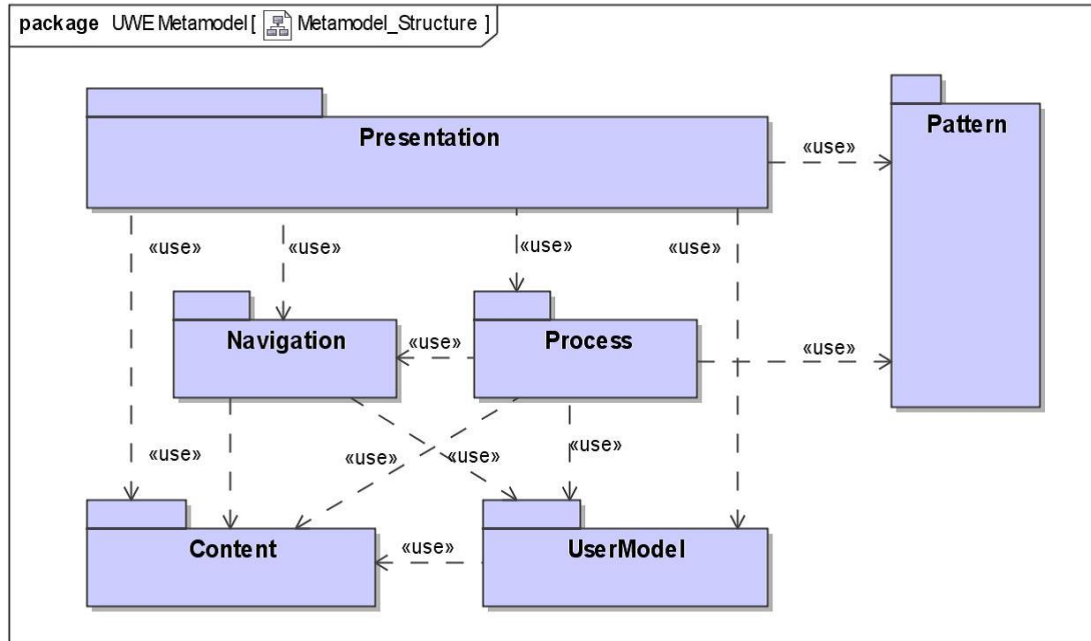
De esta manera, se obtiene una notación UML adecuada a un dominio en específico a la cual se le conoce como perfil UML, se menciona que UWE está especializada en la especificación de aplicaciones adaptativas, por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario. (Paz, 2017, págs. 22-23)

2.6.2 ACTIVIDADES DEL MODELADO UWE

El modelo que propone UWE está compuesto por etapas o sub-modelos:

- **Modelo casos de uso (requerimientos)**
- **Modelo conceptual (contenido)**
- **Modelo de navegación (usuario)**
- **Modelo de presentación (abstracto)**
- **Modelo de proceso**
- **Modelo de adaptación**

Figura 2. 5: Modelos de UWE



Fuente: (UWE, 2016)

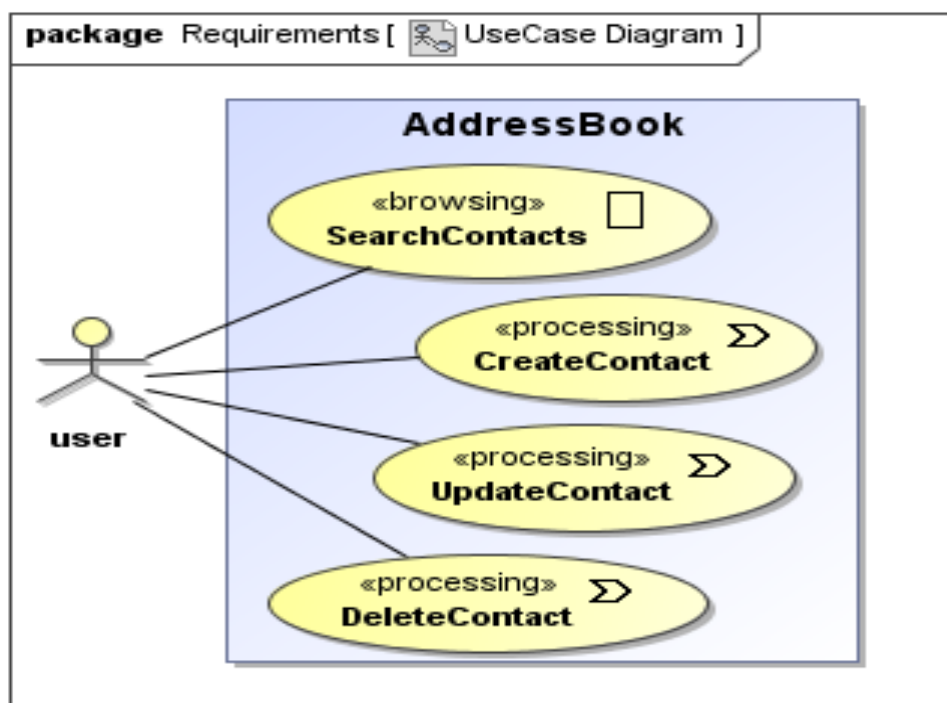
MODELO CASOS DE USO (REQUERIMIENTOS)

Para la obtención de una idea clara de lo que un usuario puede realizar en un módulo o sistema, lo primero que se realiza en diseño web es la modelación de casos de uso. Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario.

Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del usuario la interacción entre actores no se ve en el diagrama de casos de uso. Si esta interacción es esencial para una descripción coherente del comportamiento deseado, quizás los límites del sistema o del caso de uso deban de ser re-examinados.

Alternativamente, la interacción entre actores puede ser parte de suposiciones usadas en el caso de uso. Sin embargo, los actores son una especie de rol un usuario humano u otra entidad externa pueden jugar varios papeles o roles.

Figura 2. 6: Modelado de casos de uso



Fuente: (UWE, 2016)

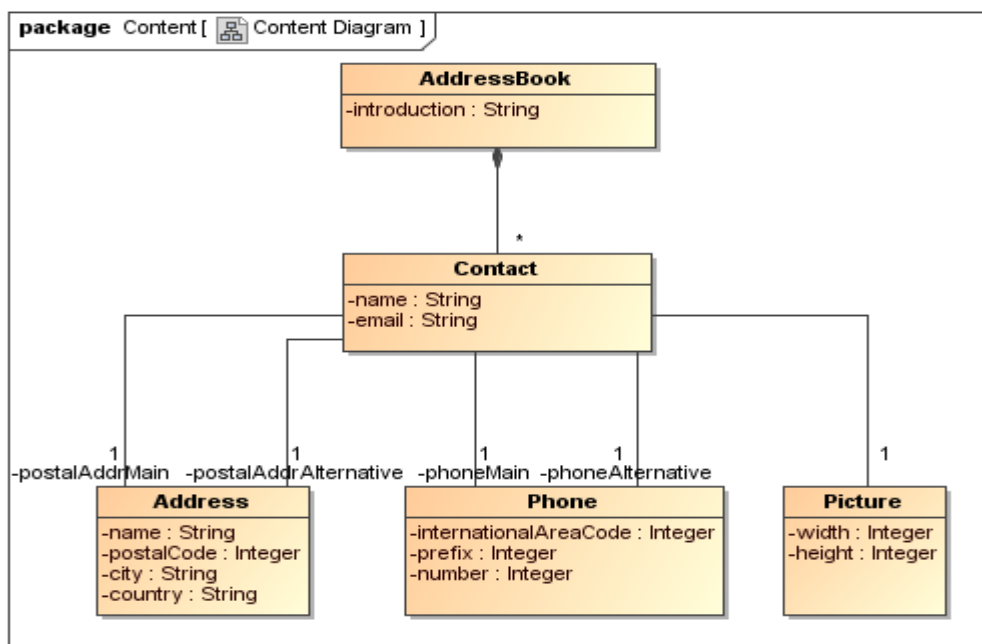
MODELO CONCEPTUAL (CONTENIDO)

El diseño conceptual se basa en el modelo de análisis e incluye los objetos involucrados en las actividades típicas que los usuarios realizan con la aplicación.

El propósito del modelo de contenido es proporcionar una especificación visual de la información relevante para el dominio del sistema web, que comprende principalmente el contenido de la aplicación web y se debe llevar a cabo de acuerdo con los casos de uso que se definen en la especificación de requerimientos.

A continuación se muestra en la imagen 2.7 un ejemplo del modelo.

Figura 2. 7: Modelo de contenido



Fuente: (UWE, 2016)

MODELO DE USUARIO (NAVEGACIÓN)

El objetivo del modelo de navegación es el de especificar las rutas de navegación del contenido de una aplicación web. En este modelo los nodos representan la información del modelo de contenido y los enlaces (links) expresan las rutas de navegación entre los nodos.

En este modelo los nodos son representados por clases UML especializadas y los enlaces por asociaciones UML.

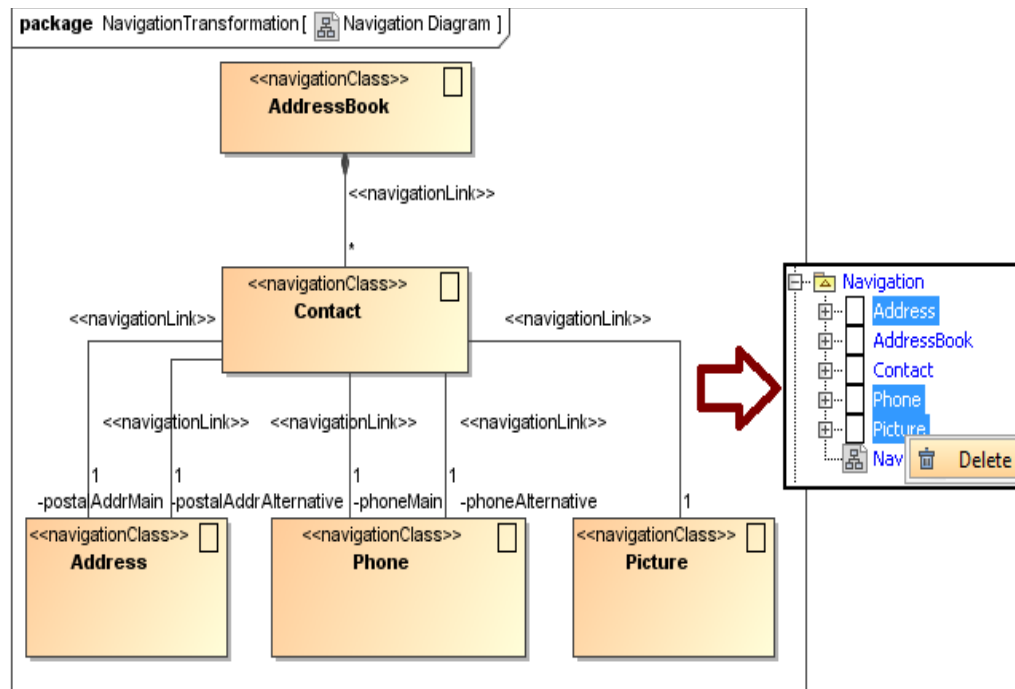
Consta de la construcción de dos modelos de navegación, el modelo del espacio de navegación y el modelo de la estructura de navegación, el primero especifica que objetos serán visitados por el navegador a través de la aplicación. El segundo define como se relacionaran.

El modelo de navegación se construye de la siguiente forma:

- Creación del modelo del espacio de navegación, el cual especifica los nodos que serán visitados por navegación directa a partir de otros.
- Adición de estructuras de acceso al modelo de navegación.

- Adición de menús al modelo para organizar los enlaces de salida de las clases de navegación.
- Los elementos básicos del metamodelo de navegación son Navigation Node y Link, los cuales se derivan de los elementos UML Class y Association, respectivamente.

Figura 2. 8: Modelo de navegacion

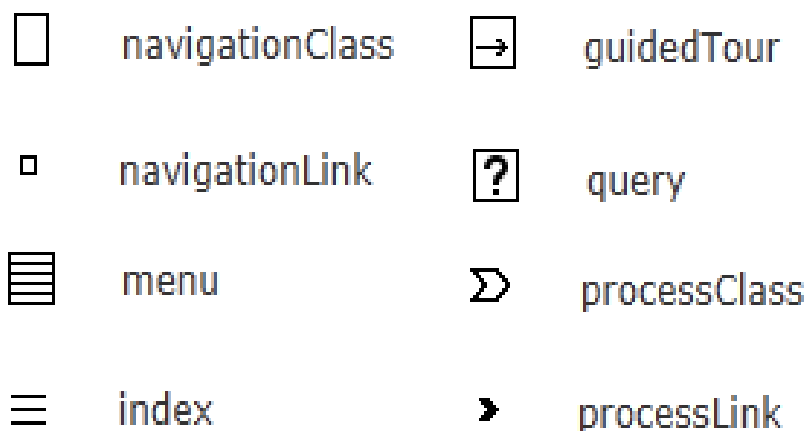


Fuente: (UWE, 2016)

MODELO PRESENTACIÓN (ABSTRACTO)

Describe dónde y cómo los objetos de navegación y accesos primitivos serán presentados al usuario, es decir, una representación esquemática de los objetos visibles al usuario, para expresar, que el elemento está ubicado en una página web. Las propiedades pueden anidarse, por ejemplo cada contacto («presentationGroup»-property) cubre diferentes textos y botones. Ello significa, que para cada contacto la correspondiente dirección de correo y los correspondientes campos de teléfonos y direcciones serán visualizados en la página.

Figura 2. 9: Modelo de presentación



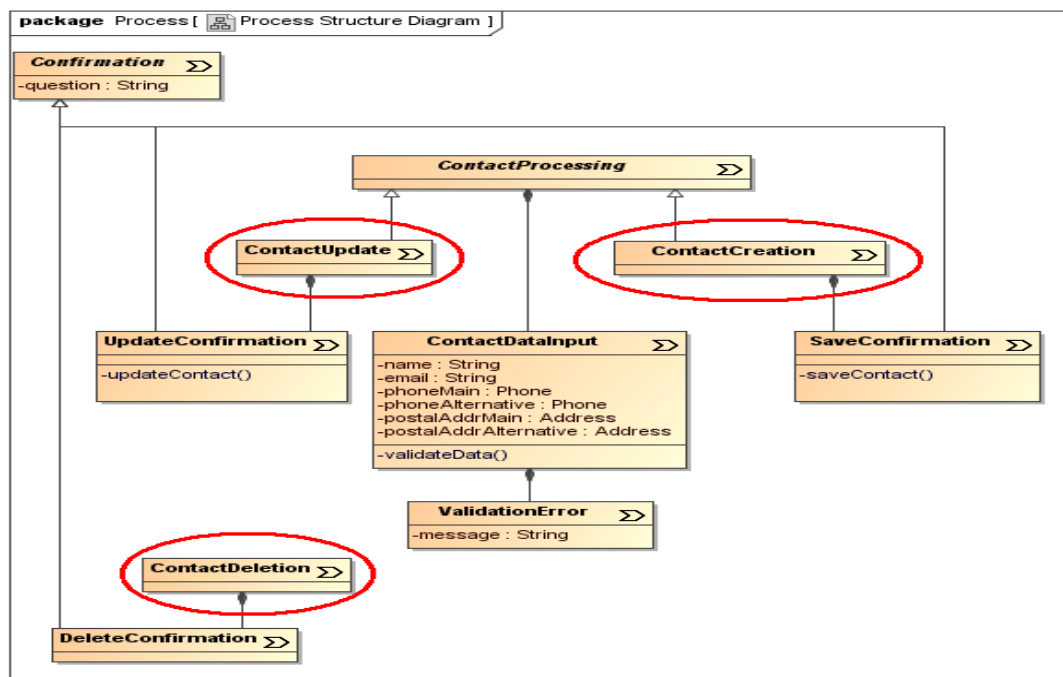
Fuente: (UWE, 2016)

MODELO DE PROCESO

Este modelo especifica las acciones que realiza cada clase de proceso, en este modelo se incluye:

- a) **Modelo de Estructura de Procesos:** Con el fin de describir las relaciones entre las diferentes de clases en el proceso, creamos un diagrama de clases, usando la transformación de navegación a estructura de proceso. Después de ejecutar la transformación tenemos un diagrama de clases con tres clases enmarcadas con un borde rojo:

Figura 2. 10: Modelo de proceso



Fuente: (UWE, 2016)

MODELO DE ADAPTACIÓN

En cuanto a los requisitos, UWE los clasifica dependiendo del carácter de cada uno. Además distingue entre las fases de captura, definición y validación de requisitos.

2.6.3 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:

- **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 prototipo de la interfaz de usuario.

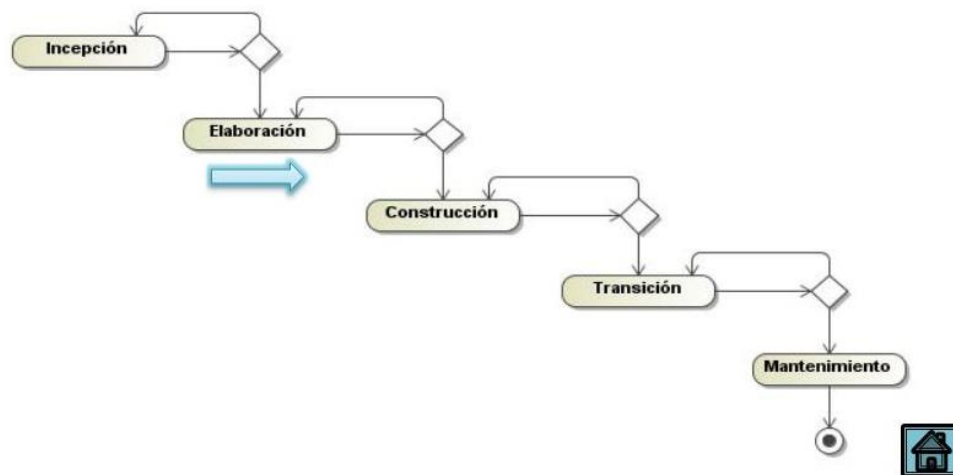
- **Diseño del sistema:** Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación web.
- **Codificación del software:** Durante esta etapa se realizan 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.
- **Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- **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.

- **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.

(UWE, 2016)

Figura 2. 11: Fases de UWE



Fuente: (UWE, 2016)

2.7 MÉTRICAS DE CALIDAD

Los desarrolladores de software más experimentados estarán de acuerdo en obtener software de alta calidad, es una meta importante. Pero, ¿cómo se define la calidad del software?, en el sentido más general se define como: 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. La misma sirve a fin de enfatizar tres puntos importantes:

1. 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. Las prácticas de ingeniería de software permiten al desarrollador analizar el problema y diseñar una solución sólida, ambas actividades críticas de la construcción de software de alta calidad. Por último, las actividades sombilla, tales como administración del cambio y revisiones técnicas, tienen tanto que ver con la calidad como cualquier otra parte de la práctica de la ingeniería de software.

2. Un producto útil entrega contenido, funciones y características que el usuario final desea, sin embargo, de igual importancia es que entrega estos activos en forma confiable y libre de errores. Un producto útil siempre satisface los requerimientos establecidos en forma explícita por los participantes.
3. 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.

(Presman, 2010, pág. 340)

La calidad del software es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

La Calidad del Software (CS) es una disciplina más dentro de la Ingeniería del Software, el principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el plan de calidad, el cual se basa en normas o estándares genéricos y en procedimientos particulares.

Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y que sean cumplidos. (altp, 2019)

- El software es un producto mental, no restringido por las leyes de la Física o por los límites de los procesos de fabricación.
- Se desarrolla, no se fabrica. El coste está fundamentalmente en el proceso de diseño, no en la producción. Y los errores se introducen también en el diseño, no en la producción.
- El software no se deteriora con el tiempo. No es susceptible a los efectos del entorno, y su curva de fallos es muy diferente a la del hardware. Todos los problemas que surjan durante el mantenimiento estaban desde el principio, y afectan a todas las copias del mismo; no se generan nuevos errores.

- Es artesanal en gran medida. El software, en su mayoría, se construye a medida, en vez de ser construido ensamblando componentes existentes y ya probados, lo que dificulta aún más el control de su calidad.
- El mantenimiento del software es mucho más complejo que el mantenimiento del hardware. Cuando un componente de hardware se deteriora se sustituye por una pieza de repuesto, pero cada fallo en el software implica un error en el diseño o en el proceso mediante el cual se tradujo el diseño en código de máquina ejecutable.
- Es engañosamente fácil realizar cambios sobre un software, pero los efectos de estos cambios se pueden propagar de forma explosiva e incontrolada.
- El software con errores no se rechaza. Se asume que es inevitable que el software presente errores.

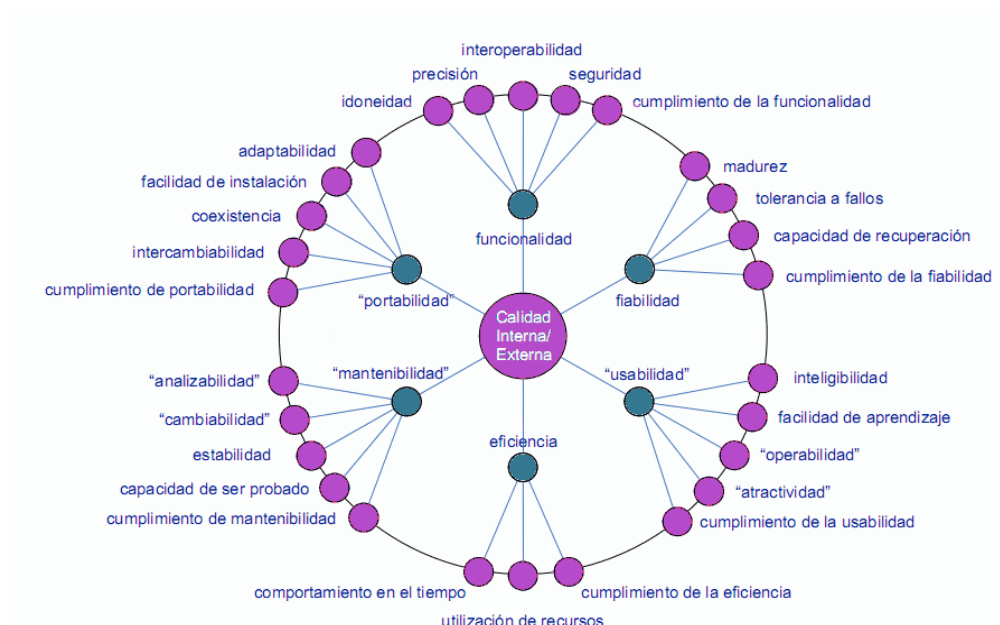
2.7.1 NORMA ISO 9126

Es un estándar internacional para evaluar la calidad del software en base a un conjunto de características y sub-características de la calidad. Cada sub-característica consta de un conjunto de atributos que son medidos por una serie de métricas. Estas métricas miden artefactos obtenidos en etapas tardías del desarrollo de software, aumentando el costo de detección y corrección de errores. Por esta razón, en la literatura ha surgido un mayor interés por la definición de métricas que pretenden evaluar una o varias de las características de calidad definidas en el estándar ISO 9126, en etapas tempranas del desarrollo de software.

Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado (Information technology Software product evaluation Quality characteristics and guidelines for their use), conocido como ISO 9126 o ISO/IEC 9126.

El modelo ISO 9126 fue desarrollado por la ISO (organización de estandarización internacional por sus siglas en Ingles ISO) y este es uno de los grandes grupos reconocidos por los estándares aplicados internacionalmente a través de un amplio rango de solicitudes.

Figura 2. 12: Características y sub-características del modelo ISO 9126



Fuente: (Prieto, ISO/IEC 9126, 2018)

CARACTERÍSTICAS DEL MODELO ISO 9126.

Funcionalidad: Un conjunto de atributos que se relacionan con la existencia de un conjunto de funciones y sus propiedades específicas. Las funciones son aquellas que satisfacen las necesidades implícitas o explícitas.

Es posible expresar la funcionalidad de acuerdo a la siguiente expresión matemática.

$$PF = \text{Cuenta Total} * (X + \text{Min} (Y) * \sum Fi)$$

Dónde:

PF: Medida de funcionalidad

Cuenta Total: Es la suma de los siguientes datos (Nro. de entradas, Nro. de salidas, Nro. de peticiones, Nro. de archivos, Nro. de interfaces externas).

X: Confiabilidad del proyecto, varía entre 1 a 100%.

Min (Y): Error mínimo aceptable al de la complejidad.

$\sum Fi$: Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$.

Para calcular el PF se usa la siguiente ecuación:

$$PF = \text{cuenta total} * (X + \text{Min}(Y) * \sum Fi)$$

$$PF = \text{Cuenta total} * [0.65 + (0.01 * \sum Fi)]$$

Fiabilidad: Un conjunto de atributos relacionados con la capacidad del software de mantener su nivel de prestación bajo condiciones establecidas durante un período establecido.

Se plantea la siguiente fórmula matemática para el cálculo.

$$TMEF = TMDF + TMDR$$

Donde:

TMDF: Tiempo medio de fallo.

TMDR: Tiempo medio de reparación.

Usabilidad: Un conjunto de atributos relacionados con el esfuerzo necesario para su uso, y en la valoración individual de tal uso, por un establecido o implicado conjunto de usuarios.

Eficiencia: Conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la cantidad de recursos necesitados bajo condiciones establecidas.

Mantenibilidad: Conjunto de atributos relacionados con la facilidad de extender, modificar o corregir errores en un sistema software.

El índice de madurez del software se calcula bajo la siguiente fórmula matemática:

$$IMS = [Mt - (Fa + Fb + Fc)] / Mt$$

Donde:

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.

Portabilidad: Conjunto de atributos relacionados con la capacidad de un sistema de software para ser transferido y adaptado desde una plataforma a otra.

Para calcular la portabilidad se usa la siguiente formula:

$$GP=1-(ET/ER)$$

Tabla 2. 2: Sub-características de normas ISO 9126

	Sub Característica	Descripción
Funcionalidad	Aplicabilidad	La capacidad del producto software para proveer un conjunto apropiado de funciones para las tareas y objetivos especificados por el usuario.
	Precisión	La capacidad del producto software para proveer los resultados o efectos acordados con un grado necesario de precisión.
	Interoperabilidad	La capacidad del producto software a interactuar con uno o más sistemas especificados.
	Seguridad	La capacidad del producto software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos y

		a las personas o sistemas autorizados no se les denegará el acceso.
	Conformidad de la funcionalidad	La capacidad del producto software de adherirse a los estándares, convenciones o regulaciones legales y prescripciones similares referente a la funcionalidad.
	Madurez	La capacidad del producto software para evitar fallas como resultado de errores en el software.
	Tolerancia a fallos	La capacidad del producto software para mantener un nivel especificado de funcionamiento en caso de errores del software o de incumplimiento de su interfaz especificada.
Fiabilidad	Recuperabilidad	La capacidad del producto software para reestablecer un nivel especificado de funcionamiento y recuperar los datos afectados directamente en el caso de una falla.
	Conformidad de fiabilidad	La capacidad del producto software para adherirse a las normas, convenciones o regulaciones relativas a la fiabilidad.

Usabilidad	Entendibilidad	La capacidad del producto software para permitir al usuario entender si el software es aplicable, y cómo puede ser utilizado para las tareas y las condiciones particulares de la aplicación.
	Facilidad de aprendizaje	La capacidad del producto software para permitir al usuario aprender su aplicación.
	Operabilidad	La capacidad del producto software para permitir al usuario operarlo y controlarlo.
	Atractividad	La capacidad del producto software de ser atractivo al usuario.
	Conformidad de usabilidad	La capacidad del producto software para adherirse a las normas, convenciones, guías de estilo o regulaciones relacionadas a su usabilidad.
Eficiencia	Comportamiento en el tiempo	La capacidad del producto software para proveer tiempos apropiados de respuesta y procesamiento, y ratios de rendimiento cuando realiza su función bajo las condiciones establecidas.
	Utilización de recursos	La capacidad del producto software para utilizar apropiadas cantidades y tipos

		de recursos cuando éste funciona bajo las condiciones establecidas.
	Conformidad de eficiencia	La capacidad del producto software para adherirse a normas o convenciones relacionadas a la eficiencia.
	Analizabilidad	La capacidad del producto software para ser diagnosticado por deficiencias o causas de fallas en el software o la identificación de las partes a ser modificadas.
	Cambiabilidad	La capacidad del software para permitir que una determinada modificación sea implementada.
Facilidad de Mantenimiento	Estabilidad	La capacidad del producto software para evitar efectos inesperados debido a modificaciones del software.
	Testeabilidad	La capacidad del software para permitir que las modificaciones puedan ser validadas.
	Conformidad de facilidad de mantenimiento	La capacidad del software para adherirse a estándares o convenciones relativas a la facilidad de mantenimiento.
Portabilidad	Adaptabilidad	La capacidad del producto software para ser adaptado a diferentes entornos definidos

	sin aplicar acciones o medios diferentes de los previstos para el propósito del software considerado.
Instalabilidad	La capacidad del producto software para ser instalado en un entorno definido.
Coexistencia	La capacidad del producto software para co-existir con otro producto software independiente dentro de un mismo entorno compartiendo recursos comunes.
Reemplazabilidad	La capacidad del producto software para ser utilizado en lugar de otro producto software, para el mismo propósito y en el mismo entorno.
Conformidad de portabilidad	La capacidad del software para adherirse a estándares o convenciones relacionados a la portabilidad.

Fuente: (ISO/IEC 9126, 2017)

Calidad en uso: Es la calidad del software que el usuario final refleja, la forma como el usuario final logra realizar los procesos con satisfacción, eficiencia y exactitud.

Figura 2. 13: Características del modelo de calidad en uso



Fuente: (ISO/IEC 9126, 2017)

Tabla 2. 3: Modelo de calidad del producto software para la calidad en uso.

Característica	Definición
Efectividad	La capacidad del producto software para permitir a los usuarios lograr las metas especificadas con precisión y completitud en un contexto de uso específico.
Productividad	La capacidad del producto software para permitir a los usuarios emplear cantidades apropiadas de recursos en relación a la efectividad lograda en un contexto de uso específico.
Integridad	La capacidad del producto software para lograr niveles aceptables de riesgo de daño a las personas, negocio, software, propiedad o entorno en un contexto de uso específico.
Satisfacción	La capacidad del producto software para satisfacer a los usuarios en un contexto de uso específico.

Fuente: (ISO/IEC 9126, 2017)

2.8 MÉTRICAS DE COSTOS

La estimación de recursos, costo y calendario para un esfuerzo de ingeniería de software requiere experiencia, acceso a buena información histórica (métricas) y coraje para comprometerse con las predicciones cuantitativas cuando todo lo que existe es información cualitativa. La estimación porta un riesgo inherente, y éste conduce a incertidumbre.

La complejidad del proyecto tiene un fuerte efecto sobre la incertidumbre inherente a la planificación.

El tamaño del proyecto es otro factor importante que puede afectar la precisión y la eficacia de las estimaciones. Conforme aumenta el tamaño, la interdependencia entre varios elementos del software crece rápidamente. La descomposición del problema, un importante enfoque de la estimación, se vuelve más difícil porque el refinamiento de los elementos del problema todavía puede ser formidable

El grado de incertidumbre estructural también tiene un efecto sobre el riesgo de estimación.

En este contexto, estructura se refiere al grado en el cual se solidificaron los requisitos, la facilidad con la que se dividieron las funciones y la naturaleza jerárquica de la información que debe procesarse. (Pressman, 2010, págs. 594-595)

2.8.1 COCOMO II

Modelo COCOMO II, modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software con el nombre de COCOMO, por Constructive Cost Model (Modelo Constructivo de Coste). El modelo COCOMO original se ha convertido en uno de los modelos de estimación de coste del software más utilizados y estudiados. Un modelo matemático empírico obtenido gracias a la recolección de datos que provienen de proyectos de software anteriores. (COCOMO, 2018)

Los objetivos principales que se tuvieron en cuenta para construir el modelo COCOMO II fueron:

- Desarrollar un modelo de estimación de costo y cronograma de proyectos de software que se adaptara tanto a las prácticas de desarrollo de la década del 90 como a las futuras.

- Construir una base de datos de proyectos de software que permitiera la calibración continua del modelo, y así incrementar la precisión en la estimación.
- Implementar una herramienta de software que soportara el modelo.
- Proveer un marco analítico cuantitativo y un conjunto de herramientas y técnicas que evaluaran el impacto de las mejoras tecnológicas de software sobre los costos y tiempos en las diferentes etapas del ciclo de vida de desarrollo.

COCOMO II está compuesto por tres modelos denominados: Composición de Aplicación, Diseño Temprano y Post-Arquitectura. Éstos surgen en respuesta a la diversidad del mercado actual y futuro de desarrollo de software.

Esta diversidad podría representarse con la siguiente tabla 2.4:

Tabla 2. 4: Distribución del mercado de software actual y futuro

Aplicaciones desarrolladas por usuarios finales		
Generador de aplicaciones	Aplicaciones con componentes	Sistemas integrados
	Infraestructura	

Fuente: (Gomes, 2019)

Aplicaciones desarrolladas por usuarios finales: En este sector se encuentran las aplicaciones de procesamiento de información generadas directamente por usuarios finales, mediante la utilización de generadores de aplicaciones tales como planillas de cálculo, sistemas de consultas, etc. Estas aplicaciones surgen debido al uso masivo de estas herramientas, conjuntamente con la presión actual para obtener soluciones rápidas y flexibles.

Generadores de aplicaciones: En este sector operan firmas como Lotus, Microsoft, Novell y Borland con el objetivo de crear módulos pre-empaquetados que serán usados por usuarios finales y programadores.

Aplicaciones con componentes: Sector en el que se encuentran aquellas aplicaciones que son específicas para ser resueltas por soluciones pre-

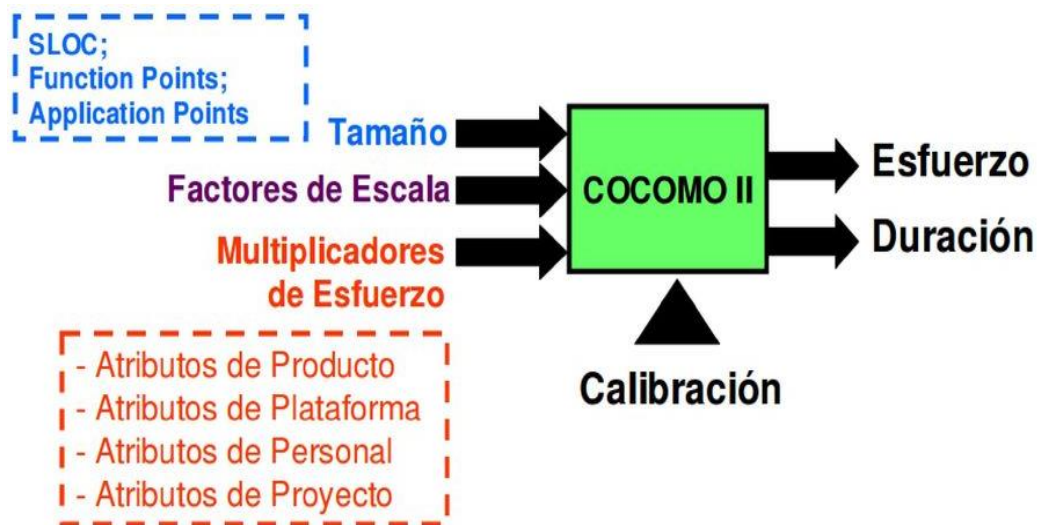
empaquetadas, pero son lo suficientemente simples para ser construidas a partir de componentes interoperables.

Sistemas integrados: Sistemas de gran escala, con un alto grado de integración entre sus componentes, sin antecedentes en el mercado que se puedan tomar como base. Porciones de 27 estos sistemas pueden ser desarrolladas a través de la composición de aplicaciones.

Infraestructura: Área que comprende el desarrollo de sistemas operativos, protocolos de redes, sistemas administradores de bases de datos entre otros.

Los tres modelos de COCOMO II se adaptan tanto a las necesidades de los diferentes sectores descriptos, como al tipo y cantidad de información disponible en cada etapa del ciclo de vida de desarrollo, lo que se conoce por granularidad de la información.

Figura 2. 14: Concepto operacional cocomo II



Fuente: (Gómez & López, s.f., pág. 28)

Composición de aplicación: Es el modelo de estimación utilizado en los proyectos de software que se construyen a partir de componentes pre-empaquetadas. En este caso, se emplean puntos objeto para estimar el tamaño del software, lo cual está acorde al nivel de información que generalmente se tiene en la etapa de planificación, y el nivel de precisión requerido en la estimación de proyectos de esta naturaleza. La fórmula propuesta en este modelo es la siguiente:

$$PM = NOP/PROD$$

Donde:

NOP (Nuevos Puntos Objeto): Tamaño del nuevo software a desarrollar expresado en Puntos Objeto y se calcula de la siguiente manera:

$$NOP = OP \times (100 - \%reusó)/100$$

OP (Puntos Objeto): Tamaño del software a desarrollar expresado en Puntos Objeto

%reusó: Porcentaje de reusó que se espera lograr en el proyecto

PROD: Es la productividad promedio determinada a partir del análisis de datos de proyectos

Diseño Temprano: Se utiliza en las primeras etapas del desarrollo en las cuales se evalúan las alternativas de hardware y software de un proyecto. En estas etapas se tiene poca información, lo que concuerda con el uso de puntos función, para estimar tamaño y el uso de un número reducido de factores de costo.

La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{\text{nominal}} = A \times (KSLOC)^b$$

$$B = 1.01 + 0.01 \times \sum_{j=1}^5 W_j$$

Donde:

PM_{Estimado}: Es el esfuerzo Nominal ajustado por 7 factores, que reflejan otros aspectos propios del proyecto que afectan al esfuerzo necesario para la ejecución del mismo.

KSLOC: Es el tamaño del software a desarrollar expresado en miles de líneas de código fuente.

A: Es una constante que captura los efectos lineales sobre el esfuerzo de acuerdo a la variación del tamaño, (A=2.94).

B: Es el factor exponencial de escala, toma en cuenta las características relacionadas con las economías y des economías de escala producidas cuando un proyecto de software incrementa su tamaño.

EM_i: Corresponde a los factores de costo que tienen un efecto multiplicativo sobre el esfuerzo, llamados Multiplicadores de Esfuerzo (Effort Multipliers).

Post-Arquitectura: Se aplica en la etapa de desarrollo propiamente dicho, después que se define la arquitectura del sistema, y en la etapa de mantenimiento. Este modelo utiliza:

- Puntos Función y/o Líneas de Código Fuente para estimar tamaño, con modificadores que contemplan el reusó, con y sin traducción automática, y el desperdicio.
- Un conjunto de 17 atributos, denominados factores de costo, que permiten considerar características del proyecto referentes al personal, plataforma de desarrollo, etc., que tienen injerencia en los costos.
- Cinco factores que determinan un exponente, que incorpora al modelo el concepto de des economía y economía de escala.

La fórmula para el cálculo del esfuerzo es la siguiente:

$$PM_{\text{estimado}} = PM_{\text{nominal}} \times \prod_{i=1}^{17} EM_i$$

El esfuerzo nominal se ajusta usando 17 factores multiplicadores de esfuerzo. El mayor número de multiplicadores permite analizar con más exactitud el conocimiento disponible en las últimas etapas de desarrollo, ajustando el modelo de tal forma que refleje fielmente el producto de software bajo desarrollo.

(Gómez & López, 2018, págs. 26-30)

2.9 SISTEMA DE INFORMACIÓN GEOGRÁFICA

Los Sistemas de Información Geográfica pueden definirse de forma provisional como sistemas que permiten almacenar datos espaciales para su consulta, manipulación y representación. La representación de datos espaciales es el campo de estudio de la cartografía.

La exactitud en la representación de las tierras emergidas se consideraba accesoria, siendo lo fundamental la exactitud en direcciones y distancias entre puertos. Las cartas náuticas actuales mantienen un esquema similar aunque la generalización de los

Sistemas de Posicionamiento Global (GPS) ha revolucionado los sistemas de navegación (Sarría, 2015, pág. 7)

Un Sistema de Información Geográfica (SIG) es un conjunto de componentes específicos que permiten a los usuarios finales crear consultas, integrar, analizar y representar de una forma eficiente cualquier tipo de información geográfica referenciada asociada a un territorio. La información geográfica va a ser aquella información que tiene algún componente espacial, es decir, una ubicación, y además, una información atributiva que nos detalle más sobre ese elemento en cuestión. Esa ubicación se podrá definir con un nombre de una calle, por ejemplo, o con coordenadas espaciales.

Mapas: Una definición genérica es el despliegue gráfico de elementos espacialmente distribuidos, llamados datos, objetos o elementos que se corresponden con entidades geográficas en el mundo real. Tradicionalmente, los mapas en soporte papel han respondido a la necesidad de sintetizar una parte de la información total del terreno, almacenando los datos de una manera económica y lo más explícita posible. Los datos geográficos tienen las dos componentes ya mencionadas, soporte y variable, y en función del objetivo cartográfico se seleccionan elementos de la realidad, asignando a cada uno de ellos atributos que definan sus características.

Creación de datos: Las modernas tecnologías SIG trabajan con información digital, para la cual existen varios métodos utilizados en la creación de datos digitales. El método más utilizado es la digitalización, donde a partir de un mapa impreso o con información tomada en campo se transfiere a un medio digital por el empleo de un programa de diseño asistido por ordenador con capacidades de georreferenciación.

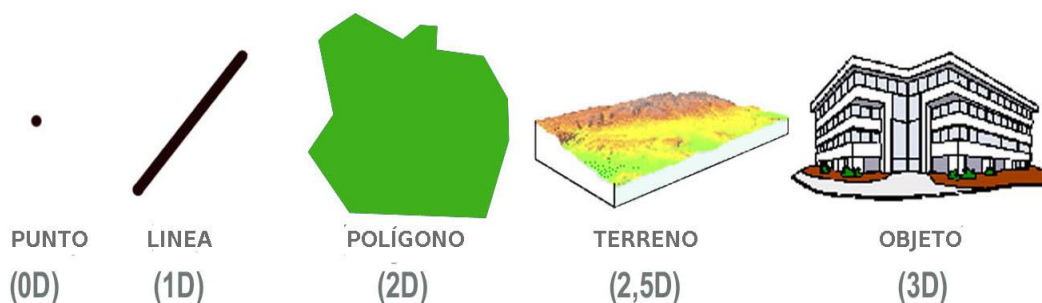
Dada la amplia disponibilidad de imágenes orto-rectificadas (tanto de satélite y como aéreas), la digitalización por esta vía se está convirtiendo en la principal fuente de extracción de datos geográficos. Esta forma de digitalización implica la búsqueda de datos geográficos directamente en las imágenes aéreas en lugar del método tradicional de la localización de formas geográficas sobre un tablero de digitalización.

Representación de datos: Los datos SIG representan los objetos del mundo real (carreteras, el uso del suelo, altitudes). Los objetos del mundo real se pueden dividir en dos abstracciones: objetos discretos (una casa) y continuos (cantidad de lluvia caída, una elevación). Existen dos formas de almacenar los datos en un SIG: raster y vectorial.

Los SIG que se centran en el manejo de datos en formato vectorial son más populares en el mercado. No obstante, los SIG raster son muy utilizados en estudios que requieran la generación de capas continuas, necesarias en fenómenos no discretos; también en estudios medioambientales donde no se requiere una excesiva precisión espacial como contaminación atmosférica, distribución de temperaturas, localización de especies marinas, análisis geológicos, etc.

(Morea Rodríguez & Huerta Rodríguez, 2017, págs. 13-25)

Figura 2. 15 Dimensión espacial de los datos en un SIG.



Fuente: (Elaboración propia)

2.10 TECNOLOGÍAS WEB

2.10.1 PATRONES DE DISEÑO

- **Modelo:** Se encarga de los datos generalmente (pero no obligatoriamente) consultando la base de datos. Actualizaciones, consultas, búsquedas, etc. todo eso va en el modelo.
- **Vista:** Son la representación visual de los datos, todo lo que tenga que ver con la interfaz gráfica. Ni el modelo ni el controlador se preocupan de cómo se verán los datos, esa responsabilidad es únicamente de la vista.

- **Controlador:** Se encarga de controlar, recibe las órdenes del usuario y se encarga de solicitar los datos al modelo y de comunicárselos a la vista. (Hernandez, 2019)

Figura 2. 16: Modelo de patrones de diseño

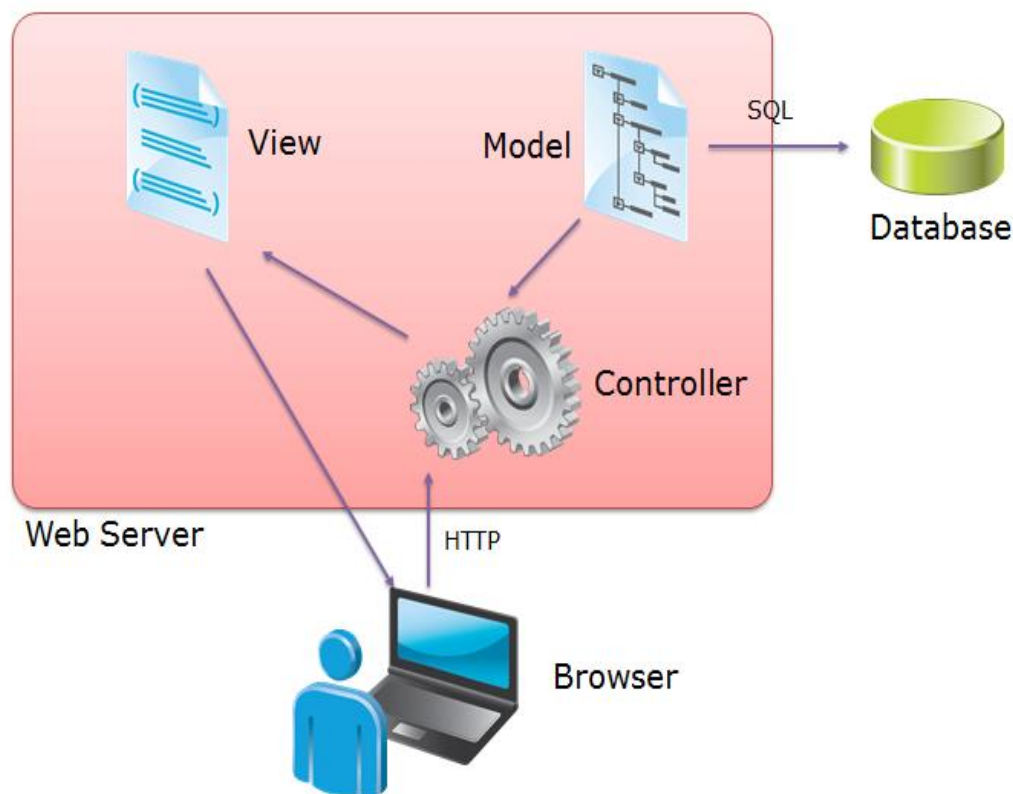


Figura Fuente: (Elaboración propia).

Pero en lo que se refiere a la seguridad de la información, las empresas tienen a su disposición los Sistemas de Gestión de la Seguridad de la Información, que se implantan bajo los requisitos de la norma ISO 27000.

La ISO 27000 es una norma que define de qué manera se debe implantar un Sistema de Gestión de la Seguridad de la Información en una empresa u organización.

Su implantación ofrece a la organización o empresa la ventaja de proteger su información de la forma más fiable posible, persiguiéndose para ello un total de tres objetivos principales:

- **Confidencialidad:** la información no se pone a disposición ni se revela a individuos, entidades o procesos no autorizados.
- **Integridad:** mantenimiento de la exactitud y completitud de la información y sus métodos de proceso.
- **Disponibilidad:** acceso y utilización de la información y los sistemas de tratamiento de la misma por parte de los individuos o procesos autorizados cuando lo requieran.

Tal es su importancia que la implantación de este tipo de herramienta garantiza que los riesgos de seguridad de la información son controlados por la organización eficientemente, tanto de forma interna como al resto de las empresas.

Es interesante tener en cuenta que los sistemas de gestión implantados bajo los requisitos de la norma ISO 27000 son totalmente compatibles con otros sistemas de gestión como son los sistemas de gestión de la calidad.

2.10.2 LENGUAJE DE PROGRAMACION

- a) Php:** Es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el preprocesador 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.
- b) JavaScript:** Lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas

2.10.3 FRAMEWORK

- **Laravel:** Framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el “código espagueti” y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP.

La mayor parte de su estructura está formada por dependencias, especialmente de Symfony, lo que implica que el desarrollo de Laravel dependa también del desarrollo de sus dependencias. Tiene soporte para desarrollar aplicaciones bajo el paradigma MVC (Modelo Vista Controlador) que permite darle un orden a la estructura de tu proyecto

- **VueJs:** Un framework open source de JavaScript, el cual nos permite construir interfaces de usuarios de una forma muy sencilla.

Una de las características más importantes de Vue es el trabajo con componentes. Un componente Vue, en términos simples, es un elemento el cual se encapsula código reutilizable, dentro de un componente podremos encontrar etiquetas HTML, estilos de CSS y código JavaScript. Los componentes nos permiten desarrollar proyectos modularizados y reactivos fáciles de escalar, si nosotros así lo deseamos podemos reemplazar un componente por otro de una forma muy amigable.

- **Bootstrap:** Biblioteca multiplataforma o conjunto de herramientas de código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como extensiones de JavaScript adicionales. (Arias P. , 2020)

- **Highcharts:** Una biblioteca de gráficos moderna que ayuda a los desarrolladores a crear visualizaciones hermosas e interactivas para páginas web.

Es un proyecto de código abierto con licencia del MIT y es de uso gratuito en aplicaciones comerciales. (Highcharts, 2020)

- **Amcharts 4:** Una librería popular que puedes usar para crear diferentes tipos de gráficos. En esta serie, estarás aprendiendo todo sobre los aspectos

importantes de esta librería. Puede ser usada para crear gráficos elegantes y responsivos.

La librería te permite mezclar diferentes tipos de gráficos y trazar datos en escalas fecha tiempo, logarítmica y personalizar con facilidad. La librería también soporta animaciones que pueden ser aplicadas cuando se cambian los datos o se actualizan colores. (Chart.js, 2020)

- **Qgis:** Son sistemas que nos permiten capturar, visualizar, analizar, modificar y compartir información geográfica. Estos sistemas se emplean en diversos ámbitos, como pueden ser: la ordenación del territorio y urbanismo, gestión de recursos, logística, transporte, desarrollo sostenible. (qgis,2029)

2.10.4 BASE DE DATOS

PostgreSQL: Sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL, se utiliza en diversos ámbitos, algunos de los ejemplos donde se hace uso de este gestor de bases de datos:

- Almacenamiento de datos (DWH).
- En servicios como Amazon Web Services Redshift.
- Para procesado de datos, almacenado tanto en la propia instancia como en otros servicios que puedan conectarse.
- En sistemas de información geográfica, como el servicio de mapas web o también en servicios móviles OpenStreetMap.
- En bases de datos para servicios web.

PgAdmin 4: una herramienta indispensable para gestionar y administrar PostgreSQL, la base de datos de código abierto más avanzada. (Marin, 2020).

PostGIS: Es un extensor de base de datos espacial para la base de datos relacional de objetos PostgreSQL . Agrega soporte para objetos geográficos permitiendo que las consultas de ubicación se ejecuten en SQL. (postgis, 2020)



CAPITULO III

MARCO APLICATIVO

3.1 INTRODUCCIÓN

Este capítulo constituye el elemento central para la solución al problema expuesto en el capítulo de Marco Introdutorio, también se considera lo expuesto en el capítulo del Marco Teórico, según la ingeniería de software, en el diseño y desarrollo de todo producto existe una serie de actividades que deben realizarse en un orden determinado que abarca no solo la producción sino también la explotación y el mantenimiento, por lo cual se desarrolla en base a las fases de la metodología de desarrollo UWE en el módulo financiero para el sistema de forestaciones y reforestaciones caso proyecto UNJP/BOL/045/UNJP.

UWE, (UML Based Web Engineering - Ingeniería Web Basado en UML). Las fases de esta metodología son: análisis de requerimiento, diseño conceptual, diseño de navegación y diseño de presentación, que nos permiten diversos diagramas para el desarrollo del software.

El presente proyecto se enfoca en dar una solución a la problemática que existe en la generación de datos económicos, generación de reportes, reportes dinámicos, por diferentes niveles y la exportación de datos en diferentes formatos.

Analizando los datos del sistema de forestaciones y reforestaciones de la institución se concluyó en desarrollar un módulo financiero con fin de mejorar la generación de la información, en base a reportes que les permitan a las diferentes instituciones y usuarios obtener rápidamente los datos financieros que necesitan para su información.

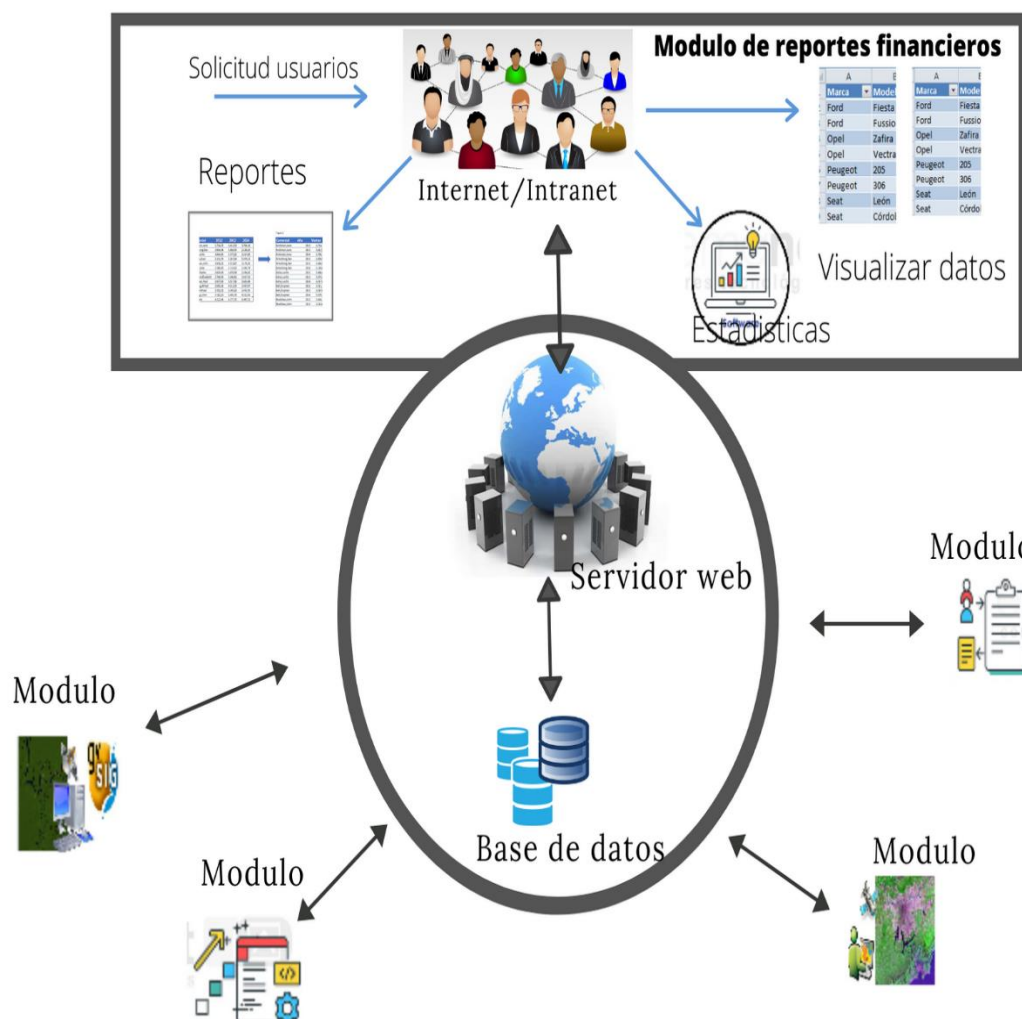
3.2 ESQUEMA DEL SISTEMA

El módulo de reportes financiero para el sistema de forestaciones y reforestaciones se desarrolla bajo las siguientes tuiciones que facilitan los elementos básicos para el proceso de desarrollo de software.

Actualmente el sistema de forestaciones y reforestaciones existe en la institución que consta de varios módulos, el módulo de reportes financieros se integra como un módulo más al sistema de forestaciones y reforestaciones, por lo cual el módulo de reportes financieros permite visualizar, generar datos y exportar en diferentes formatos

en función a los datos de la base de datos del sistema, a continuación se muestra el siguiente esquema.

Figura 3. 1: Esquema del sistema



Fuente: (Elaboración propia)

3.3 METODOLOGÍA UWE

La metodología UWE propone construir un conjunto de resultado de las fases de análisis, diseño e implementación del proceso, la fase de análisis apunta a obtener un conjunto de requerimientos estables, los requerimientos funcionales son capturados a través del modelo de requerimientos, el modelo de requerimientos conforma casos de uso especializados y un modelo de clases para la aplicación web.

La fase de diseño consiste en construir una serie de modelos para el contenido, navegación, proceso, presentación y aspectos adaptativos a un nivel de plataforma independiente. Las transformaciones implementan la construcción sistemática de modelos dependientes por la generación de modelos por defecto, los cuales pueden ser refinados por el diseñador y finalmente, los modelos de diseño son transformados a una implementación de plataforma específica. Este proceso se extiende con la construcción de diagramas de estado UML (denominado la gran foto) que integran los modelos de diseño.

3.3.1 MODELO DE CASOS DE USO

Dentro de los casos de uso se divide en vistas de acuerdo al requerimiento del usuario, esto dependiendo de la funcionalidad que tenga el sistema, una técnica que hace la fuerza para definir quiénes son los actores (usuario) de la aplicación web dando a ofrecer una mejor manera de representar la funcionalidad y la aplicación en cada uno de los actores.

3.3.1.1 DESCRIPCION DE CASOS DE USO

Todo actor representa a los usuarios del sistema, donde se comprende como un usuario a cualquier actor (usuario) que llega a interactuar con el sistema, en la siguiente tabla se describe los actores del módulo financiero para el sistema reforestaciones y forestaciones caso, proyecto unjp/bol/045/unj.

Tabla 3. 1: Tabla de descripción de casos de uso

Actor:	Descripción:
M.A.E.	Autoridades Gubernamentales y Autoridades Máximas de las diferentes Instituciones.
SuperAdministrador	Encargado general del sistema de forestación y reforestación, es el que tiene los privilegios máximos del sistema.
Administrador	Personal encargado del sistema (técnico)

 Usuario

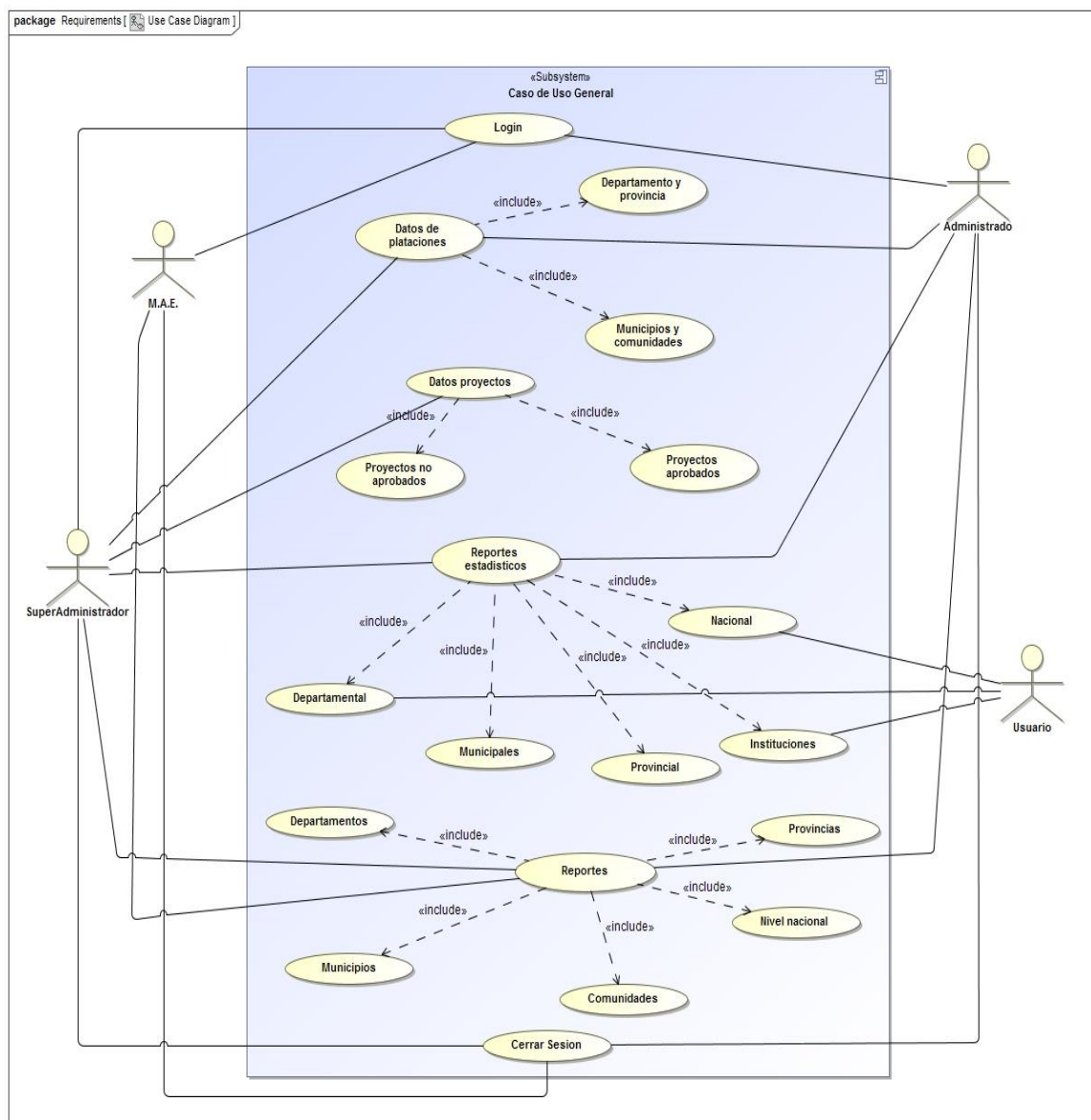
 Cualquier persona natural y/o jurídica

Fuente: (Elaboración propia)

3.3.1.2 CASO DE USO GENERAL

En el siguiente caso de uso general se describe las actividades que pueden realizar los diferentes actores, MAE (máxima autoridad ejecutiva), SuperAdministrador, Administrador y usuarios.

Figura 3. 2: Diagrama de casos de uso general



Fuente: (Elaboración propia)

3.3.1.3 ESPECIFICACIÓN DE CASOS DE USO

a) Autenticación de usuario

La actividad que se muestra en la siguiente tabla y diagrama de caso de uso corresponde a la autenticación de usuarios con privilegios para ingresar al sistema.

Tabla 3. 2: Tabla descripción de caso de uso autenticación

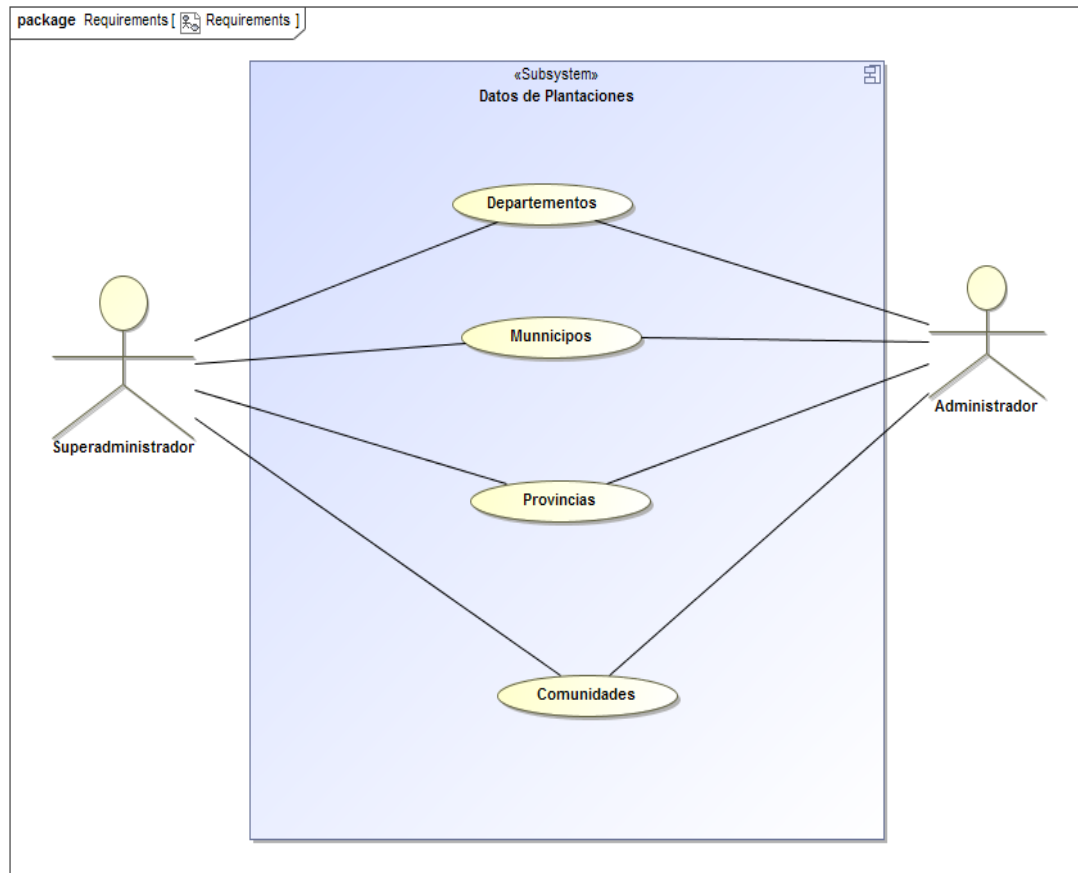
Caso de uso:	Autenticación de usuario
Actor:	M.A.E., superAdministrador y el administrador
Descripción:	Personal con credenciales de acceso al sistema para su autenticación y de esta forma poder realizar las actividades correspondientes.
Precondición:	Los actores deben ser registrados previamente en la base de datos del sistema.
Postcondición:	Para la autenticación el usuario debe contar con credenciales registradas y su nivel de privilegio.
Flujo básico del evento:	<ol style="list-style-type: none"> 1. El usuario ingresa al modulo 2. Si necesita más datos, solicita usuario y contraseña. 3. El modulo valida los datos y verifica el nivel de privilegio que tiene
Flujo alternativo	Si los datos son incorrectos el modulo no le permite el acceso.

Fuente: (elaboración propia)

b) Generar datos de plantaciones

En la generación de datos de plantaciones los actores principales son el superAdministrador y el administrador

Figura 3. 3: Casos de uso datos de plantaciones



Fuente: (Elaboración propia)

Tabla 3. 3: Tabla generar datos de plantaciones

Caso de uso:	Datos de plantaciones
Actor:	SuperAdministrador y el administrador
Descripción:	La generación de datos de plantaciones se realiza por departamentos, provincias, municipal y comunidades.
Precondición:	El usuario debe ser registrado previamente en la base de datos del sistema y autenticado.
Postcondicion:	Puede generar los datos en forma dinámica.
Flujo básico del evento:	<ol style="list-style-type: none"> 1. El usuario ingresa al módulo por la opción datos de plantaciones. 2. Puede generar los datos por diferentes niveles.

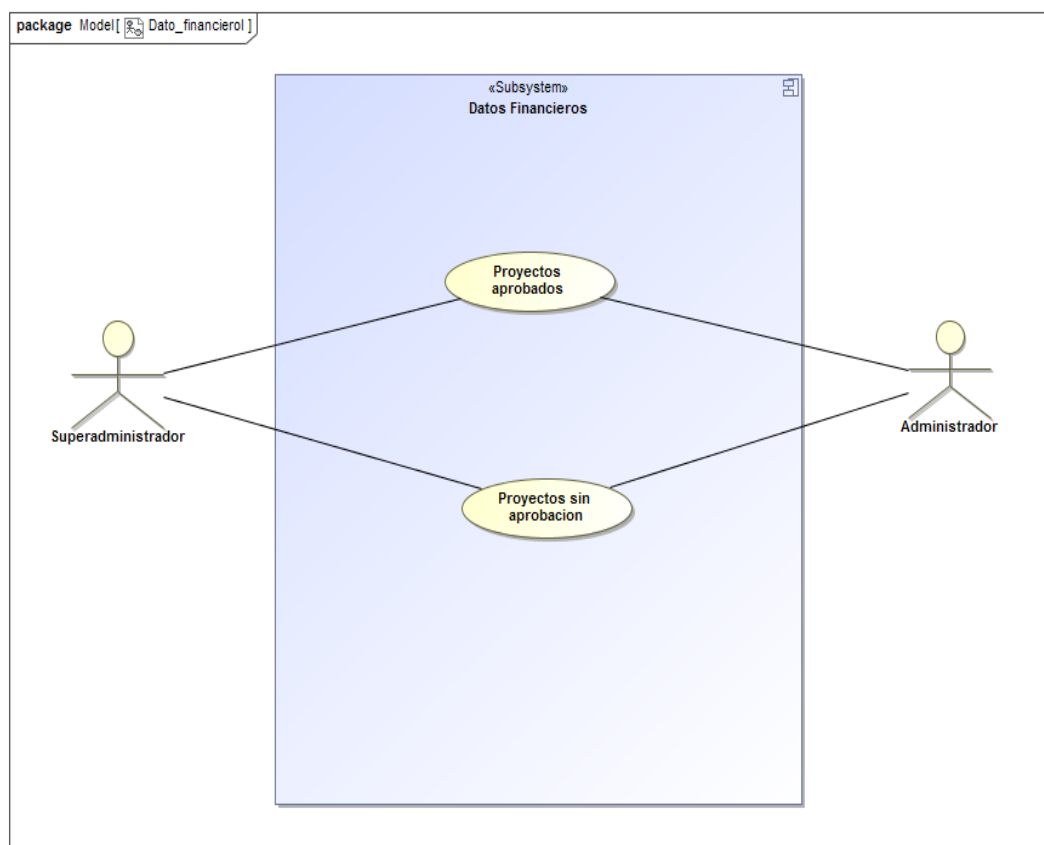
	3. Nivel, departamental, provincial, municipal y comunidad.
	4. Generar reportes
Flujo alternativo	No puede realizar modificaciones

Fuente: (Elaboración propia)

c) Visualizar proyectos con y sin ubicación geográfica

El mencionado caso de uso consiste en visualizar proyectos que son aprobados bajo las planificaciones por ende estos proyectos son ejecutados y por otra parte los proyectos que no fueron realizados por alguna razón pero son registrados en la base de datos.

Figura 3. 4: Casos de uso visualización de proyectos con y sin ubicación.



Fuente: (Elaboración propia)

Tabla 3. 4: Tabla obtener datos de proyectos sin y con ubicación geográfica

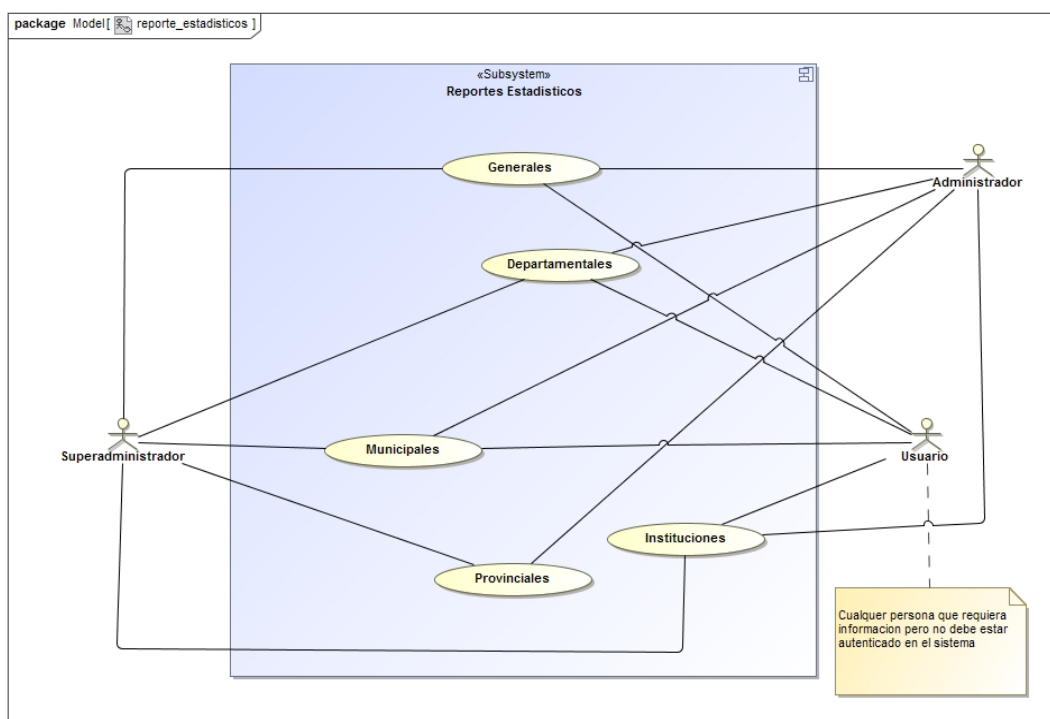
Caso de uso:	Proyectos
Actor:	SuperAdministrador y el administrador
Descripción:	Se puede visualizar proyectos con o sin ubicación geográfica según si están aprobados y no aprobados.
Precondición:	El usuario debe ser registrado previamente en la base de datos del sistema y autenticado.
Postcondicion:	Puede generar los datos en forma dinámica y visualizar la ubicación.
Flujo básico del evento:	<ol style="list-style-type: none"> 1. El usuario ingresa al módulo por la opción proyectos. 2. Puede visualizar los diferentes proyectos. 3. Visualizar la ubicación exacta de cada lugar.
Flujo alternativo	No puede realizar modificaciones

Fuente: (Elaboración propia)

d) Generar reportes estadísticos

El caso de uso generación de reportes estadísticos permite generar reportes dinámicos en formas estadísticas en el módulo financiero por diferentes niveles, nivel nacional, nivel departamental, nivel provincial y por nivel municipal los principales actores para este módulo son: el superAdministrador y administrador.

Figura 3. 5: Casos de uso reportes estadísticos



Fuente: (Elaboración propia)

Tabla 3. 5: Tabla caso de uso reportes estadísticos

Caso de uso:	Reportes estadísticos
Actor:	SuperAdministrador y el administrador
Descripción:	Permite al usuario a generar y visualizar datos en cuadros estadísticos.
Precondición:	El usuario debe ser registrado previamente en la base de datos del sistema y autenticado.
Postcondicion:	Puede generar los datos en forma dinámica y visualizar en gráficos estadísticos.
Flujo básico del evento:	<ol style="list-style-type: none"> 1. Usuario ingresa al módulo por la opción reportes estadísticos. 2. Puede generar los datos por diferentes niveles departamental, provincial y municipal.

	3. Visualizar en formas estadísticas.
	4. Generar reportes
Flujo alternativo	No puede realizar modificaciones

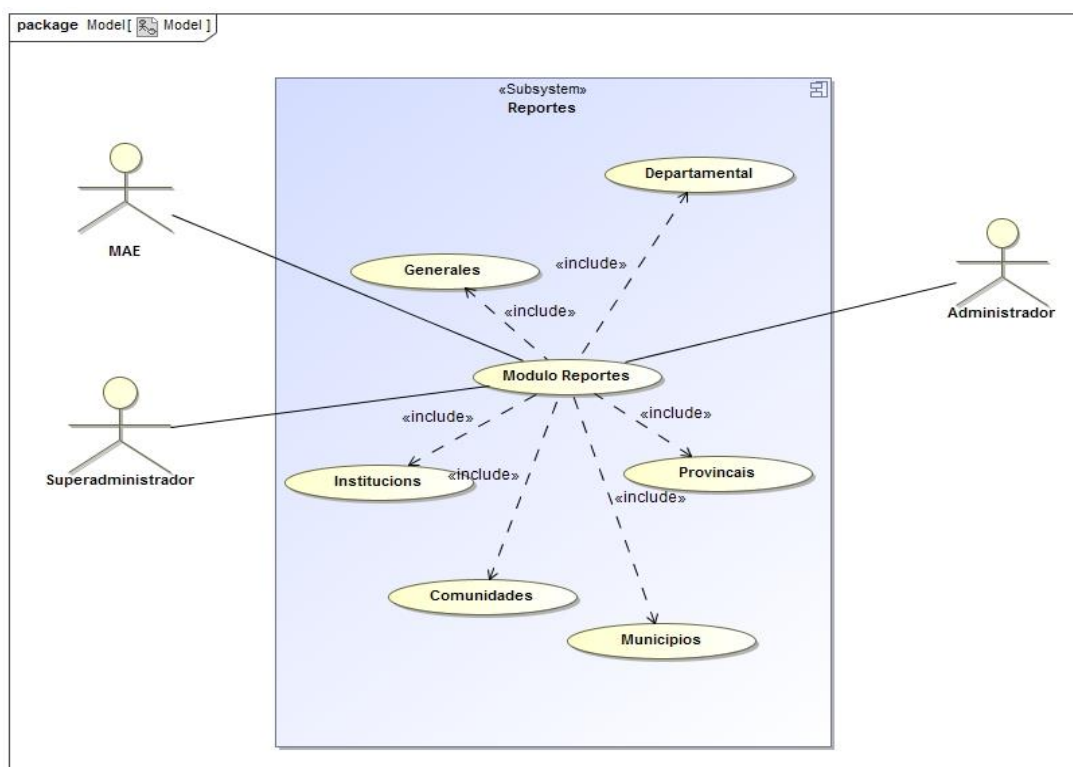
Fuente: (Elaboración propia)

e) Reportes

El caso de uso reportes realiza la generación de reportes detallados por diferentes niveles como es: Nacional, Departamental, Municipal, Comunidad, por Instituciones que participan en diferentes proyectos, demás por tipos de plantaciones y proyectos monitoreados con ubicación geográfica como también de los que no tienen una ubicación.

Los actores involucrados en este caso de uso de reportes son M.A.E, SuperAdministrador, Administrador.

Figura 3. 6: Casos de uso reportes



Fuente: (Elaboración propia)

Tabla 3. 6: Tabla de casos de uso reportes

Caso de uso:	Reportes
Actor:	MAE, SuperAdministrador y administrador
Descripción:	Permite generar reportes dinámicos por diferentes niveles.
Precondición:	El usuario debe ser registrado previamente en la base de datos del sistema y autenticado.
Postcondicion:	Puede generar los datos en forma dinámica y visualizar en tablas.
Flujo básico del evento:	<ol style="list-style-type: none"> 1. Usuario ingresa al módulo por la opción reportes. 2. Puede generar los reportes por diferentes niveles, nacional, departamental, provincial, municipal y comunidad. 3. Visualizar en tablas. 4. Generar reportes en diferentes formatos.
Flujo alternativo	No puede realizar modificación alguna.

Fuente: (Elaboración propia)

f) Obtener datos sin autenticación

El mencionado caso de uso que permite a todas las personas externas e internas que no tienen privilegios de autenticación, ver los datos limitados de forestaciones y reforestación sin la autenticación al módulo financiero.

Tabla 3. 7: Tabla de caso de uso generar datos sin autenticación.

Caso de uso:	Reportes
Actor:	Usuarios
Descripción:	Permite al usuario a visualizar datos.
Precondición:	El usuario solo necesita acceder al módulo financiero.
Postcondicion:	-----

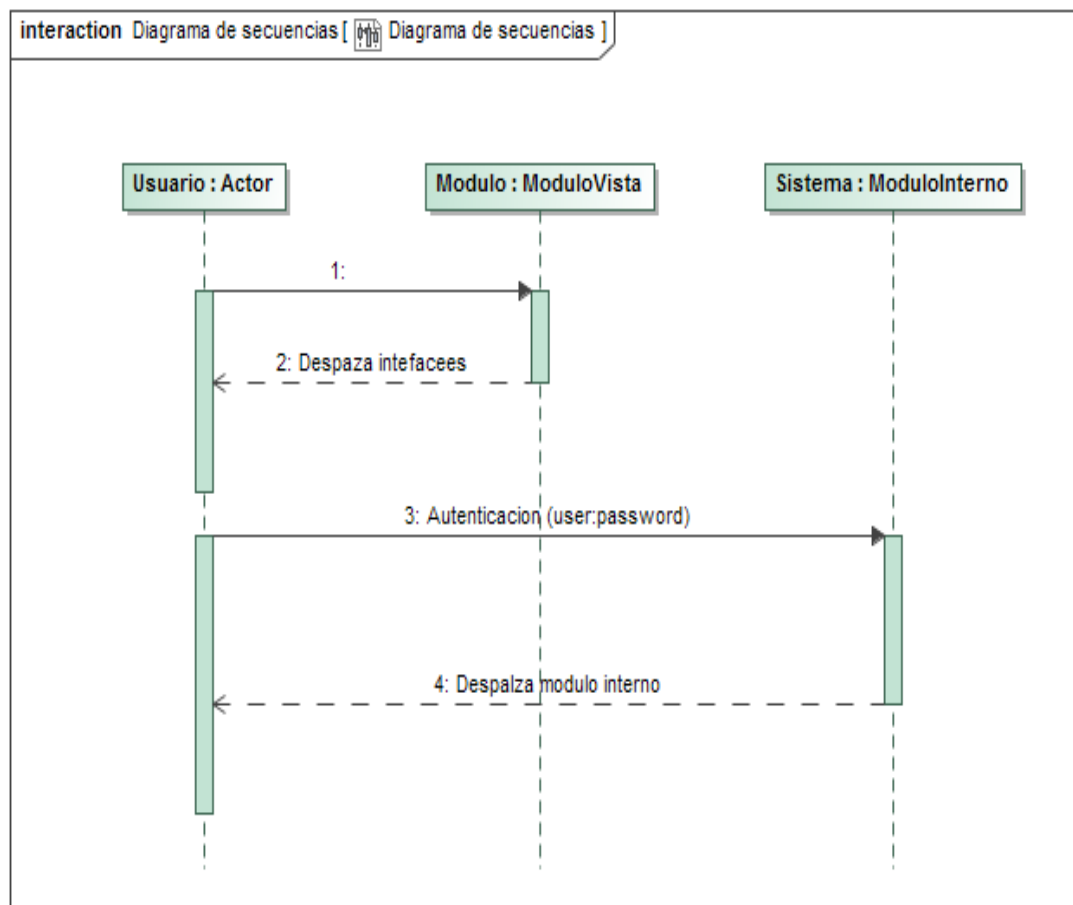
Flujo básico del evento:	<ol style="list-style-type: none"> 1. Usuario accede al módulo. 2. Visualizar datos según el nivel requerido por el botón generar datos. 3. Visualizar en figuras estadísticas a nivel nacional departamental y por instituciones.
Flujo alternativo	Solo puede visualizar datos limitados y generales.

Fuente: (Elaboración propia)

3.3.2 DIAGRAMA DE SECUENCIAS

- Diagrama de secuencia ingreso a la vista de página principal, el diagrama de secuencias describe el acceso según el usuario que cuenta con privilegios y usuario que no tiene privilegios, para su autenticación al sistema.

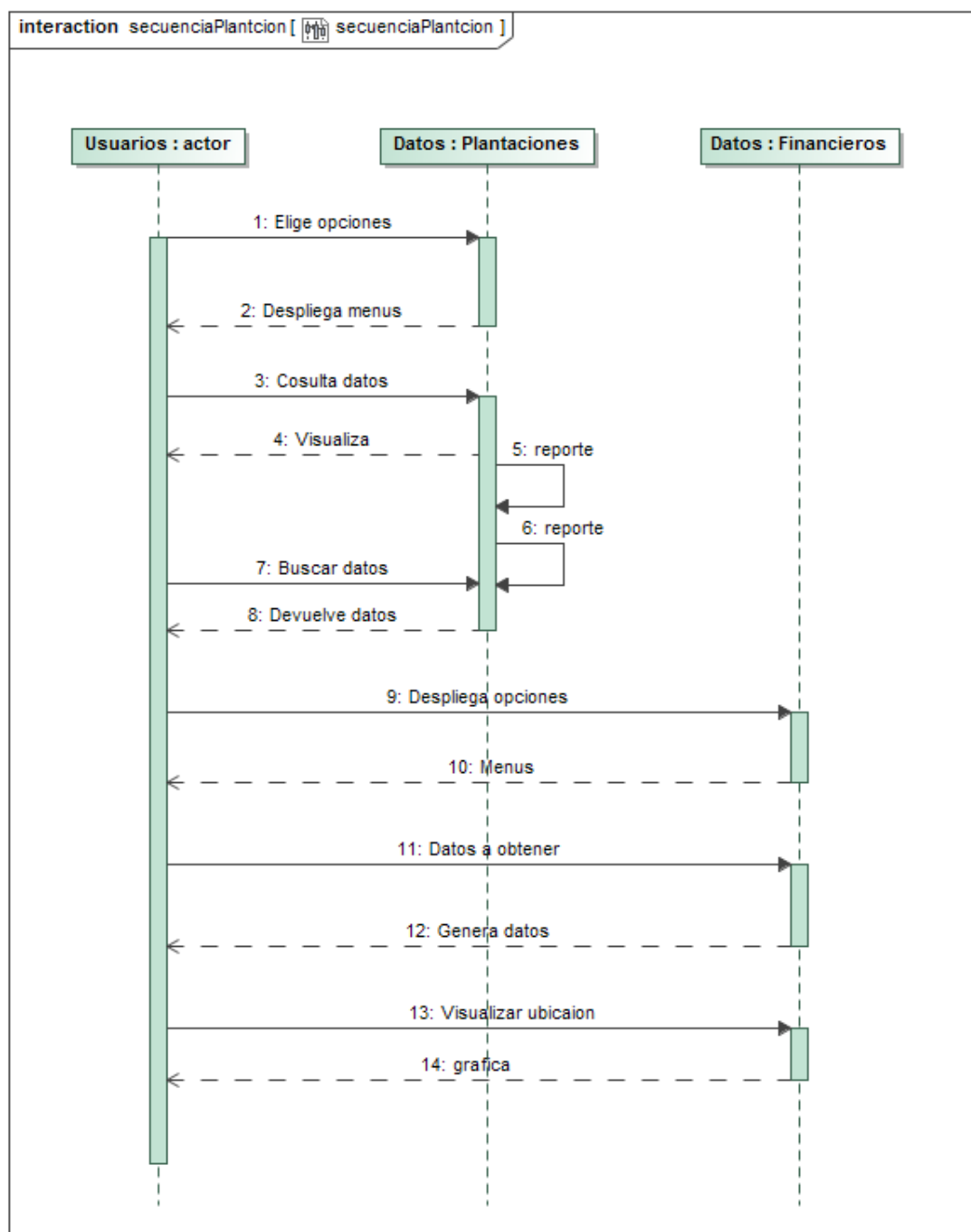
Figura 3. 7: Diagrama de secuencias autenticación de usuarios



Fuente: (Elaboración propia)

- Diagrama de secuencia de datos de plantaciones y proyectos aprobados con y sin ubicación geográfica.

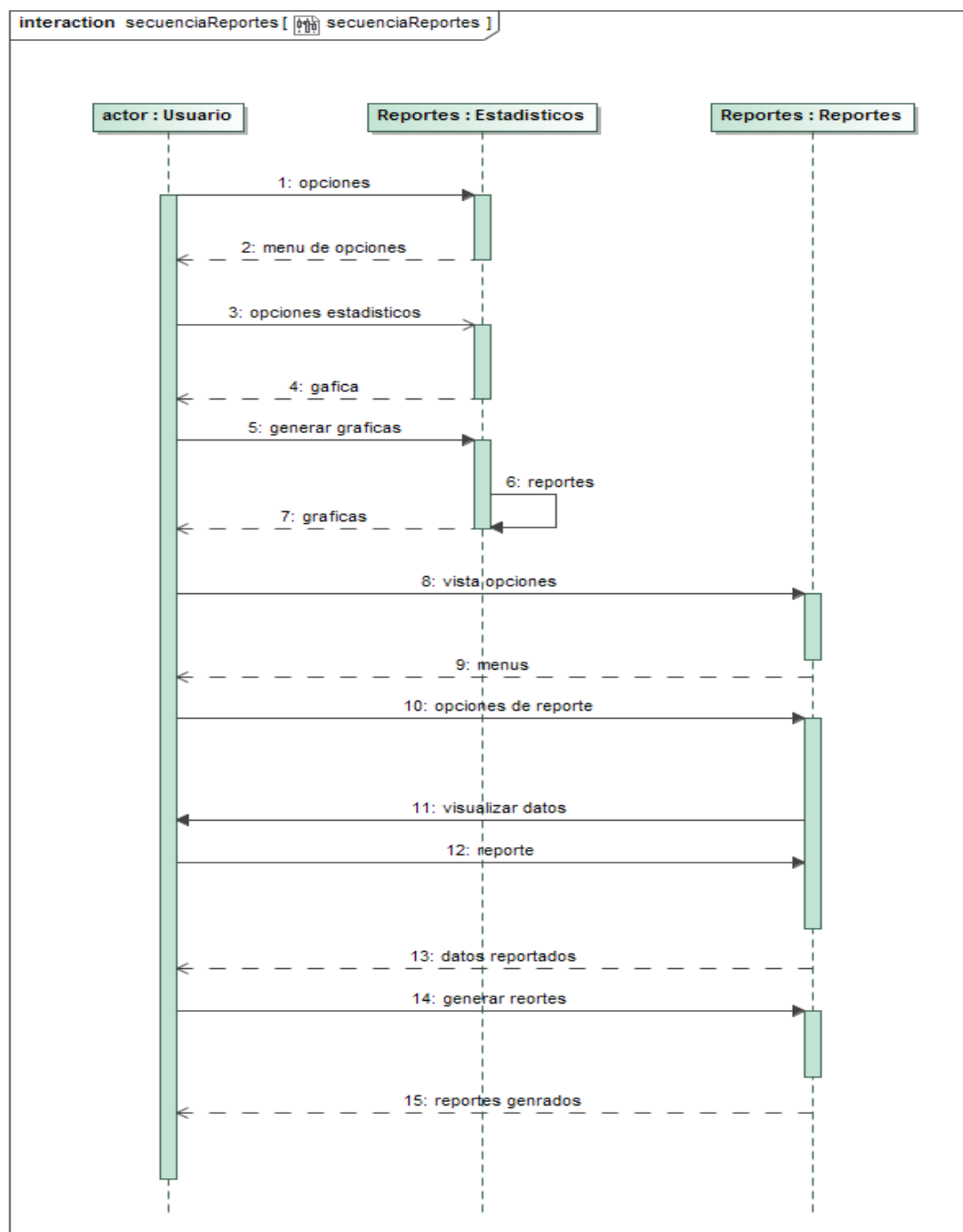
Figura 3. 8: Diagrama de secuencias de plantaciones y proyectos



Fuente: (Elaboración propia)

- Diagrama de secuencia de generación de gráficos estadísticos y reportes dinámicos por diferentes niveles.

Figura 3. 9: Diagrama de secuencias de gráficos estadísticos y reportes

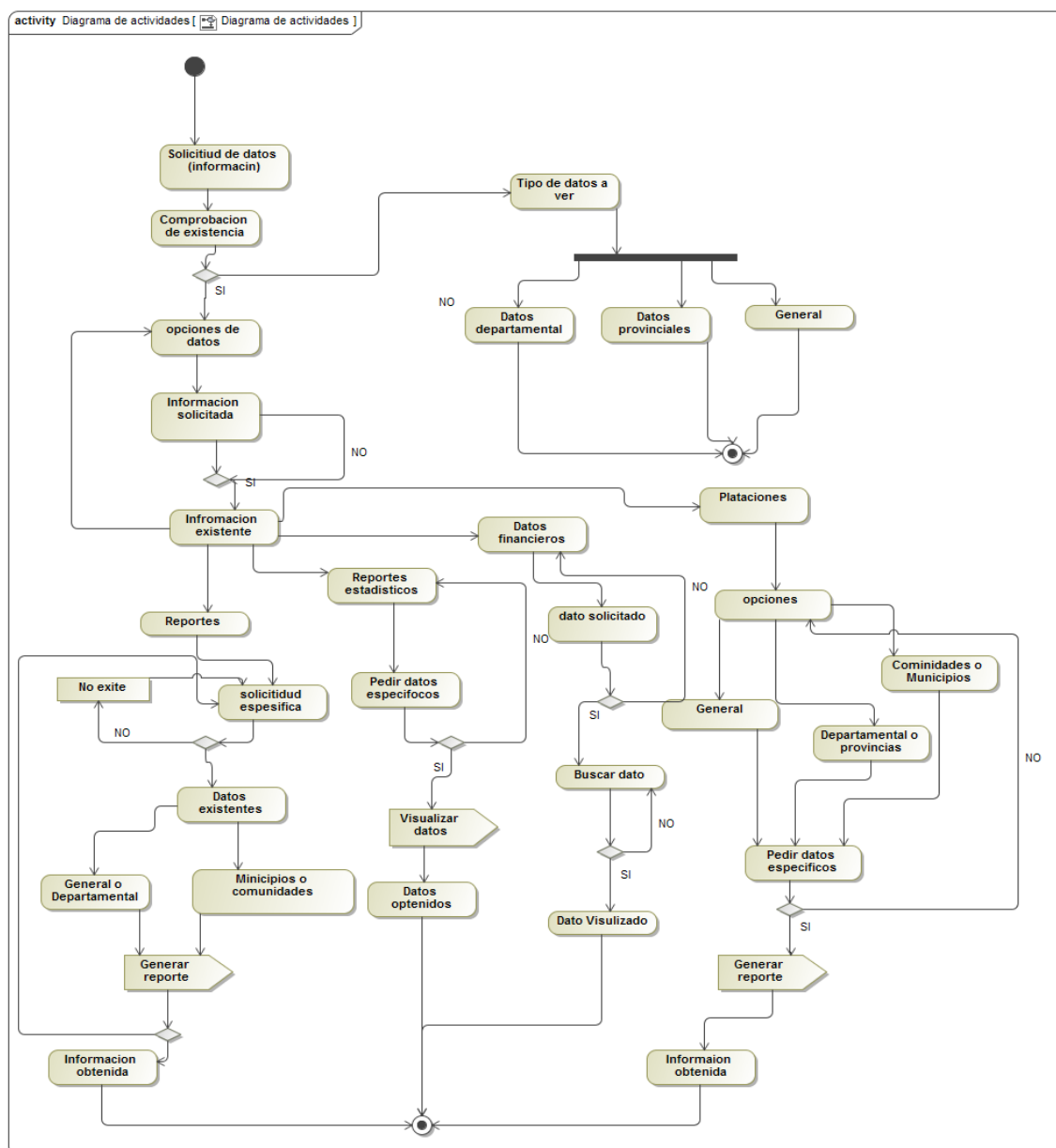


Fuente: (Elaboración propia)

3.3.3 DIAGRAMA DE ACTIVIDADES

Los diagramas de actividades nos muestran los flujos que ocurren entre actividad a actividad. El usuario inicia el proceso de autenticación, esta es verificada, si tiene credenciales ingresa al sistema según sus roles y puede realizar las actividades que requiera, de lo contrario el sistema lo niega el acceso, un usuario sin credenciales solo puede ingresar a la página principal.

Figura 3. 10: Diagrama de actividades



Fuente: (Elaboración propia)

3.3.4 MODELO DE CONTENIDO

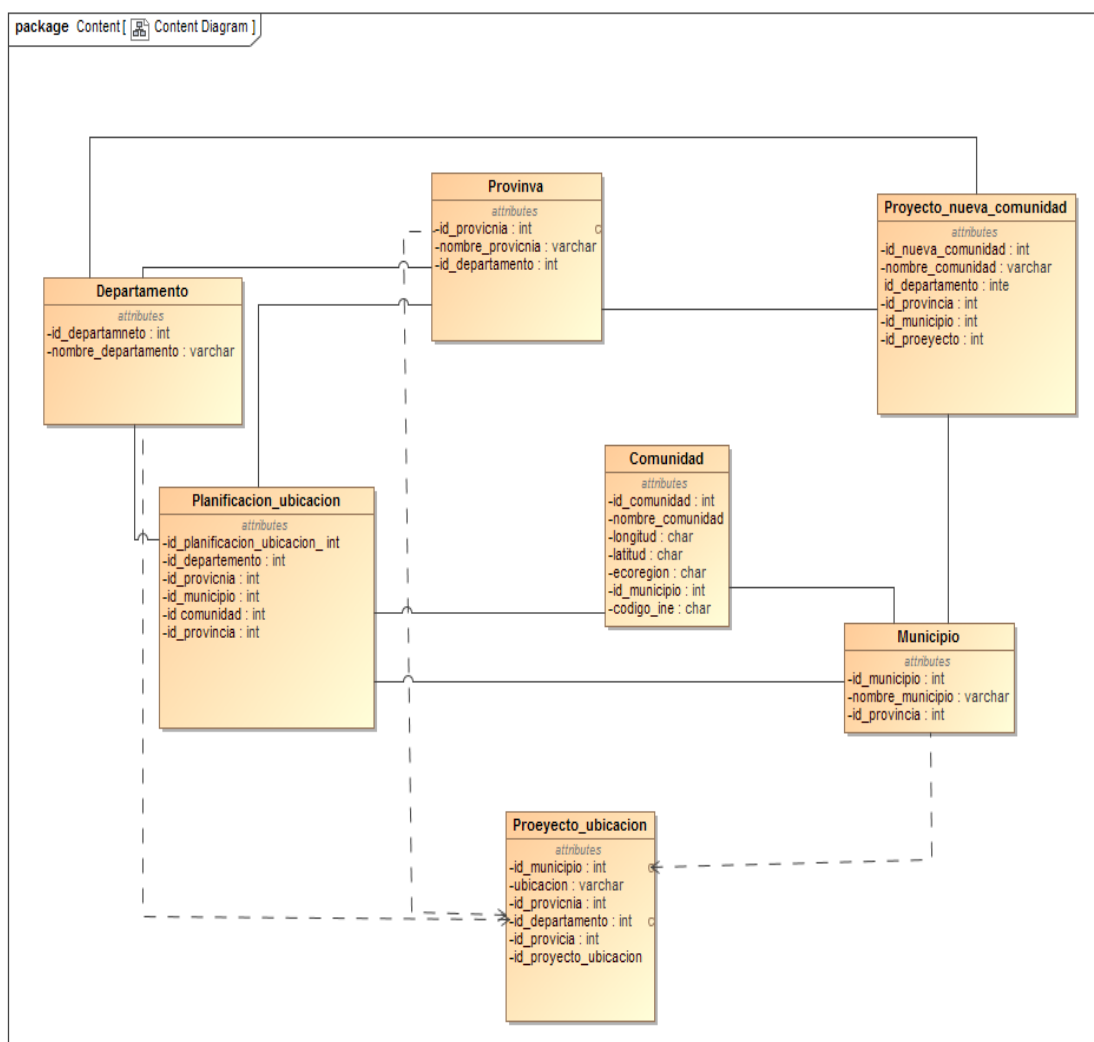
3.3.4.1 DISEÑO CONCEPTUAL

Los diagramas del diseño conceptual, describen cada una de las clases haciendo una especificación visual de la información del módulo web en diagramas UML.

Los modelos de datos aportan la base conceptual para diseñar la aplicación que hace un uso intensivo de datos, así como la base formal para las herramientas y técnicas empleadas en el desarrollo y en el uso del sistema de forestaciones y reforestaciones.

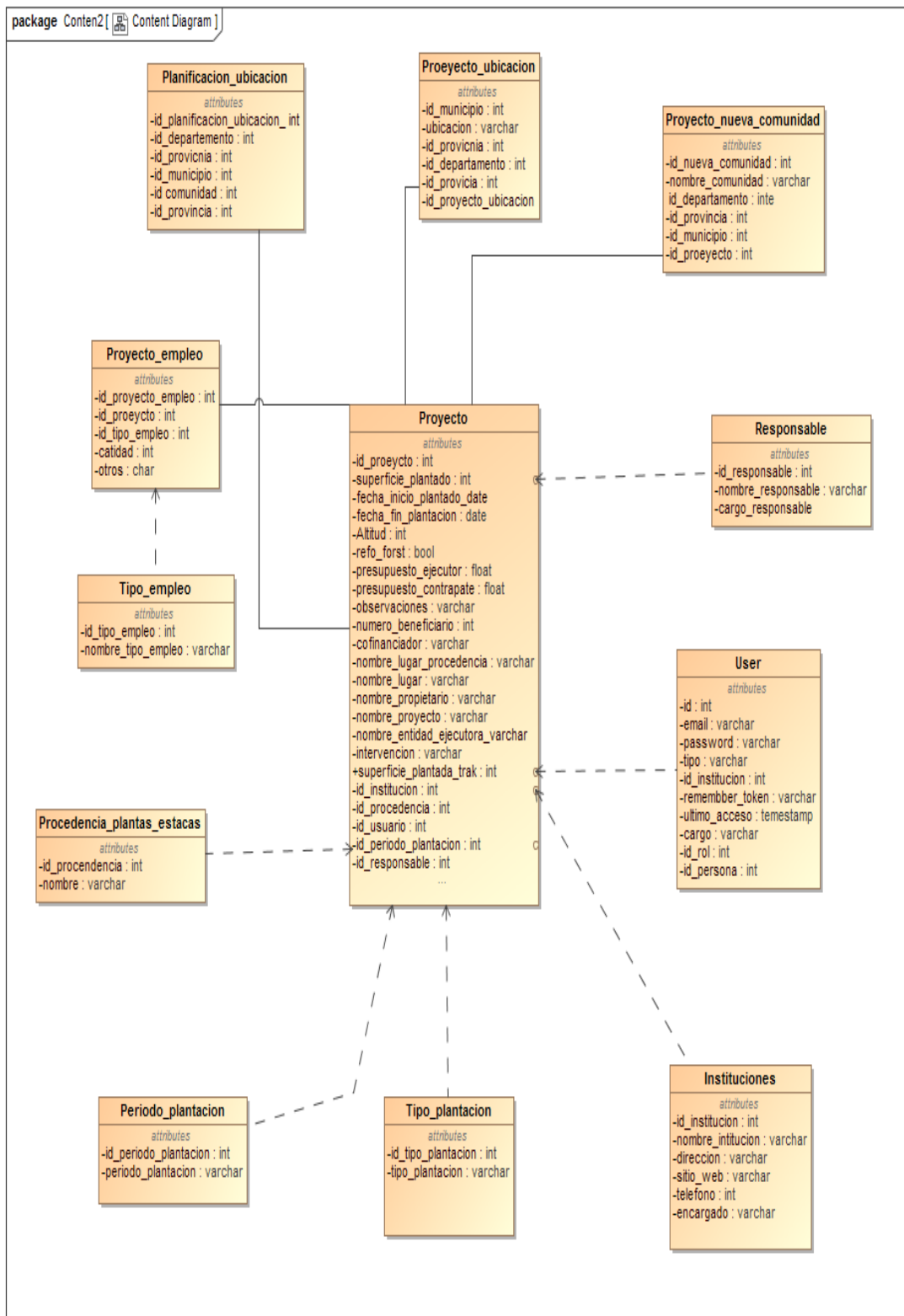
En los siguientes gráficos se visualiza el modelado conceptual de datos:

Figura 3. 11: Ilustración del diseño conceptual parte 1



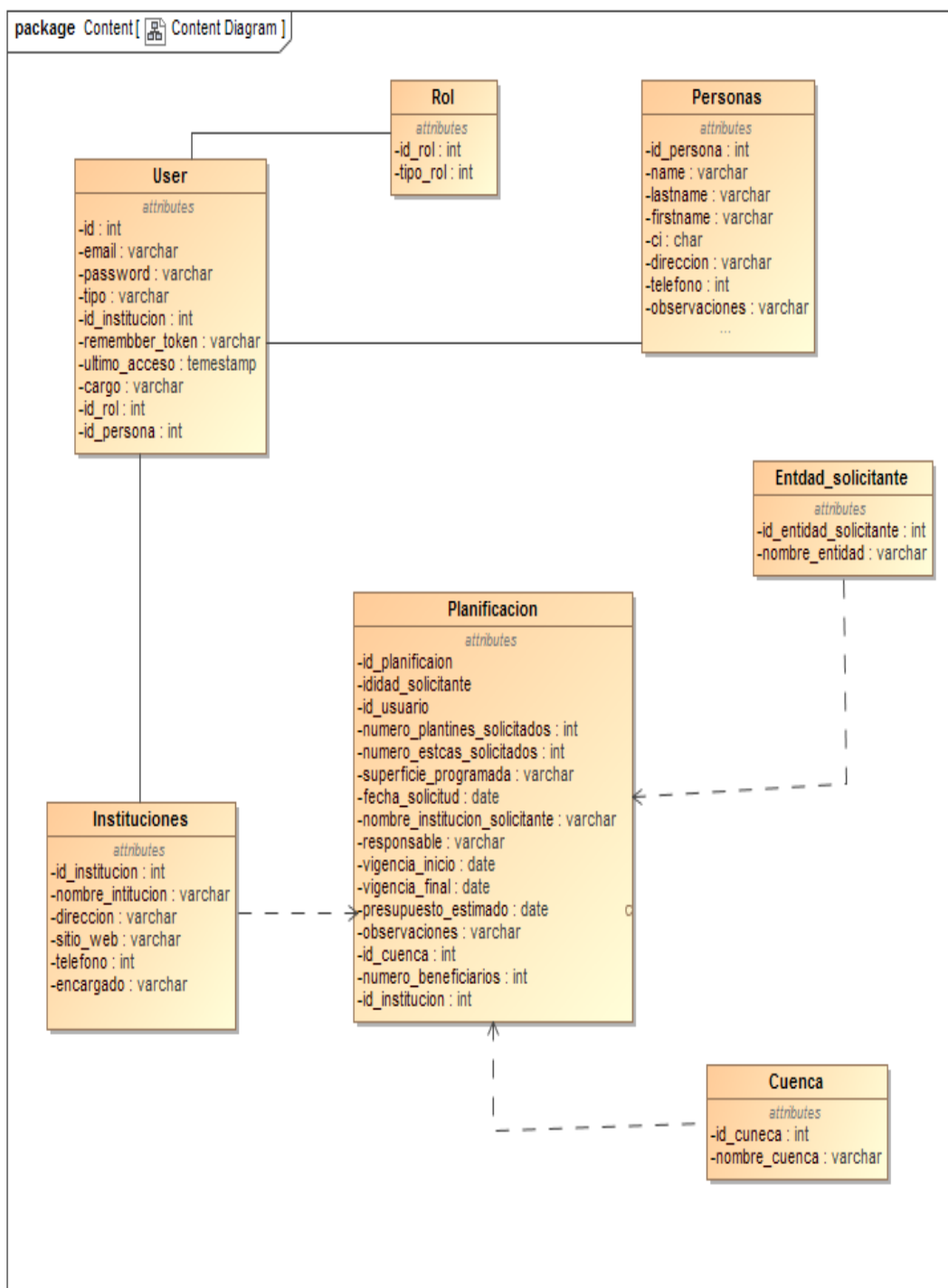
Fuente: (Elaboración propia)

Figura 3. 12: Ilustración del diseño conceptual parte 2



Fuente: (Elaboración propia)

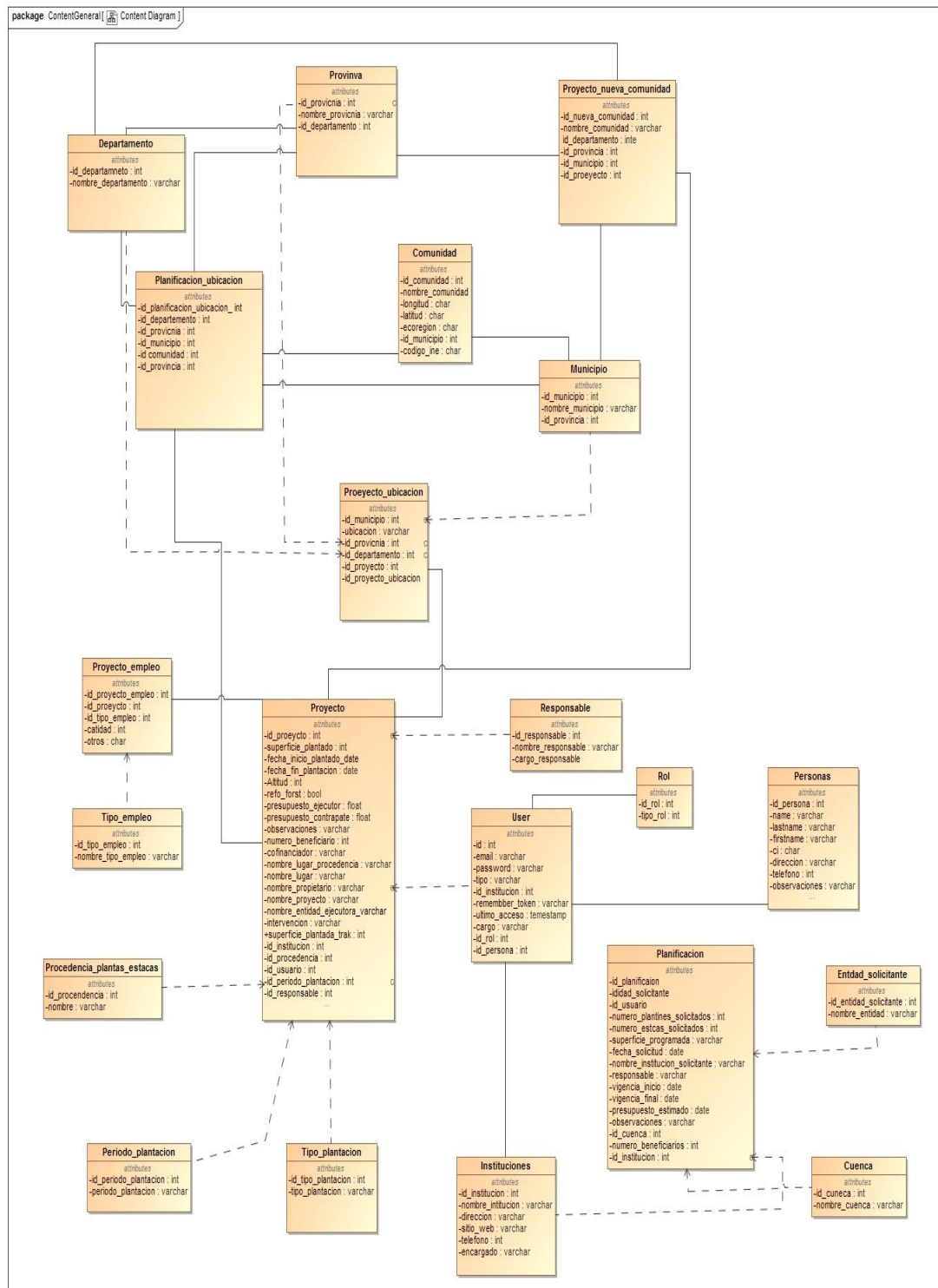
Figura 3. 13: Ilustración del diseño conceptual parte 3



Fuente: (Elaboración propia)

Diseño conceptual general:

Figura 3. 14: Ilustración del diseño conceptual general



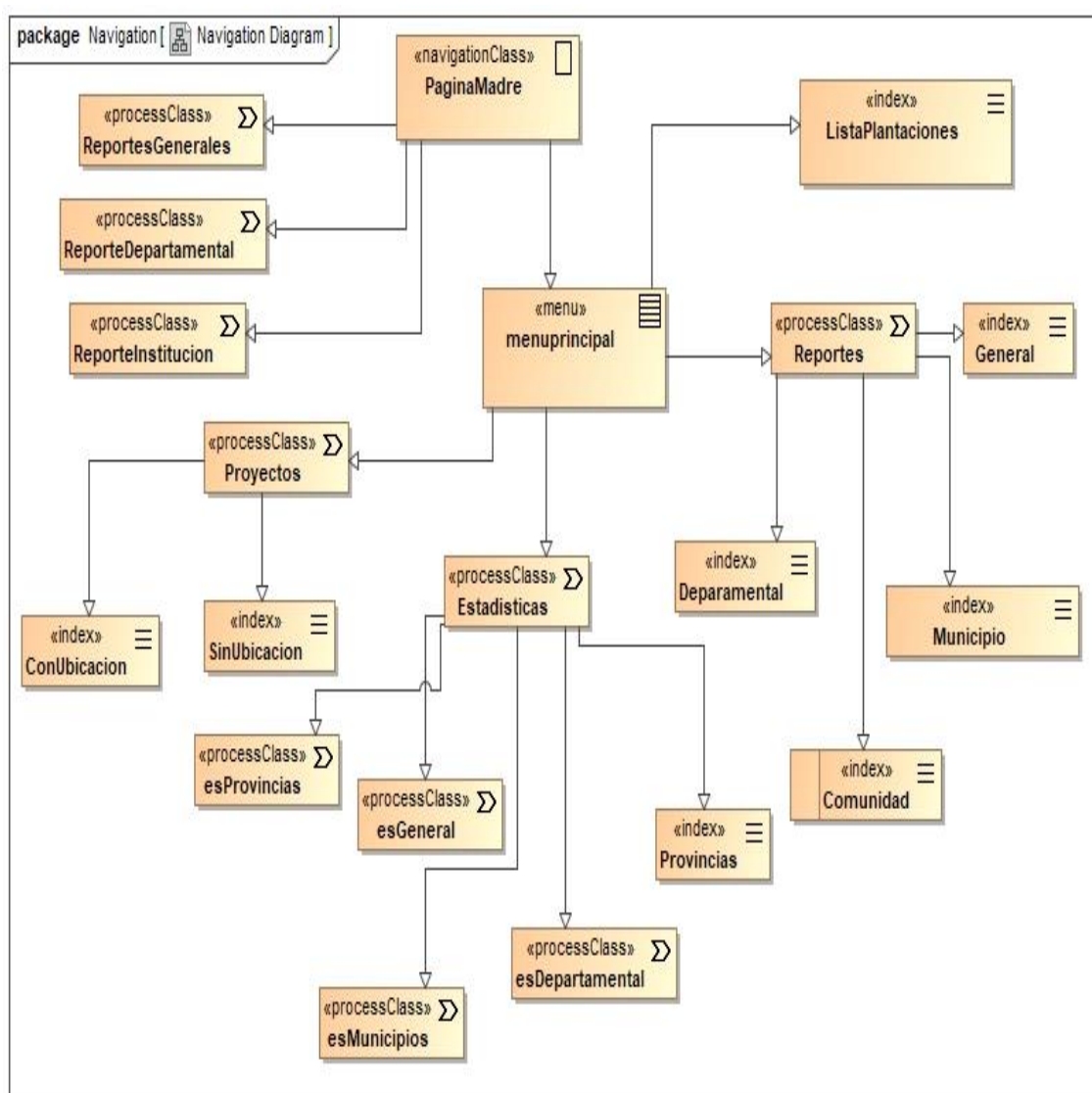
Fuente: (Elaboración propia)

3.3.5 MODELO DE NAVEGACIÓN

En la fase del modelo navegación la metodología UWE, selecciona los diagramas apropiados para, mejorar la expresión o la visualización de la construcción de la aplicación web. Adicionalmente UWE introduce clases navegacionales que son parte del modelo de navegación y otros elementos.

El usuario visualiza primero la página madre del sistema, si necesita más datos ingresa con sus credenciales de autenticación al módulo principal para poder navegar en los distintos módulos internos que tiene, se observa en la figura 3.15.

Figura 3. 15: Modelo de navegación



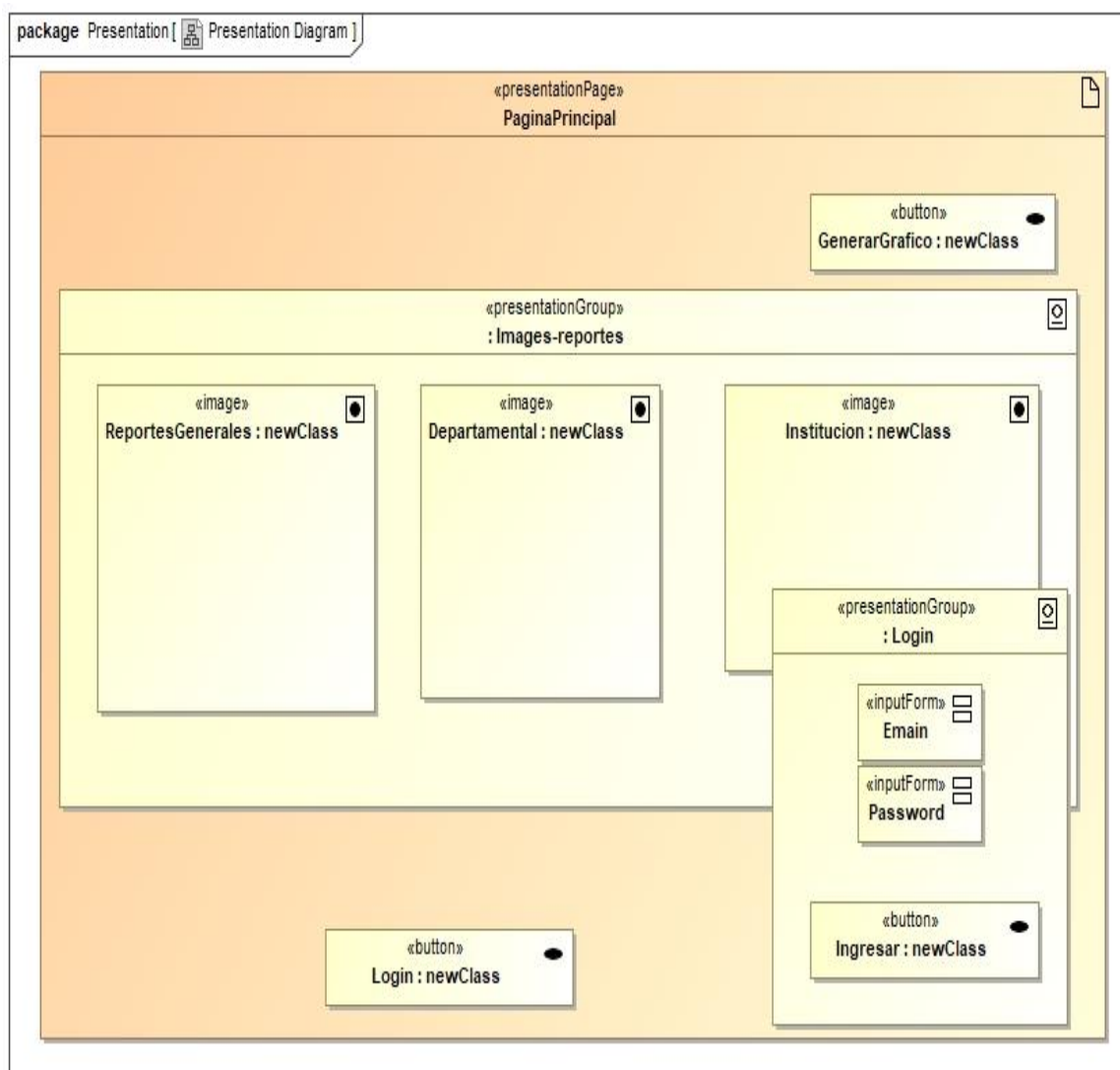
Fuente: (Elaboración propia)

3.3.6 MODELO DE PRESENTACIÓN

El modelo de navegación no muestra qué clases de navegación y proceso pertenecen a qué parte de la página web. Por lo cual podemos utilizar un diagrama de presentación para proporcionar esta información.

La metodología UWE propone construir un bosquejo de la página web y luego asociarlo en escenarios para mostrar al usuario en que forma interactuara con la paliación web, se muestra en la siguiente figura la página principal del sistema incluyendo el módulo de autenticación de usuario.

Figura 3. 16: Diagrama de presentación, pagina madre del módulo

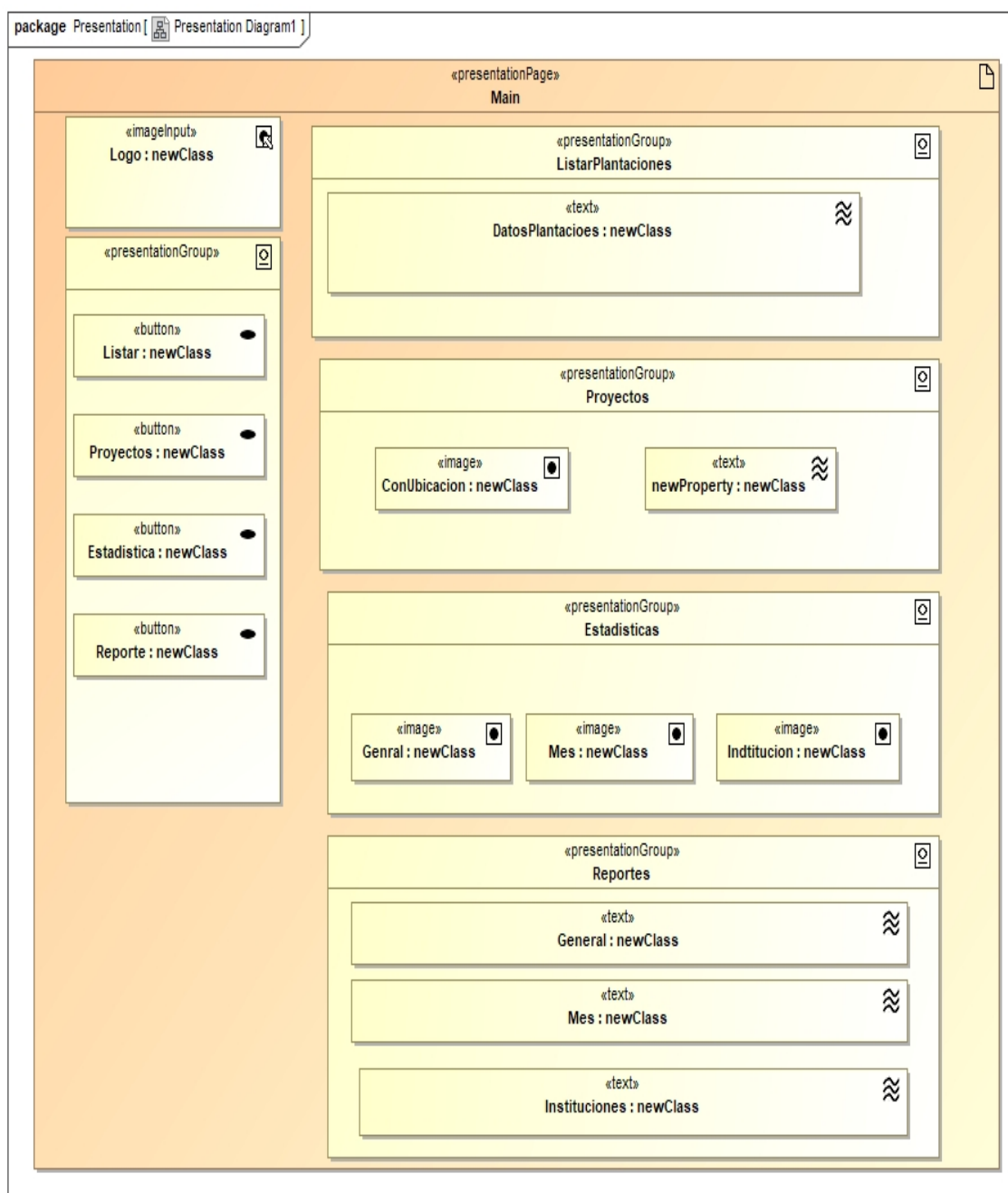


Fuente: (Elaboración propia)

Diagrama de presentación del módulo principal del sistema con los sub módulos, datos de plantaciones, proyectos, reportes estadísticos y reportes.

Para poder acceder a este módulo el usuario debe estar previamente registrado en la base de datos y autenticarse con los roles correspondientes.

Figura 3. 17: Diagrama de presentación, modulo principal



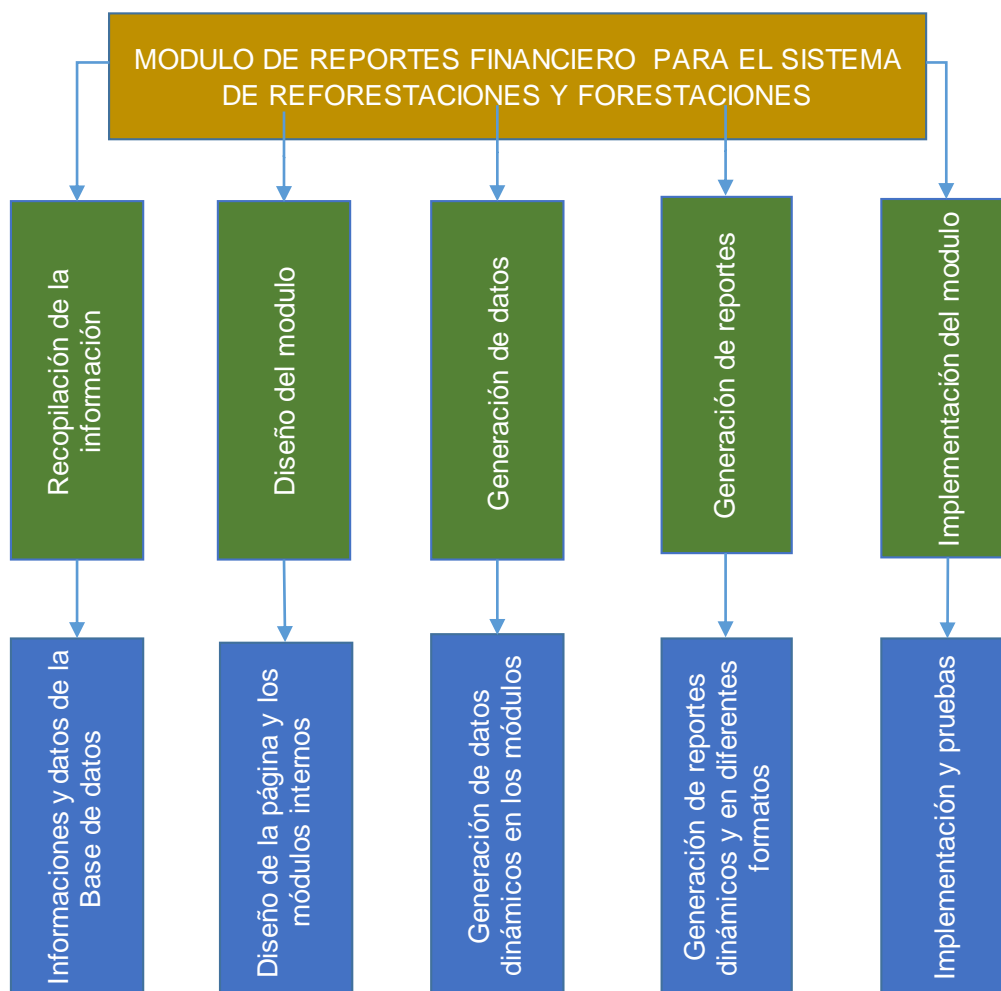
Fuente: (Elaboración propia)

3.4 DESARROLLO DEL SISTEMA

El objetivo de un proceso de software es la producción de manera eficaz y eficiente del producto, donde involucra creatividad y decisiones para las personas involucradas en el desarrollo del sistema, en las actividades realizadas se constituye desde el análisis, la recopilación de datos (requisitos) del sistema, desarrollo del sistema y la puesta en producción del nuevo sistema.

A continuación se describen las diferentes actividades realizadas en el marco del trabajo del módulo financiero de forestaciones y reforestaciones.

Figura 3. 18: Actividad realizada en el desarrollo de sistema



Fuente. (Elaboración propia)

3.4.1 ESTRUCTURA MVC

Es un patrón arquitectónico de software que separa una aplicación en tres capas descritas como su acrónimo lo indica (MVC). En el framework laravel en base a su lenguaje de programación PHP implementan este patrón de diseño, donde cada capa maneja un aspecto de la aplicación separando el código en tres partes bien definidas el modelo, la vista y el controlador.

Modelos

Los modelos son clases encargadas de trabajar con las consultas de la base de datos, es decir que por cada tabla tendremos una clase, cada registro será un objeto y las consultas se llamarán a través de métodos de esas clases. A su vez laravel trabaja con eloquent que es un ORM que nos facilitará el trabajo de las consultas a través de métodos ya establecidos, estos nos permitirán realizar las tareas más comunes y que más se repiten en una base de datos como insertar, recuperar registros por su id, modificar esos registros, listarlos, eliminarlos, etc.

Vistas

Es el producto final de una petición, el código html que se le devuelve al cliente, aquí no debería haber ninguna lógica, sin embargo puede contener impresiones de variables, condicionales o bucles; pero no más que eso. La vista tiene un fin y es ése, entregar el código html de respuesta.

Controladores

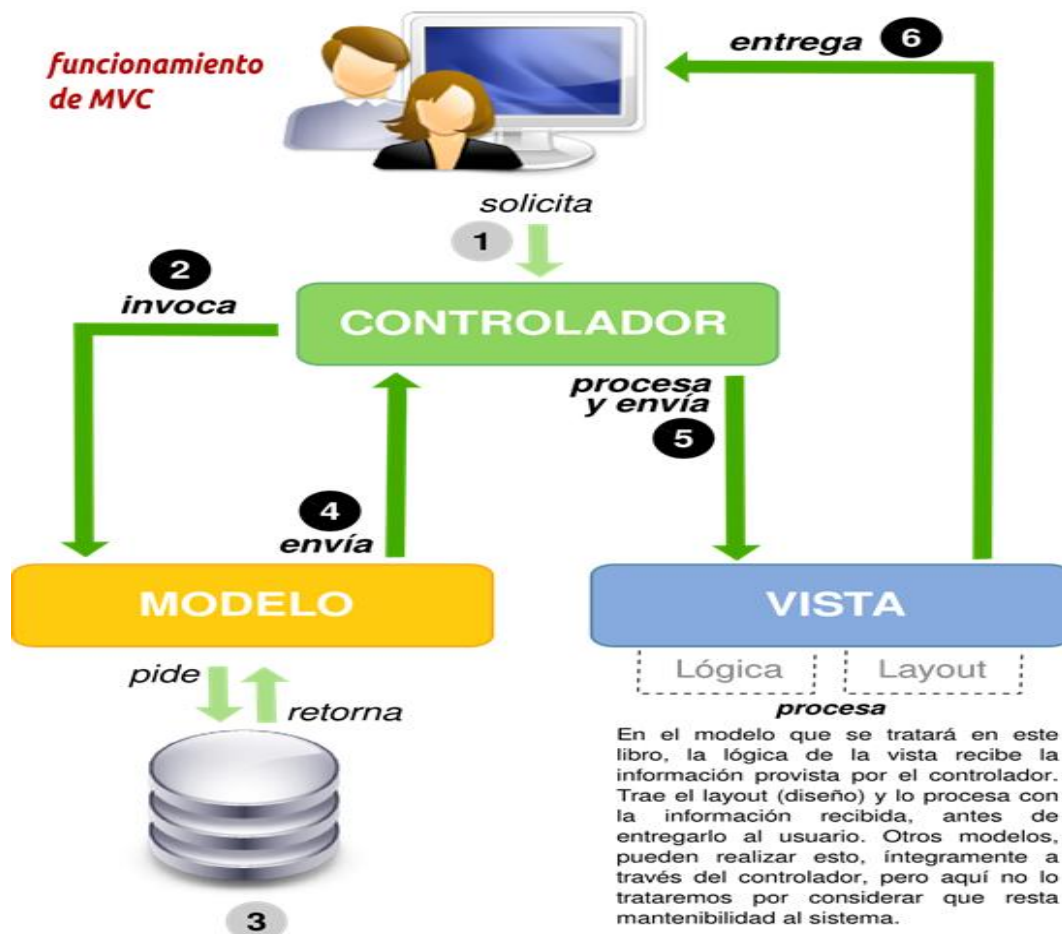
Los controladores son clases con métodos, también llamados acciones, estas acciones se comunicarán con los modelos para hacer consultas a la base de datos, y con las vistas para devolver una respuesta al cliente.

Componentes

Los componentes son una de las características más poderosas de vue. Permiten extender elementos HTML básicos para encapsular código reutilizable. En un nivel alto, los componentes son elementos personalizados a los que el compilador de vue les añade comportamiento. En algunos casos pueden aparecer como elementos HTML nativos extendidos con el atributo especial.

Estos componentes son utilizados en las vistas para la generación dinámica de datos en los diferentes módulos.

Figura 3. 19: Arquitectura MVC



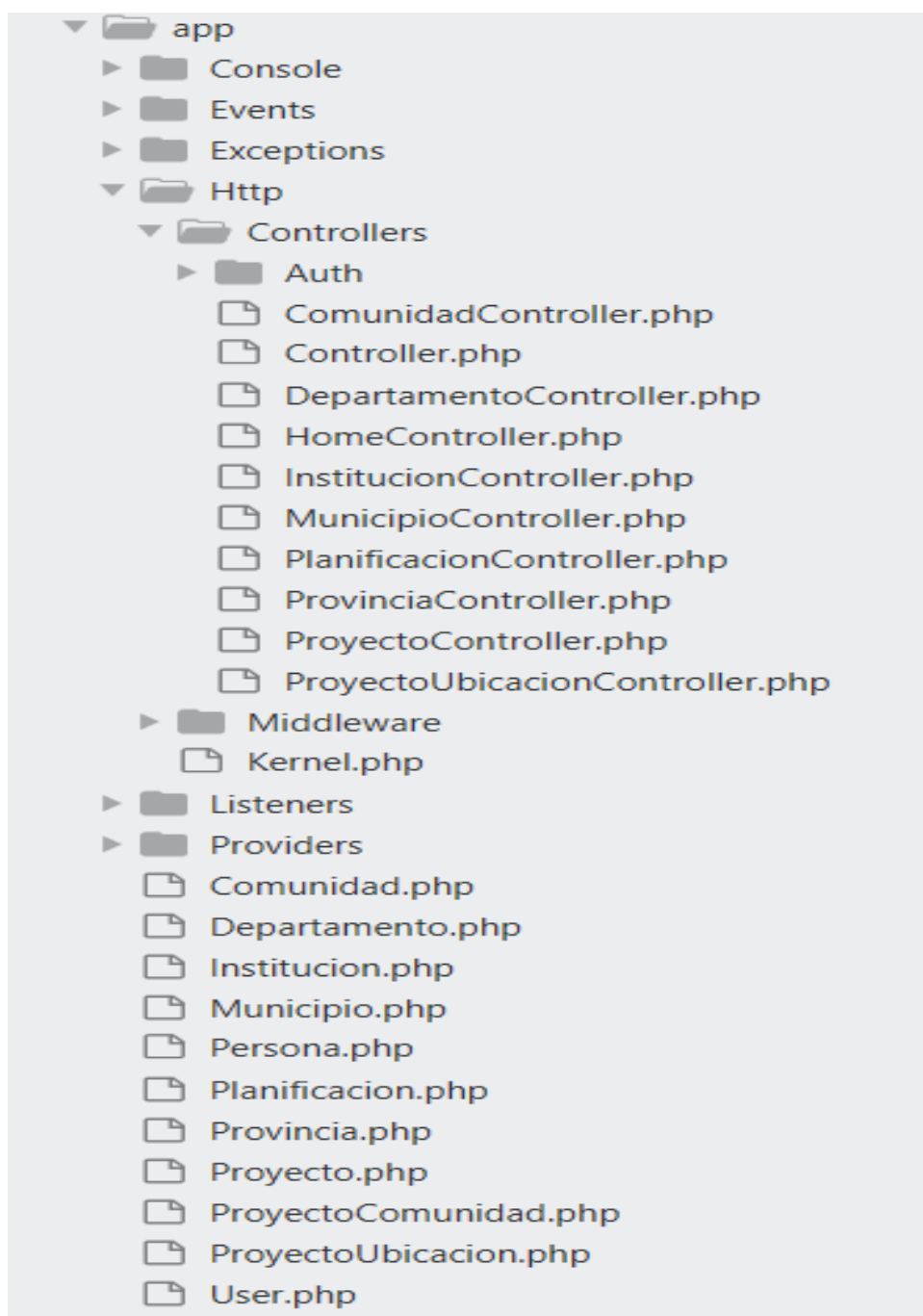
Fuente: (Elaboración propia)

3.4.2 FICHEROS

Los ficheros son archivos que están distribuidos bajo la estructura MVC de una aplicación web jerárquica, a continuación se visualiza los archivos del proyecto en los gráficos 3.20 y 3.21.

Archivos de modelo controlador (MC).

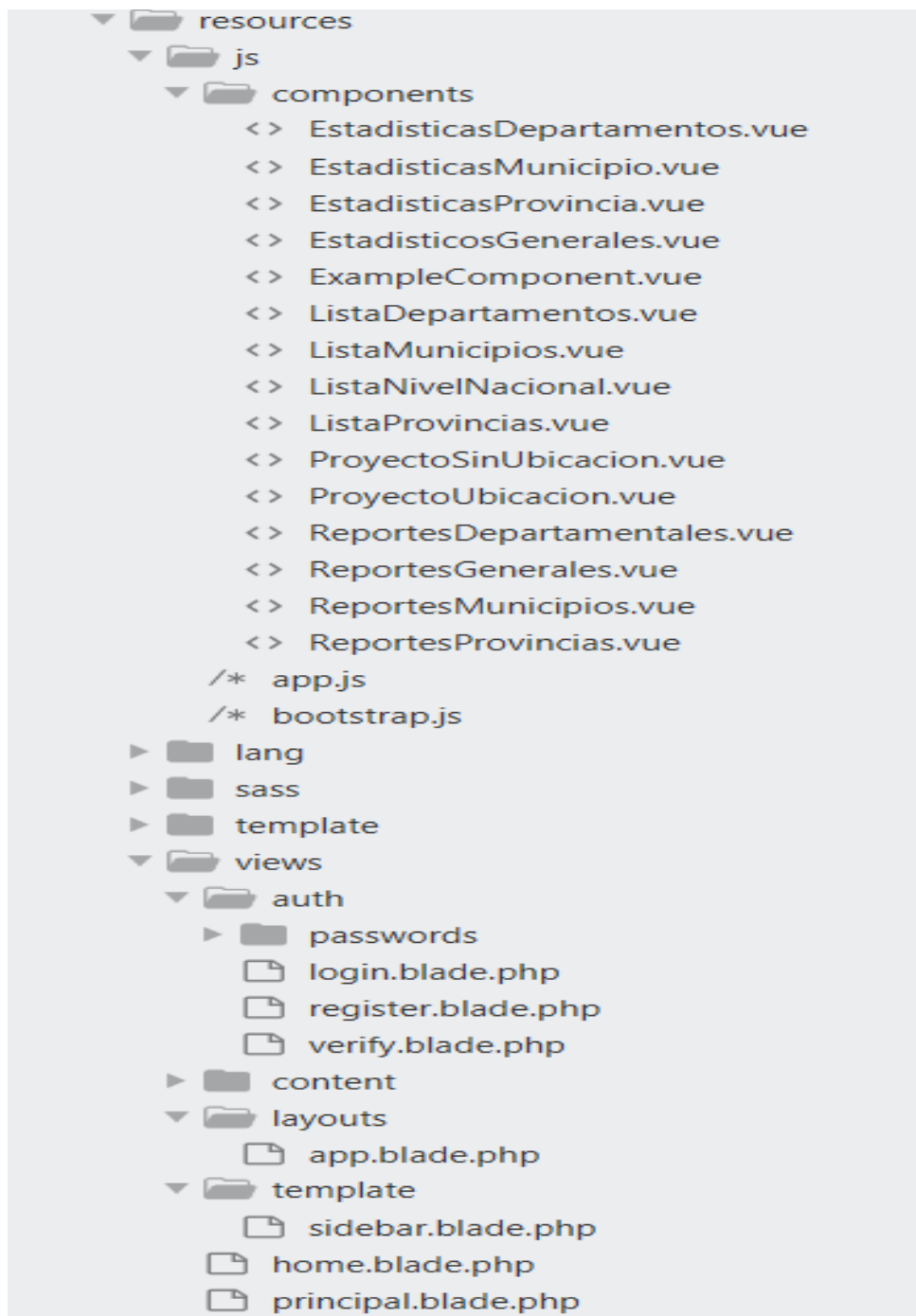
Figura 3. 20: Ficheros de Modelo y controlador



Fuente: (Elaboración propia)

Archivos de la vista (V), donde se incluyen los componentes generados para cada módulo con el framework de vue.

Figura 3. 21: Archivos de vista



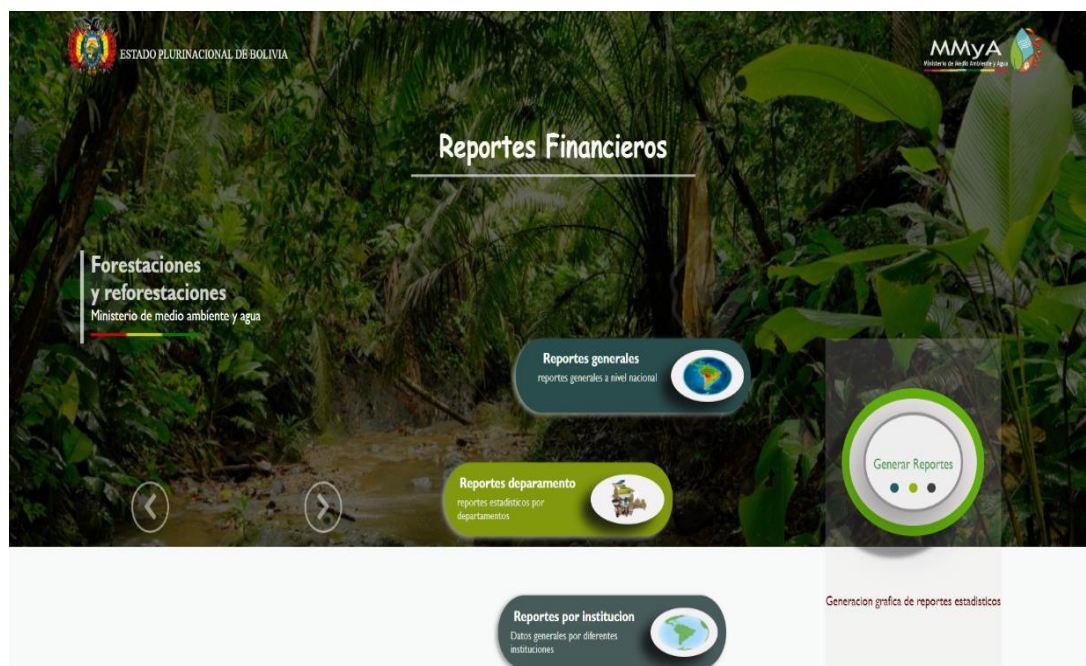
Fuente: (Elaboración propia)

3.4.3 PANTALLAS DE SISTEMA

Se muestran los interfaces de la plantilla principal del módulo financiero para el sistema de forestación y reforestación, la pantalla principal y el módulo de autenticación.

- **Pantalla principal:** Muestra el interfaz de la página web y los reportes estadísticos generales que son limitados para los usuarios visitantes que no tienen acceso al sistema.

Figura 3. 22: Pantalla principal del módulo



Fuente: (Elaboración propia)

Desarrollo del código fuente de reportes generales, función que generara información en gráfico estadístico en forma de tortas con los campos monto ejecutor, monto contraparte y cantidad de proyectos.

```

function generateCharts() {
  var respuesta='';
  var datos=[];
  var cod = document.getElementById("añosgen").value;
  var url="/principalgeneral?cod="+cod;
  axios.get(url).then(response => {
    respuesta=response.data;
    respuesta.reportegeneral.map(function(item) {
      datos.push({"country": 'Proyecto', "total": item.cantidad_plantado});
      datos.push({"country": 'Ejecutor', "total": item.monto_ejecutor});
      datos.push({"country": 'Contraparte', "total": item.monto_contraparte});
    });

    am4core.useTheme(am4themes_animated);

    var chart = am4core.create("chartgen", am4charts.PieChart3D);
    chart.hiddenState.properties.opacity = 0;
    chart.fontSize = 10;
    chart.legend = new am4charts.Legend();
    chart.legend.labels.template.maxWidth =5;
    chart.logo.disabled = true;
    chart.data =datos;

    var series = chart.series.push(new am4charts.PieSeries3D());
    series.dataFields.value = "total";
    series.dataFields.category = "country";
    series.stroke = am4core.color("#49535C");
    series.tooltip.getFillFromObject = false;
    series.tooltip.background.fill = am4core.color("#fff");
    series.tooltip.label.fill = am4core.color("#80F054");
    series.colors.list = [
      am4core.color("#D6725F"),
      am4core.color("#D9C279"),
      am4core.color("#82BF6F"),
    ];
  })
  .catch(error => {
    console.log(error);
  });
}

```

Desarrollo del código fuente para los reportes por departamentos

```

function departamentoEsta() {
  var respuesta='';
  var datos=[];
  var año = document.getElementById("añosdep").value;
  var dept = document.getElementById("dept").value;
  var url="/principaldepartamental?año="+año+"&dept="+dept;
  axios.get(url).then(response => {
    var respuesta=response.data;
    respuesta.departamento.map(function(item) {
      datos.push({"country": 'Proyecto', "total": item.cantidad_plantado});
      datos.push({"country": 'Ejecutor', "total": item.monto_ejecutor});
      datos.push({"country": 'Contraparte', "total": item.monto_contraparte});
    });

    am4core.useTheme(am4themes_animated);
    var chart = am4core.create("chartdept", am4charts.PieChart);
    chart.fontSize = 8;
    chart.data = datos;
    chart.logo.disabled = true;
    chart.legend = new am4charts.Legend();
    var pieSeries = chart.series.push(new am4charts.PieSeries());
    pieSeries.dataFields.value = "total";
    pieSeries.dataFields.category = "country";
    pieSeries.stroke = am4core.color("#49535C");
    pieSeries.tooltip.getFillFromObject = false;
    pieSeries.tooltip.background.fill = am4core.color("#fff");
    pieSeries.tooltip.label.fill = am4core.color("#80F054");
    pieSeries.colors.list = [
      am4core.color("#71892F"),
      am4core.color("#8F685B"),
      am4core.color("#4F756C"),
    ];
    pieSeries.hiddenState.properties.opacity = 1;
    pieSeries.hiddenState.properties.endAngle = -90;
    pieSeries.hiddenState.properties.startAngle = -90;

    chart.hiddenState.properties.radius = am4core.percent(0);
  })
  .catch(error => {
    console.log(error);
  });
}

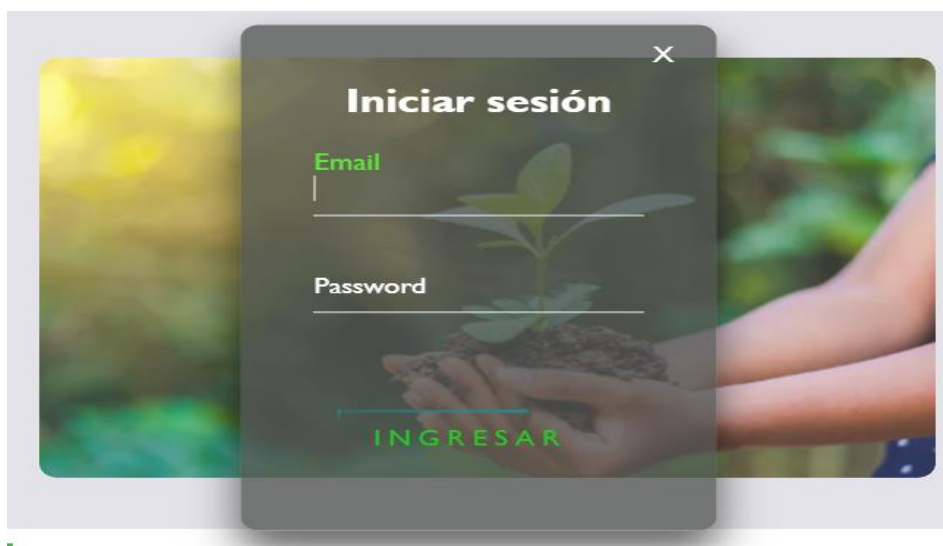
```

Desarrollo del código fuente para generar datos por instituciones.

```
function generateInstCharts(){
  var datos=[];
  var respuesta='';
  var añoinst = document.getElementById("instYear").value;
  var nameinst = document.getElementById("instName").value;
  let url = '/principalinstituciones?añoinst='+añoinst+"&nameinst="+nameinst;
  axios.get(url).then(response => {
    respuesta=response.data;
    respuesta.institucion.map(function(item) {
      datos.push({"country": 'Proyecto', "total": item.candidad});
      datos.push({"country": 'Ejecutor', "total": item.ejecutor});
      datos.push({"country": 'Contraparte', "total": item.contraparte});
    });
    am4core.useTheme(am4themes_animated);
    var chart = am4core.create("chartinst", am4charts.PieChart3D);
    chart.hiddenState.properties.opacity = 0;
    chart.fontSize = 10;
    chart.legend = new am4charts.Legend();
    chart.legend.labels.template.maxWidth =5;
    chart.logo.disabled = true;
    chart.data =datos;
    var series = chart.series.push(new am4charts.PieSeries3D());
    series.dataFields.value = "total";
    series.dataFields.category = "country";
    series.stroke = am4core.color("#49535C");
    series.tooltip.getFillFromObject = false;
    series.tooltip.background.fill = am4core.color("#fff");
    series.tooltip.label.fill = am4core.color("#80F054");
    series.colors.list = [
      am4core.color("#526B41"),
      am4core.color("#C26BA0"),
      am4core.color("#DA856C"),
    ];
  })
  .catch(error => {
    console.log(error);
  });
}
```

- **Autenticación:** La autenticación de usuario es para los usuarios que cuentan con credenciales de autenticación y los privilegios correspondientes registrados en la base de datos del sistema para su posterior acceso por un módulo de autenticación.

Figura 3 23: Menú de autenticación del usuario



Fuente: (Elaboración propia)

Código fuente de la funciones de autenticación de usuario:

```

public function showLoginForm()
{
    return view('/layouts/app');
}

public function login(Request $request)
{
    $credentials = $request->only('email', 'password');

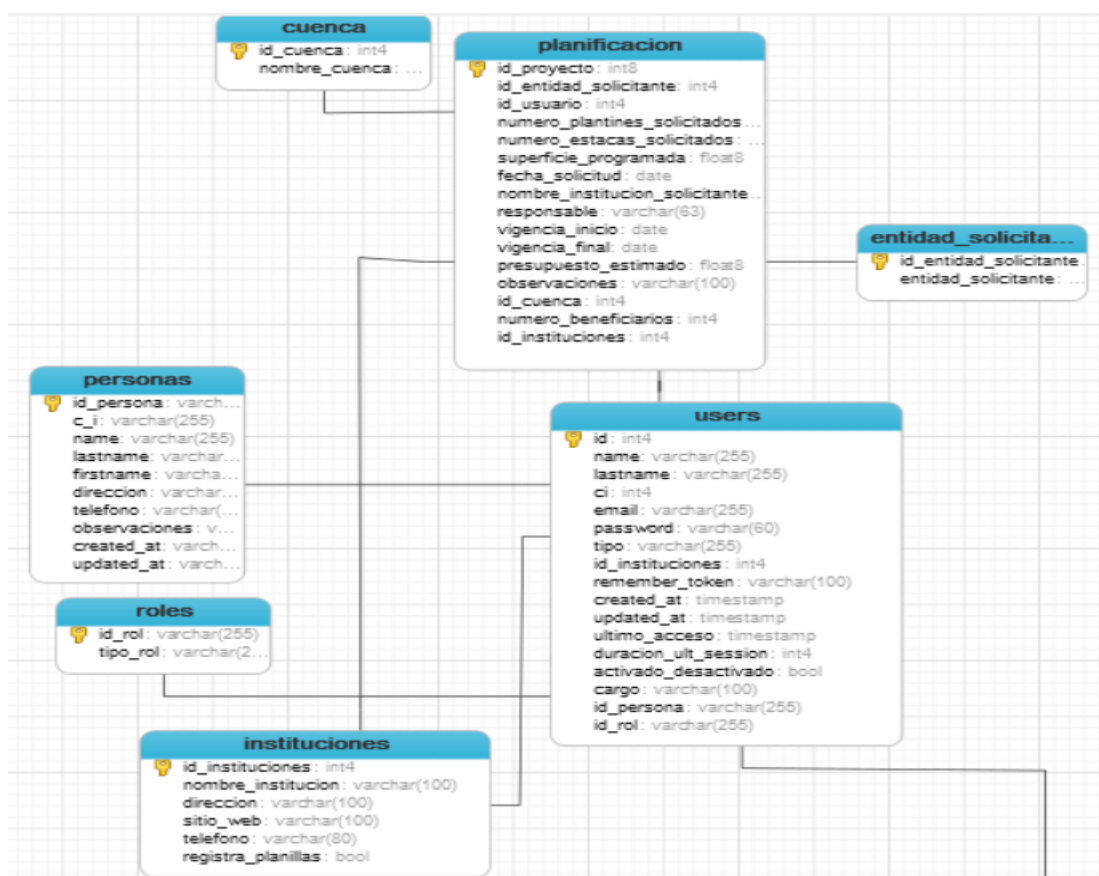
    if (Auth::attempt($credentials)) {
        return redirect('main');
    } else
        return view('Auth\login');
}

```

3.4.4 MODELO DE BASE DE DATOS

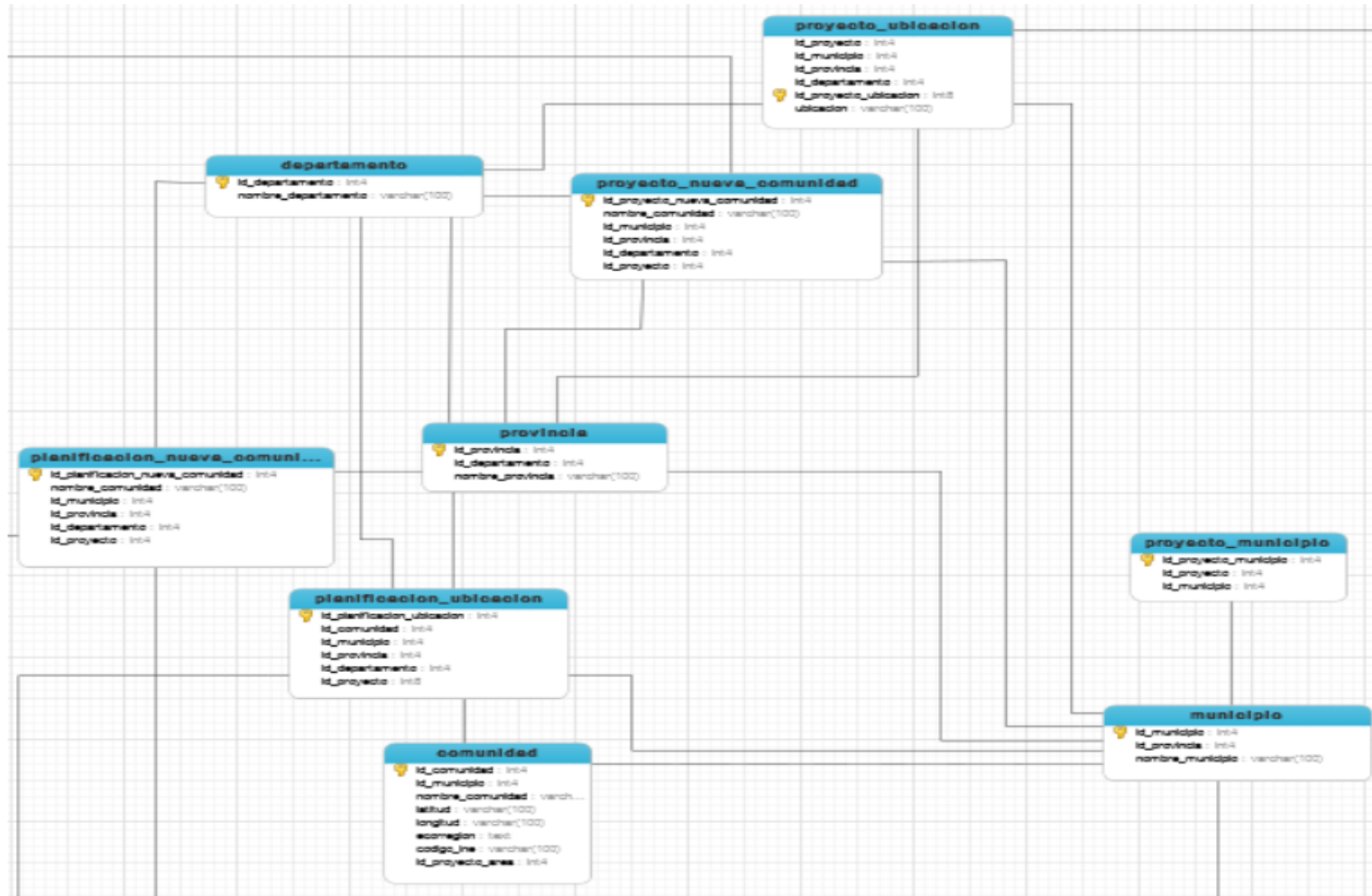
En los siguientes gráficos se visualiza el modelado de la base de datos del módulo de reportes financiero para el sistema de forestaciones y reforestaciones caso, proyecto/unjp/bol/045/unj.

Figura 3 24. Modelo de base de datos parte 1



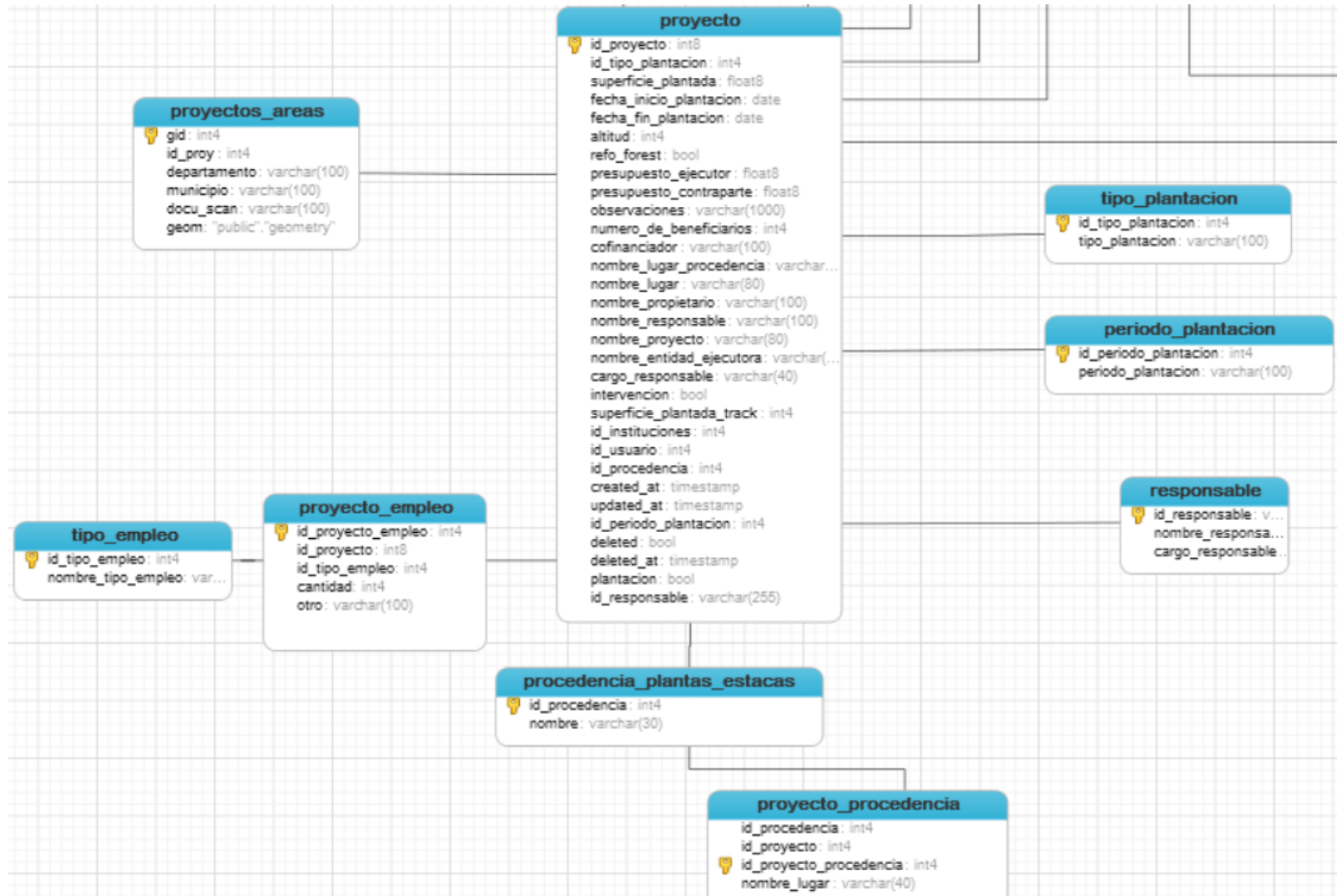
Fuente: (Elaboración propia)

Figura 3. 25: Modelo de base de datos parte 2



Fuente: (Elaboración propia)

Figura 3. 26: Modelo de base de datos parte 3



Fuente: (Elaboración propia)

3.4.5 DICCIONARIO DE DATOS

Un diccionario de datos es un conjunto de definiciones que contiene las características lógicas y puntuales de los datos que se utilizaron en la base de datos, se detalla a continuación las diferentes tablas.

Users

Tabla 3. 8: Tabla de usuarios

Columna	Nulo	Tipo de dato	Longitud
ci	NO	integer	32
email	NO	character varying	255
password	NO	character varying	60
duracion_ult_session	YES	integer	32
activado_desactivado	YES	boolean	
cargo	YES	character varying	100
name	NO	character varying	255
ultimo_acceso	YES	timestamp without time zone	
created_at	NO	timestamp without time zone	
remember_token	YES	character varying	100
id_instituciones	NO	integer	32
tipo	NO	character varying	255
id	NO	integer	32
id_persona	YES	integer	32
id_rol	YES	integer	32
updated_at	NO	timestamp without time zone	
lastname	NO	character varying	255

Fuente: (Elaboración propia)

Tipo_plantacion

Tabla 3. 9: Tabla tipo de plantaciones

Columna	Nulo	Tipo de dato	Longitud
nombre	NO	character varying	100
id_tipo_participante	NO	integer	32
descripcion	YES	character varying	100
tipo_plantacion	NO	character varying	100
id_tipo_plantacion	NO	integer	32

Fuente: (Elaboración propia)

Tipo_empleo

Tabla 3. 10: Tipo de empleo

Columna	Nulo	Tipo de dato	Longitud
id_tipo_empleo	NO	integer	32
nombre_tipo_empleo	YES	character varying	

Fuente: (Elaboración propia)

Departamento

Tabla 3. 11: Departamentos

Columna	Nulo	Tipo de dato	Longitud
nombre_departamento	NO	character varying	100
id_departamento	NO	integer	32

Fuente: (Elaboración propia)

Roles

Tabla 3. 12: Tabla de roles para usuarios

Columna	Nulo	Tipo de dato	Longitud
id_rol	NO	integer	32
tipo_rol	NO	character varying	255

Fuente: (Elaboración propia)

Responsable

Tabla 3. 13: Responsable

Columna	Nulo	Tipo de dato	Longitud
cargo_responsable	YES	character varying	150
nombre_responsable	YES	character varying	40
id_responsable	NO	integer	32

Fuente: (Elaboración propia)

Proyecto_ubicacion

Tabla 3. 14: Proyecto ubicación

Columna	Nulo	Tipo de dato	Longitud
id_proyecto	YES	integer	32
id_municipio	YES	integer	32
id_provincia	YES	integer	32
id_departamento	YES	integer	32
id_proyecto_ubicacion	NO	bigint	64
ubicacion	YES	character varying	100

Fuente: (Elaboración propia)

Proyecto_nueva_comunidad*Tabla 3. 15: Proyecto nueva comunidad*

Columna	Nulo	Tipo de dato	Longitud
nombre_comunidad	YES	character varying	100
id_proyecto_nueva_comunidad	NO	integer	32
id_proyecto	YES	integer	32
id_departamento	YES	integer	32
id_provincia	YES	integer	32
id_municipio	YES	integer	32

*Fuente: (Elaboración propia)***Proyecto_empleo***Tabla 3. 16: Proyecto empleo*

Columna	Nulo	Tipo de dato	Longitud
id_proyecto	YES	bigint	64
id_tipo_empleo	YES	integer	32
id_proyecto_empleo	NO	integer	32
cantidad	YES	integer	32
otro	YES	character varying	100

*Fuente: (Elaboración propia)***Proyecto***Tabla 3. 17: Proyectos*

Columna	Nulo	Tipo de dato	Longitud
altitud	YES	integer	32
intervencion	YES	boolean	
cargo_responsable	YES	character varying	40
nombre_entidad_ejecutora	YES	character varying	100
nombre_proyecto	YES	character varying	80

nombre_responsable	YES	character varying	100
nombre_propietario	YES	character varying	100
nombre_lugar	YES	character varying	80
nombre_lugar_procedencia	YES	character varying	100
cofinanciador	YES	character varying	100
numero_de_beneficiarios	YES	integer	32
observaciones	YES	character varying	1000
presupuesto_contraparte	YES	double precision	53
presupuesto_ejecutor	YES	double precision	53
refo_forest	YES	boolean	
fecha_fin_plantacion	YES	date	
id_proyecto	NO	bigint	64
id_tipo_plantacion	YES	integer	32
superficie_plantada	YES	double precision	53
fecha_inicio_plantacion	YES	date	
id_responsable	YES	integer	32
plantacion	YES	boolean	
deleted_at	YES	timestamp without time zone	
deleted	YES	boolean	
id_periodo_plantacion	YES	integer	32
updated_at	YES	timestamp without time zone	
created_at	YES	timestamp without time zone	
id_procedencia	YES	integer	32
id_usuario	YES	integer	32
id_instituciones	YES	integer	32
superficie_plantada_track	YES	integer	32

Fuente: (Elaboración propia)

Procedencia_plantas_estacas*Tabla 3. 18: Procedencia plantas estacas*

Columna	Nulo	Tipo de dato	Longitud
nombre	YES	character varying	30
id_procedencia	NO	integer	32

*Fuente: (Elaboración propia)***Planificacion_ubicacion***Tabla 3. 19: Planificaciones de ubicación*

Columna	Nulo	Tipo de dato	Longitud
id_departamento	YES	integer	32
id_municipio	YES	integer	32
id_provincia	YES	integer	32
id_proyecto	YES	bigint	64
id_planificacion_ubicacion	NO	integer	32
id_comunidad	YES	integer	32

*Fuente: (elaboración propia)***Planificación***Tabla 3. 20: Planificaciones*

Columna	Nulo	Tipo de dato	Longitud
presupuesto_estimado	YES	double precision	53
id_proyecto	NO	bigint	64
id_entidad_solicitante	YES	integer	32
id_usuario	YES	integer	32
numero_plantines_solicitados	YES	integer	32
numero_estacas_solicitados	YES	integer	32
superficie_programada	YES	double precision	53
fecha_solicitud	YES	date	

nombre_institucion_solicitante	YES	character varying	100
responsable	YES	character varying	63
vigencia_inicio	YES	date	
vigencia_final	YES	date	
observaciones	YES	character varying	100
id_cuenca	YES	integer	32
numero_beneficiarios	YES	integer	32
id_instituciones	YES	integer	32

Fuente: (elaboración propia)

Personas

Tabla 3. 21: Personas

Columna	Nulo	Tipo de dato	Longitud
direccion	YES	character varying	150
telefono	YES	integer	32
observaciones	YES	character varying	150
created_at	YES	timestamp without time zone	
updated_at	YES	timestamp without time zone	
lastname	NO	character varying	100
name	NO	character varying	100
c_i	NO	character varying	9
id_persona	NO	integer	32
firstname	YES	character varying	100

Fuente: (elaboración propia)

Periodo_plantacion

Tabla 3. 22: Periodo de plantaciones

Columna	Nulo	Tipo de dato	Longitud
id_periodo_plantacion	NO	integer	32
periodo_plantacion	YES	character varying	100

Fuente: (elaboración propia)

Municipio

Tabla 3. 23: Municipios

Columna	Nulo	Tipo de dato	Longitud
id_provincia	YES	integer	32
nombre_municipio	NO	character varying	100
id_municipio	NO	integer	32

Fuente: (elaboración propia)

Instituciones

Tabla 3. 24: Instituciones

Columna	Nulo	Tipo de dato	Longitud
direccion	YES	character varying	100
id_instituciones	NO	integer	32
nombre_institucion	NO	character varying	100
sitio_web	YES	character varying	100
telefono	YES	character varying	80
registra_planillas	YES	boolean	
encargado	YES	character varying	100

Fuente: (elaboración propia)

Entidad_solicitante*Tabla 3. 25: Entidad solicitante*

Columna	Nulo	Tipo de dato	Longitud
entidad_solicitante	NO	character varying	100
id_entidad_solicitante	NO	integer	32

*Fuente: (elaboración propia)***Cuenca***Tabla 3. 26: Cuencas*

Columna	Nulo	Tipo de dato	Longitud
id_cuenca	NO	integer	32
nombre_cuenca	YES	character varying	60

*Fuente: (elaboración propia)***Comunidad***Tabla 3. 27: Comunidades*

Columna	Nulo	Tipo de dato	Longitud
longitud	YES	character varying	100
codigo_ine	YES	character varying	100
latitud	YES	character varying	100
ecorregion	YES	text	
id_comunidad	NO	integer	32
id_municipio	YES	integer	32
nombre_comunidad	NO	character varying	100

Fuente: (elaboración propia)

Provincia*Tabla 3. 28: Provincias*

Columna	Nulo	Tipo de dato	Longitud
id_departamento	YES	integer	32
id_provincia	NO	integer	32
nombre_provincia	NO	character varying	100

Fuente: (elaboración propia)



CONTROL DE CALIDAD IV

4.1 MÉTRICAS DE CALIDAD DE SOFTWARE

La calidad del software es el desarrollo de software basado en estándares con la funcionalidad y el rendimiento total que satisfaga los requerimientos del cliente, todo desarrollador pone el máximo esfuerzo y dedicación, pero no siempre llega a la perfección en el producto terminado.

4.1.1 NORMA ISO 9126

La norma ISO 9126 es estándar internacional para evaluar la calidad del software en base a un conjunto de caracteres que plantea.

Las características de la norma ISO 9126 son las siguientes:

- **Funcionalidad**
- **Confiabilidad**
- **Usabilidad**
- **Portabilidad**
- **Mantenibilidad**
- **Eficiencia**

4.1.2 FUNCIONALIDAD

En este grupo se conjunta una serie de atributos que permiten calificar si un producto de software maneja en forma adecuada, el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado.

Para el cálculo de la funcionalidad se debe determinar cinco características de dominios de información los valores de información se definen de la siguiente manera.

- **Número de entradas de usuario.** Se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.
- **Número de salidas de usuario.** Se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc.

- **Número de peticiones de usuario.** Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado.
- **Número de archivos.** Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).
- **Número de interfaces externas.** Se cuentan todas las interfaces legibles por la máquina (por ejemplo: archivos de datos de cinta o disco) que se utilizan para transmitir información a otro sistema.

Tabla 4. 1: Factores de ponderación

Parámetros de medición	Cuenta	Factores de ponderación			Valor obtenido
		Simple	Medio	Complejo	
Nro. de entradas de usuario	5	-	4	-	20
Nro. de salidas de usuario	8	-	5	-	40
Nro. de peticiones de usuario	2	-	4	-	8
Nro. de archivos	18	-	10	-	72
Nro. de interfaces externas	0	-	7	-	0
Cuenta total					140

Fuente: (elaboración propia)

En la siguiente tabla se muestra el factor de ajuste de complejidad basado en las respuestas de las siguientes preguntas evaluadas.

Tabla 4. 2: Ajuste de complejidad

Factores de complejidad	Sin influencia	Inicial	Moderado	Medio	Significativo	Esencial	Fi
	0	1	2	3	4	5	
¿Requiere el sistema copias de seguridad y de recuperación fiables?				X			3
¿Se requiere comunicación de datos?					X		4
¿Existen funciones de procesamiento distribuido?						X	5
¿Es crítico el rendimiento?				X			3
¿Se ejecutará el sistema con un entorno operativo existente y fuertemente utilizado?					X		4
¿Requiere el sistema entrada de datos interactiva?				X			3
Facilidad Operativa					X		4
¿Se actualiza los archivos maestros de forma interactiva?						X	5
¿Son complejas las entradas, las salidas, los archivos o las peticiones?				X			3
Procesamiento interno complejo				X			3
Diseño de código reutilizable					X		4
Facilidad de Instalación				X			3
Facilidad de cambios					X		4
¿Soporta múltiples instalaciones en diferentes sitios?					X		4
Factor ajuste de complejidad							52

Fuente: (elaboración propia)

La funcionalidad es la medida a través del punto función (PF), que proporciona una medida objetiva del tamaño de la aplicación web que está basada en la visión del usuario de la aplicación web.

Para calcular el punto función se utiliza la siguiente formula:

$$PF = \text{Cuenta Total} * (X + \text{Min} (Y) * \sum Fi)$$

Dónde:

PF: Medida de funcionalidad

Cuenta Total: Es la suma de los siguientes datos (Nro. de entradas, Nro. de salidas, Nro. de peticiones, Nro. de archivos, Nro. de interfaces externas).

X: Confiabilidad del proyecto, varía entre 1 a 100%.

Min (Y): Error mínimo aceptable al de la complejidad.

$\sum Fi$: Son los valores de ajuste de complejidad, donde $(1 \leq i \leq 14)$.

Para calcular el PF se usa la siguiente ecuación:

$$PF = \text{cuenta total} * (X + \text{Min} (Y) * \sum Fi)$$

$$PF = \text{Cuenta total} * [0.65 + (0.01 * \sum Fi)]$$

Reemplazando los valores obtenidos en las tablas 4.1 y 4.2 se obtiene el siguiente resultado:

$$PF = 140 * [0.65 + (0.01 * 52)]$$

$$PF = 163.8$$

A continuación, calculamos el PF ideal:

$$PF_{\text{ideal}} = 140 * [0.65 + (0.01 * 62)]$$

$$PF_{\text{ideal}} = 177.8$$

Entonces la funcionalidad del sistema es:

$$\text{Funcionalidad} = (PF / PF_{\text{ideal}}) * 100$$

$$\text{Funcionalidad} = (163.8/177.8) * 100$$

$$\text{Funcionalidad} = 92.13\%$$

Con el resultado obtenido de los cálculos realizados se puede interpretar de qué 92.13% esta funcionalidad esto determinan que el sistema responde de manera óptima a las funcionalidades requeridas por la institución.

4.1.3 FIABILIDAD

Es la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido.

Para determinar la confiabilidad se tomó en cuenta las fallas que puedan ocurrir en un tiempo determinado después su implantación, para medir el tiempo promedio de fallos se calcula bajo la siguiente formula.

$$\text{TMEF} = \text{TMDF} + \text{TMDR}$$

Donde:

TMDF: Tiempo medio de fallo.

TMDR: Tiempo medio de reparación.

Se estima que un fallo puede ocurrir cada 30 días hábiles y su reparación del fallo en promedio pueda tomar 2 horas, entonces:

$$\text{TMEF} = (30 * 8) + 2$$

$$\text{TMEF} = 242 \text{ horas}$$

Entonces la disponibilidad es un buen indicador de fiabilidad, en base de la siguiente formula se tiene que:

$$\text{Disponibilidad} = (\text{TMDF} / \text{TMEF}) * 100$$

$$\text{Disponibilidad} = (240 / 242) * 100 = 98.59\%$$

Con lo que se llega a la conclusión de que el sistema tiene un 98.59% de confiabilidad

4.1.4 USABILIDAD

La usabilidad consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema, dado también que se puede referir a la usabilidad como la capacidad del software de ser entendido, aprendido, y usado de forma fácil y atractiva.

Para determinar el porcentaje de la usabilidad del sistema se optó por realizar una encuesta a 5 personas. La siguiente tabla nos muestra los resultados de la encuesta que se realizó

Tabla 4. 3: Encuesta sobre la usabilidad del sistema

N°	Presuntas	Respuesta		% de si
		Si	No	
1	¿No es complicado usar el sistema?	4	1	80
2	¿Las pantallas que vista fueron de su agrado	3	2	60
3	¿Los datos obtenidos que vio fueron fáciles de comprender?	4	1	80
4	¿El sistema responde rápido a sus solicitudes?	4	1	80
5	¿El sistema facilita el trabajo para generar datos?	5	0	100
6	¿El sistema reduce el tiempo de trabajo en generar informes?	5	0	100
7	¿Es fácil navegar por las distintas opciones?	5	0	100
8	¿La generación de datos que se realizan no es complicada?	4	1	80
9	¿El sistema no presento errores?	4	1	80
Promedio				84.4%

Fuente: (Elaboración propia)

4.1.5 MANTENIBILIDAD

Es el esfuerzo necesario para diagnosticar las deficiencias o causas de fallas o para identificar las partes que deberán ser modificadas, la calidad que tiene el software para ser modificado incluyendo correcciones o mejoras del software a cambios en el entorno y especificaciones de requerimientos funcionales, para poder medir la calidad del mantenimiento del sistema utilizaremos el índice de madurez del software (IMS), que indica la estabilidad de un producto de software.

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

$$IMS = [Mt - (Fa + Fb + Fc)]/t$$

Donde:

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.

Recopilando la información requerida por la fórmula se obtuvo la información que se muestra en la siguiente tabla:

Tabla 4. 4: Información requerida por el IMS

Información	Valor
Mt	11
Fa	0
Fb	0
Fc	0

Fuente: (Elaboración propia)

Ahora calculamos el IMS, usando los valores obtenidos:

$$IMS = [11 - (0 + 0 + 0)]/11$$

$$IMS = 11/11 = 100\%$$

Con la interpretación a este resultado establece un 100% de madures, lo que indica que no requiere de mantenimientos inmediatamente.

4.1.6 PORTABILIDAD

Se evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones, es decir que el software se puede trasladar 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.

$$GP = 1 - (ET/ER)$$

Donde:

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.

Entonces la factibilidad estimada para trasportar el sistema es de 1/2 días a otro entorno y la implementación del mismo en otro entorno es de 5 días.

$$GP = 1 - ((1/2)/ 5) = 1 - 0.1 = 90\%$$

Por lo que se concluye que el sistema tiene un grado de portabilidad del 90%.

4.1.7 CALIDAD GLOBAL

La calidad global del sistema es el total, de todos los cálculos realizados según lo propuesto por el estándar de calidad ISO 9126, los cálculos realizados se visualizan a continuación.

Tabla 4. 5: Calidad global del sistema

Atributos	Valor (%)
Funcionalidad	92.13
Fiabilidad	98.59
Usabilidad	84.4
Mantenibilidad	100
Portabilidad	90
Calidad global	92.99

Fuente: (Elaboración propia)

Con el resultado obtenido se llega a la conclusión de que el 92.99% del sistema es de calidad.

4.2 EVALUACIÓN DE COSTOS

La evaluación del costo determina la cantidad de los recursos necesarios en términos de dinero, esfuerzo, capacidad, conocimientos y tiempo. Se evaluará los costos y beneficios para el sistema en base al modelo de costos como II.

4.2.1 MODELO COCOMO II

El Modelo como II, es un modelo de estimación que se encuentra en la jerarquía de modelos de estimación de software, permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y escala, COCOMO II consta con tres modelos de estimación, los mismos se representan en 3 ecuaciones:

$$E = a(KLDC)^b ; \text{Personas} \cdot \text{mes}$$

$$D = c(E)^d ; \text{Meses}$$

$$P = \frac{E}{D} ; \text{Personas}$$

Dónde:

E: Esfuerzo requerido por el proyecto expresado en persona-mes.

D: Tiempo requerido por el proyecto expresado en meses.

P: Número de personas requeridas para el proyecto.

A, B, C y D: Constantes con valores definidos según cada sub-modelo.

KLDC: Cantidad de líneas de código distribuidas en miles

A la vez cada modelo se subdivide en tres modos:

Modo orgánico: Es un pequeño grupo de programadores experimentados desarrollando proyectos de software en un entorno familiar.

Modo semilibre: Corresponde a un esquema intermedio entre el modo orgánico y el rígido, el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.

Modo rígido: El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas.

A continuación, se describe las constantes de acuerdo a los modos mencionados anteriormente.

Tabla 4. 6: Constantes a,b,c,d COCOMO II

Modo	A	B	C	D
Organo	2.4	1.05	2.5	0.38
Semilibre	3.0	1.12	2.5	0.35
Rígido	3.6	2.20	2.5	0.32

Fuente: (Elaboración propia)

4.2.2 COSTO DEL SISTEMA

El costo del sistema se lo plantea en dos partes:

4.2.2.1 COSTO DE DESARROLLO

Para el cálculo del desarrollo del software tiene como partida el punto función calculada anteriormente en funcionalidad de software, cuyo valor obtenido es:

$$PF = 163.8$$

La tabla 3.35: muestra la relación para convertir el valor de PF a KLDC (Kilos de líneas de código).

Tabla 4. 7: Muestra de relación para convertir el valor de PF a KLDC.

Lector	Nivel	Factor LDC / PF
C	2.5	128
Java	6	53
Ansi Cobol	3	107
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34
SQL	12	12

Fuente: (elaboración propia)

Entonces, realizando los cálculos y escogiendo el valor del lenguaje de programación PHP de la tabla 5.35, tenemos:

$$\text{LCD} = \text{PF} * \text{Factor LCD/PF}$$

$$\text{LCD} = 163.8 * 29$$

$$\text{LCD} = 4750.2$$

$$\text{KLCD} = 4750.2 / 1000$$

$$\text{KLCD} = 4.75$$

Ahora, para hallar el esfuerzo reemplazamos los valores obtenidos hasta ahora:

$$E = 2.4 * (4.75)^{1.05} = 12.32$$

$$D = 2.4 * (4.75)^{0.38} = 4.34$$

Para el cálculo del número de programadores para el desarrollo de software:

$$E = \frac{12.32}{4.34} = 3.83 \approx 2$$

Estimando que el salario medio de un programador es de Bs. 4000, esta cifra será tomada en cuenta para la siguiente estimación:

Costo del software desarrollado = Numero de programadores * Salario de un programador.

Costo del Software Desarrollado por Persona = 2 * 4000 = 8000 Bs.

Costo total del Software Desarrollado = 8000 * 4= 32000 Bs.

Lo que significa que el costo del sistema desarrollado por los 4 meses es 32000, equivalente en dólares a 4630.56\$.

4.2.2 COSTO ELABORACIÓN DEL PROYECTO

Se calcula los costos de inversión de los recursos que se usaron para la elaboración del sistema.

Tabla 4. 8: Costos de recursos empleados para la elaboración del sistema.

Recursos	Costo Bs
Materia de escritorio	300
Investigación del proyecto	250
Internet	500
Otros	280
Total	1330

Fuente: (elaboración propia)

Por tanto, el costo de la elaboración del proyecto es de Bs. 1330

4.2.3 COSTO TOTAL DEL SISTEMA

El costo total del sistema se obtiene de la sumatoria del costo de desarrollo y el costo de elaboración del proyecto, se puede observar los resultados en la siguiente tabla, todos los costos están expresados en moneda nacional bs.

Tabla 4. 9: Costos total del sistema

Detalle	Costo Bs
Costo de desarrollo	32000
Costo de elaboración del proyecto	1330
Costo Total	33330

Fuente: (elaboración propia)

Entonces, el costo total del sistema es de Bs. 33330 equivalente en dólares 4832.06\$



CAPITULO V

PRUEBAS Y RESULTADOS

5.1 PRUEBAS

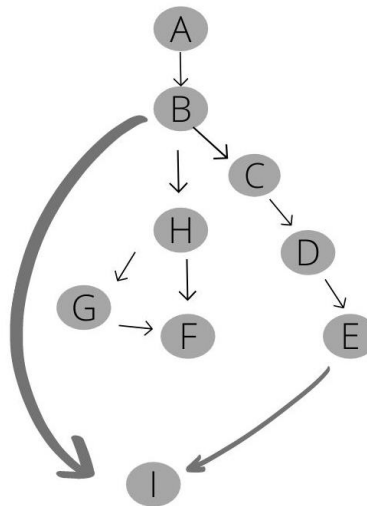
- **Pruebas de caja blanca.** Este tipo de prueba se realiza, todo el código con la noción de probar el desenvolvimiento del sistema recorrido por cada uno de los casos presentados por los algoritmos que se utilizaron en la codificación, es decir son casos de prueba que se aplica al código fuente. En las pruebas de caja blanca se consideran lo siguiente:

Se realizarán las pruebas utilizando el conocimiento del funcionamiento interno del código.

Las pruebas de caja blanca solo pueden ser realizadas por programadores.

Se visualiza en la siguiente figura las pruebas de caja blanca del sistema por nodos realizando el recorrido genera el siguiente grafo.

Figura 5. 1: Pruebas de Caja negra

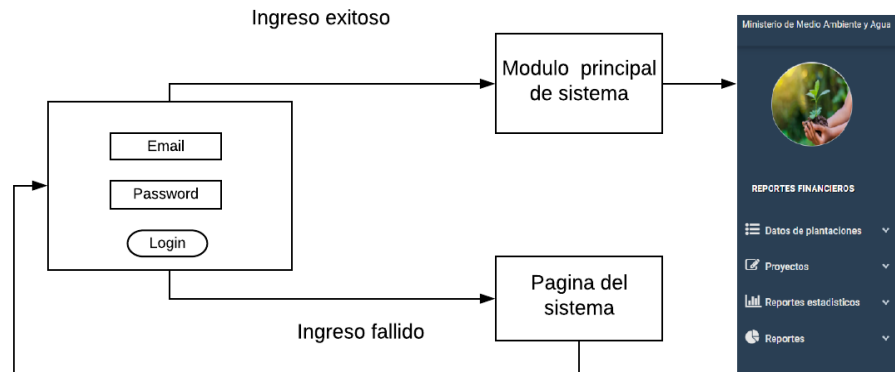


Fuente: (elaboración propia)

- **Pruebas de caja negra.** La prueba de caja negra consiste en probar cada una de las funciones del sistema que fueron descritas anteriormente. Con este tipo de prueba se debe buscar que las funciones sean operativas, además se debe agotar al sistema de tal manera buscar la mayor cantidad de errores estas pruebas son pruebas sobre la interfaz del software. (Peña, 2016, págs. 1-30)

A continuación se muestra la prueba de autenticación del usuario, si el usuario accede con las credenciales correctas del sistema ingresara a los módulos internos en caso será denegado el acceso.

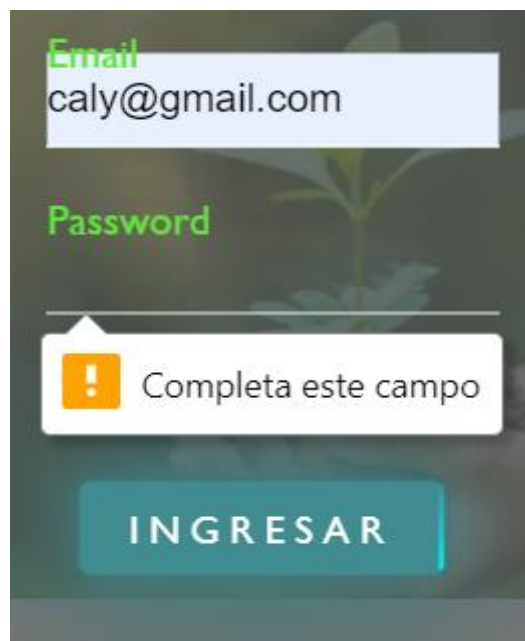
Figura 5. 2: Prueba de autenticación de usuario.



Fuente: (Elaboración propia)

Pruebas de validación del formulario para el ingreso de datos, si un formulario no es llenado correctamente muestra una alerta y en caso contrario el ingreso con los datos será exitoso.

Figura 5. 3: Pruebas de validación de formulario



Fuente: (Elaboración propia)

5.1.1 SEGURIDAD DEL SISTEMA

En la seguridad esta implementada en base al framework utilizado para el desarrollo del software, laravel tiene las siguientes características.

- HTTP Middleware: Middleware se encarga de analizar y filtrar las llamadas HTTP en tu servidor. Puedes instalarlo para que se encargue de verificar que se trate de un usuario registrado, de evitar problemas de tipo Cross-SiteScripting (XSS) y otras medidas de seguridad.
- Autenticación: Laravel viene listo para implementar autenticación de usuarios de forma nativa e incluye la opción de recordar al usuario. Además, permite incluir parámetros adicionales lo que nos asegurará, por ejemplo, si se trata de un usuario activo.
- Encriptación: Una aplicación segura necesita ser capaz de encriptar sus datos. Con laravel se tiene todo lo necesario para empezar a usar seguridad OpenSSL y cifrado AES-256-CBC. Adicionalmente, todos los valores encriptados están firmados por un código de autenticación de mensaje que detecta si el mensaje encriptado fue alterado.
- Adicionalmente, todos los valores encriptados están firmados por un código de autenticación de mensaje "Token" que detecta si el mensaje encriptado fue alterado.
- Para finalizar, cuando un desarrollador crea un sistema que tenga acceso a datos, siempre debe estar atento a los tipos de ataques como los ataques de inyección SQL, Laravel preocupado por esto, incorpora un ORM para que el desarrollador deje de preocuparse por este tipo de ataques ya que el ORM está basado en una capa de objetos y así no ser capaz de interpretar el lenguaje SQL.

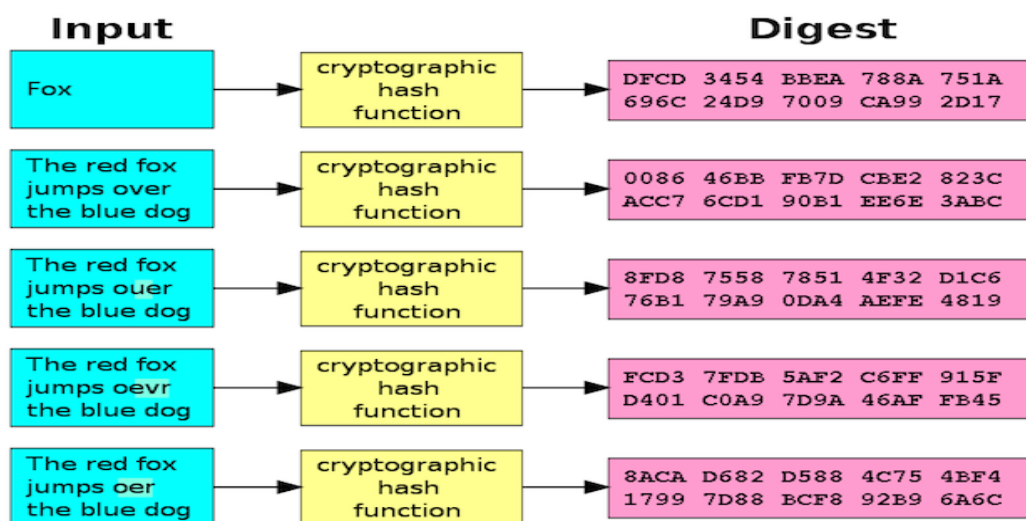
5.1.2 ENCRIPCIÓN

La encriptación es de suma importancia para la autenticación de usuarios laravel para encripta las contraseñas se utilizó el cifrado de hash laravel. Proporciona hash seguro de Bcrypt y Argon2 para almacenar las contraseñas de los usuarios, donde Bcrypt es una gran opción para las contraseñas de hash porque su factor de trabajo es ajustable,

lo que significa que el tiempo que lleva genera un hash puede aumentar a medida que aumente un hardware.

Una función criptográfica hash usualmente conocida como “hash” es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor hash de salida tendrá siempre la misma longitud.

Figura 5. 4: Función hash



Fuente: (B. Doinuhe, s.f.)

5.2 RESULTADOS

La realización del proyecto establece los objetivos a cumplir estos objetivos se fueron desglosando en los diferentes acápite, llegando haber concluido con resultados esperados, además de ser implementa el modulo financiero para el sistema de forestación y reforestaciones, caso proyecto unjp/bol/045/unj.

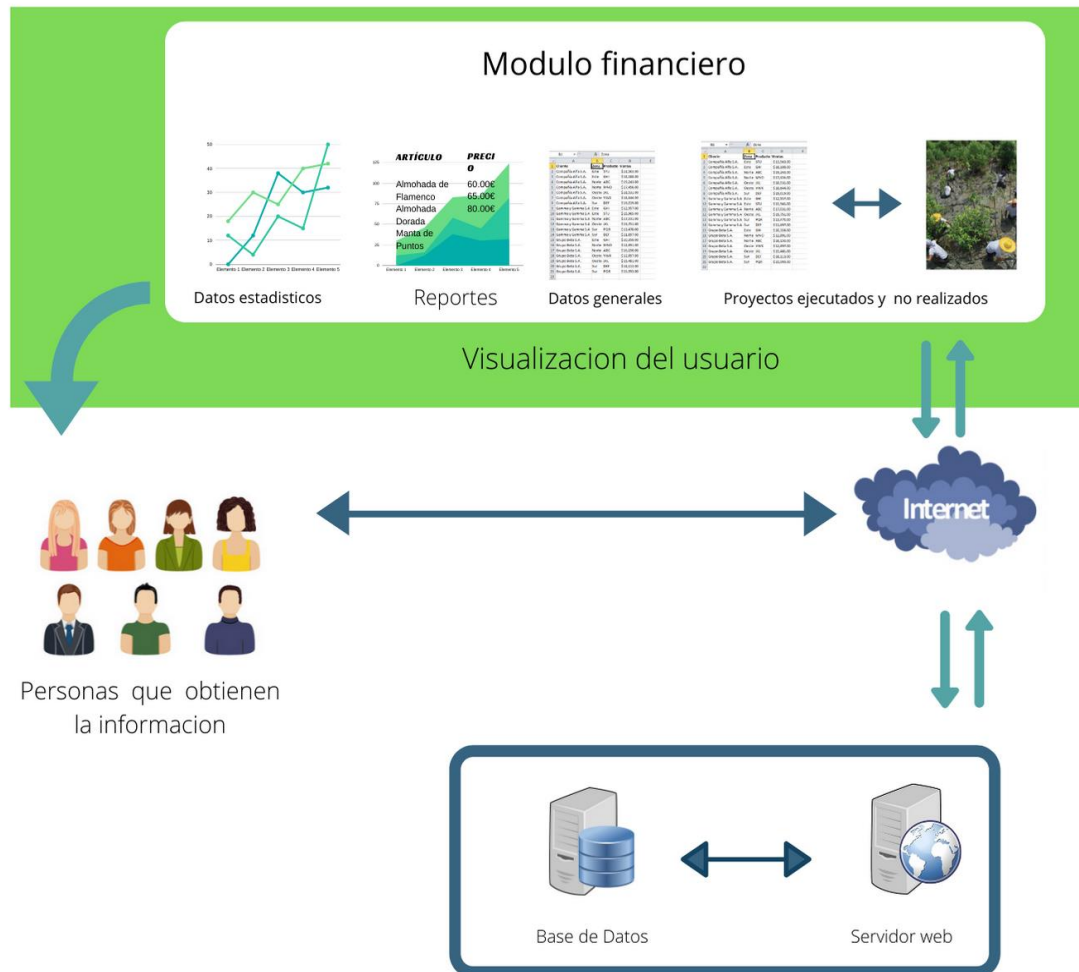
La información que se desconocía anteriormente, fue solucionada de gran manera dado que ahora la información se genera dinámicamente en gráficos estadísticos, permite generar reportes por distintos niveles de usuario, generar datos a nivel nacional, departamental, por provincias, municipios, comunidades e instituciones.

El modulo financiero es sumamente importante para el sistema de forestaciones y reforestaciones, los administradores del sistema obtienen la información sin dificultades

lo que anterior mente era moroso y dificultoso para quienes necesitaban la información.

En la siguiente imagen se visualiza el esquema funcional actual del módulo financiero de forestaciones y reforestaciones.

Figura 5. 5: Esquema funcional del módulo financiero



Fuente: (Elaboración propia)



CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

Se desarrolló el modulo financiero para el sistema de forestaciones y reforestaciones caso, proyecto UNJP/BOL/045/NJP, Donde a través del módulo se puede obtener la información de plantaciones y forestaciones por diferentes niveles, nivel nacional, departamental, provincial, municipal y por comunidades.

De la misma forma permite generar datos dinámicos, reportes dinámicos en diferentes formatos y gráficos estadísticos, con estos módulos implementados se solucionó la problemática de existe en la generación de reportes en el sistema de forestaciones y reforestaciones.

Fue desarrollado todos sus sub módulos que compone el modulo financiero, los cuales se mencionan en la siguiente descripción:

- Módulo de datos de plantaciones permite la obtención de datos a nivel nacional, departamental, provincias, municipios en forma grupal e individual por comunidades.
- Módulo de proyectos se visualiza los datos de proyectos realizados que son monitoreados por su ubicación geográfica y proyectos que no tienen ubicación en este caso los que no son aprobados o realizados.
- Módulo de reportes estadísticos que facilita la generación de datos en histogramas, líneas, mapas por diferentes niveles.
- Módulo de reportes permite obtener reportes en formatos pdf y xls de forma dinámica por diferentes niveles.
- Página principal del sistema es el portal del módulo de reportes financiero que permite el acceso a módulo de autenticación de usuario por intermedio de sus respectivos privilegios y por otra parte permite generar gráficos estadísticos a nivel nacional departamental e instituciones por cualquier usuario (persona), que no tenga privilegios de autenticación de acceso al módulo.

6.2 RECOMENDACIONES

En el proceso de desarrollo de software se hizo las pruebas de caja blanca y pruebas de caja negra, esto no significa que el software no tenga ningún error, por lo cual se recomienda a la institución que cuente con personal que del área para que realice mantenimientos a largo plazo.

- Si bien el desarrollo se realizó con éxito se recomienda la actualización y el mantenimiento, esto para el correcto funcionamiento del módulo.
- Realizar las copias de seguridad en la base de datos del sistema constante para evitar pérdida de datos debido a algún tipo de desastre.
- Se recomienda cambiar contraseñas mensual o semestral para dar mayor seguridad.

BIBLIOGRAFÍA

- Arias, A. (2016). Analisis de software. En *Analisis se software* (pág. 250).
- Armendáriz, A. M. (2016). Diseño e implementación de un sistema financiero contable. La Paz, Bolivia.
- Bolivia, F. (2015). Desarrollo e implementacion del sistema de monitoreo Hollistico e integral de bosques. *Fao Bolivia*, 1,2.
- Cotas, A. (1994). *Ingeniería de Software*. Soluciones Avanzadas (pág. 38)
- Daniel, Ramo. (2016). *Ingenieria de software*. 2da edicion.
- FAO. (2015). Programas Nacionales de ONU- Reducción de Emisiones de la Deforestación y Degradación. *Programa ONU-REDD*, (pág. 1-2).
- Gomes, A. (2019). UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE.
- Gómez, A., & López, M. C. (s.f.). UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE.
- Gonzales, P. A. (2016). Sistema Web de Generación de Reportes Gerenciales de Indicadores Financieros de Responsabilidad Social. La Paz, Bolivia.
- Gonzalez, D., & Renato R. (2016). Metrica de Ccomplejidad del Software y Procesos y Cognitivos. En *Metrica de Ccomplejidad del Software y Procesos y Cognitivos* (pág. 2).
- Levy, L. H. (2017). Planeacion financiera. En L. H. Levy, *Planeacion financiera* (pág. 42).
- Lewis, G. (1994). *What is Software Engineering*. DataPro (4015).
- Luis, G. (2017). *Informe de metodologia*. Bolivar (pág. 22-25).
- Marin, R. (2020). Lod getores de bases de datos. *RebistaDigital*.
- Mendez, J. A. (2015). Sistema contable informatico. Paraninfo.
- Muñiz, L. (2014). *El Reporting herramienta*. Sisconges y Estrategia.
- Niño, M. B. (2015). Contabilidad Sistema y gerencia. El Nacional.
- Obando, M. A. (2016). Plataforma Web para la Programacion y Control de Recursos Financieros. La Paz, Bolivia.
- Presman, R. (2010). *Ingenieria de software*. Mexico: Educación.
- Rojas, R. G. (2016). Sistema de Administración de proyectos mediante aplicaciones Web y Móvil. La Paz, Bolivia.
- Solis, C., Figueroa-Diaz, R., & Cabrera Silva, A. (2018). Metodologias Tradicionales vs. Metodologias Agiles. 9.
- Sommerville, I. (2011). *Ingeniería de Software*. Mexico: PEARSON EDUCACIÓN

WEBGRAFÍA

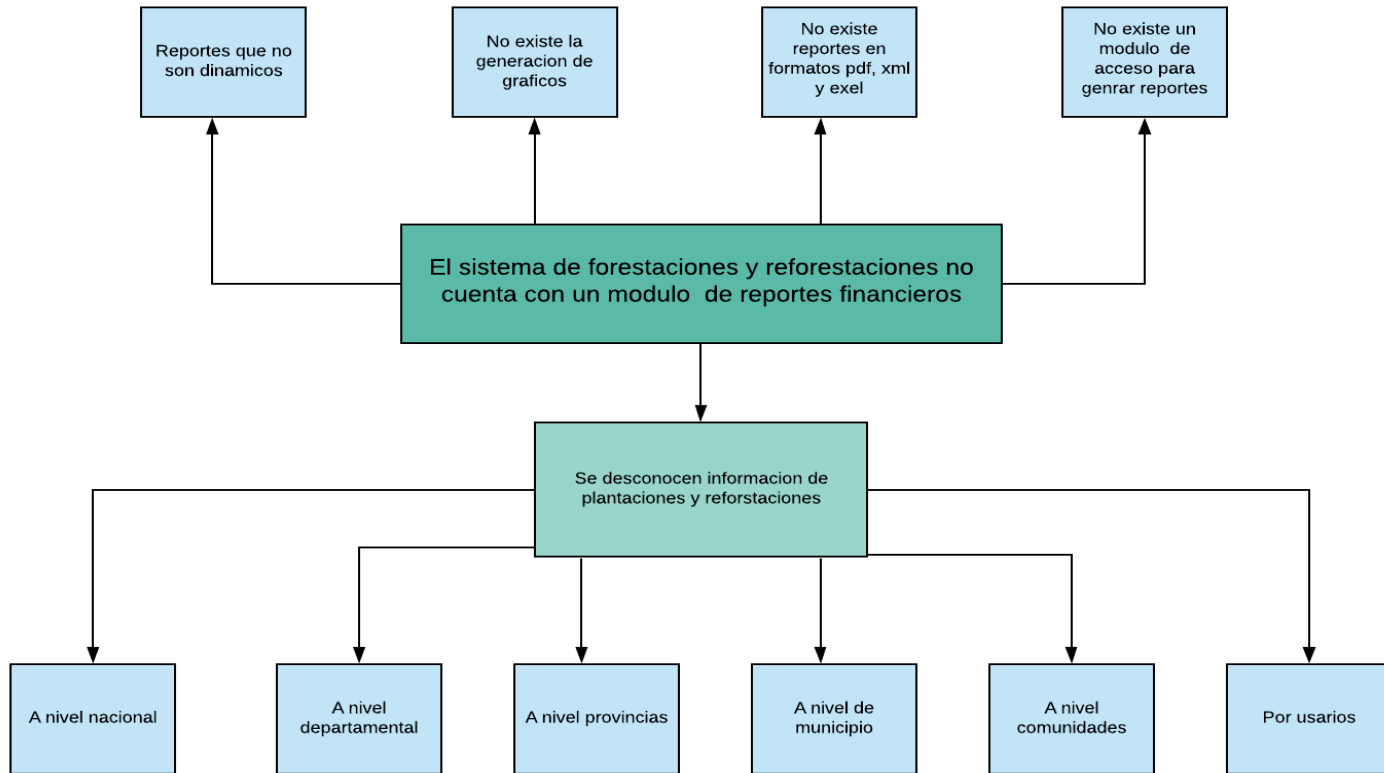
- Alegsa. (2018, Hrs: 16:36). *Definición de modulo*. Obtenido de <http://www.alegsa.com.ar/Dic/modulo.php>
- altp. (9 de mayo de 2019, Hrs: 18:12). *Introducción a la Calidad del Software*. Obtenido de <https://gsitic.wordpress.com/2019/05/09/biii17-la-calidad-del-software-y-su-medida-modelos-metricasnormas-y-estandares/>
- Álvarez, L., Helmer, A., & Yeison, G. (2017, Hrs: 12:55). Obtenido de Programación Orientada a Objetos: https://ferestrepoca.github.io/paradigmas-de-programacion/poo/poo_teoría/concepts.html
- Arias, P. (16 de enero de 2020, Hrs: 20:51). *Herramientas para el desarrollo web*. Obtenido de <https://www.pabloarias.eu/herramientas-desarrollo-web>
- atlp. (9 de mayo de 2019, Hrs: 06:12). *Introducción a la Calidad del Software*. Obtenido de <https://gsitic.wordpress.com/2019/05/09/biii17-la-calidad-del-software-y-su-medida-modelos-metricasnormas-y-estandares/>
- Belshaw, D. (16 de enero de 2019 Hrs: 13:07). *La Ingeniería Web*. Education. Obtenido de https://es.slideshare.net/uni_fcys_sistemas/la-ingeniera-web
- Bolaños, L. F., Vanessa Urrea, C., & Anyeli Gomez, R. (2017, Hrs: 18:12). *La ingeniería web*. Obtenido de <https://laingenieriaweb.wordpress.com/la-ingenieria-web/>
- Chart.js. (9 de febrero de 2020, Hrs: 16:32). *Chart.js*. Obtenido de <https://www.chartjs.org/>
- COCOMO. (17 de agosto de 2018, Hrs: 09:27). Obtenido de El Modelo Constructivo de Costos: <https://ipmoguide.com/cocomo-el-modelo-constructivo-de-costos/>
- Daniel, T., & Jadisha, Y. (20 de 05 de 2019, Hrs: 18:45). *Metodologías de desarrollo UWE*. Obtenido de <https://metodologiauwe.wordpress.com>
- Dimas, C. (23 de 11 de 2019, Hrs: 12:22). *Programación modular*. Obtenido de <https://www.docsity.com/es/programacion-modular-1/5144425/>
- Eliannys, S. (14 de noviembre de 2019, Hrs: 09:16). Obtenido de <https://conceptodefinicion.de/metodo/>
- Felipe, A. (14 de noviembre de 2017, Hrs: 17:39). *Metodología UWE*. Obtenido de <https://view.genial.ly/5a0bbf8021cf372478d5b94e/interactive-content-metodologia-uwe>
- Forestación*. (17 de Abril de 2014, Hrs: 10:05). Obtenido de <https://definicion.mx/forestacion/>
- Garza, J. (2019, Hrs: 14:33). *Metodologías ágiles*. Obtenido de <https://blog.conectart.com/metodologias-agiles>
- Gómez, A., López, M., & Migani, S. (20 de 05 de 2019, Hrs: 14:33). *Cocomo II*. Obtenido de https://www.ecured.cu/COCOMO_II

- Graus. (06 de febrero de 2017, Hrs: 07:18). *Informacion*. Obtenido de <https://www.significados.com/informacion/>
- Guaman, F. (7 de diciembre de 2015, Hrs: 22:44). *Fundamentos de Programacion*. 4-5. Obtenido de <https://es.slideshare.net/foguaman1/programacion-modular-55900674>
- Hernandez, U. (25 de octubre de 2019, Hrs: 06:55). *MVC*. Obtenido de <https://codigo.com/articulos/mvc-model-view-controller-explicado>
- Highcharts. (9 de Febrero de 2020, Hrs: 18:12). *Highcharts*. Obtenido de <https://apexcharts.com/>
- Hosting. (s.f. , Hrs: 21:46). Obtenido de <https://okhosting.com/blog/herramientas-de-desarrollo-de-software/>
- Huerta, L. N. (2014, Hrs: 19:31). Obtenido de Ingeniería web: <https://sg.com.mx/revista/14/ingenieria-web-las-aplicaciones-web-ingenieria>
- ISO/IEC 9126*. (2017, Hrs: 08:14). Obtenido de <https://sites.google.com/site/iso-9126>
- Jacobson, & Ivar. (s.f. ,Hrs: 17:29). Obtenido de https://www.ecured.cu/Metodologias_de_desarrollo_de_Software#Fuentes
- Kit, M. (23 de 04 de 2019, Hrs: 15:02). *Tecnologias en area financiera* . Obtenido de <https://www.forbes.com>
- Leandro, A. (13 de 06 de 2018, Hrs: 09:30). Obtenido de <http://www.alegsa.com.ar/Dic/dato.php>
- Martinez, A. (20 de 05 de 2015, Hrs: 18:12). *Metodologias para el desarrollo del software*. Obtenido de <https://es.slideshare.net/yeltsintorres18/metodologias-para-el-desarrollo-del-software>
- MegaInbound. (2017, Hrs: 19:55). Obtenido de Metodologias de desarrollo de software recuperado de <https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/metodologias-de-desarrollo-de-software>
- Navarro, J. (2015, Hrs: 13:08). Obtenido de <https://www.definicionabc.com/medio-ambiente/reforestacion.php>
- Palomar. (2016, Hrs: 16:12). Obtenido de <https://disenowebakus.net/control-y-desarrollo-de-proyectos.php>
- Perez, L. S. (12 de Diciembre de 2016, Hrs: 11:09). Obtenido de <https://conceptodefinicion.de/modelo/>
- PÉREZ, L. S. (17 de diciembre de 2016, Hrs: 08:18). Obtenido de <http://softwareverde.blogspot.com/2016/09/definicion-de-modelo.html>
- Porto, J. P., & Merino, M. (2016, Hrs: 20:31). Obtenido de <https://definicion.de/reporte-de-lectura/>
- Prieto, R. M. (21 de junio de 2018, Hrs: 10:01). *ISO 9126*. Obtenido de http://iso9126uts.blogspot.com/p/i_8.html

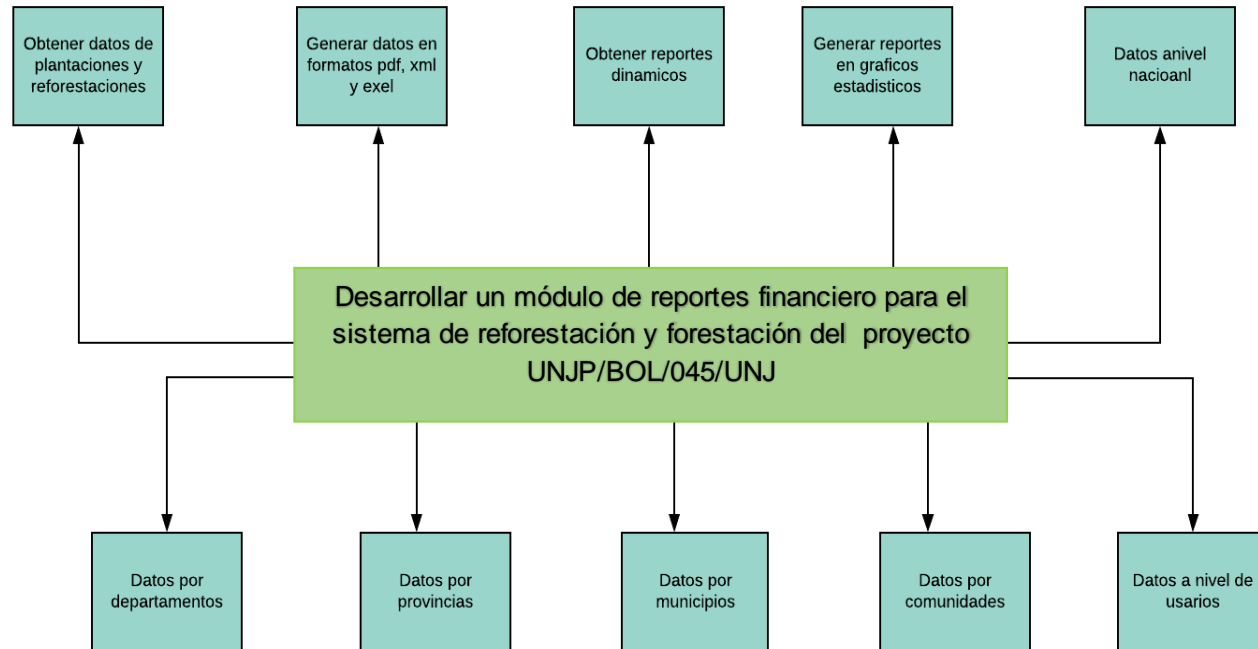
- Prieto, R. M. (21 de junio de 2018, Hrs: 12:52). *ISO/IEC 9126*. Obtenido de http://iso9126uts.blogspot.com/p/i_8.html
- Raffino, M. E. (2019, Hrs: 09:05). Obtenido de <https://concepto.de/software/#ixzz6D3G3pnWH>
- Redaccion*. (14 de noviembre de 2019, Hrs: 18:12). Obtenido de Concepto Definicion: <https://conceptodefinicion.de/registro/>
- Ruiz, L. F. (2018, Hrs: 07:16). Obtenido de <https://www.siigo.com/blog/empresario/que-es-un-costo-en-contabilidad/>
- Senn, J. (20 de enero de 2020, Hrs: 18:12). *Metodología de desarrollo de software*. Obtenido de https://Metodolog%C3%ADa_de_desarrollo_de_software
- Tarradellas, J. (2019, Hrs: 19:42). *Parners Academico*. Obtenido de <https://obsbusiness.school/int/blog-investigacion/finanzas/estados-financieros-tipos-de-informes-y-utilidades>
- Ubiquim. (s.f., Hrs: 15:33). Obtenido de <https://ubiquim.com/es/blog/20-herramientas-de-desarrollo-de-software-que-te-haran-mas-productivo/>
- Ucha, F. (2018, Hrs: 18:12). Obtenido de <https://www.definicionabc.com/medio-ambiente/forestacion.php>
- UWE. (10 de agosto de 2016, Hrs: 14:00). *Web Engineering Group*. Obtenido de <http://uwe.pst.ifi.lmu.de/publicationsMetamodelAndProfile.html>
- Velasco, R. (02 de 03 de 2019, Hrs: 13:15). Obtenido de <https://gestiondeproyectosplus.com/que-es-un-proyecto/>

ANEXOS

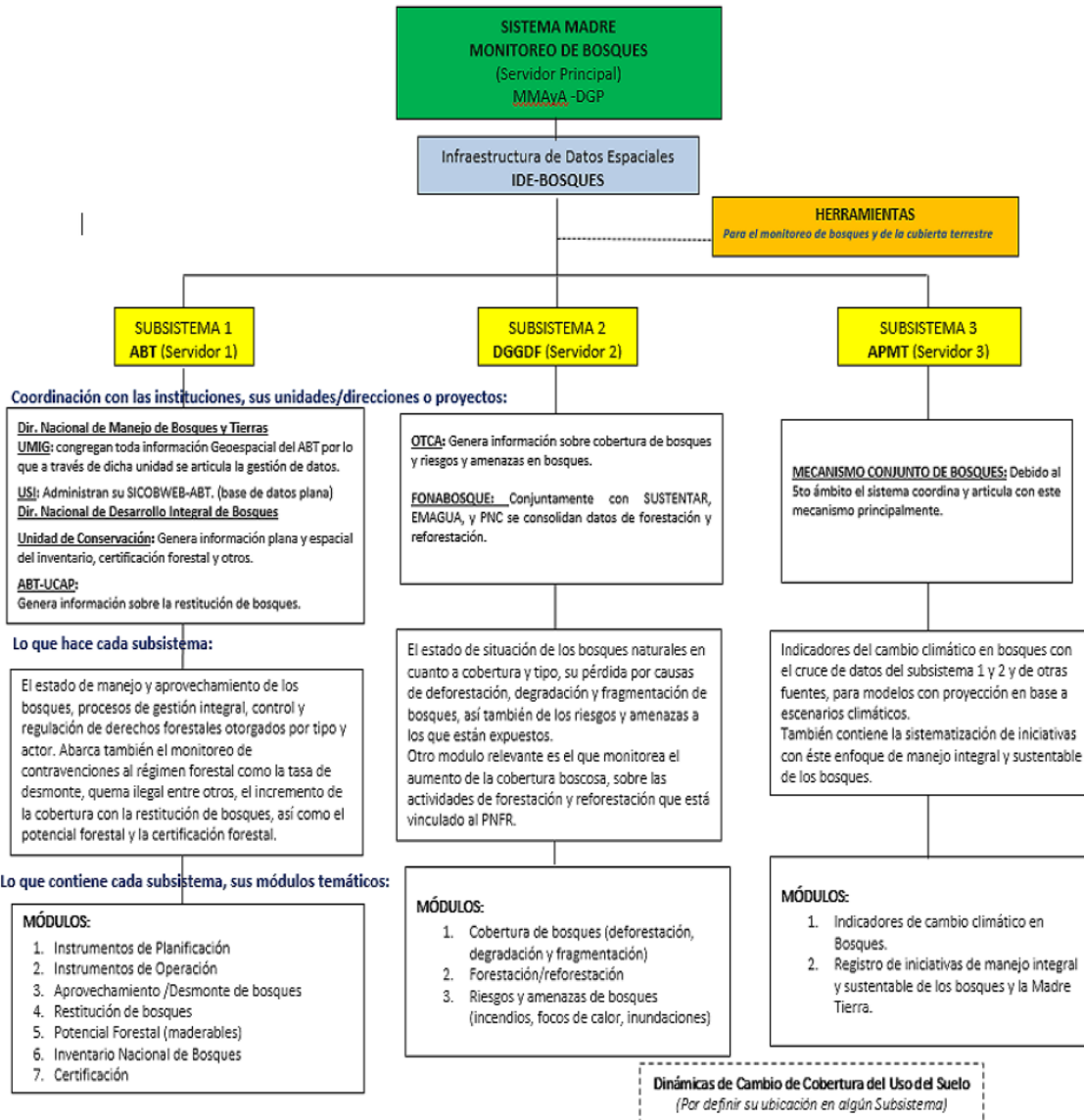
ÁRBOL DE PROBLEMAS



ÁRBOL DE OBJETIVOS



ORGANIGRAMA DE SISTEMA DEL PROYECTO UNJP/BOL/045/UNJ.



ENCUESTA SOBRE LA USABILIDAD DEL SISTEMA

ENCUESTA SOBRE LA USABILIDAD DEL SISTEMA

Módulo Financiero para el Sistema de Forestaciones y Reforestaciones

Caso: Proyecto unjp/bol/045/njp

N° <i>2</i>			
Datos personales: <i>Paulino Choque 60707590</i>			
N°	Presuntas	Respuesta	
		Si	No
1	¿No es complicado usar el sistema?		X
2	¿Las pantallas que vista fueron de su agrado	X	
3	¿Los datos obtenidos que vio fueron fáciles de comprender?	X	
4	¿El sistema responde rápido a sus solicitudes?	X	
5	¿El sistema facilita el trabajo para generar datos?		
6	¿El sistema reduce el tiempo de trabajo en generar informes?	X	
7	¿Es fácil navegar por las distintas opciones?	X	
8	¿La generación de datos que se realizan no es complicada?		X
9	¿El sistema no presento errores?		X
Total			



Firma

MANUAL DE USUARIO

Introducción



La documentación es la guía final para la administración del **módulo financiero para el sistema de forestaciones y reforestaciones, caso proyecto unjp/bol/045/unj**.

El personal autorizado para la administración del mencionado modulo deberán revisar minuciosamente y contrastar con el **módulo financiero para el sistema de forestaciones y reforestaciones** esto para el mejor uso y no tener dificultades en la manipulación del mismo con los distintos módulos interno que tiene.

I. Acceso al Módulo web

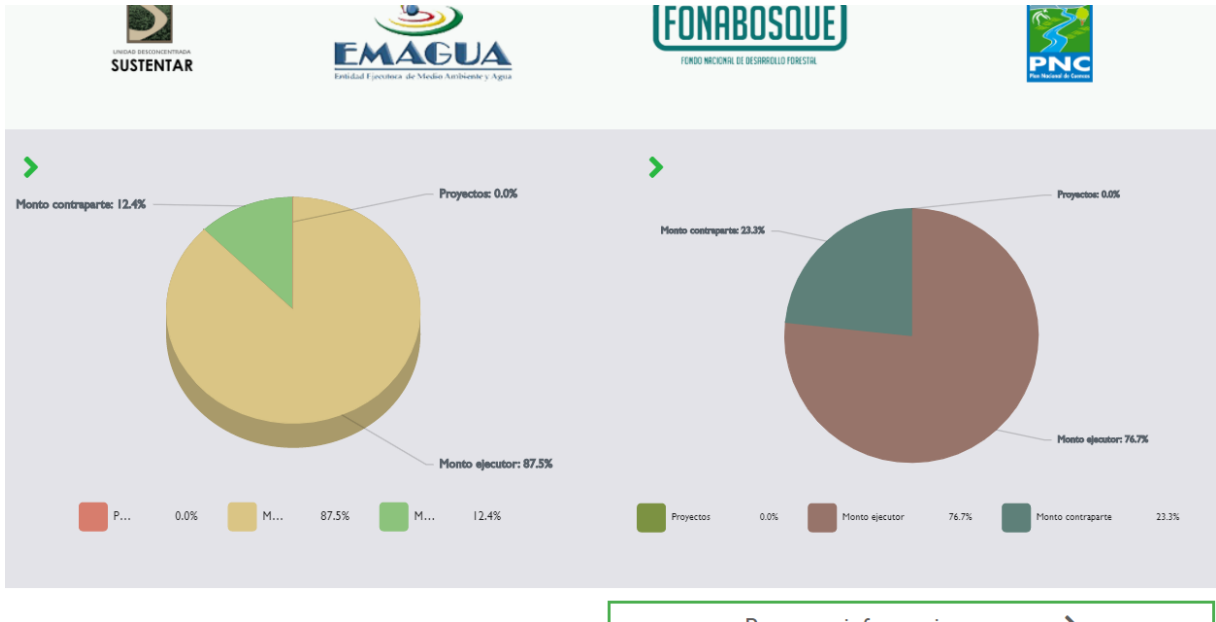
El sistema se encuentra subido el servidor de la institución por lo cual cualquier persona puede acceder a la página principal del módulo.

Generará reportes estadísticos sin la autenticación, no se necesita un usuario ni la contraseña debe hacer **click** en botón generar reportes luego se desglosara tres opciones, puede hacer **click** en la opción que dese para su información.

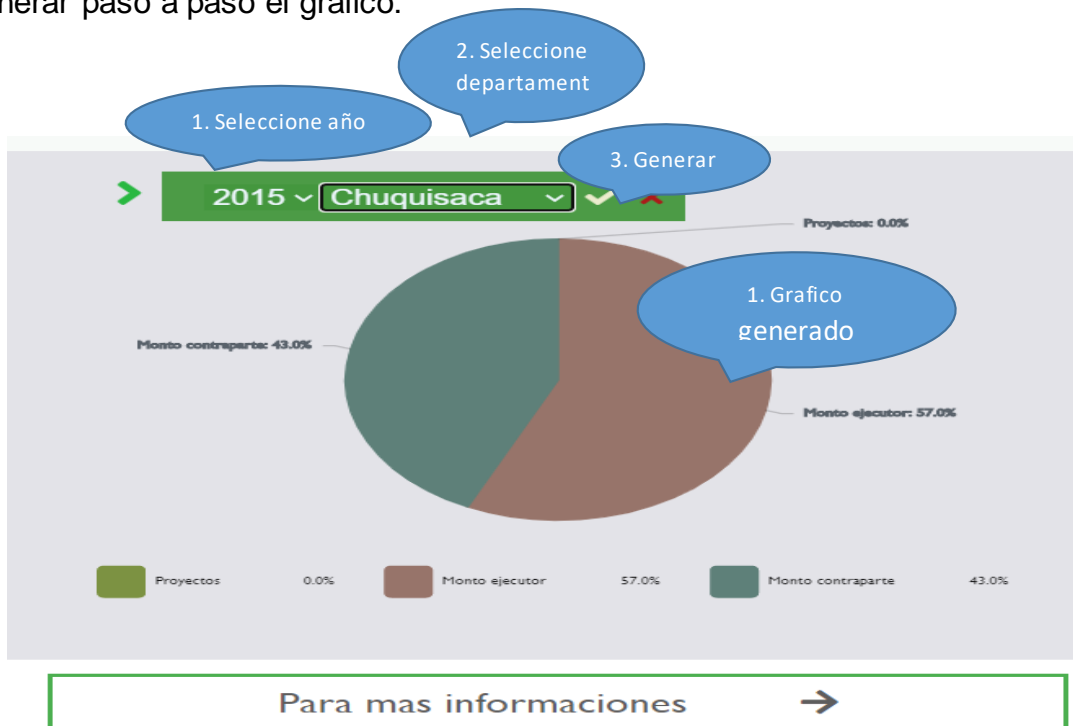
Después de elegir una opción se generara los datos en gráficos estadísticos, para generar más datos dígase al botón  haga click y se desplegará  Año puede filtrar por las opciones que guste, en los tres gráficos que se muestran a nivel general, departamental e instituciones



Generacion de graficos por cualquier usuario que accede al sistema, en el siguiente imagen se muestra dos graficos en tortas los campos tomados son monto contraparte, monto ejecutor y la cantidad de proyectos.



Generar paso a paso el grafico:





II. Autenticación de usuario

Si cuenta con credenciales de autenticación y privilegios para su acceso al sistema de forstaciones y reforestaciones dirigese al boton [Para mas informaciones](#) y se desglosara un panel de Login de seccion. Ingrese su correo y sus contraseña luego dirigire a la opcion Login para poder acceder al sistema.

The login form is titled 'Login' and features two input fields: 'Email' and 'Password'. A 'LOGIN' button is located at the bottom. Blue callout bubbles provide instructions: 'Ingrese Email' for the email field, 'Ingrese contraseña' for the password field, and 'Inicie sesión' for the login button.

III. Pantalla principal

Después de haber realizado los pasos de inicio de sesión se visualiza en modulo principal para la administración del sistema.

Ministerio de Medio Ambiente y Agua

Forestaciones y reforestaciones en Bolivia Gestion 2020

179 2013 Proyectos	179 2014 Proyectos	179 2015 Proyectos	179 2016 Proyectos
Total presupuesto 231,809	Total presupuesto 1,231,809	Total presupuesto 231,809	Total presupuesto 231,809

← Opciones para generar datos

Click, salir

IV. Módulo datos de plantaciones

Ministerio de Medio Ambiente y Agua

La Paz

Mostrar 10 registros

Cod. proyecto	Departamento	Provincia	Municipio	Entidad ejecutora	Periodo plantacion	Tipo plantacion	Ubicacion	Responsable
3	La Paz	Abel Iturralde	San Buenaventura	FONABOSQUE	2015-2016	Conservación	Com. San Miguel	NELLY M. MITA POMA
5	La Paz	Abel Iturralde	San Buenaventura	FONABOSQUE	2015-2016	Conservación	Com. San Miguel	NELLY M. MITA POMA
6	La Paz	Abel Iturralde	San Buenaventura	FONABOSQUE	2015-2016	Conservación	Com. San Miguel	NELLY M. MITA POMA
8	La Paz	Abel Iturralde	San Buenaventura	FONABOSQUE	2015-2016	Conservación	Com. San Miguel	NELLY M. MITA POMA
10	La Paz	Abel Iturralde	San Buenaventura	FONABOSQUE	2015-2016	Protección	Com. Buena Vista	NELLY M. MITA POMA
12	La Paz	Abel Iturralde	San Buenaventura	FONABOSQUE	2015-2016	Conservación	Com. Buena Vista	NELLY M. MITA POMA
14	La Paz	Abel Iturralde	San Buenaventura	FONABOSQUE	2015-2016	Conservación	Com. Buena Vista	NELLY M. MITA POMA
17	La Paz	Franz Tamayo	Apolo	FONABOSQUE	2015-2016	Conservación	Com. Charque	NELLY M. MITA POMA
18	La Paz	Franz Tamayo	Apolo	FONABOSQUE	2015-2016	Protección	Com. Charque	NELLY M. MITA POMA
21	La Paz	Franz Tamayo	Apolo	FONABOSQUE	2015-2016	Conservación	Com. Charque	NELLY M. MITA POMA

Mostrando registros del 1 al 10 de un total de 223 registros

Anterior 1 2 3 4 5 ... 23 Siguiente

1. click

2. click

3. generar

4. Click, para generar por

Buscar

2. Click

3. Click, generar

Buscar

1. Click, tipo proyecto

Codigo proyecto	Departamento	Provincia	Municipio	Ubicacion	Periodo platacion	Tipo platacion	Cofinanciador
441	La Paz	Murillo	Nuestra Señora de La Paz		2015-2016	Ornamental	GAM NUESTRA SEÑORA DE LA PAZ
442	La Paz	Inquisivi	Licoma (Villa Libertad)	Com. Charapaxi	2015-2016	Comercial	GAM LICOMA
450	La Paz	Inquisivi	Licoma (Villa Libertad)	Com. Charapaxi	2015-2016	Comercial	GAM LICOMA
463	La Paz	Abel Iturralde	San Buenaventura	Com. Villa Alcira	2015-2016	Comercial	GAM SAN BUENAVENTURA
486	La Paz	Abel Iturralde	San Buenaventura	Com. San Miguel	2015-2016	Conservación	GAM SAN BUENAVENTURA
488	La Paz	Abel Iturralde	San Buenaventura	Com. San Miguel	2015-2016	Conservación	GAM SAN BUENAVENTURA
489	La Paz	Abel Iturralde	San Buenaventura	Com. San Miguel	2015-2016	Conservación	GAM SAN BUENAVENTURA
490	La Paz	Abel Iturralde	San Buenaventura	Com. San Miguel	2015-2016	Conservación	GAM SAN BUENAVENTURA
495	La Paz	Abel Iturralde	San Buenaventura	Com. San Miguel	2015-2016	Conservación	GAM SAN BUENAVENTURA

Opciones de

Opciones de página

VII. Módulo de reportes estadísticos

Para obtener información en diferentes gráficos estadísticos, diríjase a la opción de reportes estadísticos, las opciones son varias no obstante para generar estos gráficos los pasos son similares, se visualiza a continuación el módulo de reportes por departamentos.

2. 1ra fecha

3. 2da fecha

4. generar

1. Seleccione, click

Total de proyectos por departamento

bamba: 13.1%	Beni: 4.8%
Tarja: 3.6%	Pando: 2.1%
La Paz: 12.1%	Santa Cruz: 21
Oruro: 10.1%	Chuquisaca: 2
Potosi: 5.0%	

Estadísticas departamentales por mes

Presupuesto ejecutor por departamento

bamba: 14.1%	Beni: 3.5%
Tarja: 2.9%	Pando: 6.3%
La Paz: 8.9%	Santa Cruz: 9.
Oruro: 4.9%	Chuquisaca: 3
Potosi: 15.4%	

Para visualizar datos por mes e instituciones los pasos a seguir son las siguientes.



VIII. Módulo de reportes

Es la información por diferentes niveles de forestaciones y reforestaciones los pasos para generar son de similar forma. A continuación, se muestra los pasos de generación de información por departamentos.

1. Selecciona,

2.1ra y 2da

3. Generar

5. Filtrar por tipo de plantación y tipo de proyecto

Datos

Descar


Selecciona

Departamento	Cantidad Plantado	Beneficiarios	Empleos	Monto Ejecutor	Monto Contraparte	Total Monto	Superficie plantado
Pando	21	21	1100	4305100	1604694	5909794	1561.3235
Beni	35	35	533	4644581.02	548816	5193397.02	2727.57
Tarija	63	63	418	3472710.28	411666	3884376.28	869.716
Potosí	71	70	88934	20329623.53	4329654.2	24659277.73	49694.1
La Paz	124	124	486	11776701.06	761295.12	12537996.18	2473.9098
Santa Cruz	125	125	811	12821151.94	3493460.56	16314612.5	2789.2624
Cochabamba	134	134	709	17562955.87	2673711.5	20236666.87	7908.13
Chuquisaca	180	180	363	6513884.52	547600	7061484.52	1311.992
Oruro	314	314	517	42185524.07	24745361.5	66931885.57	9924.47
Total	1066	1066	93871	123612231.79	39117258.88	162729490.67	526513.3965

Año	Cantidad Plantado	Beneficiarios	Empleos	Monto Ejecutor	Monto Contraparte	Total Monto	Superficie plantado
2012	7	7	284	1857280.6	565013	2422293.6	390.925
2013	20	20	65	4206674.55	0	4206674.55	517.44
2014	35	35	25	5286456.28	0	5286456.28	717.326
2016	54	54	100	260194.12	123.12	260317.24	109.0208
2015	107	107	93	619477.5	551632	1171109.5	1000.858
Total	223	223	567	12230083.06	1116768.12	13346851.18	2735.5698

IX. Módulo de ubicaciones geográficas

En este módulo se visualiza los diferentes proyectos que son monitoreados por su ubicación geográfica.

Para generar desplace el módulo de proyectos  **Proyectos** luego seleccione proyectos aprobados se visualiza dos partes primera parte solo un listado de todo los proyectos y una segunda parte donde muestra la geo localización en mapas.

Para visualizar un proyecto debe buscar por el código del proyecto que puede generar fácilmente en la parte del listado del proyecto.

Proyectos no aprobados o no ejecutados:

1. Generar datos

La Paz

Mostrar 10 registros

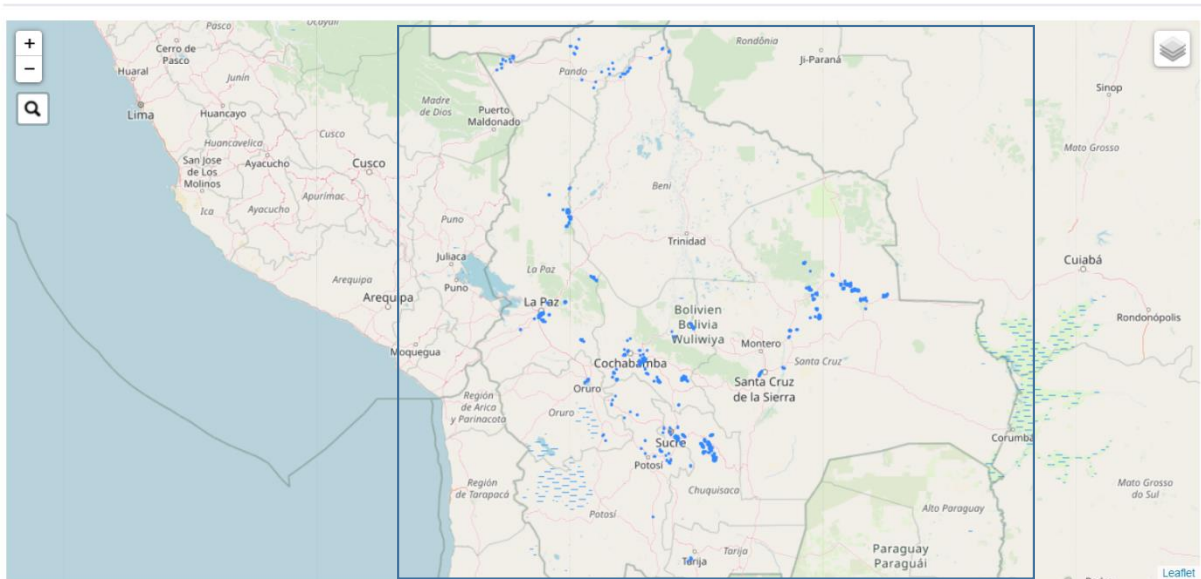
Código proyecto	Departamento	Provincia	Municipio	Periodo plantacion	Tipo plantacion	Cofinanciador	Superficie plantado
3	La Paz	Abel Iturralde	San Buenaventura	2015-2016	Conservación	GAM SAN BUENAVENTURA	8.1
			San Buenaventura	2015-2016	Conservación	GAM SAN BUENAVENTURA	23.2
6	La Paz	Abel Iturralde	San Buenaventura	2015-2016	Conservación	GAM SAN BUENAVENTURA	13.5
8	La Paz	Abel Iturralde	San Buenaventura	2015-2016	Conservación	GAM SAN BUENAVENTURA	1.4
10	La Paz	Abel Iturralde	San Buenaventura	2015-2016	Protección	GAM SAN BUENAVENTURA	11.1
12	La Paz	Abel Iturralde	San Buenaventura	2015-2016	Conservación	GAM SAN BUENAVENTURA	0.1
14	La Paz	Abel Iturralde	San Buenaventura	2015-2016	Conservación	GAM SAN BUENAVENTURA	4.6
17	La Paz	Franz Tamayo	Apolo	2015-2016	Conservación	GAM APOLO	8.8
18	La Paz	Franz Tamayo	Apolo	2015-2016	Protección	GAM APOLO	0.8
		Franz Tamayo	Apolo	2015-2016	Conservación	GAM APOLO	2.5

Mostrar 10 registros total de 99 registros ...

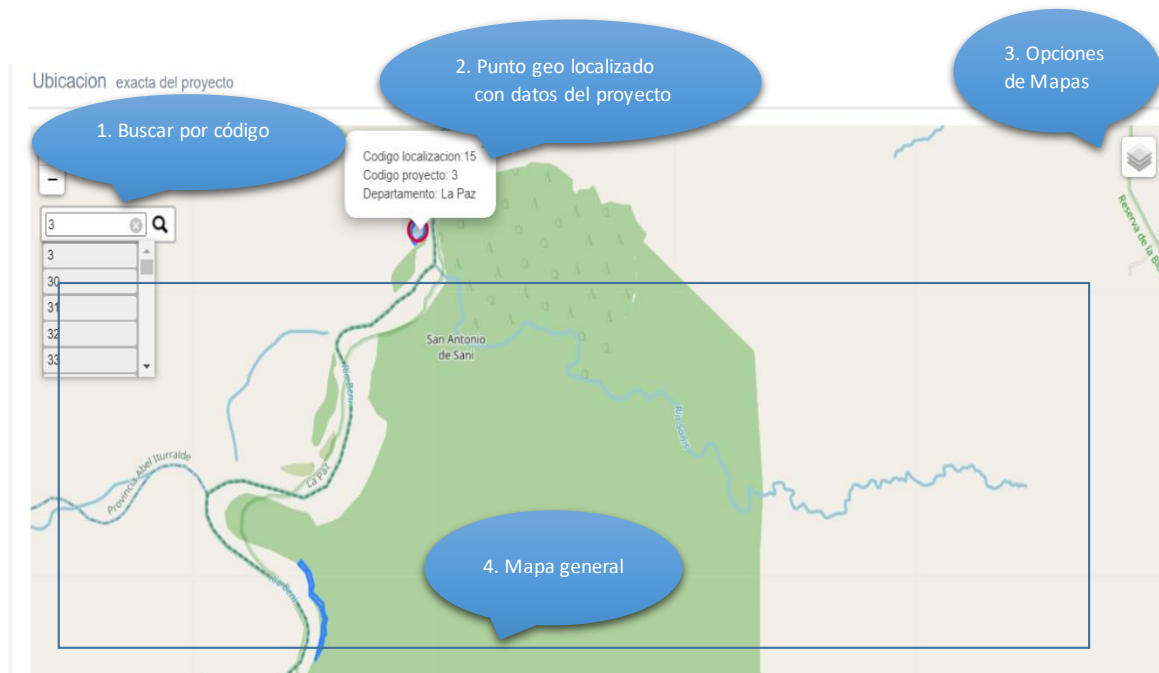
2. Código de proyectos

3. Exportar

En el siguiente manual se muestra la visualización geográfica con los proyectos monitoreados.



Generar la ubicación exacta de cada proyecto paso por paso:



AVAL DE CONFORMIDAD

El Alto, 10 de julio de 2020

Señor:

Ing. Enrique Flores Baltazar

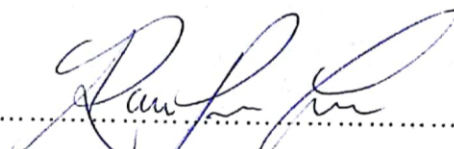
TUTOR METODOLÓGICO TALLER II**Presente:****Ref.- Aval de conformidad**

Distinguido ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado **“MODULO FINANCIERO PARA EL SISTEMA DE REFORESTACION Y FORESTACION, CASO: PROYECTO UNJP/BOL/045/UNJ”**, que propone el postulante Univ. Calixto Torrez Uruña, con cedula de identidad 9176035 LP. Para su defensa publica, evaluación correspondiente a la materia de Taller de licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de EL Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.



.....
Lic. Rosa Patricia Nina Chura

AVAL DE CONFORMIDAD

El Alto, 27 de agosto de 2020

Señor:

Ing. Enrique Flores Baltazar

TUTOR METODOLÓGICO TALLER II**Presente:****Ref.- Aval de Conformidad**

Distinguido ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto de grado “**MODULO FINANCIERO PARA EL SISTEMA DE REFORESTACION Y FORESTACION, CASO: PROYECTO UNJP/BOL/045/UNJ**”, que propone el postulante Univ. Calixto Torrez Uruña, con cedula de identidad 9176035 L.P. Para su defensa publica, evaluación correspondiente a la materia de Taller de licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de EL Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.

.....
Ing. Roberto Camilo Choque Apaza

CI.: 6953913 LP

AVAL DE CONFORMIDAD

El Alto, 28 de agosto de 2020

Señor:

Ing. David Carlos Mamani Quipe

DIRECTOR DE CARRERA DE INGENIERÍA DE SISTEMAS

Presente:


Ref.- Aval de Conformidad

Distinguido ingeniero:

Mediante la presente tengo a bien comunicarle mi conformidad del proyecto del grado "MODULO FINANCIERO PARA EL SISTEMA DE REFORESTACION Y FORESTACION, CASO: PROYECTO UNJP/BOL/045/UNJ", que propone el postulante Univ. Calixto Torrez Uruña, con cedula de identidad 9176035 LP. Para su defensa publica, evaluación correspondiente a la materia de Taller de licenciatura II, de acuerdo a reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de EL Alto.

Sin otro particular, reciba saludos cordiales.

Atentamente.



Ing. Enrique Flores Baltazar
TUTOR METODOLOGICO
TALLER DE LICENCIATURA II



Santa Cruz, 24 de agosto 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 en el perfil de proyecto de grado, para su elaboración y desarrollo, titulado **“MÓDULO FINANCIERO PARA EL SISTEMA DE REFORESTACIÓN Y FORESTACIÓN”** caso: **(PROYECTO UNJP/BOL/045/UNJ)** elaborado por el universitario **Calixto Torrez Uruña**, con cedula de identidad **9176035 LP.** y **R.U. 14002245**, 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,

[Handwritten signature]
Tomás Tapia Suño
AREA INFORMATICA
UNJP/045/BOL