

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“PLATAFORMA WEB PARA ADMINISTRACIÓN Y CONTROL DE PROYECTOS DE CONSTRUCCIÓN EN OBRAS CIVILES”

CASO: EMPRESA DE SERVICIOS ELÉCTRICOS, CIVILES AMBIENTALES
BOLIVIA S.R.L.

Para Optar al Título de Licenciatura en Ingeniería de Sistemas
MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Carlos Colque Huanca

Tutor Metodológico: M. Sc. Lic. Ing. Maricel Yarari Mamani

Tutor Especialista: Lic. Fredy Alanoca Coareti

Tutor revisor: Lic. Ing. Yolanda Escobar Mancilla

EL ALTO – BOLIVIA

2023

DEDICATORIA

Dedico el presente proyecto principalmente a Dios, por darme la fuerza necesaria para culminar esta meta.

A mis padres, por acompañarme en cada paso que doy en la búsqueda de ser mejor persona, profesional y por motivarme a seguir hacia adelante.

A todos mis amigos y compañeros por su ayuda desinteresada durante el transcurso de nuestro tiempo en la carrera.

AGRADECIMIENTOS

Agradezco a Dios por siempre cuidarme, por darme fuerza y salud e inteligencia para culminar con mi trabajo de grado.

A mi padres y hermanos por todo el apoyo incondicional que me brindaron en todo el transcurso de mi carrera.

A la M. Sc. Lic. Ing. Maricel Yarari Mamani por su tiempo y paciencia en la revisión del trabajo de grado.

Al Lic. Fredy Alanoca Coareti por su tiempo, enseñanza y paciencia en el desarrollo del presente proyecto de grado.

A la Lic. Ing. Yolanda Escobar Mancilla por su tiempo, enseñanza y paciencia en la documentación del proyecto de grado.

RESUMEN

El presente proyecto de grado denominado Plataforma Web para Administración y Control De Proyectos De Construcción En Obras Civiles la Empresa boliviana de servicios eléctricos, civiles y ambientales S.R.L., lleva muchos años prestando servicios a la construcción en Bolivia, realizando diseños arquitectónicos muy innovadores. Sin embargo, como empresa constructora de buena reputación y de tan rápido crecimiento, es necesario obtener información en el menor tiempo posible y conocer el avance de las obras en curso, para un control continuo, tener conocimiento diario de la cantidad de herramientas de trabajo y materiales, para que de esta manera se puedan brindar servicios eficientes a todos los clientes atendidos por la empresa.

El proyecto se utilizó tecnologías emergentes como MariaDB, un sistema de gestión de bases de datos relacionales, para el almacenamiento de los datos. El framework Laravel, un framework de desarrollo y diseño web basado en PHP, que proporciona una estructura sólida y herramientas avanzadas para la creación de aplicaciones web eficientes.

El proyecto se desarrolló utilizando la metodología UWE (UML - Based Web Engineering), ya que sus fases se adaptan y cumplen con los requisitos para el desarrollo del sistema. Se buscó garantizar la calidad del sistema de acuerdo con el estándar ISO 25000, que permite realizar una evaluación exhaustiva del sistema y asegurar su cumplimiento de los estándares de calidad establecidos.

El presente proyecto se desarrolló con una interfaz gráfica amigable diseñada para ofrecer una experiencia intuitiva y fácil de usar para los usuarios, mediante una navegación sencilla.

Palabras Clave: UWE, ISO 25000

ÍNDICE GENERAL

CAPÍTULO I	3
1. MARCO PRELIMINAR.....	1
1.1 INTRODUCCIÓN	1
1.2 ANTECEDENTES.....	2
1.2.1 Antecedentes Institucionales	2
1.2.1.1 Visión... ..	3
1.2.1.2 Misión	3
1.2.1.3 Organigrama.....	3
1.2.2 Antecedentes afines al Proyecto de Grado	4
1.2.2.1 Antecedentes Nacionales	4
1.2.2.2 Antecedentes Internacionales	4
1.3 PLANTEAMIENTO DEL PROBLEMA.....	6
1.3.1 Problema Principal.....	6
1.3.2 Problemas Secundarios.....	6
1.3.3 Formulación del Problema	7
1.4 OBJETIVOS.....	7
1.4.1 Objetivo General.....	7
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 METODOLOGÍA	10
1.6.1 Metodología UWE.....	10

1.6.2	Métricas de Calidad	10
1.6.3	Seguridad	11
1.6.4	Pruebas de Software	11
1.7	HERRAMIENTAS	13
1.8	LÍMITES Y ALCANCES	14
1.8.1	Límites	14
1.8.2	Alcances	15
1.9	APORTES.....	15
CAPÍTULO II		17
2.	MARCO TEÓRICO	17
2.1.	INTRODUCCIÓN	17
2.2.	PLATAFORMA WEB	17
2.3.	ADMINISTRACIÓN DE PROYECTOS	17
2.4.	CONTROL DE PROYECTOS.....	17
2.5.	PROYECTOS DE CONSTRUCCIÓN	18
2.6.	WEB 2.0.	18
2.6.1.	Características de la Web 2.0.....	18
2.6.2.	Usos de la Web 2.0	19
2.7.	METODOLOGÍA INGENIERÍA WEB BASADA EN UML (UWE).....	20
2.7.1.	Características.....	21
2.7.2.	Modelos de la metodología UWE	21
2.7.2.1.	Modelo de análisis de requerimientos	22
2.7.2.2.	Modelo de Casos de Uso	25
2.7.2.3.	Modelo contenido	26
2.7.2.4.	Modelo de Navegación.....	27

2.7.2.5. Modelo de Presentación	28
2.8. ARQUITECTURA DE SOFTWARE (MVC)	29
2.8.1. Modelo	30
2.8.2. Vista	31
2.8.3. Controlador	31
2.8.4. Características del Modelo Vista Controlador	32
2.9. HERRAMIENTAS	33
2.9.1. MagicDraw	33
2.9.2. Gestor de base de datos MySql	35
2.9.3. Lenguaje de Programación PHP	38
2.9.4. Bootstrap	39
2.9.5. JavaScript	40
2.9.6. jQuery	41
2.10. MÉTRICAS DE CALIDAD DE SOFTWARE	42
2.10.1. Calidad de Software	43
2.10.2. Norma ISO/IEC 25000	44
2.10.3. Clasificación de las características y sub características	46
2.10.3.1. Adecuación funcional	46
2.10.3.2. Eficiencia de desempeño	47
2.10.3.3. Compatibilidad:	48
2.10.3.4. Usabilidad	49
2.10.3.5. Fiabilidad	51
2.10.3.6. Seguridad	53
2.10.3.7. Mantenibilidad	54
2.10.3.8. Portabilidad	56

2.11. ANÁLISIS DE COSTOS.....	57
2.11.1. Cocomo II	57
2.11.2. Características.....	57
2.11.3. Ecuaciones del modelo cocomo	58
2.11.4. Modelos de COCOMO II.....	59
2.11.4.1. Modelo Básico.....	59
2.11.4.2. Modelo Intermedio.....	60
2.11.4.3. Atributos de Coste.....	60
2.12. SEGURIDAD	63
2.12.1. SGSI (Sistema de Gestión de la Seguridad de la Información).....	63
2.12.2. Aspectos de seguridad	64
2.13. PRUEBAS DE SOFTWARE	66
2.13.1. Pruebas de caja blanca	66
2.13.2. Pruebas de ruta básica.....	67
2.13.3. Pruebas de caja negra.....	69
2.13.4. Pruebas de estrés.....	70
CAPÍTULO III	72
3. MARCO APLICATIVO	72
3.1. DESARROLLO DE LA METODOLOGÍA	72
3.1.1. Requerimientos funcionales	73
3.1.2. Requerimientos no funcionales	74
3.1.3. Descripción de actores	75
3.1.4. Diagrama de caso de uso general del sistema.....	76
3.1.5. Diseño conceptual	86
3.1.6. Modelo de Navegación.....	87

3.1.7. Modelo presentación	89
3.1.8. Modelo de implementación	90
CAPÍTULO IV	104
4. CALIDAD, PRUEBAS, COSTOS Y SEGURIDAD DEL SOFTWARE.....	104
4.1. MÉTRICAS DE CALIDAD.....	104
4.1.1. Funcionalidad	104
4.1.2. Confiabilidad.....	109
4.1.3. Usabilidad	110
4.1.4. Eficiencia	112
4.1.5. Mantenibilidad	113
4.1.6. Portabilidad.....	114
4.2. ANÁLISIS DE COSTOS.....	116
4.3. SEGURIDAD	121
4.3.1. Seguridad Lógica.....	121
4.3.2. Seguridad Física.....	122
4.4. PRUEBAS AL SOFTWARE	123
4.4.1. Prueba de caja blanca	123
4.4.2. Prueba de caja negra	125
4.4.3. Prueba de estrés	126
CAPÍTULO V	128
5. CONCLUSIONES Y RECOMENDACIONES	115
5.1. CONCLUSIONES	115
Bibliografía	117
ANEXOS	122
ANEXO A	

ANEXO B
ANEXO C

ÍNDICE DE TABLAS

Tabla 1. Definición de métricas de calidad y fórmulas de la Adecuación Funcional.	46
Tabla 2. Definición de métricas y fórmulas de la Eficiencia de desempeño.	47
Tabla 3. Definición de métricas y fórmulas de la Compatibilidad.	48
Tabla 4. Definición de métricas de calidad y fórmulas de la usabilidad.	50
Tabla 5. Definición de métricas de calidad y fórmulas de fiabilidad.	51
Tabla 6. Definición de métricas de calidad y fórmulas de seguridad.....	53
Tabla 7. Definición de métricas de calidad y fórmulas de mantenibilidad.	54
Tabla 8. Definición de métricas de calidad y fórmulas de la portabilidad.	56
Tabla 9. Coeficientes del modelo Básico.	59
Tabla 10. Coeficientes del modelo intermedio.	60
Tabla 11. Atributos de costes totales.	61
Tabla 12. Herramientas usadas para la ingeniería de requerimientos.	72
Tabla 13. Requerimientos funcionales.	73
Tabla 14. Requerimientos no funcionales.	75
Tabla 15. Descripción de actores.	75
Tabla 16. Descripción de caso de uso – Gestionar usuarios.	78
Tabla 17. Descripción de caso de uso de administradores.	79
Tabla 18. Caso de uso agregar ocupación.....	81
Tabla 19. Descripción de caso de uso de gestionar presupuestos.	82
Tabla 20. Descripción de caso de uso habilitar formulario.	84
Tabla 21. Descripción caso de Uso - Seguimiento del cliente.....	85
Tabla 22. Número de Entradas de usuario.....	104
Tabla 23. Número de Salidas Usuario.....	105
Tabla 24. Número de Peticiones del Usuario.	105
Tabla 25. Parámetros de medición de punto función.	106
Tabla 26. Punto función sin ajustar.	106
Tabla 27. Factores de Complejidad.	107
Tabla 28. Parámetros de medición de usabilidad.	110
Tabla 29. Usabilidad del sistema.	111

Tabla 30. Escala de valores de eficiencia.	112
Tabla 31. Valoración de la eficiencia del sistema.....	112
Tabla 32. Valores para la mantenibilidad.	114
Tabla 33. Resultados	116
Tabla 34. Caso de prueba de Ingreso al sistema.....	125
Tabla 35. Caso de prueba de registro de usuario.	126
Tabla 36. Coeficientes de a,b,c,d de COCOMO II.	117
Tabla 37. Ecuaciones de Cocomo II.	118
Tabla 38. Cálculo de Atributos FAE.	118

ÍNDICE DE FIGURAS

Figura 1. Organigrama ESSECAB.....	3
Figura 2. Modelo de análisis de requerimientos.	24
Figura 3. Modelo de Caso de Uso de la Ingeniería Web.	25
Figura 4. Modelo Contenido de la Ingeniería Web.....	26
Figura 5. Modelo Navegación de la Ingeniería Web.	28
Figura 6. Modelo de Presentación de la Ingeniería Web.	29
Figura 7. Flujo de MVC.....	32
Figura 8. División de calidad de Software.....	45
Figura 9. Diagrama de flujo.....	67
Figura 10. Grafo de flujo.	68
Figura 11. Matriz de Grafo.....	69
Figura 12. Diagrama de caso de uso general.....	76
Figura 13. Gestionar usuarios.....	77
Figura 14. Diagrama de caso de uso de administradores.....	79
Figura 15. Caso de uso agregar ocupación.....	80
Figura 16. Caso de uso – Gestionar Presupuesto.....	82
Figura 17. Caso de uso - habilitación de formulario.....	83
Figura 18. Caso de Uso - Seguimiento del cliente.....	85
Figura 19. Diseño Conceptual.	86
Figura 20. Modelo de navegación –administrador.....	87
Figura 21. Modelo de navegación – técnico.	88
Figura 22. Modelo de navegación – Usuario.	88
Figura 23. Modelo presentación administrativos.....	89
Figura 24. Modelo presentación usuarios.....	89
Figura 25. Modelo presentación clientes.	90
Figura 26. Página principal.	90
Figura 27. Inicio de información.....	91
Figura 28. Menú principal clientes.	91
Figura 29. Menú principal servicios.	92
Figura 30. Menú servicios eléctricos.....	92

Figura 31.	Menú proyectos.....	92
Figura 32.	Menú nosotros.....	93
Figura 33.	Contactos.....	94
Figura 34.	Inicio de sesión.....	94
Figura 35.	Menú principal administrador.....	95
Figura 36.	Roles y permisos.....	95
Figura 37.	Crear permisos.....	96
Figura 38.	Listado de permisos.....	96
Figura 39.	Crear rol.....	97
Figura 40.	Roles y permisos.....	97
Figura 41.	Menús administrativos.....	98
Figura 42.	Agregar administrativo.....	98
Figura 43.	Lista clientes.....	99
Figura 44.	Agregar cliente.....	99
Figura 45.	Lista de ocupaciones.....	100
Figura 46.	Registro de ocupación.....	100
Figura 47.	Lista de obreros.....	101
Figura 48.	Registro de obreros.....	101
Figura 49.	Lista de proyectos activos.....	102
Figura 50.	Lista de proyectos finalizados.....	102
Figura 51.	Lista de proyectos cancelados.....	103
Figura 52.	Registro de proyectos.....	103
Figura 53.	Prueba de caja blanca.....	123

CAPÍTULO I

1. MARCO PRELIMINAR

1.1 INTRODUCCIÓN

En la actualidad los sistemas de información se han convertido en un factor muy importante en el desarrollo de las empresas nacionales, debido a la gran cantidad de información que en ellos se procesa, el procesamiento automático de la información es de gran utilidad e importancia para las empresas. Además, los sistemas de información están basados en computadora, que es una herramienta importante para tomar decisiones oportunas, confiables y efectivas en la planificación y gestión de la tecnología para asegurar su éxito, reducir costos y riesgos

La Empresa de Servicios Eléctricos, Civiles Ambientales Bolivia S.R.L. lleva varios años al servicio de la construcción en Bolivia, realizando diseños arquitectónicos muy innovadores, ofreciendo servicios desde aprobación de planos en las alcaldías correspondientes, obra bruta y vaciados hasta la obra fina siempre bajo los altos estándares internacionales de calidad, también el servicio de armado de sistemas eléctricos, sin embargo al ser una constructora de alto prestigio y al haber crecido tanto, es importante que incluya el acceso a la información en el menor tiempo posible, ver el avance de obras que se tiene para tener un control constante, tener el conocimiento diario de la cantidad de las herramientas y materiales de obra para que de esta forma se pueda brindar un servicio eficiente a todos sus clientes a los cuales la empresa brinda sus servicios. Estos procesos al no haber sido automatizados provocan que la empresa y los clientes no tengan la misma información en el mismo tiempo.

El presente proyecto tiene como finalidad desarrollar e implementar un sistema web de administración y controla de proyectos de construcción en obras civiles para la empresa de servicios eléctricos, civiles ambientales Bolivia S.R.L, que permita a la empresa tener un mejor manejo de las obras a su cargo y poder coadyuvar la toma de decisiones entre los clientes, la empresa y el usuario, donde

además se tenga el acceso a toda la información concerniente a los proyectos de construcción en los distintos estados de avance, tiempos estipulados y visualizar los diferentes tipos de archivos, entre ellos, la plataforma Web como plan estratégico para el acceso rápido a recursos y servicios logrando un paso más a la modernización tecnológica.

1.2 ANTECEDENTES

1.2.1 Antecedentes Institucionales

La Empresa de Servicios Eléctricos, Civiles Ambientales Bolivia S.R.L. lleva varios años al servicio de la construcción en Bolivia, realizando diseños arquitectónicos muy innovadores, ofreciendo servicios desde aprobación de planos en las alcaldías correspondientes, obra bruta y vaciados hasta la obra fina siempre bajo los altos estándares internacionales de calidad, también el servicio de armado de sistemas eléctricos, sin embargo al ser una constructora de alto prestigio y al haber crecido tanto, se le hace una necesidad tener acceso a la información en el menor tiempo posible, ver el avance de obras que se tiene para tener un control constante, tener el conocimiento diario de la cantidad de las herramientas y materiales de obra para que de esta forma se pueda brindar un servicio eficiente a todos sus clientes a los cuales la empresa brinda sus servicios.

La Empresa de Servicios Eléctricos, Civiles Ambientales Bolivia S.R.L. lleva varios años construyendo en Bolivia, diseño y construcción de proyectos generales en electricidad, redes eléctricas, subestaciones eléctricas, líneas de transmisión puesto a tierra, construcción de edificios completos o de partes de edificio; obras de ingeniería civil, estudio y diseño de control de sistema de potencia, fibra óptica, instalación en baja tensión e instalaciones complementarias, venta de materiales eléctricos y materiales de construcción.

La empresa fue fundada en octubre de 2014 y comenzó ofreciendo servicios de consultoría en diseño final de proyectos en las áreas civil, eléctrica y ambiental. Con el tiempo, realizó instalaciones eléctricas en baja y media tensión, así como en medios de transmisión y en la parte civil realizó estudios de puentes y

construcción de centros de salud, casas culturales, unidades educativas, sistemas de riegos y micro riegos. También en la parte ambiental realizo en la obtención de licencias para diferentes proyectos tanto en el ámbito eléctrico como civil.

1.2.1.1 Visión

Ser reconocidos como líderes en la prestación de servicios integrales en las áreas civil, ambiental y eléctrica, aportando soluciones innovadoras y sostenibles que contribuyan al desarrollo y bienestar de la sociedad.

1.2.1.2 Misión

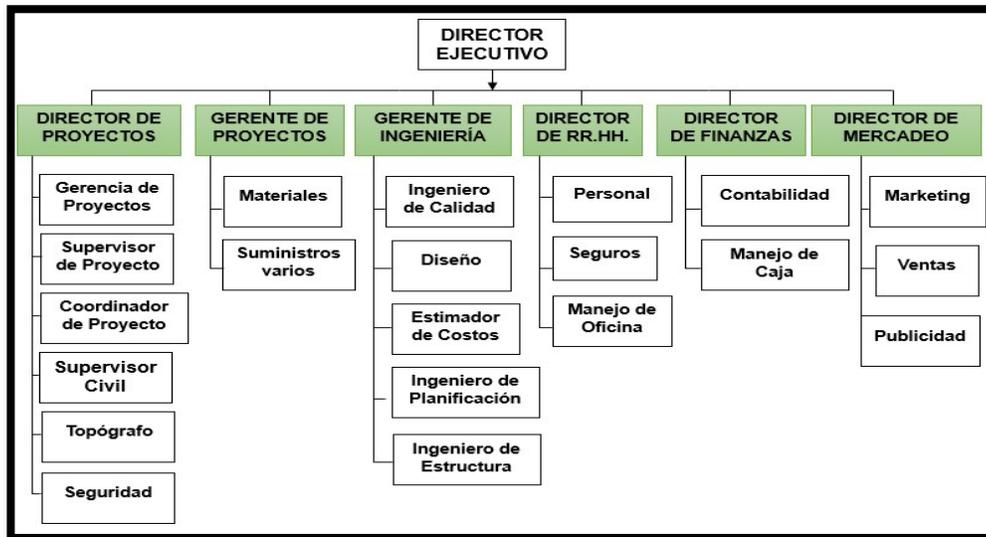
Brindar soluciones integrales y personalizadas en consultoría, diseño y construcción de proyectos en las áreas civil, ambiental y eléctrica, mediante el uso de tecnologías avanzadas, la optimización de recursos y el compromiso con la calidad, la seguridad y el medio ambiente.

1.2.1.3 Organigrama

A continuación, se muestra la organización de la empresa de acuerdo a las unidades de gerencia y cargos.

Figura 1

Organigrama ESSECAB.



Nota. Esquema de cargos de la Empresa de Servicios Eléctricos, Civiles

Ambientales Bolivia S.R.L.

1.2.2 Antecedentes afines al Proyecto de Grado

Haciendo referencia a los trabajos realizados a nivel internacional, nacional se pueden citar a las siguientes:

1.2.2.1 Antecedentes Nacionales

- Francisco Ramos Porto (2016), “SISTEMA WEB DE ADMINISTRACIÓN DE LA EJECUCIÓN DE PROYECTOS DE CONSTRUCCIÓN CIVIL PARA LA EMPRESA CONSTRUCTORA “MULRAHU S.R.L.” El presente trabajo precisa los procesos, procedimientos, herramientas tecnológicas y metodológicas en desarrollo del proyecto, el objetivo de realizar el proyecto es administrar la ejecución de los proyectos, registrar y controlar los avances de las construcciones civiles, generar informes financieros e informes de ejecución de proyectos además de generar propuestas de ejecución de actividades del director de obra según el tiempo y beneficio.
- Johnny Chavez Quispe (2014), “SISTEMA WEB DE GESTIÓN DE PROYECTOS “CONSTRUCTORA Y MANTENIMIENTO ELÉCTRICO ATWOOD S.R.L.” el presente proyecto tiene el objetivo el mejorar el funcionamiento en el proceso de la gestión de proyectos además de atraer nuevos clientes y ampliar el mercado de las construcciones, el sistema podrá generar formularios, informes, establecer un cronograma y otras funcionalidades que serán de beneficio a la constructora. La metodología empleada en el desarrollo del proyecto es ágil XP, en cuanto a las herramientas de base de datos se utilizó en gestor MySQL, lenguajes de programación PHP, Ajax, y en cuanto a la calidad empleando la ISO 9126.

1.2.2.2 Antecedentes Internacionales

- Diego Andrés Contreras Rivas, Bryan Alexander Lara Ávila (2015), “CONTROL DE PRESUPUESTO DE OBRAS CIVILES BASADO EN DISEÑO WEB ORIENTADO A LAS EMPRESAS CONSTRUCTORAS”. La presente investigación fue desarrollada para el control del presupuesto de las obras

civiles con el objetivo de cumplir la accesibilidad de la información, las herramientas utilizadas para el desarrollo web fue Asp.net siendo este un framework desarrollado y comercializado por Microsoft, servicios de XML, librerías de código abierto, el sistema operativo Windows para la implementación del proyecto y así cumplir con los objetivos propuestos.

- Marlon Yesid Tunarosa Muñoz, Elkin Mauricio Ávila Bonilla (2020), “SISTEMA DE INFORMACIÓN WEB PARA EL CONTROL Y SUPERVISIÓN DE OBRAS CIVILES EN EL CARMEN DE APICALÁ, PARA LA EMPRESA INSTALACIONES HIDROSANITARIAS Y DE GAS JATS S.A.S. “PLUMBINGSOFT”, el trabajo desarrollado tiene la objetivo de diseñar, desarrollar e implementar un sistemas de información web que realice los controles y supervise las tareas de la obra civil para la reducción de los costos y controlar el tiempo de entrega de las obras, la metodología aplicada fue Extrema Programming por su adaptabilidad a los cambios que se presentan en el transcurso del proyecto, la implementación de PlumbingSoft cumplió con los requerimientos de controlar y supervisar un obra civil el cual evidencia el desarrollo y culminación de las tareas asignadas.
- Alex PatricioTroncoso Contreras (2017), “DESARROLLO DE UNA APLICACIÓN WEB PARA LA GESTIÓN DE EMPRESAS CONSTRUCTORAS”, el presente proyecto describe el desarrollo de una aplicación web para realizar la gestión administrativa de empresas dedicados al rubro de las construcciones, a falta de la implementación de sistemas se vio el registro de la información en planillas de cálculo y archivos de texto, visto la necesidad de una aplicación web se implementó este proyecto y de esta forma poder centralizar la información y permitir el acceso de la información desde cualquier dispositivo con el fin de apoyar la toma de decisiones.

1.3 PLANTEAMIENTO DEL PROBLEMA

La Empresa de Servicios Eléctricos, Civiles Ambientales Bolivia S.R.L., está experimentando un crecimiento en los servicios de consultoría en construcción de obras eléctricas, civiles y ambientales que realiza, lo cual provoca que la administración y control de sus documentos en libros de registros, la información acerca de los equipos, materiales y herramientas de construcción, así también la información del personal sea escasa y también son manejados de manera manual y estos no están de manera organizada, así también mejorar los procesos administrativos, como el control del personal, control del avance de manera efectiva, control de los equipos y herramientas, control del material, control de todos los procesos que se tiene en una obra.

Luego de realizar un análisis de problemas con la consecuente construcción del árbol de problemas (ver anexo A), el cual no permite identificar que la constructora no cuenta con una sistematización de todo el movimiento de la información en relación a las convocatorias a licitaciones tanto nacionales como internacionales, no se cuenta con los registros de actas, controles e informes lo cual afecta en la organización y eficiencia de los servicios prestados por la Empresa de Servicios Eléctricos, Civiles Ambientales Bolivia S.R.L.

1.3.1 Problema Principal

La constructora ESECAB no cuenta con un sistema dedicado al manejo de registro de información, informes del estado de los proyectos, este se realiza de manera manual lo cual genera que la información se maneje de manera dispersa la información y organización de los proyectos, generando que no se pueda proporcionar información del estado de proyectos.

1.3.2 Problemas Secundarios

- No se obtiene información adecuada y a tiempo del estado de los proyectos realizados y en proceso, provocando confusiones y mala información de los proyectos.

- No se cuenta con los registros de los trabajadores de la empresa se realizan en cuadernos con hojas de días trabajados, la información que se obtiene es poca fiable.
- La pérdida de equipos, materiales y herramientas, generan pérdidas económicas de la empresa.
- Las licitaciones y convocatorias no son de fácil acceso, provoca la falta de proyectos de la empresa.
- No se tiene un adecuado control y organización de archivos de los proyectos lo que ocasiona demora en la consulta de información.
- No se cuenta con acceso a documentaciones respecto a controles y actas los cuales afectan en el avance de la obra.
- Falta de información estadística respecto a la cantidad de obras realizadas, tipos de servicios realizados etc. Que perjudican respecto a las metas y objetivos por cumplir en la constructora.

1.3.3 Formulación del Problema

¿Cómo optimizar la administración y control de proyectos en la Empresa de Servicios Eléctricos Civiles Ambientales Bolivia S.R.L.?

1.4 OBJETIVOS

El objetivo del presente proyecto de grado nace con una respuesta a las necesidades de la institución, luego de un análisis de los problemas y su relación causa – efecto se realizó el árbol de objetivos (ver anexo B).

1.4.1 Objetivo General

Desarrollar e implementar una plataforma web administrativa para realizar el control a los proyectos de construcción de obras civiles en la Empresa de Servicios Eléctricos Civiles Ambientales Bolivia S.R.L., como herramienta para que facilite

la gestión de información, seguimiento de tareas, comunicación entre equipos y supervisión de avance.

1.4.2 Objetivos Específicos

- Diseñar interfaz amigable e intuitivo para el usuario, con la que pueda interactuar y obtener los procesos e información requerida.
- Centralizar la información de inventario en una base de datos para el correcto registro de administrativos, proyectos y licitaciones de la constructora.
- Establecer la información actualizada de los materiales de construcción, de esta forma se evitará pérdidas y se tendrá un mejor control.
- Mostrar toda la información sobre las licitaciones públicas y privadas, en una página Web entendible y expícito para que el usuario tenga acceso estas convocatorias.
- Registrar y organizar todos los archivos de la constructora, entonces el usuario podrá acceder a la documentación de acuerdo a la obra especificada.
- Organizar y mostrar los datos estadísticos más importantes de la constructora, con el fin de mejorar y tener presente las obras como los procesos administrativos de la empresa.

1.5 JUSTIFICACIÓN

El desarrollo de la plataforma web de seguimiento y control de proceso de construcción, que mejora el desarrollo de los proyectos de la Empresa de Servicios Eléctricos Civiles Ambientales Bolivia S.R.L., se justifica principalmente en el manejo eficiente de la información, además se justifica de manera técnica, económica y social de la siguiente forma:

1.5.1 Justificación Técnica

Se justifica técnicamente porque la empresa ESECAB SRL. cuenta con las herramientas tecnológicas de software y hardware requeridas para el desarrollo e

implementación del sistema Web de seguimiento y control de proceso de construcción.

El desarrollo de la plataforma web utilizará componentes técnicos como computadoras, internet y lenguajes de programación de software libre, que impulsaran a un flujo de información eficiente y la innovación tecnológica.

1.5.2 Justificación Económica

Se justifica económicamente ya que la implementación de la plataforma tendrá un flujo económico y control eficiente de la información de acuerdo a los requerimientos de la Empresa de Servicios Eléctricos, Civiles Ambientales Bolivia S.R.L., reduciendo la pérdida de tiempo en procesos de registro y consultas requeridas.

Los proyectos serán mejor administrados evitando información confusa, los controles y las consultas de la gerencia serán atendidas eficientemente para la toma de decisiones de la empresa. El estado de los equipos y herramientas de la empresa tendrán un mejor control evitando pérdidas.

El sistema será desarrollado bajo la premisa de software libre, que implica la no distribución de gastos en licencias de uso.

1.5.3 Justificación Social

Se justifica de manera social porque brinda información segura, oportuna y actualizada de los proyectos que se realiza en la empresa, mejorando la calidad de los proyectos de construcción.

El sistema es útil beneficioso en el ámbito social el cual beneficia a la empresa principalmente a la gerencia general, la cual tiene la función de planificación, coordinación, dirección y ejecución de los proyectos, proveerá información rápida y actualizada del personal, proyectos, equipos y herramientas de la empresa.

1.6 METODOLOGÍA

1.6.1 Metodología UWE

Se aplicará para el desarrollo de la plataforma web el uso de la metodología de desarrollo UWE (Unified Modeling Language). (Ludwig, 2014)

Para su implementación se debe seguir las siguientes etapas:

Análisis de Requerimientos plasma los requisitos funcionales de la aplicación web mediante un modelo de casos de Uso.

- Modelo de Contenido mediante un modelo de clase se define los conceptos a detalle que relación en el sistema.
- Modelo de Navegación en esta etapa se representa la navegación de los objetos dentro de la aplicación un conjunto de estructuras como son índices menús y consultas
- Modelo de presentación representa las interfaces de usuario por medio de vistas abstractas
- Modelo de proceso representa el aspecto que tienen las actividades que se conectan con cada clase de proceso.

1.6.2 Métricas de Calidad

La estimación del costo para el desarrollo del software se puede aplicar el método Cocomo II que está basado en cálculo del costo mediante el tiempo esfuerzo y el número de líneas de código del software, Cocomo II consta de tres modelos composición de aplicación, diseño temprano y Post Arquitectura. (Miss Juarez , 2020, pág. 1)

- Modelo de composición de aplicación que es utilizado durante la ingeniería de software, prototipado de interfaces de usuario, interacción del sistema y del software, evaluación del rendimiento y la evaluación del crecimiento de la tecnología.

- Modelo de diseño temprano que se aplica cuando se ha establecido la arquitectura del software.
- Modelo Post Arquitectura aplicado durante la construcción del software.

Estos modelos se adaptan a las necesidades como al tipo y la cantidad de la información. (Miss Juarez , 2020, pág. 25)

1.6.3 Seguridad

La norma ISO-27000 es un estándar que define ciertos requisitos para un sistema de gestión de la seguridad de la información (SGSI) que tiene el objetivo de garantizar el control de la seguridad adecuada protegiendo la información de cualquier empresa, institución, organización, etc. Ofreciendo una cantidad de recomendaciones de seguridad. (Baena, Joel Coronado, & Mendoza M., 2019)

Entre sus características de las normas ISO – 27000 se basan en:

- Confidencialidad
- Integridad
- Disponibilidad

1.6.4 Pruebas de Software

Las pruebas son muy importantes en la fase de implementación es por eso que es muy necesario realizarlas se propone las siguientes pruebas:

1.6.4.1 Caja Negra

Las pruebas de caja negra, también llamadas pruebas de comportamiento, se enfocan en los requerimientos funcionales del software; es decir, las técnicas de prueba de caja negra le permiten derivar conjuntos de condiciones de entrada que revisarán por completo todos los requerimientos funcionales para un programa. Las pruebas de caja negra no son una alternativa para las técnicas de caja blanca. En vez de ello, es un enfoque complementario que es probable que descubra una clase de errores diferente que los métodos de caja blanca. Las pruebas de caja negra intentan encontrar errores en las categorías siguientes:

funciones incorrectas o faltantes, errores de interfaz, errores en las estructuras de datos o en el acceso a bases de datos externas, errores de comportamiento o rendimiento y errores de inicialización y terminación. (Pressman, 2010, pág. 423)

1.6.4.2 Caja Blanca

La prueba de caja blanca, en ocasiones llamada prueba de caja de vidrio, es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba de caja blanca, puede derivar casos de prueba que: garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez, revisen todas las decisiones lógicas en sus lados verdadero y falso, ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y revisen estructuras de datos internas para garantizar su validez. (Pressman, 2010, pág. 414)

1.6.4.3 Pruebas de Estrés

Las pruebas de estrés se diseñan para simular situaciones de carga del mundo real. Conforme aumenta el número de usuarios simultáneos de la aplicación o el número de transacciones en línea o la cantidad de datos descargados o subidos (Pressman, 2010, pág. 473), estas pruebas de rendimiento ayudarán a responder las siguientes preguntas:

- ¿El tiempo de respuesta del servidor se degrada a un punto donde es apreciable e inaceptable?
- ¿En qué punto (en términos de usuarios, transacciones o carga de datos) el rendimiento se vuelve inaceptable?
- ¿Qué componentes del sistema son responsables de la degradación del rendimiento?

- ¿Cuál es el tiempo de respuesta promedio para los usuarios bajo diversas condiciones de carga?
- ¿La degradación del rendimiento tiene impacto sobre la seguridad del sistema?
- ¿La confiabilidad o precisión de la aplicación resulta afectada conforme crece la carga sobre el sistema?
- ¿Qué sucede cuando se aplican cargas que son mayores que la capacidad máxima del servidor?
- ¿La degradación del rendimiento tiene impacto sobre los ingresos de la compañía?

1.7 HERRAMIENTAS

- **Xampp**

Es un acrónimo de X indicando la disponibilidad de distintos sistemas operativos, Apache, MariaDB, PHP y Perl. Es una distribución de Apache de software libre que actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas, siendo su principal objetivo la gestión de base de datos (MySQL y MariaDB), gestión del servidor Apache además de la interpretación del lenguaje de Script (PHP Y Perl). (Valencia Pavón , 2012-2013, pág. 75)

- **Laravel**

Es un framework de código abierto del Modelo Vista Controlador (MVC) ofreciendo un conjunto de herramientas y recursos que tiene por finalidad de desarrollar aplicaciones y sistemas web con PHP de forma sencilla y ordenada, este framework fue desarrollado por Taylor Otwell y un grupo de desarrolladores contribuidores. (Arango Cordoba , 2016, pág. 17)

- **Mysql**

Es un gestor de bases de datos que se encuentra dentro de la categoría de los programas open-source, está basado en el lenguaje de consultas estructurados

(SQL) y a su vez es un lenguaje de base de datos normalizados que puede crear consultas SQL de paso para manipular bases de datos remotas cliente servidor mediante los comandos, clausulas, operadores y funciones de agregado y mediante sus combinaciones realizar instrucciones de creación, actualización y manipulación de la base de datos. (Harb Hoecker, pág. 3)

- **Ajax**

Es el asíncrono de JavaScript y XML que es un conjunto de técnicas de desarrollo web que permite generar aplicaciones web de manera interactiva que accede la comunicación entre el servidor web y el navegador del usuario, a su vez permite que el funcionamiento del sistema web sea veloz, asíncrona procesando las solicitudes del servidor en segundo plano. (Flores Fernandez, 2010, pág. 5)

- **JavaScript**

Es un lenguaje de programación muy conocido. Entre otras funciones, gestiona el contenido dinámico de un sitio web y permite la interacción dinámica del usuario. XML es otra variante de un lenguaje de marcado como HTML, como lo sugiere su nombre: extensible Markup Language. Mientras HTML está diseñado para mostrar datos, XML está diseñado para contener y transportar datos.

Tanto JavaScript como XML funcionan de forma asíncrona en AJAX. Como resultado, cualquier aplicación web que use AJAX puede enviar y recuperar datos del servidor sin la necesidad de volver a cargar toda la página. (B., 2023)

1.8 LÍMITES Y ALCANCES

1.8.1 Límites

- El software que se implementará solo estará disponible para departamento de planificación y administración.
- La información almacenada en la base de datos, solo registrará los proyectos que realiza la Empresa de Servicios Eléctricos Civiles Ambientales Bolivia S.R.L.

- El módulo control del personal se aplicará a la asistencia y planillas de sueldo y salarios del personal de la empresa.

1.8.2 Alcances

La plataforma web, mejora el registro, control y seguimiento en los procesos administrativos que realiza la constructora ESECAB SRL. a continuación, se detalla en módulos las actividades:

- Se registrarán a los usuarios que ingresarán al sistema y se les asignan el rol de acceso.
- Se registrarán a los empleados de la empresa para el control de asistencia.
- Se agregarán nuevos proyectos de construcción para su seguimiento por parte de la empresa.
- Se guardará la documentación administrativa, los proyectos realizados y en proceso de seguimiento en la empresa.
- Se registrarán la agrupación de los distintos archivos entre planillas de asistencia, de sueldo y salario en la base de datos.
- Se realizará las consultas a la plataforma estatal del SICOES esto para las convocatorias.
- Se realizará gráficos estadísticos acerca de los proyectos realizados y en proceso.

1.9 APORTES

El sistema Web de seguimiento y control de proceso de construcción aportara a la empresa ESECAB de las siguientes maneras:

La apertura de proyectos por parte de usuarios registrados y con su respectivo permiso de acceso.

El seguimiento de los diferentes proyectos de la empresa, para la consulta de la información en el estado en que se encuentra cada uno.

El control de los equipos y herramientas de la empresa, con especificaciones del estado, en que proyecta se encuentra y la ubicación actual en la que se encuentra a cargo del personal

El marco teórico para una plataforma web de administración y control de proyectos de construcción en obras civiles abarcaría los siguientes aspectos.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. INTRODUCCIÓN

En este capítulo se definirán los conceptos y definiciones como base para poder desarrollar el presente proyecto, con bases teóricas de diferentes fuentes donde se extrajo la información.

2.2. PLATAFORMA WEB

Según Sedera (2016, pág.8), indica que “Las plataformas web son una propuesta flexible, individualizada e interactiva, con el uso y combinación de diversos materiales, formatos y soportes de fácil e inmediata actualización”.

Plataforma web es una arquitectura tecnológica que permite el desarrollo de sus propias funcionalidades informáticas y permite la integración de las plataformas tecnológicas de información, informática y conectividad disponibles para una mejor organización en la web (Da Silva & Nuñez, 2018, pág. 9).

2.3. ADMINISTRACIÓN DE PROYECTOS

La administración de proyectos es la forma de planear, organizar, dirigir y controlar una serie de actividades realizadas por un grupo de personas que tienen un objetivo específico; el cual puede ser (crear, diseñar, elaborar, mejorar, analizar, etc.) un problema o cosa (Rodríguez, 2002).

Baker (2009, pág.2), menciona que: “La administración de proyectos es el proceso de combinar sistemas, técnicas y personas para completar un proyecto dentro de las metas establecidas de tiempo, presupuesto y calidad”.

2.4. CONTROL DE PROYECTOS

Los controles del proyecto son un conjunto de procesos que se implementan para comprender e incidir en los tiempos y el dinero necesarios para un proyecto. Cada control del proyecto se centra en una parte específica del plan del proyecto, como el cronograma, los recursos o los riesgos potenciales. (Asana, 2013, pág. 11)

2.5. PROYECTOS DE CONSTRUCCIÓN

Es el documento, o conjunto de documentos, que definen el diseño de una obra civil, y sobre el cual se desarrolla el trabajo de los arquitectos, ingenieros y proyectistas de distintas especialidades.

Neve (2019, pág.12), menciona que: “Un proyecto de construcción, ya sea de arquitectura, ingeniería civil u otra especialidad es un conjunto de documentos que definen de forma precisa la /obra antes de que se lleve a cabo”.

2.6. WEB 2.0.

La web 2.0 es un concepto que empezó a utilizarse en el año 2003. Su aparición se debió al auge de diversas aplicaciones como los blogs o las redes sociales que permitieron que los usuarios dejaran de ser mejor sujetos pasivos para adoptar un rol mucho más activo y dinámico. Sin embargo, a pesar de que hace años que existe este término.

En pocas palabras, el concepto de web 2.0 es sinónimo de aplicaciones web que permiten a los usuarios compartir información en Internet. En su momento, cuando este término se puso de moda, una de las plataformas preferidas era Blogger. Posteriormente, Facebook se convirtió en la primera red social que no solo permitía compartir información, sino la posibilidad de interactuar de una forma más inmediata. (Gardey, 2021)

2.6.1. Características de la Web 2.0

Es importante tener en cuenta que no existe una definición precisa de Web 2.0, aunque es posible aproximarse a ella estableciendo ciertos parámetros.

- Una página web que se limita a mostrar información y que ni siquiera se actualiza, forma parte de la generación 1.0. En cambio, cuando las páginas ofrecen un nivel considerable de interacción y se actualizan con los aportes de los usuarios, se habla de Web 2.0.
- Cabe mencionar que las diferencias entre la primera y la segunda era de la Web no se basan en un cambio a nivel tecnológico en los servidores, aunque

naturalmente se ha dado un considerable avance en el hardware; es el enfoque de la Red, los objetivos y la forma en la que los usuarios comenzaron a percibir la información en línea lo que caracteriza este renacer, que tuvo lugar silenciosa pero velozmente, a comienzos del nuevo milenio.

- Algunas de las principales características de la web 2.0 son los servicios de redes sociales, las plataformas de auto publicación (por ejemplo, las herramientas de WordPress de creación de blogs y sitios web), las etiquetas (funcionalidad que permite añadir metadatos a páginas web o contenidos multimedia), el botón «me gusta» y la calificación de contenidos (que permite al usuario expresar su opinión sobre el contenido) y los marcadores sociales.

Hasta entonces, Internet era un universo de datos mayoritariamente estáticos, una fuente de consulta revolucionaria que atraía a millones de personas a contemplarla pasivamente. Si bien los foros y el chat datan de la Web 1.0, éstos se encontraban bien diferenciados de los sitios tradicionales (tal y como ocurre en la actualidad); navegar era similar a visitar un gran centro comercial, con infinidad de tiendas, en las cuales era posible comprar productos, pero no alterar sus vidrieras. (Wikipedia, 2021)

2.6.2. Usos de la Web 2.0

La Web 2.0 se caracteriza principalmente por la participación del usuario como contribuidor activo y no solo como espectador de los contenidos de la Web (usuario pasivo). (Wikipedia, 2021)

Esto queda reflejado en aspectos como:

- El auge de los blogs.
- El auge de las redes sociales.
- Las webs creadas por los usuarios, usando plataformas de auto-edición.
- El contenido agregado por los usuarios como valor clave de la Web.

- El etiquetado colectivo (folcsonomía, marcadores sociales...).
- La importancia de la larga cola.
- La beta perpetua: la Web 2.0 se inventa permanentemente.
- Aplicaciones web dinámicas.
- La World Wide Web como plataforma.

2.7. METODOLOGÍA INGENIERÍA WEB BASADA EN UML (UWE)

La Ingeniería web basada en UML es una metodología basada en el Proceso Unificado y UML (Lenguaje Unificado de Modelado) para el desarrollo de aplicaciones Web, cubre todo el ciclo de vida de las aplicaciones Web es un lenguaje y una metodología adecuada para modelar el funcionamiento actual de la empresa. (SlideShare, 2015)

UWE es un método de ingeniería del software para el desarrollo de aplicaciones web basado en UML. Cualquier tipo de diagrama UML puede ser usado, porque UWE es una extensión de UML.

El enfoque UWE proporciona una notación específica de dominio, un proceso de desarrollo basado en modelos y soporte de herramientas para la ingeniería de aplicaciones web. La característica de UWE es el hecho de ser un enfoque basado en estándares que no se limita al uso de la UML "lingua franca", sino que también utiliza XML como formato de intercambio de modelos, MOF1 para el metamodelado, los principios basados en modelos del enfoque MDA, el lenguaje de transformación de modelos QVT2 y XML.

Las principales razones para utilizar los mecanismos de extensión de la UML en lugar de una técnica de modelado patentada es la aceptación de la UML en el desarrollo de sistemas de software, la flexibilidad para la definición de un lenguaje de modelado específico de dominio web: un llamado perfil UML y de un amplio

soporte de modelado visual por parte de las todas las herramientas existentes de UML CASE. (UWE, 2016)

2.7.1. Características

La metodología UWE define vistas especiales representadas gráficamente por diagramas UML, tales como el análisis de requerimientos, diseño conceptual modelo de navegación y el modelo de presentación. UWE no limita el número de diagramas posibles de una aplicación.

- Es una metodología orientada a objetos, iterativa e incremental basada en UML.
- Se basa también en el proceso de desarrollo del software unificado.
- Proporciona un diseño sistemático y una generación semiautomática en las aplicaciones web a través de un Framework de publicación XML.
- UWE define su propio perfil UML en el cual se definen todos los elementos necesarios para modelar los diferentes aspectos de una aplicación web que son la presentación y la navegación entre otros.
- En esta metodología se proponen 2 tipos de diagramas para el modelado de la navegación que son: El modelo de espacio , el cual se definen todos los caminos navegacionales, es decir todas aquellas asociaciones de navegación directa entre los distintos objetos de la aplicación más bien conocida como clases de navegación, y el segundo modelo de estructura de navegación el cual se define la estructura de acceso que son utilizadas en la navegación, es decir todo aquello referente a menús, índices y demás.

2.7.2. Modelos de la metodología UWE

El método UWE consiste en la construcción de seis modelos de análisis y diseño. Dicha construcción se realiza dentro del marco de un proceso de diseño iterativo e incremental. Las actividades de modelado abarcan: el análisis de requerimientos,

diseño conceptual, modelo de usuario, diseño de la navegación, de la presentación y diseño de la adaptación. (Thewolf, 2015)

UWE proporciona guías para la construcción de modelos de forma sistemática, enfocándose en la personalización y en estudio de casos de uso.

Los principales artefactos que produce el método de diseño de UWE son los siguientes:

- Un Modelo de Requerimientos que captura los requerimientos del sistema.
- Un Modelo Conceptual para el contenido (modelo de contenido).
- Un Modelo de Usuario.
- Un Modelo de Navegación que comprende la estructura de la navegación.
- Un Modelo de Presentación que abarca modelos estáticos y dinámicos (modelo de estructura de la presentación, modelo del flujo de la presentación).
- Un 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 (Sanchez, 2017)

2.7.2.1. Modelo de análisis de requerimientos

Cumple un papel primordial en el proceso de producción de software, ya que se enfoca un área fundamental, su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, de esta manera, se pretende minimizar los problemas relacionados por la mala gestión de requerimientos en el desarrollo de sistemas. Sirve como base sólida en el proceso de desarrollo de software, por lo que antes de pasar a aspectos referentes a la administración adecuada de los requerimientos, es importante definir los requerimientos y cuáles serán las características deseables que debería de tener (Chavez, 2012)

2.7.2.1 Requerimientos Funcionales

En general, los requerimientos funcionales describen acciones específicas que el ingeniero de software debe ser capaz de realizar durante el desarrollo de software. Los requerimientos funcionales a menudo se dividen en reglas de negocio y casos de uso. Las reglas de negocio son declaraciones de alto nivel que definen lo que un sistema debe hacer, mientras que los casos de uso son descripciones más detalladas de cómo debe funcionar el sistema.

Algunos de los requerimientos más comunes en virtud de él incluyen:

- Las características y funcionalidad deseadas del producto
- Plataformas para desarrollar aplicaciones.
- Especificaciones de diseño en términos de tema, colores y fuentes
- Funcionalidad de back-end: integración API y bases de datos.
- Plazos de finalización.

2.7.2.2 Requerimientos no funcionales

Los requerimientos no funcionales describen características específicas que el software debe poseer durante el desarrollo de la aplicación.

Por lo general, se dividen en tres categorías: rendimiento, seguridad y calidad.

- **Requerimientos de rendimiento**

Los requerimientos de rendimiento suelen dividirse en dos categorías: tiempo de respuesta y rendimiento. El tiempo de respuesta es el tiempo que tarda un sistema en responder a la solicitud de un usuario, mientras que el rendimiento es el número de solicitudes que un sistema puede manejar. Son más críticos para los sistemas interactivos, como las aplicaciones de escritorio y los sitios web, donde los usuarios esperan respuestas inmediatas a sus acciones.

- **Requerimientos de seguridad**

Los requerimientos de seguridad especifican las medidas que un sistema debe tomar para proteger los datos del acceso no autorizado. En algunos casos, los requerimientos de seguridad también pueden especificar el nivel de protección requerido, como confidencial o de alto secreto. Implica autenticación, autorización y cifrado.

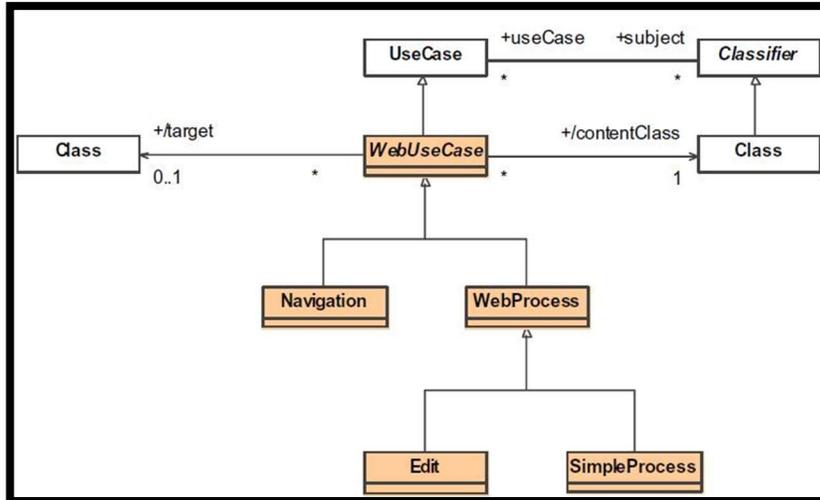
- **Requerimientos de calidad**

Especifica el nivel de calidad que debe cumplir un sistema. En algunos casos, los requerimientos de calidad también pueden especificar los métodos utilizados para medir la calidad, como la densidad de defectos o la satisfacción del cliente. Los requerimientos de calidad son generalmente cuatro medidas de calidad: conformidad, usabilidad, confiabilidad y mantenibilidad.

Los requerimientos pueden ser documentados en diferentes niveles de detalle, para este caso, UWE propone dos niveles de granularidad. En primera instancia se deben describir detalladamente las funcionalidades del sistema, las cuales son modeladas con casos de uso UML. Como segundo paso, se debe elaborar una descripción de los casos de uso más detallada, por ejemplo, realizando diagramas de actividad UML donde se delimiten las responsabilidades y acciones de los actores involucrados.

Figura 1

Modelo de análisis de requerimientos.



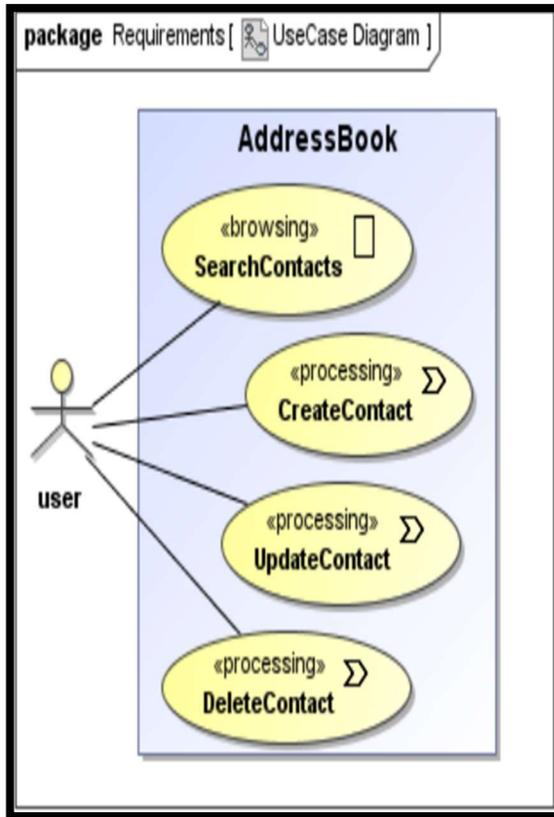
Nota. La figura muestra el modelo de análisis de requerimiento para el desarrollo del prototipo. (Vargas, 2017)

2.7.2.2. Modelo de Casos de Uso

Un diagrama de casos de uso muestra la relación entre los actores y los casos de uso del sistema. Representa la funcionalidad que ofrece el sistema en lo que se refiere a su interacción externa.

Figura 2

Modelo de Caso de Uso de la Ingeniería Web.



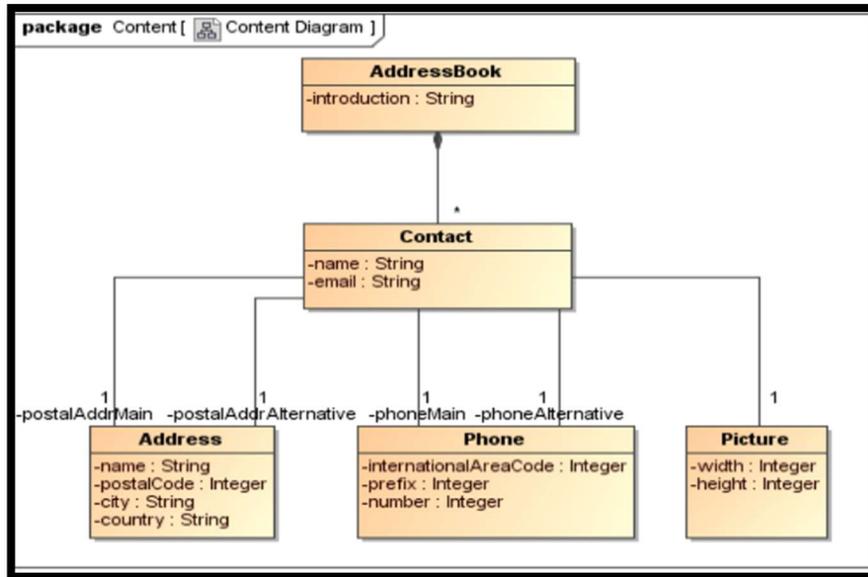
Nota. La figura muestra el formato del modelo del Caso de Uso para su elaboración. (Vargas, 2017)

2.7.2.3. Modelo contenido

El modelo conceptual de un sistema software constituye una abstracción externa que describe mediante diagramas y notaciones con distinto grado de formalidad el conocimiento que debe poseer una persona acerca de un sistema, conocimiento que se encuentra almacenado en la Memoria a Largo Plazo.

Figura 3

Modelo Contenido de la Ingeniería Web.



Nota. La figura muestra el formato del modelo de contenido para su elaboración. (Vargas, 2017)

2.7.2.4. Modelo de Navegación

El modelo de estructura de navegación define la estructura de nodos y links de una WebApp mostrando cómo se puede realizar la navegación utilizando elementos de acceso tales como índices, visitas guiadas, consultas y menús.

Los elementos de modelado son:

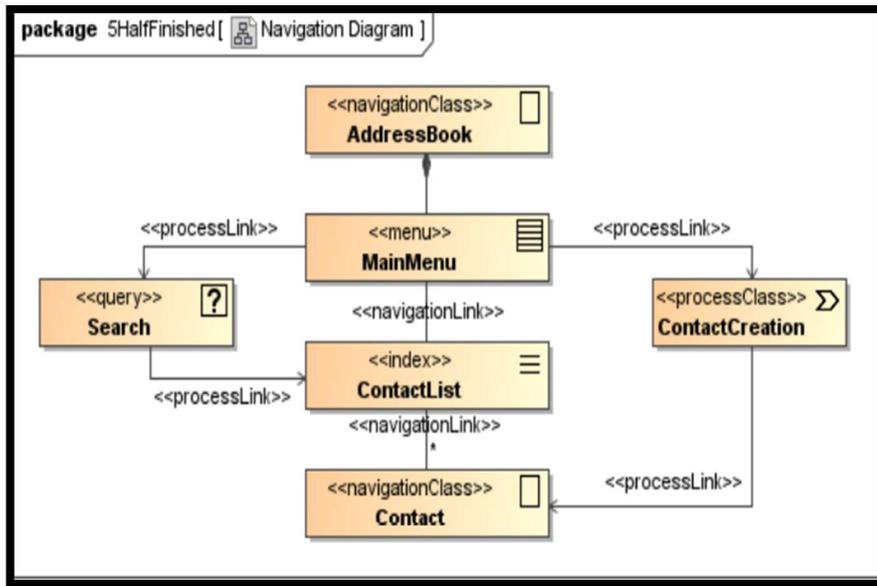
- **Clases de navegación**, que se denotan con (0), representan los nodos navegables de la estructura de hipertexto.
- **Links de navegación**, que muestran el vínculo directo entre las clases de navegación.
- **Caminos de navegación alternativos**, los cuales son visualizados con el estereotipo <<menu>> ().
- **Primitivas de acceso**, las cuales se utilizan ya sea para llegar a múltiples instancias de una clase de navegación (<<index>> o <<guided tour>>) o para seleccionar ítems (<<query>>).

- **Clases de procesos ()**, las cuales modelan los puntos de entrada y de salida de los procesos de negocio. Cada clase de proceso está asociada a un caso de uso de proceso.
- **Links de procesos**, que representan el vínculo entre las clases de proceso y de navegación.

El modelo de estructura de navegación se representa mediante diagramas de clases UML estereotipados con las clases de navegación y procesos, menús y primitivas de acceso y así también los links de navegación y proceso.

Figura 4.

Modelo Navegación de la Ingeniería Web.



Nota. La figura muestra el modelo navegación y su formato de desarrollo. (Vargas, 2017)

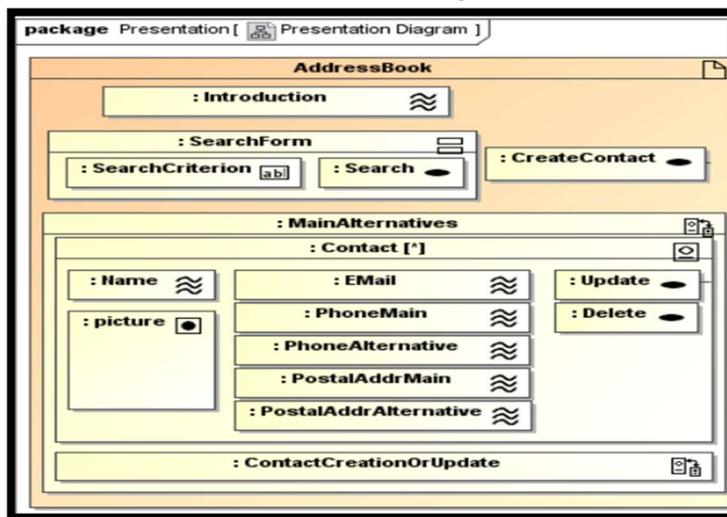
2.7.2.5. Modelo de Presentación

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

- **Clases de presentación**, las cuales se basan directamente en los nodos del modelo de navegación. Una clase de presentación () está compuesta por elementos de UI tales como, texto (<<text>>), vinculo (<<anchor>>), botón (<<button>>), imagen (<<image>>), formulario (<<form>>), y colección de vínculos (<<anchored collection>>)
- **Páginas web (<<page>>)**, que se utilizan para modelar la información proveniente de varios nodos de navegación y que se presentan en una misma página web.
- **Grupo de presentación (<<presentation group>>)**, el cual es un contenedor de clases de presentación, y a su vez de otros grupos de presentación.

Figura 5

Modelo de Presentación de la Ingeniería Web.



Nota. La figura muestra el modelo presentación y su prototipo para su desarrollo. (Vargas, 2017)

2.8. ARQUITECTURA DE SOFTWARE (MVC)

MVC es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos.

MVC se usa inicialmente en sistemas donde se requiere el uso de interfaces de usuario, aunque en la práctica el mismo patrón de arquitectura se puede utilizar para distintos tipos de aplicaciones. Surge de la necesidad de crear software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos.

MVC es un "invento" que ya tiene varias décadas y fue presentado incluso antes de la aparición de la Web. No obstante, en los últimos años ha ganado mucha fuerza y seguidores gracias a la aparición de numerosos frameworks de desarrollo web que utilizan el patrón MVC como modelo para la arquitectura de las aplicaciones web. (Desarrollo web, 2020)

El modelo MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

2.8.1. Modelo

Es la capa donde se trabaja con los datos, por tanto, contendrá mecanismos para acceder a la información y también para actualizar su estado. Los datos los tendremos habitualmente en una base de datos, por lo que en los modelos tendremos todas las funciones que accederán a las tablas y harán los correspondientes selects, updates, inserts, etc.

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio (la funcionalidad del sistema). Un ejemplo de regla puede ser: "Si la mercancía pedida no está en el almacén, consultar el tiempo de entrega estándar del proveedor".
- Lleva un registro de las vistas y controladores del sistema.
- Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes

que actualiza los datos, un temporizador que desencadena una inserción, etc.).

2.8.2. Vista

Las vistas, como su nombre nos hacen entender, contienen el código de nuestra aplicación que va a producir la visualización de las interfaces de usuario, o sea, el código que nos permitirá renderizar los estados de nuestra aplicación en HTML. En las vistas nada más tenemos los códigos HTML y PHP que nos permite mostrar la salida.

En la vista generalmente trabajamos con los datos, sin embargo, no se realiza un acceso directo a éstos. Las vistas requerirán los datos a los modelos y ellas se generará la salida, tal como nuestra aplicación requiera

Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Las vistas son responsables de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización ()", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).

2.8.3. Controlador

Contiene el código necesario para responder a las acciones que se solicitan en la aplicación, como visualizar un elemento, realizar una compra, una búsqueda de información, etc.

En realidad, es una capa que sirve de enlace entre las vistas y los modelos, respondiendo a los mecanismos que puedan requerirse para implementar las necesidades de nuestra aplicación. Sin embargo, su responsabilidad no es

manipular directamente datos, ni mostrar ningún tipo de salida, sino servir de enlace entre los modelos y las vistas para implementar las diversas necesidades del desarrollo.

Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

El controlador es responsable de:

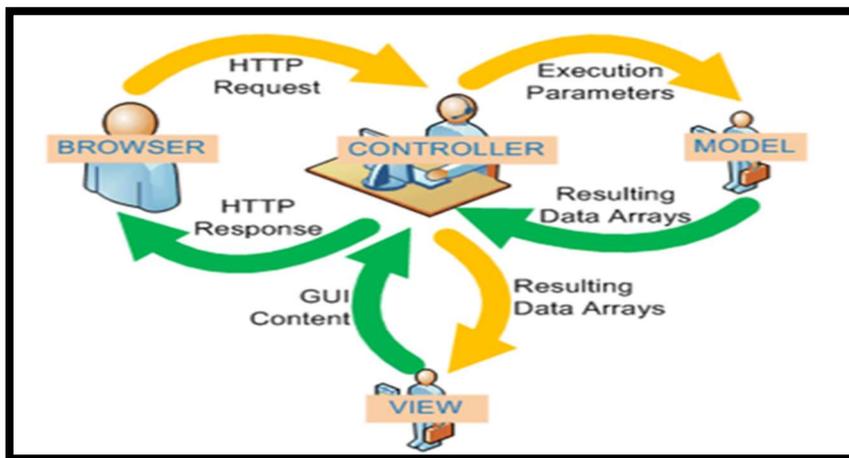
- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos.

2.8.4. Características del Modelo Vista Controlador

Aunque se pueden encontrar diferentes implementaciones de MVC, la característica del flujo que sigue el control generalmente es el siguiente:

Figura 6

Flujo de MVC



Nota. La figura muestra el movimiento de flujo del modelo, vista y controlador con el cual se interactúa. (Vargas, 2017)

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)

- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el. Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta Unidad entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente. (Alicante, 2021)

2.9. HERRAMIENTAS

2.9.1. MagicDraw

Es una herramienta visual de modelado UML y CASE con soporte de trabajo en equipo. Diseñada para analistas de negocio, analistas de software, analistas de sistemas, programadores, ingenieros de control de calidad y redactores de documentación, esta herramienta de desarrollo dinámica y versátil facilita el análisis y el diseño de sistemas orientados a objetos (OO) y bases de datos. Proporciona el mejor mecanismo de ingeniería de código de la industria (con soporte completo de ida y vuelta para los lenguajes de programación J2EE, C#, C++, CORBA IDL, .NET, XML Schema, WSDL), así como el modelado de esquemas de bases de datos, la generación de DDL y las facilidades de ingeniería inversa.

MagicDraw, herramienta CASE desarrollada por No Magic. Es compatible con el estándar UML 2.3, desarrollo de código para diversos lenguajes de programación (Java, C++ y C#, entre otros) así como para modelar datos. Cuenta con capacidad para trabajar en equipo y es compatible con varios entornos de desarrollo (IDEs) Diseñada para los analistas del negocio, los analistas del software, los programadores, los ingenieros y los escritores de la documentación. (Magicdraw, 2021)

- **Un práctico editor UML.** Una interfaz gráfica de usuario amigable y personalizable le permite expresar sus ideas en UML de la forma más rápida y directa posible. Toda la notación UML 2 y la semántica son compatibles.
- **Una poderosa herramienta de ingeniería de código.** Puede construir un modelo UML, generar código fuente a partir de él, escribir más código a mano o en su IDE favorito, luego revertir el código, hacer algunos cambios en el MagicDraw, y fusionar su modelo con el código de nuevo. No se pierde ningún detalle, no importa el camino (hacia adelante o hacia atrás) que usted vaya. Soporte para Java, C#, C++, CORBA IDL, DDL.
- **Una facilidad de informes del modelo UML.** Basado en los archivos XSL, puede generar informes HTML personalizados para cada elemento del modelo. MagicDraw le permite seleccionar qué partes del modelo incluir y cómo debe ser el informe.
- **Una herramienta de análisis de modelos OO.** MagicDraw genera diagramas de dependencia de paquetes, recupera y muestra árboles de herencia, o rastrea cualquier otro tipo de relación como un parámetro de método o un valor de retorno, tales como dependencias, asociaciones, realizaciones y uso de clases.
- **Un generador de modelos OO personalizado.** Conjunto predefinido de patrones de diseño incluyendo GoF, Java, patrón de diseño JUnit. Todos los patrones pueden ser extendidos y nuevos patrones pueden ser creados usando Java o Jython.

- **Herramienta de modelado en equipo.** Más de un desarrollador puede trabajar con el mismo modelo OO. El modelo UML se almacena en el repositorio de Teamwork Server. Cada desarrollador puede bloquear una parte del modelo y trabajar en esa parte individualmente. Más tarde, los cambios pueden confirmarse en el servidor y compartirse con el equipo. El servidor funciona como un sistema de control de versiones de código fuente.
- **Herramienta de modelado de bases de datos.** Puede diseñar el esquema de la base de datos en un diagrama de clases UML, y luego generar su código DDL. Si desea analizar la estructura de la base de datos existente, puede realizar ingeniería inversa a través del puente JDBC.

2.9.2. Gestor de base de datos MySQL

Es el sistema de administración de bases de datos (Database Management System, DBMS) más popular, desarrollado y proporcionado por MySQL AB. Es un sistema de gestión de base de datos relacional, multihilo y multiusuario. MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

MySQL es un sistema de base de datos disponible gratuitamente. Sin embargo, hay varias ediciones de pago también disponibles con las que puede utilizar la funcionalidad avanzada.

MySQL es fácil de usar en comparación con otros programas de bases de datos como Microsoft SQL Server y Oracle database etc. Puede ser usado con cualquier lenguaje de programación, pero es ampliamente usado con PHP.

MySQL puede ejecutarse en múltiples plataformas como Linux, Windows, Unix y un esquema de información para definir y administrar sus metadatos. Puede instalarlo en su sistema local o incluso en el servidor.

a) **Características clave de MySQL**

MySQL presenta algunas ventajas que lo hacen muy interesante para los desarrolladores. La más evidente es que trabaja con bases de datos relacionales, es decir, utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.

Al ser basada en código abierto es fácilmente accesible y la inmensa mayoría de programadores que trabajan en desarrollo web han pasado usar MySQL en alguno de sus proyectos porque al estar ampliamente extendido cuenta además con una ingente comunidad que ofrece soporte a otros usuarios. Pero estas no son las únicas características como veremos a continuación:

- **Facilidad de gestión:** Es bastante fácil de descargar y utilizar el software.
- **Alto rendimiento:** Le proporciona utilidades de carga rápida con diferente caché de memoria.
- **Escalable:** Con MySQL, puede escalar en cualquier momento. Es realmente fácil crear almacenes de datos que incluyen una enorme cantidad de datos.
- **Compatibilidad:** MySQL es compatible con todas las plataformas modernas como Windows, Linux, Unix.
- **Rendimiento:** MySQL le ofrece resultados de alto rendimiento sin perder la funcionalidad esencial.
- **Seguridad completa de los datos:** Sólo los usuarios autorizados pueden acceder a la base de datos. Completa seguridad de los datos.
- **Bajo Costo:** Es gratis de usar.
- **Eficiencia de memoria:** MySQL tiene baja pérdida de memoria.
- **Arquitectura Cliente y Servidor:** MySQL basa su funcionamiento en un modelo cliente y servidor. Es decir, clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento. Cada cliente puede hacer consultas a través del sistema de registro para obtener datos, modificarlos, guardar estos cambios o establecer nuevas tablas de registros, por ejemplo.

- **Compatibilidad con SQL:** SQL es un lenguaje generalizado dentro de la industria. Al ser un estándar MySQL ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a MySQL.
- **Vistas:** Desde la versión 5.0 de MySQL se ofrece compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos SQL. En bases de datos de gran tamaño las vistas se hacen un recurso imprescindible.
- **Procedimientos almacenados.** MySQL posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.
- **Desencadenantes.** MySQL permite además poder automatizar ciertas tareas dentro de nuestra base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros o optimizar su funcionalidad.
- **Transacciones.** Una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas. En caso por ejemplo de una falla de energía, cuando el monitor falla u ocurre algún otro inconveniente, el sistema opta por preservar la integridad de la base de datos resguardando la información.

b) **Ventajas de usar MySQL**

Descritas las principales características de MySQL es fácil ver sus ventajas. MySQL es una opción razonable para ser usado en ámbito empresarial. Al estar basado en código abierto permite a pequeñas empresas y desarrolladores disponer de una solución fiable y estandarizada para sus aplicaciones. Por ejemplo, si se cuenta con un listado de clientes, una tienda online con un catálogo de productos o incluso una gran selección de contenidos multimedia disponible, MySQL ayuda a gestionarlo todo debida y ordenadamente.

2.9.3. Lenguaje de Programación PHP

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web puede incrustar código PHP que se ejecutara cada vez que se visite una página. El código PHP es interpretado en el servidor Web y genera código HTML y otro contenido que el visitante verá.

PHP fue concebido en 1994 y es fruto del trabajo de un hombre, Rasmus Lerdorf. Ha sido adoptado por otras personas de talento y ha experimentado tres transformaciones importantes hasta convertirse en el producto actual. En octubre de 2002, era utilizado por más de nueve millones de dominios de todo el mundo y su número crece rápidamente. Si desea conocer el número actual de sitios que utilizan este lenguaje, visite el sitio <http://www.php.net/usage.php>. PHP es un producto de código abierto, lo que quiere decir que puede acceder a su código. Puede utilizarlo, modificarlo y redistribuirlo sin coste alguno.

Entre los competidores principales de PHP se puede citar a Perl, Microsoft Active Server Pages (ASP), Java Server Pages (JSP) y Allaire ColdFusion. En comparación con estos productos, PHP cuenta con muchas ventajas, entre las que se encuentran las siguientes: (Welling & Thompson, 2004)

- Alto rendimiento
- Interfaces para una gran cantidad de sistemas de base de datos diferentes.
- Bibliotecas incorporadas para muchas tareas Web habituales.
- Bajo coste.
- Facilidad de aprendizaje y uso.
- Portabilidad.
- Acceso al código abierto.

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir, es un lenguaje de programación con variables, sentencias condicionales, ciclos (bucles), funciones.... No es un lenguaje de marcado como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes. Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del browser, pero sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP. (Estrada, 2021)

2.9.4. Bootstrap

Bootstrap es un framework CSS desarrollado por Twitter en 2010, para estandarizar las herramientas de la compañía.

Inicialmente, se llamó Twitter Blueprint y, un poco más tarde, en 2011, se transformó en código abierto y su nombre cambió para Bootstrap. Desde entonces fue actualizado varias veces y ya se encuentra en la versión 4.4.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles. Desde que vio la luz, Bootstrap se ha caracterizado por tratarse de una excelente herramienta para crear interfaces de usuarios limpias y totalmente adaptables a cualquier tipo de dispositivo y pantalla, independientemente de su tamaño. Ya hemos dicho que Bootstrap se ha convertido en una de las herramientas más utilizadas hoy en día a la hora de realizar cualquier diseño web.

Esto es debido a algunas de sus principales características, que lo han hecho muy atractivo a los ojos de los desarrolladores. (acensTechnologies, 2016)

- Fácil e intuitivo
- Compatibles con todos los navegadores.
- Optimizado para dispositivos móviles.
- Amplia comunidad de desarrolladores tras el proyecto.

2.9.5. JavaScript

JavaScript es un lenguaje de programación diseñado en un principio para añadir interactividad a las páginas webs y crear aplicaciones web. A pesar de la similitud en el nombre, no está relacionado con Java. Se emplea en el desarrollo de páginas web para tareas como cambiar automáticamente la fecha de una página, hacer que una página aparezca en una ventana emergente al hacer clic en un enlace o que un texto o imagen cambien al pasar el ratón por encima. También suele emplearse para hacer encuestas y formularios. Se ejecuta en el ordenador del visitante a la web, por lo que no requiere descargas constantes desde el sitio web. (Armetrics, 2021)

- **Uso de JavaScript**

JavaScript también se utiliza en entornos que no se basan en web, tales como documentos PDF, navegadores específicos del sitio y widgets de escritorio. Máquinas virtuales de JavaScript (VMs) mucho más nuevas y rápidas, así como plataformas construidas sobre este lenguaje, han aumentado la popularidad de JavaScript para aplicaciones web. Por parte del cliente, JavaScript se ha aplicado tradicionalmente como un lenguaje interpretado, pero los navegadores más recientes pueden realizar la compilación al momento.

También se utiliza en el desarrollo de videojuegos, en la creación de aplicaciones de escritorio y móviles y en la programación de servidores con entornos de

ejecución como Node.js. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Normalmente se utilizaba JavaScript en las páginas web HTML para realizar operaciones y solo en el marco de la aplicación cliente, sin acceso a funciones del servidor. En la actualidad se emplea para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript es interpretado en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML. (McFarland, 2014)

2.9.6. jQuery

jQuery a una librería o biblioteca de JavaScript que facilita la programación en este lenguaje, al ofrecer una serie de funciones y métodos con los cuales los usuarios pueden producir páginas web o sitios web más rápido y fácil.

JavaScript tiene un pequeño secreto vergonzoso: escribirlo puede ser difícil. Si bien es más simple que muchos otros lenguajes de programación, JavaScript sigue siendo un lenguaje de programación. Y muchas personas, incluidos los diseñadores web, encuentran difícil programarlo.

Para complicar aún más las cosas, los diferentes navegadores web entienden JavaScript de manera diferente, por lo que un programa que funciona en, digamos, Chrome puede no responder por completo en Internet Explorer 9. Esta situación común puede costar muchas horas de prueba en diferentes máquinas y diferentes navegadores para asegurarse de que un programa funcione correctamente para toda la audiencia de su sitio.

jQuery es el arma secreta de un diseñador web en la batalla de la programación JavaScript. Con jQuery, puede realizar tareas en una sola línea de código que podrían requerir cientos de líneas de programación y muchas horas de pruebas en el navegador para lograrlo con su propio código JavaScript. (Porto & Gardey, 2019)

a) Características de JQuery

A continuación, se presentan las principales características de JQuery:

- Es un software libre, por lo que puede ser empleado por cualquier usuario de manera gratuita.
- Su librería permite actualizaciones constantes y rápidas.
- Posee un código abierto y compatible con diferentes navegadores.
- Es fácil de usar, por lo que permite ahorrar tiempo y esfuerzo.
- Su desempeño se integra muy bien con AJAX, una técnica de desarrollo web.
- Permite realizar animaciones, efectos y personalizaciones.
- Es compatible con diferentes buscadores como Google Chrome, Microsoft Edge, Firefox, IE, Safari, Android e IOS, cuyas páginas web se deben programar en formas diferentes. **(Parada, 2019)**

2.10. MÉTRICAS DE CALIDAD DE SOFTWARE

Para asegurar que el proyecto se desarrolle de aplicaron una serie de métricas de calidad seguridad, donde se calcularan los parámetros de calidad que cumple el sistema. En la mayoría de los desafíos técnicos, las métricas nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto.

El proceso para intentar mejorarlo, el producto se mide para intentar aumentar su calidad.

El principio, podría parecer que la necesidad de la medición es algo evidente después de todo es lo que nos permite cuantificar y por consiguiente gestionar de forma más efectiva.

La medición es muy común en el mundo de la ingeniería. Medimos potencia de consumo, pesos, dimensiones físicas, temperaturas, voltajes, señales de ruidos por mencionar algunos aspectos. Desgraciadamente la medición se aleja de lo

común en el mundo de la ingeniería del software. Encontramos dificultades en ponernos de acuerdo sobre que medir y como va evaluar las medidas.

- Para indicar la calidad del producto.
- Para evaluar la productividad de la gente que desarrolla el producto.
- Para evaluar los beneficios en términos de productividad y de calidad, derivados de uso de nuevos métodos y herramientas de la ingeniería de software.
- Para establecer una línea de base para la estimación
- Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

2.10.1. Calidad de Software

La calidad de un producto software depende del cumplimiento de un amplio listado de variables y de indicadores, que se deben tener en cuenta en las fases iniciales del ciclo de vida. Para alcanzarla es necesario darle mayor importancia a cuestiones como los requisitos no-funcionales, y a características como interoperabilidad, flexibilidad, usabilidad, confiabilidad, desempeño, escalabilidad, seguridad, etc. Entre los atributos más relevantes se encuentran:

- **Seguridad.** Debido a iniciativas como cero papeles, mejoramiento ambiental, gestión documental, y la tendencia actual hacia la digitalización de la información de las empresas y la sociedad en general, y debido a que prácticamente todos los dispositivos electrónicos tienen conectividad, se vuelve sensible la información confidencial y protegida por marcos legales. Es por esto que este atributo toma importancia, y se hace imperante mejorar la seguridad y sus atributos de calidad, como la integridad, la privacidad y la confidencialidad, incrementado el nivel de la complejidad y la efectividad de los mecanismos software que prevengan las diferentes modalidades de ataque.

- **Disponibilidad y Confiabilidad.** La dependencia social, organizacional y de gobierno de los sistemas de información, exige el aseguramiento de una funcionalidad ininterrumpida de todas sus funciones, y cada vez son menos tolerables las interrupciones o fallas.
- **Usabilidad.** El acceso cada vez mayor a todo tipo de usuario hace que el desarrollo de software sea más exigente en el diseño de interfaz y factores relacionados con la usabilidad, y cada vez se va ir agudizando más este problema en el análisis y diseño, lo cual incrementará los costos.

2.10.2. Norma ISO/IEC 25000

La calidad del producto, junto con la calidad del proceso, es uno de los aspectos más importantes actualmente en el desarrollo de Software. Relacionada con la calidad del producto, recientemente ha aparecido la familia de normas ISO/IEC 25000, que proporciona una guía para el uso de la nueva serie de estándares internacionales llamada Requisitos y Evaluación de Calidad de Productos de Software (SQuaRE - System and Software Quality Requirements and Evaluation).

La norma ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software.

La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software. Esta familia de normas ISO/IEC 25000 se encuentra compuesta por cinco divisiones.

Figura 7

División de calidad de Software.



Nota. La figura muestra la división de la gestión de la división de la calidad de la ISO 25000.

La calidad del producto software forma parte central como lo menciona ISO/IEC 25000, relacionando directamente las variables de funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

Según Monsalve (2010) “La calidad que pueden alcanzar los productos software, y en general cualquier producto, está sometida a como se desarrolla cada una de las etapas de la vida del producto, partiendo por la definición de la idea del producto hasta la entrega y mantención del mismo”.

ISO/IEC 25000 define tres vistas diferenciadas en el estudio de la calidad de un producto:

- **Vista interna:** Se ocupa de las propiedades del sistema como: el tamaño, la complejidad o la conformidad con las normas de orientación a objetos.

Puede utilizarse desde las primeras fases del desarrollo, permitiendo detectar deficiencias en el sistema en etapas tempranas del ciclo de vida.

- **Vista externa:** Analiza el comportamiento del sistema en producción y estudia sus atributos, por ejemplo: el rendimiento de un software en una máquina

determinada, el uso de memoria de un programa o el tiempo de funcionamiento entre fallos.

Esta vista necesita que el sistema esté completo y en la fase de producción del producto, siendo muy dependiente de la máquina donde se ejecute.

- **Vista en uso:** Mide la productividad y efectividad del usuario final al utilizar el sistema.

2.10.3. Clasificación de las características y sub características

La Calidad del Software se clasifica en un conjunto estructurado de características y sub características de la siguiente manera:

2.10.3.1. Adecuación funcional

Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes sub características:

Tabla 1

Definición de métricas de calidad y fórmulas de la Adecuación Funcional.

CARACTERÍSTICA	MÉTRICA	DESCRIPCIÓN	FÓRMULA
A		N	
Complejidad Funcional	Complejidad de la implementación funcional.	Grado en el cual todas las funcionalidades cubren correctamente las tareas para lo que fueron diseñadas.	$CF = \frac{ {(x - y) * 100} }{x}$
Correlación Funcional	Correlación entre las	Capacidad de la plataforma	

	funcionalidades	para relacionar sus funcionalidades y por ende exista un común denominador entre ellas.	$CF = \frac{ {(x - y) * 100} }{x}$
Pertinencia Funcional	Complejidad de la Idoneidad Funcional	Grado en el cual todas las funcionalidades cubren los objetivos de enseñanza relacionado a la programación.	$CF = \frac{ {(x - y) * 100} }{x}$

Nota. La plataforma tiene la capacidad de ser funcionalmente correcto, se divide en 3 subcaracterísticas para evaluar más detalladamente sus métricas de calidad y fórmulas.

2.10.3.2. Eficiencia de desempeño

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas, que se mostrara en la siguiente tabla 2.

Tabla 2

Definición de métricas y fórmulas de la Eficiencia de desempeño.

CARACTERÍSTICA	MÉTRICA	DESCRIPCIÓN	FÓRMULA
Comportamiento Temporal	Tiempo de	Tiempos de respuesta cuando la	$TPR = \frac{(x_1 + x_2 + x_3)}{3}$

	respu esta.	plataforma a cabo una función.	$x = \text{Tiempo 1 de respuesta en 1 minuto.}$ $CP = (10 * TP) - [11 * (TP - 1)]$
Utilización De Recursos	Utiliza ción de CPU	Cantidad de recursos utilizados cuando la plataforma lleva a cabo una función bajo condiciones determinadas.	UR $= \left[x - \left(11 * \frac{x}{10 - 1} \right) \right]$ $x = \text{Porcentaje del uso del procesador.}$
Capacidad	Tama ño de memo ria usada por el proce so	Cantidad de memoria que emplea por el proceso que se está ejecutando de la plataforma	C $= \left(\frac{x}{10} \right)$ $- \left[11 * \left(\left(\frac{x}{100} \right) - 1 \right) \right]$ $x = \text{Tamaño de la memoria usada por el proceso.}$

Nota. Funciona de acuerdo a una cantidad específica de recursos bajo distintos escenarios ya sean hardware o software, se divide en 3 subcaracterísticas para la evaluar aspectos más detallados de la eficiencia.

2.10.3.3. Compatibilidad:

Capacidad de dos o más sistemas o componentes de intercambiar información y llevar a cabo funciones específicas bajo el mismo entorno de hardware y/o software.

Tabla 3.

Definición de métricas y fórmulas de la Compatibilidad.

CARACTERÍSTIC	MÉTRICA	DESCRIPCIÓN	FÓRMULA
A		N	

Coexistencia	Número de programas causantes de conflictos	Grado en el que la plataforma puede coexistir con otro software independiente, sea o no compartiendo recursos.	$C = (10 * x) - [(10 - x) - (10 * x)]$ $x = \text{Número de programas causantes de conflictos.}$
Interoperabilidad	Capacidad de intercambiar información con otras plataformas o programas.	Grado en el que dos o más plataformas tienen la capacidad de intercambiar información y emplear la información que fue compartida.	$I = \text{Capacidad de intercambiar información con otras plataformas o programas}$

Nota. Llevar a cabo sus funciones cuando comparten el mismo entorno sea hardware o software, se divide en dos subcaracterísticas para evaluar aspectos más detallados de la compatibilidad.

2.10.3.4. Usabilidad

Capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones.

Esta característica se subdivide a su vez en las siguientes sub características:

Tabla 4

Definición de métricas de calidad y fórmulas de la usabilidad.

CARACTERÍSTIC A	MÉTRICA	DESCRIPCIÓN N	FÓRMULA
Inteligibilidad	Capacidad del software de detectar vacíos conceptuales	Capacidad de la plataforma para detectar de acuerdo a los resultados obtenidos.	$I = \frac{\text{Capacidad del software de detectar vacíos conceptuales}}{\dots}$
Aprendizaje	Número de características game-based-learning que cumple una plataforma	Capacidad de la plataforma de cumplir con las características que potencialicen las habilidades de la tendencia game based-learning.	$A = \frac{(10 * x)}{7}$ $x = \text{Número de características game-based-learning.}$
Operabilidad	Claridad en los mensajes	Capacidad de la plataforma para ser de fácil uso a través de mensajes para la aclaración de cualquier punto que pueda ser sujeto a dudas.	$O = \frac{ {(x - y) * 100} }{x} * 10$ $x = \text{Número total de entradas}$ $y = \text{Número de entradas de validación}$
Protección Frente a Errores de Usuario	Verificación de entradas validadas	Capacidad de la plataforma para proteger a los usuarios	$PEU = \frac{ {(x - y) * 100} }{x} * 10$

		a través de validaciones y evitar posibles errores.	
Estética	Número de principios de la evaluación heurística de la interfaz de usuario.	Grado en el que la plataforma cumple con los principios heurísticos de la estética en la interfaz de usuario.	$E = \frac{(10 * x)}{7}$ $x = \text{Número de principios de estética que cumple.}$
Accesibilidad	Número de Sistemas operativos compatibles	Capacidad de la plataforma de ser accesible a distintos entornos de software.	$E = \frac{(10 * x)}{7}$ $x = \text{Número total de sistemas operativos compatibles.}$

Nota. Debe ser comprendida, entendida, usada y resuelta por parte del usuario final.

2.10.3.5. Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo determinados. Esta característica se subdivide a su vez en las siguientes sub características:

Tabla 5

Definición de métricas de calidad y fórmulas de fiabilidad.

CARACTERÍSTIC A	MÉTRICA	DESCRIPCIÓN N	FÓRMULA
Madurez	Número de fracasos en casos de prueba.	Capacidad de la plataforma para detectar fallas que determinen la	$M = \frac{ {(x - y) * 100} }{x}$

		completitud a través de pruebas de validación.	$x = \text{Número de pruebas realizadas.}$ $y = \text{Número total de fallas detectadas.}$
Disponibilidad	Tiempo de operación de servicios.	Capacidad de la plataforma o componente de estar operativo para su uso cuando se requiere en un periodo de tiempo específico.	$D = \frac{ {(x - y) * 100} }{x}$ $x = \text{Tiempo de uso en la plataforma en segundos}$ $y = \text{Tiempo de no disponibilidad de la plataforma}$
Tolerancia Fallos	A Tiempo perdido por fallas.	Capacidad de la plataforma para operar según lo previsto en presencia de fallos, sean estos de hardware o software en un periodo de tiempo.	$TF = (x - y) * NI$ $x = \text{Tiempo total de uso de la plataforma en minutos.}$ $y = \text{Sumatoria de tiempos de duración de fallos.}$
Capacitación	Tiempo de promedio de recuperación	Capacidad de la plataforma de recuperarse ante un fallo inesperado y por ende restablecerse en un período de tiempo específico.	$TPR = \frac{(x_1 + x_2 + x_3)}{3}$ $x = \text{Tiempo de respuesta en 1 minuto.}$ $CR = (10 * TP) - [11 * (TP - 1)]$

Nota. Capacidad de desempeñar una función específica, se divide en 4 características para evaluar aspectos más detallados de la fiabilidad, con sus métricas y fórmulas.

2.10.3.6. Seguridad

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes sub características:

Tabla 6

Definición de métricas de calidad y fórmulas de seguridad.

CARACTERÍSTICA	MÉTRICA	DESCRIPCIÓN	FÓRMULA
Confidencialidad	Número de métodos de recuperación y validación de datos.	Capacidad de la plataforma para notificar accesos o modificaciones por personas externas no autorizadas. <i>Si se cumple que $x=y$ es afirmativo</i>	M $= (x - y)$ $* NI$ $x=$ <i>Número de formas de recuperación o validación de datos.</i> $y =$ <i>Número de formas de recuperación o validación de datos no válidas.</i>
Integridad	Número de solicitudes y mensajes de alerta recibidos.	Formas que tiene la plataforma para demostrar la identidad correcta de un usuario	I $= (x - y)$ $* NI$ $x =$ <i>Número de solicitudes de cambio</i> $y =$ <i>Número de mensajes de alerta recibidos</i>
Autenticidad	Número de formas de autenticación	Capacidad de la plataforma para rastrear de forma precisa las	A $= (x - y)$ $* NI$ $x =$ <i>Número de formas</i>

		acciones maliciosas de una persona externa no autorizada.	posibles de autenticación y = Número de formas que no permite autenticación
Responsabilidad	Capacidad de la plataforma de rastrear ingresos maliciosos	Capacidad de la plataforma para proteger la información y los datos y que por ende personas	x = Número de ingresos malicioso. y = Número de rastreos exitosos.

Nota. Para proteger la información y los datos y por ende personas no autorizadas no puedan leer o modificarlos, se divide en 4 características para evaluar aspectos de la seguridad, como sus métricas y fórmulas.

2.10.3.7. Mantenibilidad

Esta característica representa la capacidad del producto software para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas.

Esta característica se subdivide a su vez en las siguientes sub características:

Tabla 7

Definición de métricas de calidad y fórmulas de mantenibilidad.

CARACTERÍSTIC A	MÉTRICA	DESCRIPCIÓN N	FÓRMULA
Modularidad	Número de módulos concatenados.	Grado en el que la plataforma tiene separados todos sus componentes para que la afección en	$M = \frac{ {(x - y) * 100} }{x}$ x = Número total de módulos y = Número de módulos

		alguno tenga un impacto mínimo en los demás.	<i>no concatenados</i>
Reusabilidad	Número de atributos y funciones heredadas.	Grado en el que el código de la plataforma puede ser utilizado en más de un sistema software o en la construcción de otros.	$R = (x - y) * NI$ <i>x = Suma de atributos y funciones de todos los objetos del software</i> <i>y = Suma de atributos y funciones heredadas de todos los objetos</i>
Analizabilidad	Fallas diagnosticadas	Grado en el que la plataforma tiene la facilidad de encontrar fallas en el mismo y diagnosticar las causas de los fallos	$A = \frac{ {(x - y) * 100} }{x}$ <i>x = Número total de fallas encontradas</i> <i>y = Número total de fallas no diagnosticadas</i>
Capacidad De Ser Modificado	Complejidad de la modificación	Grado en el que la plataforma permite que sea modificado de forma eficiente sin afectar su desempeño.	$CM = \frac{x}{y}$ <i>x = Número de modificaciones</i> <i>y = Tiempo que le toma al desarrollador hacer las modificaciones en minutos</i>

Capacidad De Ser Probado	Número de reinicios para pruebas	Grado en el que la plataforma tiene la facilidad de reiniciar sus funciones y restablecerse después de las pruebas realizadas.	$A = \frac{ {(x - y) * 100} }{x}$ $x = \text{Número de total de reinicios}$ $y = \text{Número de reinicios fallidos}$
---------------------------------	----------------------------------	--	---

Nota. Se divide en 5 características para evaluar aspectos más detallados de la mantenibilidad, como sus métricas y fórmulas.

2.10.3.8. Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno hardware, software, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes sub características:

Tabla 8

Definición de métricas de calidad y fórmulas de la portabilidad.

CARACTERÍSTIC A	MÉTRICA	DESCRIPCIÓN N	FÓRMULA
Adaptabilidad	Capacidad de adaptarse a otros entornos	Capacidad del producto para ser adaptado de forma exitosa a diferentes entornos determinados de hardware o software.	$A = \text{Capacidad de adaptarse a otros entornos}$ <i>Tabla de cumplimiento</i>
Facilidad De Instalación	Número de pasos previo a la de Instalación	Grado en el que la plataforma tiene la	$D = (10 * x) - [10 * (x - 1)]$

		facilidad para ser instalado de forma exitosa.	$x = \text{Número de pasos para la instalación}$
Capacidad De Ser Reemplazado	Capacidad de ser reemplazado o	Capacidad de la plataforma para que pueda ser utilizada en lugar de otro software con el mismo propósito.	$CR = \text{Capacidad de ser reemplazado}$ <i>Tabla de Cumplimiento</i>

Nota. Capacidad de la plataforma para ser transferida exitosamente a cualquier entorno, se divide en 3 características para evaluar aspectos más detallados de la portabilidad, como sus métricas y fórmulas.

2.11. ANÁLISIS DE COSTOS

2.11.1. 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 en la industria. (COCOMO, 2013)

2.11.2. Características

- Es una herramienta basada en las líneas de código la cual la hace muy poderosa para la estimación de costos y no como otros que solamente miden el esfuerzo en base al tamaño.
- Representa el más extenso modelo empírico para la estimación de software.
- Existen herramientas automáticas que estiman costos basados en COCOMO como ser: Costar, COCOMO 81.

2.11.3. Ecuaciones del modelo cocomo

Las ecuaciones que se utilizan en los tres modelos son:

$$E = a(KI)^b * m(X), \text{ en persona} - \text{mes.}$$

$$Tdev = c(E)^d, \text{ en meses.}$$

$$P = \frac{E}{Tdev}, \text{ en personas.}$$

Dónde:

- E = Es el esfuerzo requerido por el proyecto, en persona – mes.
- $Tdev$ = Es el esfuerzo requerido por el proyecto, en meses.
- P = Es el numero de personas requerido por el proyecto.
- a, b, c y d , Son constantes con valores definidos en una tabla, segun cada submodelo
- KI = Es la cantidad de lineas de codigo, en miles.
- $m(X)$ = Es un multiplicador que depende de 15 atributos.

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

- **Modo Orgánico:** Un pequeño grupo de programadores experimentados desarrollan software en un entorno familiar. El tamaño del software varía desde unos pocos miles de líneas (tamaño pequeño) a unas decenas de miles (medio).
- **Modo Semilibre o Semiencajado:** Corresponde a un esquema intermedio entre el orgánico y el rígido; el grupo de desarrollo puede incluir una mezcla de personas experimentadas y no experimentadas.
- **Modo Rígido o Empotrado:** El proyecto tiene fuertes restricciones, que pueden estar relacionadas con la funcionalidad y/o pueden ser técnicas. El

problema a resolver es único y es difícil basarse en la experiencia, puesto que puede no haberla.

2.11.4. Modelos de COCOMO II

2.11.4.1. Modelo Básico

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

Tabla 9

Coefficientes del modelo Básico.

Modo	A	b	C	d
Orgánico	2.40	1.05	2.5	0.38
Semi acoplado	3.00	1.12	2.5	0.35
Empotrado	3.60	1.20	2.5	0.32

Nota. La siguiente tabla muestra los coeficientes multiplicadores del modelo básico.

Estos valores son para las fórmulas:

- Personas necesarias por mes para llevar adelante el proyecto

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

- Tiempo de desarrollo del proyecto.

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

- Personas necesarias para realizar el proyecto.

$$(CosteH) = MM/Tdev$$

- Costo total del proyecto

$$(CosteM = CosteH * \text{salario medio entre los programadores y analistas.})$$

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

2.11.4.2. Modelo Intermedio

En este modelo se introducen 15 atributos de coste para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación.

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

Los valores de las constantes a reemplazar en la fórmula se muestran en la siguiente tabla:

Tabla 10

Coeficientes del modelo intermedio.

Proyecto Software	A	B	C	d
Orgánico	3.2	1.05	2.5	0.38
Semi acoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.20	2.5	0.32

Nota. La siguiente tabla muestra los coeficientes multiplicadores del modelo intermedio.

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

2.11.4.3. Atributos de Coste

Cada atributo se cuantifica para un entorno de proyecto. La escala es muy baja bajo - nominal - alto - muy alto - extremadamente alto. Dependiendo de la

calificación de cada atributo, se asigna un valor para usar de multiplicador en la fórmula el significado de los atributos es el siguiente, según su tipo de:

A continuación, los atributos de costes totales:

Tabla 11

Atributos de costes totales.

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	<u>0,88</u>	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		<u>0,87</u>	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	<u>0,86</u> 0,90	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00			
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						

Técnicas actualizadas de programación	1,00	0,91	0,82
1,24	1,10		
Utilización de herramientas de software	1,00	0,91	0,83
1,24	1,10		
Restricciones de tiempo de desarrollo	1,00	1,04	1,10
1,22	1,08		

Valor

Atributos	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	<u>0,88</u>	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		<u>0,87</u>	1,00	1,15	1,30	
Tiempo de respuesta		0,87	1,00	1,07	1,15	
Atributos de personal						
Capacidad de análisis	1,46	1,19	1,00	0,86	0,71	
Experiencia en la aplicación	1,29	1,13	1,00	0,91	0,82	
Calidad de los programadores	1,42	1,17	1,00	<u>0,86</u> 0,90	0,70	
Experiencia en la máquina virtual	1,21	1,10	1,00			
Experiencia en el lenguaje	1,14	1,07	1,00	0,95		
Atributos del proyecto						

Técnicas actualizadas de programación	1,00	0,91	0,82
1			
,24	1,10		
Utilización de herramientas de software	1,00	0,91	0,83
1			
,24	1,10		
Restricciones de tiempo de desarrollo	1,00	1,04	1,10
1			
,22	1,08		

Nota. La siguiente tabla muestra los coeficientes multiplicadores de los atributos totales.

2.12. SEGURIDAD

2.12.1. SGSI (Sistema de Gestión de la Seguridad de la Información)

Toda la información almacenada y procesada por una organización está expuesta ante amenazas de ataque (por intereses comerciales, intelectuales y/o chantaje y extorsión), error (intencionado o por negligencia), ambientales (por ej. Inundación o incendio), fallo en los sistemas (de almacenamiento de datos, informáticos, redes telemáticas), entre otras y también está sujeta a vulnerabilidades que representan puntos débiles inherentes a su propio uso en el ciclo de vida representado a continuación. Permitir que una información precisa y completa esté disponible de manera oportuna para aquellos autorizados que tienen una necesidad es un catalizador para la eficiencia del negocio. Para poder interrelacionar y coordinar las actividades de protección para la seguridad de la información, cada organización necesita establecer su propia política y objetivos para la seguridad de la información dentro de la coherencia del marco de globales de la organización. Una vez fijados los objetivos en seguridad de la información necesitamos asegurar el modo de poder lograrlos eficazmente, en definitiva, un sistema de gestión de la seguridad de la información o SGSI en su forma abreviada. Por tanto, un SGSI consiste en el conjunto de políticas, procedimientos y directrices junto a los recursos y actividades asociados que son administrados colectivamente por una organización, en la búsqueda de proteger sus activos de información esenciales. Un SGSI desde la visión del estándar internacional

ISO/IEC 27001 es un enfoque sistemático para establecer, implementar, operar, monitorear, revisar, mantener y mejorar la seguridad de la información de una organización y lograr sus objetivos comerciales y/o de servicio (p.ej. en empresas públicas, organizaciones sin ánimo de lucro, etc.).

El alcance de un SGSI puede incluir, en función de dónde se identifiquen y ubiquen los activos de información esenciales, total o sólo una parte de la organización, funciones específicas e identificadas de la organización, secciones específicas e identificadas de la organización, o una o más funciones en un grupo de organizaciones. El término seguridad de la información generalmente se basa en que la información se considera un activo que tiene un valor que requiere protección adecuada, por ejemplo, contra la pérdida de disponibilidad, confidencialidad e integridad.

- **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, entidades o procesos autorizados cuando lo requieran.

2.12.2. Aspectos de seguridad

La seguridad de las instalaciones de cómputo, almacén de datos y la información generada es parte de una conversión exitosa. El reconocimiento de la necesidad de seguridad es una consecuencia natural de la creencia de que la información es un recurso organizacional importante. Con las transacciones cada vez más complejas y muchos intercambios innovadores, la Web ha producido un incremento en las preocupaciones de seguridad para el mundo profesional de SI.

Es útil pensar en la seguridad de sistemas, datos e información en un continuo imaginario que va desde seguridad total hasta acceso abierto por completo. Aunque no hay tal cosa como un sistema totalmente seguro, las acciones de los

analistas y usuarios pretenden mover los sistemas hacia el lado más seguro del espectro, disminuyendo la vulnerabilidad del sistema. Se debe observar que conforme más personas en la organización obtienen mayor potencia de cómputo, obtienen acceso a la Web, o se conectan a las intranets y extranets, la seguridad se vuelve cada vez más difícil y compleja.

La seguridad tiene tres aspectos interrelacionados: físico, lógico y conductual. Los tres deben trabajar juntos si deseamos que la calidad de seguridad permanezca alta. Seguridad física La seguridad física se refiere a proteger el sitio donde se encuentra la computadora, su equipo y software a través de medios físicos. Puede incluir acceso controlado a las salas de cómputo por medio de signos legibles por la máquina, sistemas biométricos o un registro de entrada y salida del sistema por un humano, así como el uso de cámaras de televisión de circuito cerrado para supervisar las áreas de cómputo, respaldando con frecuencia los datos y almacenando los respaldos en un área a prueba de fuego o de agua, a menudo en una ubicación remota segura. Además, el equipo de cómputo pequeño se debe asegurar para que un usuario típico no pueda moverlo y se debe garantizar el suministro ininterrumpido de energía eléctrica. Las alarmas que notifican a las personas apropiadas en caso de fuego, inundación o intrusión no autorizada de una persona deben estar en todo momento en funcionamiento activo.

- **Seguridad lógica**

La seguridad lógica se refiere a los controles lógicos en el software. Los controles lógicos son conocidos por la mayoría de los usuarios como contraseñas o códigos de autorización de alguna clase. Cuando se usan, permiten al usuario entrar al sistema o a una parte específica de una base de datos con una contraseña correcta. Sin embargo, las contraseñas se manejan de manera descuidada en muchas organizaciones. Los empleados han escuchado por casualidad gritar una contraseña en las oficinas atestadas, grabar las contraseñas para sus pantallas y compartir las contraseñas personales con empleados autorizados que han olvidado las suyas. Seguridad conductual Las expectativas conductuales de una

organización están implícitas en sus manuales de políticas e incluso en letreros anunciados en las salas de trabajo y los comedores. Sin embargo, la conducta que los miembros de la organización interiorizan también es crítica para el éxito de los esfuerzos de seguridad. Parte del aspecto conductual de la seguridad es supervisar la conducta a intervalos irregulares para cerciorarse de que se están siguiendo los procedimientos apropiados y corregir cualesquier conductas que se podrían deteriorar con el tiempo. Hacer que el sistema registre el número de inicios de sesión fallidos de los usuarios es una forma de supervisar si usuarios no autorizados están intentando iniciar sesión en el sistema. La salida generada por el sistema se debe reconocer por su potencial de poner a la organización en riesgo en algunas circunstancias. Los controles para la salida incluyen pantallas que sólo se pueden acceder mediante la contraseña, la clasificación de información (es decir, a quién se puede distribuir y cuándo) y el almacenamiento seguro de documentos impresos y almacenados, sin importar el formato. (Kendall & Kendall, 2011)

2.13. PRUEBAS DE SOFTWARE

2.13.1. Pruebas de caja blanca

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican al software, logrando como resultado

que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. (Anonimo, 2011)

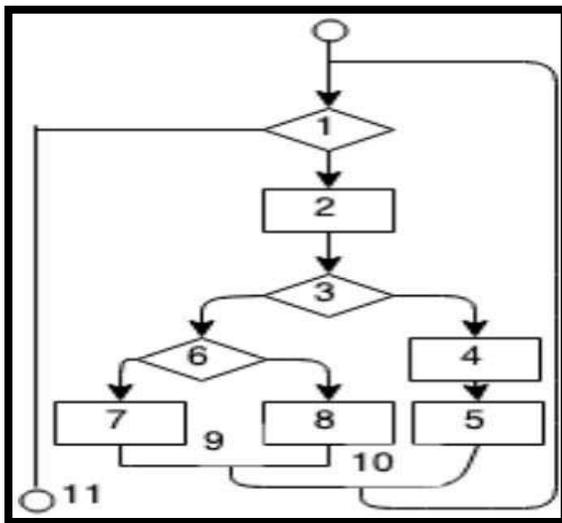
2.13.2. Pruebas de ruta básica

Este tipo de pruebas se basa en diseñar un caos de prueba por cada camino, permite obtener una medida de complejidad lógica de un procedimiento y usar esta medida como guía, se debe conocer la presentación que se conoce como grafo de flujo y las rutas de programa independiente.

- **Notación de grafo de flujo:** se usará para la presentación de flujo de control lógico. Se realiza a partir de PDL (Program Design Language) o pseudocódigo, con el objetivo de mapear el grafo de flujo correspondiente.

Figura 8

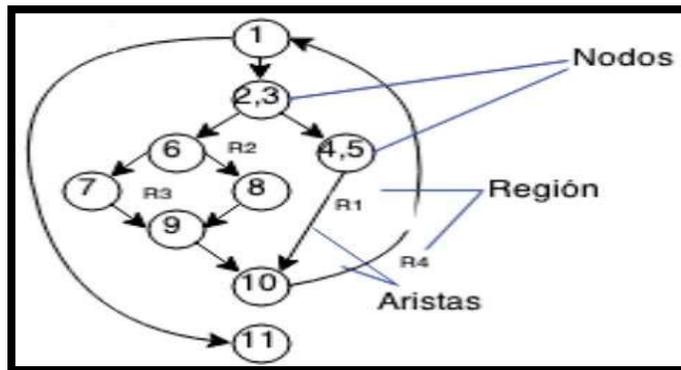
Diagrama de flujo.



Nota. Adaptado de Ingeniería de Software (pág. 415), por (Pressman, 2010).

Figura 9

Grafo de flujo.



Nota. Adaptado de Ingeniería de Software (pág. 415), por (Pressman, 2010).

- **Complejidad Ciclomática:** proporciona una medición de software extremadamente útil, conjunto básico que garantiza que toda la ilustración será ejecutada al menos una vez.

Permite definir el número de caminos independientes del conjunto básico de un programa.

- Número de regiones de gráfico de flujo que corresponde a la complejidad Ciclomática
- La complejidad Ciclomática $V(G)$ para un grafo de flujo G se define como $V(G) = E - N + 2$

Donde E es el número de aristas del grafo de flujo

N el número de nodos del grafo de flujo

- También se define como

$$V(G) = P + 1$$

Donde P es el número de nodos predicado contenidos en el grafo de flujo.

- **Matrices de Gráfos:** Una matriz de grafo es una matriz cuadrada cuyo tamaño (es decir, número de filas y columnas) es igual al número de nodos del gráfico de flujo. Cada fila y columna corresponde a un nodo identificado y las entradas de la matriz corresponden a conexiones (una arista) entre nodos (Pressman, pág.420).

Figura 10

Matriz de Grafo.

Nodo	Conectado a nodo				
	1	2	3	4	5
1			a		
2					
3		d		b	
4		c			f
5		g	e		

Matriz de grafo

Nota. Adaptado de Ingeniería de Software (pág. 420), por (Pressman, 2010).

2.13.3. Pruebas de caja negra

Estas pruebas permiten obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. En ellas se ignora la estructura de control, concentrándose en los requisitos funcionales del sistema y ejercitándolos.

La prueba de Caja Negra no es una alternativa a las técnicas de prueba de la Caja Blanca, sino un enfoque complementario que intenta descubrir diferentes tipos de errores a los encontrados en los métodos de la Caja Blanca.

Muchos autores consideran que estas pruebas permiten encontrar:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las Bases de Datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para preparar los casos de pruebas hacen falta un número de datos que ayuden a la ejecución de los estos casos y que permitan que el sistema se ejecute en todas sus variantes, pueden ser datos válidos o inválidos para el programa según si lo que se desea es hallar un error o probar una funcionalidad. Los datos se escogen atendiendo a las especificaciones del problema, sin importar los detalles internos del programa, a fin de verificar que el programa corra bien.

Para desarrollar la prueba de caja negra existen varias técnicas, entre ellas están:

- Técnica de la Partición de Equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del software.
- Técnica del Análisis de Valores Límites: esta Técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables.
- Técnica de Grafos de Causa-Efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

Dentro del método de Caja Negra la técnica de la Partición de Equivalencia es una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de pruebas que descubran clases de errores, reduciendo así en número de clases de prueba que hay que desarrollar. (Anonimo, 2011)

2.13.4. Pruebas de estrés

Una prueba de estrés (stress) consiste en probar los límites que un sistema puede soportar. En este tipo de pruebas se suele enviar más peticiones de las que el software podría atender normalmente para saber el comportamiento de la aplicación.

Por otro lado, las pruebas de estrés (stress testing) son realizadas sobrecargando un sistema más allá de sus especificaciones, para verificar cómo

y cuándo fallará. Dentro de informática podemos colocar una gran carga en la base de datos, entradas (peticiones) continuas al sistema o almacenar información más allá de la capacidad de memoria del sistema. (Platzi, 2018)

CAPÍTULO III

3. MARCO APLICATIVO

En este capítulo se dará paso al diseño y desarrollo del sistema, los diversos procedimientos y comportamientos más relevantes que se desarrollan, con el fin de identificar los requerimientos que se precisan en el sistema, también se identificarán los principales actores, de esta forma se logrará preparar un adecuado desarrollo del sistema.

3.1. DESARROLLO DE LA METODOLOGÍA

Se utilizará como base de requerimientos la ingeniería de requerimientos, requerimientos funcionales y no funcionales. Para obtener la información clara y precisa a través de la captura, análisis y especificación de requisitos, para el correcto desarrollo de esta fase se empleó el uso de herramientas provistas para la captura de las necesidades a las que debe responder el sistema, aplicando las herramientas descritas.

Tabla 12

Herramientas usadas para la ingeniería de requerimientos.

HERRAMIENTA	CARACTERÍSTICAS
Observación	Se hizo la observación del proceso de la formulación de contratación que realiza la empresa y los tipos de servicios que ofrece y funciones que desempeñan cada uno de los actores involucrados en el proceso.
Documentación	Se revisaron los documentos que se manejan dentro del proceso, el tipo de presentación, las normas que se debe seguir y los formatos para la realización de los reportes
Entrevistas	Se realizó entrevistas con el personal involucrado dentro del proceso, quienes son: <ul style="list-style-type: none">• Personal administrativo.• Profesional.• Obreros de ocupación.• Clientes.

Nota. Herramientas usadas para la ingeniería de requerimientos del sistema.

Después del análisis de requerimientos se empleó las herramientas que provee la metodología UWE.

3.1.1. Requerimientos funcionales

Los requerimientos funcionales, describen las características mínimas que necesita el sistema para los diferentes procesos.

Tabla 13

Requerimientos funcionales.

ROL	DESCRIPCIÓN	FUNCIÓN
RF-1	Acceder al sistema	Accede al sistema 3 tipos de usuarios (Administrador, administrativo, profesional y clientes), mediante su usuario y contraseña
RF-2	Gestionar Usuarios	Permite crear, listar, actualizar, reset de password y eliminar el registro.
RF-3	Gestionar Roles	Permite la asignación de roles a los usuarios.
RF-4	Gestionar Permisos	Permite la asignación de permisos a los usuarios dependiendo del rol.
RF-5	Administración del software	Permite al usuario con el perfil de Administrador, tener acceso total a los módulos y operaciones de la gestión de la información: creación, listado, actualización, inactivación de los recursos del sistema.
RF-6	Perfil de Usuario	Muestra la información del usuario y permitela actualización de foto de perfil y la Contraseña

RF-7	Gestionar los proyectos	Permite crear, editar, activar, inactivar y estado del proyecto.
RF-8	Administrar Clasificadores	Permite crear, editar, activar, inactivar cada uno de los clasificadores presupuestarios.
RF-9	Gestionar Financiamiento	Permite listar de acuerdo a la gestión seleccionada asignar financiamiento y un documento privado.
RF-10	Habilitar proyecto	Permite habilitar de acuerdo a una gestión, el tipo de servicio del cliente.
RF-11	Generar reporte de seguimiento	Permite de acuerdo al tipo de servicio se realice la verificación y estado del proyecto.
RF-12	Seguimiento de proyectos	El sistema permitirá a los usuarios autorizados el ingresar planes y cronogramas de proyecto.
RF-13	Reporte de proyectos	Al ingresar el proyecto, se podrá realizar reportes.
RF-14	Imprimir seguimiento	Se podrá imprimir la documentación creada.
RF-15	Modificación del proyecto	Antes de que el proyecto se de alta, se podrá modificar las veces que haga falta.

Nota. Requerimientos funcionales los cuales se aplicarán para el desarrollo del funcionamiento.

3.1.2. Requerimientos no funcionales

Requerimientos no funcionales (RFN), se describen cualidades y características del sistema que el usuario puede visualizar.

Tabla 14*Requerimientos no funcionales.*

ROL	FUNCIÓN
RNF-1	El sistema debe funcionar correctamente en los navegadores como ser: Google Chrome, Mozilla Firefox, IE/Edge y otros.
RNF-2	El sistema debe tener una interfaz gráfica, amigable para que el usuario pueda tener una mejor experiencia
RNF-3	El sistema debe tener un menú lateral para que el usuario pueda navegar a través de ellas.
RNF-4	El soporte y mantenimiento periódico del sistema asegura un buen rendimiento al sistema.
RNF-5	El sistema deberá responde a todas las solicitudes del usuario
RNF-6	El sistema deberá encriptar la contraseña de los usuarios
RNF-7	La sesión tendrá un determinado periodo de tiempo activo.
RNF-8	El sistema debe permitir agregar nuevos módulos y funciones a futuro.

*Nota. Requerimientos no funcionales son los cuales se describen en la tabla.***3.1.3. Descripción de actores**

Este proceso permitirá la identificación de los actores que interactuarán con el sistema se tomó en cuenta los roles que cumplen diferentes personas de la empresa.

Tabla 15*Descripción de actores.*

DESCRIPCIÓN	RESPONSABILIDADES
Administrador	<ul style="list-style-type: none"> Se encarga de gestionar los usuarios dentro del sistema web, la asignación de roles y permisos.

	<ul style="list-style-type: none"> • Se encarga de las configuraciones del sistema web. • Se encarga de habilitar formulado, editar fechas. • Encargado de generar reportes.
Técnico	<ul style="list-style-type: none"> • Se encarga de habilitar el formulado. • Se encarga de asignar los respectivos proyectos y las solicitudes. • Se encarga de revisar las solicitudes de proyectos. • Encargado de generar reportes del avance del proyecto.
Usuario	<ul style="list-style-type: none"> • De acuerdo a las normas de la empresa.

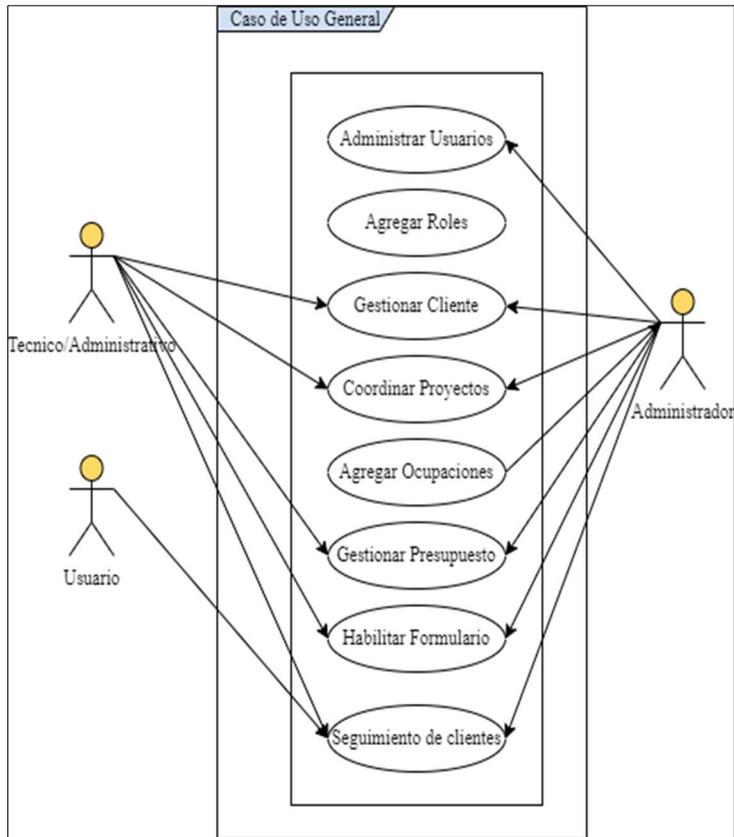
Nota. Descripción de actores.

3.1.4. Diagrama de caso de uso general del sistema

En la siguiente figura presentamos el caso de uso general del sistema el cual nos ayuda a identificar a los actores.

Figura 11

Diagrama de caso de uso general.



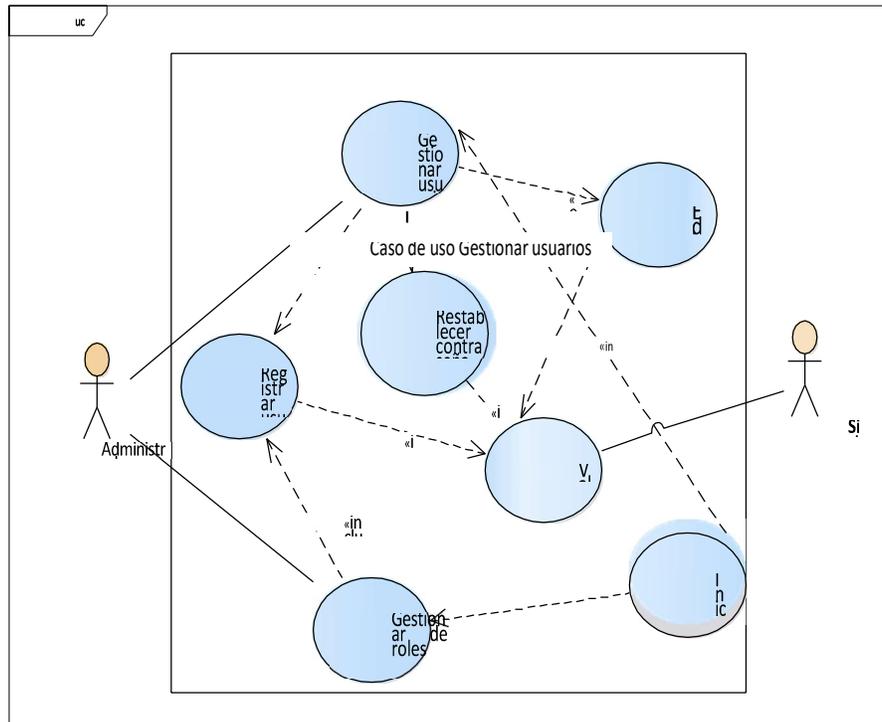
Nota. Diagrama de caso de uso general.

Diagramas de casos de uso específicos del sistema

➤ Diagrama de caso de uso: Gestionar Usuarios

Figura 12

Gestionar usuarios



Nota. Gestionar usuarios.

Tabla 16

Descripción de caso de uso – Gestionar usuarios.

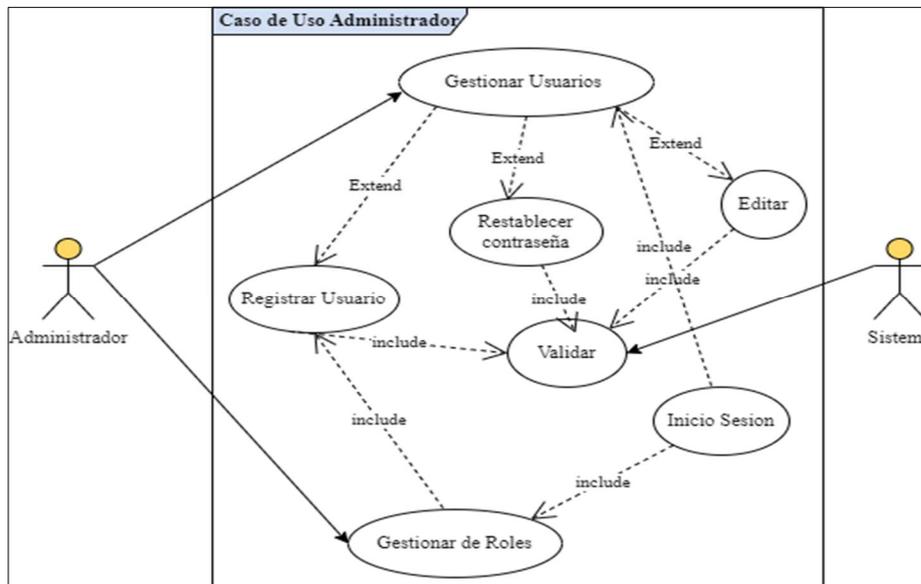
Caso de Uso	Gestionar Usuarios
Identificador:	CU01
Descripción:	El sistema permite gestionar los usuarios registrados en el mismo
Actores:	Administrador
Precondiciones:	El usuario tiene que estar registrado en el sistema, haber iniciado la sesión con éxito e ingresar al panel de administración de usuarios.
Flujo de Eventos:	<p>El caso de uso comienza cuando el usuario selecciona la opción de Administración de usuarios</p> <ol style="list-style-type: none"> 1. El sistema despliega una lista de usuarios registrados

	<ol style="list-style-type: none"> 2. El usuario solicita crear, modificar, reset password, activar o desactivar un usuario. 3. El sistema solicita la información para crear, modificar, reset de password, activar o desactivar un usuario 4. El sistema valida los datos y realiza la acción solicitada
Post Condiciones:	El usuario logra crear modificar, reset password, activar o desactivar un usuario
Flujos alternativos:	Si el usuario trata de crear o modificar un usuario con datos ya existentes muestra un mensaje de error "El usuario ya existe"

Nota. Descripción de caso de uso – Gestionar usuarios.

Figura 13.

Diagrama de caso de uso de administradores.



Nota. Diagrama de caso de uso de administradores.

Tabla 17

Descripción de caso de uso de administradores.

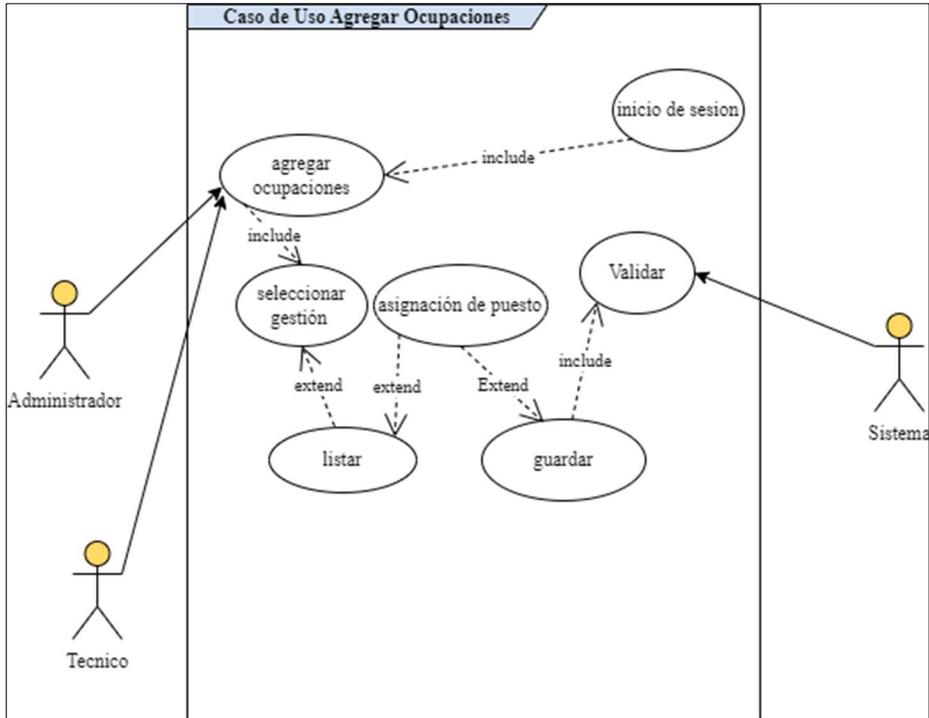
Caso de Uso	Administradores
-------------	-----------------

Identificador:	CU-01
Descripción:	El sistema permite gestionar los usuarios registrados en el mismo
Actores:	Administrador
Precondiciones:	El usuario tiene que estar registrado en el sistema, haber iniciado la sesión con éxito e ingresar al panel de administración de usuarios
Flujo de Eventos:	<p>El caso de uso comienza cuando el usuario selecciona la opción de Administración de usuarios</p> <ol style="list-style-type: none"> 1. El sistema despliega una lista de usuarios registrados 2. El usuario solicita crear, modificar, reset password, activar o desactivar un usuario. 3. El sistema solicita la información para crear, modificar, reset de password, activar o desactivar un usuario 4. El sistema valida los datos y realiza la acción solicitada
Post Condiciones:	El usuario logra crear modificar, reset password, activar o desactivar un usuario
Flujos alternativos:	Si el usuario trata de crear o modificar un usuario con datos ya existentes muestra un mensaje de error "El usuario ya existe"

Nota. Descripción de caso de uso de administradores.

Figura 14

Caso de uso agregar ocupación



Nota. Caso de uso agregar ocupación.

Tabla 18

Caso de uso agregar ocupación.

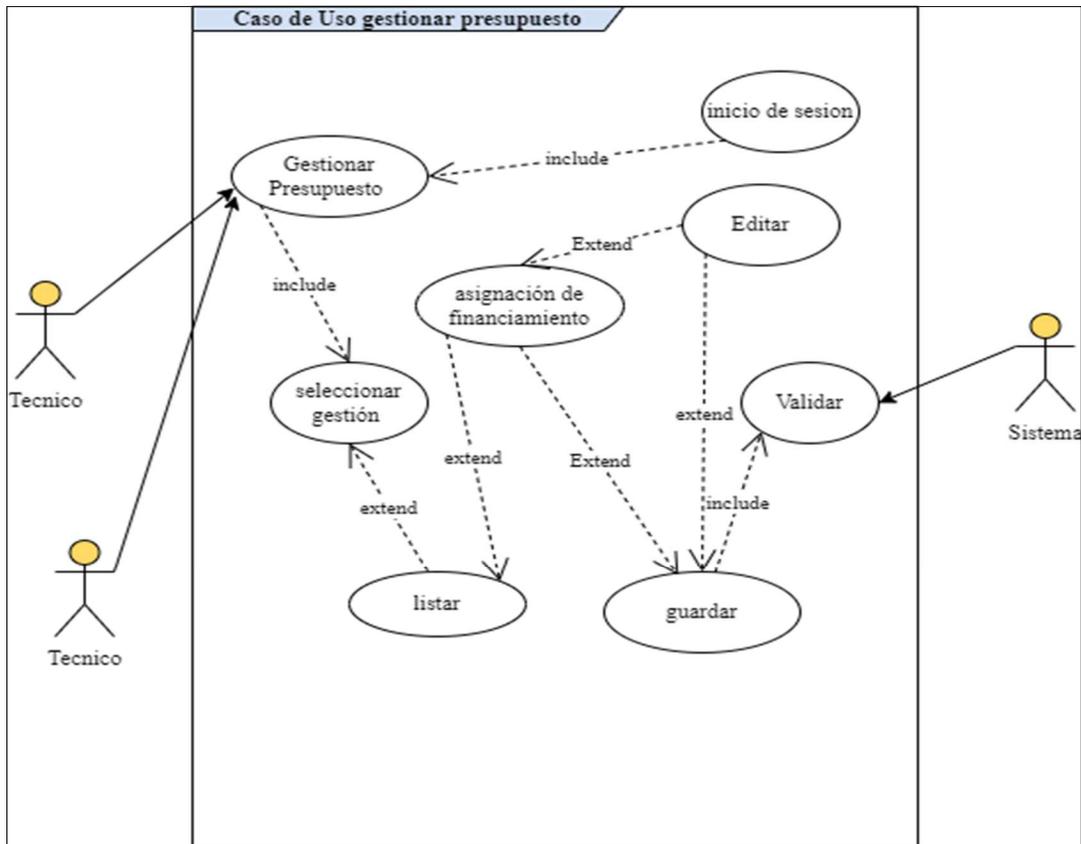
Caso de Uso	Agregar ocupación
Identificador:	CU – 05
Descripción:	En este caso se podrá ver los cargos y asignaciones de los trabajadores en la empresa
Actores:	Administrador, técnico
Precondiciones:	El usuario tiene que estar registrado en el sistema, haber iniciado la sesión con éxito e ingresar al panel agregar ocupación
Flujo de Eventos:	<p>El caso comienza ingresando al menú de ocupación</p> <ol style="list-style-type: none"> 1. El sistema permite seleccionar un cargo del trabajador 2. El sistema muestra un listado de todos los cargos 3. El usuario pulsa en el botón de guardar 4. El sistema valida los datos

Post Condiciones:	El técnico logra asignar las ocupaciones
Flujos alternativos:	El técnico validara para su asignación

Nota. Caso de uso agregar ocupación.

Figura 15

Caso de uso – Gestionar Presupuesto.



Nota. Caso de uso – Gestionar Presupuesto.

Tabla 19

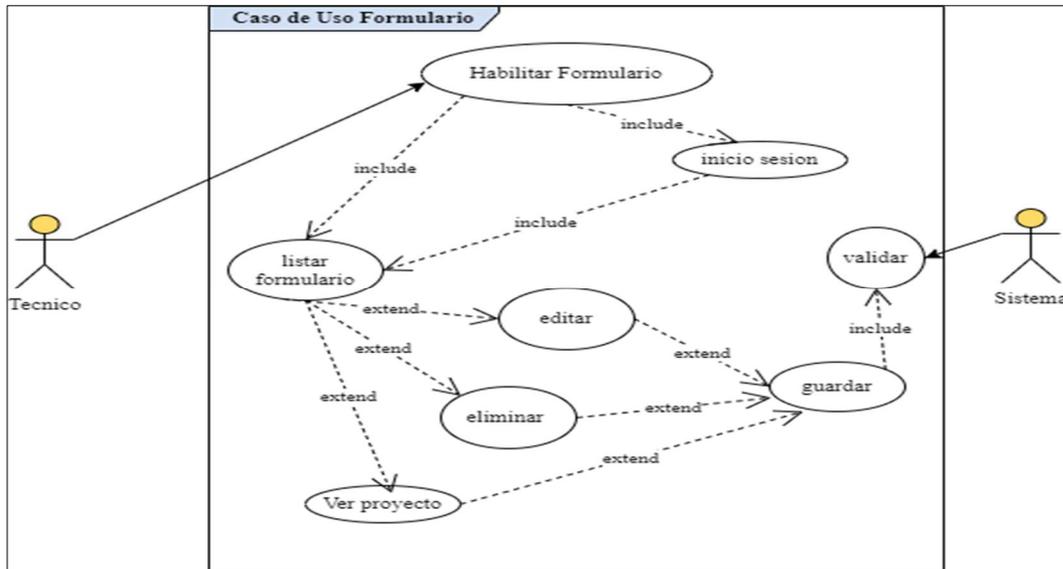
Descripción de caso de uso de gestionar presupuestos.

Caso de Uso		Gestionar Usuarios
Identificador:	CU – 02	
Descripción:	En este caso se podrá ver el financiamiento por obra civil en la gestión específica	
Actores:	Administrador, técnico	
Precondiciones:	El usuario tiene que estar registrado en el sistema, haber iniciado la sesión con éxito e ingresar al panel de Asignar Financiamiento	
Flujo de Eventos:	<p>El caso comienza ingresando al menú de presupuesto</p> <ol style="list-style-type: none"> 1. El sistema permite seleccionar una gestión específica 2. El sistema muestra un listado de todos los registros 3. El sistema solicita información de la asignación del presupuesto 4. El usuario pulsa en el botón de guardar 5. El sistema valida los datos 	
Post Condiciones:	El técnico logra asignar el financiamiento	
Flujos alternativos:	El técnico deberá validar para su asignación.	

Nota. Descripción de caso de uso de gestionar presupuestos.

Figura 16

Caso de uso - habilitación de formulario.



Nota. Caso de uso - habilitación de formulario.

Tabla 20

Descripción de caso de uso habilitar formulario.

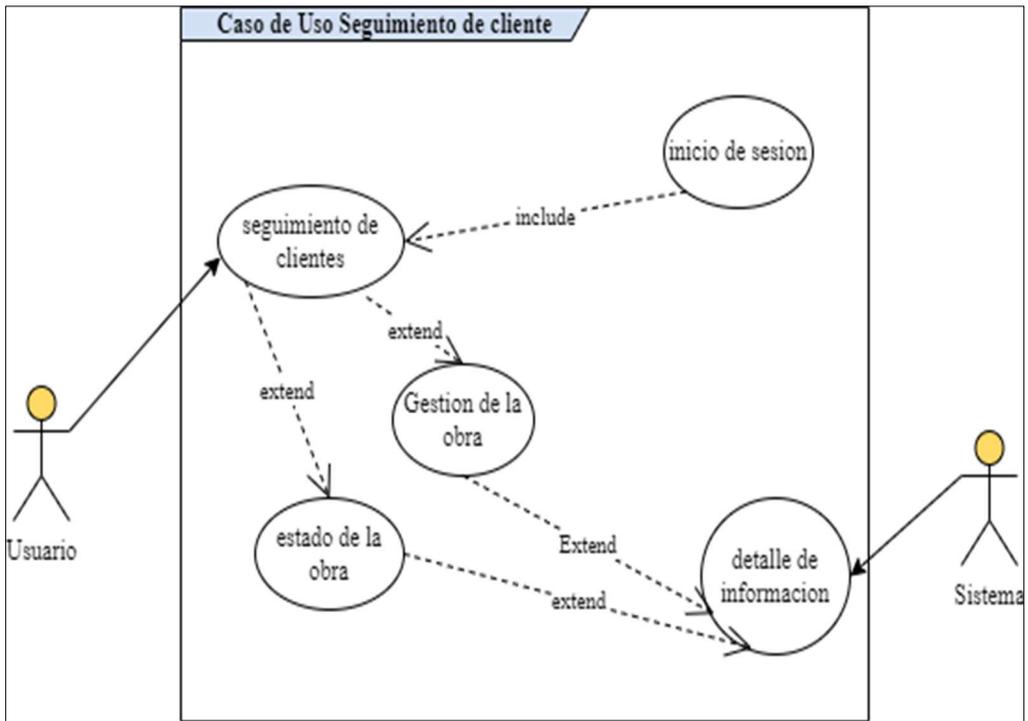
Caso de Uso	Habilitar formulario.
Identificador:	CU-03
Descripción:	Permite el uso de listar los formularios de proyectos habilitados
Actores:	Técnico
Precondiciones:	El técnico deberá haber iniciado sesión e ingresar al formulario
Flujo de Eventos:	El caso de uso inicia al ingresar al sistema
Post Condiciones:	El sistema muestra la lista de los formularios habilitados
Flujos alternativos:	Si el proyecto no este habilitado en la gestión no se habilitara el formulario
Descripción:	Permite el uso de listar los formularios de proyectos habilitados
Actores:	Técnico
Precondiciones:	El técnico deberá haber iniciado sesión e ingresar al formulario
Flujo de Eventos:	El caso de uso inicia al ingresar al sistema

Post Condiciones:	El sistema muestra la lista de los formularios habilitados
Flujos alternativos:	Si el proyecto no este habilitado en la gestión no se habilitara el formulario

Nota. Descripción de caso de uso habilitar formulario

Figura 17

Caso de Uso - Seguimiento del cliente.



Nota. Caso de Uso - Seguimiento del cliente.

Tabla 21

Descripción caso de Uso - Seguimiento del cliente.

Caso de Uso	Seguimiento de cliente
Identificador:	CU-04
Descripción:	En este caso de uso el usuario podrá hacer seguimiento del estado de su obra civil
Actores:	Usuario

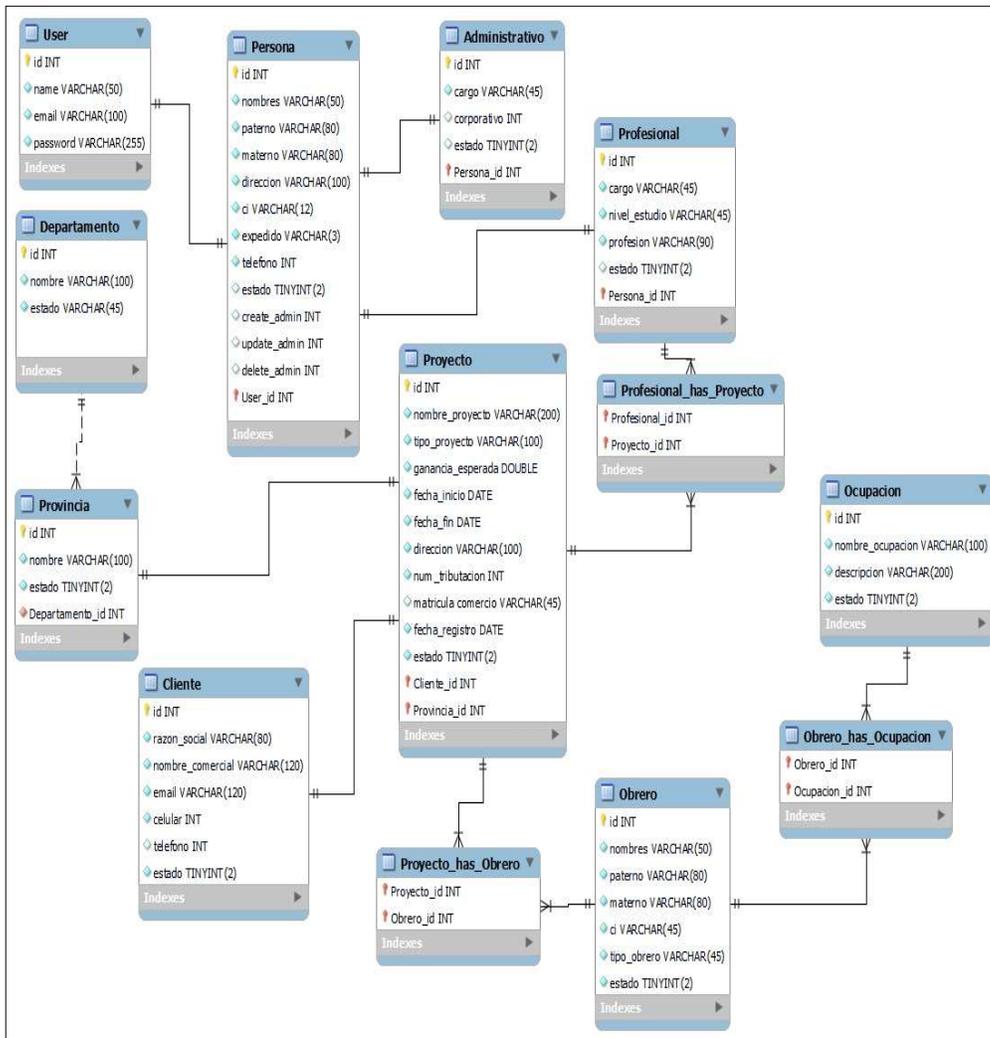
Precondiciones:	El usuario deberá estar registrado y haber iniciado sesión
Flujo de Eventos:	El usuario deberá iniciar sesión para su seguimiento del estado de su obra
Post Condiciones:	Detalle de la información
Flujos alternativos:	El usuario no selecciona la obra ni la gestión saldrá un error

Nota. Descripción caso de Uso - Seguimiento del cliente.

3.1.5. Diseño conceptual

Figura 18

Diseño Conceptual.

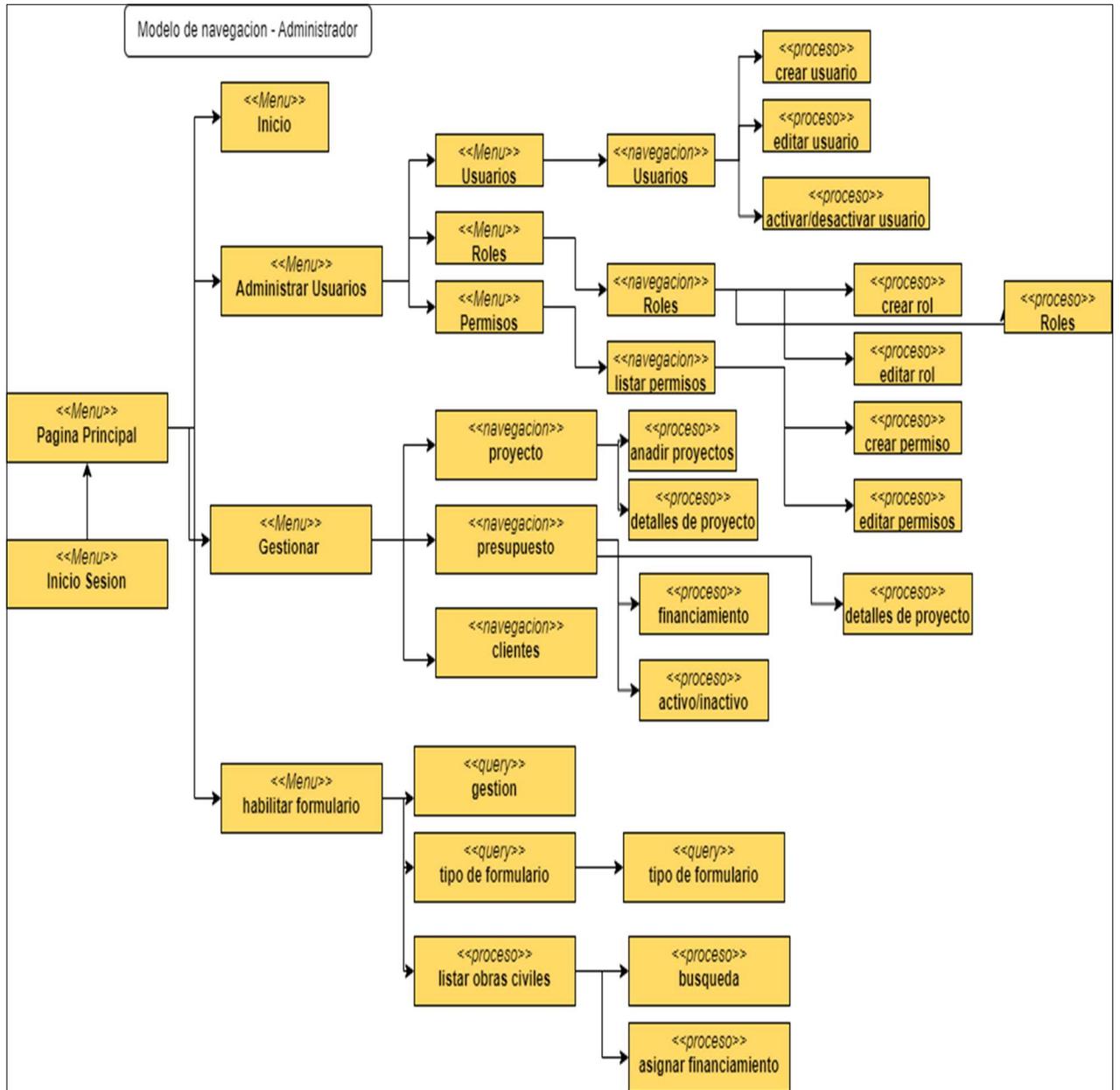


Nota. Diseño Conceptual.

3.1.6. Modelo de Navegación

Figura 19

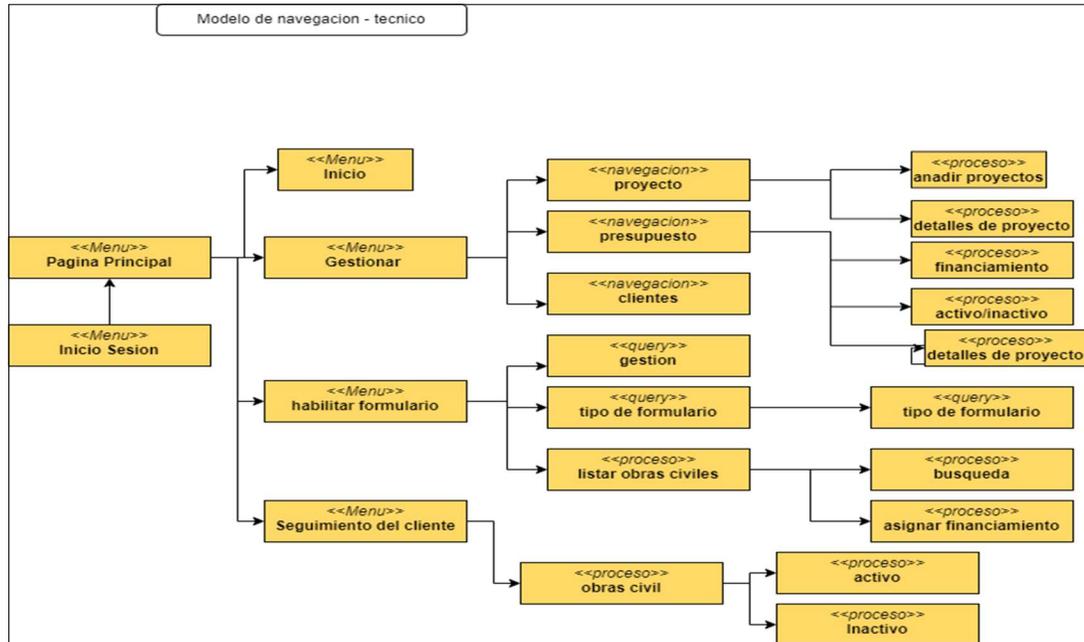
Modelo de navegación –administrador.



Nota. Modelo de navegación –administrador.

Figura 20

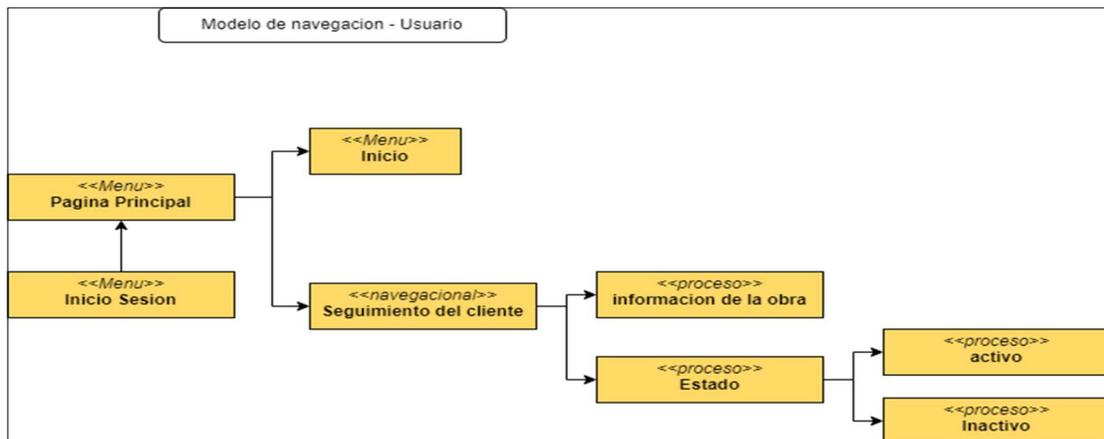
Modelo de navegación – técnico.



Nota. Modelo de navegación – técnico.

Figura 21

Modelo de navegación – Usuario.



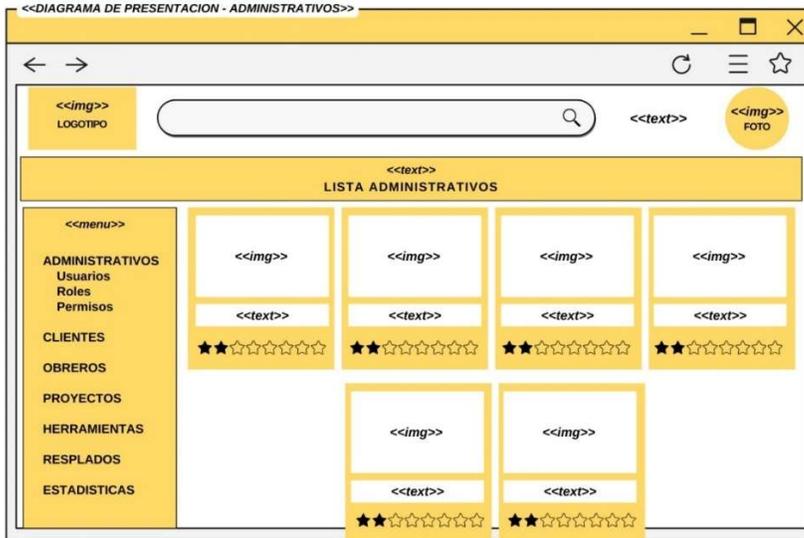
Nota. Modelo de navegación – Usuario.

3.1.7. Modelo presentación

El modelo de presentación de los administrativos y usuarios, a continuación, en la siguiente figura:

Figura 22.

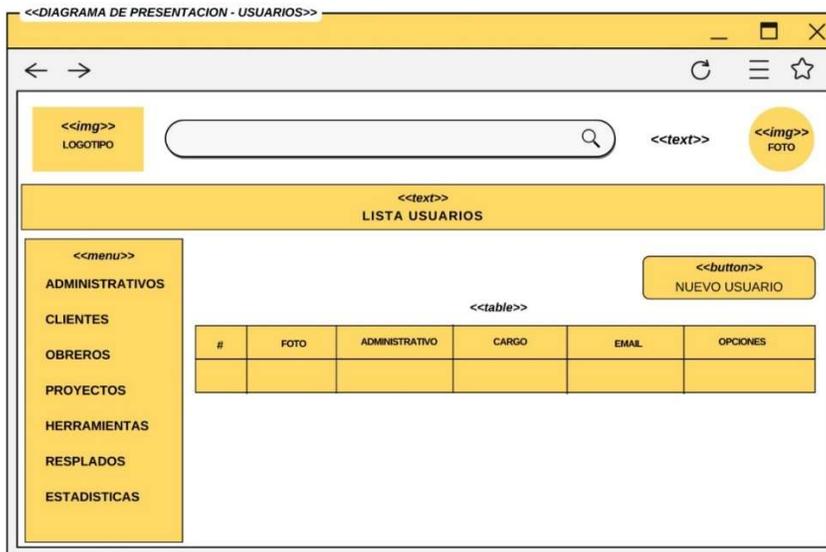
Modelo presentación administrativos



Nota. Modelo presentación administrativos

Figura 23.

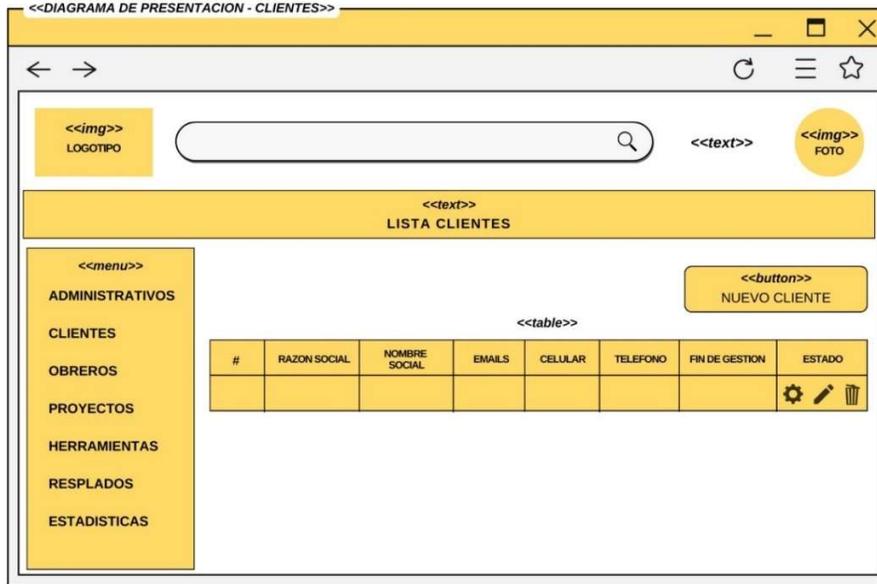
Modelo presentación usuarios.



Nota. Modelo presentación usuarios

Figura 24.

Modelo presentación clientes.



Nota. Modelo presentación clientes.

3.1.8. Modelo de implementación

En este modelo se realiza la implementación del sistema, al dirigirse a la página principal, se podrá ver la página web de la empresa.

Figura 25

Página principal.



Nota. Página principal, con los menús de opciones y login.

A continuación, navegando en la página de inicio podemos observar el resumen de los proyectos.

Figura 26

Inicio de información.



Nota: Inicio de información

También podemos navegar en la página de inicio, ahí podemos observar la cantidad y clientes de la Empresa.

Figura 27

Menú principal clientes.



Nota. Menú principal clientes.

A continuación, en la opción de servicios vemos los detalles de servicios de obras

civiles.

Figura 28

Menú principal servicios.



Nota. Menú principal servicios.

A continuación, en la opción de servicios vemos los detalles de servicios eléctricos.

Figura 29

Menú servicios eléctricos.



Nota. Menú servicios eléctricos.

En la siguiente figura podemos observar el menú de proyectos.

Figura 30

Menú proyectos.



NUESTROS PROYECTOS

En ESECAB S.R.L. nos especializamos en ofrecer soluciones integrales en obras civiles, eléctricas y ambientales. Nuestro equipo de profesionales altamente capacitados y con amplia experiencia se dedica a brindar servicios de calidad, cumpliendo con los más altos estándares de ingeniería y sostenibilidad.

Nota. Menú proyectos.

En la siguiente figura podemos observar el menú Nosotros donde brinda la información de la empresa Esecab S.R.L.

Figura 31

Menú nosotros.



La empresa fue fundada en octubre de 2014 y comenzó ofreciendo servicios de consultoría en diseño final de proyectos en las áreas de civil eléctrica y ambiental.

ESECAB S.R.L.

En general la empresa se ha destacado por su capacidad para brindar soluciones integrales y personalizadas a sus clientes en el campo de la consultoría y construcción en áreas civiles, ambientales y eléctricos. **Octubre 2014**

Nota. Menú nosotros, se brinda la información de la empresa Esecab S.R.L.

A continuación, en la opción de contactos vemos los detalles de formulario para

solicitar información.

Figura 32

Contactos.



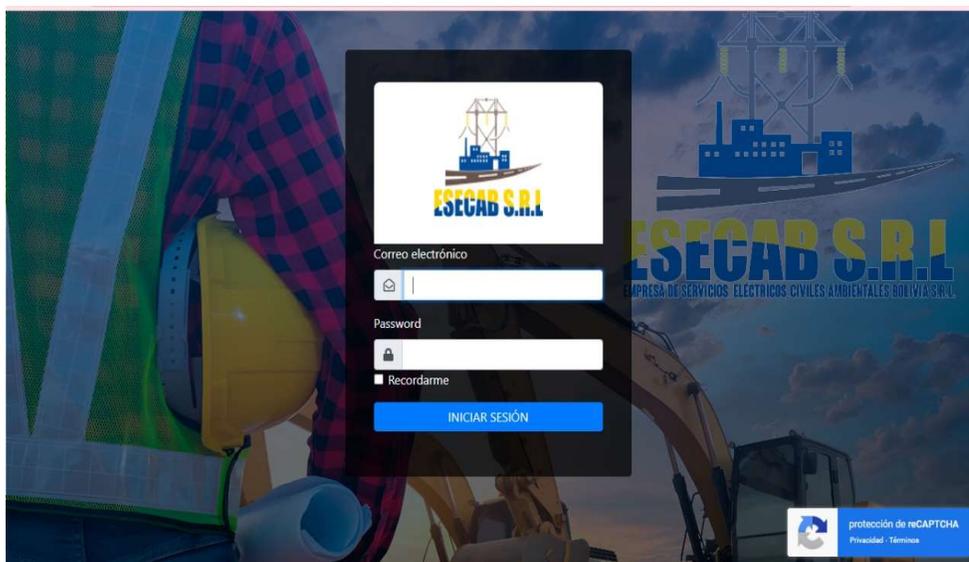
The screenshot shows the top navigation bar of the ESEGAB S.R.L. website with links for SERVICIOS, PROYECTOS, LICITACIÓN, NOSOTROS, CONTACTOS, and INGRESAR. Below the navigation bar is a contact form titled "ENVIANOS UN MENSAJE A ese_cab_srl@gmail.com". The form includes fields for "NOMBRE:" (with a placeholder "Escribe tu nombre Completo"), "CORREO ELECTRONICO:" (with a placeholder "Escribe tu correo Electronico"), and "MENSAJE:". A yellow "ENVIAR" button is located at the bottom right of the form. To the left of the form is an image of a woman in a white shirt and yellow hard hat holding a smartphone.

Nota. Contactos, en donde podemos ver el formulario de consultas.

A continuación, ingresamos a la página de login.

Figura 33

Inicio de sesión.

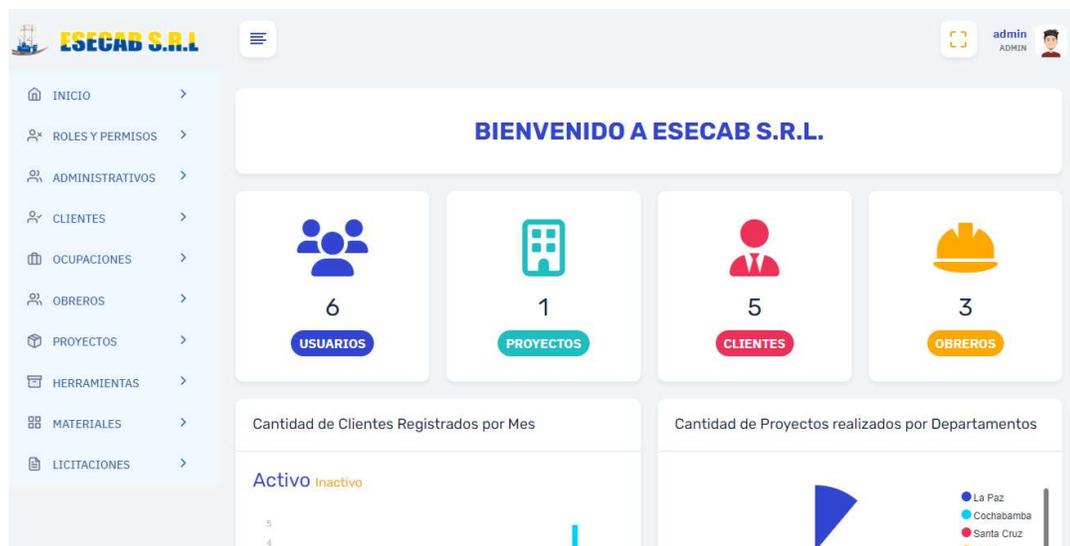


The screenshot shows the login page of the ESEGAB S.R.L. website. The page features a dark background with a construction site image. A central login form contains the ESEGAB S.R.L. logo, a "Correo electrónico" field, a "Password" field, a "Recordarme" checkbox, and a blue "INICIAR SESIÓN" button. At the bottom right, there is a reCAPTCHA logo and links for "Privacidad" and "Términos".

Nota. Inicio de sesión, para verificar las credenciales de acceso.

Figura 34

Menú principal administrador.

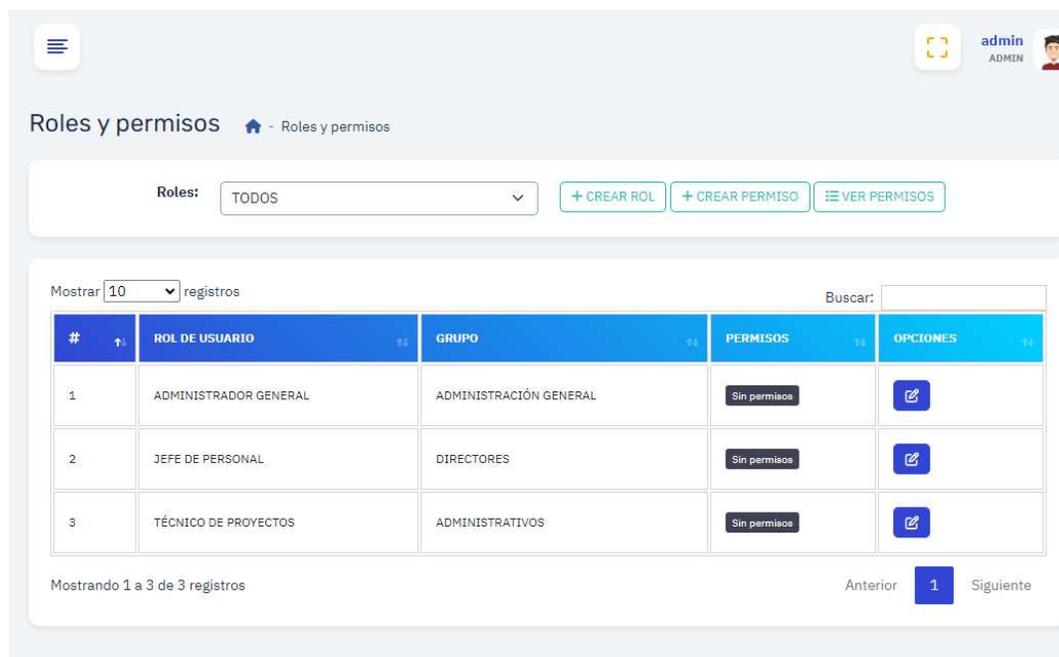


Nota. Menú principal administrador.

En la siguiente figura vemos la lista de roles y permisos.

Figura 35

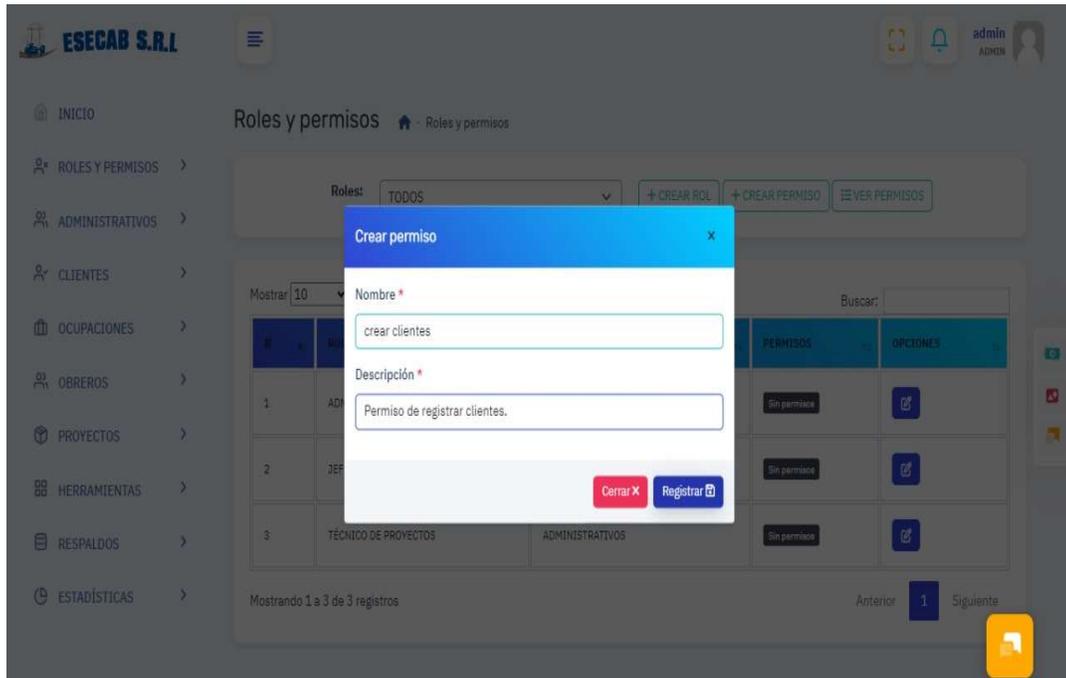
Roles y permisos



Nota. Roles y permisos, con la opción de crear y actualizar.

Figura 36

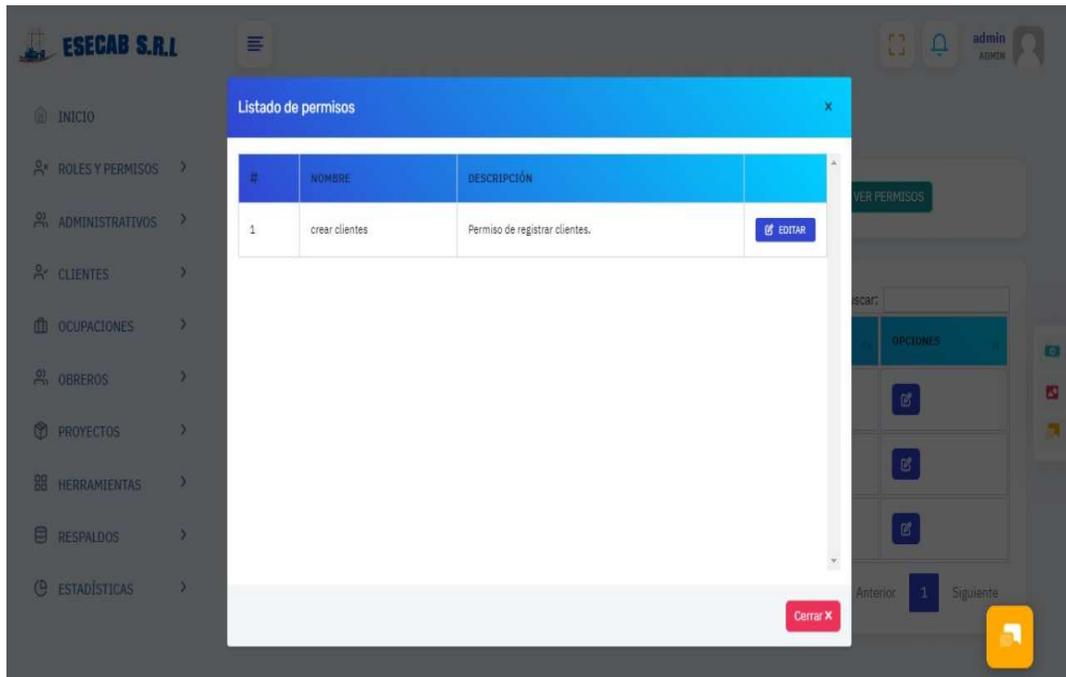
Crear permisos.



Nota. Crear permisos.

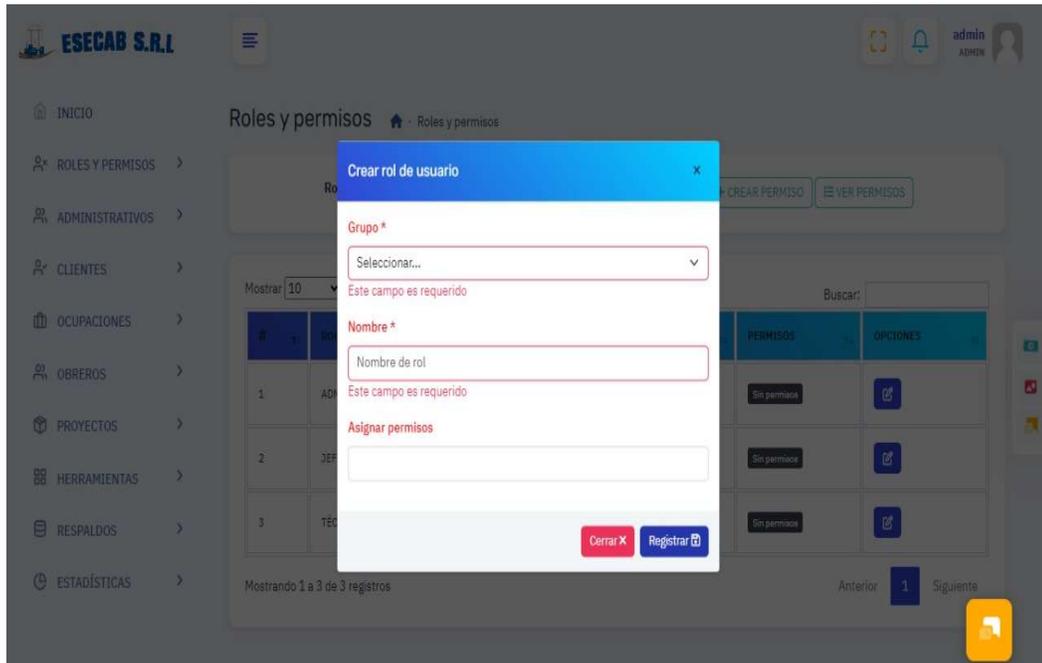
Figura 37

Listado de permisos.



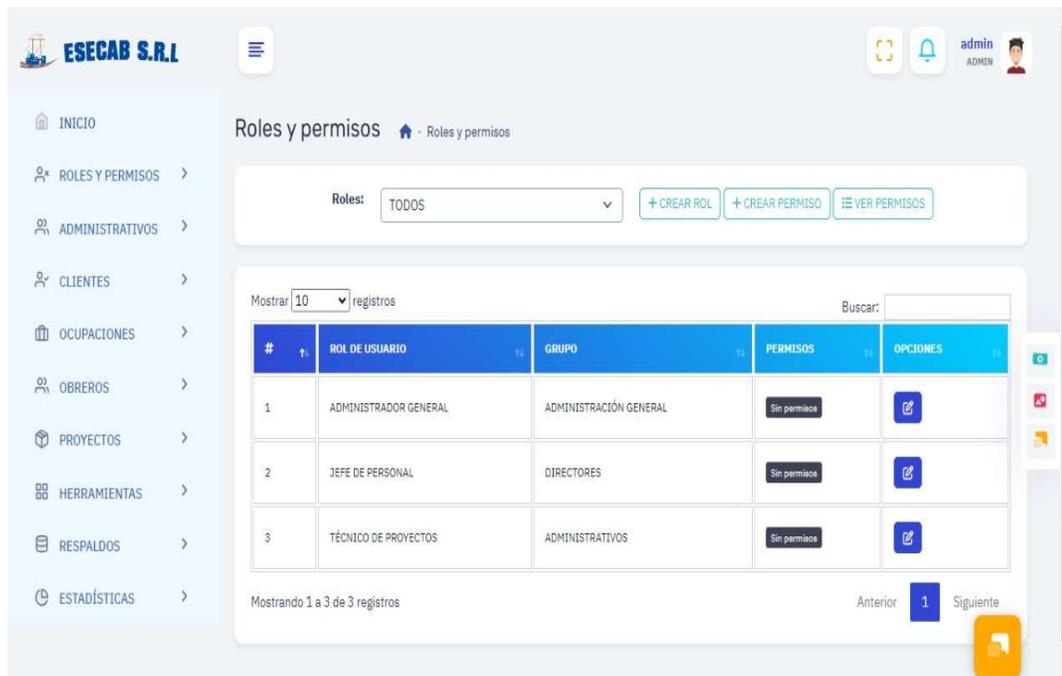
Nota. Listado de permisos.

Figura 38
Crear rol.



Nota. Crear rol.

Figura 39
Roles y permisos.

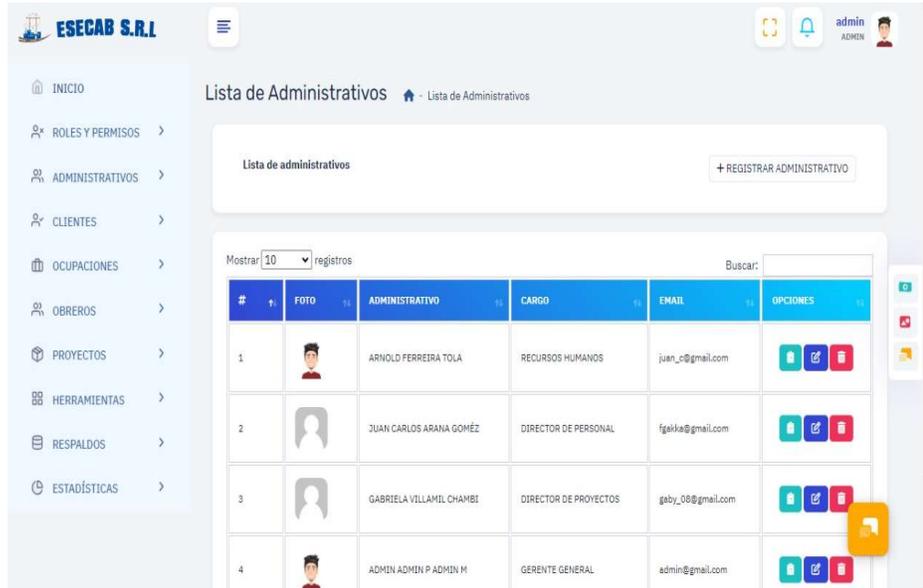


Nota. Roles y permisos.

A continuación, se muestra la lista de administrativos, con las opciones correspondientes.

Figura 40

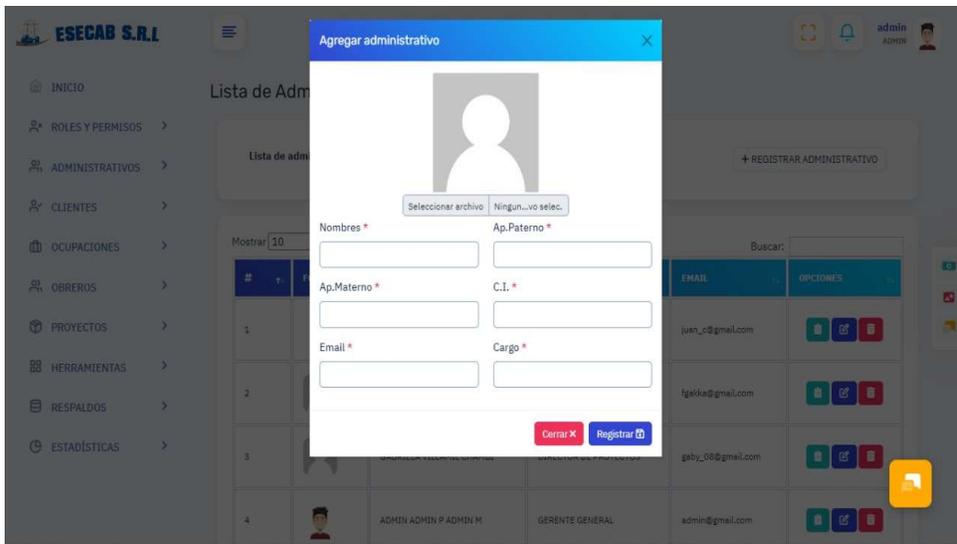
Menús administrativos.



Nota. Menús administrativos, con las opciones de crear, editar y eliminar.

Figura 41

Agregar administrativo.



Nota. Agregar administrativo, con las validaciones correspondientes
A continuación, se muestra la lista de clientes, con las opciones correspondientes.

Figura 42

Lista clientes.

Lista de Clientes

Mostrar 10 registros

REPORTE PDF REGISTRAR CLIENTES

#	RAZÓN SOCIAL	NOMBRE COMERCIAL	EMAIL	CELULAR	TÉLFONO	OPCIONES
1	MINISTERIO DE TELECOMUNICACIONES	MDETTBB	ministeriotelecomunica@gmail.com	75802141	22835438	[Iconos de edición y eliminación]
2	MINISTERIO DE OBRAS PÚBLICAS	MGBB	ministeriogobierno@gmail.com	75802141	22835438	[Iconos de edición y eliminación]
3	ALCALDIA DE EL ALTO	GAMEA	gameas@gmail.com	75802141	22835438	[Iconos de edición y eliminación]
4	MINISTERIO DE CULTURAS	MCBBBS	ministeriocultura@gmail.com	75802141	22835438	[Iconos de edición y eliminación]
5	MINISTERIO DE LA PRESIDENCIA	ASEFT	ministerio@gmail.com	75802141	22835438	[Iconos de edición y eliminación]

Mostrando 1 a 5 de 5 registros

Anterior 1 Siguiente

Nota. Lista clientes.

Figura 43

Agregar cliente.

Agregar cliente

Razón social * Nombre Comercial *

Email * Celular *

Teléfono *

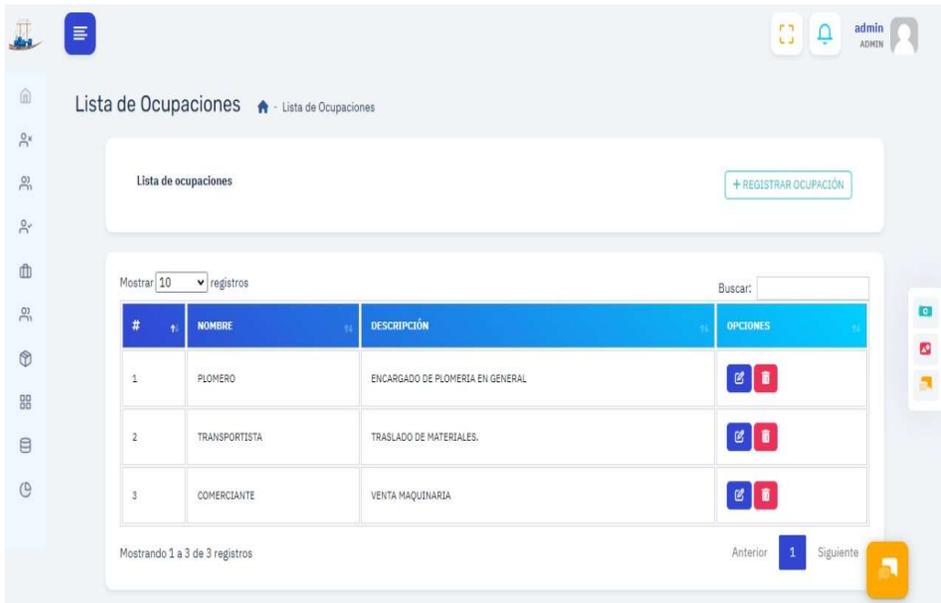
Cerrar X Registrar

Nota. Registro de cliente, formulario con las validaciones correspondientes.

A continuación, se muestra la lista de ocupaciones, con las opciones correspondientes.

Figura 44

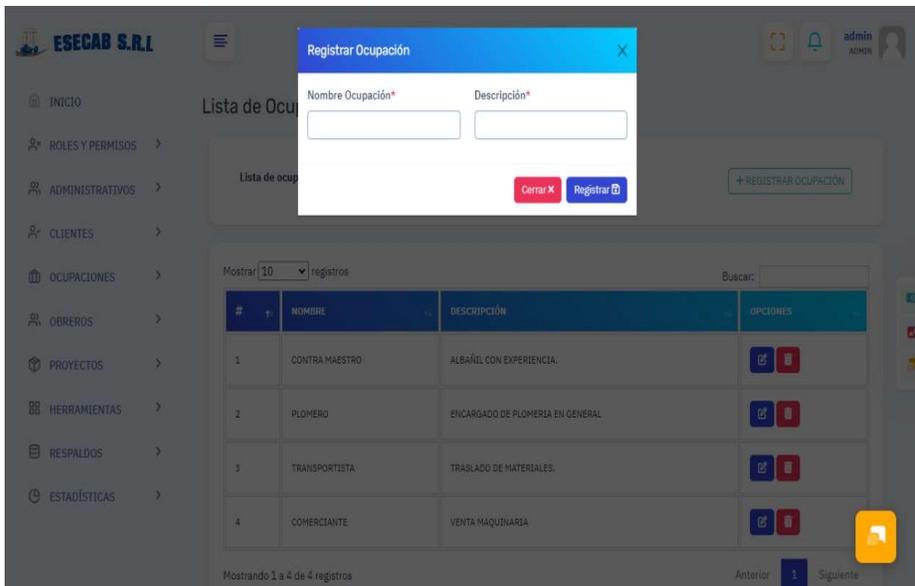
Lista de ocupaciones.



Nota. Lista de ocupaciones, con la opción de crear, editar y eliminar.

Figura 45

Registro de ocupación.

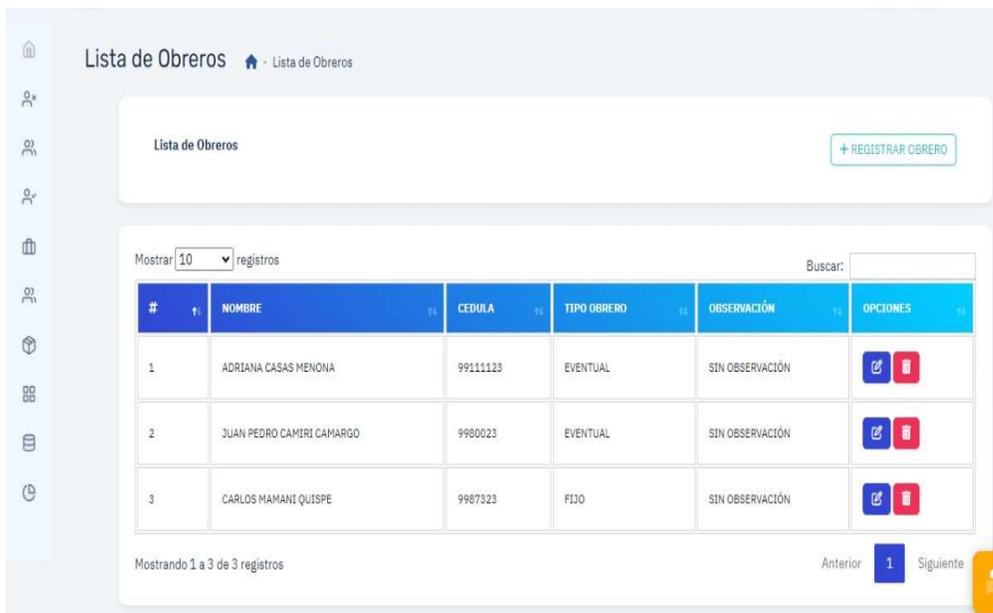


Nota. Registro de ocupación, con las validaciones correspondientes.

A continuación, se muestra la lista de obreros, con las opciones correspondientes.

Figura 46

Lista de obreros.



Lista de Obreros

+ REGISTRAR OBRERO

Mostrar 10 registros

Buscar:

#	NOMBRE	CEDULA	TIPO OBRERO	OBSERVACIÓN	OPCIONES
1	ADRIANA CASAS MENONA	99111123	EVENTUAL	SIN OBSERVACIÓN	 
2	JUAN PEDRO CAMIRI CAMARGO	9980023	EVENTUAL	SIN OBSERVACIÓN	 
3	CARLOS MAMANI QUISPE	9987323	FIJO	SIN OBSERVACIÓN	 

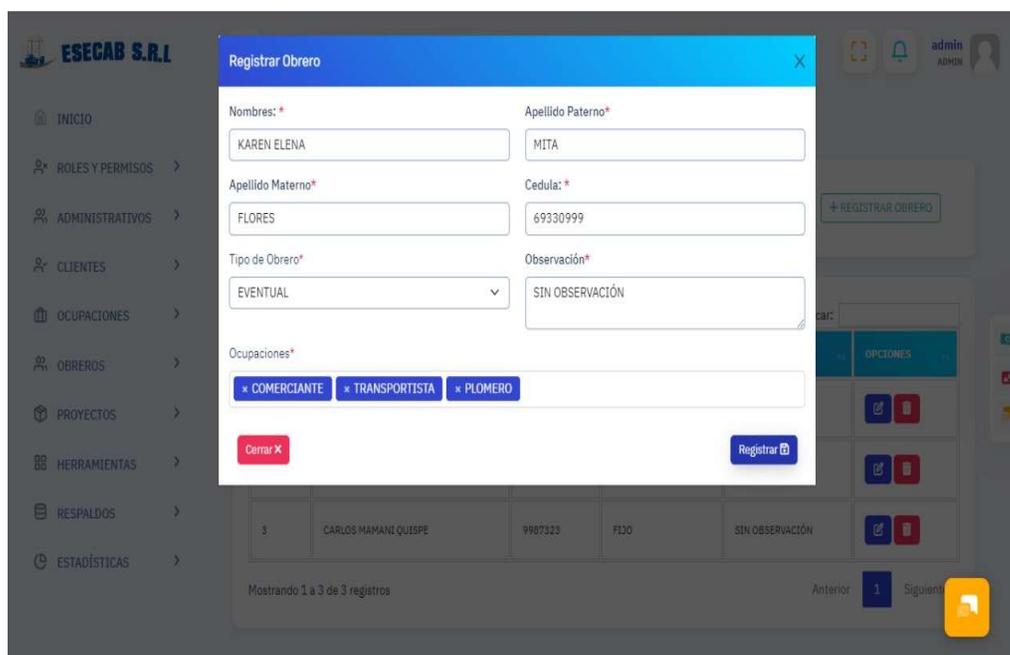
Mostrando 1 a 3 de 3 registros

Anterior 1 Siguiente

Nota. Lista de obreros.

Figura 47

Registro de obreros.



Registrar Obrero

Nombres*: KAREN ELENA Apellido Paterno*: MITA

Apellido Materno*: FLORES Cedula*: 69330999

Tipo de Obrero*: EVENTUAL Observación*: SIN OBSERVACIÓN

Ocupaciones*: COMERCIANTE TRANSPORTISTA PLOMERO

Cerrar X Registrar

Nota. Registro de obreros, con las validaciones correspondientes.

A continuación, se muestra la lista de proyectos, con los diferentes estados.

Figura 48

Lista de proyectos activos.

Lista de Proyectos

Lista de proyectos

REPORTE PDF REGISTRAR PROYECTO

PROYECTOS ACTIVOS PROYECTOS FINALIZADOS PROYECTOS CANCELADOS

Mostrar 10 registros

Buscar:

#	NOMBRE DE PROYECTO	TIPO DE PROYECTO	PROVINCIA	TELEFONO	ESTADO	OPCIONES
1	CANCHAS MUNICIPALES	PÚBLICO	Omasuyos	66646027	Actualizar	[Iconos de acciones]

Mostrando 1 a 1 de 1 registros

Anterior 1 Siguiente

Nota. Lista de proyectos activos.

Figura 49

Lista de proyectos finalizados.

Lista de Proyectos

Lista de proyectos

REPORTE PDF REGISTRAR PROYECTO

PROYECTOS ACTIVOS PROYECTOS FINALIZADOS PROYECTOS CANCELADOS

Mostrar 10 registros

Buscar:

#	NOMBRE DE PROYECTO	TIPO DE PROYECTO	PROVINCIA	TELEFONO	ESTADO	OPCIONES
Ningún dato disponible en esta tabla						

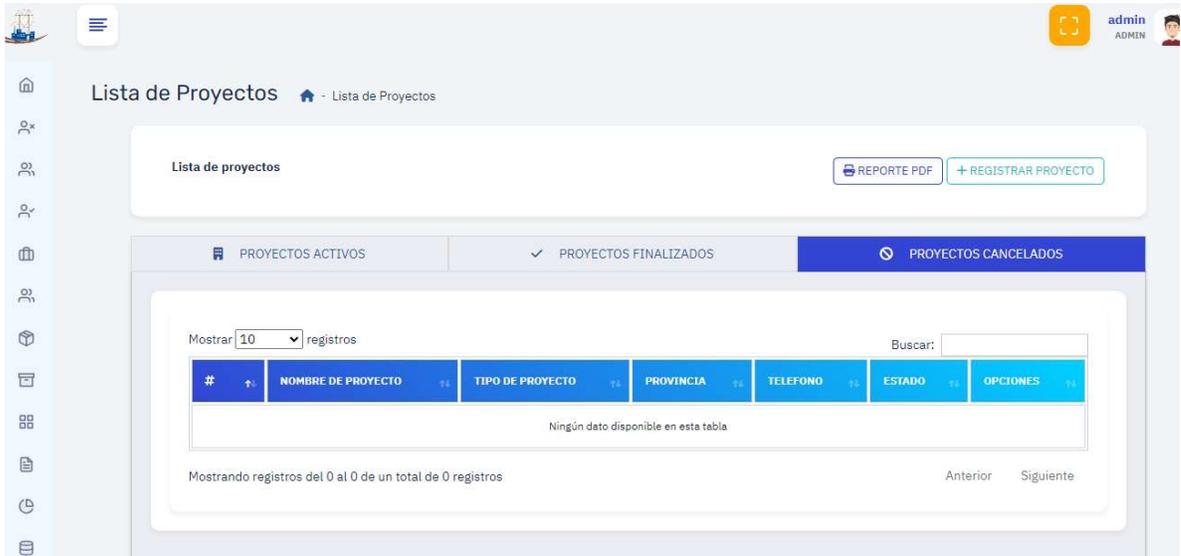
Mostrando registros del 0 al 0 de un total de 0 registros

Anterior Siguiente

Nota. Lista de proyectos finalizados

Figura 50

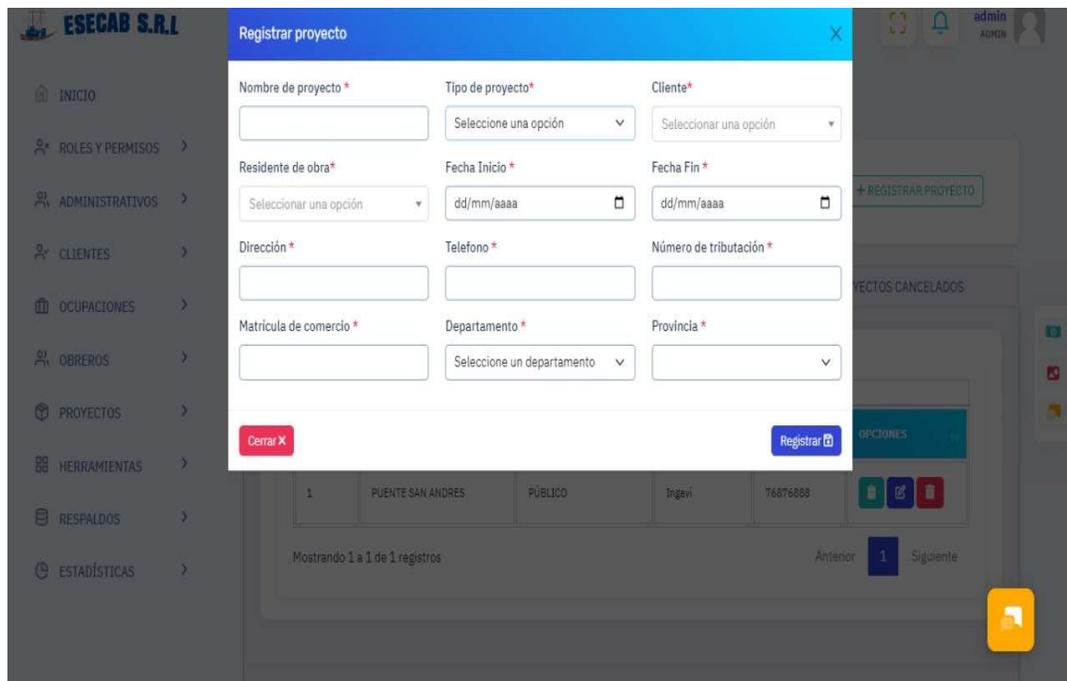
Lista de proyectos cancelados.



Nota. Lista de proyectos cancelados.

Figura 51

Registro de proyectos.



Nota. Registro de proyectos para que se realice el registro de los proyectos.

CAPITULO IV

4. CALIDAD, PRUEBAS, COSTOS Y SEGURIDAD DEL SOFTWARE

4.1. MÉTRICAS DE CALIDAD

Para asegurar que el proyecto se desarrolle se aplicaron las métricas de calidad, donde se calcularon los parámetros que cumple el sistema, que permita especificar ordenadamente dichas características y atributos.

Se pueden destacar varios modelos y criterios de calidad para productos de software, algunos de ellos podemos destacar como el ISO 25000.

Se pueden destacar varios modelos y criterios de calidad para productos de software, algunos de ellos podemos destacar como el ISO 25000, del cual se aplicará para evaluar la calidad del sistema los siguientes:

4.1.1. Funcionalidad

Punto función

El desarrollo de esta técnica de estimación de software, toma en cuenta 5 características:

- **Número de entradas de usuario:** El número de entradas en la que el usuario proporciona datos al sistema.

Tabla 22

Número de Entradas de usuario.

Entradas de Usuario		
1	Administración de usuarios	1
2	Administración de técnicos	1
3	Administración de usuarios	28
Total		30

Nota. El número de entradas de los usuarios al sistema.

- **Número de salidas de usuario:** El número de salidas que el sistema proporciona al sistema: mensajes, notificaciones, reportes, alertas, etc.

Tabla 23

Número de Salidas Usuario.

Salidas de Usuario	
	Administración de usuarios
	Administración de técnicos
	Administración de usuarios
	Total

Nota. Número de Salidas Usuario del sistema.

- **Número de peticiones de usuario:** Es el número de entradas interactivas que genera una respuesta del software como salidas interactivas.

Tabla 24

Número de Peticiones del Usuario.

Peticiones del Usuario		
1	Administración de usuarios	3
2	Administración de técnicos	3
3	Administración de usuarios	6
	Total	12

Nota. Número de peticiones al sistema.

- **Número de Archivos:** Es el número de archivos, se toma en los grupos lógicos de datos, o archivos independientes.
- **Número de interfaces externas:** Es la suma de todas las interfaces legibles por la máquina.

Tabla 25*Parámetros de medición de punto función.*

Parámetros de medición	Cuenta
Número de entradas de Usuario	30
Número de salidas de usuario	26
Número de peticiones de usuario	12
Número de archivos	20
Número de interfaz externa	2

Nota. Son Parámetros de medición de punto función

Habiendo realizado el conteo de los parámetros de función en la tabla se hace el cálculo de los puntos de función sin ajustar en la siguiente tabla:

Tabla 26*Punto función sin ajustar.*

Parámetro de medición	Cuenta	Factor	Total
Número de entradas de Usuario	24	6	180
Número de salidas de usuario	26	7	182
Número de peticiones de usuario	12	6	72
Número de archivos	20	15	300
Número de interfaz externa	2	10	20
Cuenta total			754

Nota. Punto función sin ajustar.

En la tabla se puede apreciar la cuenta total de los puntos de función si ajustar, para determinar los valores ajustados se hace el empleo de los factores de complejidad listados a continuación:

Tabla 27

Factores de Complejidad.

Importancia		2	4	6	8	1	
		0	0	0	0	0	
		%	%	%	%	%	
Factor	Escola	No influencia	Incidental	Moderado	Medio	Significativo	Esencial
		0	1	2	3	4	5
	¿Requiere el sistema copias de seguridad y de recuperación fiable?						X
	¿Se requiere comunicación de datos?						X
	¿Existe funciones de procesos distribuidos?						X
	¿Es crítico el rendimiento?					X	
	¿El sistema web será ejecutado el SO Actual?						X
	¿Se requiere una entrada interactiva para el sistema?					X	
	¿Se requiere que el sistema tenga entradas a datos con múltiples ventanas?					X	
	¿Se actualiza los archivos de forma interactiva?					X	
	¿Son complejas las entradas, salidas, los archivos o las peticiones?						X
	¿Es complejo el procesamiento interno del sistema?					X	
	¿Se ha diseñado el código para ser reutilizado?						X
	¿Se ha diseñado el sistema para facilitar al usuario el trabajo y ayudarlos a encontrar la información?					X	
TOTAL							

Nota. Factores de Complejidad.

Para el cálculo del punto de función ajustado se utiliza la siguiente formular:

$$PF = \text{Conteo total}[0.65 + 0.01 * \sum fi]$$

Donde:

$$\sum (fi)$$

- Sumatoria de los valores de los factores de ajuste.

Se tiene:

$$\sum(fi)= 54$$

$$PF = 754 * [0.65 + 0.01 * 54]$$

$$PF = 897.26$$

Considerando el máximo valor en la sumatoria de los factores de ajuste $\sum(fi) = 70$, se calcula el siguiente valor:

$$PFmax = \text{Cuenta Total} [0.65 + (0.01 * \sum fi)]$$

$$PFax = 754 * [0.65 + (0.01 * 70)]$$

$$PF max = 1017.9$$

El cálculo de la funcionalidad se obtiene de la relación entre el valor máximo con el valor obtenido del punto de función ajustado:

$$\text{Funcionalidad} = \frac{PF}{PF max} * 100$$

$$\text{Funcionalidad} = \frac{854.42}{969.3}$$

$$\text{Funcionalidad} = 0.8814 * 100\%$$

$$\text{Funcionalidad} = \mathbf{88.14 \%}$$

Por lo tanto, la funcionalidad del sistema desarrollado llega a ser del 88.14 %, esto quiere decir que el sistema tiende a funcionar un 88.14 %, sin riesgo de fallar, y un 11.86 % de que el sistema colapse.

4.1.2. Confiabilidad

Es la capacidad de un sistema para mantener su nivel de ejecución bajo condiciones normales en un periodo de prueba establecido, cuyo criterio son la madurez, recuperabilidad, tolerancia a fallos y cumplimiento de fiabilidad. Para determinar la confiabilidad del sistema especificamos el tiempo en que empieza a funcionar, a partir de allí se realiza las observaciones hasta un tiempo determinado.

Donde se toma en cuenta que:

$$P(T \leq t) = F(t) \Rightarrow \text{Probabilidad de fallas}$$

$$P(T \leq t) = 1 - F(t) \Rightarrow \text{Probabilidad de trabajo sin fallas}$$

Para calcular la confiabilidad del sistema se toma en cuenta la siguiente función:

$$F(t) = f * e^{-\mu * t}$$

Donde:

- f = funcionalidad del sistema.
- μ = Probabilidad de error del sistema.
- t = tiempo de prueba del sistema.

En un periodo de 20 días como tiempo de prueba se define de cada 10 ejecuciones 1 falla.

$$F(t) = 0.88 * e^{-\frac{1}{10} * 20}$$

$$F(t) = 0.1191 * 100\%$$

$$F(t) = 11.91 \%$$

Reemplazando:

$$P(T \leq t) = 0.1191 = 11.91\%$$

$$P(T \leq t) = 1 - 0.1191$$

$$P(T \leq t) = 0.8809 * 100\%$$

$$P(T \leq t) = \mathbf{88.09\%}$$

Por lo tanto, la confiabilidad del sistema desarrollado es del 88.09%. en un periodo de 20 días como el tiempo de prueba.

4.1.3. Usabilidad

Para calcular la usabilidad de un sistema se toma en cuenta el factor humano, para conocer si el sistema cumple con los requisitos establecidos para el usuario se realiza la evaluación del sistema mediante encuestas a los usuarios del sistema.

Para determinar la usabilidad del sistema, se utiliza la siguiente formular:

Formula:

$$FU = \left[\frac{\sum x_i}{n} * 100 \right]$$

Donde:

- *xi = Valores de evaluación de cada pregunta.*
- *n = Numero de preguntas.*

Se toma en cuenta la siguiente tabla:

Tabla 28

Parámetros de medición de usabilidad.

Escala	Valor
Muy bueno	5
Bueno	4
Regular	3
Malo	2
Pésimo	1

Nota. Parámetros de medición de usabilidad.

Tabla 29*Usabilidad del sistema.*

Pregunta	Si	Resultado xi
¿Puede Utilizar con facilidad el sistema?	9	0.9
¿Puede Controlar operaciones que el sistema solicita?	8	0.8
¿Las Respuestas del sistema son complicadas?	2	0.8
¿El Sistema permitió la retroalimentación de información?	9	0.9
¿El sistema cuenta con interface agradable a la vista?	9	0.9
¿La respuesta del sistema es satisfactoria?	9	0.9
¿Le parece complicada las funciones del sistema?	8	0.8
¿Se hace difícil o dificultoso aprender a manejar el sistema?	1	0.9
¿Los resultados que proporciona el sistema facilitan el trabajo?	10	1
¿Durante el uso del sistema se produjo errores?	2	0.8
TOTAL		8.7

Nota. Usabilidad del sistema.

Con los resultados obtenidos del cuestionario obtenemos calculamos la usabilidad:

$$FU = \left[\frac{8.7}{10} * 100 \right]$$

$$FU = [0.87 * 100]$$

$$FU = 87\%$$

Con esto concluimos que la usabilidad del sistema es del 87%.

4.1.4. Eficiencia

La eficiencia mide si el sistema emplea óptimamente los recursos del sistema. Para calcular el grado de eficiencia que tiene el sistema se toma en cuenta la siguiente escala:

Tabla 30

Escala de valores de eficiencia.

Escala	Valor
Excelente	5
Bueno	4
Aceptable	3
Deficiente	2
Pésimo	1

Nota. Escala de valores de eficiencia.

Para valorar la eficiencia del sistema se tiene la siguiente tabla:

Tabla 31

Valoración de la eficiencia del sistema.

Nº	PREGUNTA	PORCENTAJE
	¿La distribución y estilo de la interfaz permite que un usuario introduzca con eficiencia las operaciones y la información?	5
	¿Una secuencia de operaciones (o entrada de datos) puede realizarse con facilidad de movimientos?	4

¿Los datos de salida están presentados de modo que se entienden de inmediato?	4
¿Las operaciones jerárquicas están organizadas de manera que minimizan la navegación del usuario para hacer que alguna se ejecute?	4
¿Procesa y responde adecuadamente cuando realiza alguna consulta o búsqueda?	5
Total	22

Nota. Valoración de la eficiencia del sistema.

Para calcular la eficiencia tenemos la siguiente formula:

$$E = \frac{\sum x_i}{n} * \frac{100}{n}$$

Donde:

- $\sum xi$ = Sumatoria de los valores de eficiencia
- n = Numero de preguntas

Reemplazando se tiene lo siguiente:

$$E = \frac{22}{5} * \frac{100}{5}$$

$$E = 88 \%$$

Con esto concluimos que la eficiencia del sistema es del 88%.

4.1.5. Mantenibilidad

Permite medir el esfuerzo necesario para realizar modificaciones del sistema, ya sea por corrección de errores o por el incremento de módulos o funciones dentro del sistema. El estándar IEE94 sugiere la implementación del índice de madurez del sistema que proporciona un indicador de estabilidad del sistema desarrollado.

De esto modo tenemos que la siguiente ecuación para el índice de madurez del sistema:

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt}$$

Donde:

- **Mt:** Número de módulos total de la versión actual
- **Fa:** Número de módulos de la versión actual que se añadieron.
- **Fc:** Número de módulos de la versión actual que se cambiaron.
- **Fd:** Número de módulos de la versión anterior que se eliminaron en la versión actual.

Tabla 32

Valores para la mantenibilidad.

Descripción	Valor
Mt	5
Fa	0
Fc	1
Fd	0

Nota. Valores para la mantenibilidad.

Reemplazando en la ecuación se tiene:

$$IMS = \frac{[5 - (0 + 1 + 0)]}{5}$$

$$IMS = 0.8 * 100\%$$

$$IMS = 80\%$$

Se concluye que el índice de mantenibilidad del sistema desarrollado es del 80%.

4.1.6. Portabilidad

Es la capacidad de un software para ser transferido de un ambiente a otro donde se considera lo siguiente:

- **Adaptabilidad:** Evalúa la oportunidad para adaptar el software a diferentes ambientes sin necesidad de aplicarle modificaciones.

- **Facilidad de Instalación:** Es el esfuerzo necesario para instalar el software en un ambiente determinado.
- **Conformidad:** Permite evaluar si el software se adhiere a estándares o convenciones relativas a portabilidad.
- **Capacidad de reemplazo:** Se refiere a la oportunidad y el esfuerzo usado en sustituir el software por otro producto con funciones similares.

El sistema está desarrollado para ser implementado en cualquier servidor que tenga

instalado los servicios de apache y las herramientas que sean necesarias.

En el caso de del sistema desarrollado, al estar bajo el marco de trabajo Laravel, se cuenta con la facilidad de configurar el software para casi la mayoría de los ambientes.

En cuanto a la ejecución, al estarse aplicando una tecnología web, el sistema puede ser ejecutado en cualquier dispositivo que cuente con conexión a internet y un navegador web.

Para obtener la portabilidad, se tiene la siguiente formula:

$$Portabilidad = 1 - \left(\frac{\text{numero de dia para portar el sistema}}{\text{numero de dias para implementar el sistema}} \right)$$

Reemplazando la formula se obtiene el siguiente resultado:

$$Portabilidad = 1 - \left(\frac{1}{7} \right)$$

$$Portabilidad = 0.85 * 100\%$$

$$Portabilidad = 86\%$$

Resultados:

De acuerdo a los resultados obtenidos se puede establecer la calidad total del sistema en base a los parámetros medidos anteriormente. La calidad está

directamente relacionada con el grado de satisfacción con el usuario que ingresa al sistema.

A continuación de acuerdo a las operaciones realizadas y resultados obtenidos, obtenemos el siguiente resumen:

Tabla 33

Resultados

Características	Resultados
Funcionalidad	88 %
Confiabilidad	88 %
Usabilidad	87%
Eficiencia	88%
Mantenibilidad	80%
Portabilidad	86%
Evaluación total	86 %

Nota. Resultados

Evaluación de calidad total del sistema es de un 86 %.

4.2. ANÁLISIS DE COSTOS

Para realizar el presente proyecto es necesario planificar y estimar los costos durante y hasta la finalización del mismo. Es útil para estimar el costo total del sistema de la empresa ESECAB SRL se tomarán en cuenta los siguientes costos:

COCOMO II permite hacer estimaciones en base al tamaño del software, y un conjunto de factores de coste y de escala. COCOMO II posee tres modelos:

- **ORGÁNICO:** Proyectos relativamente sencillos, menores a 5000 líneas de código, implica procesamiento de datos, uso de la base de datos se focaliza en transacciones y recuperación de datos.

- **SEMIACOPLADO:** Proyectos intermedios en complejidad y tamaño. La experiencia en este tipo de proyectos es variable y las restricciones intermedias.
- **EMPOTRADO:** Proyectos bastantes complejos, en los que apenas se tiene experiencia y en un entorno de gran innovación técnica. Para la estimación de costos de este proyecto se empleará el submodelo post arquitectura.

Para calcular el esfuerzo, necesitaremos hallar la variable KLDC (kilo-líneas de código). Este proyecto se implementa 8504 Líneas de Código. Aplicando Conversiones se tiene.

$$LCD = 8504$$

$$KLCD = \frac{LCD}{1000}$$

$$KLCD = \frac{8504}{1000}$$

$$KLCD = 8,504$$

Es un modelo Intermedio y semi acoplado.

Tabla 34

Coefficientes de a,b,c,d de COCOMO II.

Proyecto Software	A	b	c	D
Orgánico	3.2	1.05	2.5	0.38
Semi acoplado	3.0	1.12	2.5	0.35
Empotrado	2.8	1.20	2.5	0.32

Nota. Coeficientes de a,b,c,d de COCOMO II.

Debido a que el proyecto no supera las 50000 líneas de código se utilizaran los coeficientes para proyectos semi-acoplado. A continuación, se muestra las ecuaciones correspondientes para el coeficiente seleccionado de acuerdo a las líneas de código, para calcular el costo de software:

Ecuaciones de Cocomo II.

Variable	Ecuación	Tipo / Unidad
Esfuerzo requerido por el proyecto	$E = a * (KLDC)^b * FAE$	Persona/mes
Tiempo requerido por el proyecto	$T = c * (E)^d$	Mes
Número de personas requeridos para el proyecto	$NP = \frac{E}{T}$	Personas
Costo Total	$CT = SueldoMes * NP * T$	\$us

Nota. Ecuaciones de Cocomo II.

Para hallar los valores de FAE, se utilizará la tabla de atributos multiplicadores.

Tabla 36.

Cálculo de Atributos FAE.

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0 , 7 5	0 , 8 8	1 , 0 0	1 , 1 5	1 , 4 0	
Tamaño de Base de datos		0 , 9 4	1 , 0 0	1 , 0 8	1 , 1 6	

Complejidad	0 , 7 0	0 , 8 5	1 , 0 0	1 , 1 5	1 , 3 0	1 , 6 5
Atributos de hardware						
Restricciones de tiempo de ejecución			1 , 0 0	1 , 1 1	1 , 3 0	1 , 6 6
Restricciones de memoria virtual			1 , 0 0	1 , 0 6	1 , 2 1	1 , 5 6
Volatilidad de la máquina virtual		0 , 8 7	1 , 0 0	1 , 1 5	1 , 3 0	
Tiempo de respuesta		0 , 8 7	1 , 0 0	1 , 0 7	1 , 1 5	
Atributos de personal						
Capacidad de análisis	1 , 4 6	1 , 1 9	1 , 0 0	0 , 8 6	0 , 7 1	
Experiencia en la aplicación	1 , 2 9	1 , 1 3	1 , 0 0	0 , 9 1	0 , 8 2	
Calidad de los programadores	1 ,	1 ,	1 ,	0 ,	0 ,	

	4	1	0	8	7	
	2	7	0	6	0	
Experiencia en la máquina virtual	1	1	1	0		
	,	,	,	,		
	2	1	0	9		
	1	0	0	0		
Experiencia en el lenguaje	1	1	1	0		
	,	,	,	,		
	1	0	0	9		
	4	7	0	5		
Atributos del proyecto						
Técnicas actualizadas de programación	1	1	1	0	0	
	,	,	,	,	,	
	2	1	0	9	8	
	4	0	0	1	2	
Utilización de herramientas de software	1	1	1	0	0	
	,	,	,	,	,	
	2	1	0	9	8	
	4	0	0	1	3	
Restricciones de tiempo de desarrollo	1	1	1	1	1	
	,	,	,	,	,	
	2	0	0	0	1	
	2	8	0	4	0	

Nota. Cálculo de Atributos FAE.

Por tanto, nuestro Factor de ajuste será

$$FAE = 1.15 * 1.00 * 1.15 * 1.00 * 1.00 * 0.87 * 1.00 * 0.86 * 0.82 * 0.86 * 0.90 * 0.95 * 0.82 * 0.91 * 1.00 = 0.4452$$

Aplicando y reemplazando valores a la fórmula de esfuerzo, se tiene:

$$E = a * KLCD^b * FAE(\text{persona/mes})$$

$$E = 3 * 8,504^{1.05} * 0.4452 (\text{persona/mes})$$

$$E = 12.69 (\text{persona/mes})$$

Cálculo del Tiempo

$$T = c * Esfuerzo^d(\text{meses})$$

$$T = 2.5 * 12.69^{0.35}(\text{meses})$$

$$T = 6.08 (\text{meses}) \text{ esto equivale 6 meses}$$

Cálculo del personal requerido

$$NP = \frac{E}{T} \text{ personas}$$

$$NP = \frac{12.69}{6.08} \text{ personas}$$

$$NP = 2.08 \cong 2 \text{ persona}$$

Cálculo de Costo Total

$$CT = SueldoMes * NP * T$$

$$CT = 1000 * 2 * 6$$

$$CT = 12000 \$us$$

En resumen, se requiere 2 personas estimando un trabajo de 6 meses y con costo total de 12000 \$us. Equivalente en bolivianos a 829,740.60 Bs.

4.3. SEGURIDAD

Uno de los aspectos más importantes a tomar en cuenta en este proyecto, es la implementación de normas de seguridad. La norma ISO 27001 evalúa y rectifica el cumplimiento de las normas, así como la mejora continua en base a un conjunto de controles que permiten reducir el riesgo de sufrir incidentes de seguridad en el funcionamiento del sistema dentro de la institución, para ello se toman en cuenta los siguientes tipos de seguridad:

4.3.1. Seguridad Lógica

Para la seguridad del sistema se consideran las siguientes precauciones.

- **Copias de Seguridad:** Las copias de seguridad son respaldos que se tiene de las bases de datos, estos se realizan de manera automática.
- **Autenticación:** Para la seguridad de datos del sistema se tienen registrado

de nombre de usuario y contraseña de acceso encriptadas, según su nivel de acceso pueda realizar actividades en el sistema.

Para asegurar la seguridad de la información dentro del sistema se aplicaron los controles mediante el uso de sesiones y su verificación.

En el sistema se tiene 3 niveles de usuario, cada uno con diferentes privilegios.

También la implementación del código captcha

- **Encriptación:** El sistema permite encriptar las rutas y las contraseñas de los usuarios con hash.
- **Base de datos:** En cuanto a la forma de resguardo se realizó los siguientes puntos: una vez que se accedió a los registros o se realizó las diferentes acciones dentro del sistema se hace la desconexión de la base de datos para que no se tenga ningún problema con terceros. Además los respaldos o back-up de la base de datos se deberá realizar.

La información en una institución es muy valiosa, por tanto, su resguardo es fundamental, la conexión a la base de datos y el cierre de la conexión es de forma automática.

4.3.2. Seguridad Física

- **Seguridad física y del entorno:** Se prevé que los equipos donde se almacenan los datos y el sistema se encuentran resguardados en lugares seguros, con las condiciones necesarias para funcionar con el menor número de riesgos con seguridad de acceso.
- **Equipamiento:** Se tiene una adecuada protección física y mantenimiento de los servidores donde se almacenan los sistemas de la empresa.
- **Acceso físico:** Se restringe el acceso físico a las áreas críticas a toda persona no autorizada, para reducir el riesgo de accidentes fraudulentos y robos de contraseñas.

4.4. PRUEBAS AL SOFTWARE

4.4.1. Prueba de caja blanca

En este tipo de prueba, observamos 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:

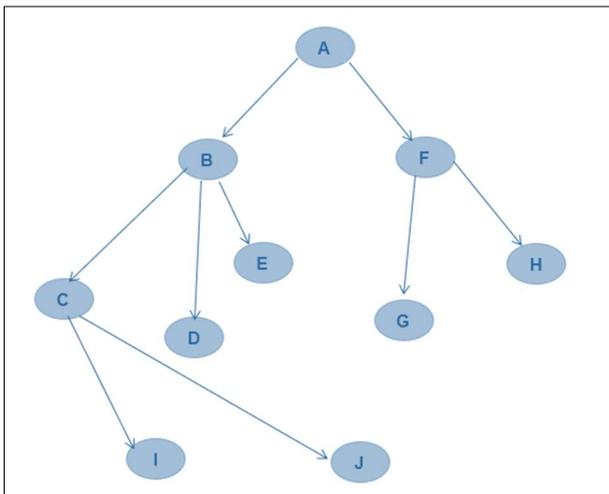
- Se realizarán las pruebas utilizando el conocimiento del funcionamiento interno del código.
- Las pruebas de caja blanca solo se pueden realizar por programadores.

La aplicación del caso de prueba de caja blanca se lo realiza utilizando métrica de complejidad dicromática el cual brinda la cantidad aproximada de casos de prueba que se deben aplicar en el código fuente como se muestra en la siguiente figura.

De forma general, se debe seguir:

Figura 52

Prueba de caja blanca



Nota. Prueba de caja blanca, del sistema

Donde:

- Inicio del sistema (A)
- Módulo de administración de usuarios (B)

- Creación de usuarios (C)
- Cambiar de estado (D)
- Listado de usuarios (E)
- Módulo de reportes (F)
- Listado de proyectos (G)
- Reporte general (H)
- Reporte específico (I)
- Corroboración de proyectos (J)

Después de realizar el grafico se procede a determinar la complejidad ciclo matica del grafo mediante el siguiente formula:

$$V(G) = A - N + 2$$

Donde:

$$A = 8(\text{Aristas})$$

$$N = 7(\text{Nodos})$$

Por tanto:

$$V(G) = 8 - 7 + 2$$

$$V(G) = 3$$

Determinar los caminos linealmente independientes los caminos que deben ser probados.

Camino 1: A – B – C – I

Camino 2: A – B – C – J

Camino 3: A – B – D

Camino 4: A – B – E

Camino 5: A – F – H

Camino 6: A – F – G

Presentar los casos de prueba para forzar ejecución de cada camino. Esta última condición establece que para la ejecución de ciertos caminos.

Camino 1: El usuario podrán ingresar a la página principal revisar los módulos de usuarios, ver los reportes y la impresión de formularios.

Camino 2: El usuario puede ver los reportes y corroborar proyectos en qué estado se encuentran.

Camino 3: El usuario ingresa y administra a los usuarios, puede cambiar los estados.

Camino 4: El usuario ingresa y administra a los usuarios, los puede listar.

Camino 5: El usuario ingresa y administra puede generar los reportes generales de los proyectos.

Camino 6: El usuario cuando ingresa y puede generar los reportes específicos.

4.4.2. Prueba de caja negra

Para hacer las pruebas de caja negra, se hace el uso de la técnica de partición de equivalencia que es una de las más efectivas para analizar los valores válidos e inválidos de las siguientes interfaces:

Tabla 37

Caso de prueba de Ingreso al sistema.

Caso de prueba de ingreso al sistema	
Código:	Caso de Prueba 01
Descripción de la Prueba	Ingresar al sistema mediante usuario y contraseña, después de la validación correspondiente.
Condiciones de ejecución	El usuario debe encontrarse registrado previamente.
Pasos de Ejecución	El sistema muestra la pantalla de autenticación, donde pide usuario y contraseña para ingresar.
Resultado esperado	El sistema valida los datos ingresados y permite al usuario el ingreso de acuerdo a su rol asignado.

Evaluación de prueba	Se realiza el ingreso al sistema con la verificación previa, en caso de que la verificación falle, el usuario es devuelto a la página de inicio
-----------------------------	---

Nota. Caso de prueba de Ingreso al sistema.

Tabla 38.

Caso de prueba de registro de usuario.

Caso de prueba de registro de usuario	
Código:	Caso de Prueba 02
Descripción de la Prueba	Registro de usuarios
Condiciones de ejecución	El usuario debe haber iniciado sesión en el sistema.
Pasos de Ejecución	<ul style="list-style-type: none"> ✓ El sistema muestra la pantalla para buscar personas registradas en la universidad. ✓ El usuario encuentra la persona y presiona la opción de asignar usuario. ✓ El sistema le muestra un formulario para la creación de nuevo usuario. ✓ El sistema valida los datos ingresados y crea al nuevo usuario.
Resultado esperado	El sistema valida y crea el nuevo usuario
Evaluación de prueba	Se realizó la creación de usuario de manera efectiva.

Nota. Caso de prueba de registro de usuario.

4.4.3. Prueba de estrés

Las pruebas de estrés se centran en someter la plataforma a situaciones que superen su capacidad normal de procesamiento, con el objetivo de identificar posibles debilidades, errores o fallos en el sistema. Estas pruebas ayudan a determinar si la plataforma puede manejar volúmenes de usuarios concurrentes,

cargas de trabajo intensas o picos repentinos de tráfico.

Para esta prueba se utilizó un servidor de los de la Universidad con algunas variaciones en el Sistema Operativo donde se obtuvieron resultados que van con el rendimiento óptimo que exigen a la plataforma

- Usuarios establecidos: 50
- Tiempo de prueba: 5 min
- Solicitudes realizadas: 987
- Fallas HTTP: 0
- Pico de solicitud: 8 (Solicitudes por segundo)

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

El objetivo general del proyecto se ha logrado satisfactoriamente con el desarrollo de la Plataforma web para administración y control de proyectos de construcción en obras civiles utilizando las herramientas de diseño y desarrollo propuestos para la culminación del proyecto.

- Se elaboró una base de datos confiable y segura para el almacenamiento de toda la información que conlleva la elaboración la constructora.
- Se logró realizar un análisis del proceso, lo que permitió diseñar los módulos del sistema de manera ordenada para cada proceso.
- Se realizó una ingeniería de requerimientos para la recolección de datos del sistema para tener datos precisos.
- Los diseños de interfaz gráfica del sistema fueron diseñados de forma amigable, donde los usuarios realizan los registros con facilidad mostrando una información rápida.
- El sistema permite generar reportes de acuerdo al requerimiento de los usuarios del sistema.

5.2. RECOMENDACIONES

A partir del presente proyecto se proponen las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema.

- Se recomienda capacitar a los nuevos administradores para que puedan realizar operaciones del sistema y así poder tener mejor control de los proyectos que ofrece el sistema.
- El Mantenimiento del software en determinados periodos de tiempo, tomando en cuenta las normas preestablecidas.
- Se recomienda cambiar continuamente las contraseñas para la seguridad del sistema y proteger el acceso a personas ajenas.
- Cuando se requiera la ampliación y creación de nuevos módulos, se

recomienda primero revisar la documentación para tomar una buena decisión, ya que el sistema presenta elementos reutilizables para el desarrollo de nuevos módulos.

Bibliografía

- acensTechnologies*. (2016). Obtenido de <https://www.acens.com/wp-content/images/2016/10/bootstrap-framework-acens-wp.pdf>
- Alicante. (18 de octubre de 2021). *Servicio de informatica*. Obtenido de <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Anonimo. (2011). *Capacitacion y guia para el desarrollo de Software*. Obtenido de <http://materias.fi.uba.ar/7548/PruebasSoftware.pdf>
- Arango Cordoba , J. F. (2016). *Laravel framework para la visualización de variables de entorno en el sector agrícola*. CALI .
- Arimetrics*. (12 de agosto de 2021). Obtenido de <https://www.arimetrics.com/glosario-digital/javascript>
- Asana. (19 de enero de 2013). Obtenido de <https://asana.com/es/resources/project-controls>
- B., G. (11 de ENERO de 2023). *¿QUÉ ES AJAX Y CÓMO FUNCIONA?* Obtenido de HOSTINGER TUTORIALES: <https://www.hostinger.es/tutoriales/que-es-javascript-introduccion-basica/>
- Baena, G. R., Joel Coronado, E., & Mendoza M., R. (05 de 06 de 2019). *Enumed.Net*. Obtenido de Enumed: <http://www.eumed.net/rev/ce/2019/norma-iso-eic.html>
- Baker. (2009).
- Barrera, A. (2021). *next U*. Obtenido de <https://www.nextu.com/blog/que-es-json/>
- Bustos, G. (29 de abril de 2021). *Hostinger Tutoriales*. Obtenido de ¿Qué es Apache? Descripción completa del servidor web Apache: <https://www.hostinger.es/tutoriales/que-es-apache/>

- Cardenas, O. (26 de mayo de 2016). *google sites*. Obtenido de Ingenieria WEB:
<https://sites.google.com/site/ingenieriaweb23/home>
- Chavez. (2012). *Ingenieria Web*. 30-50.
- Chavez Quispe, J. (2014). *Sistema web de gestión de proyectos*.
- COCOMO. (28 de enero de 2013). Obtenido de <https://es.wikipedia.org/wiki/COCOMO>
- Contreras Rivas, D. A., & Lara Ávila, B. A. (2015). *Control de presupuesto de obras civiles basado en diseño web orientado a las empresas constructoras*. Guayaquil.
- Da Silva, F., & Nuñez, G. (2018). En *La era de las plataformas digitales y el desarrollo de los mercados de datos* (pág. 9). Naciones Unidas: CEPAL.
- Desarrollo web*. (28 de julio de 2020). Obtenido de <https://desarrolloweb.com/articulos/que-es-mvc.html>
- EcuRed*. (s.f.). Obtenido de <https://www.ecured.cu/MagicDraw>
- Estrada, J. G. (03 de agosto de 2021). *Desarrollo Web con PHP y MySQL*. Obtenido de <https://cupdf.com/document/desarrollo-web-con-php-y-mysql-52.html>
- Flores Fernandez, H. A. (2010). *ARQUITECTURA MULTICAPA MEDIANTE AJAX Y ORM MULTILAYER ARCHITECTURE DRIVEN AJAX AND ORM*. BOGOTA: VINCULOS.
- Gardey, J. P. (16 de 05 de 2021). *Definicion.de*. Obtenido de <https://definicion.de/web-2-0/>
- Harb Hoecker, R. A. (s.f.). *Mysql my struct Query Language*. Chile.
- Instituto de Tecnologia de comumbia Britanica*. (19 de septiembre de 2019). Obtenido de <https://codeigniter.com/userguide3/general/welcome.html>

IONOS. (16 de marzo de 2020). Obtenido de <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/codeigniter-framework-php-rapido-y-versatil/>

ISO, E. (2021). Obtenido de <https://danieltrabajoiso25000.blogspot.com/2020/11/ventajas-y-desventajas-norma-iso-25-000.html>

Lopez, J. (2010). Ingenieria WEB. *Ingenieria de software*.

Lozada, A. (18 de agosto de 2020). *horizonte*. Obtenido de <https://soyhorizonte.com/blog/ventajas-y-desventajas-de-usar-bootstrap/>

Ludwig, M. (2014). *UWE*. Obtenido de UWE: <https://uwe.pst.ifi.lmu.de/teachingTutorialProcessSpanish.html>

Magicdraw. (31 de octubre de 2021). Obtenido de https://www.magicdraw.com/main.php?ts=download_demo&cmd_go_to_login=1&menu=download_demo&back_cmd=cmd_show

McFarland. (29 de agosto de 2014). Obtenido de <https://www.um.es/docencia/barzana/DAWEB/Lenguaje-de-programacion-JavaScript-1.pdf>

Miss Juarez , F. G. (2020). *Método Cocomo II*. GUATEMALA.

MySql.com. (13 de abril de 2021). Obtenido de <https://www.mysql.com/>

Neve. (2 de noviembre de 2019). Obtenido de <https://www.losmejoresrecursos.online/proyecto-construccion/>

Parada, M. (31 de octubre de 2019). *OpenWebinars*. Obtenido de <https://openwebinars.net/blog/que-es-jquery/>

Platzi. (9 de septiembre de 2018). Obtenido de <https://platzi.com/blog/pruebas-esenciales-para-evaluar-el-rendimiento-de-software/>

- Porto, J. P., & Gardey, A. (2019). Obtenido de <https://definicion.de/json/>
- Porto, J. P., & Gardey, A. (5 de noviembre de 2019). Obtenido de <https://definicion.de/jquery/>
- Pressman, R. S. (2010). *Ingeniería del Software, Un enfoque práctico*. Mc Graw Hill.
- Ramos Porto, F. (2016). *Sistema web de administración de la ejecución de proyectos de construcción civil para la empresa constructora "Mulrahu SRL"*. La Paz.
- Salguera, H. (2012). Metodología UWE.
- Sanchez, G. (7 de febrero de 2017). *slideshare*. Obtenido de <https://es.slideshare.net/GermnSnchezDomnguez/metodologa-uwe-umlbased-web-engineering>
- Sedera. (2016). Obtenido de https://repositorio.cepal.org/bitstream/handle/11362/47540/1/S2100764_es.pdf
- SlideShare. (06 de junio de 2015). *Metodología UWE*. Obtenido de <https://es.slideshare.net/omar1023/uwe-49070035>
- Thewolf, D. (25 de junio de 2015). *Metodología UWE*. Obtenido de <https://metodologiauwe.wordpress.com/>
- Troncoso Contreras, A. P. (2017). *Desarrollo de una aplicación web para la gestión de empresas constructoras*. Valpariso.
- Tunarosa Muñoz, M. Y., & Ávila Bonilla, E. M. (2020). *Sistema de información web para el control y supervisión de obras civiles en el Carmen de Apicalá, para la empresa instalaciones hidrosanitarias y de gas Jats S.A.S "Plumbingsoft"*. Colombia.
- UWE, L. –L.-M.-U. (2016). *UWE – UML-based Web Engineering*. Obtenido de LMU – Ludwig-Maximilians-Universität München: <https://uwe.pst.ifi.lmu.de/teachingTutorialRequirementsSpanish.html>

Valencia Pavón , E. (2012-2013). *Sistema Académico web utilizando software libre*. QUITO.

Vargas, M. S. (25 de junio de 2017). *Ingeniería basada en modelos*. Obtenido de <http://marcelosalasvargas.blogspot.com/2017>

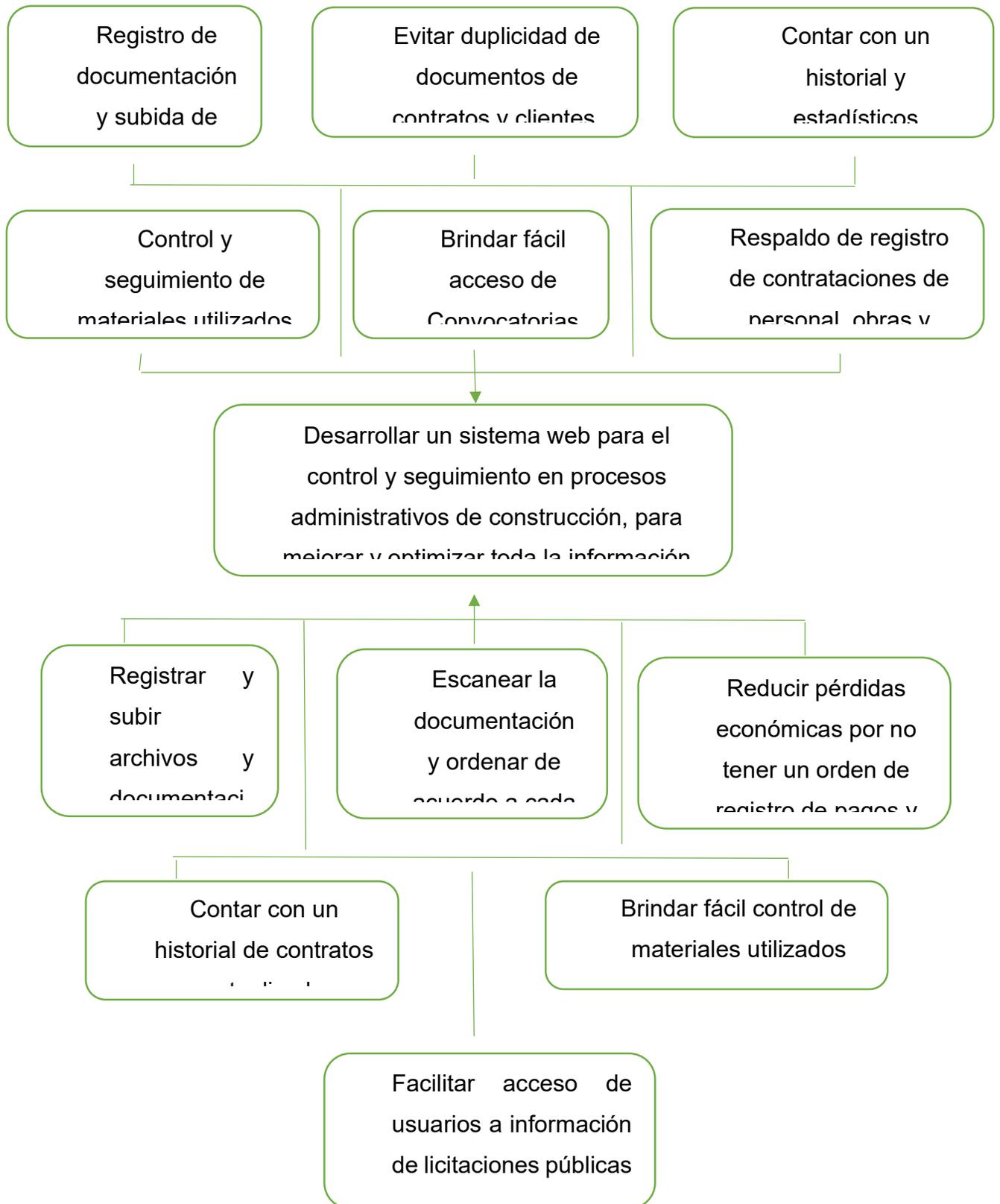
Wikipedia. (31 de octubre de 2021). Obtenido de https://es.wikipedia.org/wiki/Web_2.0

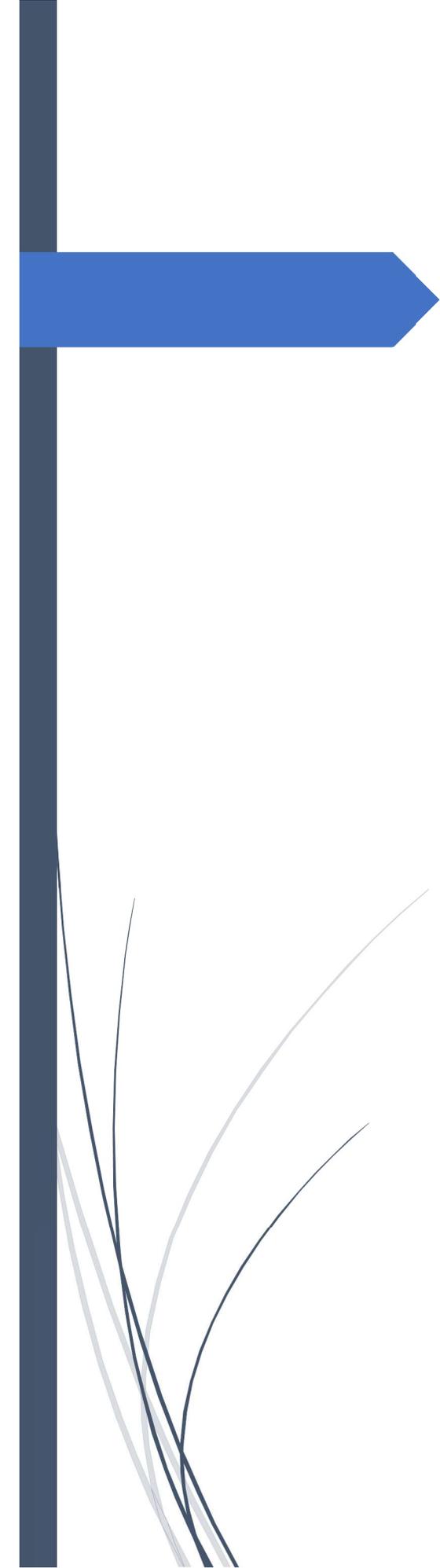
ANEXOS

Anexo A. Árbol de problemas



Anexo B. Árbol de Objetivos





*“Plataforma Web Para
Administración Y Control De
Proyectos De Construcción En
Obras Cíviles”*

MANUAL DE USUARIO

Carlos Colque Huanca

INDICE DEL MANUAL

1.	Introducción	1
2.	Objetivo del sistema	1
3.	Guía del sistema.....	1
a.	Inicio	1
4.	Roles y permisos	2
5.	Administrativos	5
6.	Clientes.....	13
7.	Ocupaciones.....	15
8.	Obreros.....	18
9.	Proyectos.....	20
2	20	
10.	Herramientas	22
11.	Partidas	25
12.	Backups.....	27
13.	Estadísticas	28

1. Introducción

El presente documento pretende servir de guía para el uso adecuado del sistema, ya que con esta plataforma se pretende realizar el registro adecuado de los proyectos.

2. Objetivo del sistema

El objetivo del manual, es permitir al usuario final, que pueda tener la administración adecuada de la PLATAFORMA WEB PARA ADMINISTRACIÓN Y CONTROL DE PROYECTOS DE CONSTRUCCIÓN EN OBRAS CIVILES.

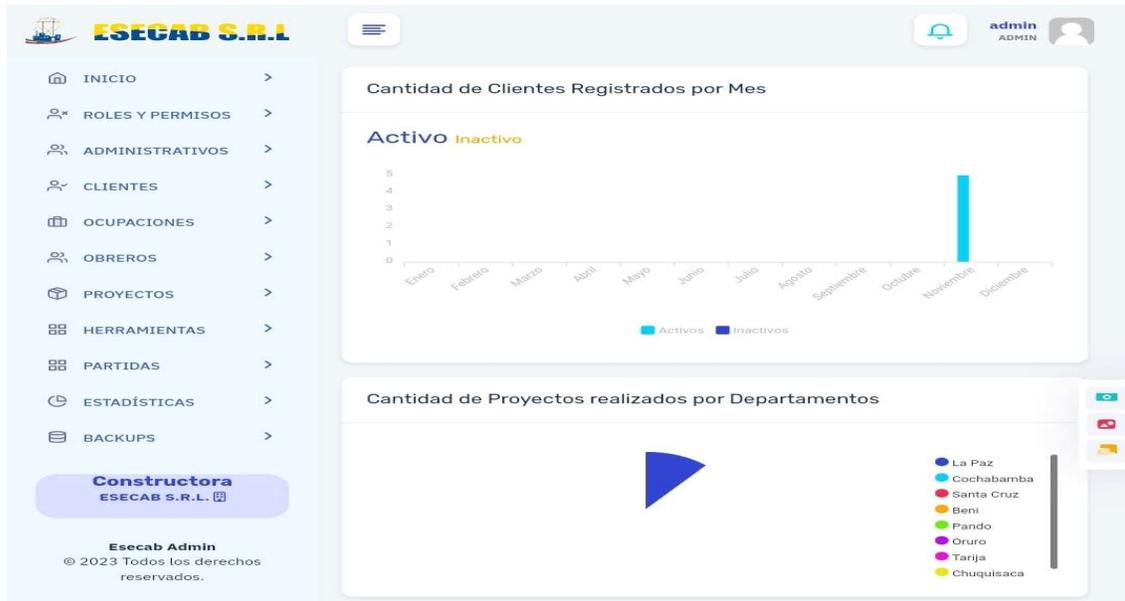
3. Guía del sistema

a. Inicio

Después de haber realizado el inicio de sesión nos muestra el un reporte de la cantidad de usuarios, proyectos, clientes y obreros.



También nos muestra el menú del sistema junto a cuadros gráficos estadísticos de los proyectos activos y registrados por mes, presentan también la cantidad de proyectos realizados por departamentos lo que asegura que se mantiene el control adecuado de la plataforma para realizar los registros.



4. Roles y permisos

Cuando se realice clic en el menú roles y permisos nos despliega la siguiente pantalla donde se podrá realizar las acciones de crear rol, crear permiso y ver los permisos.

ESECAB S.R.L.

Inicio Roles y permisos

Roles: TODOS + CREAR ROL VER PERMISOS

Presione

Mostrar 10 registros Buscar:

#	ROL DE USUARIO	GRUPO	PERMISOS	OPCIONES
1	ADMINISTRADOR GENERAL	ADMINISTRACIÓN GENERAL	Sin permisos	
2	JEFE DE PERSONAL	DIRECTORES	Sin permisos	
3	TÉCNICO DE PROYECTOS	ADMINISTRATIVOS	Sin permisos	

Mostrando 1 a 3 de 3 registros Anterior 1 Siguiete

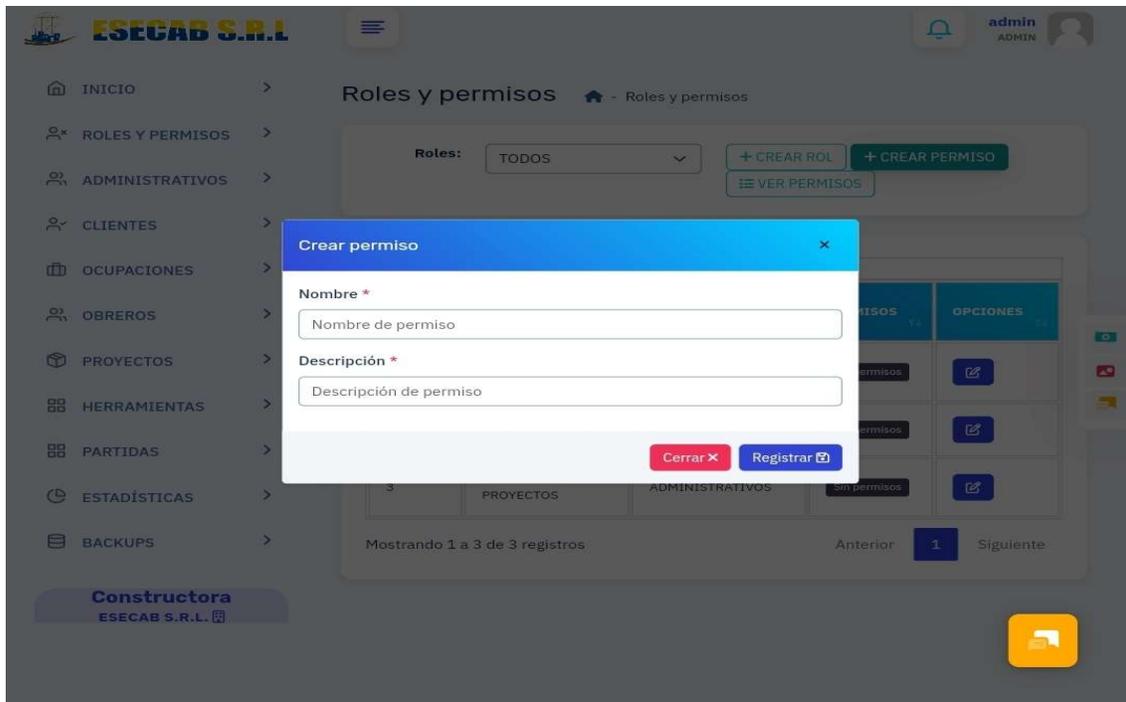
Constructora ESECAB S.R.L.

Esecab Admin © 2023 Todos los derechos reservados.

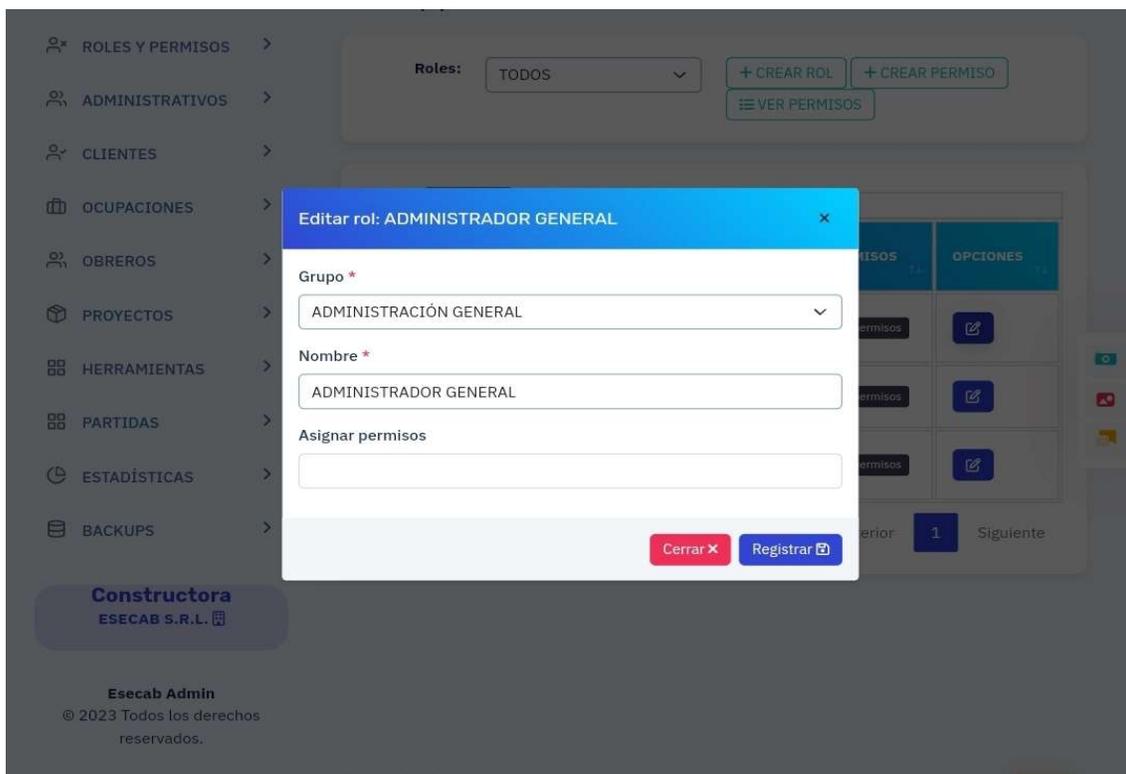
Para la creación de un rol presionamos en el botón crear crear rol y nos muestra la siguiente ventana. Donde se registra los siguientes datos:

Después se crea el permiso con los campos de asignar el nombre del permiso y la descripción de permiso.

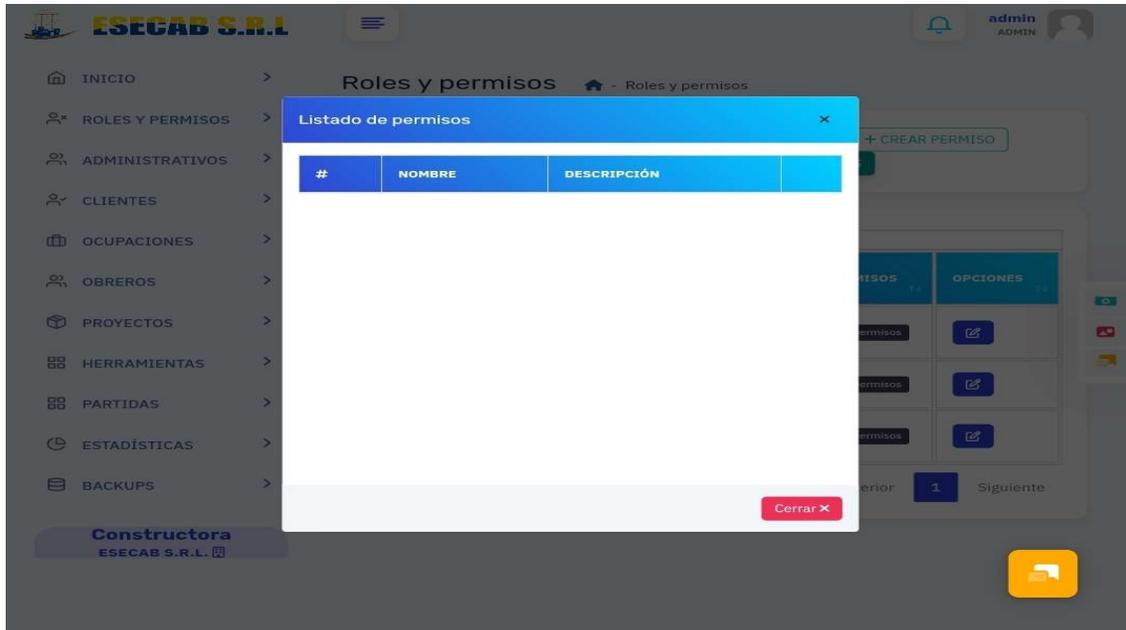




También podrá el administrador realizar la edición del rol de la siguiente manera.



Se visualizará el listado del permiso de la siguiente manera.



5. Administrativos

Listado de administrativos

En el menú de listado de administrativos se podrá visualizar la lista de los administrativos su foto el cargo su cuenta de correo y el estado en el que se encuentra como se ve en la siguiente imagen:



Lista de Administrativos

Lista de administrativos

+ REGISTRAR ADMINISTRATIVO

Mostrar 10 registros

Buscar:

#	FOTO	ADMINISTRATIVO	CARGO	EMAIL	ESTADO	OPCIONES
1		FIDEL FLORES ULLOA	ING SIST	dani_f@gmail.com	ACTIVO	Actualizar
2		DANIEL CASAS FLORES	ING SIST	delf@gmail.com	ACTIVO	Actualizar
3		ROCIO QUISPE CHAMBI	ING SIST	rovina@gmail.com	INACTIVO	Actualizar

Si desea actualizar el estado del administrativo mostrará la siguiente ventana:

ESECAB S.R.L.

Actualizar estado

Si selecciona la opción inactivo, el usuario no podrá acceder al sistema.

Estado del administrativo*

ACTIVO

Cerrar X Registrar

ADMINISTRATIVO	CARGO	EMAIL	ESTADO	OPCIONES
FIDEL FLORES ULLOA	ING SIST	dani_f@gmail.com	ACTIVO	Actualizar
DANIEL CASAS FLORES	ing sist	delf@gmail.com	ACTIVO	Actualizar
ROCIO QUISPE CHAMBI	ing sist	rovina@gmail.com	ACTIVO	Actualizar

Anterior 1 Siguiete



Para poder realizar el registro de los datos de los administrativos nos despliega la siguiente pantalla donde se muestra datos principales como ser:

Datos de Administrativo

DATOS DEL ADMINISTRATIVO

Nombres y Apellidos	FIDEL FLORES ULLOA	Número de Celular	66565679
Cargo	ING SIST	C.I.	9896001
Corporativo		Email	dani_f@gmail.com

Para poder editar los datos de los administrativos dónde se realiza las siguientes acciones:

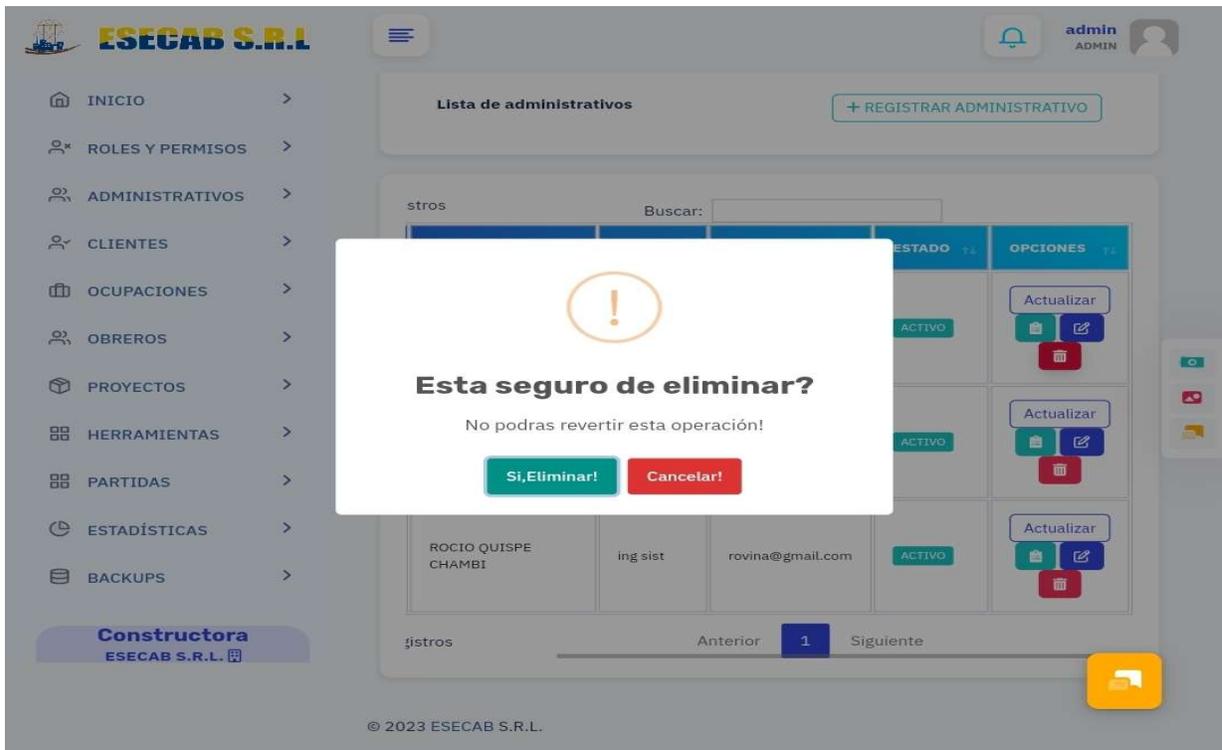
Editar Administrativo

Seleccionar archivo Ningun...vo selec.

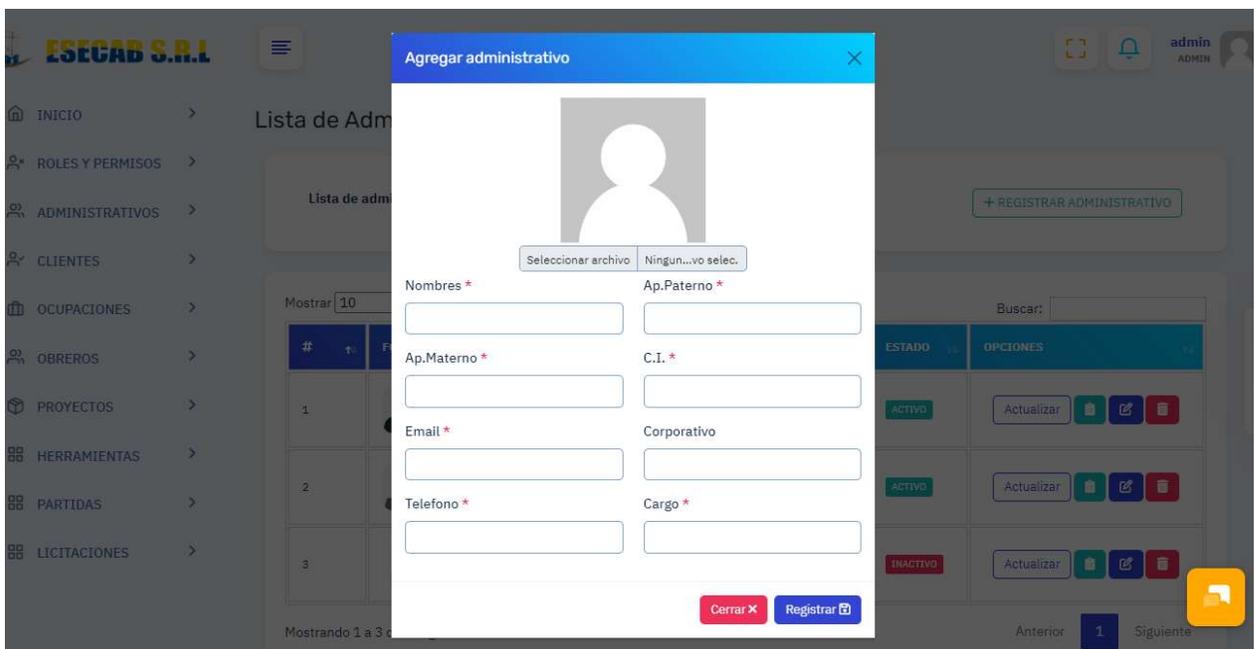
Nombres *	Ap.Paterno *
FIDEL	FLORES
Ap.Materno *	C.I. *
ULLOA	9896001
Email *	Corporativo
dani_f@gmail.com	
Telefono *	Cargo *
66565679	ING SIST

Cerrar Registrar

También se puede realizar la acción de editar los datos de los administrativos donde nos mostrará la siguiente ventana:



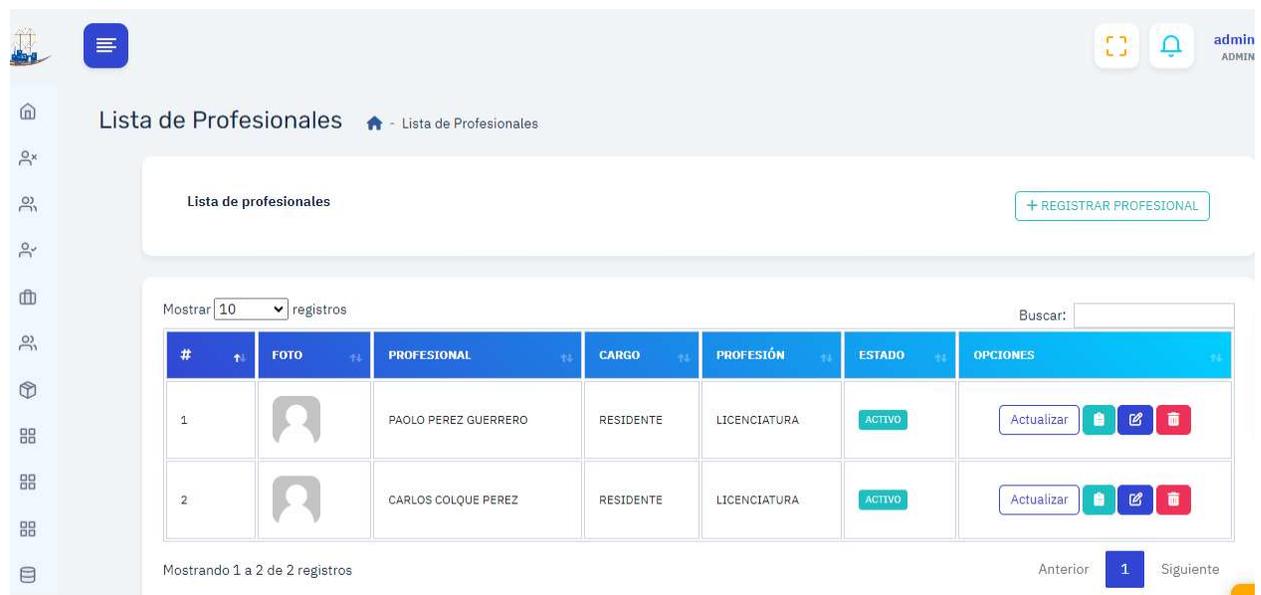
El agregar un nuevo administrativo nos despliega la siguiente ventana donde se llenan los siguientes campos



Listados de profesionales

El listado de profesionales nos mostrará con la siguiente pantalla donde se visualizará los siguientes campos:

Foto, profesión, cargo y estado.



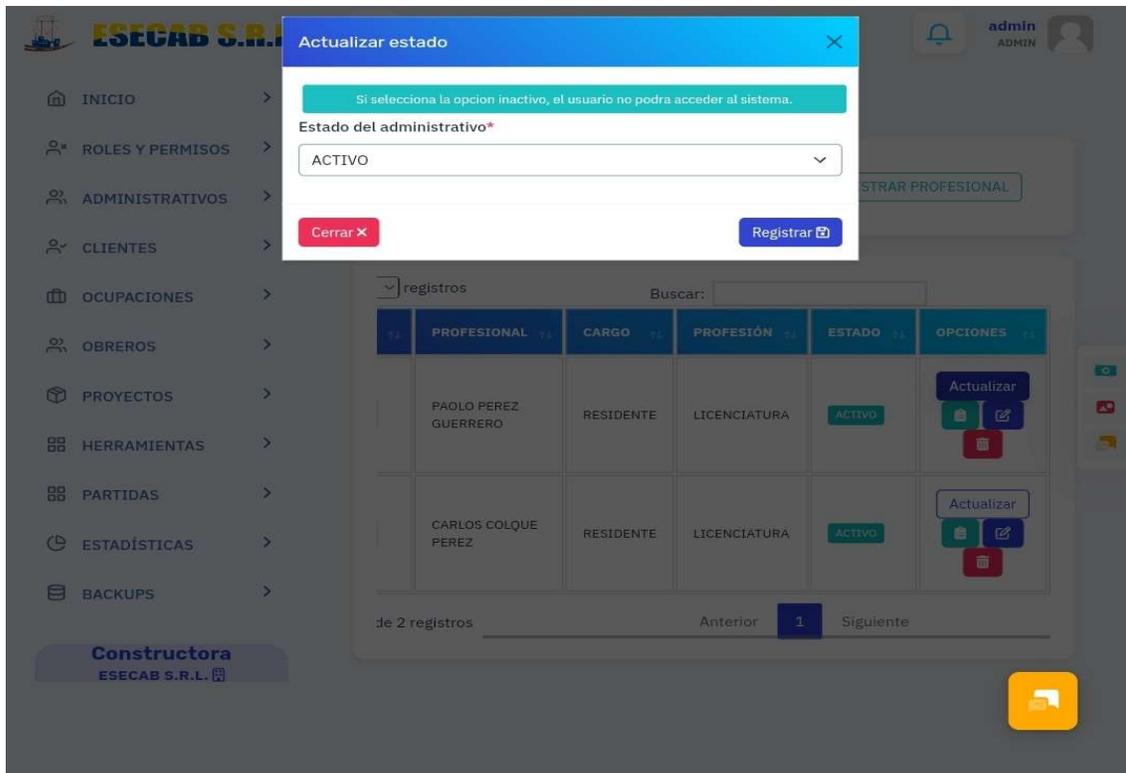
The screenshot displays a web application interface for managing professionals. The header shows the user is logged in as 'admin ADMIN'. The main content area is titled 'Lista de Profesionales' and features a '+ REGISTRAR PROFESIONAL' button. Below this, there is a table with the following data:

#	FOTO	PROFESIONAL	CARGO	PROFESIÓN	ESTADO	OPCIONES
1		PAOLO PEREZ GUERRERO	RESIDENTE	LICENCIATURA	ACTIVO	Actualizar   
2		CARLOS COLQUE PEREZ	RESIDENTE	LICENCIATURA	ACTIVO	Actualizar   

At the bottom of the table, it indicates 'Mostrando 1 a 2 de 2 registros' and navigation controls for 'Anterior', '1', and 'Siguiente'.

En la siguiente pantalla mostrará la ventana de actualizar el estado del profesional de la siguiente manera:





La siguiente pantalla muestra los datos de los profesionales con los siguientes campos:

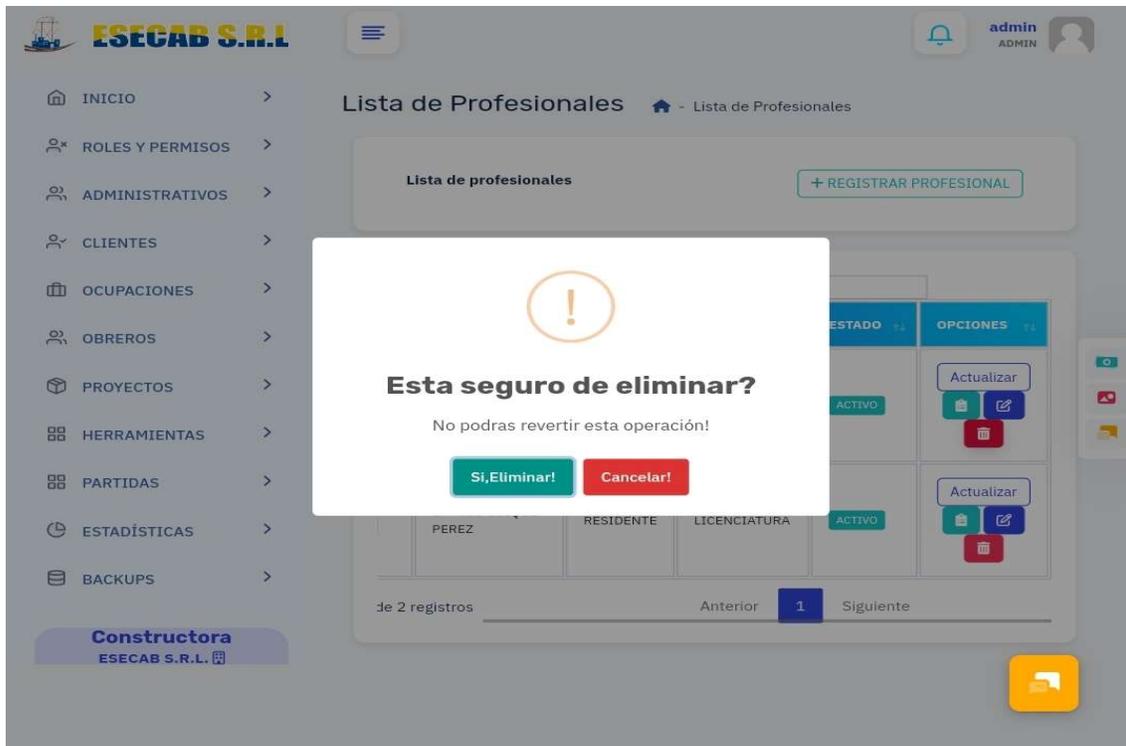


También los campos del profesional se podrán editar como muestra la siguiente pantalla:

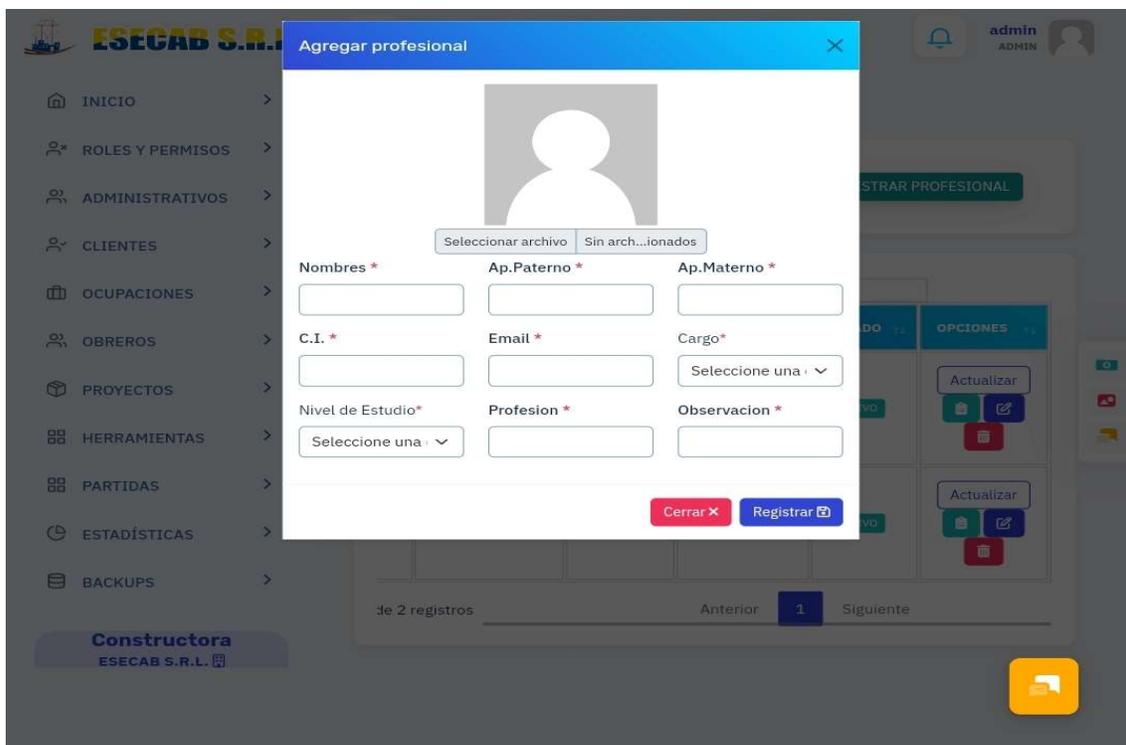
The screenshot displays the 'Editar Profesional' (Edit Professional) form. The form is a modal window with a blue header and a white body. It contains several input fields for personal and professional information, a 'Registrar' button, and a 'Cerrar' button. The background shows a sidebar menu and a table of records.

Editar Profesional		
		
Seleccionar archivo Sin archivos seleccionados		
Nombres *	Ap.Paterno *	Ap.Materno *
PAOLO	PEREZ	GUERRERO
C.I. *	Email *	Cargo*
9896001	profesional2@gmail	RESIDENTE
Nivel de Estudio*	Profesion *	Observacion *
LICENCIATURA	ARQUITECTO	
Cerrar X		Registrar

Eliminar al profesional el sistema podrá realizar la acción de eliminar de la siguiente manera:

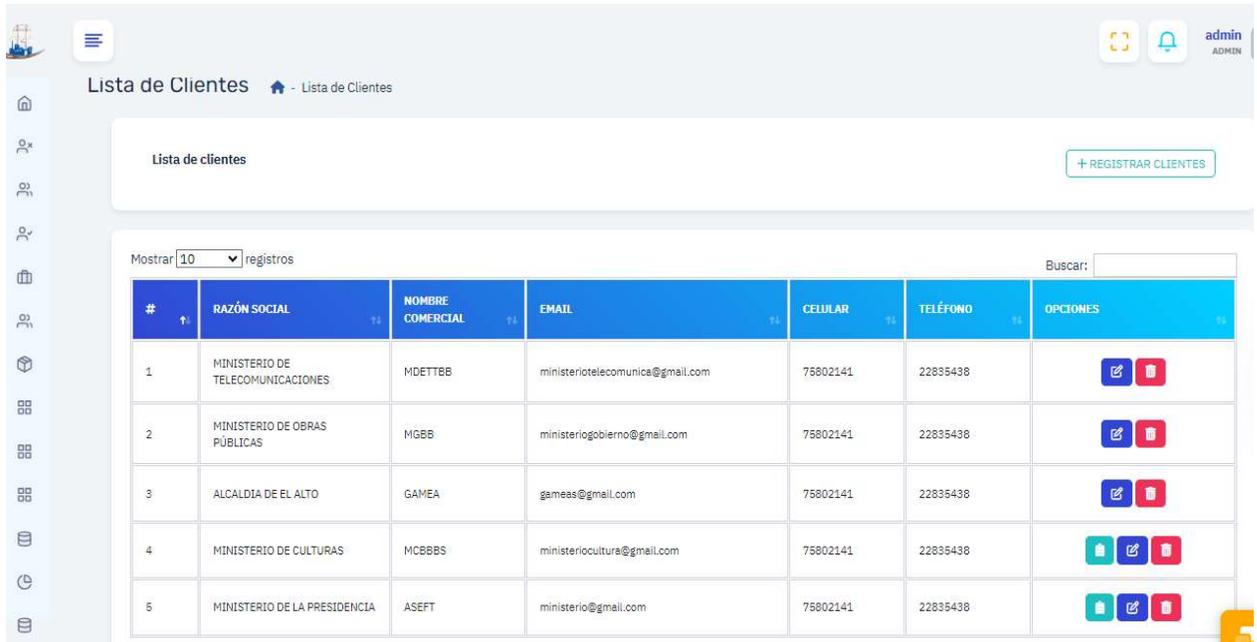


Se podrá agregar un nuevo profesional completando los siguientes campos:



6. Clientes

En el menú administrativos también nos mostrará la lista de los clientes con los datos de la razón social, nombre comercial y el email.



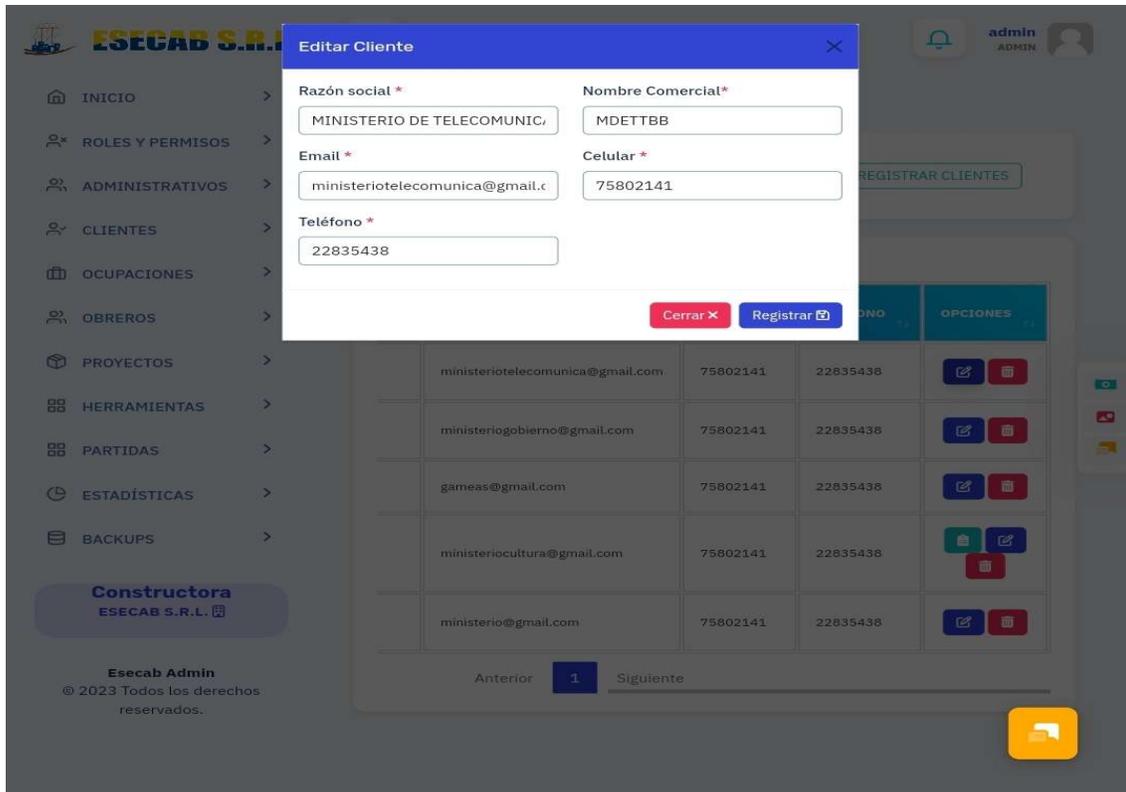
Lista de Clientes

Lista de clientes + REGISTRAR CLIENTES

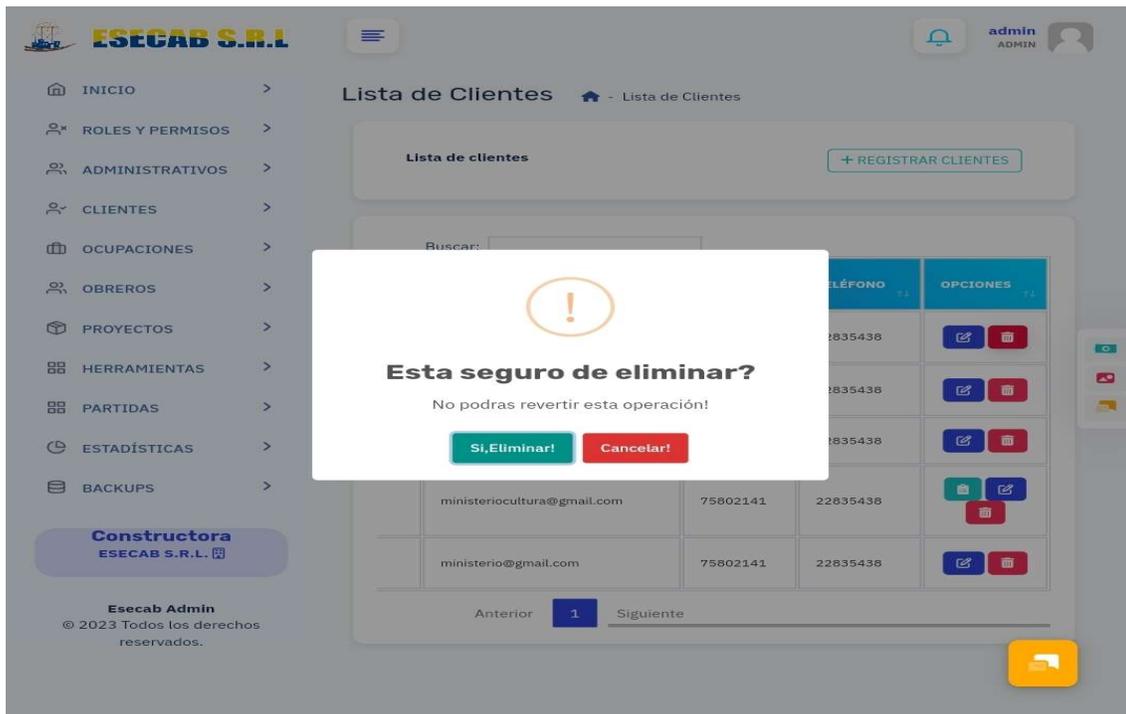
Mostrar 10 registros Buscar:

#	RAZÓN SOCIAL	NOMBRE COMERCIAL	EMAIL	CELULAR	TELÉFONO	OPCIONES
1	MINISTERIO DE TELECOMUNICACIONES	MDETTBB	ministeriotelecomunica@gmail.com	75802141	22835438	 
2	MINISTERIO DE OBRAS PÚBLICAS	MGBB	ministeriogobierno@gmail.com	75802141	22835438	 
3	ALCALDIA DE EL ALTO	GAMEA	gameas@gmail.com	75802141	22835438	 
4	MINISTERIO DE CULTURAS	MCBBS	ministeriocultura@gmail.com	75802141	22835438	  
5	MINISTERIO DE LA PRESIDENCIA	ASEFT	ministerio@gmail.com	75802141	22835438	  

Se realiza la acción de editar cliente en los cuales los siguientes campos se podrán editar:



La siguiente ventana mostrara el mensaje de Eliminar y si está de acuerdo con la acción de la siguiente manera:



También se podrán agregar un nuevo profesional con los siguientes campos:

Agregar cliente

Razón social *

Nombre Comercial*

Email *

Celular *

Teléfono *

Cerrar X Registrar

ministeriotelecomunica@gmail.com	75802141	22835438		
ministeriogobierno@gmail.com	75802141	22835438		
gameas@gmail.com	75802141	22835438		
ministeriocultura@gmail.com	75802141	22835438		
ministerio@gmail.com	75802141	22835438		

Anterior 1 Siguiente

7. Ocupaciones

Lista de Ocupaciones

+ REGISTRAR OCUPACIÓN

Mostrar 10 registros

#	NOMBRE	DESCRIPCIÓN	OPCIONES
1	PLOMERO	ENCARGADO DE PLOMERIA EN GENERAL	
2	TRANSPORTISTA	TRASLADO DE MATERIALES.	
3	COMERCIANTE	VENTA MAQUINARIA	

Mostrando 1 a 3 de 3 registros

Anterior 1 Siguiente

En esta sección del sistema, el usuario podrá ver la lista de todas las ocupaciones que un obrero puede tener, además que se visualiza las opciones de crear, editar y eliminar.

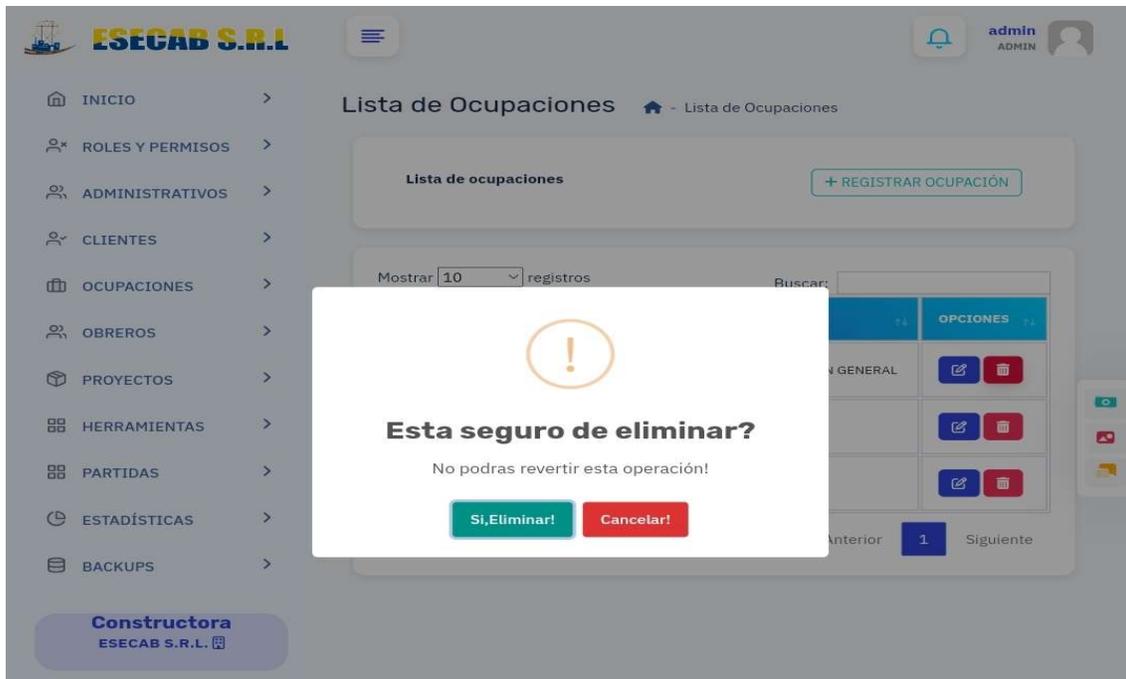
Mostrar 10 registros

#	NOMBRE	DESCRIPCIÓN	OPCIONES
1	PLOMERO	ENCARGADO DE PLOMERIA EN GENERAL	
2	TRANSPORTISTA	TRASLADO DE MATERIALES.	
3	COMERCIANTE	VENTA MAQUINARIA	

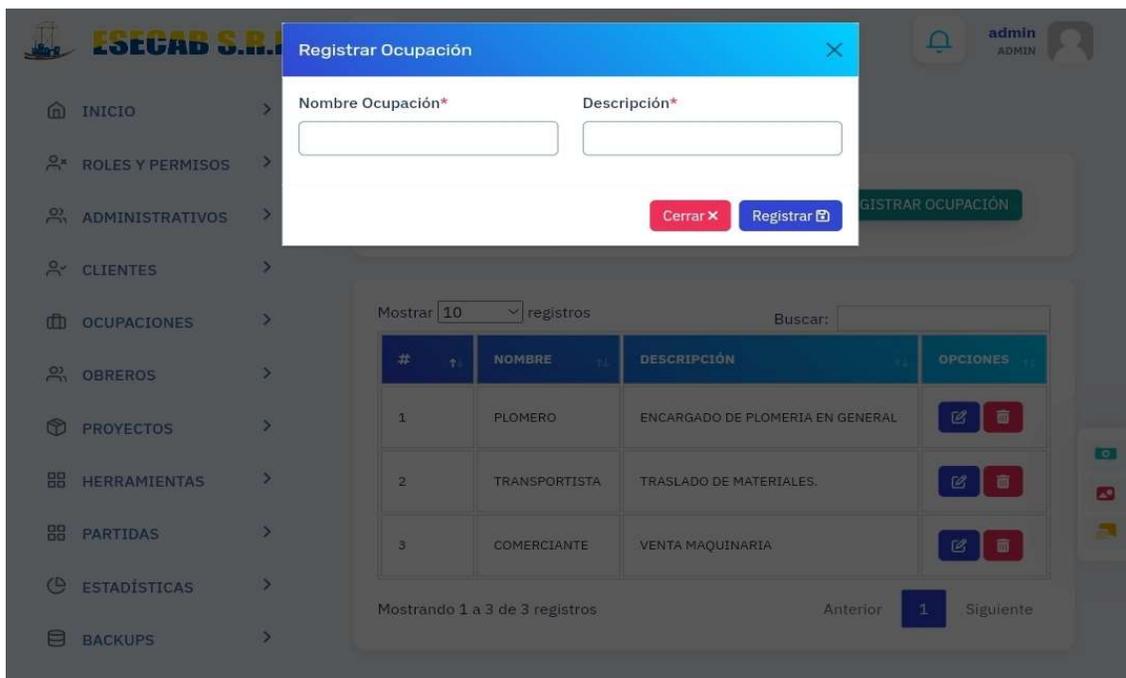
Mostrando 1 a 3 de 3 registros

Anterior 1 Siguiente

En la opción de editar, tenemos la opción de actualizar los datos de ocupación de acuerdo al dato seleccionado.



En la opción de eliminar, tenemos la opción de eliminar los datos de ocupación de acuerdo al dato seleccionado.



En la opción de registrar ocupación, podemos registrar nuevas ocupaciones de acuerdo a lo requerido en las obras.

8. Obreros

Lista de Obreros

Mostrar 10 registros

#	NOMBRE	CEDULA	TIPO OBRERO	OBSERVACIÓN	OPCIONES
1	ADRIANA CASAS MENONA	99111123	EVENTUAL	SIN OBSERVACIÓN	 
2	JUAN PEDRO CAMIRI CAMARGO	9980023	EVENTUAL	SIN OBSERVACIÓN	 
3	CARLOS MAMANI QUIISPE	9987323	FIJO	SIN OBSERVACIÓN	 

Mostrando 1 a 3 de 3 registros

En esta sección del sistema, el usuario podrá ver la lista de todos los obreros inscritos en la empresa, además que se visualiza las opciones de crear, editar y eliminar.

Editar Obrero

Nombres: * ADRIANA

Apellido Paterno* CASAS

Apellido Materno* MENONA

Cedula: * 99111123

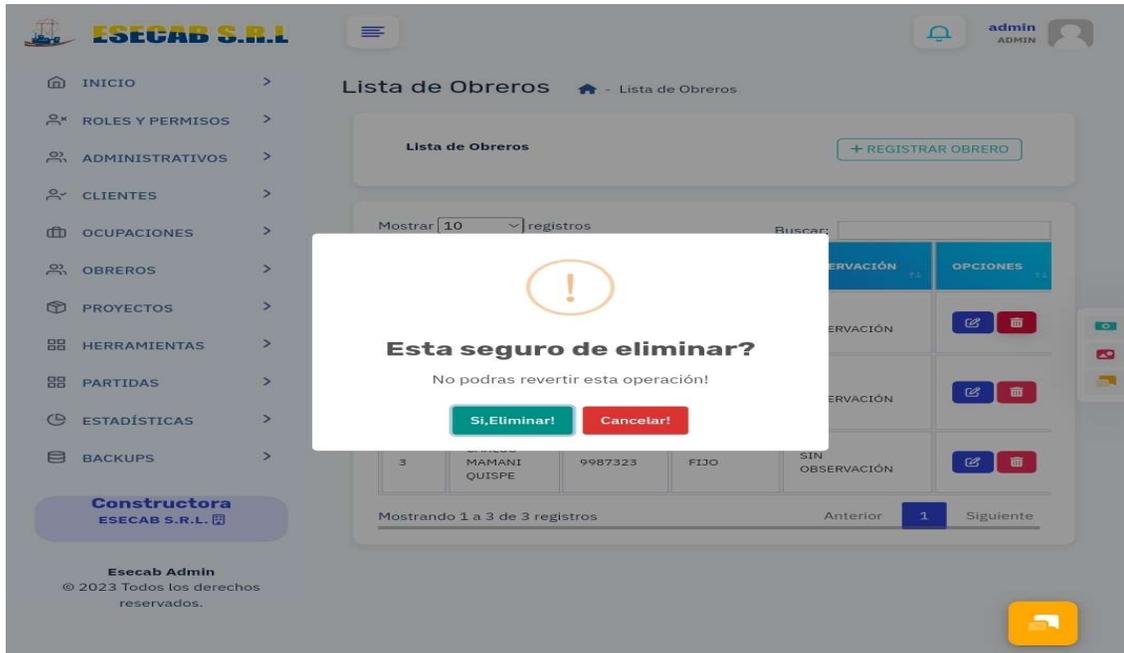
Tipo de Obrero* EVENTUAL

Observación* SIN OBSERVACIÓN

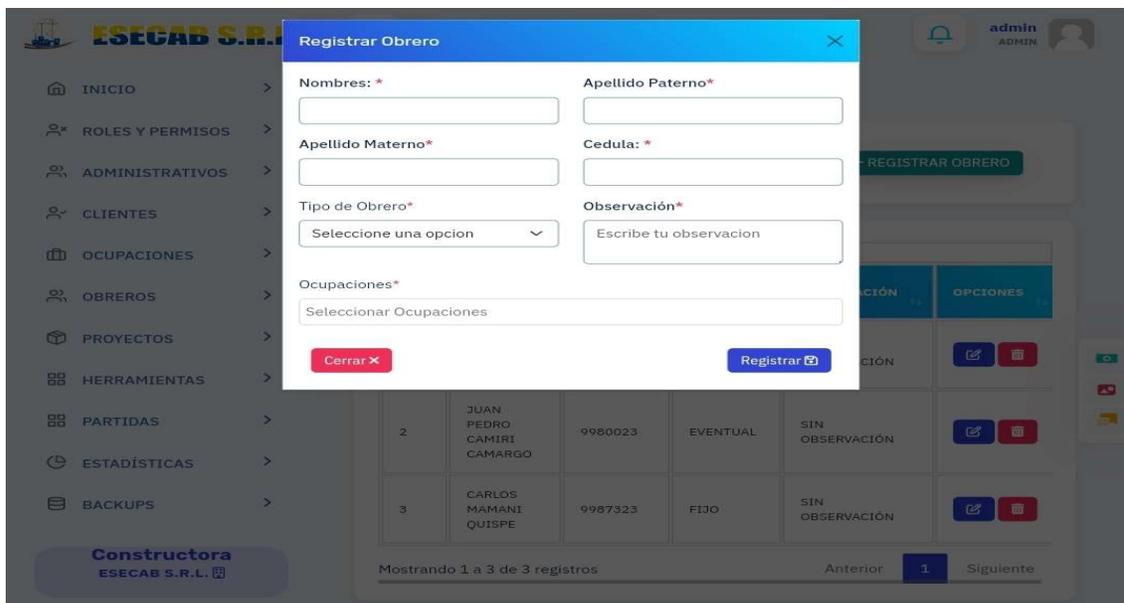
Ocupaciones*

Cerrar X Registrar

En la opción de editar, tenemos la opción de actualizar los datos de ocupación de acuerdo al dato seleccionado, donde se puede asignar una o más ocupaciones.



En la opción de eliminar, tenemos la opción de eliminar los datos de obreros de acuerdo al dato seleccionado.



En la opción de registrar obrero, podemos registrar los datos del obrero de la empresa, con las ocupaciones del registrado

9. Proyectos

Lista de Proyectos

Lista de proyectos + REGISTRAR PROYECTO

PROYECTOS ACTIVOS PROYECTOS FINALIZADOS PROYECTOS CANCELADOS

Mostrar 10 registros

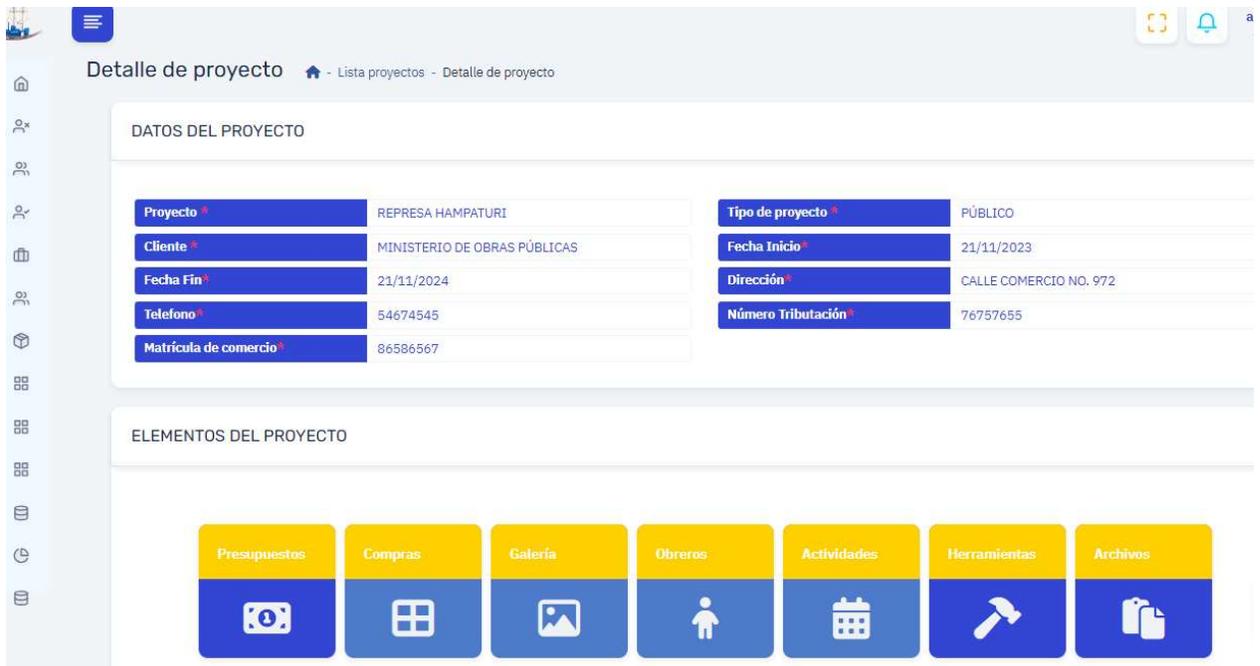
#	NOMBRE DE PROYECTO	TIPO DE PROYECTO	PROVINCIA	TELEFONO	ESTADO	OPCIONES
1	REPRESA HAMPATURI	PÚBLICO	Abel Iturraide	54674545	Actualizar	
2	EDIFICIO CALOMAI	PÚBLICO	Abel Iturraide	54674545	Actualizar	
3	PUENTE SAN ANDRES	PÚBLICO	Abel Iturraide	54674545	Actualizar	
4	Canchas Municipales	PÚBLICO	Omasuyos	65646027	Actualizar	

Mostrando 1 a 4 de 4 registros

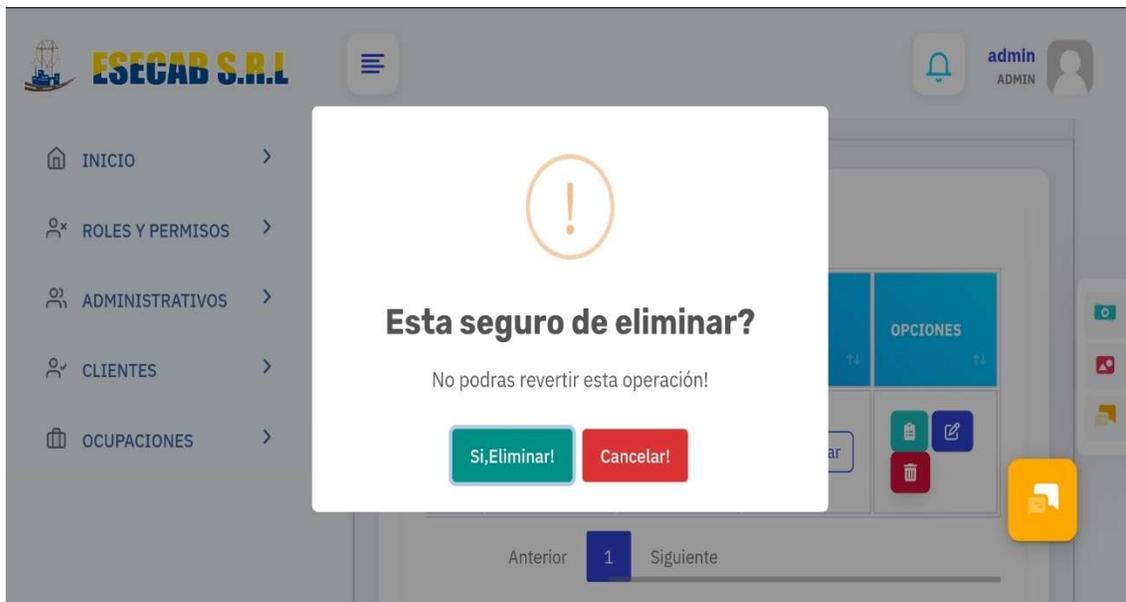
Anterior 1 Siguiente

En esta sección podemos ver la lista de todos los proyectos, filtrándolos de acuerdo al estado de acuerdo al estado en que se encuentre, este puede ser estado del proyecto activo, finalizado y cancelados.

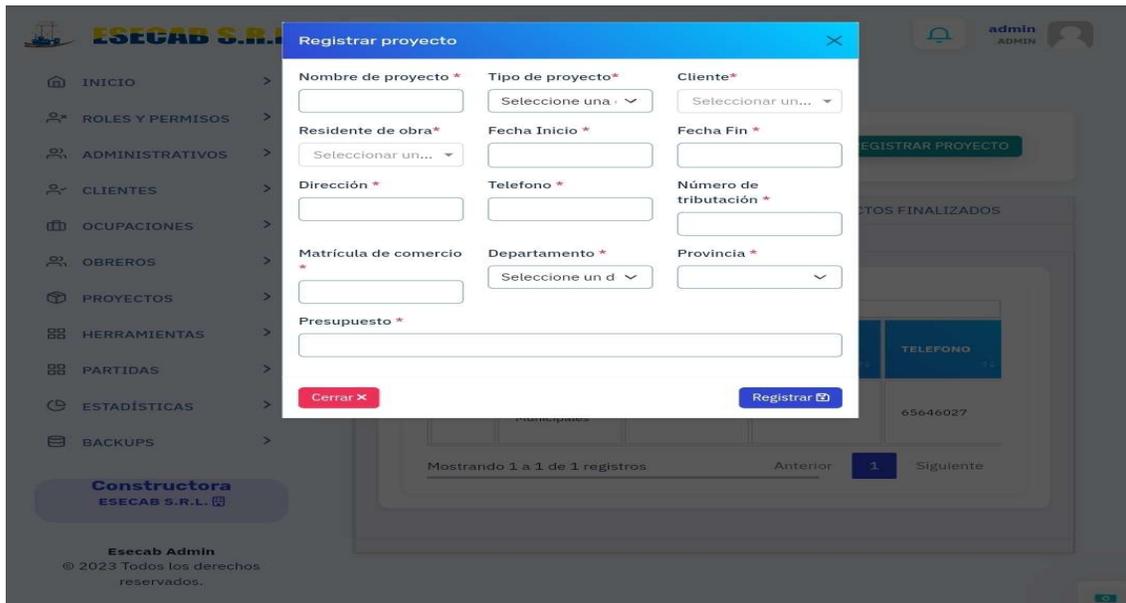
En esta sección podemos ver las opciones de ver los botones de editar, eliminar y ver datos del proyecto.



El botón de ver datos del proyecto, nos envía a esta vista donde podemos ver los elementos de cada proyecto.



Opción de eliminar proyecto, se muestra la opción de alerta y advertencia antes proceder esta opción.



El botón de registrar proyecto nos muestra los siguientes campos, que son obligatorios para el registro de cada proyecto.

10. Herramientas

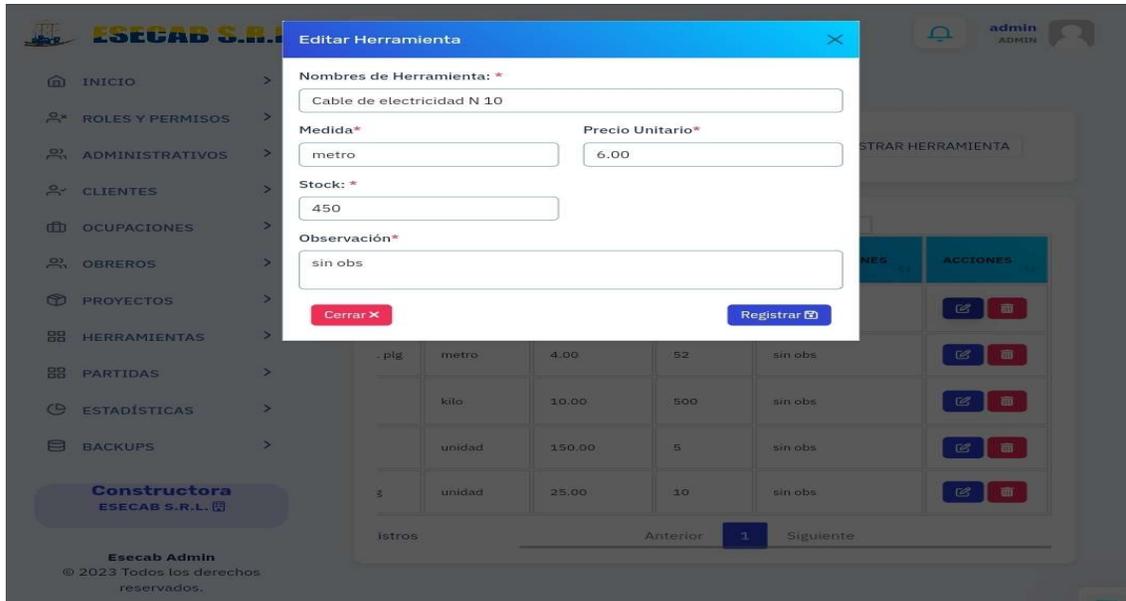
Lista de Herramientas [Lista de Herramientas](#)

Mostrar registros Buscar:

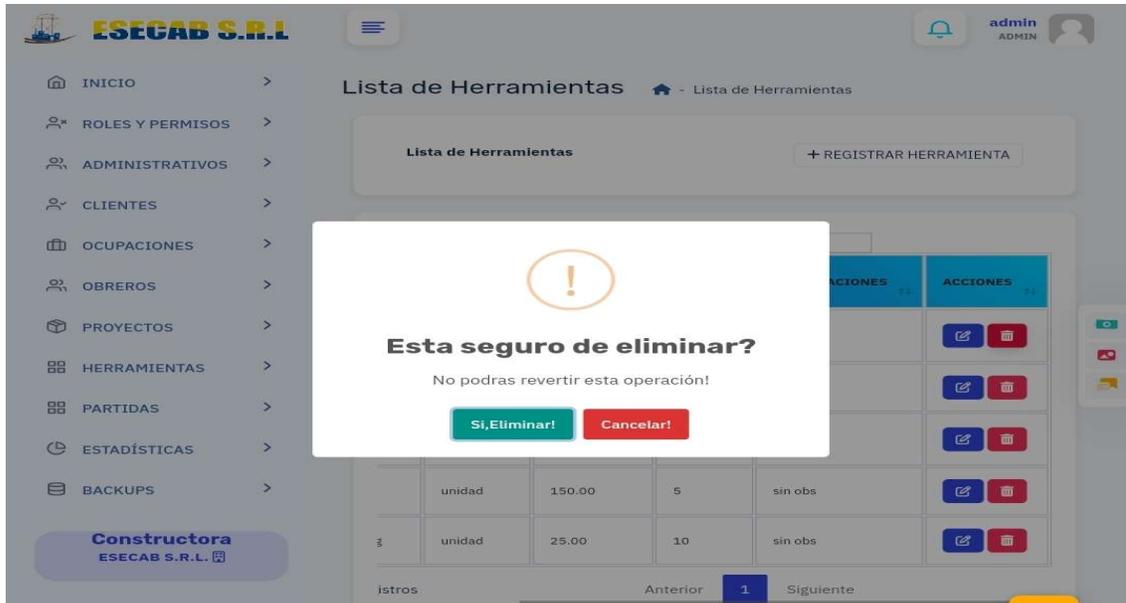
[REPORTE PDF](#) [+ REGISTRAR HERRAMIENTA](#)

#	HERRAMIENTA	MEDIDA	PRECIO UNITARIO	STOCK	OBSERVACIONES	ACCIONES
1	Cable de electricidad N 10	metro	6.00 Bs.	450	sin obs	✎ 🗑
2	Tuberman de 1 plg	metro	4.00 Bs.	52	sin obs	✎ 🗑
3	Alambre corrugado	kilo	10.00 Bs.	500	sin obs	✎ 🗑
4	carretilla	unidad	150.00 Bs.	5	sin obs	✎ 🗑
5	Martillo de 1 kg	unidad	25.00 Bs.	10	sin obs	✎ 🗑

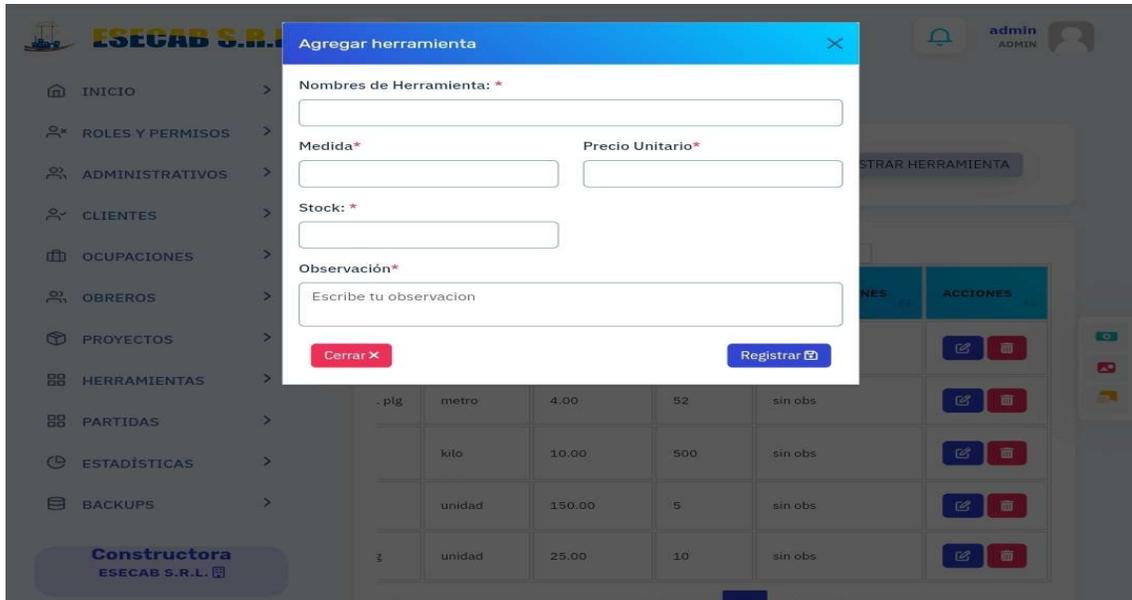
En esta sección del sistema, el usuario podrá ver la lista de todas las herramientas pertenecientes a la empresa, además que se visualiza las opciones de crear, editar y eliminar.



El botón de editar, nos muestra la opción de editar la herramienta, para actualizar los datos de la partida de las herramientas.



Opción de eliminar herramienta, se muestra la opción de alerta y advertencia antes proceder esta opción.



El botón de registrar herramienta nos muestra los siguientes campos, que son obligatorios para el registro de cada herramienta.

11. Partidas

En esta sección del sistema, el usuario podrá ver la lista de todas las partidas pertenecientes a la empresa, además que se visualiza las opciones de crear, editar y eliminar.

The screenshot displays the 'Editar partida' (Edit item) modal form in the 'ESECAB S.R.L.' system. The form is overlaid on a table of budget items. The modal form contains the following fields:

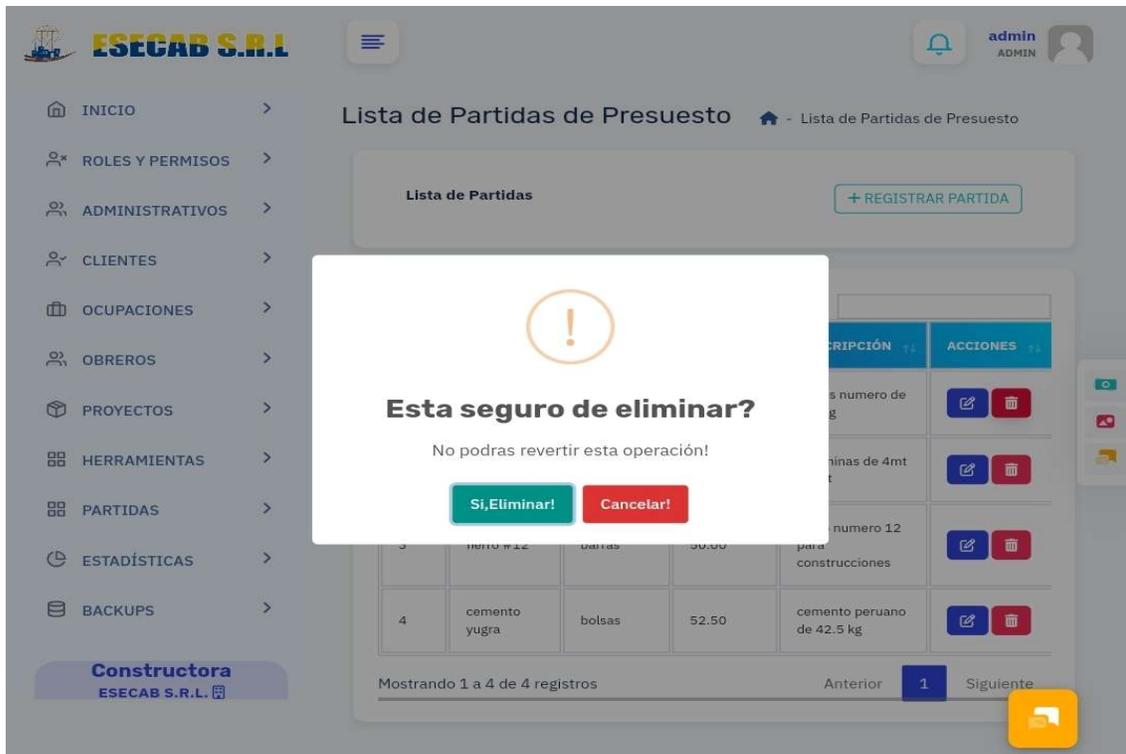
- Nombres de Partida:** Clavos 2.5
- Medida:** kilo
- Precio:** 12.00
- Descripción:** clavos numero de 2.5plg

The background table shows the following data:

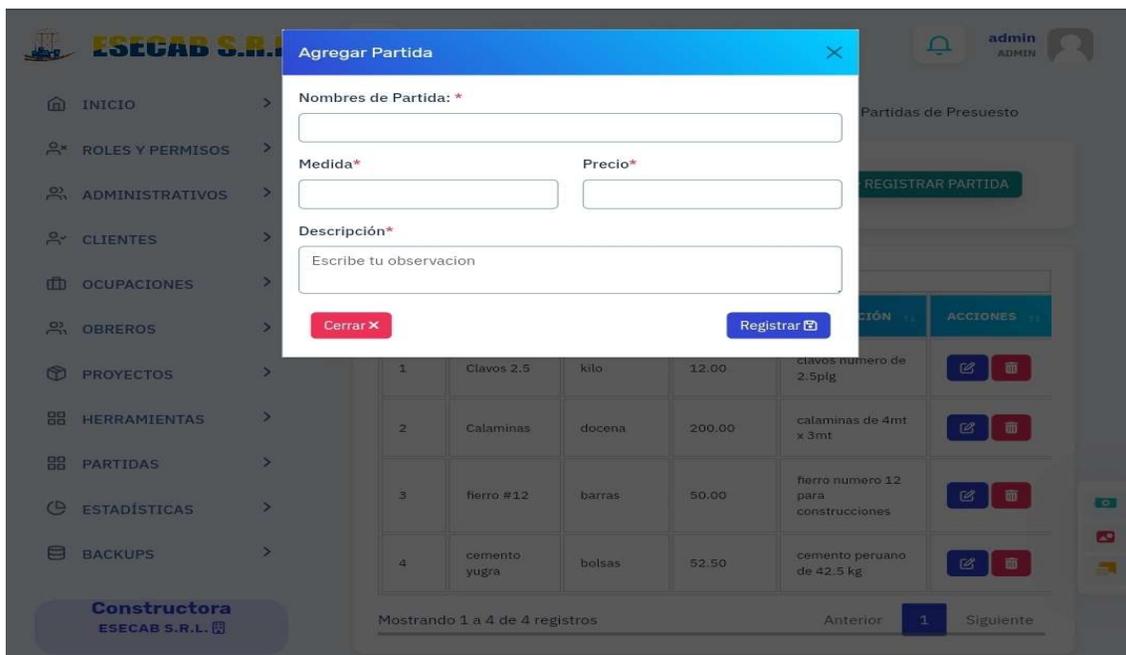
ID	Nombre	Unidad	Precio	Descripción	Acciones
1	Clavos 2.5	kilo	12.00	clavos numero de 2.5plg	[Editar] [Eliminar]
2	Calaminas	docena	200.00	calaminas de 4mt x 3mt	[Editar] [Eliminar]
3	fierro #12	barras	50.00	fierro numero 12 para construcciones	[Editar] [Eliminar]
4	cemento yugra	bolsas	52.50	cemento peruano de 42.5 kg	[Editar] [Eliminar]

The interface also includes a sidebar menu with options like INICIO, ROLES Y PERMISOS, ADMINISTRATIVOS, CLIENTES, OCUPACIONES, OBREROS, PROYECTOS, HERRAMIENTAS, PARTIDAS, ESTADÍSTICAS, and BACKUPS. The footer shows 'Constructora ESECAB S.R.L.' and 'Esecab Admin © 2023 Todos los derechos reservados.'.

El botón de editar, nos muestra la opción de editar la partida, para actualizar los datos de la partida de los presupuestos.

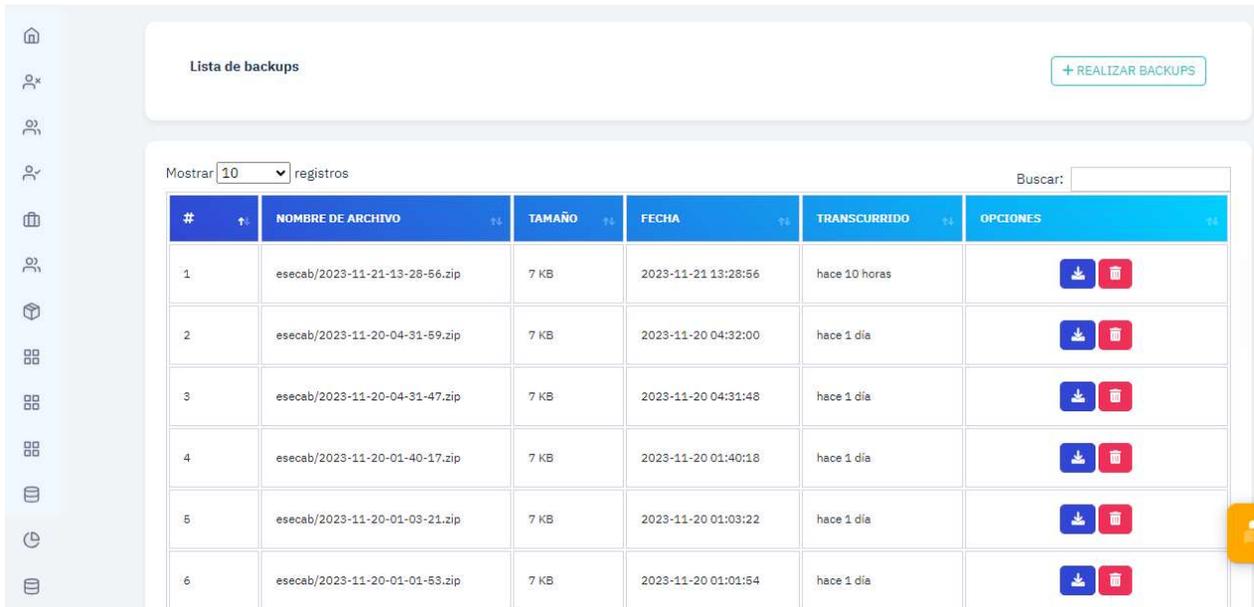


Opción de eliminar partida de presupuesto, se muestra la opción de alerta y advertencia antes proceder esta opción.



El botón de registrar partida nos muestra los siguientes campos, que son obligatorios para el registro de la partida de presupuesto.

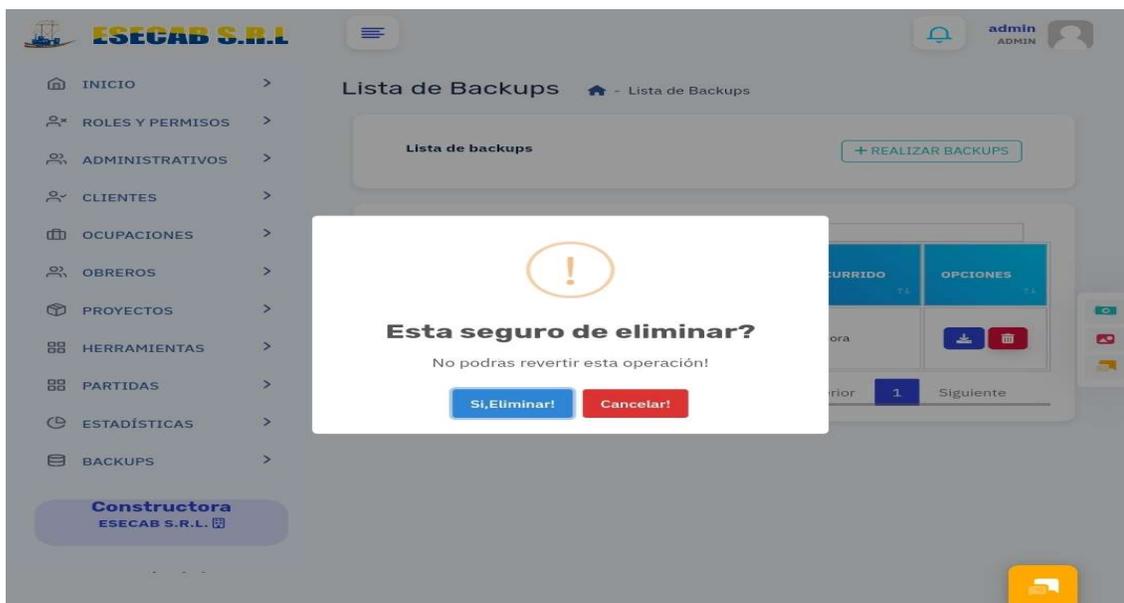
12. Backups



The screenshot shows a web interface for managing backups. At the top, there is a header 'Lista de backups' and a button '+ REALIZAR BACKUPS'. Below the header, there is a search bar and a dropdown menu for 'Mostrar' set to '10' registros. The main content is a table with the following columns: '#', 'NOMBRE DE ARCHIVO', 'TAMAÑO', 'FECHA', 'TRANSCURRIDO', and 'OPCIONES'. The table contains six rows of backup records, each with a download and delete icon in the 'OPCIONES' column.

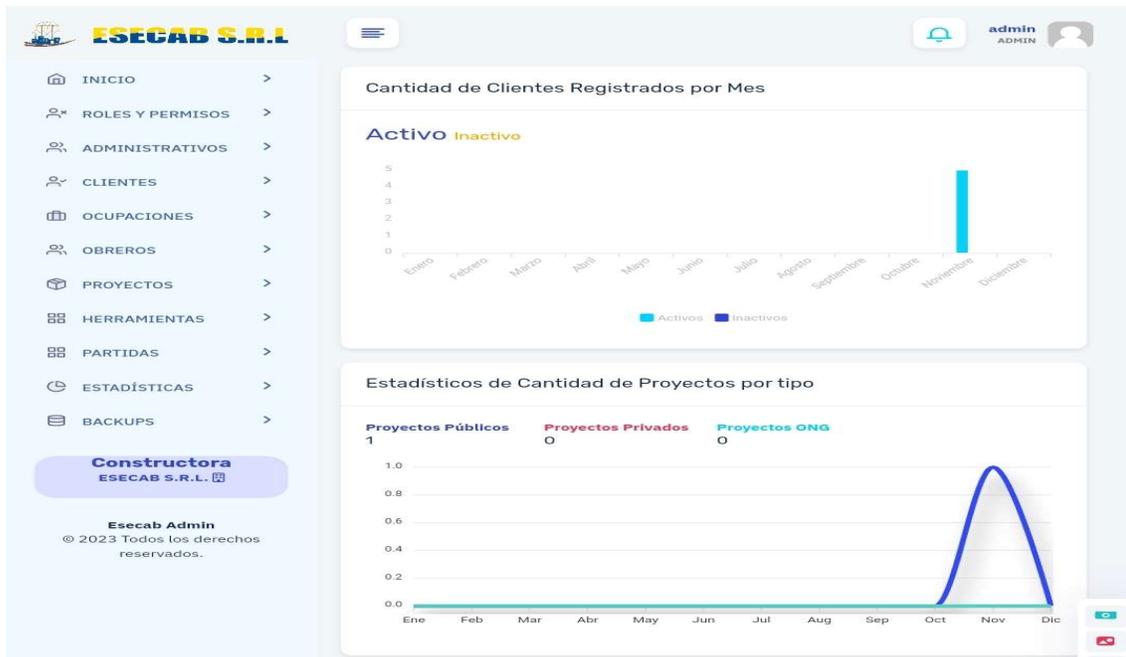
#	NOMBRE DE ARCHIVO	TAMAÑO	FECHA	TRANSCURRIDO	OPCIONES
1	esecab/2023-11-21-13-28-56.zip	7 KB	2023-11-21 13:28:56	hace 10 horas	[Download] [Delete]
2	esecab/2023-11-20-04-31-59.zip	7 KB	2023-11-20 04:32:00	hace 1 día	[Download] [Delete]
3	esecab/2023-11-20-04-31-47.zip	7 KB	2023-11-20 04:31:48	hace 1 día	[Download] [Delete]
4	esecab/2023-11-20-01-40-17.zip	7 KB	2023-11-20 01:40:18	hace 1 día	[Download] [Delete]
5	esecab/2023-11-20-01-03-21.zip	7 KB	2023-11-20 01:03:22	hace 1 día	[Download] [Delete]
6	esecab/2023-11-20-01-01-53.zip	7 KB	2023-11-20 01:01:54	hace 1 día	[Download] [Delete]

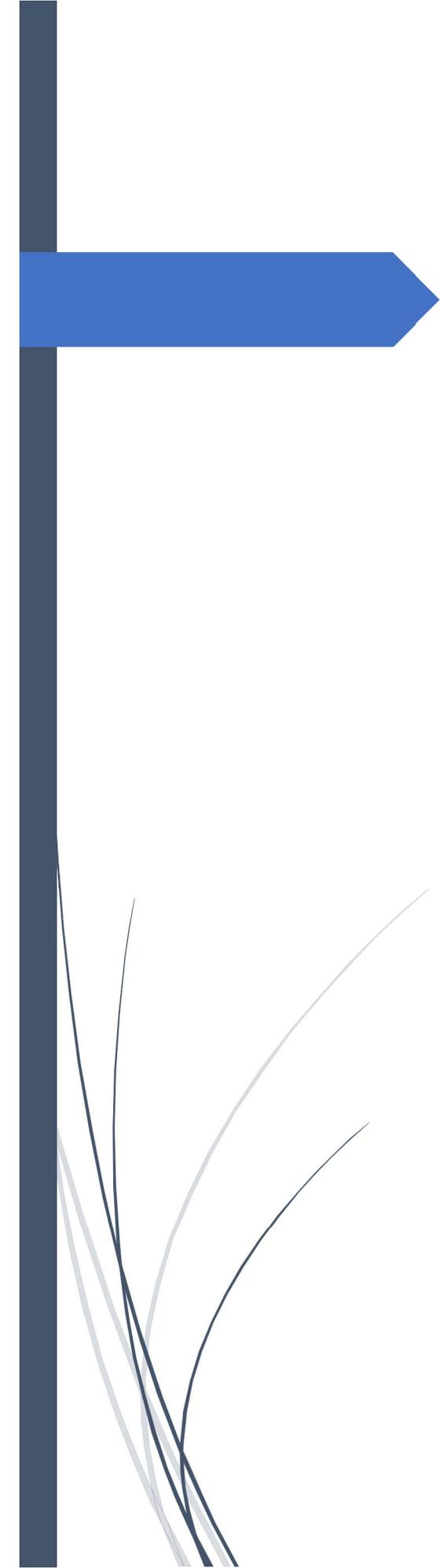
En esta sección del sistema, el administrador podrá ver todos los respaldos de la base de datos crear uno nuevo, descargar estos respaldos.



Opción de eliminar, se muestra la opción de alerta y advertencia antes proceder esta opción.

13. Estadísticas





*“Plataforma Web Para
Administración Y Control
De Proyectos De
Construcción En Obras
Cíviles”*

MANUAL TÉCNICO

Carlos Colque Huanca

1. PRINCIPALES REQUERIMIENTOS

1.1 Requerimientos Funcionales

- Acceso al Sistema mediante verificación de datos de acceso.
- Administrar los roles y permisos de acuerdo a los usuarios registrados.
- Administración de Administrativos.
- Administración de Profesionales.
- Administración de Proyectos.
- Reporte Estadístico.
- Respaldos de Base de Datos.

1.2 Requerimientos No Funcionales

- Sistema Compatible con los diferentes navegadores.
- Interfaz gráfica debe responder al usuario cuando lo requiera
- Transacciones deben responder las necesidades de los usuarios.
- Soporte y Mantenimiento, para disponibilidad del sistema.

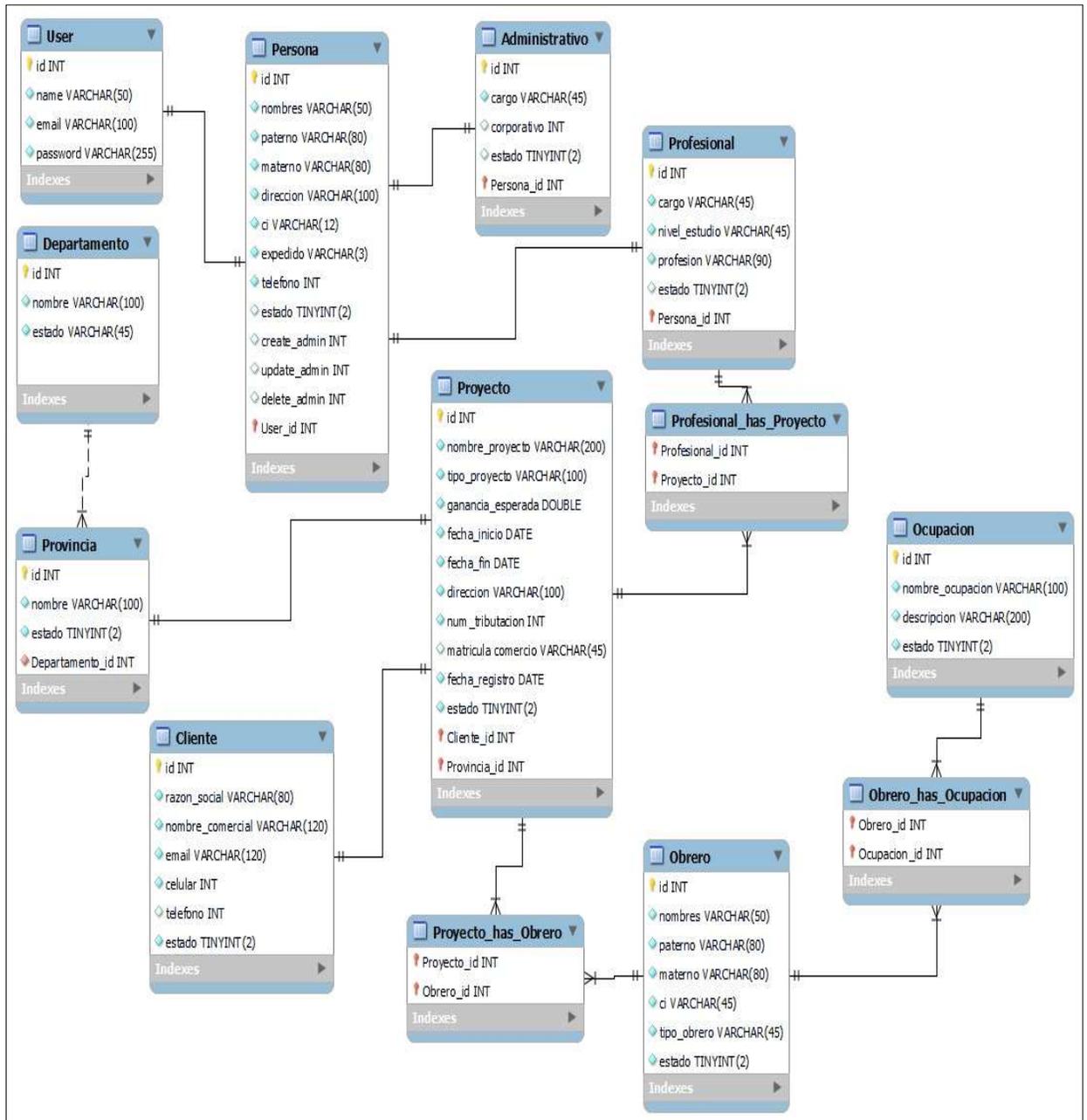
2 DISEÑO DEL SISTEMA

2.1 Caso de Uso General



3 BASE DE DATOS

3.1 Modelo de Base de Datos Relacional



4 HERRAMIENTA DE DESARROLLO

Las herramientas para el desarrollo del sistema serán los siguientes:

➤ **Hypertext Pre.Procesor “PHP”**

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

➤ **MySQL**

MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation.

➤ **Laravel**

Es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP, basado en el modelo vista controlador, intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP

5 REQUERIMIENTO E INSTALACIÓN

5.1 Requerimientos

A continuación, se lista los requerimientos del sistema para la implementación:

- **PHP versión mínima 8.2 para arriba.**
- **Servidor Apache instalado.**
- **Acceso a Terminal para ejecución de comandos.**
- **Composer.**
- **Node JS**

El sistema cuenta con un flujo considerable de archivos pdf e imágenes por lo que se recomienda tener la capacidad de acuerdo a la siguiente especificación:

- **100 GB de almacenamiento como mínimo.**