UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS

PROYECTO DE GRADO

SISTEMA WEB PARA EL REGISTRO, SEGUIMIENTO Y CONTROL EFICIENTE DE LUMINARIAS PÚBLICAS, MATERIALES Y HERRAMIENTAS

CASO: GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO G.A.M.E.A

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Wilson Viracocha Laura

Tutor Metodológico: M.Sc.Lic.Ing. Marisol Arguedas Balladares

Tutor Especialista: Lic.Ing. Luis Francisco Páez Rocha

Tutor Revisor: Lic.Ing. Yuly Ramírez Limachi

EL ALTO – BOLIVIA

2024

DECLARACIÓN JURADA DE NO PLAGIO DE

PROYECTO DE GRADO

Yo, Wilson Viracocha Laura, estudiante de Taller de Grado II, de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto, Identificado con CI. 8377140 L.P. y Registro Universitario: 200020708.

Declaro bajo juramento que:

1. Soy autor del Trabajo de Grado titulado:

“SISTEMA WEB PARA EL REGISTRO, SEGUIMIENTO Y CONTROL EFICIENTE DE LUMINARIAS PÚBLICAS, MATERIALES Y HERRAMIENTAS” CASO: “GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO G.A.M.E.A”.

El mismo que presento bajo la modalidad de Proyecto de Grado.

2. El texto de mi trabajo de grado respeta y no vulnera los derechos de terceros, incluidos los derechos de propiedad intelectual. En tal sentido, declaro que este Trabajo de Grado no ha sido plagiado total ni parcialmente, para la cual he respetado las normas internacionales de citas y referencias de las fuentes consultadas.
3. El texto del Trabajo de Grado que presento no ha sido publicado ni presentado antes en cualquier medio electrónico o físico.
4. Por tanto, declaro que mi Trabajo de Grado cumple con las normas de la carrera de ingeniería de Sistemas de la Universidad Pública de El Alto.

El incumplimiento de lo declarado da lugar a responsabilidad del declarante, en consecuencia; a través del presente documento asumo frente a terceros, la carrera de ingeniería de sistemas de la Universidad Pública de El Alto toda responsabilidad que pueda derivarse por el Trabajo de Grado presentado.

Fecha: 18 de noviembre de 2024

Univ. Wilson Viracocha Laura

C.I. 8377140 LP

AGRADECIMIENTOS

Agradezco, en primer lugar, a Dios, por darme la vida y la fortaleza para alcanzar este objetivo tan importante en mi vida académica y personal.

A mis padres, Eleodoro Viracocha y Victoria Laura, quienes con su amor, esfuerzo y apoyo incondicional me han guiado y motivado en cada paso. Su dedicación y sus enseñanzas me han dado la fuerza para perseverar.

Expreso también mi sincero agradecimiento a mis tutores, quienes con su conocimiento y orientación hicieron posible la culminación de este proyecto:

- A la M.Sc.Lic.Ing. Marisol Arguedas Balladares, mi tutora metodológica, por su paciencia por su disponibilidad de tiempo y guía en el desarrollo de esta investigación.
- Al Lic.Ing. Luis Francisco Páez Rocha, tutor especialista, por su invaluable experiencia y aportes técnicos.
- A la Lic.Ing. Yuly Ramírez Limachi, mi tutora revisora, por sus observaciones y dedicación en la revisión de este trabajo.

agradezco a Jhenny Pilco, por su apoyo constante y palabras de ánimo en los momentos difíciles.

A todos ustedes, mi más sincero agradecimiento.

Wilson Viracocha Laura

DEDICATORIA

Dedico este trabajo de grado con profundo cariño y gratitud a Dios, por darme la vida, la salud y la fortaleza para alcanzar este logro.

A mis queridos padres, Eleodoro Viracocha y Victoria Laura, quienes con su amor, esfuerzo y sacrificio me han acompañado en cada paso de este camino. Gracias por enseñarme el valor del trabajo y la perseverancia, por sus consejos y apoyo incondicional en cada momento. Este logro es también suyo.

Wilson Viracocha Laura

1. CAPÍTULO I 1
1.1. INTRODUCCIÓN 1
1.2. ANTECEDENTES 1
1.2.1. Antecedentes de la Institución 1
1.2.1.1. Misión 2
1.2.1.2. Visión 2
1.2.1.3. Objetivos 3
1.2.1.4. Organigrama 4
1.2.2. Antecedentes afines al Proyecto de Grado 4
1.2.2.1. Antecedentes Internacionales 4
1.2.2.2. Antecedentes Nacionales 5
1.3. PLANTEAMIENTO DEL PROBLEMA 5
1.3.1. Problema Principal 5
1.3.2. Problemas Secundarios 6
1.4. Formulación del Problema 6
1.5. OBJETIVOS 6
1.5.1. Objetivo General 6
1.5.2. Objetivos Específicos 6
1.6. JUSTIFICACIÓN 7
1.6.1. Justificación Técnica 7
1.6.2. Justificación Económica 7
1.6.3. Justificación Social 8
1.7. METODOLOGÍA 8
1.7.1. Metodología de Desarrollo 8
1.7.2. Métrica de Calidad al Software 8
1.7.3. Estimación de Costos 9

1.7.4. Seguridad 9
1.7.5. Pruebas al Software 10
1.7.5.1. Pruebas Estructurales 10
1.7.5.2. *Pruebas Funcionales* 10
1.8. METODO DE RECOLECCION DE DATOS 11
1.8.1. Entrevistas 11
1.8.2. Encuestas y Cuestionarios 11
1.8.3. Revisión Documental 11
1.9. HERRAMIENTAS DE DESARROLLO 11
1.9.1. Lenguaje de Programación PHP 12
1.9.2. Gestor de Base de Datos MySQL 12
1.9.3. Framework Laravel 12
1.9.4. HTML 13
1.9.5. CSS 13
1.9.6. JavaScript 13
1.9.7. Framework Bootstrap 5 14
1.9.8. XAMPP 14
1.9.9. Plugins 14
1.9.10. Visual Studio Code 15
1.10. LIMITES Y ALCANCES 15
1.10.1. Limites 15
1.10.2. Alcances 15
1.11. APORTES 16
1.11.1. Teórico 16
1.11.2. Práctico 16
2. CAPÍTULO II 18

2.1. DIRECCION DE ALUMBRADO PUBLICO....18
2.1.1. Hojas de Ruta....18
2.1.2. Registro....18
2.1.3. Seguimiento....19
2.1.4. Control....19
2.1.5. Eficiencia....19
2.1.6. Luminarias Públicas....20
2.1.7. Materiales....20
2.1.8. Herramientas....20
2.2. SISTEMAS WEB....21
2.2.1. Arquitectura de los Sistemas Web....21
2.2.2. Tecnologías y Herramientas....22
2.3. EFICIENTE....23
2.4. ALUMBRADO PÚBLICO....24
2.4.1. Dimensiones y Espaciamiento....24
2.4.2. Lámparas y Luminarias....25
2.4.3. Componentes....25
2.4.4. Tipos de Luminarias....27
2.4.4.1. Luminarias HID....27
2.4.4.2. Luminaria LED....28
2.5. METODOLOGIA....28
2.5.1. Metodología de Desarrollo UWE....28
2.5.1.1. Análisis de Requerimientos....28
2.5.1.2. Diseño del Sistema....29
2.5.1.3. Codificación del Software:....30
2.5.1.4. Pruebas....30

2.5.1.5. Instalación o Fase de Implementación 31
2.5.1.6. Mantenimiento 31
2.5.2. Ciclos de la Metodología UWE 31
2.6. FRAMEWORKS Y TECNOLOGIAS WEB 35
2.6.1. PHP 35
2.6.2. Framework Laravel v10 37
2.6.3. MySQL 39
2.6.4. HTML, CSS y JavaScript en el Desarrollo Web 40
2.6.5. Bootstrap 5 41
2.6.6. XAMPP 42
2.6.7. Plugins 43
2.6.8. Visual Studio Code 44
2.7. PRUEBAS DE SOFTWARE 46
2.7.1. Tipos de Pruebas 46
2.7.1.1. Pruebas Funcionales 46
2.7.1.2. Pruebas Estructurales 46
2.7.2. Importancia de las Pruebas en el Desarrollo de Sistemas Web 47
2.8. SEGURIDAD EN SISTEMAS WEB 48
2.8.1. Estándar ISO 27001 48
2.8.2. Importancia de la Seguridad en Sistemas de Gestión Municipal 50
2.9. CICLO DE VIDA DEL SISTEMA 50
2.10. METRICAS DE CALIDAD DE SOFTWARE 53
2.10.1. ISO/IEC 25000 53
2.10.1.1. ISO/IEC 25010: Modelo de Calidad del Producto Software 54
2.11. ESTIMACIÓN DE COSTOS EN PROYECTOS DE SOFTWARE 55
2.11.1. COCOMO II 55

3. CAPÍTULO III 58
3.1. DISEÑO METODOLOGICO UWE 58
3.1.1. Fase de Análisis de Requisitos 58
3.1.1.1. Análisis de Situación Actual 58
3.1.1.2. Ingeniería de Requerimientos 58
3.1.1.3. Requerimientos Funcionales 59
3.1.1.4. Requerimientos No Funcionales 60
3.1.1.5. Identificación de Actores 60
3.1.2. Diagrama de Caso de Uso General 63
3.1.3. Diagrama de Caso de Uso Dirección 63
3.1.4. Diagrama de Caso de Uso Distrital 64
3.2. FASE DE DISEÑO 65
3.2.1. Modelo de Caso de Uso 65
3.2.2. Diagrama de Clases 75
3.2.3. Modelo Conceptual 77
3.2.4. Modelo Navegacional 78
3.2.4.1. Modelo Navegacional Administrador 78
3.2.4.2. Modelo Navegacional Veedor 79
3.2.4.3. Modelo Navegacional Coordinador 79
3.2.4.4. Modelo Navegacional Técnico 80
3.2.5. Modelo de Presentación 80
3.3. FASE DE CODIFICACION 89
3.4. PRUEBAS DEL SISTEMA 132
3.4.1. Pruebas Estructurales 132
3.4.2. Pruebas Funcionales 138
4. CAPÍTULO IV 141

4.1. CICLO DE VIDA DEL SISTEMA 141
4.2. METRICAS DE CALIDAD DE SOFTWARE 142
4.2.1. Funcional 142
4.2.2. Confiabilidad 145
4.2.3. Usabilidad 147
4.2.4. Eficiencia 149
4.2.5. Mantenibilidad 150
4.2.6. Portabilidad 151
4.3. ESTIMACION DE COSTOS 153
4.3.1. COCOMO II 153
4.4. SISTEMA DE GESTION DE SEGURIDAD DE LA INFORMACION 156
4.4.1. Norma ISO 27001 157
4.4.1.1. Autenticación, Autorización y Control de Acceso 157
4.4.1.2. Encriptación de Contraseña 158
4.4.1.3. Csrf en Formularios 158
4.5. PRUEBAS 159
4.5.1. Introducción 159
4.5.2. Pruebas de Estructurales 159
4.5.3. Descripción General 159
4.5.3.1. Áreas Probadas 159
4.5.3.2. Resultados 161
4.5.4. Pruebas Funcionales 161
4.5.4.1. Descripción General 161
4.5.4.2. Casos de Prueba Ejecutados 161
4.5.4.3. Resultados y Correcciones 165
4.5.5. Pruebas de Seguridad 165

4.5.5.1. Controles Implementados 165

4.5.5.2. Resultados de Pruebas de Seguridad 166

5. CAPÍTULO V 168

5.1. CONCLUSIONES 168

5.1.1. Conclusiones Objetivos Específicos 168

5.2. RECOMENDACIONES 169

6. BIBLIOGRÁFIA 171

INDICE DE *FIGURAS*

Pagina

Figura 1. Organigrama Dirección de Alumbrado Público 4

Figura 2. Cono de luz 24

Figura 3. Lámpara 25

Figura 4. Balasto 26

Figura 5. Condensadores 26

Figura 6. Soportes 27

Figura 7. Luminarias HID 27

Figura 8. Luminaria LED 28

Figura 9. Análisis de Requerimientos 29

Figura 10. Diseño de Sistema 30

Figura 11. Estereotipos del Diagrama de Caso de Uso 32

Figura 12. Diagrama de Caso de Uso en la Metodología UWE 32

Figura 13. Diagrama de Contenido en la Metodología UWE 33

Figura 14. Diagrama de Navegación en la Metodología UWE 34

Figura 15. Diagrama de Presentación en la Metodología UWE 34

Figura 16. Diagrama de Caso de Uso General 63

Figura 17. Diagrama de Caso de Uso Dirección 64

Figura 18. Diagrama de Caso de Uso Distrital 65

Figura 19. Caso de uso Administración de Usuario 66

Figura 20. Caso de Uso Dirección 67

Figura 21. Caso de Uso Agendar Trabajo 68

Figura 22. Caso de Uso Detalles de Trabajo 69

Figura 23. *Caso de Uso Proyectos* 70

Figura 24. Caso de Uso Luminarias Retiradas 71

Figura 25. Caso de Uso Inspección 72

Figura 26. Caso de Uso Equipamiento 73

Figura 27. Caso de Uso Accesorios 74

Figura 28. Caso de Uso Urbanización 75

Figura 29. Diagrama de Clases 76

Figura 30. Modelo Conceptual 77

Figura 31. Modelo Navegacional Administrador 78

Figura 32. Modelo Navegacional Veedor 79

Figura 33. Modelo Navegacional Coordinador 79

Figura 34. Modelo Navegacional Técnico 80

Figura 35. Modelo de Presentación - Inicio de Sesión al Sistema 81

Figura 36. Modelo de Presentación - Administración de Usuarios 81

Figura 37. Modelo de Presentación - Detalles de Trabajos 82

Figura 38. Modelo de Presentación - Agendar 82

Figura 39. Modelo de Presentación - Proyectos 83

Figura 40. Modelo de Presentación - Registro de Proyectos 83

Figura 41. Modelo de Presentación - Empezar Instalación de Proyecto 84

Figura 42. Modelo de Presentación - Luminarias Retiradas 85

Figura 43. Modelo de Presentación - Registro de Luminarias Retiradas 85

Figura 44. Modelo de Presentación - Inspecciones 86

Figura 45. Modelo de Presentación - Registrar Inspección 86

Figura 46. Modelo de Presentación - Equipamiento 87

Figura 47. Modelo de Presentación - Registro de Equipamiento 87

Figura 48. Modelo de Presentación - Urbanizaciones 88

Figura 49. Modelo de Presentación - Registrar Urbanización 88

Figura 50. Modelo de Presentación - Realizar Trabajo.....89

Figura 51. Dashboard Actividades de Mantenimiento, Instalación y Proyectos 90

Figura 52. Módulo de Usuarios: Lista de Usuarios 90

Figura 53. Módulo de Usuario: Registrar.....91

Figura 54. Módulo de Usuario: Modificar.....91

Figura 55. Módulo de Usuario: Eliminar 92

Figura 56. Módulo de Usuario: Restablecer Contraseña 92

Figura 57. Módulo de Usuario: Buscar 93

Figura 58. Módulo de Usuario: Exportar.....93

Figura 59. Agendar Trabajo para las Urbanizaciones de los diferentes distritos.....94

Figura 60. Realizar Trabajo: Lista de trabajos pendientes del distrito.....94

Figura 61. Realizar Trabajo: Buscar.....95

Figura 62. Realizar Trabajo: Empezar Mantenimiento.....95

Figura 63. Módulo Trabajos Registrados: Lista de trabajos Pendientes 96

Figura 64. Módulo Trabajos Registrados: Modificar 96

Figura 65. Módulo Trabajos Registrados: Eliminar.....97

Figura 66. Módulo Trabajos Registrados: Exportar 97

Figura 67. Módulo Trabajos Registrados: Buscar.....98

Figura 68. Módulo Trabajos Registrados Realizados: Listar.....98

Figura 69. Módulo Trabajos Registrados Realizados: Modificar.....99

Figura 70. Módulo Trabajos Registrados Realizados: Detalles 99

Figura 71. Módulo Trabajos Registrados Realizados: Eliminar 100

Figura 72. Módulo Trabajos Registrados Realizados: Reporte.....100

Figura 73. Módulo Trabajos Registrados Realizados: Buscar 101

Figura 74. Módulo Trabajos Registrados Realizados: Exporta Lista de Trabajos 101

Figura 75. Módulo Proyectos: Listar.....102

Figura 76. Módulo Proyectos: Empezar Instalación 102

Figura 77. Módulo de Proyecto: Registrar 103

Figura 78. Módulo Proyectos: Modificar 103

Figura 79. Módulo Proyectos: Reporte 104

Figura 80. Módulo Proyectos: Eliminar 104

Figura 81. Módulo Proyectos: Detalles 105

Figura 82. Módulo Proyectos: Buscar 105

Figura 83. Módulo Proyectos: Reporte General 106

Figura 84. Módulo Proyectos Obras Ejecutadas: Listar 106

Figura 85. Módulo Proyectos Obras Ejecutadas: Modificar 107

Figura 86. Módulo Proyectos Obras Ejecutadas: Eliminar 107

Figura 87. Módulo Proyectos Obras Ejecutadas: Reporte 108

Figura 88. Módulo Proyectos Obras Ejecutadas: Detalles 108

Figura 89. Módulo Proyectos Obras Ejecutadas: Buscar 109

Figura 90. Módulo Proyectos Obras Ejecutadas: Reporte General 109

Figura 91. Módulo Accesorios y Luminarias Retiradas: Listar 110

Figura 92. Módulo Accesorios y Luminarias Retiradas: Registrar 110

Figura 93. Módulo Accesorios y Luminarias Retiradas: Modificar 111

Figura 94. Módulo Accesorios y Luminarias Retiradas: Eliminar 111

Figura 95. Módulo Accesorios y Luminarias Retiradas: Reporte 112

Figura 96. Módulo Accesorios y Luminarias Retiradas: Reporte General 112

Figura 97. Módulo Accesorios y Luminarias Retiradas: Detalles 113

Figura 98. Módulo Accesorios y Luminarias Retiradas: Buscar 113

Figura 99. Módulo de Inspección: Listar 114

Figura 100. Módulo de Inspección: Registrar 114

Figura 101. Módulo de Inspección: Modificar 115

Figura 102. Módulo de Inspección: Eliminar.....115

Figura 103. Módulo de Inspección: Reporte General 116

Figura 104. Módulo de Inspección: Buscar 116

Figura 105. Módulo de Inspección Realizadas: Listar 117

Figura 106. Módulo de Inspección Realizadas: Modificar.....117

Figura 107. Módulo de Inspección Realizadas: Eliminar 118

Figura 108. Módulo de Inspección Realizadas: Reporte 118

Figura 109. Módulo de Equipamientos: Por Distritos.....119

Figura 110. Módulo de Equipamientos: Registrar.....119

Figura 111. Módulo de Equipamientos: Listar 120

Figura 112. Módulo de Equipamientos: Modificar.....120

Figura 113. Módulo de Equipamientos: Eliminar 121

Figura 114. Módulo de Equipamientos: Buscar.....121

Figura 115. Módulo de Accesorios: Listar 122

Figura 116. Módulo de Accesorios: Registrar.....122

Figura 117. Módulo de Accesorios: Modificar.....123

Figura 118. Módulo de Accesorios: Eliminar 123

Figura 119. Módulo de Accesorios: Reporte General.....124

Figura 120. Módulo de Accesorios: Buscar 124

Figura 121. Módulo de Reelevamiento: Por Distrito 125

Figura 122. Módulo de Reelevamientos: Registrar.....125

Figura 123. Módulo de Reelevamiento: Listar 126

Figura 124. Módulo de Reelevamiento: Modificar 126

Figura 125. Módulo de Reelevamiento: Eliminar.....127

Figura 126. Módulo de Reelevamientos: Reporte General Por Distrito.....127

Figura 127. Módulo de Reelevamiento: Buscar.....128

Figura 128. Módulo de Urbanización: Listar 128

Figura 129. Módulo de Urbanización: Registrar 129

Figura 130. Módulo de Urbanización: Modificar 129

Figura 131. Módulo de Urbanización: Eliminar 130

Figura 132. Módulo de Urbanización: Reporte General.....130

Figura 133. Módulo de Urbanización: Buscar.....131

Figura 134. Perfil de Usuario.....131

Figura 135. Cambiar Contraseña 132

Figura 136. Diagrama de pruebas estructurales.....133

Figura 137. Autenticación y Validación de Usuarios.....157

Figura 138. Encriptación de Contraseña 158

Figura 139. Csrf en Formularios.....159

Figura 140. Validación de Formulario.....160

Figura 141. Validación de Inicio de Sesión.....160

Figura 142. Verificación de middleware.....161

Figura 143. Verificación de bloqueo 162

Figura 144. Encriptación de contraseñas 162

Figura 145. Validación de Campos Usuarios 163

Figura 146. Validación de Campos Proyectos.....164

Figura 147. Validación de Campos Luminarias Retiradas 165

INDICE DE TABLAS

Pagina

Tabla 1. Requerimientos Funcionales 59

Tabla 2. Requerimientos No Funcionales 60

Tabla 3. Especificación de Caso de Uso Administración de Usuarios 66

Tabla 4. Especificación de Caso de Uso Dirección.....67

Tabla 5. Especificación de Caso de Uso Agendar Trabajo 68

Tabla 6. Especificación de Caso de Uso Detalles de Trabajo.....69

Tabla 7. Especificación de Caso de Uso Proyectos.....70

Tabla 8. Especificación de Caso de Uso Luminarias Retiradas 72

Tabla 9. Especificación de Caso de Uso Inspección.....73

Tabla 10. Especificación de Caso de Uso Equipamiento.....73

Tabla 11. Especificación de Caso de Uso Accesorios 74

Tabla 12. Especificación de Caso de Uso Urbanización.....75

Tabla 13. Prueba Funcionales.....138

Tabla 14. Parámetros de Entrada.....143

Tabla 15. Ajuste de Complejidad.....144

Tabla 16. Parámetros de medición de usabilidad 148

Tabla 17. Preguntas de Usabilidad del Sistema 148

Tabla 18. Escala de Valores de Eficiencia.....149

Tabla 19. Valoración de la Eficiencia del Sistema 149

Tabla 20. Valores para la Mantenibilidad.....151

Tabla 21. Resultados.....152

Tabla 22. Coeficientes del Modelo COCOMO II 153

Tabla 23. Ecuaciones del Modelo COCOMO II.....154

Tabla 24. Cálculo de los Atributos FAE 154

RESUMEN

El presente proyecto detalla el desarrollo de un sistema web para el registro, seguimiento y control eficiente de luminarias públicas, materiales y herramientas en el Gobierno Autónomo Municipal de El Alto. Se establece como objetivo principal mejorar la gestión de información y el seguimiento de trabajos, proporcionando datos confiables y de fácil acceso para optimizar el mantenimiento y la planificación de recursos. Se implementa la metodología UWE (UML-Based Web Engineering) que permite un análisis sistemático y desarrollo estructurado. El sistema utiliza herramientas como PHP con el framework Laravel 10, MySQL para la base de datos, y Bootstrap 5 para el frontend, garantizando una solución robusta y escalable. La arquitectura implementada sigue el patrón MVC y cumple con estándares de seguridad ISO 27001. Las pruebas realizadas demuestran un 92% de efectividad según las métricas de calidad ISO/IEC 25000, con resultados destacables en usabilidad (96%) y eficiencia (96%). El sistema optimiza significativamente los procesos de gestión municipal, reduciendo errores en el registro y seguimiento de luminarias, y facilitando el control de recursos. Los resultados evidencian una mejora sustancial en la administración de la información y la toma de decisiones en la gestión del alumbrado público municipal.

Palabras clave: *sistema web, Gestión municipal, Luminarias públicas*

GLOSARIO DE ABREVIATURAS

API (Application Programming Interfaces): Interfaces de Programación de Aplicaciones

CMS (Content Management System): Sistema de Gestión de Contenidos

COCOMO II (Constructive Cost Model II): Modelo Constructivo de Costos II

CSS (Cascading Style Sheets): Hojas de Estilo en Cascada

DOM (Document Object Model): Modelo de Objetos del Documento

GAMEA: Gobierno Autónomo Municipal de El Alto

HTML (HyperText Markup Language): Lenguaje de Marcado de Hipertexto

HTTP (Hypertext Transfer Protocol): Protocolo de Transferencia de Hipertexto

ISO (International Organization for Standardization): Organización Internacional de Normalización

KLOC (Kilo Lines of Code): Miles de Líneas de Código

MVC (Model-View-Controller): Modelo-Vista-Controlador

ORM (Object-Relational Mapping): Mapeo Objeto-Relacional

PDCA (Plan-Do-Check-Act): Planificar-Hacer-Verificar-Actuar

PHP (Hypertext Preprocessor): Preprocesador de Hipertexto

RDBMS (Relational Database Management System): Sistema de Gestión de Bases de Datos Relacionales

SEO (Search Engine Optimization): Optimización para Motores de Búsqueda

SQL (Structured Query Language): Lenguaje de Consulta Estructurada

UML (Unified Modeling Language): Lenguaje Unificado de Modelado

UWE (UML-Based Web Engineering): Ingeniería Web Basada en UML

CAPÍTULO I
MARCO PRELIMINAR

INGENIERÍA
DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO

1. CAPÍTULO I

1.1. INTRODUCCIÓN

Las tecnologías de la información facilitan tareas antes manuales, permitiendo un acceso ágil a la información. Un sistema web organiza y gestiona datos en tiempo real, optimizando procesos.

El desarrollo urbano de la ciudad de El Alto requiere sistemas eficientes para gestionar problemas de alumbrado público. Por ello, la Alcaldía (GAMEA) implementa un “Sistema Web para el Registro, Seguimiento y Control Eficiente de Luminarias Públicas, Materiales y Herramientas”.

La zona urbana crece continuamente, demandando una mejor gestión de alumbrado. Hasta ahora, la Alcaldía usaba métodos manuales, generando desorganización, retrasos, y mala administración de datos.

Para el desarrollo del sistema web se utilizará la metodología de desarrollo de software UWE y para el funcionamiento del sistema web por parte de Backend y Frontend se aplicarán las siguientes herramientas: Lenguaje de programación PHP, Framework Laravel, Gestor de Base de Datos MySQL, HTML (lenguaje de marcas de hipertexto), Framework Bootstrap, administrador de base de datos DBeaver, para la parte de prototipado del sistema Balsamiq Mockups y otros.

1.2. ANTECEDENTES

1.2.1. *Antecedentes de la Institución*

El Gobierno Autónomo Municipal de El Alto es el nombre oficial de la instancia de GAMEA, que administra el territorio comprendido por el municipio.

La entidad inicio su funcionamiento bajo el nombre de Alcaldía tras la creación de la entidad territorial en 1985. Previamente este territorio era administrado como parte del municipio vecino de La Paz.

De acuerdo a la ley de Autonomías y la organización territorial de Bolivia los municipios bolivianos tienen la potestad de elegir sus alcaldes en elecciones locales.

El Gobierno Autónomo Municipal de El Alto, también conocido por sus siglas GAMEA, se compone del poder ejecutivo, representado por el alcalde y su equipo, y el consejo municipal constituido por representantes elegidos igualmente por voto popular a través de elecciones municipales cada 5 años. Para el periodo 2020-2025 se eligió a Eva Copa como alcaldesa de acuerdo a los resultados de las elecciones Subnacionales en Bolivia.

El municipio de El Alto se halla dividido para su administración, en 14 distritos, 4 de ellos rurales, cada uno bajo una Sub-Alcaldía y una autoridad denominada Subalcaide, a través de esas entidades se descentralizan algunas actividades administrativas.

1.2.1.1. **Misión**

El Alto ciudad emergente con alma y espíritu de unidad, con planificación del bienestar social, participativo, desconcentrado, incluyente que garantiza derechos, seguridad del espacio público, con acceso a la salud, educación y protagonista del desarrollo nacional.

1.2.1.2. **Visión**

Gobierno Autónomo Municipal de El Alto, con identidad, ciudad ajayu con derecho a la salud, educación y sin violencia, con empleo, modernidad, industrias y emprendimientos. Para el desarrollo sustentable y planificado, articulador de la región metropolitana.

1.2.1.3. **Objetivos**

Realiza, implementar y ejecutar proyectos de alumbrado público: de acuerdo a competencias, así mismo realizar el mantenimiento preventivo, correctivo y ampliación de la cobertura del sistema de alumbrado público en el municipio de El Alto, en cumplimiento de la norma boliviana de alumbrado público a solicitud y requerimiento de la dirección de atención Ciudadana y población en general.

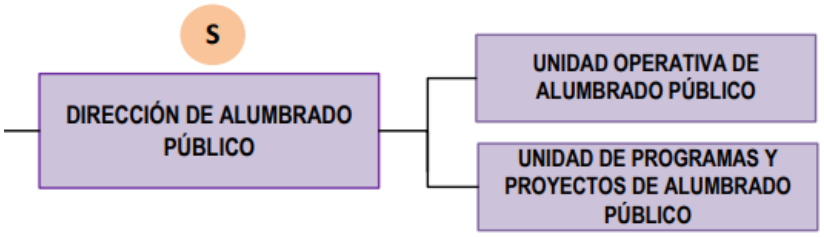
- Realizar los parámetros técnicos y las especificaciones técnicas para proyectos de alumbrado público
- Elaborar y ejecutar proyectos de alumbrado público de acuerdo a las competencias de la dirección de atención ciudadana de la secretaria municipal gestión institucional
- Coadyuvar y coordinar con las Sub-Alcaldías mediante la unidad de coordinación con sub alcaldías dependiente de la dirección de atención ciudadana. Actividades relacionadas a iluminación urbana de los diferentes distritos municipales
- Planificar, diseñar, implementar y realizar el seguimiento de nuevas aplicaciones tecnológicas para soluciones tempranas en las organizaciones institucionales e interinstitucionales con el fin de construir políticas públicas que incluyan el desarrollo digital, así mismo generar campañas de ahorro de energía eléctrica, impulsar la implementación de nuevas y mejores tecnologías
- Realizar el cronograma de salidas de carros canastas para el mantenimiento del sistema de alumbrado público
- Otras funciones asignadas por inmediato superior y la máxima autoridad ejecutiva

1.2.1.4. **Organigrama**

Este describe la estructura jerárquica y funcional de la organización encargada de gestionar y mantener el sistema de iluminación pública en la ciudad de El Alto, como se ve en la Figura 1.

Figura 1.

Organigrama Dirección de Alumbrado Público



Nota. Organigrama de la dirección de alumbrado público. (Gobierno Autónomo Municipal de El Alto, 2024).

La Dirección de Alumbrado Público se encarga de gestionar todas las dotaciones de luminarias y accesorios que se realizan en las diferentes urbanizaciones de la ciudad de El Alto. Asimismo, es responsable del mantenimiento y la instalación.

1.2.2. ***Antecedentes afines al Proyecto de Grado***

1.2.2.1. **Antecedentes Internacionales**

- (Rubén Alejandro Mora Romero, Liz Joseane Sánchez Jarquín, Álvaro Rafael Blanco Doña, 2018) “sistema web para el control y registro de los proyectos investigativos en la dirección de grado y postgrado de unan” Desarrollar un sistema web para el control y registro de los proyectos investigativos en la dirección de Grado y Postgrado de UNAN. (Managua, 2018)
- (Jorge Enrique Mendoza Rivilla, 2017) “implementación de sistemas web para la gestión y control de los procesos de la unidad de titulación de la carrera de

ingeniería en sistemas de la universidad salesiana” Implementar una herramienta que cubra las necesidades de la carrera Ingeniería en Sistemas de la Universidad Politécnica Salesiana sede guayaquil, para la gestión y seguimiento en cada una de las etapas de las respectivas opciones de titulación mediante una aplicación web.(Rivilla & Enrique, s. f.)

1.2.2.2. **Antecedentes Nacionales**

- (Max Juan Mamani Quispe, 2020) “sistema web de registro y control de programas educativos” Diseñar, desarrollar e implementar un sistema web de registro y control para la unidad de programas educativos del Gobierno Autónomo Municipal de El Alto. (Quispe, 2020)
- (Jeaneth Delicia Bautista Luque, 2020) “sistema de información web para la gestión de compra y venta e inventarios de componentes electrónicos caso: empresa techsbol” Desarrollar un Sistema Web para la gestión de compra y venta e inventarios de componentes electrónicos de la empresa TECHSBOL que coadyuve en la administración y control de la información más adecuado y eficiente en las actividades.(Luque, 2020)

1.3. **PLANTEAMIENTO DEL PROBLEMA**

1.3.1. ***Problema Principal***

El Gobierno Autónomo Municipal de El Alto enfrenta desafíos en el registro, seguimiento y control de luminarias, materiales y herramientas, lo que genera información dispersa, desactualizada, dificultando así una gestión eficiente, lo que puede conllevar a posibles procesos administrativos y judiciales. (Ver Anexo A)

1.3.2. *Problemas Secundarios*

- La información no está centralizada, por lo que no se puede controlar la existencia y el estado de material como luminarias, brazos, cables y herramientas.
- El seguimiento de las luminarias es tradicional debido a que dicha información se encuentra dispersa en forma física, así como en hojas de cálculo y procesador de palabras, lo que conlleva a una mala administración.
- Descontrol en las hojas de rutas, generalmente se encuentran archivadas y difícil de saber si fueron atendidas o despachadas, lo que ocasiona una planificación ineficiente.

1.4. **Formulación del Problema**

¿De qué manera se podrá mejorar el registro, seguimiento y control de luminarias materiales y herramientas, en el Gobierno Autónomo Municipal de El Alto para evitar la generación de información dispersa y desactualizada que dificulte una gestión eficiente?

1.5. **OBJETIVOS**

1.5.1. *Objetivo General*

Desarrollar un Sistema Web para el Registro, Seguimiento y Control eficiente de luminarias públicas, materiales y herramientas en la ciudad de El Alto, tal que genere información confiable y oportuna para una adecuada toma de decisiones.

1.5.2. *Objetivos Específicos*

- Identificar los procesos actuales para definir los requisitos.
- Realizar un análisis de requerimientos necesarios para el desarrollo del sistema web.

- Diseñar una base de datos acorde a todos los requerimientos obtenidos para el desarrollo del sistema web.
- Sistematizar la información para una administración transparente.
- Optimizar la gestión de la información para facilitar el registro y control, mejorando así la eficiencia y el mantenimiento.

1.6. JUSTIFICACIÓN

1.6.1. *Justificación Técnica*

La Alcaldía cuenta con todos los recursos necesarios para el desarrollo e implementación del sistema web como ser:

- Equipos de computación
- Servicio de internet
- Centro de datos
- Hosting
- dominio

1.6.2. *Justificación Económica*

El proyecto está desarrollado con lo que es el software libre, lo que indica de no gastos en la parte de licencias.

Toda la información que se maneja en la Alcaldía de la ciudad de El Alto es de vital importancia, este sistema web está dedicado para el área de luminarias públicas, una herramienta necesaria y fundamental ya que no se cuenta con ningún sistema de este tipo, gracias a esta herramienta se optimizará procesos como también costos asociados con el almacenamiento físico de estas, la impresión de documentos, el envío de correspondencias, evitar posibles procesos administrativos y judiciales que deba asumir la alcaldia y diferentes gastos operativos.

1.6.3. *Justificación Social*

El desarrollo de un sistema web para el registro, seguimiento y control de luminarias públicas, materiales y herramientas es crucial para mejorar la gestión del alumbrado público en la ciudad de El Alto. Los informes y datos generados por el sistema no solo facilitarán el acceso a información precisa y detallada, sino que también optimizarán el proceso de auditoría. La centralización de datos en una base de datos robusta permitirá al director, jefe de unidad y a los técnicos de cada distrito gestionar eficientemente los recursos disponibles. La capacidad de emitir informes de forma oportuna fortalecerá la toma de decisiones, impactando positivamente en la calidad del servicio y en la confianza de la ciudadanía hacia las autoridades.

1.7. **METODOLOGÍA**

1.7.1. *Metodología de Desarrollo*

La elección de una metodología de desarrollo que combine enfoques investigativos y descriptivos es fundamental para el éxito del sistema. La metodología UWE permite un análisis profundo y sistemático, lo cual es necesario para abordar las particularidades del entorno urbano de El Alto. Al utilizar UML, se asegura que el diseño del sistema sea claro y estructurado, facilitando la comunicación entre los involucrados y la comprensión de los procesos. Las fases clave de esta metodología garantizan que cada aspecto del desarrollo se aborde con rigurosidad, lo que resulta en un sistema más eficiente y alineado con las necesidades reales de la comunidad.

1.7.2. *Métrica de Calidad al Software*

El estándar ISO/IEC 25000 presenta un marco conceptual para el modelo de calidad y define un conjunto de características y sub características, las que debe cumplir todo producto software.(*NORMAS ISO 25000*, s. f.)

La norma ISO/IEC 25000, también conocida como SQuaRE (System and Software Quality Requirements and Evaluation), proporciona un marco para la evaluación y gestión de la calidad de productos de software y sistemas informáticos. Está diseñada para ayudar a los desarrolladores, evaluadores y usuarios a definir y medir la calidad del software de manera estructurada y objetiva. La norma organiza la calidad en 8 características principales, como adecuación funcional, usabilidad, eficiencia de rendimiento, y seguridad, entre otras, cada una con subcaracterísticas específicas. Con ISO/IEC 25010, las organizaciones pueden asegurar que el software cumple con los estándares de calidad requeridos para satisfacer las necesidades y expectativas de los usuarios finales (Organización Internacional de Normalización, 2014).

1.7.3. ***Estimación de Costos***

La estimación de costos utilizando el modelo COCOMO permite calcular el esfuerzo y tiempo necesarios para desarrollar un proyecto de software basado en varios factores que afectan su complejidad y tamaño. COCOMO II se estructura en tres submodelos: el modelo de aplicaciones tempranas (Early Design Model), el modelo de arquitectura y diseño posterior (Post-Architecture Model), y el modelo de componentes reutilizables. Cada uno de estos modelos se adapta a las distintas fases del desarrollo, brindando estimaciones más precisas según la cantidad de información disponible y la madurez del proyecto. El modelo se enfoca en variables como el tamaño del proyecto, expresado en KLOC (líneas de código), así como factores de ajuste que incluyen la complejidad del producto, el equipo y el entorno de desarrollo (Boehm, 2000).

1.7.4. ***Seguridad***

En cuanto a seguridad, para el presente proyecto se usará la norma ISO 27001. Este estándar internacional establece los requisitos para implementar, mantener y mejorar un Sistema de Gestión de Seguridad de la Información (SGSI). Su objetivo principal es proteger

la confidencialidad, integridad y disponibilidad de la información, asegurando que las organizaciones gestionen los riesgos relacionados con la seguridad de manera eficaz y sistemática. (Normas ISO, 2018)

1.7.5. ***Pruebas al Software***

Este es un proceso de evolución donde vemos el funcionamiento del software que se está desarrollando como también los errores que se tiene hasta ahora para ver que cumplan todos los estándares requeridos, esto nos ayudara a poder corregirlos antes que el sistema esté en funcionamiento

1.7.5.1. **Pruebas Estructurales**

Las pruebas estructurales constituyen una técnica de evaluación que se enfoca en examinar la estructura interna del software. Este método implica un análisis profundo del código fuente para verificar la lógica del programa, las rutas de ejecución y los flujos de control, permitiendo identificar posibles defectos en la implementación del sistema (Spillner et al., 2021).

1.7.5.2. **Pruebas Funcionales**

Las pruebas funcionales evalúan el comportamiento del software desde una perspectiva externa, centrándose exclusivamente en verificar que las entradas produzcan los resultados esperados. En este enfoque, el evaluador se concentra en validar que el sistema responda correctamente a las acciones del usuario según los requerimientos establecidos, sin necesidad de conocer cómo se procesa internamente la información (Myers et al., 2023).

1.8. METODO DE RECOLECCION DE DATOS

1.8.1. *Entrevistas*

La entrevista es una técnica cualitativa que permite obtener información detallada y profunda directamente de los participantes, basándose en preguntas abiertas. Es ampliamente utilizada en investigaciones sociales y de mercado (Comparasoftware, 2023).

1.8.2. *Encuestas y Cuestionarios*

Las encuestas recopilan datos cuantitativos mediante un conjunto de preguntas estandarizadas dirigidas a un grupo específico. Estas pueden administrarse en línea, en papel o por teléfono, permitiendo una amplia cobertura y resultados estadísticamente significativos (Clientify, 2022).

Los cuestionarios son instrumentos estructurados compuestos por preguntas abiertas o cerradas, diseñados para recopilar datos específicos de los encuestados. Son flexibles en su formato y pueden ser administrados presencialmente o de forma remota (Clientify, 2022).

1.8.3. *Revisión Documental*

La revisión documental se centra en la recopilación y análisis de información contenida en documentos previamente elaborados, como libros, artículos, informes y bases de datos. Este método es clave en investigaciones bibliográficas y hemerográficas, ya que permite interpretar datos secundarios para complementar los hallazgos empíricos. Según Cubillo (2009), esta técnica facilita la obtención de un contexto más amplio y enriquecedor sobre el tema de estudio

1.9. HERRAMIENTAS DE DESARROLLO

Para el desarrollo de este sistema web utilizaremos todas las herramientas que mencionaremos a continuación con el fin de desarrollar un sistema web optimo

1.9.1. *Lenguaje de Programación PHP*

Es un lenguaje de código abierto muy popular, adecuado para el desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.(Souza, 2020)

1.9.2. *Gestor de Base de Datos MySQL*

Schwartz et al. (2012) describen a MySQL como el sistema de gestión de bases de datos de código abierto más utilizado en Internet, con una amplia presencia en servidores y proyectos web.

En su configuración como sistema gestor de bases de datos relacionales, MySQL destaca por su robustez y versatilidad, características que lo han posicionado como una solución efectiva para diversos tipos de aplicaciones y sitios web (Schwartz et al., 2012).

Una de las ventajas fundamentales de MySQL es su naturaleza multiplataforma, que permite su implementación en diferentes sistemas operativos como Windows, Linux y Mac. Se destaca particularmente su integración en entornos Linux, donde forma parte esencial de la arquitectura LAMP, trabajando en conjunto con Linux, Apache y PHP (Schwartz et al., 2012).

1.9.3. *Framework Laravel*

Según Stauffer (2019), Laravel se ha consolidado como el framework PHP líder en el desarrollo backend, destacando por su enfoque moderno y su amplia adopción en la comunidad de desarrollo web. Este framework proporciona una robusta base de código que optimiza el desarrollo y mantenimiento de aplicaciones web mediante dos ventajas fundamentales: una arquitectura refinada que promueve altos estándares de calidad en el

desarrollo, y un conjunto completo de bibliotecas que simplifican la implementación de funcionalidades comunes.

La versatilidad de Laravel permite abordar proyectos de cualquier escala, desde pequeñas aplicaciones hasta sistemas empresariales complejos. Como señala Stauffer (2019), el framework enfatiza tres aspectos cruciales: la calidad del código, la facilidad de mantenimiento y la escalabilidad. Su estructura promueve la estandarización del desarrollo, lo que facilita el trabajo colaborativo y mejora la mantenibilidad del código a largo plazo.

1.9.4. ***HTML***

Es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponden a HyperText Markus Language, es decir, Lenguaje de Marcas de Hipertexto Según (Herrera, 2015).

1.9.5. ***CSS***

Son las siglas en inglés para hojas de estilo en cascada (Cascading Style Sheets). Básicamente, es un lenguaje que maneja el diseño y presentación de las páginas web, es decir, cómo lucen cuando un usuario las visita. Funciona junto con el lenguaje HTML que se encarga del contenido básico de los sitios (Santos, s. f. 2019).

1.9.6. ***JavaScript***

Es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado. Es la tercera capa del pastel de las tecnologías web estándar (HTML y CSS. 2019)**.**

1.9.7. ***Framework Bootstrap 5***

Bootstrap surgió en 2010 como un framework CSS desarrollado por Twitter, evolucionando hasta convertirse en una herramienta integral para el desarrollo web moderno. Como señalan Otto y Thornton (2022), este framework integra CSS y JavaScript para crear interfaces web dinámicas, ofreciendo una amplia gama de componentes predefinidos como menús de navegación, controles de página y barras de progreso, que mejoran significativamente la interacción con el usuario.

La característica distintiva de Bootstrap es su enfoque en el diseño responsive. Los autores enfatizan que el framework permite desarrollar sitios web que se adaptan automáticamente a diferentes dispositivos como computadoras de escritorio, tablets y smartphones, manteniendo una organización coherente y una experiencia de usuario optimizada (Otto & Thornton, 2022).

1.9.8. ***XAMPP***

Dvorski (2007) define XAMPP como un entorno de desarrollo local multiplataforma que integra un conjunto de herramientas esenciales para el desarrollo web, incluyendo un servidor web y sistemas de gestión de bases de datos, lo que facilita la creación y prueba de aplicaciones web en un entorno controlado.

1.9.9. ***Plugins***

Los plugins, según describe Erixon (2018), son extensiones de software que añaden funcionalidades específicas a aplicaciones existentes sin alterar su código base. Su aplicación más común se encuentra en navegadores web, donde permiten el procesamiento de diferentes tipos de contenido, como la visualización de documentos PDF, extendiendo así las capacidades nativas del navegador.

1.9.10. ***Visual Studio Code***

Es un editor de código fuente liviano pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con suporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes y tiempos de ejecución (como C++, C#, Java, Python, PHP, Go, .NET).(Documentation for Visual Studio Code, s. f. 2019).

1.10. **LIMITES Y ALCANCES**

1.10.1. ***Limites***

Este proyecto está pensado para el área de alumbrado público de la Alcaldía, de acuerdo a sus necesidades, especificaciones y requerimientos.

- Este sistema no podrá realizar ningún control fuera de los 14 distritos de la ciudad de El Alto
- No se podrá compartir información con sistemas externos o internos
- Este sistema no podrá ser utilizado en otras áreas o unidades
- El sistema web contara con niveles de administración, por lo que no todos los usuarios tendrán acceso a la misma información
- El sistema solo almacenara información de todos los procesos en la dirección de alumbrado público

1.10.2. ***Alcances***

Para la implementación de este proyecto, se alcanzará los siguientes módulos:

- Módulo de Usuario
- Módulo de Trabajos Registrados
- Módulo de Proyectos

- Módulo de Luminarias Retiradas
- Módulo de Inspecciones
- Módulo de Equipamientos
- Módulo de Accesorios
- Módulo de Reelevamientos de Luminarias
- Módulo de Urbanizaciones

1.11. **APORTES**

1.11.1. ***Teórico***

- Implementar medidas de seguridad para proteger el sistema contra posibles ataques, garantizando la integridad y la confidencialidad de los datos.
- Utilizar herramientas de análisis de datos para procesar la información recopilada por las luminarias y generar informes que ayuden en la toma de decisiones

1.11.2. ***Práctico***

- Implementar el sistema de control central con un enfoque modular y escalable.
- Realizar pruebas exhaustivas del sistema en entornos controlados y en condiciones del mundo real para garantizar su rendimiento y confiabilidad.
- Proporcionar documentación clara para la instalación, configuración y mantenimiento del sistema
- Ofrecer capacitación a los técnicos y personal de mantenimiento que trabajara con el sistema

CAPÍTULO II
MARCO TEÓRICO

INGENIERÍA
DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO

2. CAPÍTULO II

2.1. DIRECCION DE ALUMBRADO PUBLICO

Según el Gobierno Autónomo Municipal de El Alto (2021) se encarga de implementar y ejecutar proyectos de alumbrado público en el municipio de El Alto, incluyendo mantenimiento preventivo, correctivo y ampliación de cobertura, cumpliendo la normativa boliviana. Realizar especificaciones técnicas y coordinar con sub-alcaldías para proyectos de iluminación urbana. Planificar, diseñar y supervisar nuevas tecnologías para mejorar el sistema de alumbrado, promover el ahorro energético y fomentar el desarrollo digital. Elaborar el cronograma de mantenimiento de vehículos de alumbrado. Otras tareas asignadas por la autoridad competente.

2.1.1. *Hojas de Ruta*

En la ciudad de El Alto, la gestión de hojas de ruta es un proceso administrativo esencial para la tramitación de solicitudes y documentos oficiales. Las hojas de ruta permiten el seguimiento adecuado de los trámites, garantizando que cada solicitud sea atendida dentro de los plazos establecidos. Según el Gobierno Autónomo Municipal de El Alto (2021), estas hojas son documentos internos que facilitan la comunicación y coordinación entre las diferentes unidades administrativas, asegurando una mayor transparencia y eficiencia en la atención a la ciudadanía.

2.1.2. *Registro*

Laudon y Laudon (2018) definen el registro en sistemas de información como un proceso sistemático de documentación y organización de datos que facilita su posterior consulta y actualización. En el contexto de este proyecto, este concepto se aplica específicamente a la gestión de luminarias públicas, donde se documenta de manera detallada su ubicación, tipología, estado operativo y especificaciones técnicas, junto con el inventario de

materiales y herramientas empleados en su mantenimiento. Este sistema digital de registro funciona como un inventario comprehensivo que permite mantener un control preciso y actualizado de todos los componentes y recursos involucrados en el sistema de iluminación pública.

2.1.3. ***Seguimiento***

El Project Management Institute (2017) define el seguimiento como un proceso continuo de monitoreo y evaluación que garantiza el cumplimiento de objetivos en sistemas de gestión. En el contexto de las luminarias públicas, este proceso implica una supervisión sistemática que abarca todo el ciclo de vida del equipamiento, monitoreando cada etapa desde su instalación inicial hasta su mantenimiento y eventual reemplazo, permitiendo así una gestión eficiente y proactiva de la infraestructura de iluminación pública.

2.1.4. ***Control***

Según Daft y Marcic (2020), el control de sistemas implica una gestión eficiente de recursos mediante la supervisión y dirección sistemática para alcanzar los objetivos organizacionales. En el contexto del sistema de luminarias públicas, esto se traduce en una administración precisa de la distribución y utilización de recursos, incluyendo el monitoreo detallado de materiales, herramientas y equipamiento en cada ubicación, lo que permite optimizar las tareas de mantenimiento y reparación mediante un seguimiento exhaustivo del inventario y su localización en las diferentes urbanizaciones.

2.1.5. ***Eficiencia***

Heizer et al. (2020) conceptualizan la eficiencia como la optimización en el uso de recursos y tiempo para alcanzar objetivos específicos. En el contexto del sistema web para la gestión de luminarias públicas, esta eficiencia se materializa a través de la automatización de procesos operativos, que incluye la optimización del mantenimiento, la gestión de inventarios

y la toma de decisiones basada en datos precisos y actualizados. Este enfoque sistemático permite minimizar errores humanos y maximizar la efectividad en la administración de recursos, asegurando una gestión más precisa y ágil de la infraestructura de iluminación pública.

2.1.6. ***Luminarias Públicas***

Las luminarias públicas, como describe Boyce (2014), son componentes esenciales de la infraestructura urbana que proporcionan iluminación en espacios públicos como calles, parques y plazas, contribuyendo significativamente a la seguridad y calidad de vida de los ciudadanos. El sistema web propuesto aborda la gestión integral de esta infraestructura crítica, implementando un ciclo completo de administración que abarca desde la instalación inicial hasta el mantenimiento preventivo y correctivo, asegurando así un servicio de iluminación pública eficiente y sostenible que satisface las necesidades de la comunidad urbana.

2.1.7. ***Materiales***

En el marco del proyecto, Stevenson (2018) define los materiales como el conjunto de componentes físicos esenciales para el funcionamiento y mantenimiento de las luminarias públicas, incluyendo bombillas, cables, transformadores y otros elementos críticos. La implementación del sistema web optimiza la gestión de estos recursos mediante un control de inventario automatizado, lo que permite prevenir tanto el desabastecimiento como la acumulación excesiva de stock, asegurando una administración eficiente de los suministros necesarios para el mantenimiento de la infraestructura de iluminación.

2.1.8. ***Herramientas***

Groover (2019) caracteriza las herramientas como instrumentos fundamentales que abarcan desde implementos manuales básicos hasta equipamiento especializado, utilizados por el personal técnico en las operaciones de instalación, mantenimiento y reparación de

luminarias públicas. El sistema incorpora un módulo específico para la gestión de estos recursos, permitiendo un seguimiento detallado de su disponibilidad, estado y ubicación, lo que optimiza los procesos de mantenimiento y asegura la reposición oportuna del equipamiento necesario para las operaciones técnicas.

2.2. SISTEMAS WEB

Los sistemas web se refieren a aplicaciones accesibles mediante navegadores a través de Internet o redes internas, permitiendo a los usuarios interactuar con información y herramientas almacenadas en servidores centrales (Shklar & Rosen, 2009).

2.2.1. *Arquitectura de los Sistemas Web*

La arquitectura típica de un sistema web consta de tres capas principales:

1. Capa de presentación (Frontend):
    - Interfaz de usuario desarrollada con HTML5, CSS3 y JavaScript.
    - Frameworks como Bootstrap para una experiencia de usuario dinámica y responsiva.
    - Diseño adaptativo para su uso en diferentes dispositivos y tamaños de pantalla.
2. Capa de lógica de negocio (Backend):
    - Implementada en lenguaje como php (con laravel).
    - Manejo de la lógica de negocio, como cálculos de eficiencia energética, programación de mantenimientos y gestión de inventario.
    - API REST full para la comunicación entre el frontend y el backend.

3. Capa de datos:
    - Base de datos relacional (MySQL) para almacenar información estructurada sobre luminarias, mantenimientos y materiales(Fowler, 2002).

2.2.2. ***Tecnologías y Herramientas***

Para el desarrollo del sistema web de gestión de luminarias públicas, se utilizó diversas tecnologías y herramientas:

1. Frontend:
    - HTML5 para la estructura del contenido
    - CSS3 y frameworks como Bootstrap para el diseño responsivo
    - JavaScript y bibliotecas como jQuery para la interactividad
2. Backend:
    - Lenguajes de programación: PHP
    - Frameworks: Laravel
    - Servidores web: Nginx o Apache
3. Base de datos:
    - Sistemas de gestión de bases de datos relacionales: MySQL
    - Herramientas ORM (Object-Relational Mapping) para facilitar la interacción con la base de datos
4. Herramientas de desarrollo y despliegue:
    - Control de versiones: Git con plataforma como GitHub
    - Editores de texto y entornos de desarrollo (IDE): Visual Studio Code

- Gestores de paquetes: Composer
- gestor de bases de datos: DBeaver
- Servidor local de desarrollo: XAMPP
- Extensión de limpieza de caché: Clear Cache
- Herramienta de prueba de diseño responsivo: Responsive Design Tester

2.3. EFICIENTE

Se refiere a la implementación de mecanismos de gestión que permiten alcanzar los objetivos organizacionales de manera óptima, minimizando el uso de recursos y maximizando los resultados. En el contexto de la gestión de luminarias públicas, materiales y herramientas, el control eficiente implica la capacidad de administrar estos elementos de manera que se reduzcan costos, se optimice el uso de recursos y se mejore la calidad del servicio público (Robbins & Coulter, 2021).

El concepto de control eficiente se basa en varios principios clave:

1. **Optimización de recursos**: Utilizar los materiales, herramientas y personal de la manera más efectiva posible (Slack et al., 2019).
2. **Reducción de desperdicios**: Minimizar el desperdicio de materiales y tiempo en los procesos de mantenimiento y reparación (Womack & Jones, 2016).
3. **Mejora continua**: Implementar un ciclo de evaluación y mejora constante en los procesos de control (Deming, 2018).
4. **Toma de decisiones basada en datos**: Utilizar información precisa y actualizada para guiar las decisiones de gestión (Davenport, 2018).

2.4. **ALUMBRADO PÚBLICO**

Boyce (2014) describe las luminarias de alumbrado público como sistemas integrales de iluminación que no solo distribuyen y transforman la luz emitida por las lámparas, sino que también incorporan componentes estructurales para su soporte, protección y conexión a la red eléctrica. Estos dispositivos cumplen una función dual al actuar como elementos protectores que resguardan las lámparas de factores ambientales externos, mientras gestionan eficientemente la distribución lumínica en espacios públicos.

2.4.1. ***Dimensiones y Espaciamiento***

Boyce (2014) establece los parámetros técnicos fundamentales para la instalación de luminarias públicas, enfatizando la importancia de una distribución uniforme que considere los obstáculos del entorno urbano. Las especificaciones de altura varían según el contexto: para áreas peatonales y ciclovías se recomiendan postes de 4.5 a 6 metros, mientras que para vías urbanas se establecen alturas de 8 a 10 metros en zonas residenciales y comerciales, alcanzando hasta 12 metros en áreas industriales. El espaciamiento óptimo entre postes se calcula como 2.5 a 3 veces su altura, considerando que el cono de luz proyectado tiene un diámetro aproximadamente igual a la altura del poste, lo que permite evitar zonas de sombra y asegurar una iluminación continua y eficiente.

Figura *2.*

Cono de luz

Nota. Iluminación de luminarias públicas. Extraído de: https://pinterest.com

2.4.2. ***Lámparas y Luminarias***

Las instalaciones modernas de alumbrado público, según describe Boyce (2014), emplean principalmente lámparas de descarga, incluyendo las variantes de vapor de mercurio a alta presión y vapor de sodio a diferentes presiones, que han reemplazado a las tradicionales lámparas incandescentes debido a su superior eficiencia energética y económica. Estas lámparas se integran en luminarias específicamente diseñadas que, además de proporcionar protección y soporte, optimizan la distribución del flujo luminoso mediante diseños asimétricos que maximizan la cobertura sobre la calzada. La versatilidad de su instalación permite su montaje en diversos soportes, desde postes y columnas hasta sistemas de suspensión sobre cables transversales, adaptándose a las necesidades específicas de cada espacio urbano.

2.4.3. ***Componentes***

La evolución tecnológica en iluminación pública ha registrado diversos tipos de lámparas, como señala Boyce (2014), desde las tradicionales de vapor de mercurio, ahora obsoletas, hasta las modernas de halogenuros metálicos que ofrecen mayor eficiencia, pasando por las ampliamente utilizadas lámparas de vapor de sodio en sus variantes de alta y baja presión, representando así la progresión histórica en la búsqueda de soluciones más eficientes para el alumbrado urbano.

Figura *3.*

Lámpara

Nota. Lámparas de alumbrado público con inducción electromagnética Extraído de: https://iluminet.com

Balasto: es el elemento estabilizador que contrarresta la tendencia al crecimiento de la intensidad consumida por una lámpara.

Figura *4.*

Balasto

Nota. Balasto para luminaria pública Extraído de: https://electrored.store

Condensador: necesario para corregir el bajo factor de los circuitos formados por las lámparas y el balasto.

Figura *5*.

Condensadores

Nota. Condensador para sistema de iluminación Extraído de: https://lifasa.com

Soportes: desde brazos hasta báculos, columnas, etc. que sirven para mantener la luminaria en la posición deseada. Lo más importante de este soporte son sus dimensiones,

altura y saliente. Deben estar fabricados en materiales resistentes para soportar la corrosión y garantizar su durabilidad.(✓✓ *Luminaria alumbrado público*, s. f., p. 2)

Figura *6.*
Soportes

Nota. Soportes de luminarias LED para alumbrado público Extraído de: https://ei.com.co

2.4.4. ***Tipos de Luminarias***

2.4.4.1. **Luminarias HID**

Se refieren a bombillas de gas a alta presión o a alta intensidad de descarga. Se incluyen las de mercurio y vapor de sodio a alta presión y haluros metálicos (Rhona, s.f.).

Figura *7.*
Luminarias HID

Nota. Luminaria tipo HID para alumbrado vial Extraído de: https://rhona.cl

2.4.4.2. **Luminaria LED**

Basadas en la más reciente tecnología LED (diodo emisor de luz) con mejores resultados en eficiencia y ahorro energético que los emisores de luz convencionales (Dioselyna, s.f.).

Figura *8.*
Luminaria LED

Nota. Luminaria LED para alumbrado público Extraído de: https://dioselyna.com

2.5. **METODOLOGIA**

2.5.1. ***Metodología de Desarrollo UWE***

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! la cual consta de las siguientes fases:

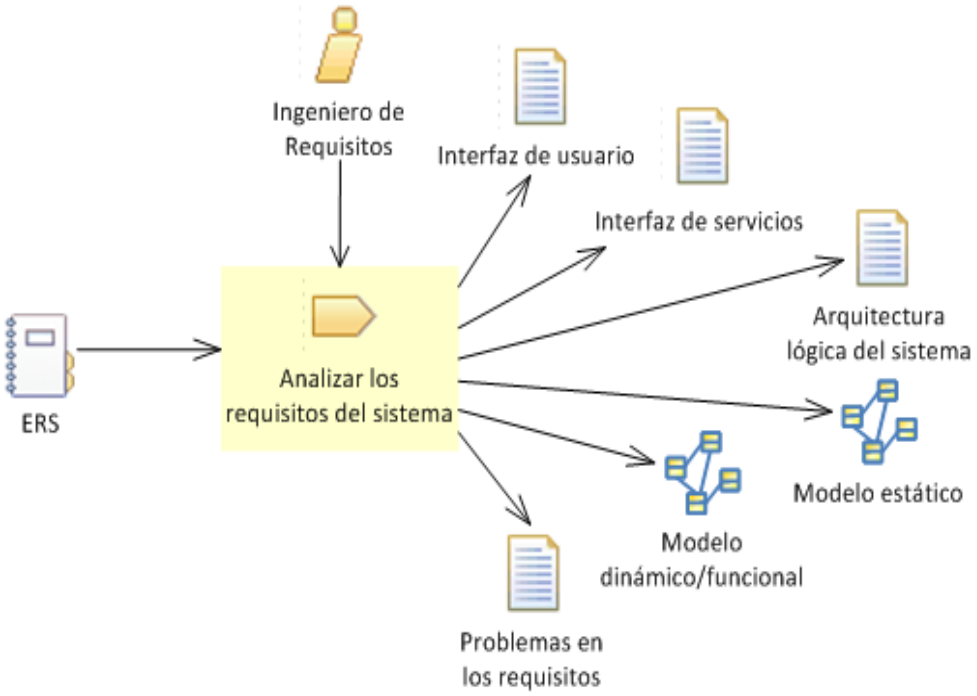
2.5.1.1. **Análisis de Requerimientos**

La fase inicial de análisis, según Rossi et al. (2017), comprende un proceso sistemático de recopilación y evaluación de requisitos del sistema web, donde se identifican tanto las funcionalidades críticas como las restricciones técnicas y organizacionales. Esta etapa emplea

diagramas de casos de uso y modelos UML para documentar de manera estructurada las interacciones previstas entre los usuarios y el sistema, proporcionando así una base sólida para el desarrollo posterior.

Figura *9.*

Análisis de Requerimientos



Nota. Diagrama de análisis de requerimientos Extraído de: https://www.juntadeandalucia.es

2.5.1.2. **Diseño del Sistema**

La fase de diseño arquitectónico, como señalan Rossi et al. (2017), establece la estructura fundamental del sistema basándose en los requisitos previamente identificados. Este proceso implica la creación de diversos diagramas UML, incluyendo diagramas de clases, secuencia y componentes, que detallan la organización de los elementos del sistema, sus relaciones y las interacciones entre los diferentes componentes, proporcionando así una base sólida para la implementación posterior del sistema web.

Figura *10.*

Diseño de Sistema

Nota. Diseño de sistema Extraído de: https://ifgeekthen.nttdata.com

2.5.1.3. **Codificación del Software:**

La fase de codificación, explican Rossi et al. (2017), representa la transformación del diseño arquitectónico en código funcional, donde se seleccionan cuidadosamente las tecnologías y lenguajes de programación que mejor se adapten a los requerimientos del sistema. Durante esta etapa, los desarrolladores traducen las especificaciones del diseño en código ejecutable, asegurando que cada componente implementado cumpla fielmente con los requisitos y la arquitectura establecida en las fases previas.

2.5.1.4. **Pruebas**

Myers et al. (2023) explican que la fase de pruebas constituye un proceso sistemático de verificación y validación, donde se evalúa tanto la funcionalidad como el cumplimiento de los requisitos establecidos. Esta fase abarca diversos niveles de prueba, incluyendo pruebas unitarias, de integración, sistémicas y de aceptación, utilizando UML como herramienta para modelar casos de prueba y asegurar una cobertura completa del sistema.

2.5.1.5. Instalación o Fase de Implementación

En cuanto a la fase de implementación, Pressman y Maxim (2024) señalan que esta etapa implica el despliegue del software en el entorno de producción real. Los autores enfatizan la importancia de una configuración adecuada que garantice la correcta integración de todos los componentes, permitiendo a los usuarios finales acceder y utilizar el sistema de manera eficiente.

2.5.1.6. Mantenimiento

La fase de mantenimiento, que sigue a la implementación, abarca múltiples aspectos críticos para la longevidad del sistema. Como destacan Rossi et al. (2017), esta etapa incluye tres actividades principales: la corrección de errores identificados, el mejoramiento de funcionalidades existentes y la adaptación a nuevos requerimientos. Este proceso puede requerir revisiones sustanciales tanto en el diseño como en el código fuente, con el objetivo primordial de mantener el sistema operativo y relevante a lo largo del tiempo.

2.5.2. *Ciclos de la Metodología UWE*

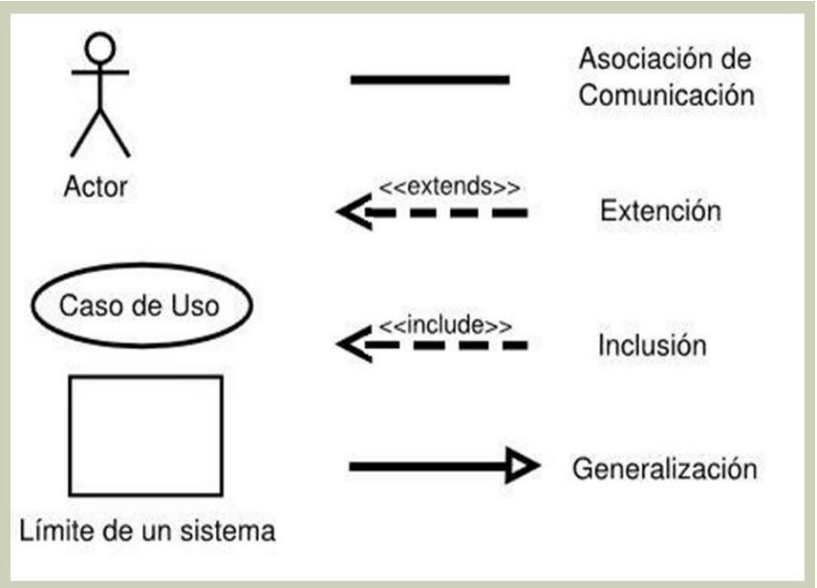
UWE propone una extensión de UML que se divide en 4 pasos

1. **Ciclo de Análisis:** dentro de este ciclo se realiza un análisis del sistema construyendo para ello diagramas de casos de uso diagrama de Caso de uso en UWE se distinguen casos de uso estereotipados con “browsing” y con “processing” para ilustrar si los datos persistentes de la aplicación son modificación o no. Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participaran en un caso de uso se denominan actores.

UWE provee diferentes estereotipos como se ve en la Figura 11.

Figura *11.*

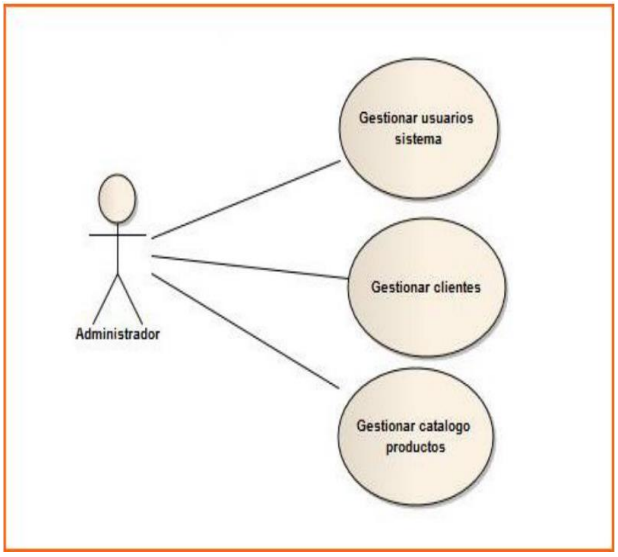
Estereotipos del Diagrama de Caso de Uso



Nota. Estereotipo del diagrama de caso de uso Extraído de: https://images.slideplayer.es

Figura *12.*

Diagrama de Caso de Uso en la Metodología UWE



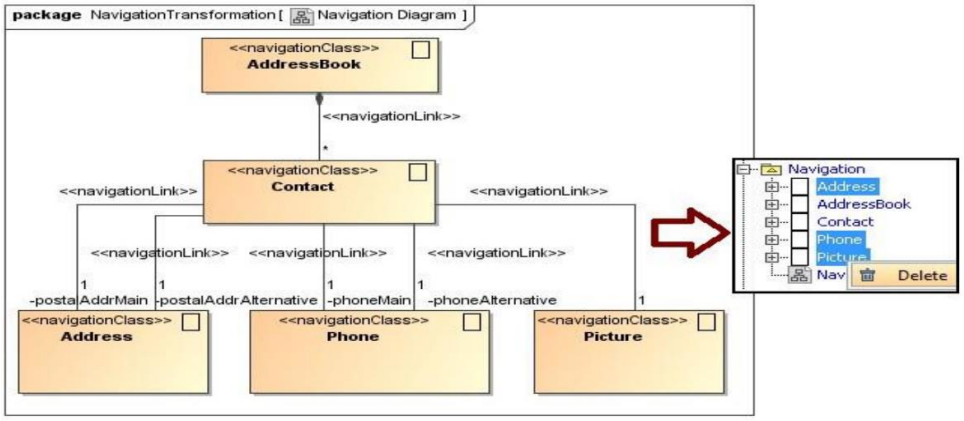
Nota. Diagrama de caso de uso Extraído de: (Requirements Model, 2016)

2. Ciclo de Diseño Conceptual: dentro de este ciclo se modela al universo de la aplicación, creando para ello el modelo de dominio.

3. Modelo conceptual Un diagrama de contenido es un diagrama de UML normal de clases. Los diagramas de clases describen la estructura estática de un sistema. UWE provee diferentes estereotipos:

Figura *13.*

Diagrama de Contenido en la Metodología UWE



Nota. Contenido de la metodología UWE Extraído de: https://encrypted-tbn0.gstatic.com

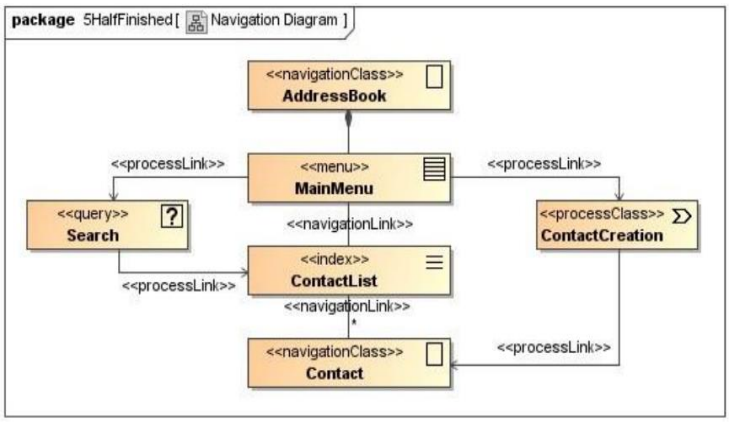
4. Ciclo de Diseño Navegaciones: Rubin (2012) describe que durante este ciclo se establece la interconexión entre los diferentes objetos del dominio, proceso que se materializa mediante la elaboración de dos modelos esenciales: el Espacio de navegación y la Estructura de navegación, los cuales definen el flujo de interacción del sistema.

➢ **Modelo de Navegación**

En un sistema para la web es útil saber cómo están enlazadas las páginas. Ello significa que necesitamos un diagrama contenido nodos (nodes) y enlaces (links). UWE provee diferentes estereotipos:

Figura *14.*

Diagrama de Navegación en la Metodología UWE



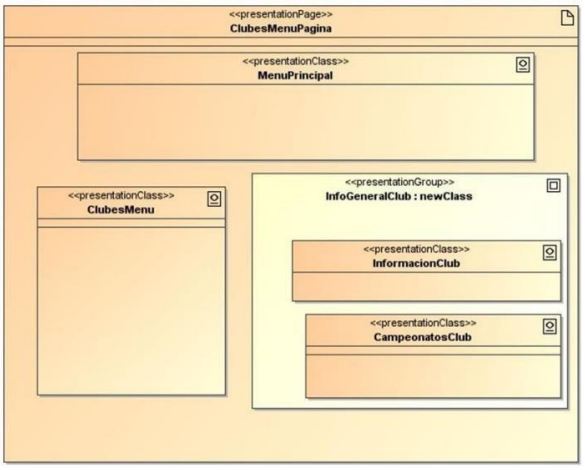
Nota. Representación del diagrama de navegación UWE Extraído de: (Tutorial-navegation Model, 2016)

5. Ciclo de Diseño de la Presentación: la presentación se describe en función de distintos modelos estándares UML. ➢ Modelo de Presentación

El modelo de navegación no indica cuales son las clases de navegación y de proceso que pertenecen a una página web. Podemos usar un Diagrama de Presentación con el fin de proveer esta información. UWE provee diferentes estereotipos:

Figura *15.*

Diagrama de Presentación en la Metodología UWE



Nota. Diagrama metodología UWE Extraído de: (Tutorial-navegation Model, 2016)

2.6. FRAMEWORKS Y TECNOLOGIAS WEB

2.6.1. *PHP*

PHP (acrónimo de "PHP: Hypertext Preprocessor") es un lenguaje de programación de código abierto diseñado específicamente para el desarrollo web del lado del servidor. Creado originalmente por Rasmus Lerdorf en 1994, PHP ha evolucionado significativamente a lo largo de los años, convirtiéndose en uno de los lenguajes más populares y versátiles para la creación de sitios web dinámicos y aplicaciones web. Su sintaxis se basa en C, Java y Perl, lo que facilita su aprendizaje para desarrolladores familiarizados con estos lenguajes. PHP se caracteriza por su capacidad para integrarse con HTML, permitiendo a los desarrolladores incrustar código PHP directamente en el marcado HTML. Esta característica, junto con su robusta funcionalidad y amplio soporte para bases de datos, hace de PHP una elección ideal para el desarrollo web moderno (Tatroe et al., 2020).

Profundizando en las características y capacidades de PHP:

- Flexibilidad y versatilidad: PHP es conocido por su flexibilidad. Puede utilizarse para desarrollar desde pequeños sitios web estáticos hasta complejas aplicaciones empresariales. Su capacidad para funcionar en diversos sistemas operativos (Windows, Linux, macOS, etc.) y su compatibilidad con la mayoría de los servidores web lo convierten en una opción muy versátil.
- Manejo de bases de datos: Una de las fortalezas de PHP es su excelente soporte para una amplia gama de bases de datos. Ofrece conexiones nativas para muchos sistemas de bases de datos, incluyendo MySQL entre otros. Esto facilita la creación de aplicaciones web que requieren almacenamiento y recuperación de datos de manera eficiente.

- Seguridad: PHP proporciona varias capas de seguridad para proteger contra amenazas comunes en aplicaciones web. Incluye funciones integradas para prevenir inyecciones SQL, cross-site scripting (XSS), y otras vulnerabilidades. Además, su constante actualización por parte de la comunidad ayuda a mantener el lenguaje seguro frente a nuevas amenazas.
- Rendimiento y escalabilidad: Con las mejoras introducidas en versiones recientes, PHP ha experimentado significativos aumentos en rendimiento. La introducción de PHP 7 y versiones posteriores ha traído consigo optimizaciones que han duplicado la velocidad de ejecución en comparación con versiones anteriores. Esto, combinado con tecnologías como opcode caching, permite que las aplicaciones PHP escalen eficientemente para manejar altos volúmenes de tráfico.
- Frameworks y CMS: PHP cuenta con una amplia variedad de frameworks y sistemas de gestión de contenidos (CMS) que simplifican y aceleran el proceso de desarrollo. Frameworks populares como Laravel, Symfony, y CodeIgniter proporcionan estructuras robustas para el desarrollo de aplicaciones, mientras que CMS como WordPress, Drupal, y Joomla permiten crear sitios web complejos con facilidad.
- Comunidad y ecosistema: PHP tiene una de las comunidades de desarrolladores más grandes y activas. Esto se traduce en una abundancia de recursos, bibliotecas de terceros, y herramientas disponibles. El gestor de paquetes Composer ha revolucionado la forma en que los desarrolladores PHP gestionan las dependencias de sus proyectos, facilitando la integración de bibliotecas externas y promoviendo la reutilización de código.
- Evolución continua: A pesar de ser un lenguaje maduro, PHP continúa evolucionando. Las versiones recientes han introducido características modernas

como tipos de retorno estrictos, clases anónimas, y mejoras en el manejo de errores. Esto demuestra el compromiso de la comunidad PHP para mantener el lenguaje relevante y competitivo en el panorama del desarrollo web actual.

2.6.2. ***Framework Laravel v10***

Laravel 10 es la décima versión principal del popular framework de PHP, lanzada en febrero de 2023. Este framework de código abierto sigue el patrón de arquitectura Modelo-Vista-Controlador (MVC) y está diseñado para facilitar el desarrollo de aplicaciones web robustas, escalables y de alto rendimiento. Laravel se destaca por su sintaxis elegante, su amplia gama de funcionalidades integradas y su enfoque en la productividad del desarrollador.

Características principales de Laravel 10:

- Arquitectura modular: Laravel 10 mantiene una estructura modular que permite a los desarrolladores organizar el código de manera eficiente. Utiliza el sistema de paquetes de Composer, lo que facilita la integración de bibliotecas de terceros y la reutilización de código.
- Eloquent ORM: El Object-Relational Mapping (ORM) de Laravel, llamado Eloquent, proporciona una forma intuitiva y expresiva de interactuar con bases de datos. Eloquent permite definir relaciones complejas entre modelos y realizar consultas de base de datos de manera sencilla y eficiente.
- Sistema de enrutamiento: Laravel 10 ofrece un sistema de enrutamiento potente y flexible que permite definir rutas de aplicación de manera clara y concisa. Soporta rutas REST ful, agrupación de rutas, y middleware para un control granular del flujo de la aplicación.
- Blade Template Engine: Blade es el motor de plantillas de Laravel, que proporciona una sintaxis sencilla y expresiva para crear vistas. Permite la

reutilización de código mediante layouts y componentes, mejorando la mantenibilidad de la interfaz de usuario.

- Artisan CLI: La interfaz de línea de comandos Artisan de Laravel facilita tareas comunes de desarrollo, como la generación de código, la gestión de migraciones de base de datos y la ejecución de tareas programadas.
- Seguridad integrada: Laravel 10 incluye características de seguridad robustas como protección CSRF, encriptación, hashing de contraseñas y autenticación de usuarios, lo que ayuda a proteger las aplicaciones contra vulnerabilidades comunes.
- Gestión de colas y trabajos en segundo plano: El framework proporciona un sistema de colas unificado que permite ejecutar tareas en segundo plano de manera eficiente, mejorando el rendimiento y la capacidad de respuesta de la aplicación.
- Testing: Laravel facilita la escritura y ejecución de pruebas unitarias y de integración, promoviendo el desarrollo basado en pruebas (TDD) y asegurando la calidad del código.
- Ecosystem: Laravel 10 se beneficia de un amplio ecosistema que incluye paquetes como Laravel Sanctum para autenticación API, Laravel Breeze para scaffolding de autenticación, para aplicaciones con funciones avanzadas.
- Rendimiento mejorado: Esta versión incluye optimizaciones de rendimiento, como la carga diferida de servicios y mejoras en el enrutamiento, que contribuyen a una mayor velocidad de ejecución.
- Compatibilidad con PHP 8.1+: Laravel 10 requiere PHP 8.1 o superior, aprovechando las características más recientes del lenguaje PHP para mejorar la calidad y el rendimiento del código.

- Actualizaciones y mantenimiento: Laravel sigue un ciclo de lanzamiento predecible, con actualizaciones frecuentes y un fuerte énfasis en la compatibilidad hacia atrás, facilitando la actualización y el mantenimiento de proyectos existentes(Stauffer, 2019).

El framework Laravel en su versión 10 representa una solución integral para el desarrollo web moderno, destacándose por su capacidad de equilibrar una interfaz intuitiva con capacidades técnicas avanzadas. La combinación de una arquitectura sólida, características de seguridad robustas y un enfoque centrado en optimizar la productividad del desarrollador hacen de este framework una opción versátil y confiable para proyectos de cualquier escala (Tatroe et al., 2020).

2.6.3. ***MySQL***

Como señalan Wilson y Johnson (2021), MySQL ha evolucionado desde su desarrollo inicial por MySQL AB hasta convertirse en uno de los sistemas de gestión de bases de datos relacionales (RDBMS) más influyentes del mundo bajo la propiedad de Oracle Corporation. Su arquitectura versátil y rendimiento optimizado lo han posicionado como una solución preferida para el desarrollo web, abarcando desde pequeñas aplicaciones hasta sistemas empresariales de gran escala.

Las características fundamentales de MySQL se estructuran en varios componentes clave. Su arquitectura cliente-servidor facilita la gestión centralizada de datos, mientras que su amplio soporte para SQL estándar permite operaciones de base de datos intuitivas. La naturaleza multiplataforma del sistema ofrece flexibilidad en la implementación, adaptándose a diversos entornos operativos como Windows, Linux y macOS (Schwartz et al., 2012).

Según Tatroe et al. (2020), la robustez de MySQL se evidencia en sus capacidades avanzadas de seguridad y rendimiento. El sistema implementa un esquema de privilegios

sofisticado con encriptación de contraseñas, soporte para transacciones ACID, y una arquitectura de almacenamiento flexible que permite elegir entre diferentes motores como InnoDB y MyISAM según las necesidades específicas del proyecto.

Las herramientas administrativas de MySQL, incluyendo MySQL Workbench, proporcionan una interfaz gráfica completa para el diseño y gestión de bases de datos. El sistema también se destaca por su optimizador de consultas avanzado y sus capacidades de backup y recuperación, incluyendo herramientas como mysqldump para la gestión de datos. La incorporación del soporte nativo para JSON en versiones recientes amplía su versatilidad para aplicaciones modernas (Welling & Thomson, 2017).

2.6.4. ***HTML, CSS y JavaScript en el Desarrollo Web***

Duckett (2014) establece que el desarrollo web moderno se fundamenta en tres tecnologías principales: HTML, CSS y JavaScript, las cuales trabajan en conjunto para crear experiencias web completas y funcionales. Estas tecnologías complementarias permiten desarrollar desde sitios web básicos hasta aplicaciones web sofisticadas.

HTML, como lenguaje de marcado fundamental, proporciona la estructura base de las páginas web. Su diseño intuitivo facilita la organización del contenido mediante elementos semánticos, desde etiquetas básicas como <body> y <p> hasta componentes más avanzados introducidos en HTML5 como <header> y <nav>, garantizando compatibilidad universal con los navegadores web (Gauchat, 2017).

Según Wilson (2021), CSS revoluciona la presentación visual de las páginas web al separar el diseño del contenido. El autor destaca que esta tecnología no solo permite un control preciso sobre la apariencia mediante selectores y propiedades, sino que también facilita la creación de diseños responsivos a través de características como Flexbox, Grid y media queries, mejorando significativamente la mantenibilidad del código.

JavaScript amplía las capacidades interactivas de las páginas web como lenguaje de programación orientado a objetos del lado del cliente. Sus funcionalidades incluyen la manipulación dinámica del DOM, gestión de eventos de usuario, validación de formularios y comunicación asíncrona con servidores. La versatilidad de JavaScript se extiende al desarrollo del lado del servidor mediante tecnologías como Node.js, permitiendo crear aplicaciones web completas y dinámicas (Zakas, 2019).

Como señala Perkovic (2021), la sinergia entre estas tres tecnologías fundamentales establece los cimientos del desarrollo web moderno. HTML proporciona la estructura semántica, CSS gestiona la presentación visual y la responsividad, mientras que JavaScript aporta la interactividad y funcionalidad dinámica. Esta combinación permite crear desde sitios web estáticos hasta sofisticadas aplicaciones web de página única (SPA).

2.6.5. ***Bootstrap 5***

Otto y Thornton (2022) presentan Bootstrap 5 como una evolución significativa en el desarrollo web frontend, destacando cómo esta versión, lanzada en 2021, revoluciona la creación de interfaces responsivas modernas. Los autores enfatizan que esta iteración marca un punto de inflexión al eliminar la dependencia de jQuery, resultando en un framework más ligero y eficiente.

Las innovaciones fundamentales de Bootstrap 5 se manifiestan a través de múltiples características clave. Su sistema de rejilla flexible, basado en Flexbox, facilita la creación de diseños adaptables mediante una estructura de 12 columnas con seis puntos de quiebre predefinidos. La colección de componentes predefinidos, que incluye elementos como botones, formularios y modales, optimiza significativamente el proceso de desarrollo (Wilson, 2021).

Según Duckett (2014), la arquitectura de Bootstrap 5 prioriza la accesibilidad y la personalización. El framework incorpora variables Sass para modificaciones globales del diseño, mientras mantiene una extensa biblioteca de utilidades CSS que permiten ajustes rápidos sin código adicional. El autor resalta la importancia de las mejoras en accesibilidad, que facilitan la integración con tecnologías de asistencia.

El framework introduce características técnicas avanzadas que mejoran la experiencia de desarrollo. La integración de Popper.js optimiza el posicionamiento de elementos interactivos, mientras que el soporte nativo para RTL facilita la internacionalización. La documentación exhaustiva y los ejemplos prácticos proporcionados simplifican el proceso de aprendizaje e implementación (Smyth, 2020).

Johnson (2022) enfatiza que Bootstrap 5 se ha convertido en una herramienta indispensable para el desarrollo web moderno, permitiendo crear interfaces consistentes y responsivas de manera eficiente. Sin embargo, el autor advierte sobre la importancia de personalizar el framework según las necesidades específicas del proyecto para evitar la homogeneización del diseño web.

2.6.6. ***XAMPP***

Dvorski (2007) introduce XAMPP como una herramienta fundamental para el desarrollo web local, destacando su capacidad para simular entornos de servidor completos sin necesidad de conexión a internet. Esta característica la hace especialmente valiosa para desarrolladores que se inician en la creación de aplicaciones web dinámicas con PHP, MySQL y Perl.

Las características distintivas de XAMPP incluyen un proceso de instalación simplificado, compatibilidad multiplataforma y un conjunto integrado de herramientas esenciales. Este paquete todo en uno incorpora el servidor Apache, bases de datos

MariaDB/MySQL, intérpretes de lenguajes de programación y utilidades adicionales como phpMyAdmin para la gestión de bases de datos (Sikos, 2020).

En cuanto a su importancia en el desarrollo web, Welling y Thomson (2017) enfatizan que XAMPP proporciona un entorno de desarrollo local que refleja fielmente las configuraciones de producción. Los autores resaltan que esta capacidad permite a los desarrolladores realizar pruebas exhaustivas en un entorno controlado antes de la implementación final.

Los componentes principales de XAMPP funcionan en perfecta sinergia. Apache actúa como el servidor web central, MariaDB ofrece una solución de base de datos robusta y de código abierto, mientras que el intérprete PHP facilita la ejecución de scripts del lado del servidor. Esta combinación crea un entorno de desarrollo completo y versátil (Bartholomew, 2019).

2.6.7. ***Plugins***

Erixon (2018) define los plugins como componentes de software que amplían las capacidades de programas existentes. En el ámbito del desarrollo web, estos módulos de código resultan fundamentales para extender la funcionalidad de diversas plataformas, desde sistemas de gestión de contenidos hasta navegadores web y servidores.

Los plugins se manifiestan en diferentes contextos del desarrollo web. En sistemas de gestión de contenidos como WordPress y Drupal, facilitan la incorporación de nuevas funcionalidades sin alterar el código central. Los navegadores web emplean plugins para mejorar la experiencia del usuario, mientras que los servidores web los utilizan para expandir sus capacidades técnicas (Williams et al., 2022).

Según Tatroe et al. (2020), en el contexto específico de XAMPP, los plugins desempeñan un papel crucial en la extensión de funcionalidades. Los autores destacan cómo

Apache utiliza módulos como mod_ssl para HTTPS y mod_security para protección adicional, mientras que PHP se beneficia de extensiones como PDO para gestión de bases de datos y GD para manipulación de imágenes.

Las ventajas del uso de plugins son múltiples: proporcionan modularidad en el desarrollo, permiten personalización según necesidades específicas, facilitan el mantenimiento del código y fomentan la colaboración comunitaria. Sin embargo, también presentan desafíos significativos relacionados con la seguridad, el rendimiento y la compatibilidad entre versiones (Robbins, 2021).

Stuttard y Pinto (2021) enfatizan la importancia de implementar mejores prácticas en el uso de plugins. Los autores recomiendan seleccionar plugins de fuentes confiables, mantener actualizaciones regulares, realizar copias de seguridad antes de las instalaciones y evaluar exhaustivamente la compatibilidad antes de la implementación en producción. Estas precauciones ayudan a mitigar los riesgos asociados con la integración de componentes de terceros.

2.6.8. ***Visual Studio Code***

Microsoft (2023) presenta Visual Studio Code como un editor de código fuente revolucionario que, desde su lanzamiento en 2015, ha transformado el panorama del desarrollo web. Su éxito se atribuye a una combinación de rendimiento excepcional, capacidad de extensión y flexibilidad en la personalización.

Las características fundamentales de VS Code incluyen una interfaz de usuario intuitiva y personalizable, junto con un robusto soporte multilingüe. El editor destaca por su sistema integrado de depuración y control de versiones, permitiendo a los desarrolladores gestionar eficientemente proyectos complejos (Perkovic, 2021).

Smyth (2020) enfatiza que el ecosistema de extensiones de VS Code representa uno de sus aspectos más potentes. El autor destaca cómo estas extensiones permiten a los desarrolladores adaptar el editor a sus necesidades específicas, abarcando desde herramientas de control de versiones hasta utilidades de optimización web.

En el contexto del desarrollo web, VS Code sobresale por su integración con tecnologías modernas. El editor proporciona soporte nativo para JavaScript y TypeScript, mientras que su sistema de extensiones facilita el trabajo con frameworks populares como React y Angular. Para el desarrollo full-stack, ofrece herramientas que simplifican tanto el trabajo frontend como backend (Wanyoike, 2020).

Johnson (2022) describe cómo VS Code optimiza el flujo de trabajo de desarrollo web a través de características como Emmet para la escritura rápida de HTML/CSS, Live Server para visualización en tiempo real, y ESLint para análisis de código. El autor señala que estas herramientas integradas reducen significativamente el tiempo dedicado a tareas rutinarias.

La integración entre VS Code y XAMPP crea un entorno de desarrollo web completo. Este sistema combinado permite a los desarrolladores gestionar servidores locales, bases de datos y código fuente desde una única interfaz. La configuración para desarrollo PHP incluye extensiones específicas y herramientas de depuración que facilitan el trabajo con aplicaciones web dinámicas (Wilson, 2021).

Según Perkovic (2021), las mejores prácticas para utilizar VS Code incluyen una gestión cuidadosa de extensiones, personalización gradual del entorno y dominio de los atajos de teclado. El autor enfatiza la importancia de mantener un equilibrio entre funcionalidad y rendimiento, recomendando instalar solo las extensiones esenciales para el flujo de trabajo específico.

2.7. PRUEBAS DE SOFTWARE

Las pruebas de software son una parte crucial del ciclo de desarrollo, especialmente en sistemas web complejos como el de gestión de luminarias públicas. Estas pruebas aseguran la calidad, funcionalidad y rendimiento del sistema antes de su implementación Myers et al. (2023).

2.7.1. *Tipos de Pruebas*

2.7.1.1. Pruebas Funcionales

Las pruebas funcionales se centran en las entradas y salidas del sistema, sin considerar su funcionamiento interno. Myers et al. (2023) explican:

"Las pruebas funcionales evalúan la funcionalidad del software sin conocer su estructura interna. Este enfoque es particularmente útil para validar que el sistema cumple con los requisitos especificados y funciona correctamente desde la perspectiva del usuario final" (p. 112).

En el contexto de un sistema de gestión de luminarias públicas, las pruebas funcionales podrían incluir:

- Verificar que el sistema registre correctamente los datos de nuevas luminarias.
- Comprobar que los informes generados contengan la información correcta.
- Asegurar que las interfaces de usuario respondan adecuadamente a diferentes entradas.

2.7.1.2. Pruebas Estructurales

Las pruebas estructurales, por otro lado, examinan la estructura interna del código. Según Spillner et al. (2022):

"Las pruebas estructurales implican el análisis del código fuente y la estructura interna del software. Estas pruebas son esenciales para identificar errores en la lógica del programa, optimizar el rendimiento y mejorar la calidad general del código" (p. 178).

Para un sistema web de gestión de luminarias, las pruebas estructurales podrían incluir:

- Verificar la eficiencia de los algoritmos de búsqueda y filtrado.
- Analizar la estructura de las consultas a la base de datos para optimizar el rendimiento.
- Examinar el manejo de excepciones y errores en el código.

2.7.2. ***Importancia de las Pruebas en el Desarrollo de Sistemas Web***

Las pruebas son fundamentales en el desarrollo de sistemas web por varias razones. Pressman y Maxim (2024) señalan:

"Las pruebas rigurosas en sistemas web son esenciales para garantizar la funcionalidad, seguridad y rendimiento en diversos navegadores y dispositivos. En un entorno web, donde los usuarios pueden acceder al sistema desde una variedad de plataformas, las pruebas exhaustivas son cruciales para asegurar una experiencia consistente y fiable" (p. 389).

La importancia de las pruebas en el desarrollo de sistemas web se manifiesta en varios aspectos:

1. Calidad del producto: Las pruebas ayudan a identificar y corregir defectos antes del lanzamiento, mejorando la calidad general del sistema.
2. Seguridad: En sistemas que manejan datos sensibles, como información de infraestructura pública, las pruebas de seguridad son críticas para prevenir vulnerabilidades.

3. Rendimiento: Las pruebas de carga y estrés aseguran que el sistema pueda manejar el volumen de tráfico esperado sin degradación del rendimiento.
4. Compatibilidad: Las pruebas en diferentes navegadores y dispositivos garantizan una experiencia de usuario consistente.
5. Usabilidad: Las pruebas de interfaz de usuario ayudan a mejorar la facilidad de uso y la accesibilidad del sistema.
6. Confiabilidad: Las pruebas exhaustivas aumentan la confiabilidad del sistema, reduciendo el riesgo de fallos en producción.

En el contexto de un sistema de gestión de luminarias públicas, las pruebas son particularmente importantes debido a la naturaleza crítica de la infraestructura que gestiona. Un sistema bien probado puede prevenir errores que podrían resultar en fallos en el alumbrado público, afectando la seguridad y el bienestar de los ciudadanos.

2.8. SEGURIDAD EN SISTEMAS WEB

2.8.1. *Estándar ISO 27001*

La ISO 27001 es un estándar internacional que proporciona un marco robusto para la gestión de la seguridad de la información. Como señala la International Organization for Standardization (2022):

"ISO 27001 ayuda a las organizaciones a proteger sus activos de información de manera sistemática y rentable, mediante la adopción de un Sistema de Gestión de Seguridad de la Información (SGSI). Este enfoque permite a las organizaciones identificar, gestionar y minimizar la gama de amenazas a las que la información está regularmente sometida" (p. 4).

Calder y Watkins (2023) profundizan en la estructura y beneficios de ISO 27001:

"El estándar se basa en el ciclo de mejora continua Plan-Do-Check-Act (PDCA), que asegura que la seguridad de la información no sea un proyecto único, sino un proceso en constante evolución. Esto permite a las organizaciones adaptarse a las amenazas emergentes y a los cambios en el panorama tecnológico" (p. 27).

Los componentes clave de ISO 27001 incluyen:

1. Contexto de la organización: Comprender las necesidades y expectativas de las partes interesadas.
2. Liderazgo: Compromiso de la alta dirección con la seguridad de la información.
3. Planificación: Evaluación de riesgos y oportunidades.
4. Soporte: Recursos, competencia, concienciación, comunicación y documentación.
5. Operación: Implementación de controles y procesos de seguridad.
6. Evaluación del desempeño: Monitoreo, medición, análisis y evaluación.
7. Mejora: Acciones correctivas y mejora continua.

Humphreys (2024) destaca la importancia de ISO 27001 en el contexto de sistemas web:

"Para sistemas web que manejan datos sensibles, como información de alumbrado público municipal, ISO 27001 proporciona un marco crucial para proteger contra amenazas cibernéticas. Implementar controles como la gestión de accesos, el cifrado de datos y la continuidad del negocio puede prevenir brechas de seguridad y asegurar la integridad de los servicios públicos" (p. 113).

La aplicación de ISO 27001 en el desarrollo y mantenimiento de sistemas web municipales puede incluir:

- Implementación de políticas de contraseñas robustas

- Cifrado de datos sensibles en tránsito y en reposo
- Realización regular de pruebas de penetración y evaluaciones de vulnerabilidad
- Establecimiento de procedimientos de respuesta a incidentes
- Formación continua del personal en mejores prácticas de seguridad

Al adherirse a los principios de ISO 27001, los sistemas web municipales pueden mejorar significativamente su postura de seguridad, protegiendo tanto los datos de los ciudadanos como la infraestructura crítica de la ciudad.

2.8.2. *Importancia de la Seguridad en Sistemas de Gestión Municipal*

La seguridad es crucial en sistemas que manejan información sensible de ciudadanos. Según García-Hernández (2023), "Los gobiernos municipales deben priorizar la seguridad de la información para mantener la confianza pública y proteger contra amenazas cibernéticas cada vez más sofisticadas" (p. 89).

2.9. CICLO DE VIDA DEL SISTEMA

El ciclo de vida del sistema es un enfoque estructurado para el desarrollo y mantenimiento de sistemas de información, que abarca todas las etapas desde la concepción inicial hasta la implementación y eventual retiro del sistema (Pressman & Maxim, 2020).

En el contexto del sistema web para el registro, seguimiento y control eficiente de luminarias públicas, materiales y herramientas, el ciclo de vida del sistema es crucial para garantizar un desarrollo ordenado y eficaz.

Fases del ciclo de vida del sistema:

1. Planificación y análisis de requisitos: En esta fase inicial, se identifican las necesidades específicas del sistema de gestión de luminarias públicas. Se realizan entrevistas con

los stakeholders, como funcionarios municipales y técnicos de mantenimiento, para comprender los procesos actuales y las mejoras deseadas (Sommerville, 2016). Los requisitos clave incluyen:

- Registro detallado de luminarias públicas
- Seguimiento del estado y mantenimiento de las luminarias
- Control de inventario de materiales y herramientas
- Generación de informes y análisis de datos

2. Diseño del sistema: Basándose en los requisitos recopilados, se procede al diseño del sistema. Esta fase implica:
    - Diseño de la arquitectura del sistema web
    - Modelado de la base de datos para almacenar información sobre luminarias, materiales y herramientas
    - Diseño de la interfaz de usuario, considerando la usabilidad para diferentes tipos de usuarios
    - Definición de los flujos de trabajo y procesos del sistema
3. Implementación: En esta fase, se lleva a cabo la codificación del sistema web. Se utilizan lenguajes de programación y frameworks adecuados para el desarrollo web, como Python con Django o JavaScript con Node.js para el backend, y HTML, CSS y JavaScript para el frontend (Gauchat, 2017). La implementación incluye:
    - Desarrollo de módulos para registro y gestión de luminarias
    - Implementación de funcionalidades de seguimiento y control
    - Creación de interfaces para la gestión de materiales y herramientas
    - Integración de sistemas de autenticación y autorización
4. Pruebas: Las pruebas son cruciales para garantizar la calidad y fiabilidad del sistema. Se realizan diversos tipos de pruebas (Myers et al., 2021):
    - Pruebas unitarias para verificar el funcionamiento de componentes individuales

- Pruebas de integración para asegurar la correcta interacción entre módulos
- Pruebas de sistema para evaluar el rendimiento y la funcionalidad global
- Pruebas de aceptación con usuarios finales para validar la usabilidad y satisfacción

5. Despliegue: Una vez que el sistema ha pasado todas las pruebas, se procede al despliegue. Esta fase implica:
    - Configuración del entorno de producción
    - Migración de datos existentes al nuevo sistema
    - Capacitación de usuarios finales
    - Implementación gradual del sistema, posiblemente comenzando con un proyecto piloto
6. Mantenimiento y evolución: Después del despliegue, el sistema entra en la fase de mantenimiento y evolución. Esta fase es continua y puede incluir:
    - Corrección de errores identificados durante el uso
    - Actualizaciones de seguridad
    - Mejoras y adición de nuevas funcionalidades basadas en el feedback de los usuarios
    - Optimización del rendimiento del sistema
7. Evaluación y revisión: Periódicamente, se realiza una evaluación completa del sistema para determinar su efectividad y relevancia continua. Esto puede llevar a:
    - Identificación de áreas de mejora
    - Planificación de actualizaciones importantes
    - Decisiones sobre la continuidad o eventual reemplazo del sistema

El ciclo de vida del sistema no es un proceso lineal, sino iterativo. A medida que se identifican nuevas necesidades o cambios en los requisitos, el sistema puede volver a fases anteriores para adaptarse y mejorar (Boehm & Turner, 2003).

La adopción de un enfoque ágil en el ciclo de vida del sistema permite una mayor flexibilidad y capacidad de respuesta a los cambios. Metodologías como Scrum o Kanban pueden ser particularmente beneficiosas para el desarrollo y mantenimiento continuo de este sistema web de gestión de luminarias públicas (Rubin, 2012).

En conclusión, el ciclo de vida del sistema proporciona un marco estructurado para el desarrollo, implementación y mantenimiento del sistema web para el registro, seguimiento y control eficiente de luminarias públicas, materiales y herramientas. Este enfoque asegura que el sistema cumpla con los requisitos iniciales, se adapte a las necesidades cambiantes y mantenga su eficacia a lo largo del tiempo (Myers et al., 2021).

2.10. **METRICAS DE CALIDAD DE SOFTWARE**

2.10.1. ***ISO/IEC 25000***

Según García-Mireles y Villa-Martínez (2023), la norma ISO/IEC 25000, también conocida como SQuaRE (System and Software Quality Requirements and Evaluation), constituye un marco de trabajo integral diseñado para evaluar la calidad del software. Esta normativa unifica y actualiza los estándares anteriores, proporcionando una estructura coherente para la especificación de requisitos y evaluación de calidad en productos de software.

La implementación de la familia de normas ISO/IEC 25000 ha demostrado ser fundamental para las organizaciones que buscan garantizar la calidad de sus productos software (Djouab y Bari, 2020). Este conjunto de normas se divide en varias divisiones, cada una enfocada en aspectos específicos del ciclo de vida del software.

2.10.1.1. ISO/IEC 25010: Modelo de Calidad del Producto Software

La ISO/IEC 25010, como componente esencial de la familia SQuaRE, establece un modelo específico para evaluar la calidad del producto software. Estdale y Georgiadou (2018) señalan que este modelo define ocho características fundamentales de calidad:

1. **Adecuación funcional**: Capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas cuando se utiliza bajo condiciones específicas.
2. **Eficiencia de rendimiento**: Representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones (Llaneza et al., 2023).
3. **Compatibilidad**: De acuerdo con la International Organization for Standardization (2019), esta característica mide la capacidad de dos o más sistemas para intercambiar información y/o llevar a cabo sus funciones requeridas mientras comparten el mismo entorno.
4. **Usabilidad**: Determina la capacidad del producto software para ser entendido, aprendido, usado y resultar atractivo para el usuario.
5. **Fiabilidad**: Como señalan Djouab y Bari (2020), esta característica representa la capacidad del sistema para desempeñar las funciones especificadas durante un período definido.
6. **Seguridad**: Capacidad de protección de la información y los datos para que personas o sistemas no autorizados no puedan leerlos o modificarlos.
7. **Mantenibilidad**: García-Mireles y Villa-Martínez (2023) la definen como el grado de efectividad y eficiencia con el que un producto puede ser modificado.

8. **Portabilidad**: Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno operativo a otro.

Cada una de estas características incluye subcaracterísticas que permiten una evaluación más detallada y específica. Llaneza et al. (2023) destacan que estas subcaracterísticas proporcionan un marco de referencia para realizar mediciones directas de la calidad del software, permitiendo una evaluación objetiva y sistemática del producto.

2.11. ESTIMACIÓN DE COSTOS EN PROYECTOS DE SOFTWARE

La estimación de costos es un proceso crucial en el desarrollo de software que permite prever los recursos financieros, humanos y de tiempo necesarios para completar un proyecto (Boehm et al., 2000). En el contexto del sistema web para el registro, seguimiento y control eficiente de luminarias públicas, materiales y herramientas, una estimación precisa es fundamental para la planificación y gestión eficaz del proyecto.

2.11.1. *COCOMO II*

Es una versión actualizada del modelo COCOMO original, diseñada para estimar proyectos de software más modernos (Boehm et al., 2000). Este modelo considera factores adicionales y es más flexible para diferentes tipos de desarrollo de software.

Para el sistema web de luminarias públicas, la aplicación de COCOMO II incluiría:

1. Selección del modelo: Se utilizaría el modelo Post-Architecture, ya que se tiene una comprensión clara de los requisitos y la arquitectura del sistema.
2. Estimación del tamaño: En lugar de KLOC, COCOMO II permite el uso de Puntos de Función (que se describirán en la sección 2.18.3) convertidos a Líneas de Código Fuente (SLOC).
3. Evaluación de factores de escala:

- Precedentes: Bastante familiar (3)
- Flexibilidad de desarrollo: Normal (3)
- Arquitectura/Resolución de riesgos: Algo (4)
- Cohesión del equipo: Mayormente (2)
- Madurez del proceso: Nominal (3)

4. Evaluación de multiplicadores de esfuerzo: Se evalúan 17 multiplicadores en categorías como producto, plataforma, personal y proyecto.
5. Cálculo del esfuerzo: Esfuerzo = A * Size^ E * EMi Donde A es una constante, Size es el tamaño en KSLOC, E es un exponente derivado de los factores de escala, y EMi son los multiplicadores de esfuerzo.
6. Estimación del cronograma y costo: Basado en el esfuerzo calculado y considerando factores como el costo por persona-mes.

CAPÍTULO III
MARCO APLICATIVO

INGENIERÍA
DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO

3. CAPÍTULO III

3.1. DISEÑO METODOLOGICO UWE

La metodología UWE se centra en el desarrollo de aplicaciones web utilizando diagramas UML. Su enfoque permite modelar de manera estructurada aspectos como la navegación, presentación y lógica de negocio, facilitando la comprensión y mantenimiento del sistema. Además, promueve la automatización en la generación de código a partir de los modelos.

3.1.1. *Fase de Análisis de Requisitos*

La fase de análisis de requisitos se enfoca en identificar y documentar las necesidades del usuario y los objetivos del sistema. Implica la recopilación de información a través de entrevistas y encuestas, asegurando que todos los requerimientos funcionales y no funcionales queden claramente definidos para guiar el desarrollo.

3.1.1.1. Análisis de Situación Actual

Actualmente, la Alcaldía de El Alto enfrenta varios desafíos en la gestión de luminarias públicas. El proceso de registro y seguimiento se realiza de forma manual, lo que genera problemas como:

- Pérdida de información
- Falta de control
- Ineficiencia operativa

3.1.1.2. Ingeniería de Requerimientos

Es el proceso sistemático de identificar, analizar, documentar y gestionar los requisitos de un sistema. Su objetivo es garantizar que las necesidades del usuario y las restricciones

del entorno se traduzcan en especificaciones claras, precisas y verificables, sirviendo como base para el diseño y desarrollo del software.

3.1.1.3. Requerimientos Funcionales

Tabla 1.

Requerimientos Funcionales

Registro de luminarias públicas	El sistema permitirá registrar la luminaria, tipo y modelo de las luminarias.
Seguimiento de mantenimiento	Se debe poder realizar un seguimiento del estado y las tareas de mantenimiento, registrando las reparaciones, componentes en mal estado.
Control de inventario de materiales y herramientas	Gestión eficiente del inventario, permitiendo agregar, actualizar y eliminar datos sobre herramientas y materiales utilizados en el mantenimiento de las luminarias.
Generación de reportes	El sistema generará reportes automáticos sobre el mantenimiento realizado, inventario disponible y las tareas de mantenimiento realizadas.
Asignación de técnicos	El sistema deberá permitir asignar técnicos a los diferentes distritos y hacer un seguimiento de las tareas que estos realicen en campo.
Historial de actividades	Registro completo de todas las intervenciones, mantenimiento e instalación realizadas sobre las luminarias.
Notificaciones	El sistema deberá notificar a los responsables sobre las urbanizaciones que necesiten mantenimiento o instalación o inspección.
Registrar urbanizaciones	El sistema deberá poder agregar nuevas urbanizaciones en los 14 distritos.
Registrar luminarias retiradas	Registro de todas las luminarias retiradas, como también la fecha, numero sisco, cantidad y lugar.

3.1.1.4. Requerimientos No Funcionales

Tabla 2.

Requerimientos No Funcionales

RNFO 1	El sistema deberá implementar mecanismos de seguridad como autenticación, cifrado de datos sensibles y roles de acceso para distintos usuarios.
RNFO 2	Deberá permitir el crecimiento en número de usuarios y luminarias sin afectar el rendimiento.
RNFO 3	Optimización para manejar grandes volúmenes de datos y múltiples usuarios simultáneos sin degradación.
RNFO 4	Interfaz intuitiva y fácil de usar para diferentes tipos de usuarios, desde técnicos de mantenimiento hasta gestores municipales.
RNFO 5	El sistema debe ser capaz de funcionar en escenarios de conectividad limitada, sincronizando los datos cuando sea posible.
RNFO 6	Diseño modular y con una estructura clara que facilite las actualizaciones y el mantenimiento del sistema.
RNFO 7	El sistema web debe funcionar correctamente en cualquier navegador, como Firefox, Chrome, Edge, opera, brave
RNFO 8	El sistema usara servicios Nginx

3.1.1.5. Identificación de Actores

Estas personas están encargadas de realizar las siguientes tareas:

- **Jefe de Dirección:**

- Designa roles a los usuarios
- Restringe el uso del sistema
- Pide reportes de trabajos realizados
- Agenda trabajos
- Información de todos los trabajos

- Detalles de equipamiento Distrital
- Detalles de accesorios
- Registrar inspecciones
- Registrar Urbanizaciones
- **Jefe de Unidad:**
- Designa roles a los usuarios
- Restringe el uso del sistema
- Pide reportes de trabajos realizados
- Agenda trabajos
- Información de todos los trabajos
- Detalles de equipamiento Distrital
- Detalles de accesorios
- Modificar Información
- Registrar inspecciones
- Registrar proyectos
- Instalar proyectos
- **Secretaria:**
- Detalles de trabajo
- Información de proyectos
- Detalles de inspección
- Detalles de equipamiento Distrital
- Detalles de accesorios
- Detalles de urbanización
- **Jefatura:**
- Detalles de trabajo

- Información de proyectos
- Detalles de inspección
- Detalles de equipamiento Distrital
- Detalles de accesorios
- Registrar trabajo
- Asignar inspección
- Registrar urbanizaciones
- Registrar equipamiento
- Registrar instalación
- **Técnico:**
- Realizar trabajo
- Detalles de trabajo
- Información de proyectos
- Detalles de inspección
- Registrar proyectos
- Instalar proyectos
- Registrar inspecciones
- Detalles de equipamiento Distrital
- Detalles de accesorios
- Registrar urbanizaciones
- Registrar equipamiento
- Registrar instalación

3.1.2. ***Diagrama de Caso de Uso General***

Figura *16.*

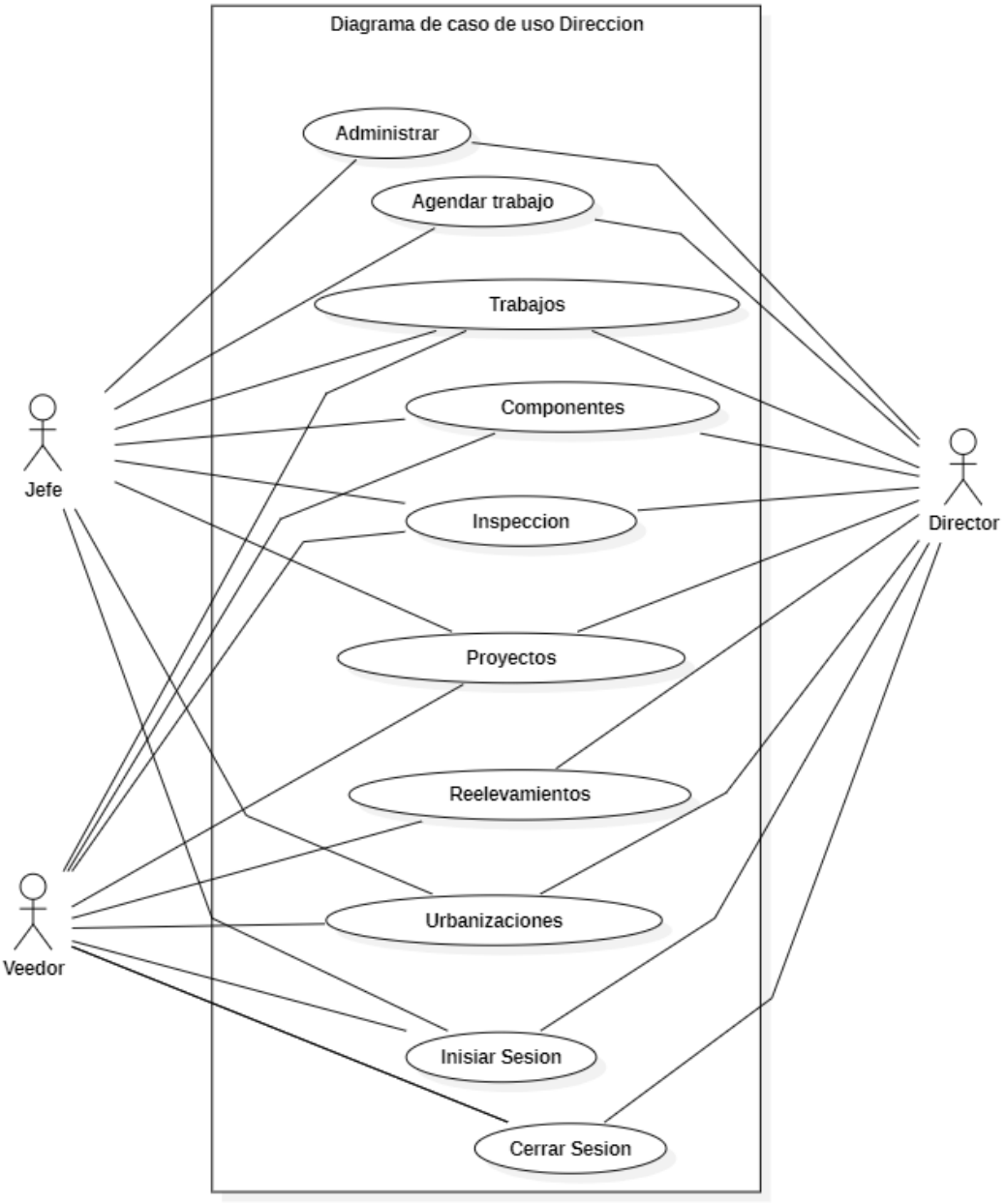
Diagrama de Caso de Uso General

3.1.3. ***Diagrama de Caso de Uso Dirección***

A continuación, se realiza el modelado donde se puede apreciar cómo interactúan los actores de la Dirección de Alumbrado Público.

Figura *17.*

Diagrama de Caso de Uso Dirección

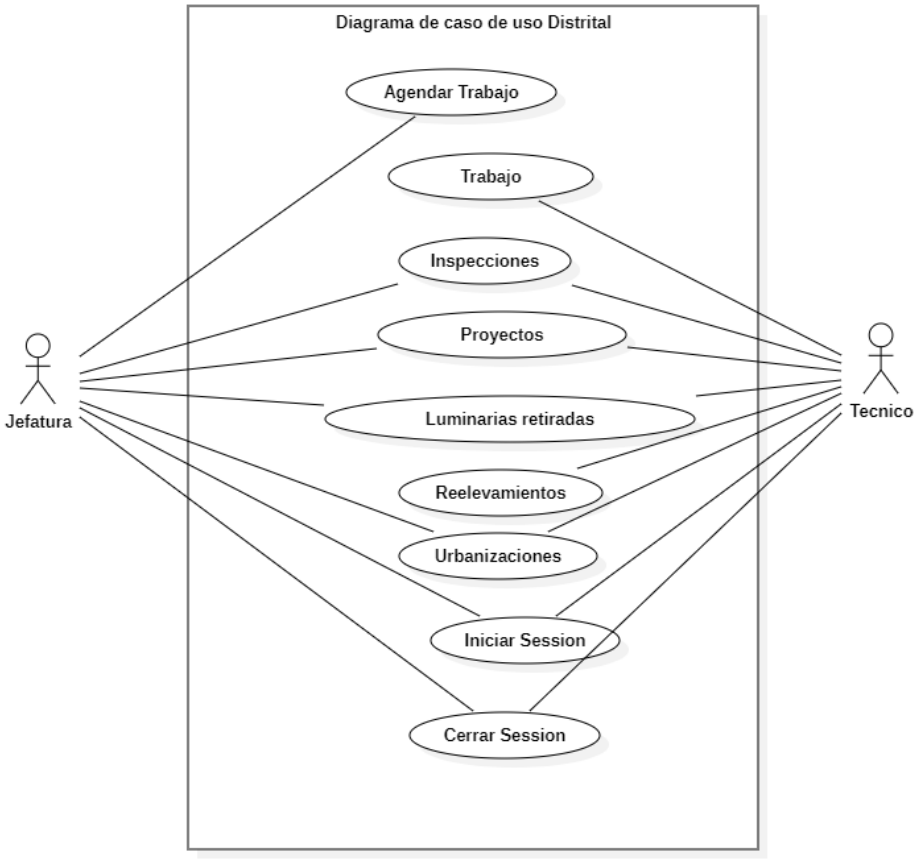


3.1.4. ***Diagrama de Caso de Uso Distrital***

El modelado presentado a continuación muestra cómo los actores de los diferentes distritos interactúan entre sí en el sistema. A través de este proceso, se visualizan las relaciones y responsabilidades que cada actor tiene dentro de la estructura organizativa.

Figura *18.*

Diagrama de Caso de Uso Distrital



3.2. FASE DE DISEÑO

A continuación, se elabora una serie de diagramas utilizando la notación UWE, siguiendo la metodología de desarrollo aplicada en el proyecto. Esto permitirá un modelado y diseño más eficiente de los distintos diagramas del sistema, asegurando una mayor claridad y organización.

3.2.1. *Modelo de Caso de Uso*

Los modelos de casos de uso describen un sistema en términos de las diversas formas en que puede ser utilizado, denominadas casos de uso. Cada caso de uso consiste en una secuencia de acciones iniciada por el usuario.

A continuación, se llevará a cabo el análisis de los requisitos del sistema, representando los casos de uso a través del comportamiento del sistema frente a las acciones de los actores involucrados.

- **Caso de Uso:** Administración de Usuarios

En la Figura 19, se observa el proceso de administración de usuarios que realiza el administrador donde puede modificar los datos como también puede eliminar a los usuarios que tengan el rol de Coordinador, Veedor y Técnico

Figura *19.*

Caso de uso Administración de Usuario

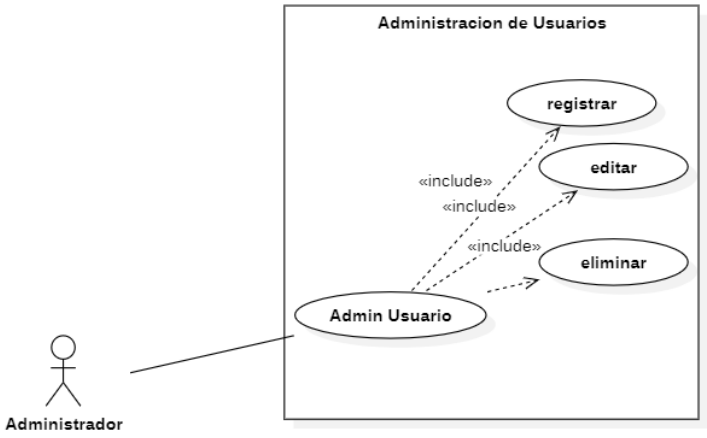


Tabla 3.

Especificación de Caso de Uso Administración de Usuarios

Administración de Usuarios	
Actores:	Administrador
Tipo:	Primario Esencial
Descripción:	El administrador registrar usuarios con los privilegios de acceso correspondientes a la función que desempeñan para que los mismos tengan acceso al sistema

- **Caso de Uso:** Dirección de Alumbrado Público

En la Figura 20 se observa el modelo de caso de uso de Administración

Figura *20.*

Caso de Uso Dirección

Tabla 4.

Especificación de Caso de Uso Dirección

	Administración de la información
Actores:	Administrador y Veedor
Tipo:	Primario Esencial

Administración de la información	
Descripción:	El administrador registra usuarios, trabajos, inspecciones, proyectos, urbanizaciones y equipamientos. Agenda trabajos, gestiona proyectos de las diferentes urbanizaciones, realiza la instalación de nuevos proyectos, emitir reportes de trabajos realizados y de proyectos

- **Caso de Uso:** Agendar Trabajo

En la Figura 21, se muestra el modelo de caso de uso del proceso de agendar trabajo para los diferentes distritos.

Figura *21.*

Caso de Uso Agendar Trabajo

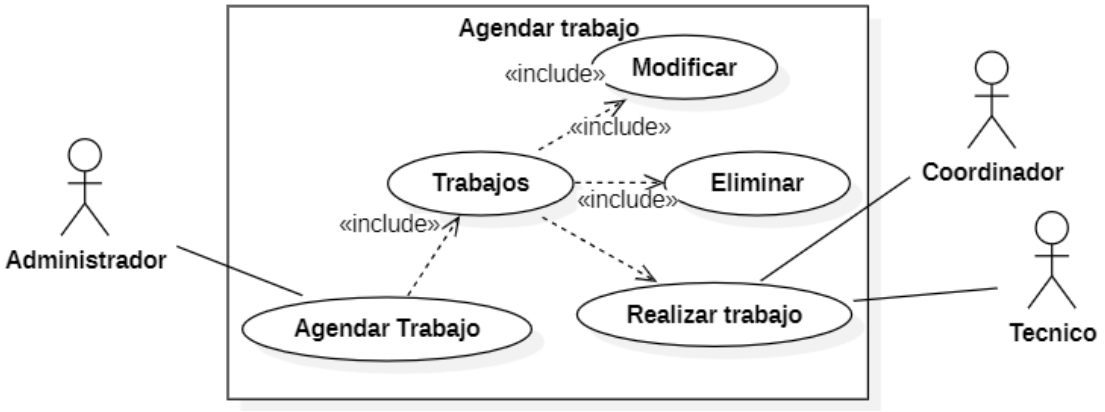


Tabla 5.

Especificación de Caso de Uso Agendar Trabajo

Agendar Trabajo	
Actores:	Administrador, Coordinador y Técnico
Tipo:	Primario Esencial
Descripción:	El administrador registra trabajo, tiene la opción de modificar y eliminar los trabajos agendados, el Coordinador y el Técnico, tiene acceso a los trabajos a realizar en su distrito, pero no pueden modificar ni eliminar.

- **Caso de Uso:** Detalles de Trabajo

En la Figura 22, se muestra el modelo de caso de uso de detalles de trabajos.

Figura *22.*

Caso de Uso Detalles de Trabajo

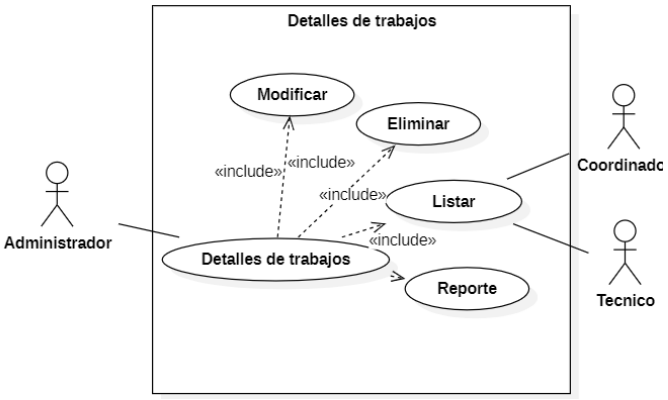


Tabla 6.

Especificación de Caso de Uso Detalles de Trabajo

	Detalles de Trabajo
Actores:	Administrador, Coordinador, Veedor y Técnico
Tipo:	Primario Esencial
Descripción:	El administrador tiene acceso a todos los detalles de los diferentes distritos, puede modificar, eliminar y sacar reporte de todos los trabajos. El Coordinador y el técnico pueden ver todos los trabajos pertenecientes a su distrito

- **Caso de Uso:** Proyectos

En la Figura 23, se representó las interacciones que los diferentes actores tienen con el sistema de gestión de proyectos. Como se puede ver en la Figura 24, los actores principales son el Administrador, el Coordinador, el Técnico y el Veedor. Tanto el Administrador como el Coordinador tienen la capacidad de registrar proyectos en el sistema. Sin embargo, el Administrador tiene más responsabilidades, ya que además de registrar, puede modificar, eliminar proyectos y generar reportes. Los actores como el Técnico y el Veedor están más enfocados en tareas específicas, como listar e instalar los proyectos. He vinculado estos casos de uso con el principal, "Proyectos", a través de <<include>>, ya que las acciones de registrar, modificar, eliminar y generar reportes dependen de la gestión básica de proyectos. De esta

forma, he estructurado el sistema para que cada actor tenga los permisos y funciones adecuados según su rol dentro de la organización.

Además, he diseñado el sistema de manera que cada actor solo pueda acceder a las funcionalidades que le corresponden según su rol, garantizando así la seguridad y el control adecuado de la información.

Figura 23.

Caso de Uso Proyectos

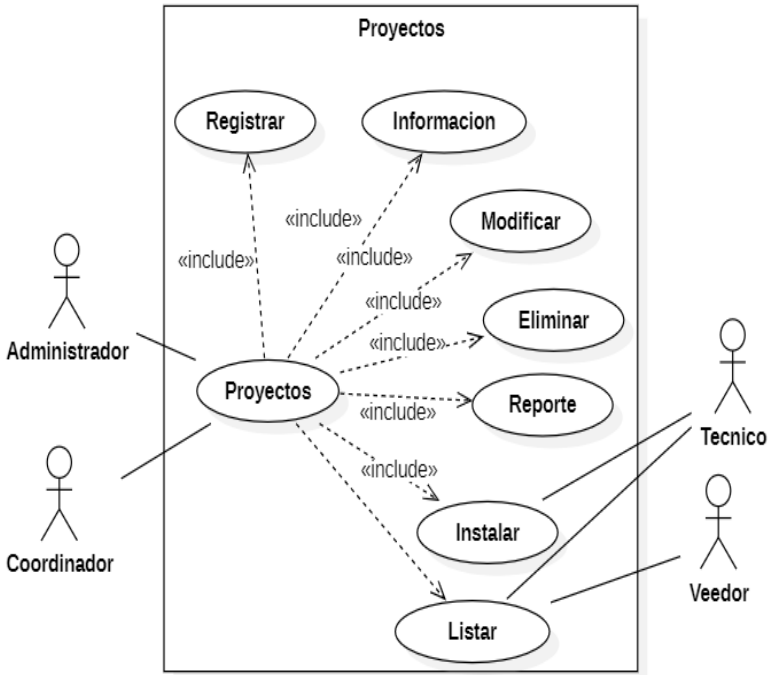


Tabla 7.

Especificación de Caso de Uso Proyectos

	Proyectos
Actores:	Administrador, Veedor, Coordinador y Técnico
Tipo:	Primario Esencial
Descripción:	Tanto como el Administrador, Coordinador y el técnico pueden agregar proyectos solo que el administrador puede modificar y eliminar, pueden hacer la instalación del mismo como también generar reporte.

- **Caso de Uso:** Luminarias Retiradas

En la Figura 24, se muestra el modelo de caso de uso de luminarias retiradas, en el cual los actores principales son el Administrador, el Coordinador y el Técnico, cada uno con roles y responsabilidades claramente definidas dentro del sistema. Tanto el Administrador como el Coordinador pueden registrar y modificar la información relacionada con las luminarias retiradas, lo que les permite mantener actualizada la base de datos del sistema. Además, el Administrador tiene la capacidad exclusiva de generar reportes, lo que le da un mayor control sobre la supervisión y análisis de los datos gestionados. Por su parte, el Técnico solo puede listar las luminarias retiradas, lo que le permite acceder a la información sin tener permisos de modificación.

Figura *24.*

Caso de Uso Luminarias Retiradas

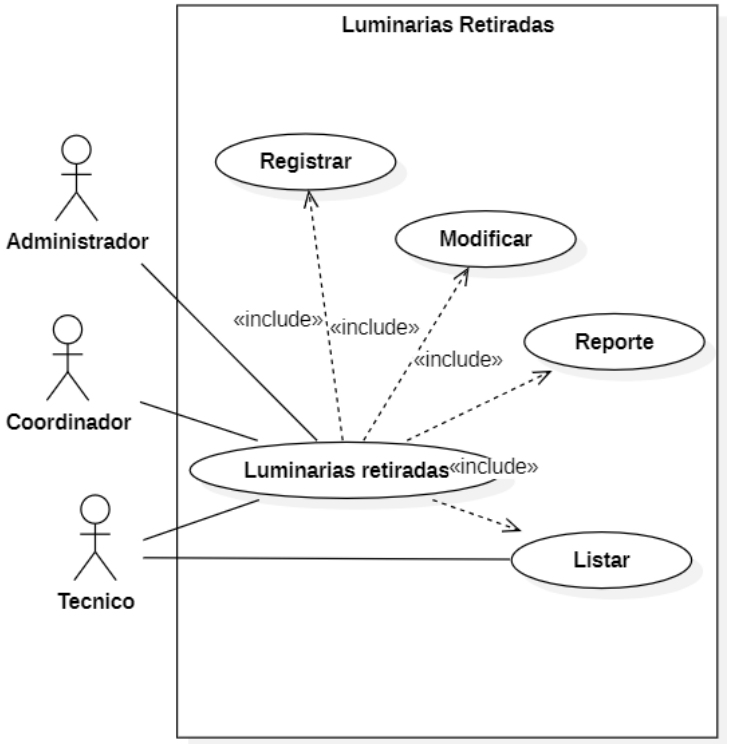


Tabla 8.

Especificación de Caso de Uso Luminarias Retiradas

	Luminarias Retiradas
Actores:	Administrador, Veedor, Coordinador y Técnico
Tipo:	Primario Esencial
Descripción:	el Administrador, Coordinador y el técnico pueden agregar Luminarias retiradas solo que el administrador puede modificar, eliminar y generar reporte. El Veedor solo puede ver la información

- **Caso de Uso:** Inspección

En la Figura 25, se muestra el modelo de caso de uso del proceso de Inspeccionar trabajos.

Figura *25.*

Caso de Uso Inspección

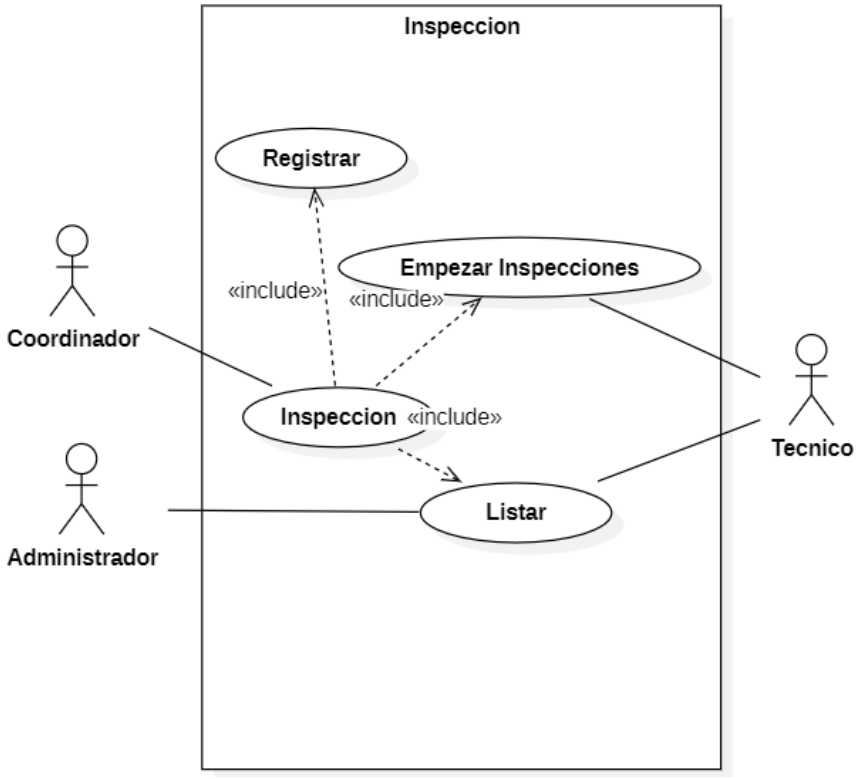


Tabla 9.

Especificación de Caso de Uso Inspección

	Inspección
Actores:	Administrador, Coordinador y Técnico
Tipo:	Primario Esencial
Descripción:	El Administrador, Coordinador y el técnico pueden agregar Inspecciones solo que el administrador puede modificar y eliminar. El técnico y el coordinador pueden hacer las inspecciones del distrito al que pertenecen. Todos tienen acceso a la información del lugar designado, los administradores tienen acceso a toda la información de todos los distritos en general

- **Caso de Uso:** Equipamiento

En la Figura 26, se muestra el modelo de caso de uso de Equipamiento

Figura *26.*

Caso de Uso Equipamiento

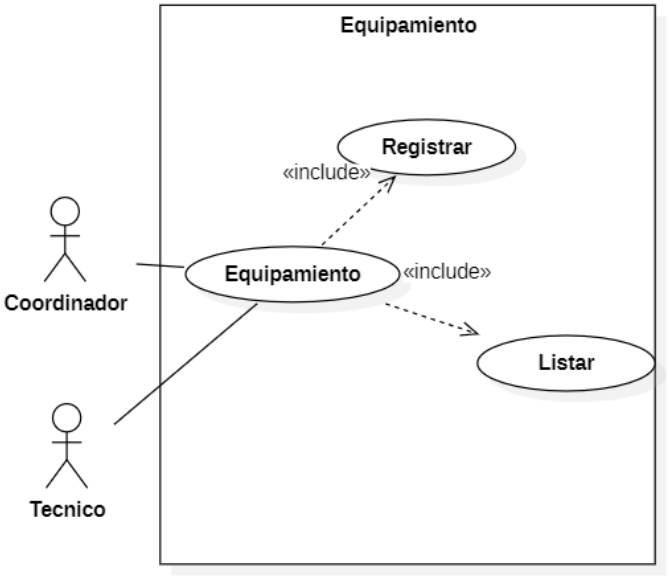


Tabla 10.

Especificación de Caso de Uso Equipamiento

	Equipamiento
Actores:	Coordinador y Técnico

Tipo:	Primario Esencial
Descripción:	El Coordinador y el técnico pueden agregar los equipos que tienen su distrito como también pueden listar, pero no pueden modificar ni eliminar la información

- **Caso de Uso:** Accesorios

En la Figura 27, se muestra el modelo de caso de uso de Accesorios

Figura *27.*

Caso de Uso Accesorios

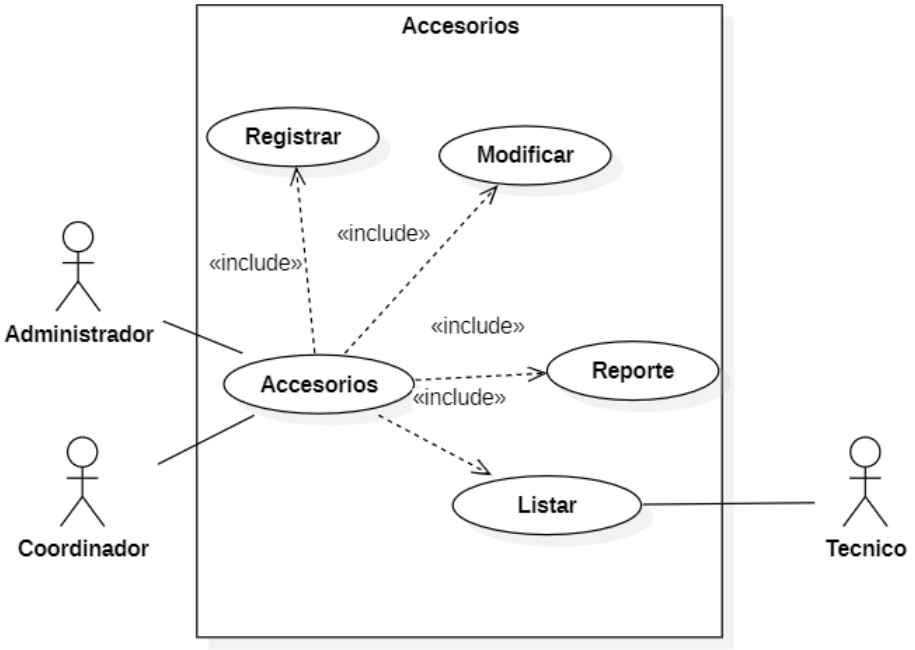


Tabla 11.

Especificación de Caso de Uso Accesorios

	Accesorios
Actores:	Administrador, Coordinador y Técnico
Tipo:	Primario Esencial
Descripción:	El Administrador, Coordinador y el técnico pueden agregar Accesorios que se utiliza en los diferentes tipos de trabajo, pero solo el administrador puede eliminar, modificar y sacar reporte. El Veedor solo puede ver la información

- **Caso de Uso:** Urbanización

En la Figura 28, se muestra el modelo de caso de uso de Urbanización

Figura *28.*

Caso de Uso Urbanización

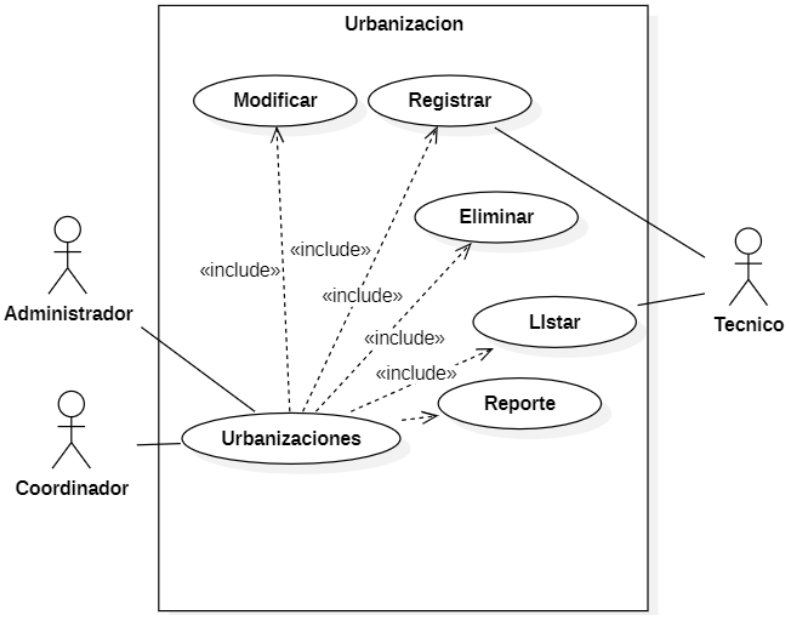


Tabla 12.

Especificación de Caso de Uso Urbanización

Urbanización	
Actores:	Administrador, Coordinador y Técnico
Tipo:	Primario Esencial
Descripción:	El Administrador, Coordinador y el técnico pueden agregar nuevas Urbanizaciones, pero solo administrador puede modificar y eliminar o generar reporte

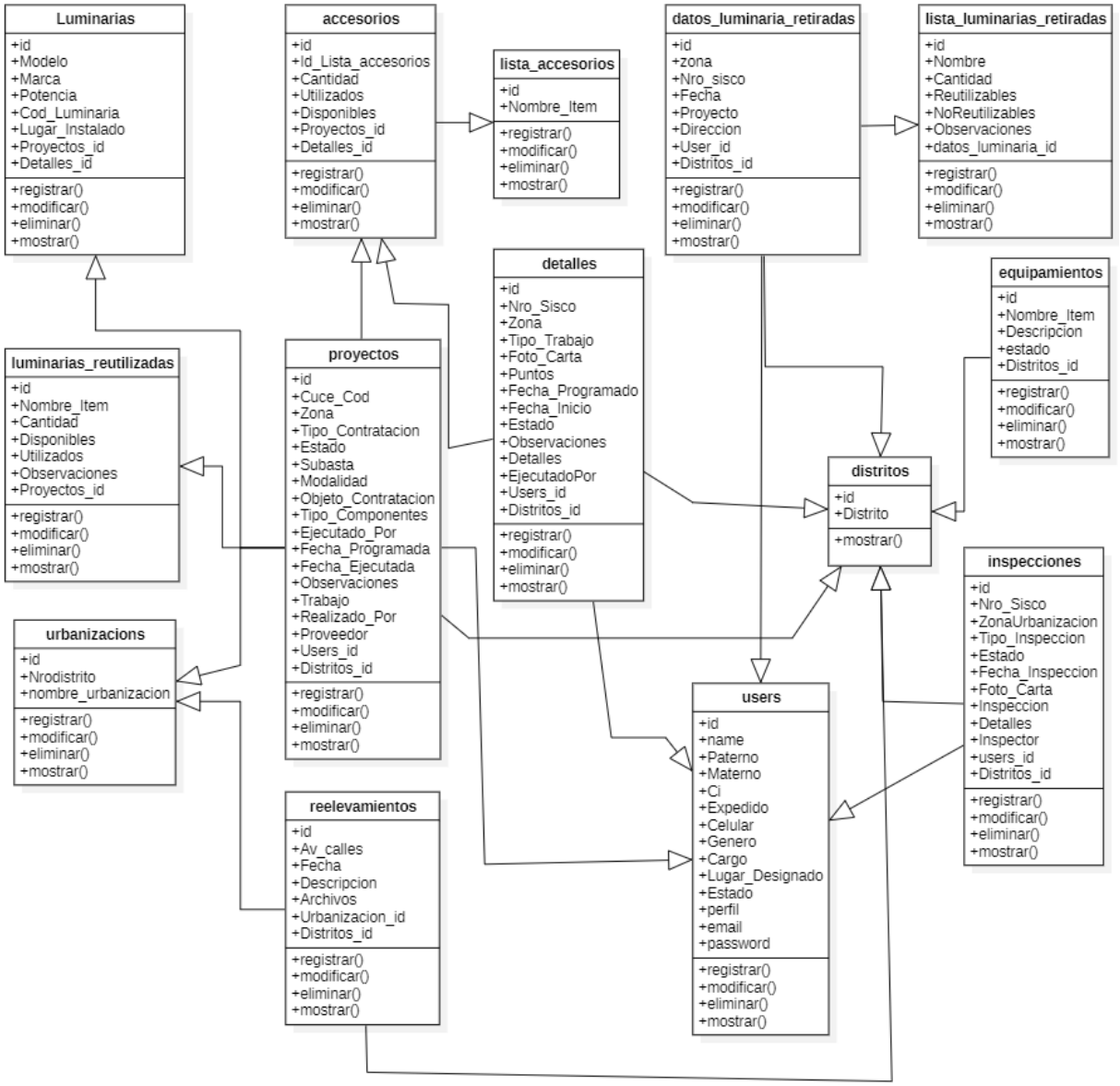
3.2.2. ***Diagrama de Clases***

El diagrama de clases es una representación gráfica de las clases que componen el sistema y sus relaciones. Este diagrama forma parte del lenguaje unificado de modelado (UML) y permite visualizar la estructura estática del sistema. Cada clase se muestra con sus

atributos (propiedades) y métodos (comportamientos), y las relaciones entre las clases pueden incluir herencia, asociación, agregación y composición. El diagrama de clases es esencial para definir cómo se organizarán los objetos en el sistema, sirviendo como base para la implementación del código, garantizando una estructura clara y coherente como se ve en la Figura 29.

Figura *29*.

Diagrama de Clases

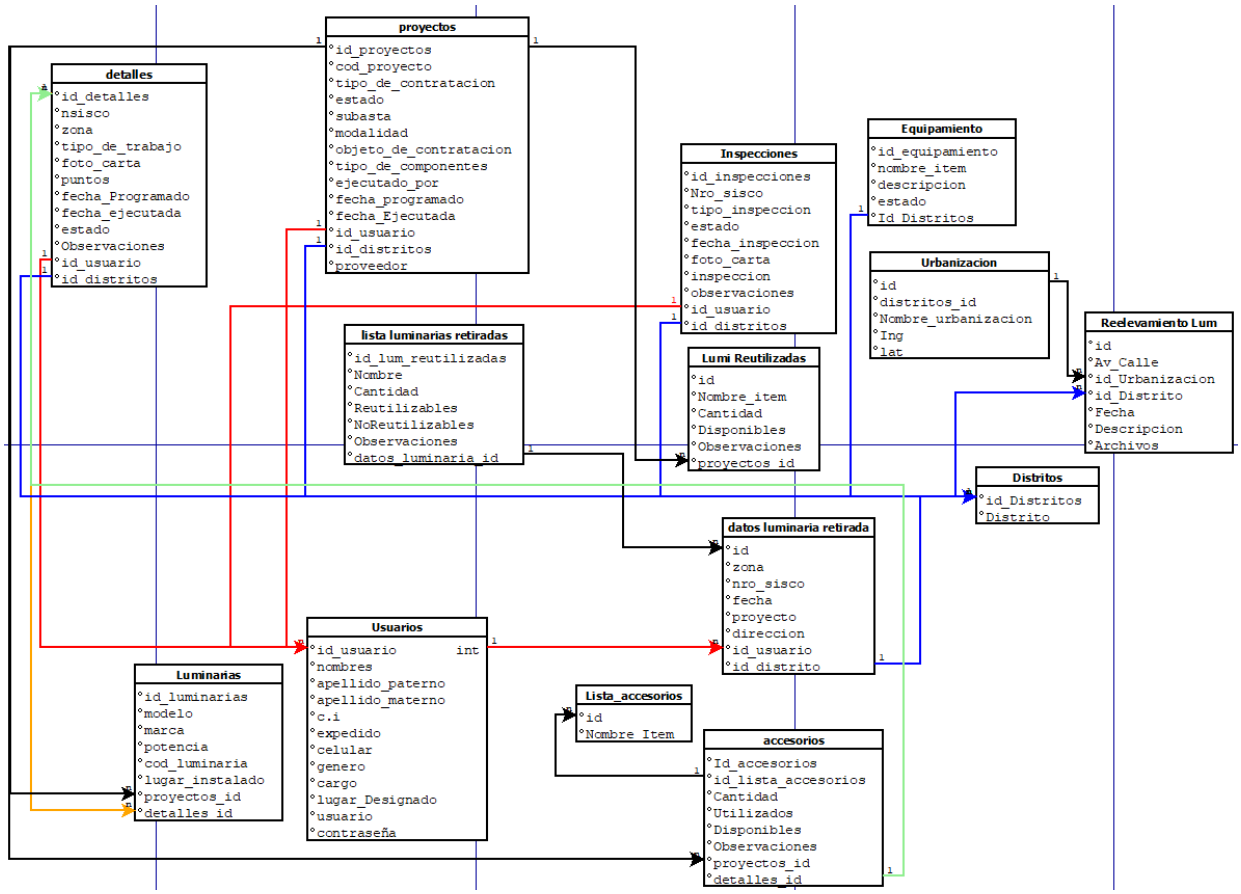


3.2.3. ***Modelo Conceptual***

El modelo conceptual establece una representación abstracta del dominio del sistema, basado en los requisitos extraídos de los casos de uso. Este modelo conceptual describe las entidades, relaciones y atributos clave que definen la estructura y comportamiento del sistema. En la Figura 30, se ilustra detalladamente el conjunto de tablas que conforman la base de datos del sistema, las cuales están diseñadas para reflejar las interacciones y dependencias entre los diferentes componentes. Estas tablas se han normalizado para optimizar la integridad de los datos y asegurar su consistencia dentro del sistema.

Figura *30.*

Modelo Conceptual



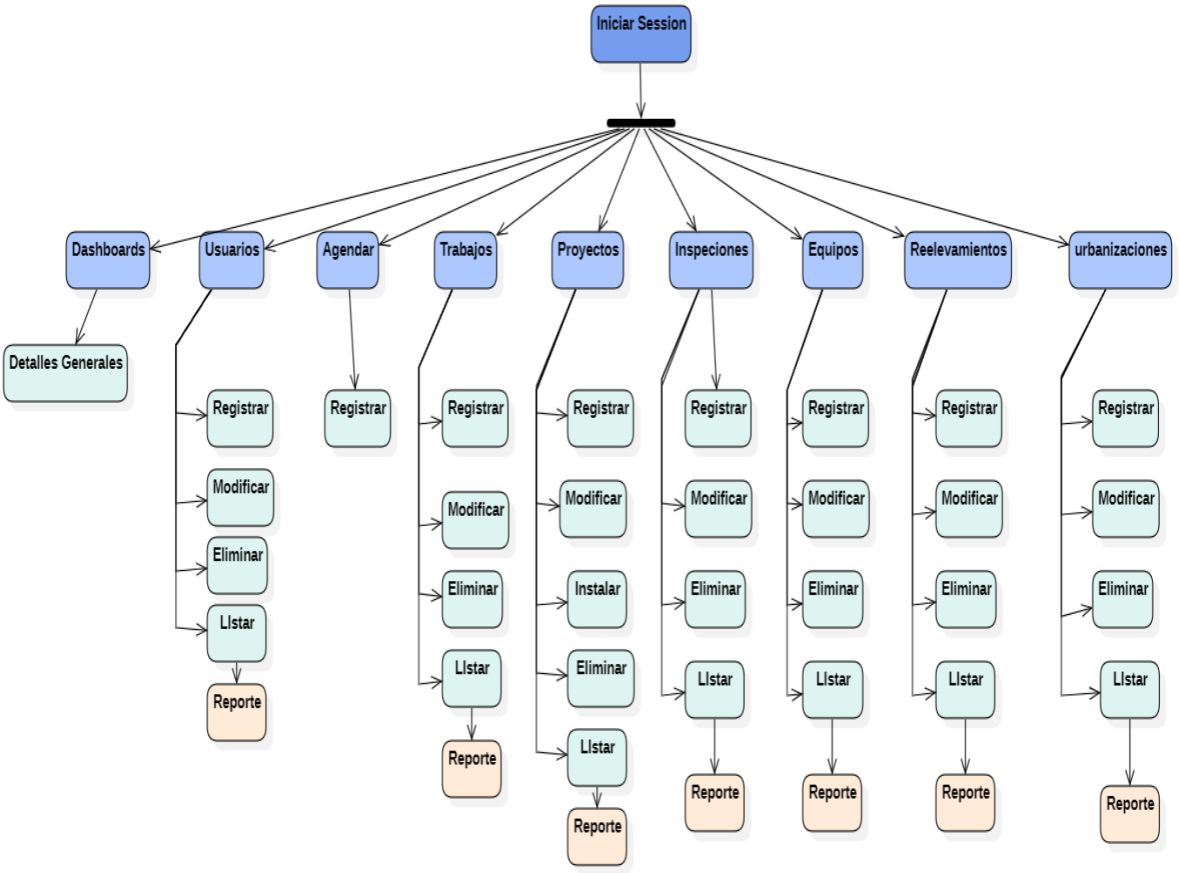
3.2.4. ***Modelo Navegacional***

En el modelo navegacional de la metodología UWE, defino las rutas y opciones de navegación que los usuarios tendrán en el sistema web. Este modelo me permite organizar de forma clara los enlaces entre las distintas páginas o vistas, asegurando que el acceso a la información y las funcionalidades sea eficiente. Baso este modelo en el modelo conceptual, garantizando que cada elemento navegacional esté directamente relacionado con los objetos y operaciones del dominio del sistema. Además, el modelo navegacional me ayuda a identificar restricciones de acceso y definir los flujos de interacción para diferentes tipos de usuarios, lo que facilita un diseño coherente y fácil de usar.

3.2.4.1. **Modelo Navegacional Administrador**

Figura *31.*

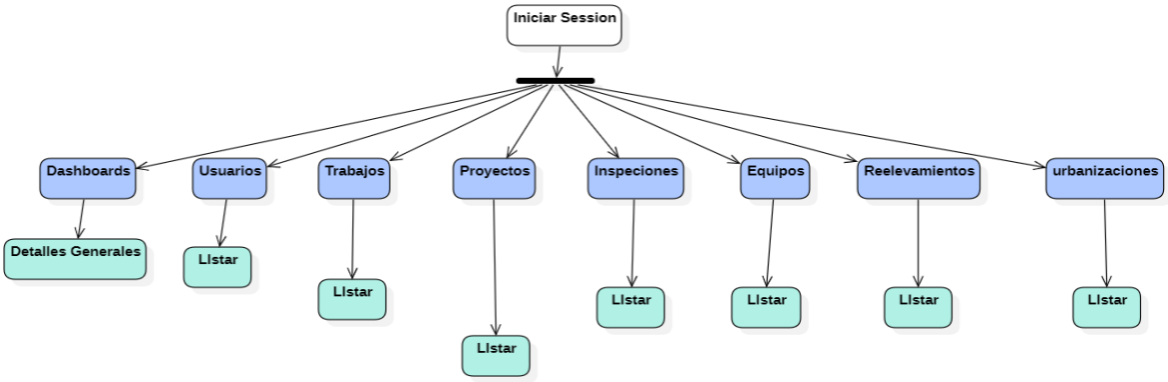
Modelo Navegacional Administrador



3.2.4.2. Modelo Navegacional Veedor

Figura *32.*

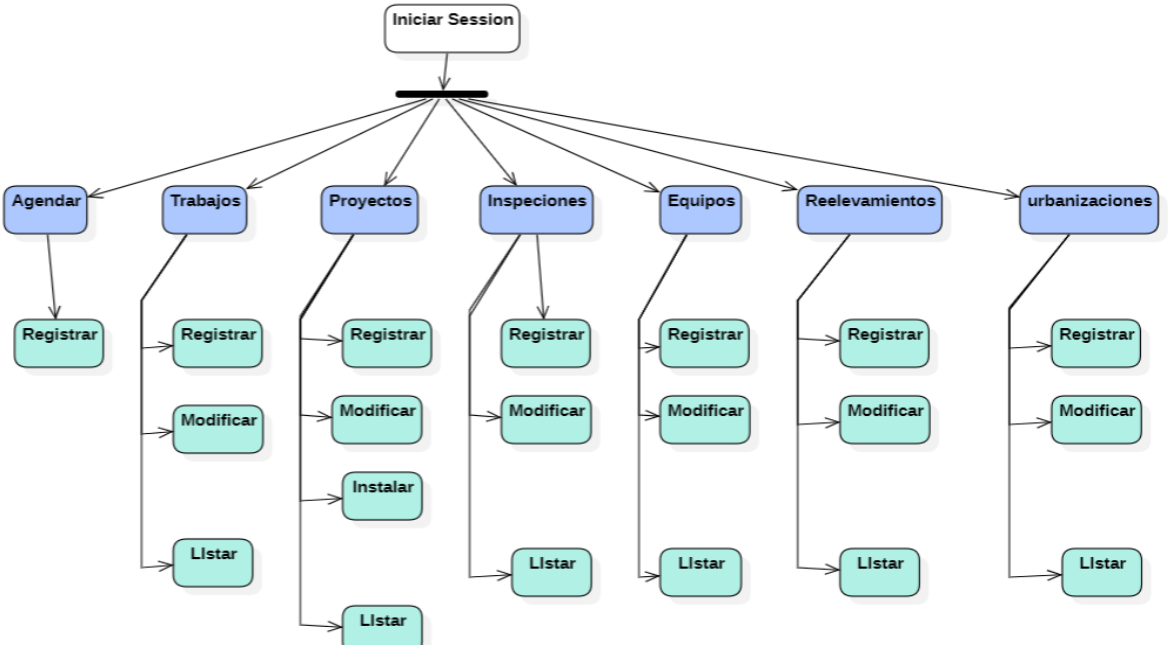
Modelo Navegacional Veedor



3.2.4.3. Modelo Navegacional Coordinador

Figura *33.*

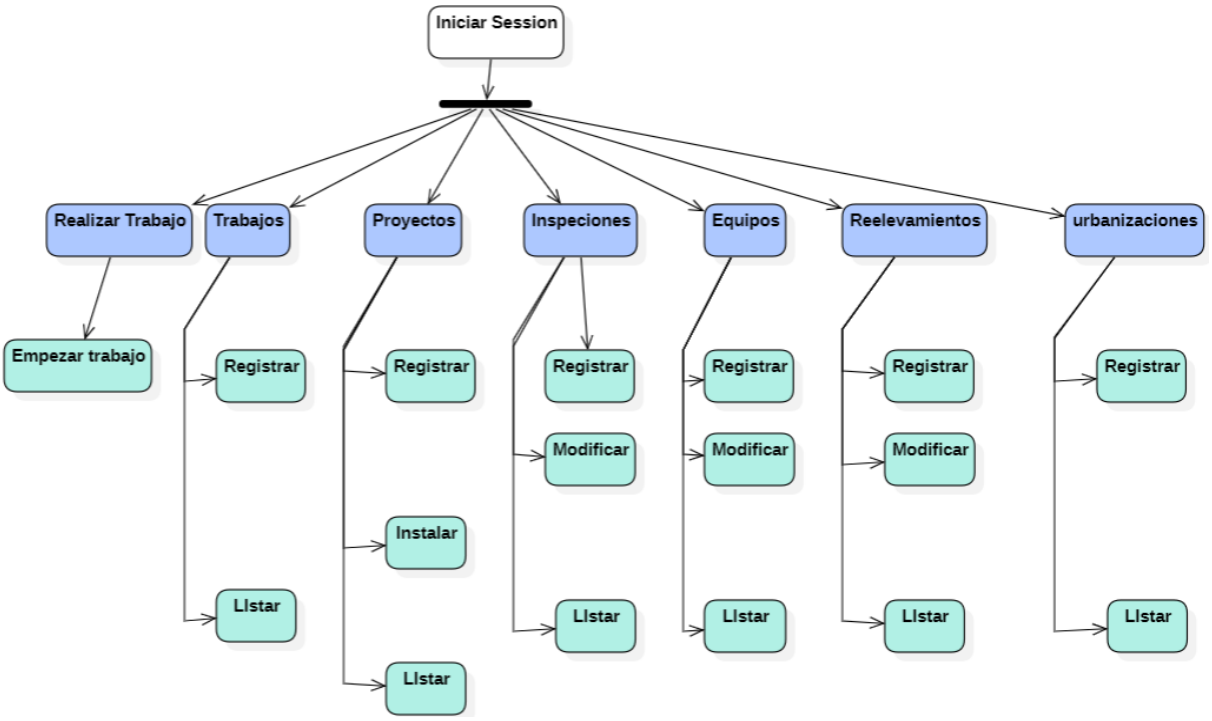
Modelo Navegacional Coordinador



3.2.4.4. Modelo Navegacional Técnico

Figura *34.*

Modelo Navegacional Técnico



3.2.5. *Modelo de Presentación*

En el modelo de presentación, me enfoco en cómo se visualiza la información y las funcionalidades del sistema web para el usuario final. Este modelo me permite definir la estructura y el diseño de las interfaces, asegurando que sean intuitivas y agradables visualmente. Aquí detallo la disposición de los elementos gráficos, textos, botones y otros componentes interactivos, siempre buscando mantener una experiencia de usuario eficiente y consistente. A través de este modelo, logro que las interfaces no solo sean funcionales, sino también atractivas y fáciles de navegar, lo que mejora la interacción del usuario con el sistema.

Figura *35.*

Modelo de Presentación - Inicio de Sesión al Sistema

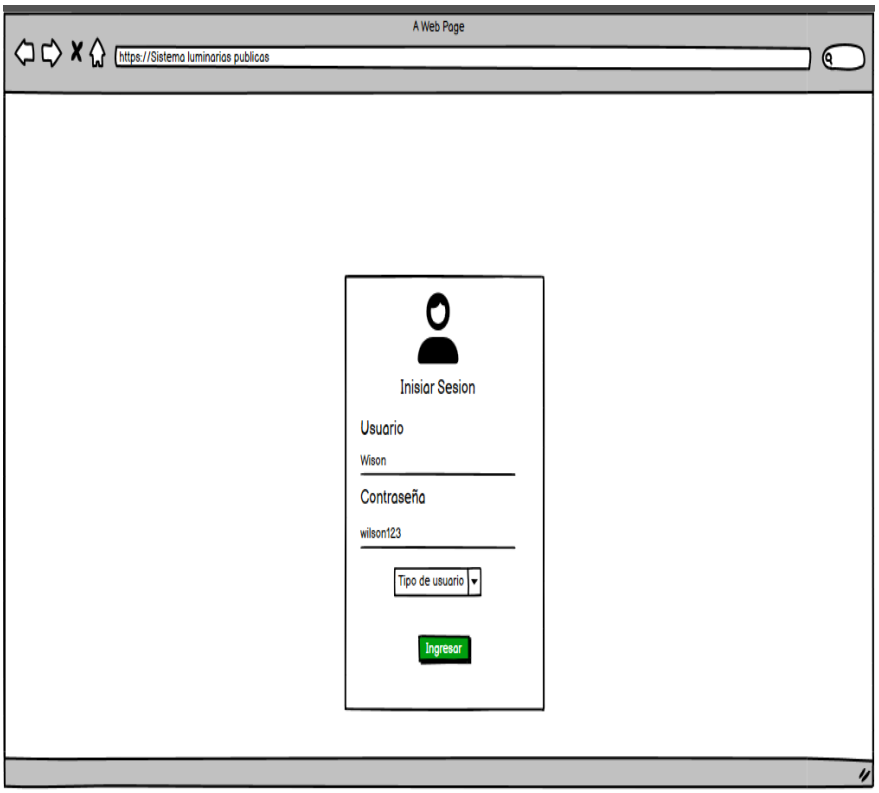


Figura *36.*

Modelo de Presentación - Administración de Usuarios

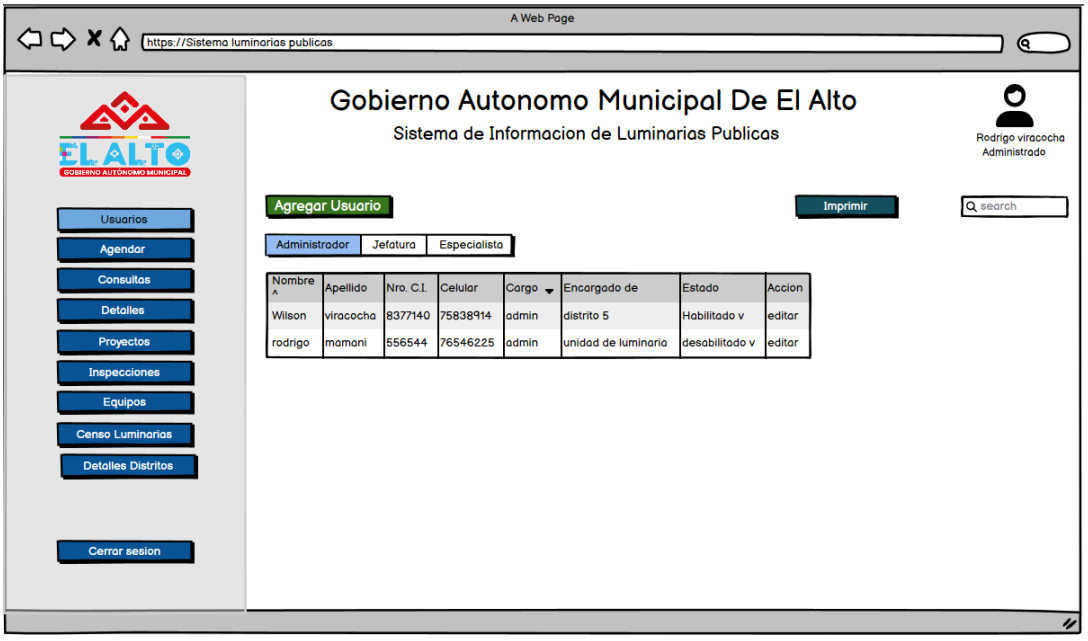


Figura *37.*

Modelo de Presentación - Detalles de Trabajos

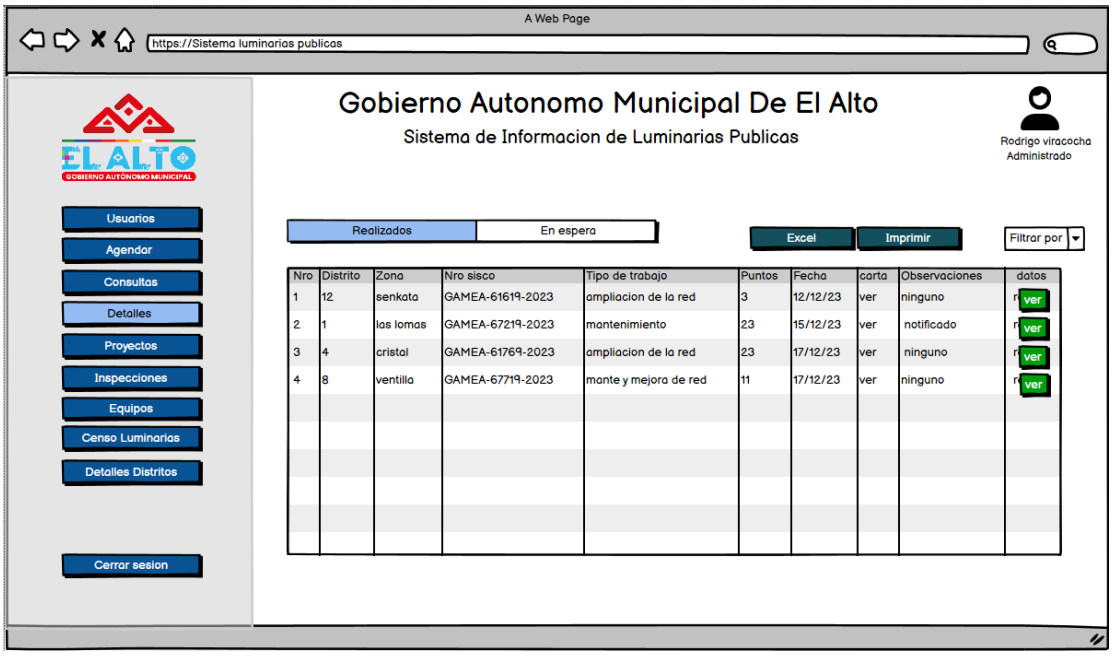


Figura *38.*

Modelo de Presentación - Agendar

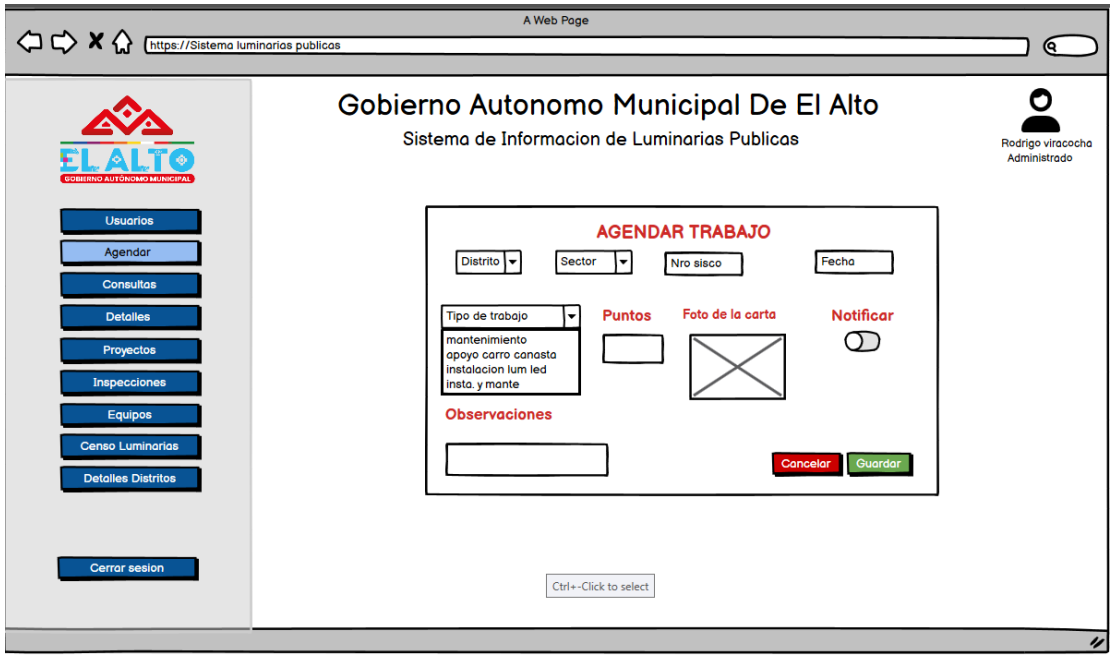


Figura *39.*

Modelo de Presentación - Proyectos

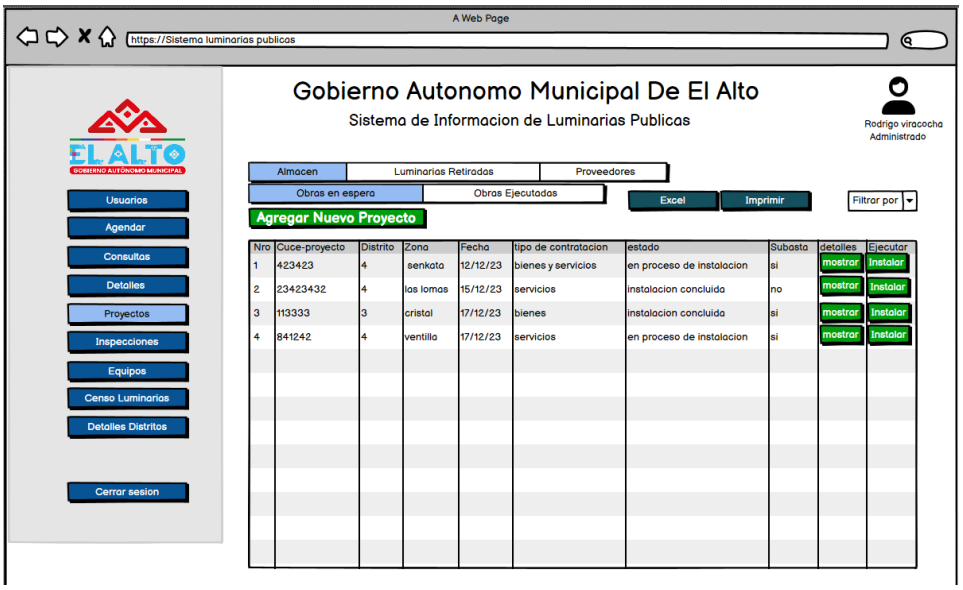


Figura *40.*

Modelo de Presentación - Registro de Proyectos

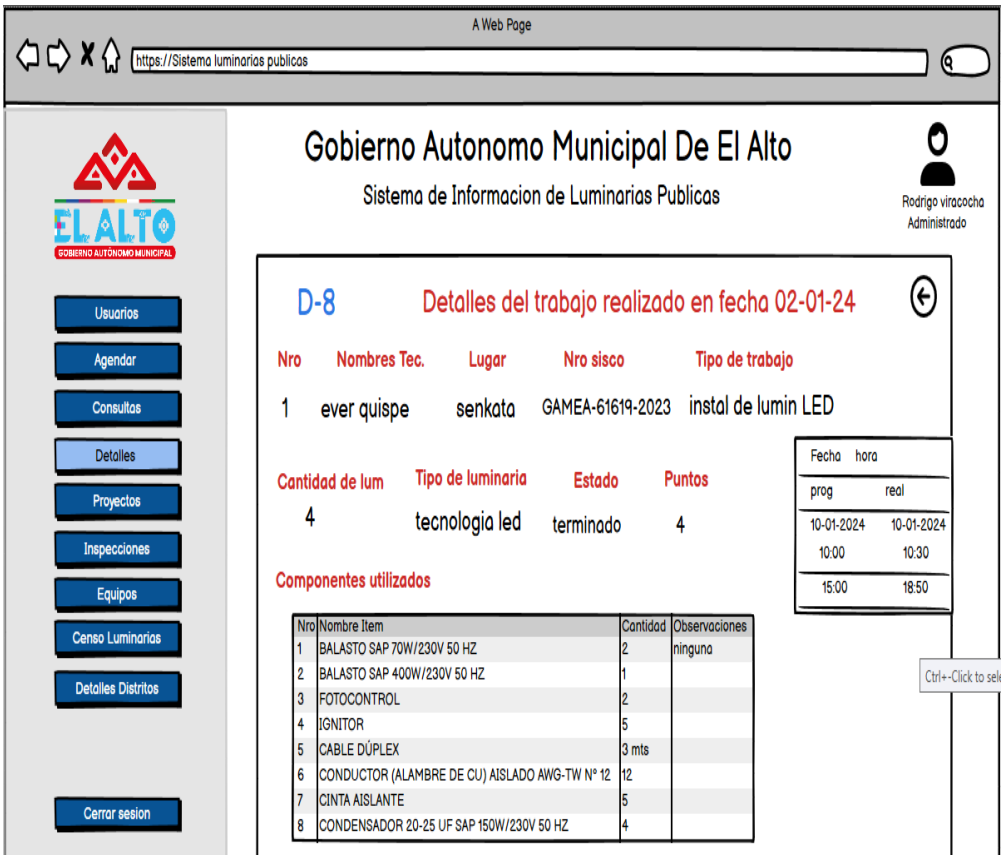


Figura *41.*

Modelo de Presentación - Empezar Instalación de Proyecto

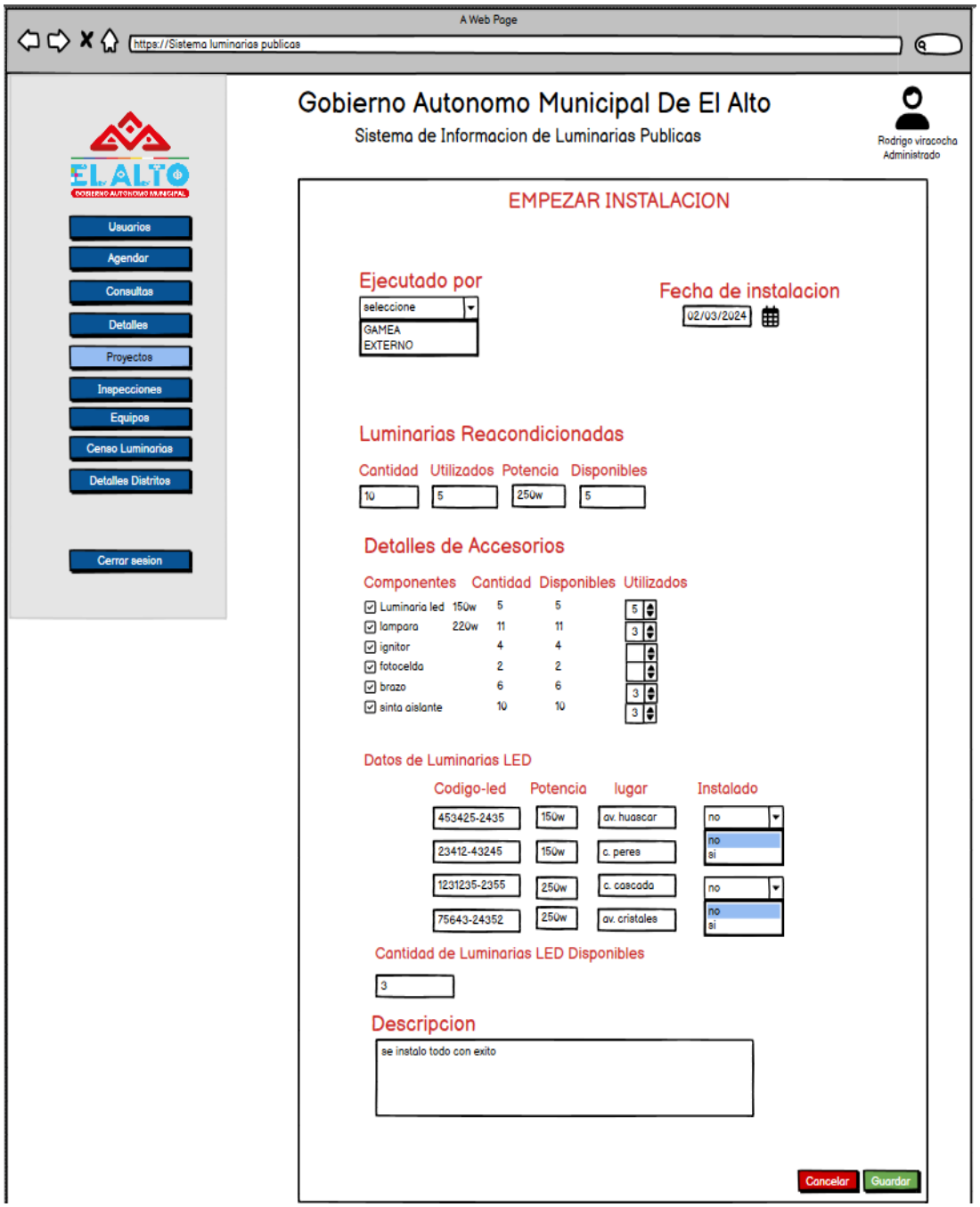


Figura *42.*

Modelo de Presentación - Luminarias Retiradas

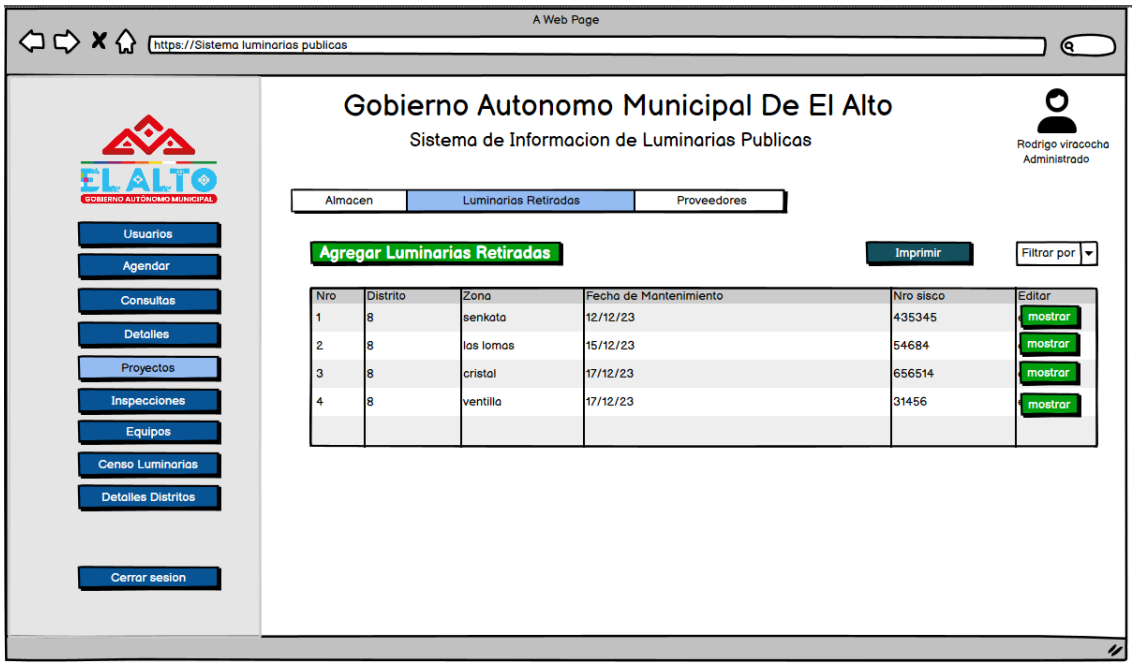


Figura *43.*

Modelo de Presentación - Registro de Luminarias Retiradas

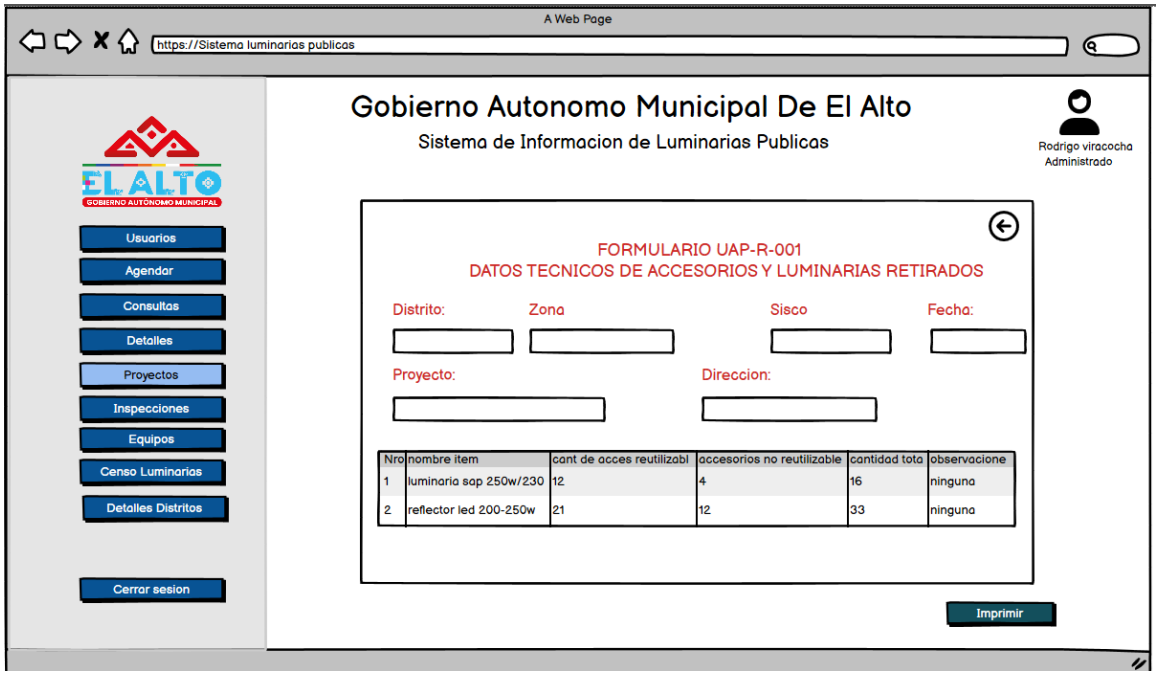


Figura *44.*

Modelo de Presentación - Inspecciones

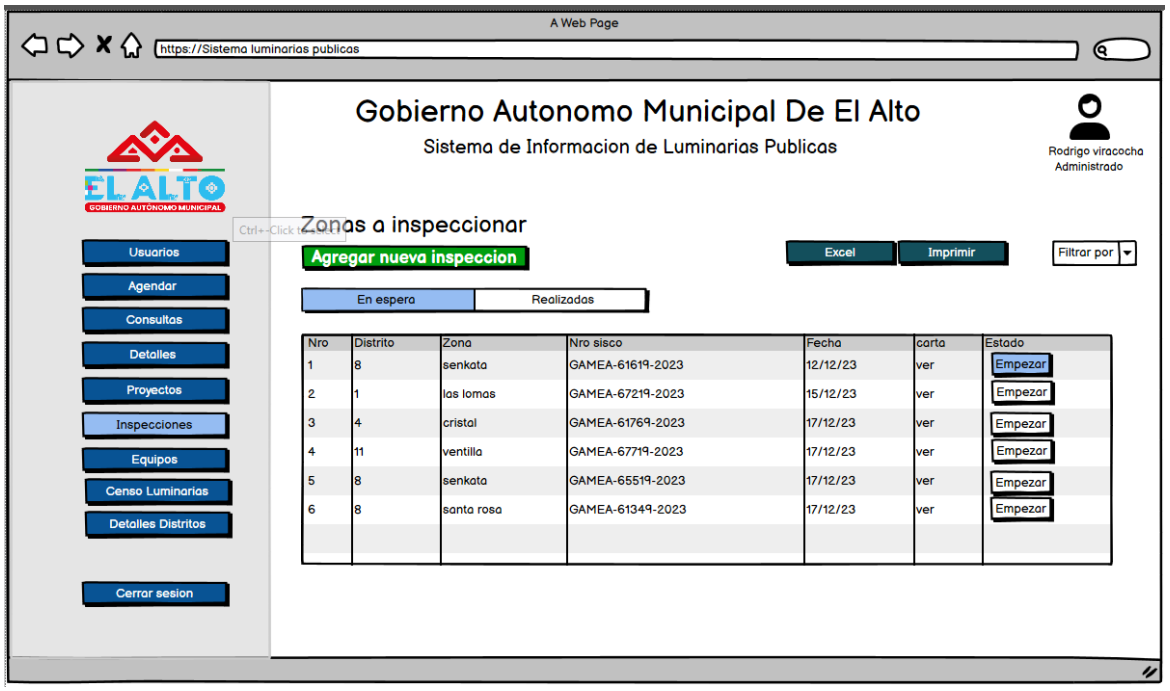


Figura *45.*

Modelo de Presentación - Registrar Inspección

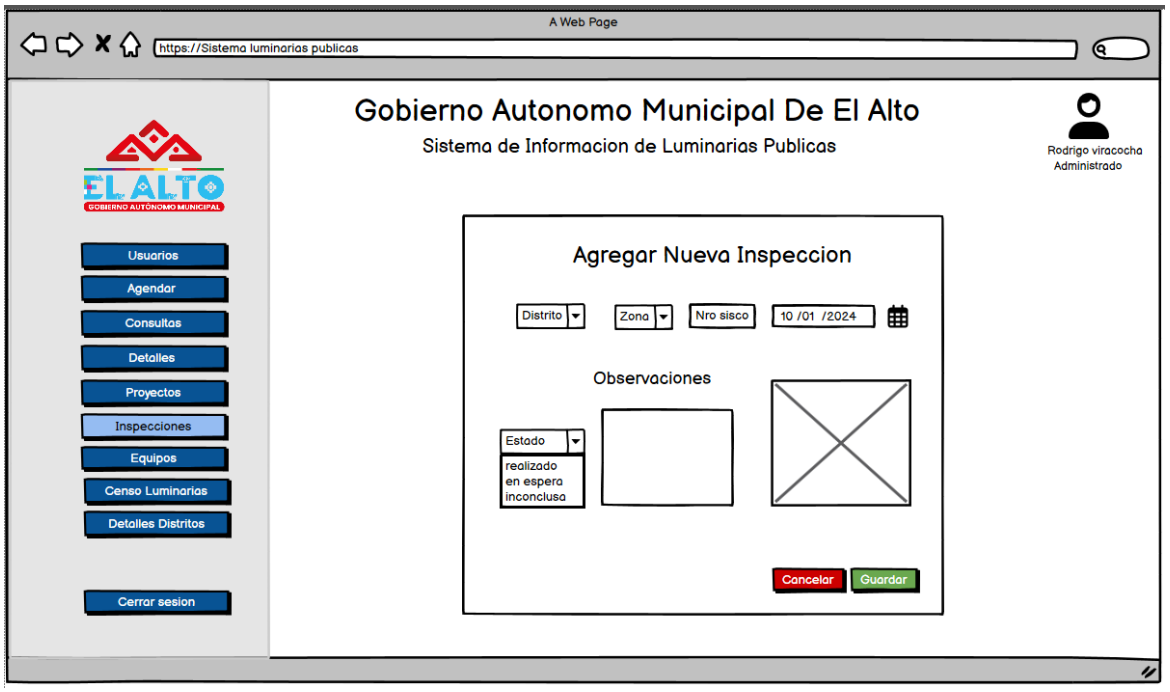


Figura *46.*

Modelo de Presentación - Equipamiento

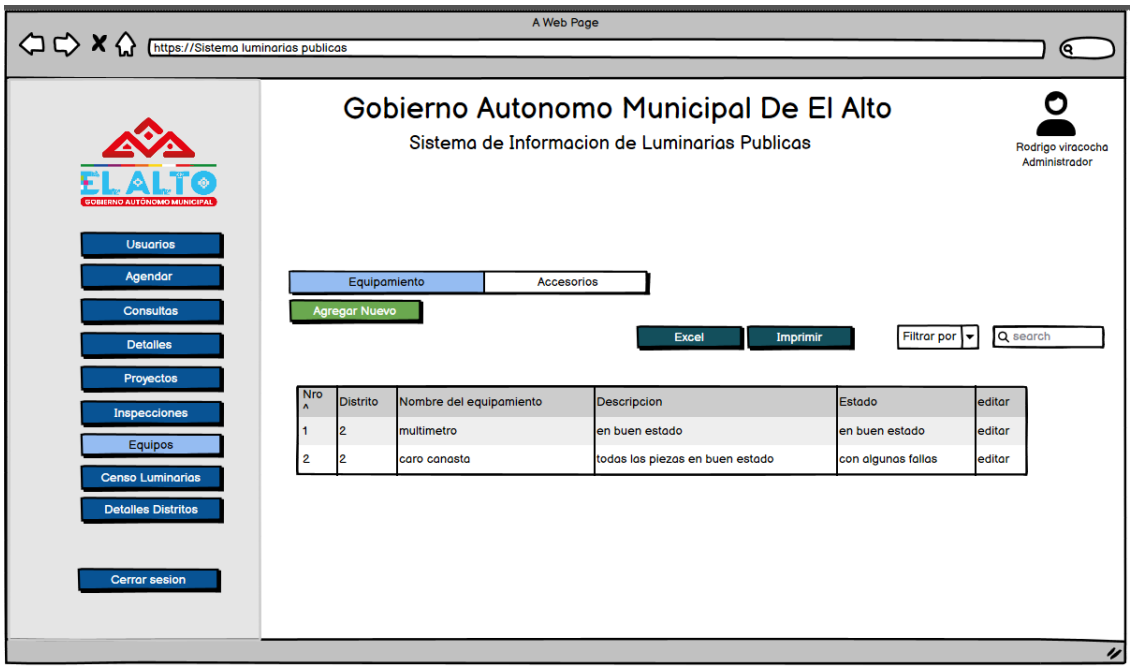


Figura *47.*

Modelo de Presentación - Registro de Equipamiento

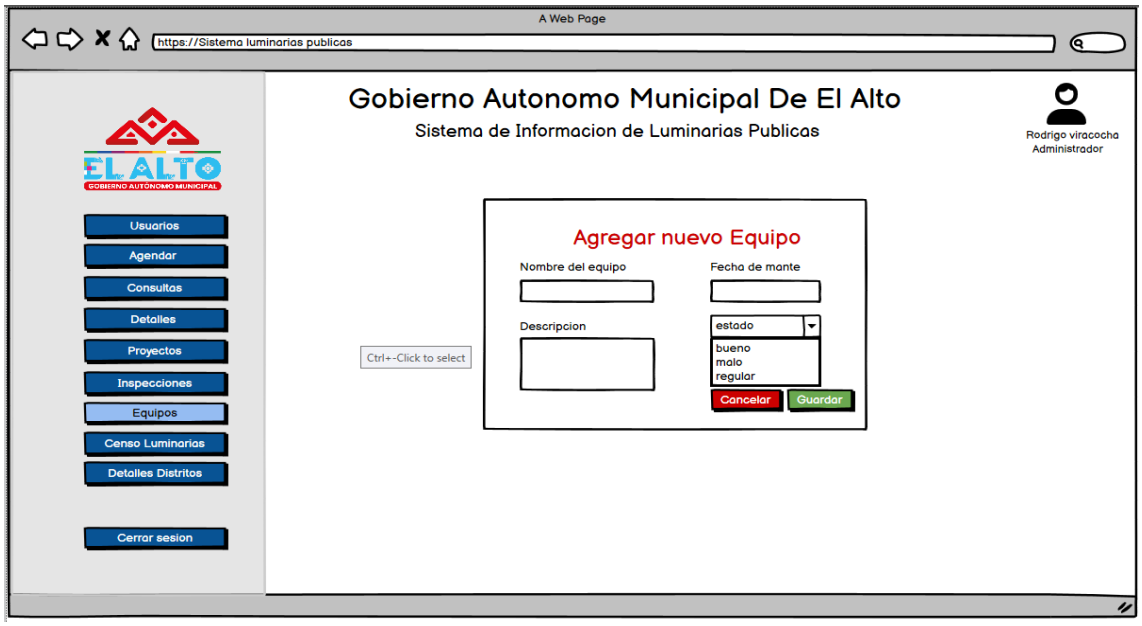


Figura *48.*

Modelo de Presentación - Urbanizaciones

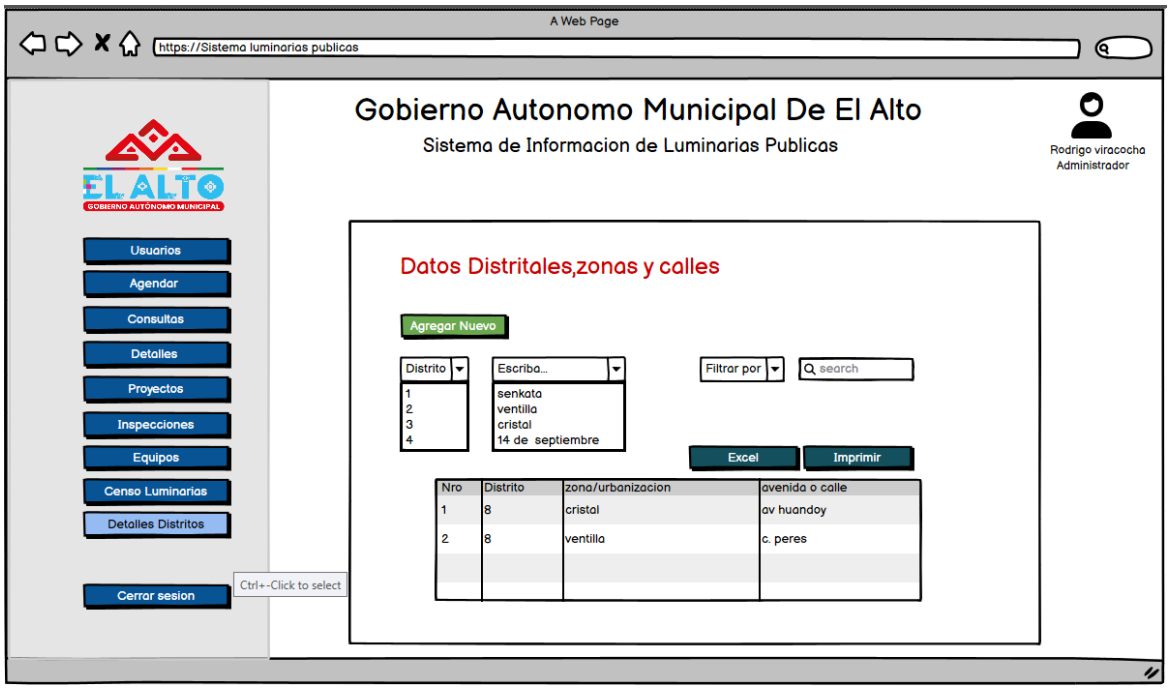


Figura *49.*

Modelo de Presentación - Registrar Urbanización

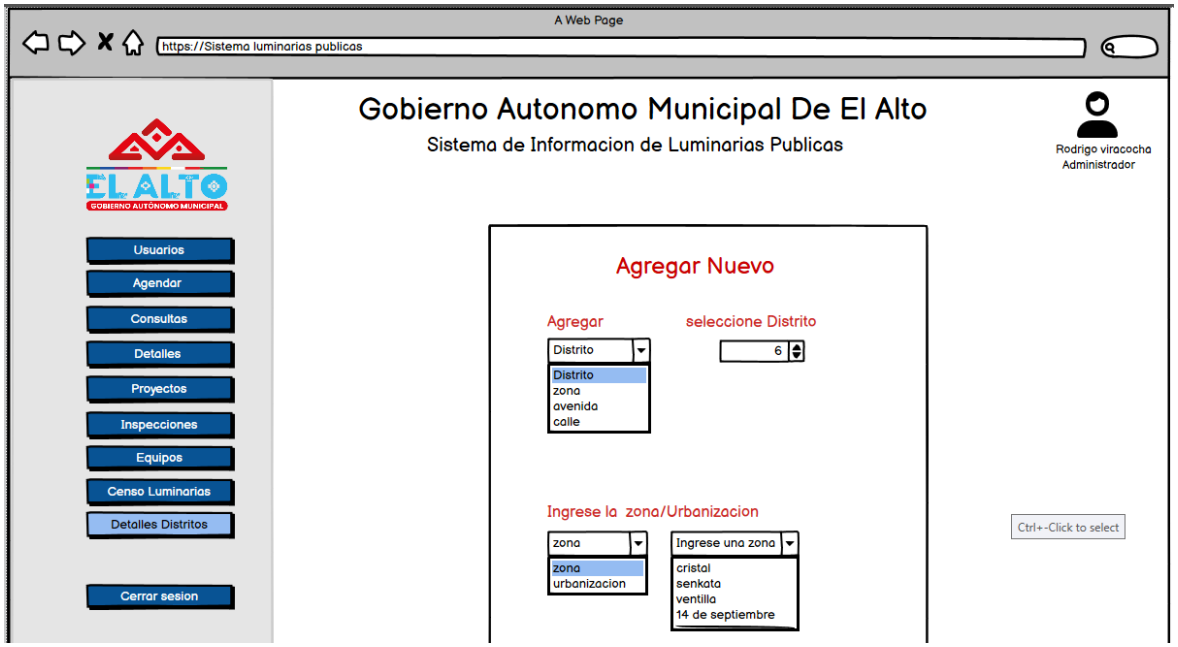
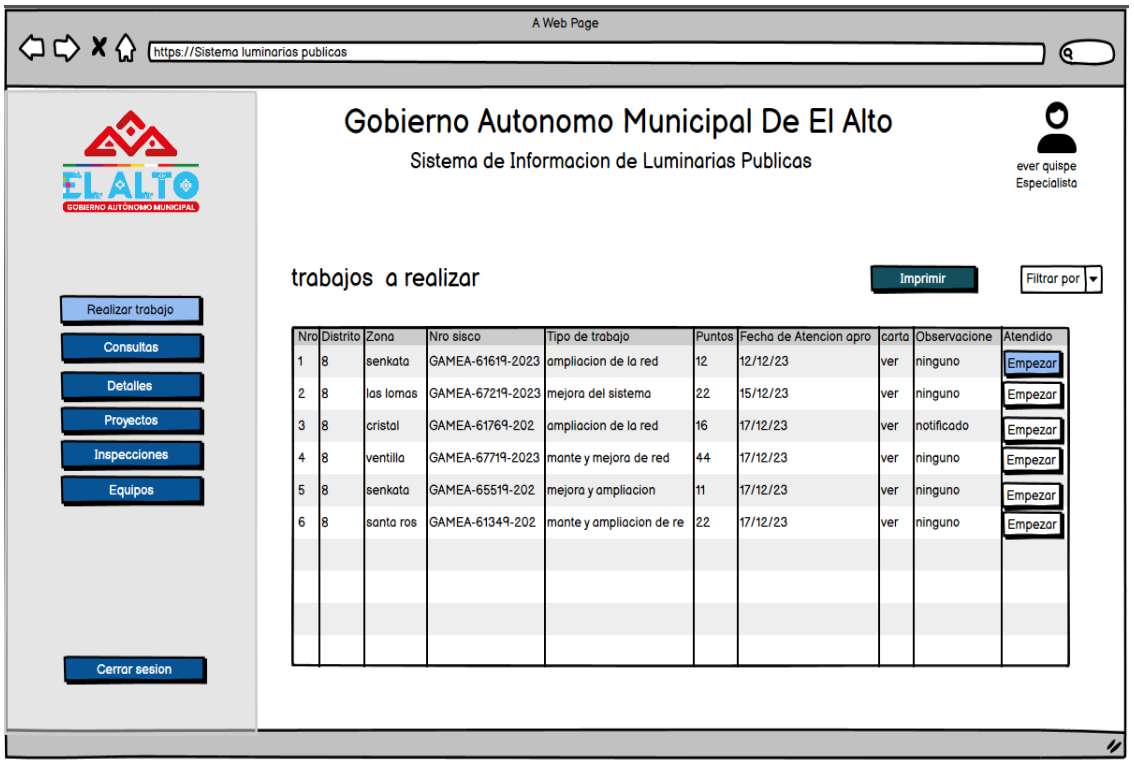


Figura *50.*

Modelo de Presentación - Realizar Trabajo



3.3. FASE DE CODIFICACION

En esta fase del desarrollo del **"**Sistema Web para el Registro, Seguimiento y Control Eficiente de Luminarias Públicas, Materiales y Herramientas**"**, me he enfocado en transformar los diseños y diagramas previos en código funcional. Utilizando tecnologías como PHP, HTML, CSS, JavaScript y el framework Laravel, he implementado cada uno de los módulos establecidos, garantizando que el sistema cumpla con los requisitos definidos en las fases anteriores. He estructurado el código de manera clara y modular, facilitando la escalabilidad y el mantenimiento futuro, y asegurando un rendimiento óptimo y una experiencia de usuario intuitiva.

Figura *51.*

Dashboard Actividades de Mantenimiento, Instalación y Proyectos

Figura *52.*

Módulo de Usuarios: Lista de Usuarios

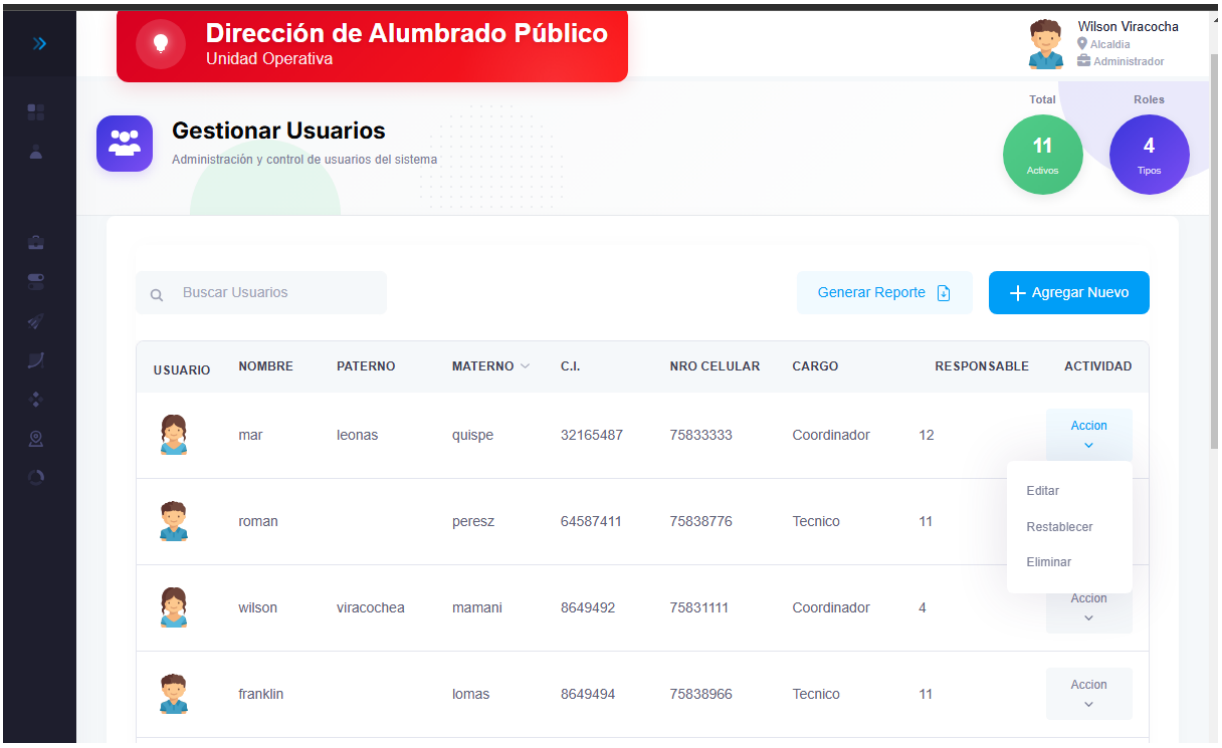


Figura *53.*

Módulo de Usuario: Registrar

Figura *54.*

Módulo de Usuario: Modificar

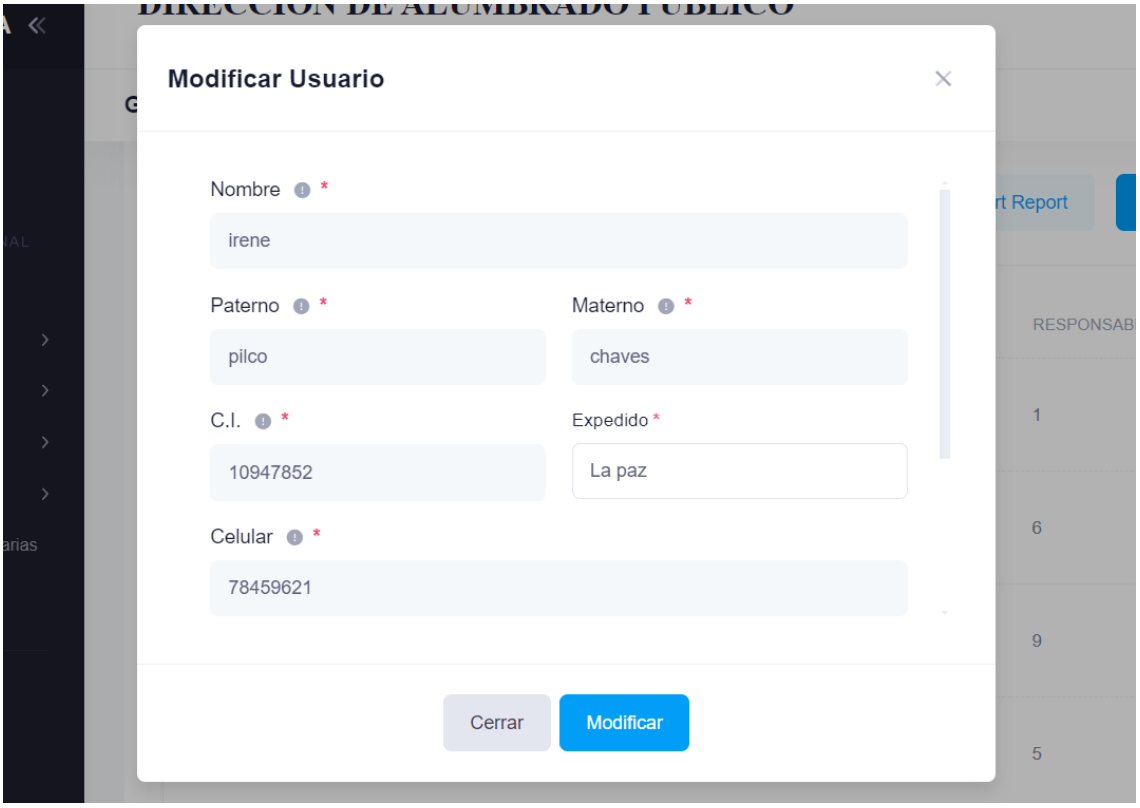


Figura *55.*

Módulo de Usuario: Eliminar

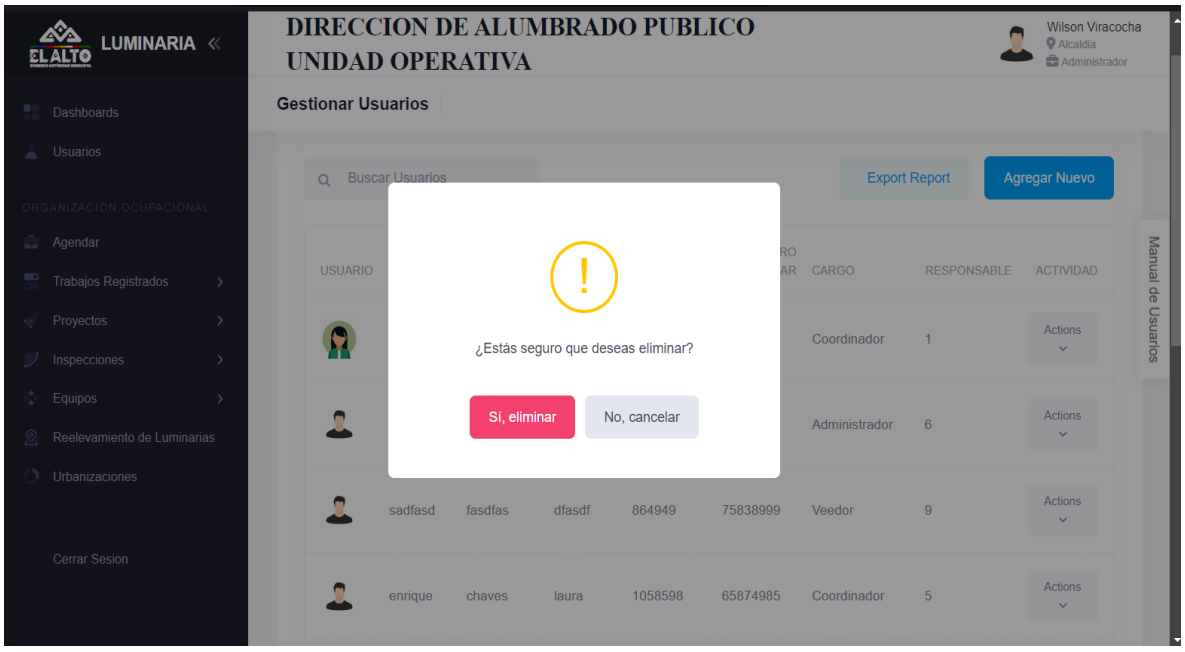


Figura *56.*

Módulo de Usuario: Restablecer Contraseña

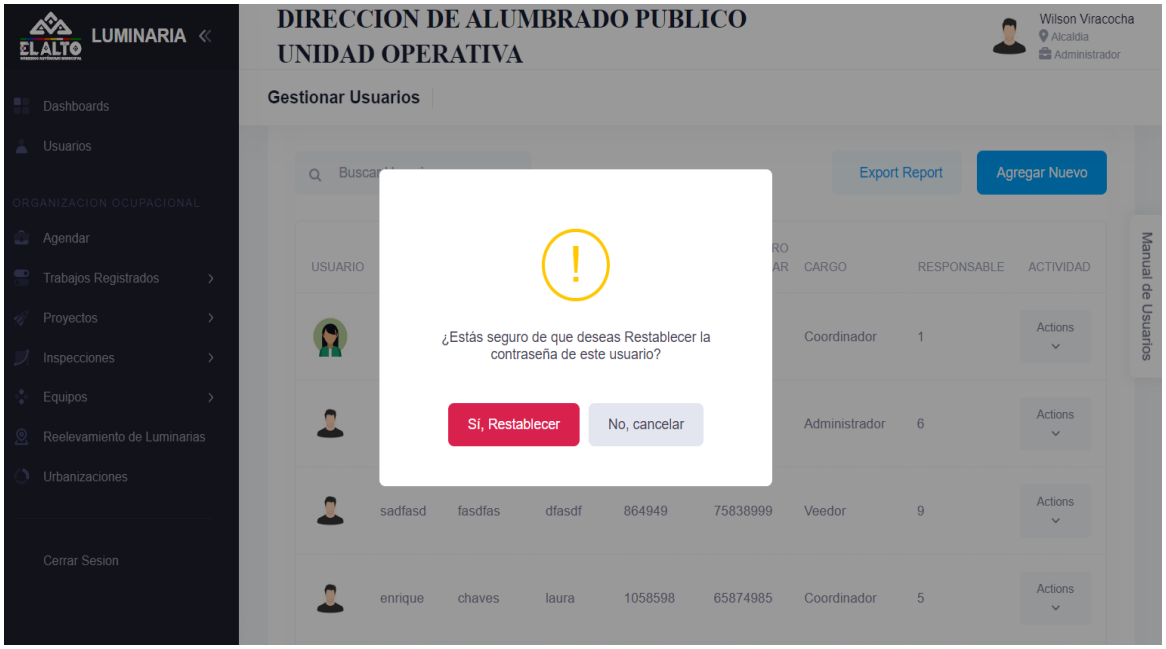


Figura *57.*

Módulo de Usuario: Buscar

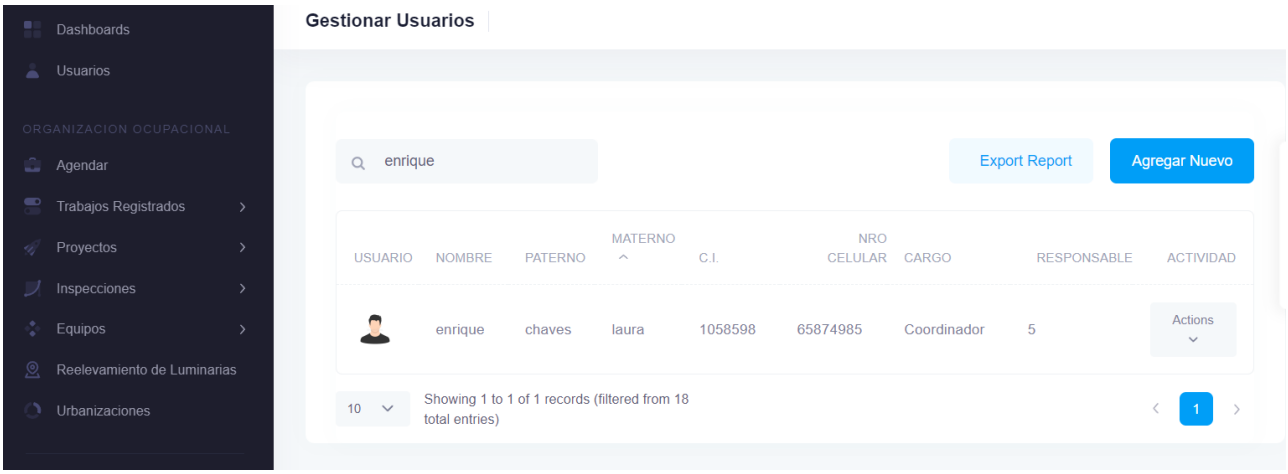
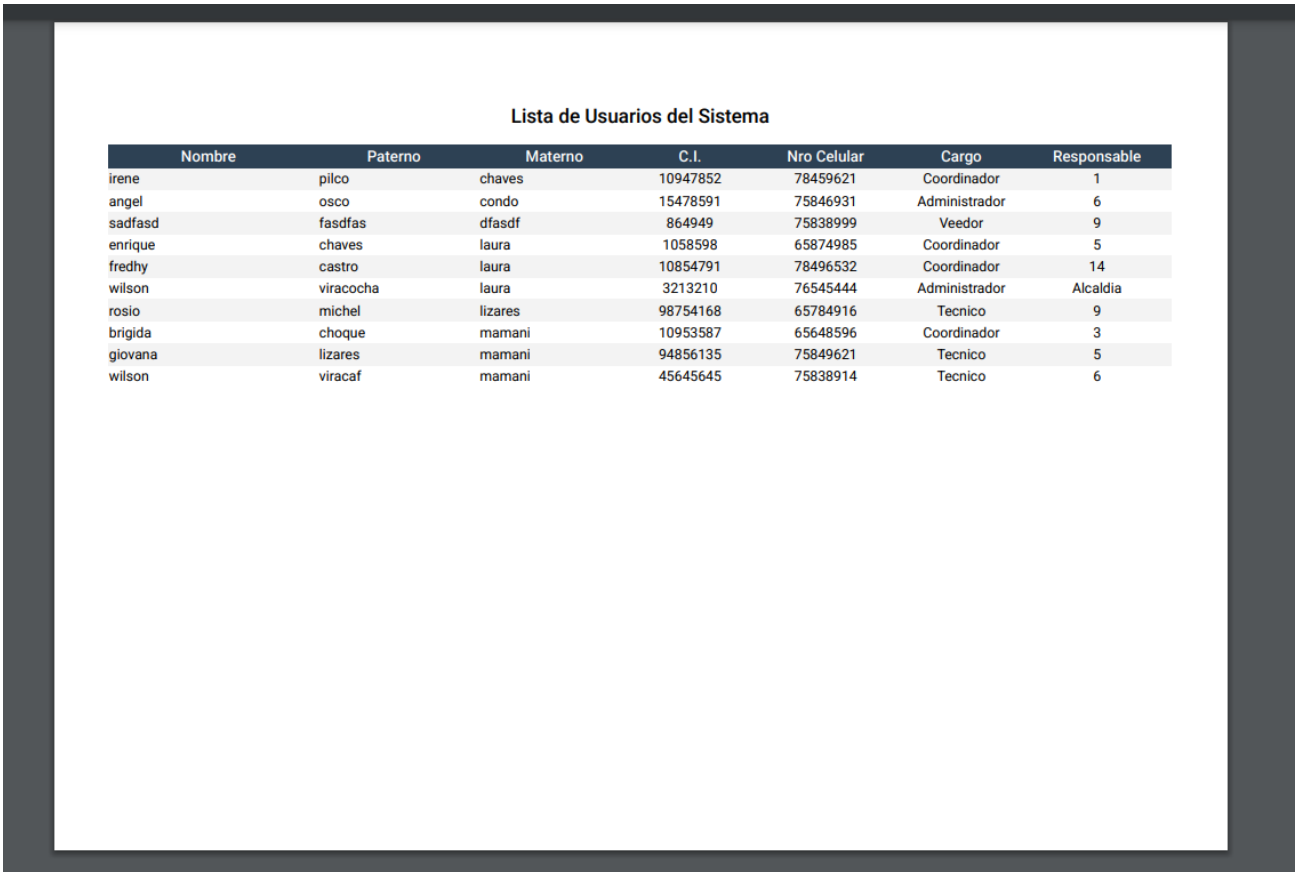


Figura *58.*

Módulo de Usuario: Exportar



Lista de Usuarios del Sistema

Nombre	Paterno	Materno	C.I.	Nro Celular	Cargo	Responsable
irene	pilco	chaves	10947852	78459621	Coordinador	1
angel	osco	condo	15478591	75846931	Administrador	6
sadfasd	fasdfas	dfasdf	864949	75838999	Veedor	9
enrique	chaves	laura	1058598	65874985	Coordinador	5
fredhy	castro	laura	10854791	78496532	Coordinador	14
wilson	viracocha	laura	3213210	76545444	Administrador	Alcaldia
rosio	michel	lizares	98754168	65784916	Tecnico	9
brigida	choque	mamani	10953587	65648596	Coordinador	3
giovana	lizares	mamani	94856135	75849621	Tecnico	5
wilson	viracaf	mamani	45645645	75838914	Tecnico	6

Figura *59.*

Agendar Trabajo para las Urbanizaciones de los diferentes distritos

Figura *60.*

Realizar Trabajo: Lista de trabajos pendientes del distrito

Figura *61.*

Realizar Trabajo: Buscar

Figura *62.*

Realizar Trabajo: Empezar Mantenimiento

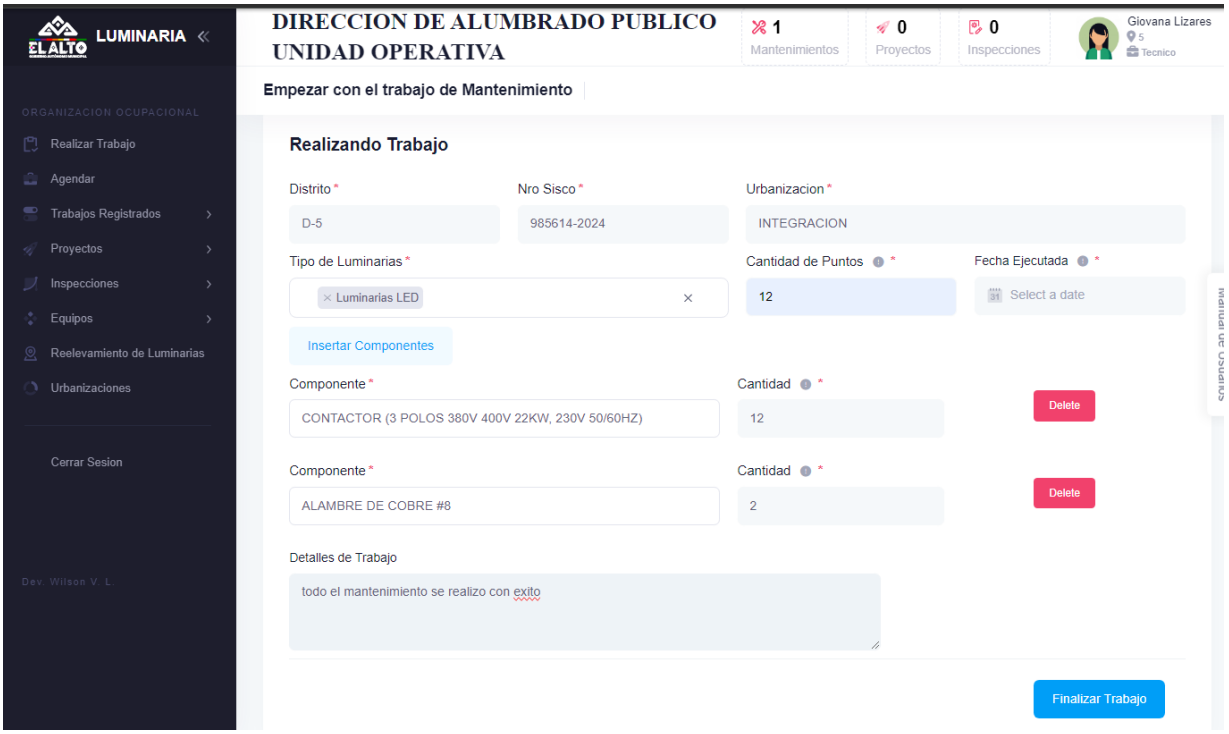


Figura *63.*

Módulo Trabajos Registrados: Lista de trabajos Pendientes

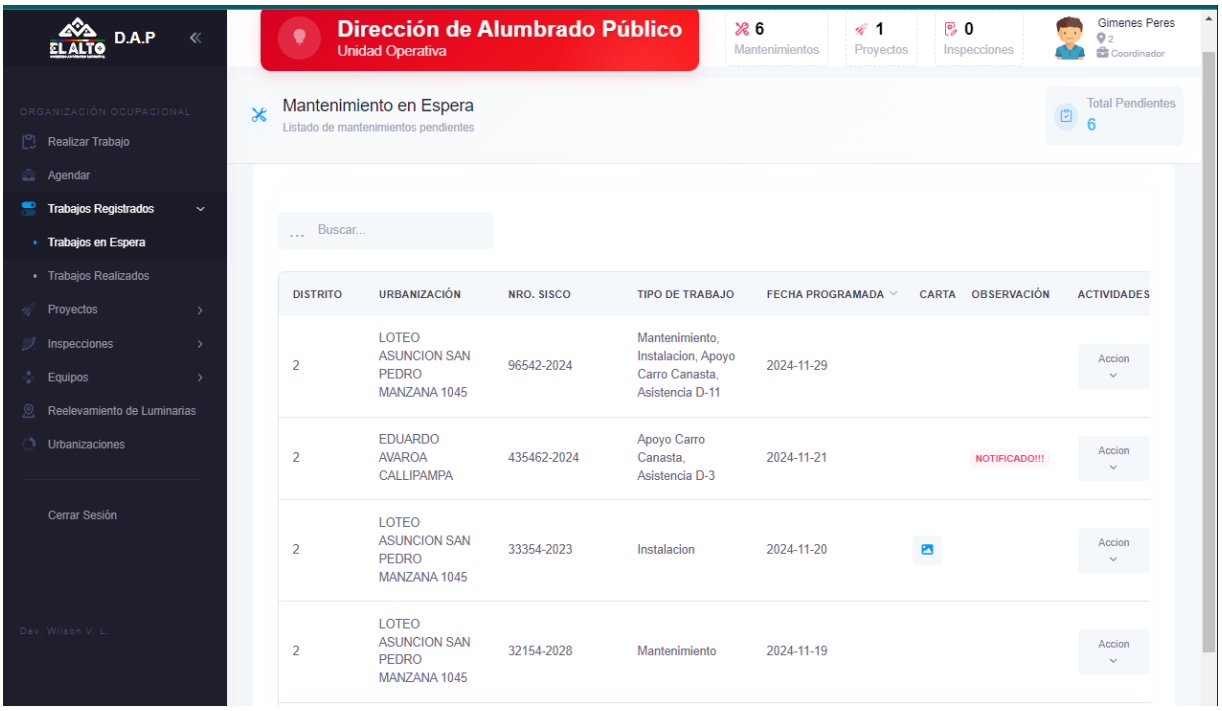


Figura *64.*

Módulo Trabajos Registrados: Modificar

Figura *65.*

Módulo Trabajos Registrados: Eliminar

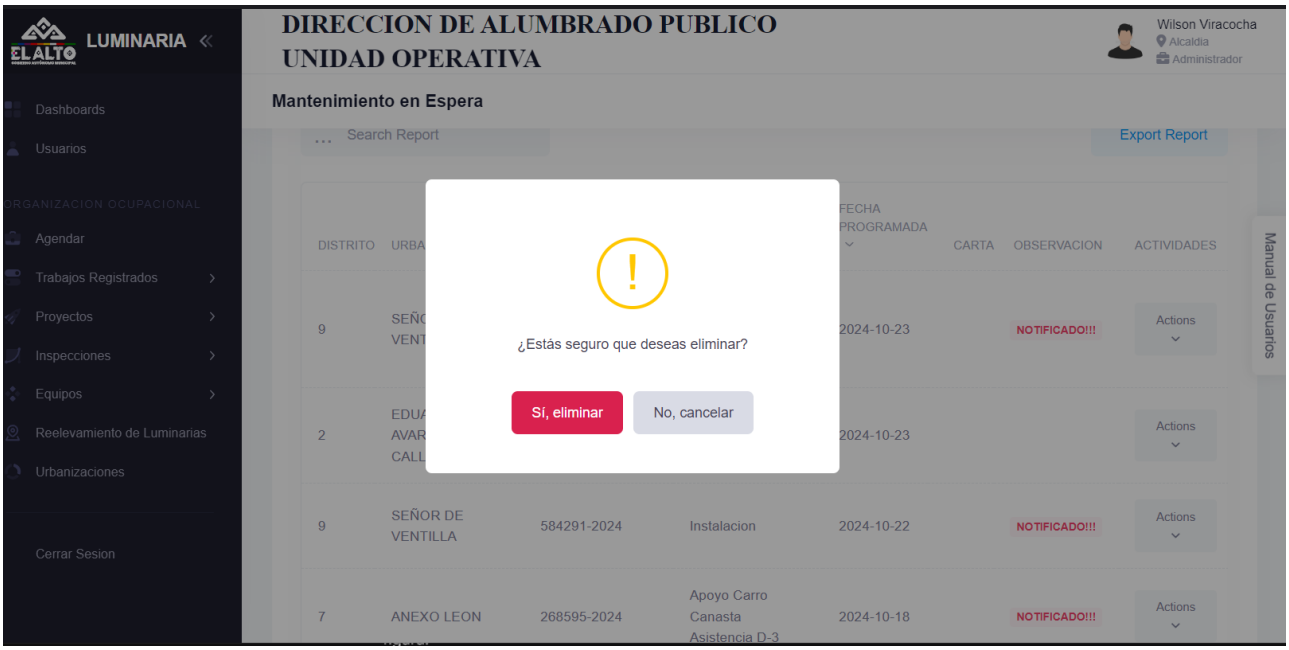
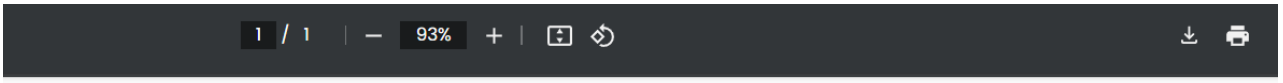


Figura *66.*

Módulo Trabajos Registrados: Exportar



Lista de Trabajos En Espera

Distrito	Urbanizacion	Nro Sisco	Tipo de Trabajo	Fecha Programada	Observacion
9	SEÑOR DE VENTILLA	987652-2024	Mantenimiento Instalacion Apoyo Carro Canasta Asistencia D-12	2024-10-23	NOTIFICADO!!!
2	EDUARDO AVAROA CALLIPAMPA	11114-2023	Instalacion	2024-10-23	
9	SEÑOR DE VENTILLA	584291-2024	Instalacion	2024-10-22	NOTIFICADO!!!
7	ANEXO LEON	268595-2024	Apoyo Carro Canasta Asistencia D-3	2024-10-18	NOTIFICADO!!!
8	RETAMAS 1	258964-2024	Mantenimiento, Instalacion	2024-10-17	NOTIFICADO!!!
2	COMPLEMENTO CUPILUPACA	65947-2023	Apoyo Carro Canasta	2024-10-17	NOTIFICADO!!!
13	VICTORIA A2	597413-2023	Mantenimiento Instalacion	2024-10-15	NOTIFICADO!!!
5	INTEGRACION	985614-2024	Mantenimiento Instalacion Apoyo Carro Canasta Asistencia D-2	2024-10-10	NOTIFICADO!!!
7	6 DE SEPTIEMBRE	259643-2023	Mantenimiento Instalacion	2024-10-10	NOTIFICADO!!!
8	UNION PUCARANI SENKATA	268793-2023	Instalacion	2024-10-10	NOTIFICADO!!!

Figura *67.*

Módulo Trabajos Registrados: Buscar

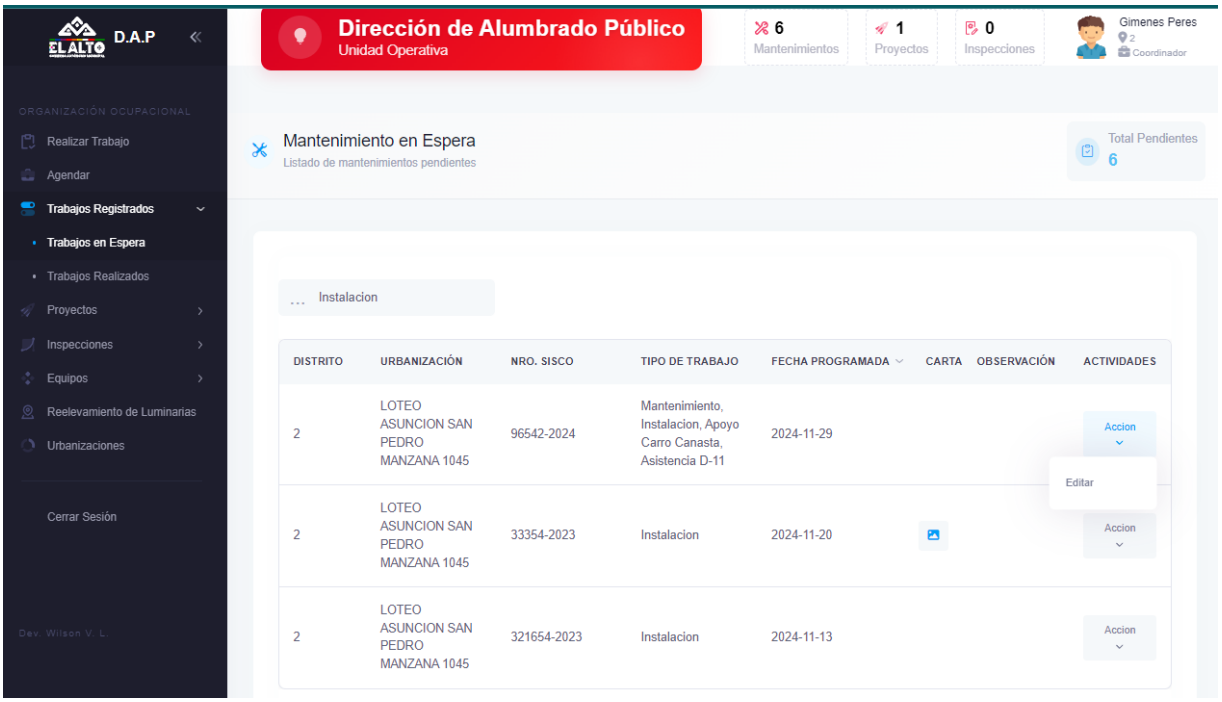


Figura *68.*

Módulo Trabajos Registrados Realizados: Listar

Figura *69.*

Módulo Trabajos Registrados Realizados: Modificar

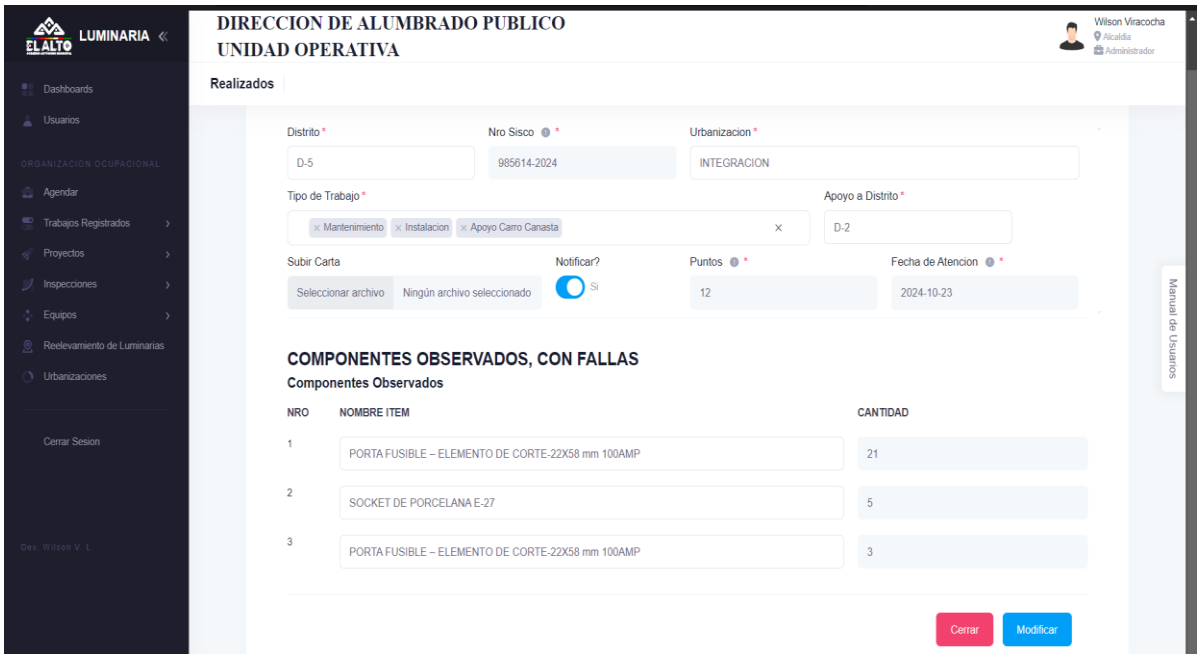


Figura *70.*

Módulo Trabajos Registrados Realizados: Detalles

Figura *71.*

Módulo Trabajos Registrados Realizados: Eliminar

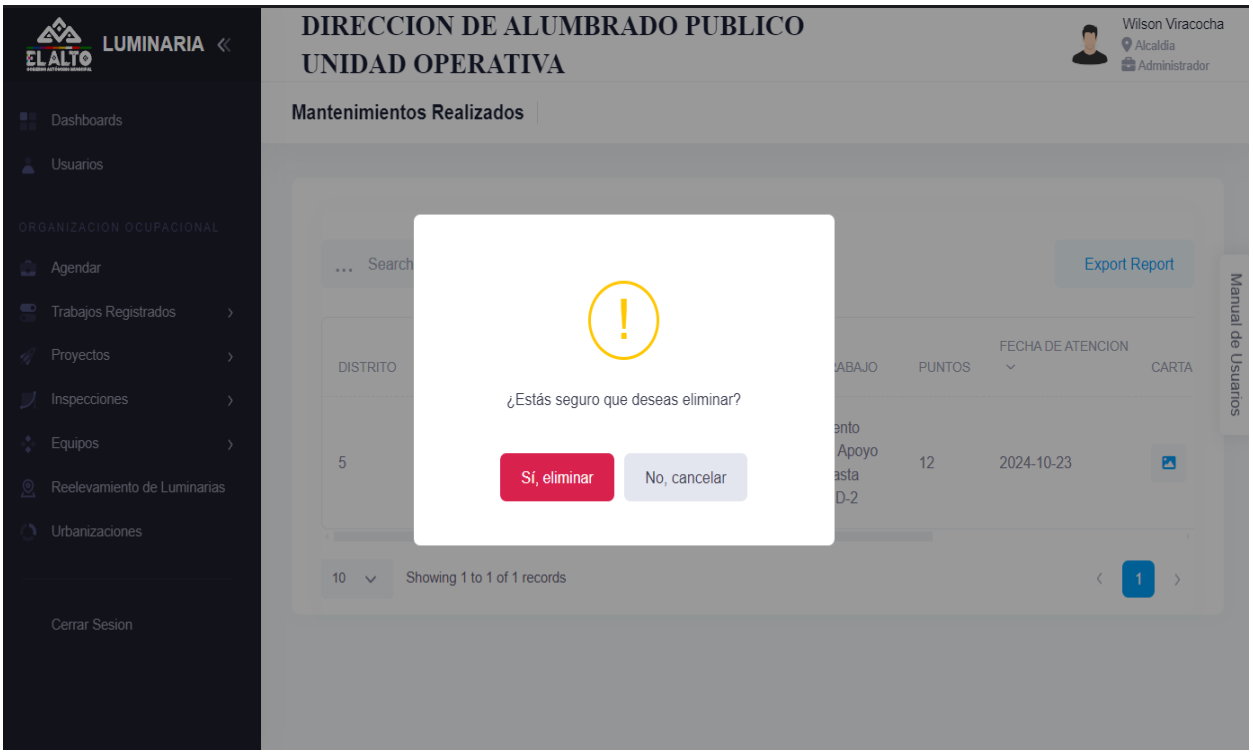
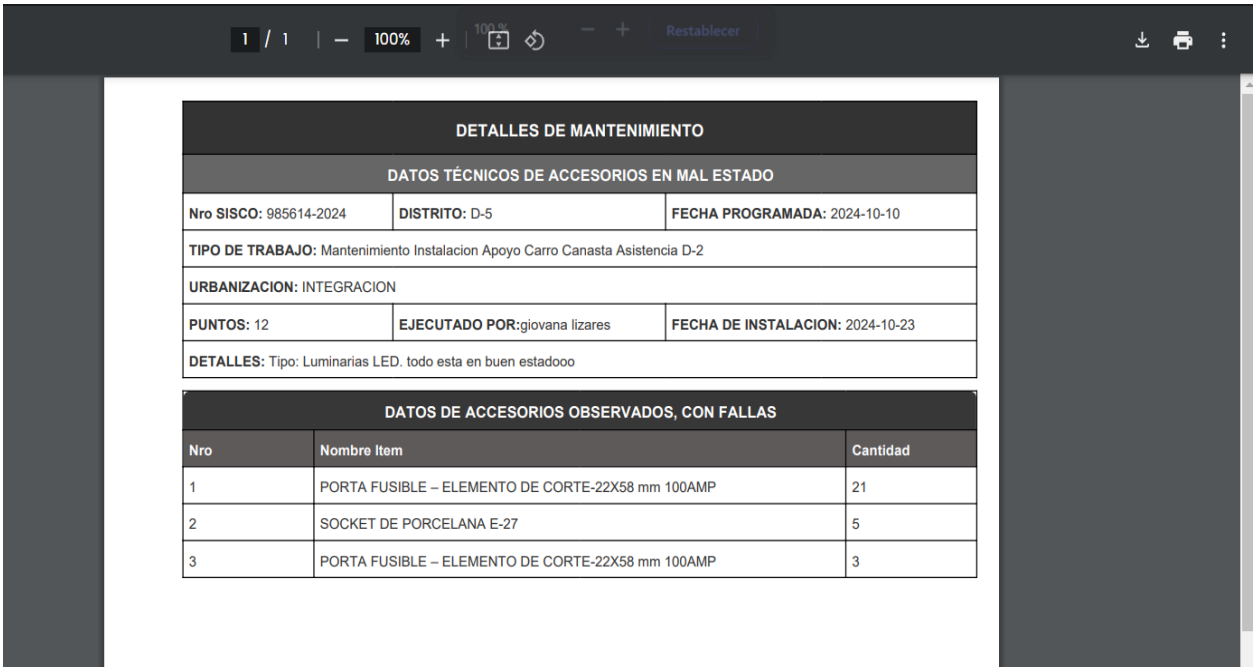


Figura *72.*

Módulo Trabajos Registrados Realizados: Reporte



DETALLES DE MANTENIMIENTO		
DATOS TÉCNICOS DE ACCESORIOS EN MAL ESTADO		
Nro SISCO: 985614-2024	**DISTRITO:** D-5	**FECHA PROGRAMADA:** 2024-10-10
TIPO DE TRABAJO: Mantenimiento Instalacion Apoyo Carro Canasta Asistencia D-2		
URBANIZACION: INTEGRACION		
PUNTOS: 12	**EJECUTADO POR:**giovana lizares	**FECHA DE INSTALACION:** 2024-10-23
DETALLES: Tipo: Luminarias LED. todo esta en buen estadooo		

DATOS DE ACCESORIOS OBSERVADOS, CON FALLAS		
Nro	**Nombre Item**	**Cantidad**
1	PORTA FUSIBLE – ELEMENTO DE CORTE-22X58 mm 100AMP	21
2	SOCKET DE PORCELANA E-27	5
3	PORTA FUSIBLE – ELEMENTO DE CORTE-22X58 mm 100AMP	3

Figura *73.*

Módulo Trabajos Registrados Realizados: Buscar

Figura *74.*

Módulo Trabajos Registrados Realizados: Exporta Lista de Trabajos

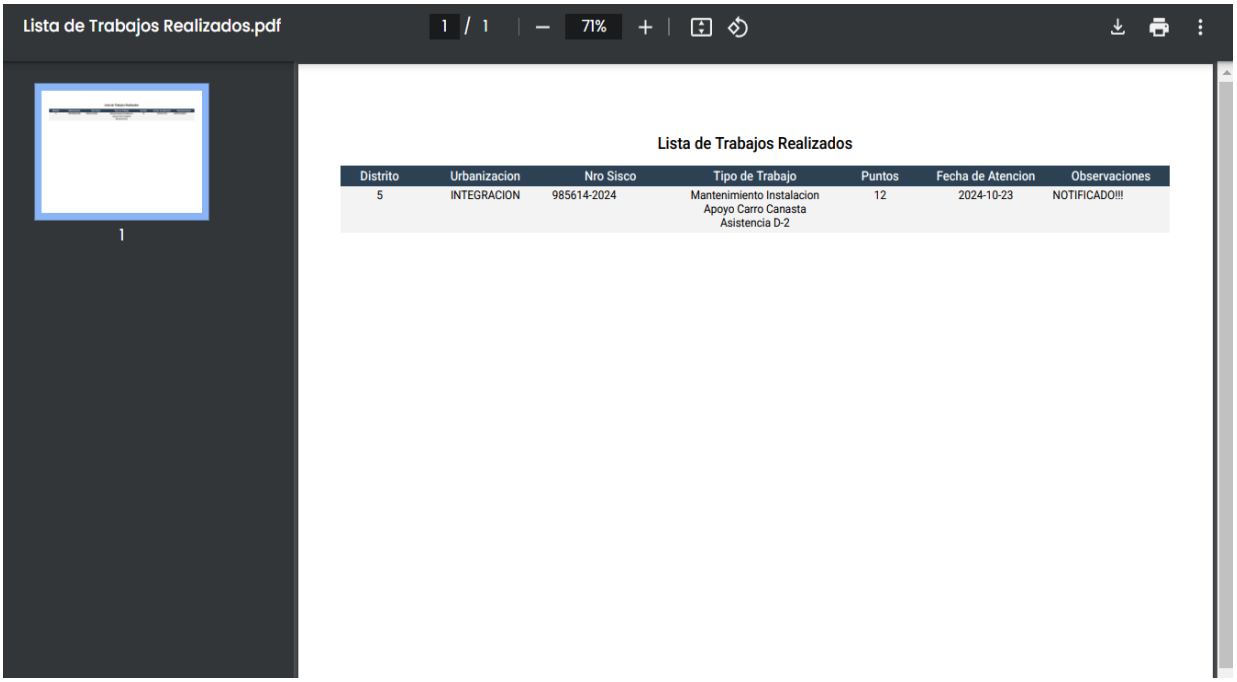


Figura *75.*

Módulo Proyectos: Listar

Figura *76.*

Módulo Proyectos: Empezar Instalación

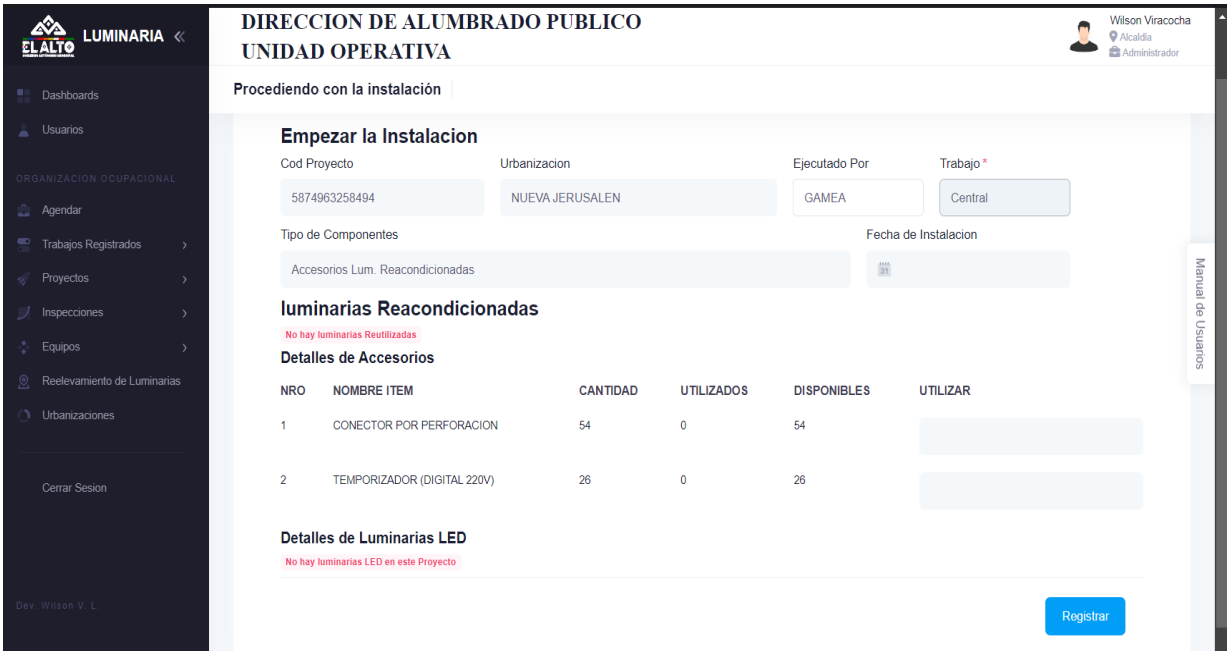


Figura *77.*

Módulo de Proyecto: Registrar

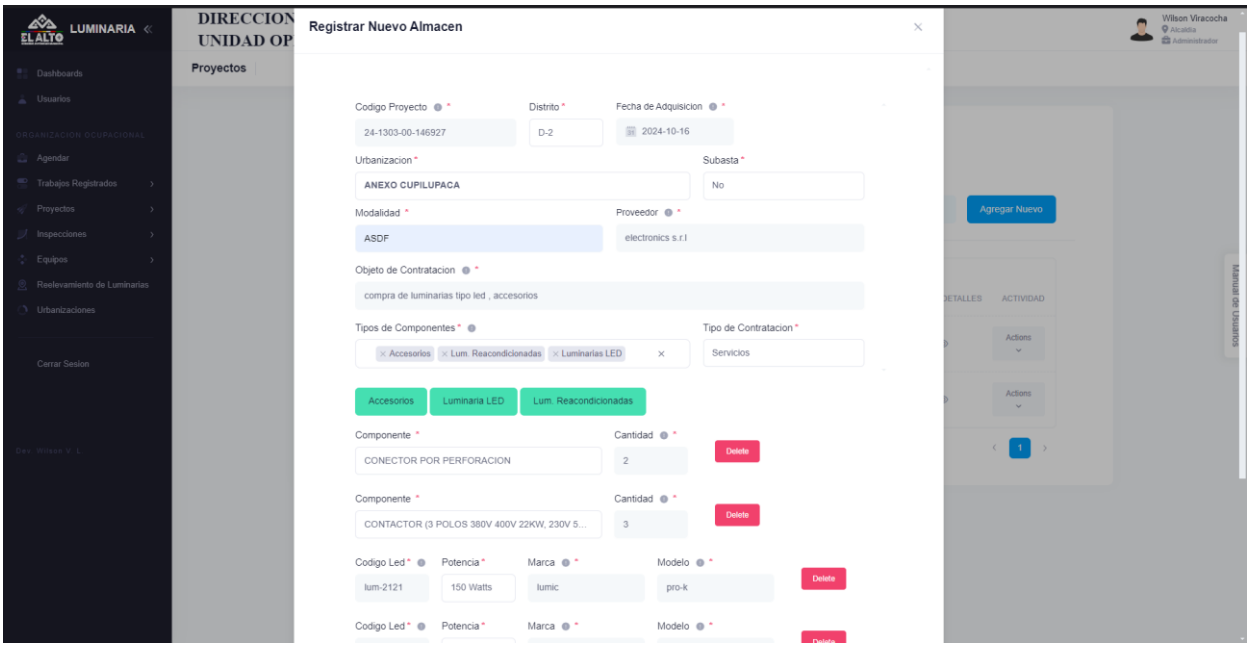


Figura *78.*

Módulo Proyectos: Modificar

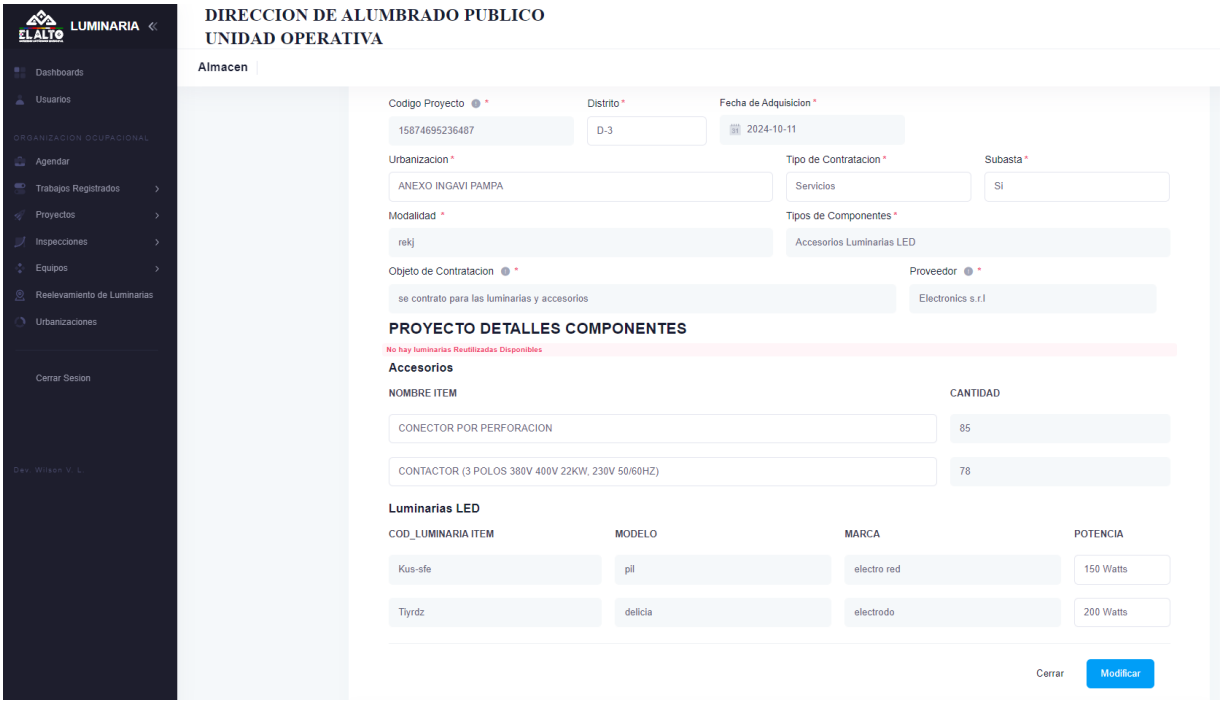
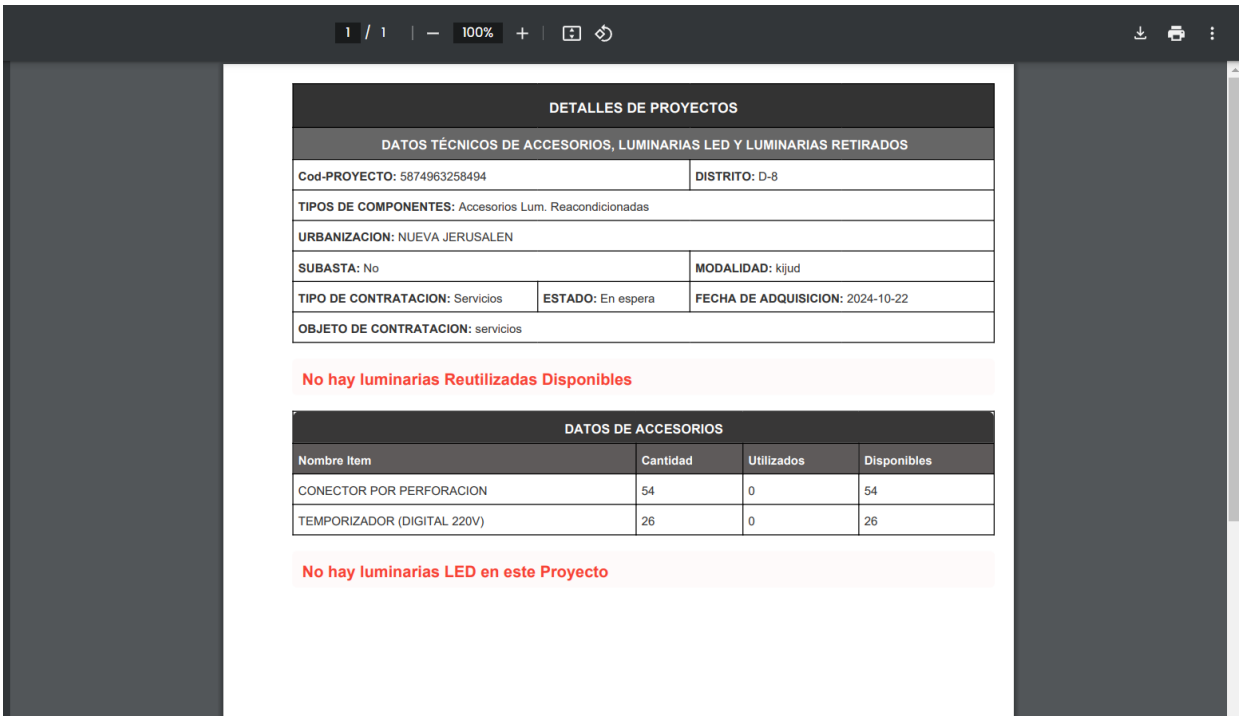


Figura *79.*

Módulo Proyectos: Reporte



DETALLES DE PROYECTOS

DATOS TÉCNICOS DE ACCESORIOS, LUMINARIAS LED Y LUMINARIAS RETIRADOS

Cod-PROYECTO: 5874963258494		**DISTRITO:** D-8
TIPOS DE COMPONENTES: Accesorios Lum. Reacondicionadas		
URBANIZACION: NUEVA JERUSALEN		
SUBASTA: No		**MODALIDAD:** kijud
TIPO DE CONTRATACION: Servicios	**ESTADO:** En espera	**FECHA DE ADQUISICION:** 2024-10-22
OBJETO DE CONTRATACION: servicios		

No hay luminarias Reutilizadas Disponibles

DATOS DE ACCESORIOS

Nombre Item	Cantidad	Utilizados	Disponibles
CONECTOR POR PERFORACION	54	0	54
TEMPORIZADOR (DIGITAL 220V)	26	0	26

No hay luminarias LED en este Proyecto

Figura *80.*

Módulo Proyectos: Eliminar

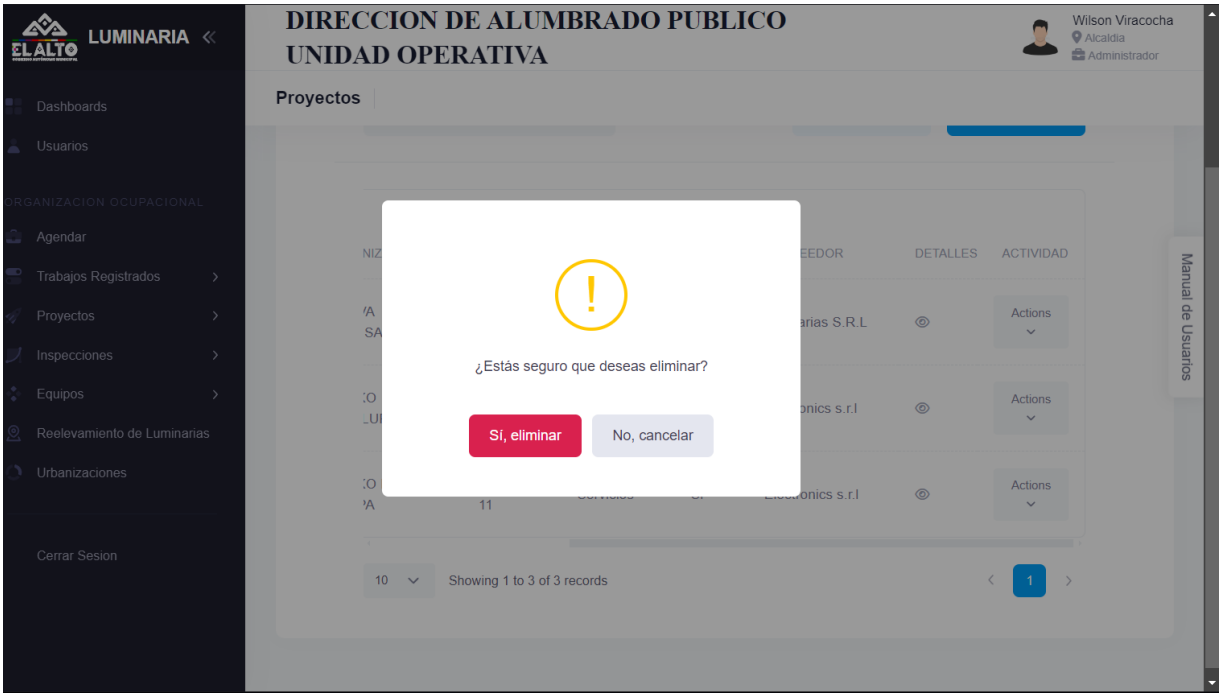


Figura *81.*

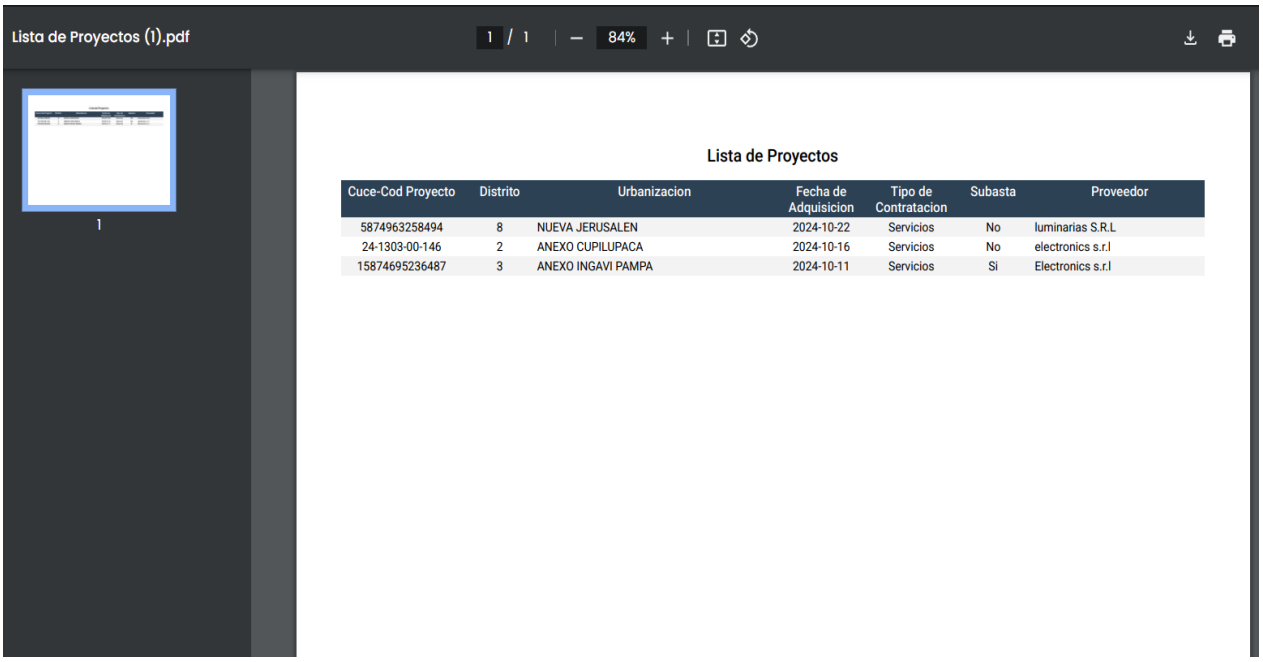
Módulo Proyectos: Detalles

Figura *82.*

Módulo Proyectos: Buscar

Figura *83.*

Módulo Proyectos: Reporte General



Cuce-Cod Proyecto	Distrito	Urbanizacion	Fecha de Adquisicion	Tipo de Contratacion	Subasta	Proveedor
5874963258494	8	NUEVA JERUSALEN	2024-10-22	Servicios	No	luminarias S.R.L
24-1303-00-146	2	ANEXO CUPILUPACA	2024-10-16	Servicios	No	electronics s.r.l
15874695236487	3	ANEXO INGAVI PAMPA	2024-10-11	Servicios	Si	Electronics s.r.l

Figura *84.*

Módulo Proyectos Obras Ejecutadas: Listar

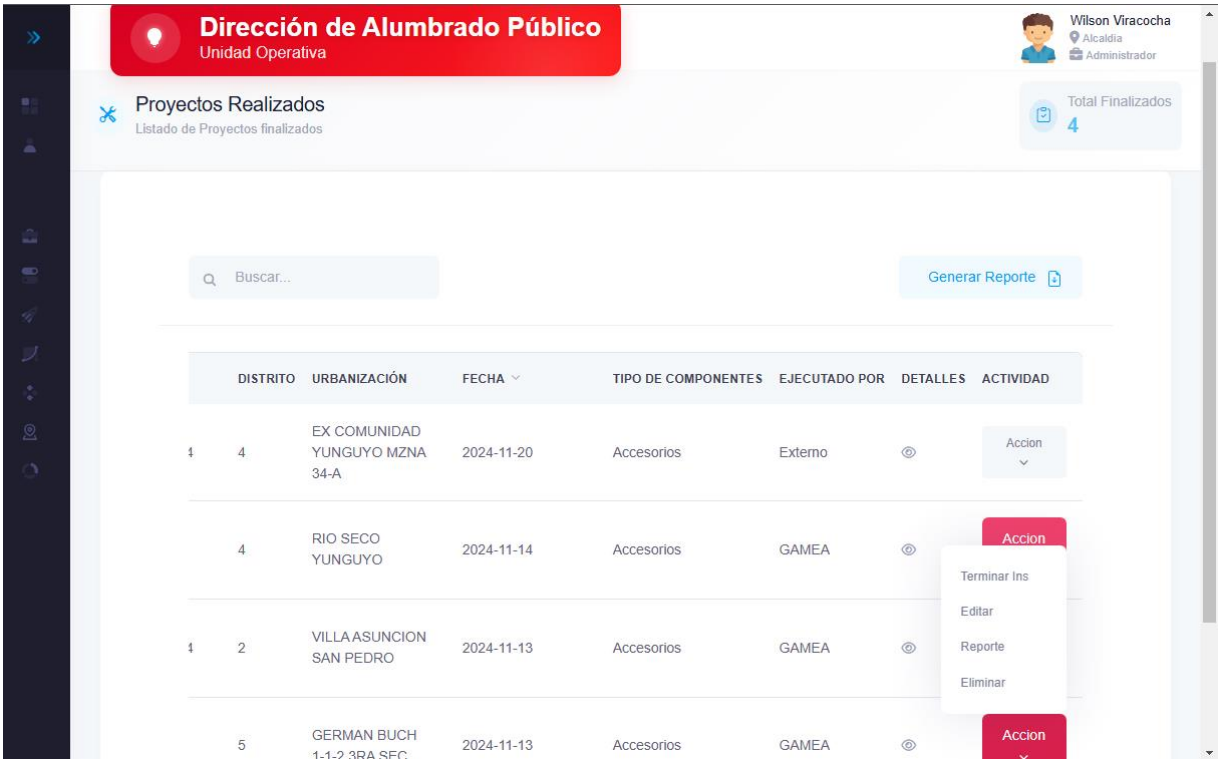


Figura *85.*

Módulo Proyectos Obras Ejecutadas: Modificar

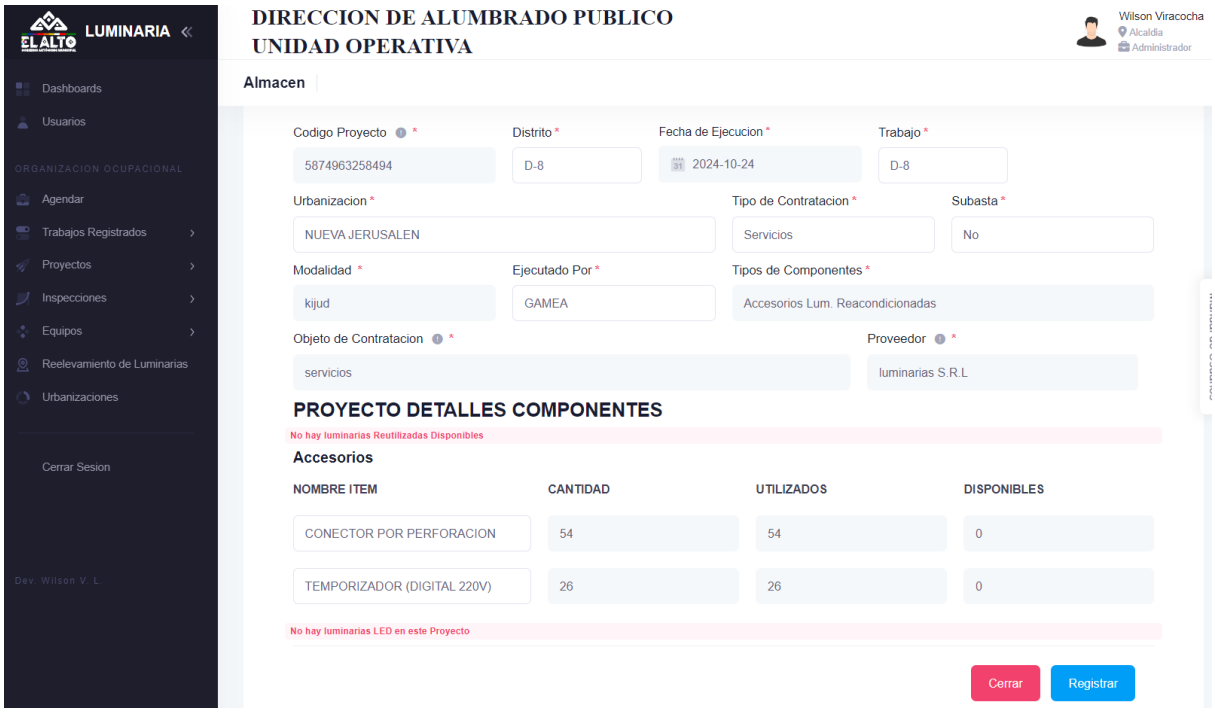


Figura *86.*

Módulo Proyectos Obras Ejecutadas: Eliminar

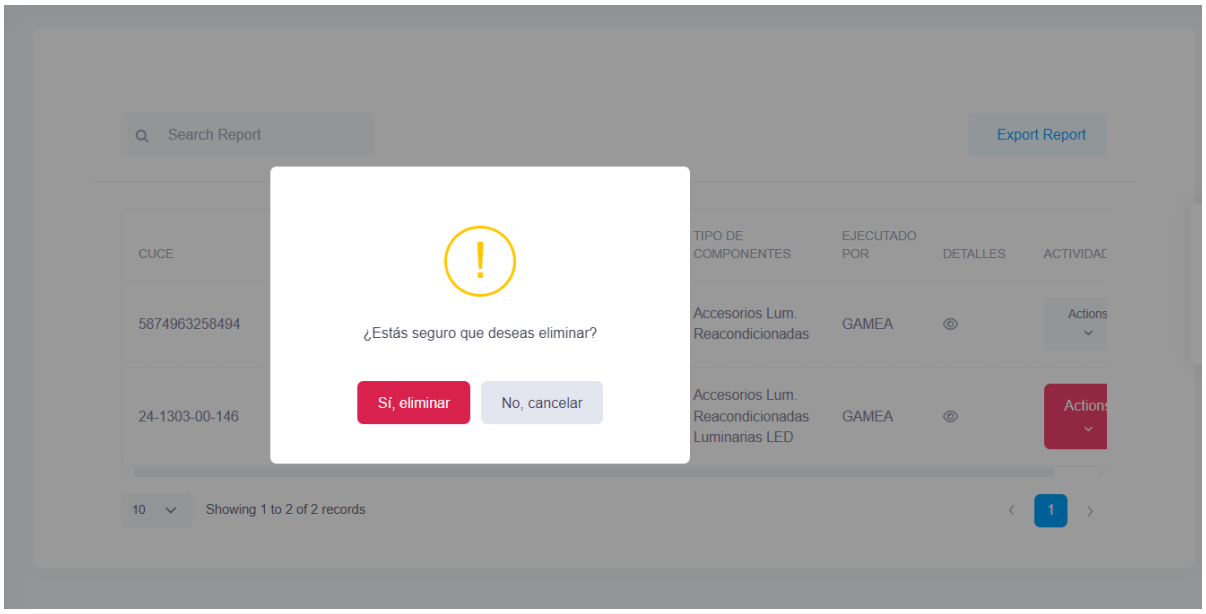
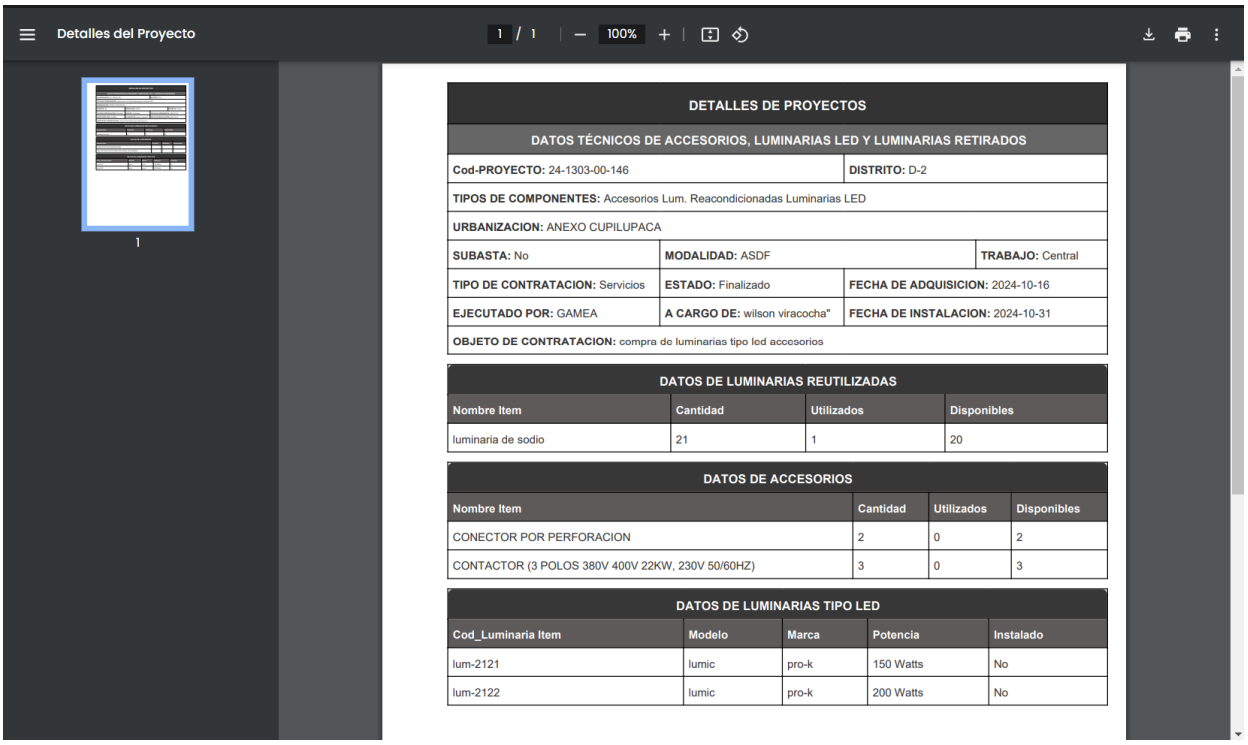


Figura *87.*

Módulo Proyectos Obras Ejecutadas: Reporte



Detalles del Proyecto

1 / 1 100%

1

DETALLES DE PROYECTOS		
DATOS TÉCNICOS DE ACCESORIOS, LUMINARIAS LED Y LUMINARIAS RETIRADOS		
Cod-PROYECTO: 24-1303-00-146		**DISTRITO:** D-2
TIPOS DE COMPONENTES: Accesorios Lum. Reacondicionadas Luminarias LED		
URBANIZACION: ANEXO CUPILUPACA		
SUBASTA: No	**MODALIDAD:** ASDF	**TRABAJO:** Central
TIPO DE CONTRATACION: Servicios	**ESTADO:** Finalizado	**FECHA DE ADQUISICION:** 2024-10-16
EJECUTADO POR: GAMEA	**A CARGO DE:** wilson viracocha"	**FECHA DE INSTALACION:** 2024-10-31
OBJETO DE CONTRATACION: compra de luminarias tipo led accesorios		

DATOS DE LUMINARIAS REUTILIZADAS

Nombre Item	Cantidad	Utilizados	Disponibles
luminaria de sodio	21	1	20

DATOS DE ACCESORIOS

Nombre Item	Cantidad	Utilizados	Disponibles
CONECTOR POR PERFORACION	2	0	2
CONTACTOR (3 POLOS 380V 400V 22KW, 230V 50/60HZ)	3	0	3

DATOS DE LUMINARIAS TIPO LED

Cod_Luminaria Item	Modelo	Marca	Potencia	Instalado
lum-2121	lumic	pro-k	150 Watts	No
lum-2122	lumic	pro-k	200 Watts	No

Figura *88.*

Módulo Proyectos Obras Ejecutadas: Detalles

LUMINARIA

Dashboards
Usuarios
ORGANIZACION OCUPACIONAL
Agendar
Trabajos Registrados
Proyectos
Inspecciones
Equipos
Reelevamiento de Luminarias
Urbanizaciones
Cerrar Sesion
Dev. Wilson V. L.

DIRECCION DE ALUMBRADO PUBLICO
UNIDAD OPERATIVA

Wilson Viracocha
Alcaldia
Administrador

Detalles Proyecto

Manual de Usuarios

Codigo Proyecto	Distrito	Tipos de Componentes
24-1303-00-146	D-2	Accesorios Lum. Reacondicionadas Luminarias LED

Urbanizacion	Tipo de Contratacion	Subasta
ANEXO CUPILUPACA	Servicios	No

Modalidad	Trabajo	Ejecutado Por	Fecha de Ejecucion *
ASDF	Central	GAMEA	31/10/2024

Objeto de Contratacion	Ejecutado
compra de luminarias tipo led accesorios	wilson viracocha

PROYECTO DETALLES COMPONENTES

luminarias Reutilizadas

NOMBRE ITEM	CANTIDAD	UTILIZADOS	DISPONIBLES
luminaria de sodio	21	1	20

Accesorios

NOMBRE ITEM	CANTIDAD	UTILIZADOS	DISPONIBLES
CONECTOR POR PERFORACION	2	0	2
CONTACTOR (3 POLOS 380V 400V 22KW, 230V 50/60HZ)	3	0	3

Luminarias LED

Figura *89.*

Módulo Proyectos Obras Ejecutadas: Buscar

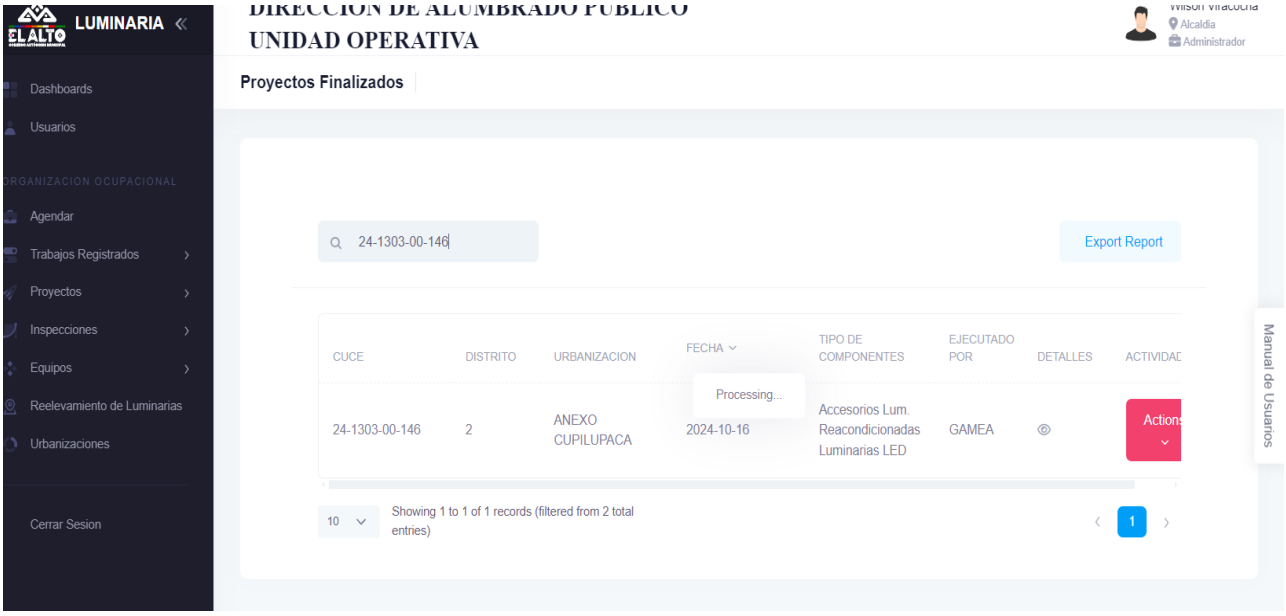


Figura *90.*

Módulo Proyectos Obras Ejecutadas: Reporte General

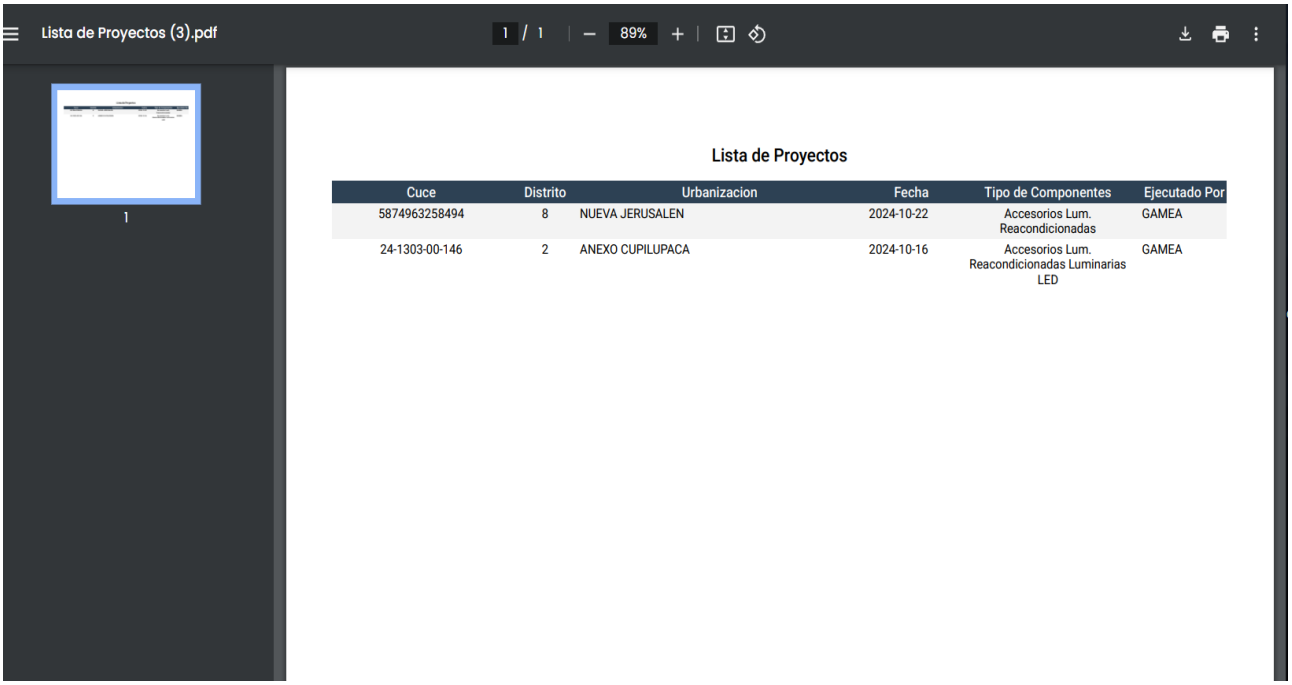


Figura *91.*

Módulo Accesorios y Luminarias Retiradas: Listar

Figura *92.*

Módulo Accesorios y Luminarias Retiradas: Registrar

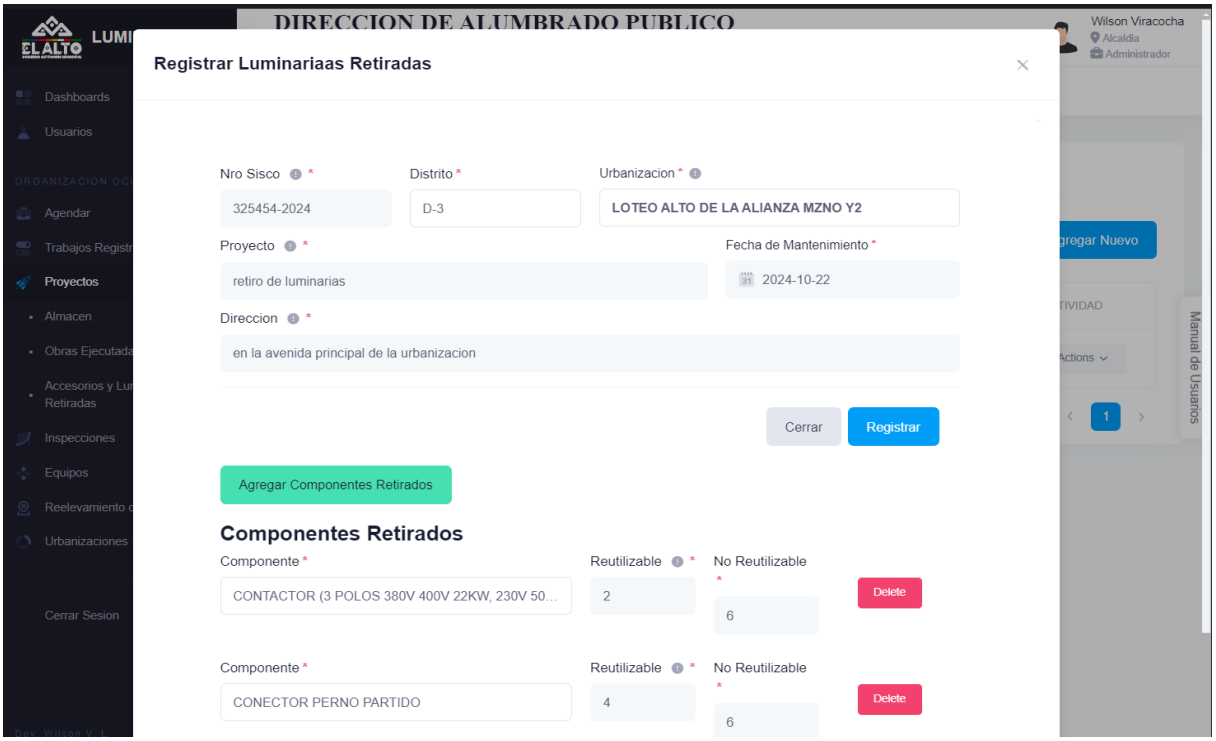


Figura *93.*

Módulo Accesorios y Luminarias Retiradas: Modificar

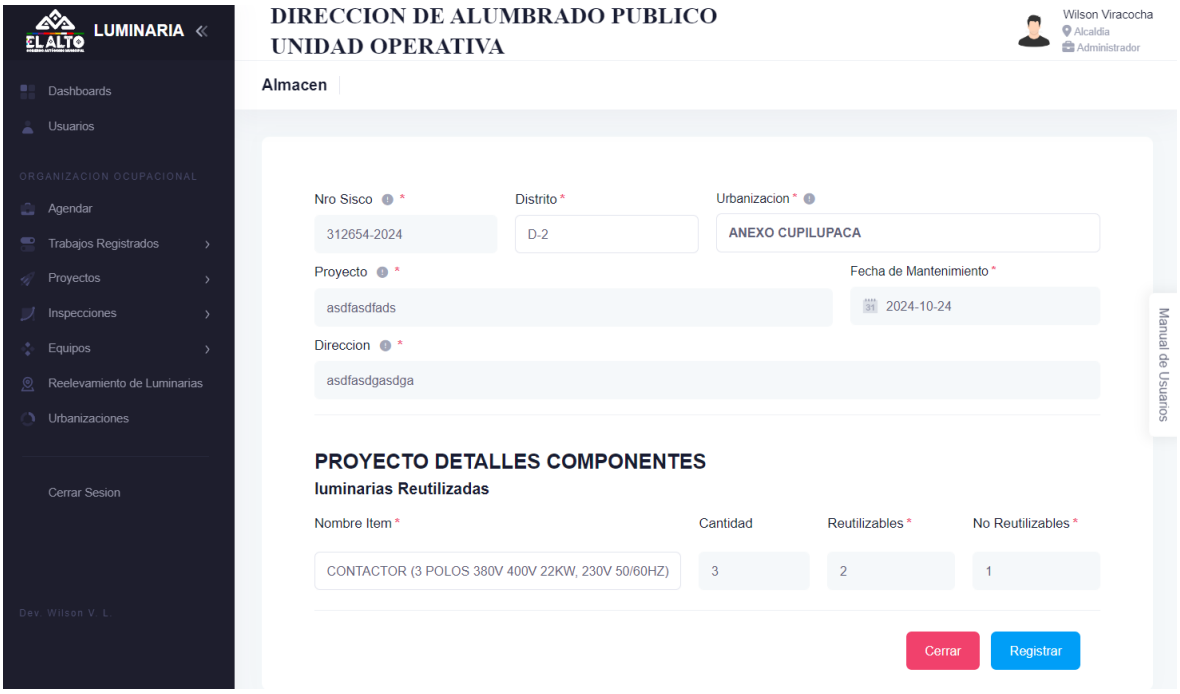


Figura *94.*

Módulo Accesorios y Luminarias Retiradas: Eliminar

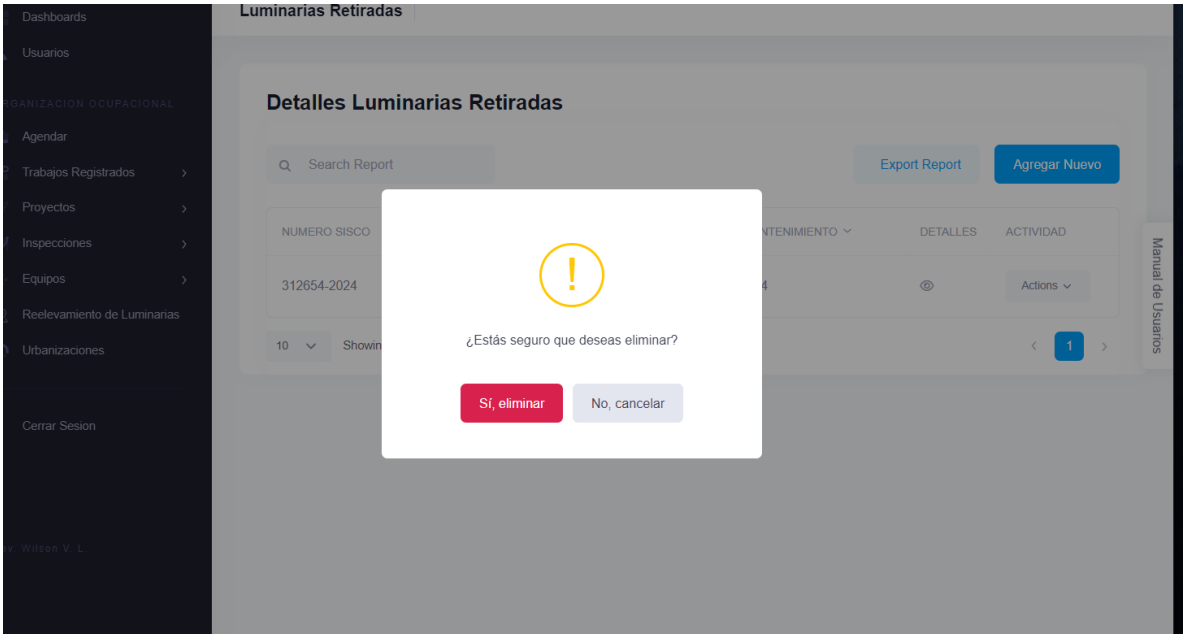
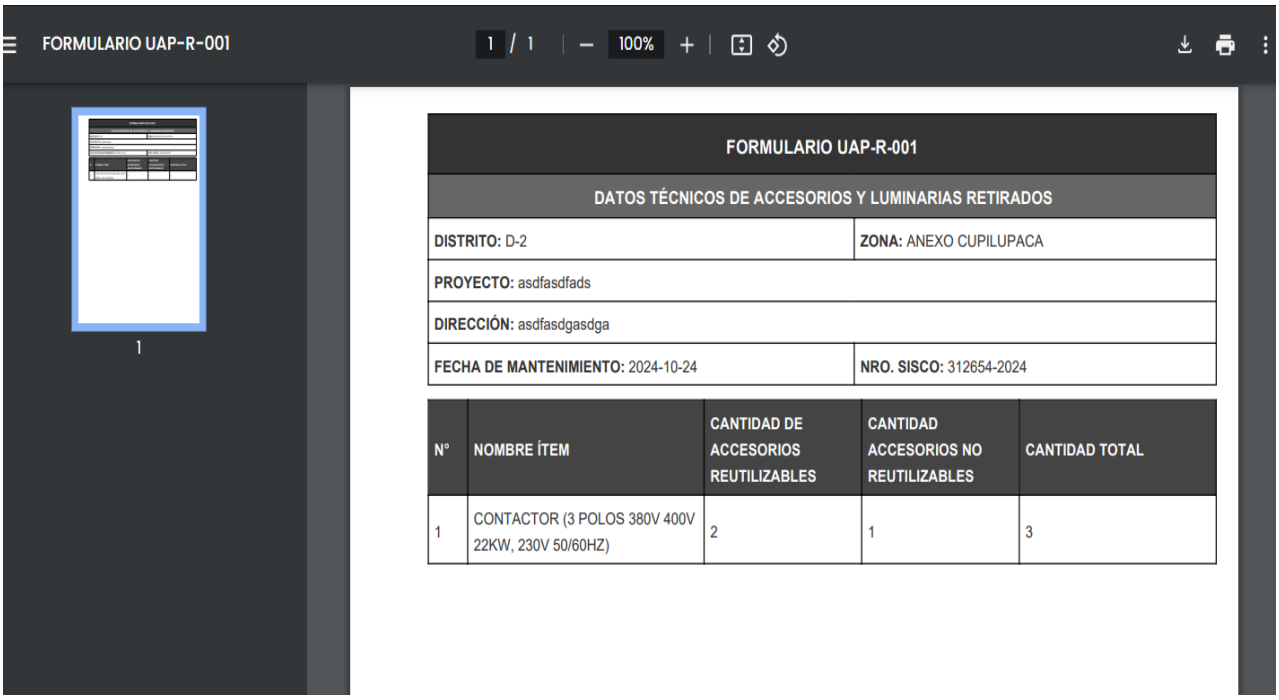


Figura *95.*

Módulo Accesorios y Luminarias Retiradas: Reporte



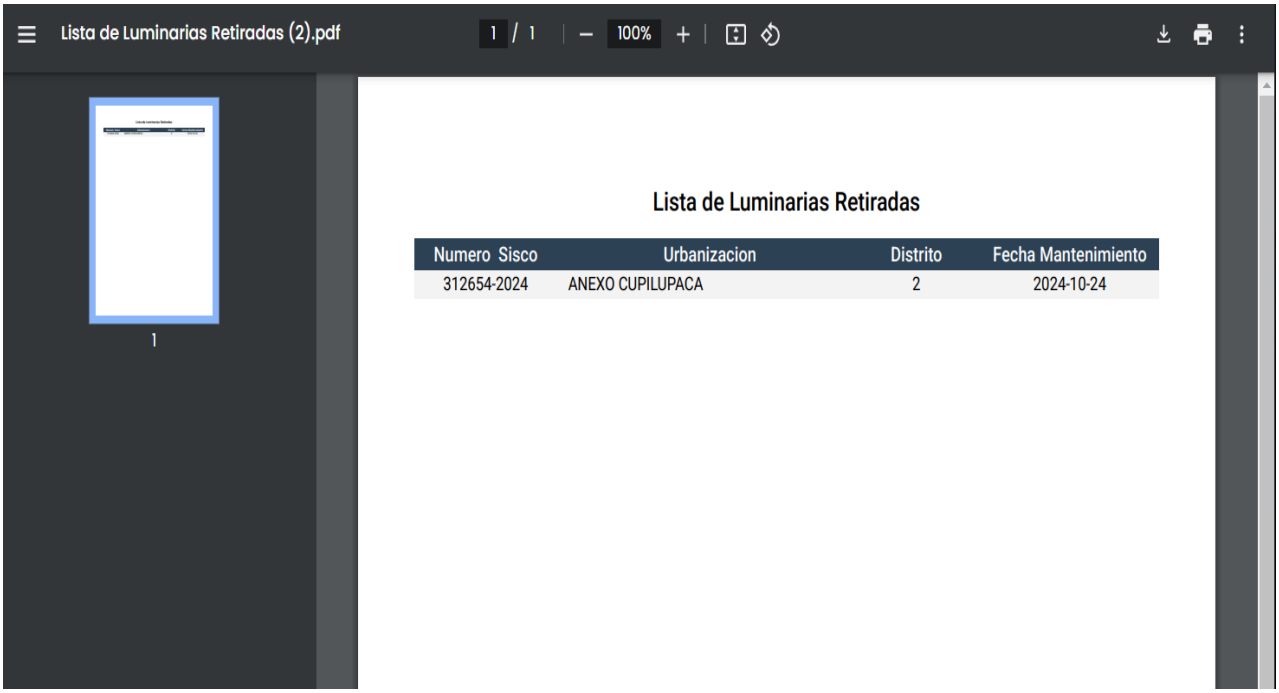
FORMULARIO UAP-R-001

FORMULARIO UAP-R-001	
DATOS TÉCNICOS DE ACCESORIOS Y LUMINARIAS RETIRADOS	
DISTRITO: D-2	**ZONA:** ANEXO CUPILUPACA
PROYECTO: asdfasdfads	
DIRECCIÓN: asdfasdgasdga	
FECHA DE MANTENIMIENTO: 2024-10-24	**NRO. SISCO:** 312654-2024

N°	NOMBRE ÍTEM	CANTIDAD DE ACCESORIOS REUTILIZABLES	CANTIDAD ACCESORIOS NO REUTILIZABLES	CANTIDAD TOTAL
1	CONTACTOR (3 POLOS 380V 400V 22KW, 230V 50/60HZ)	2	1	3

Figura *96.*

Módulo Accesorios y Luminarias Retiradas: Reporte General



Lista de Luminarias Retiradas (2).pdf

Lista de Luminarias Retiradas

Numero Sisco	Urbanizacion	Distrito	Fecha Mantenimiento
312654-2024	ANEXO CUPILUPACA	2	2024-10-24

Figura *97.*

Módulo Accesorios y Luminarias Retiradas: Detalles

Figura *98.*

Módulo Accesorios y Luminarias Retiradas: Buscar

Figura *99.*

Módulo de Inspección: Listar

Figura *100.*

Módulo de Inspección: Registrar

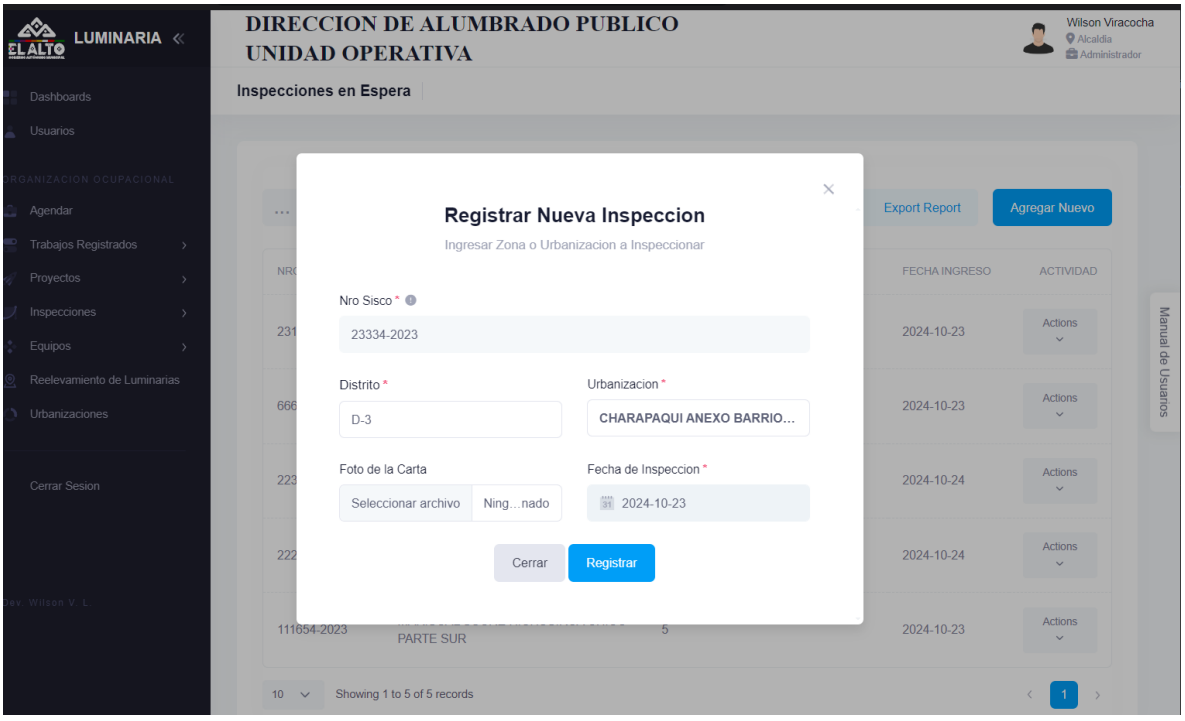


Figura *101.*

Módulo de Inspección: Modificar

Figura *102.*

Módulo de Inspección: Eliminar

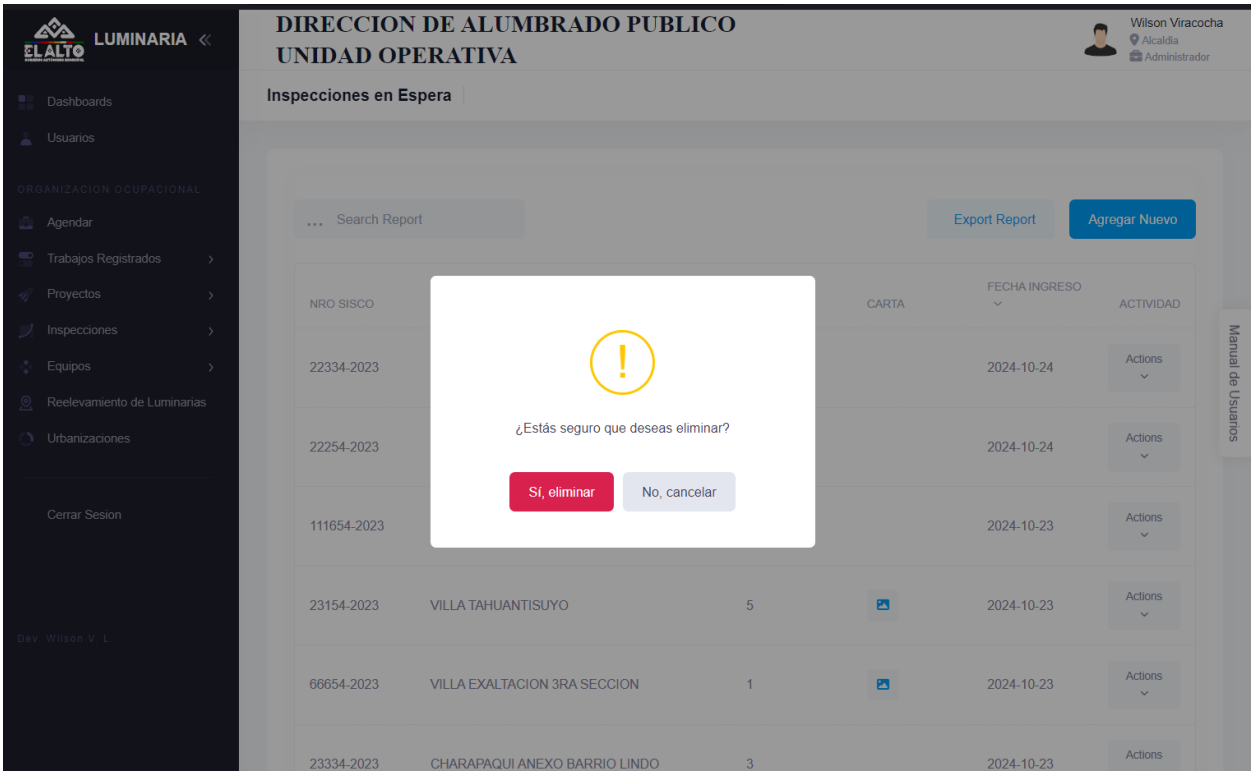


Figura *103.*

Módulo de Inspección: Reporte General

Nro Sisco	Zona/Urb	Distrito	Fecha Ingreso
22334-2023	LOTEO CHARAPAQUI MZNO B	5	2024-10-24
22254-2023	GERMAN BUCH 1-1-2 3RA SEC.	5	2024-10-24
111654-2023	MARISCAL SUCRE HICHUSIRCA CHICO PARTE SUR	5	2024-10-23
23154-2023	VILLA TAHUANTISUYO	5	2024-10-23
66654-2023	VILLA EXALTACION 3RA SECCION	1	2024-10-23
23334-2023	CHARAPAQUI ANEXO BARRIO LINDO	3	2024-10-23

Figura *104.*

Módulo de Inspección: Buscar

Figura *105.*

Módulo de Inspección Realizadas: Listar

Figura *106.*

Módulo de Inspección Realizadas: Modificar

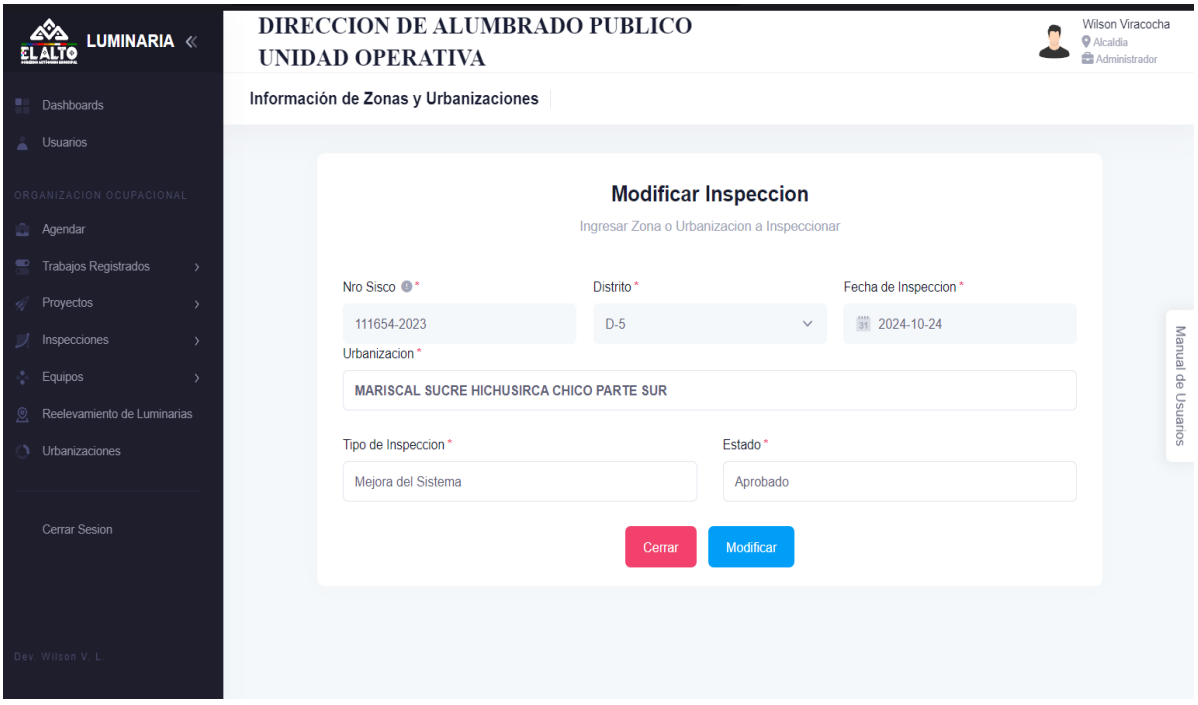


Figura *107.*

Módulo de Inspección Realizadas: Eliminar

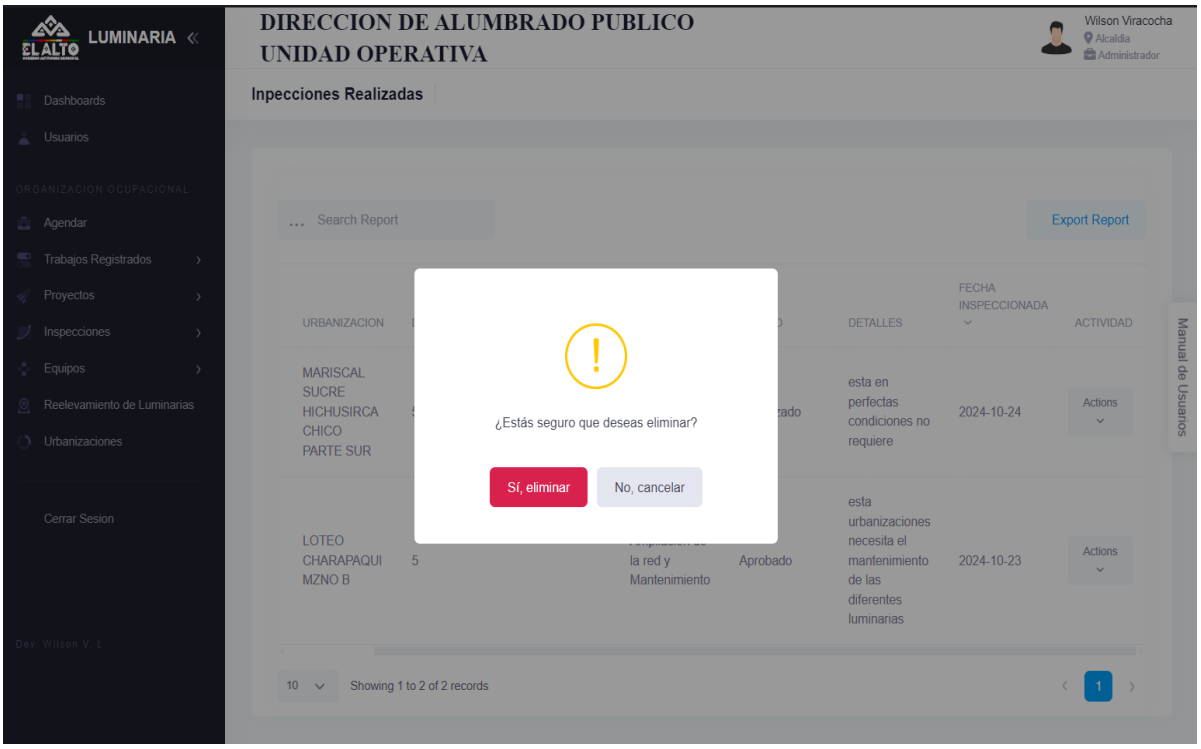


Figura *108.*

Módulo de Inspección Realizadas: Reporte

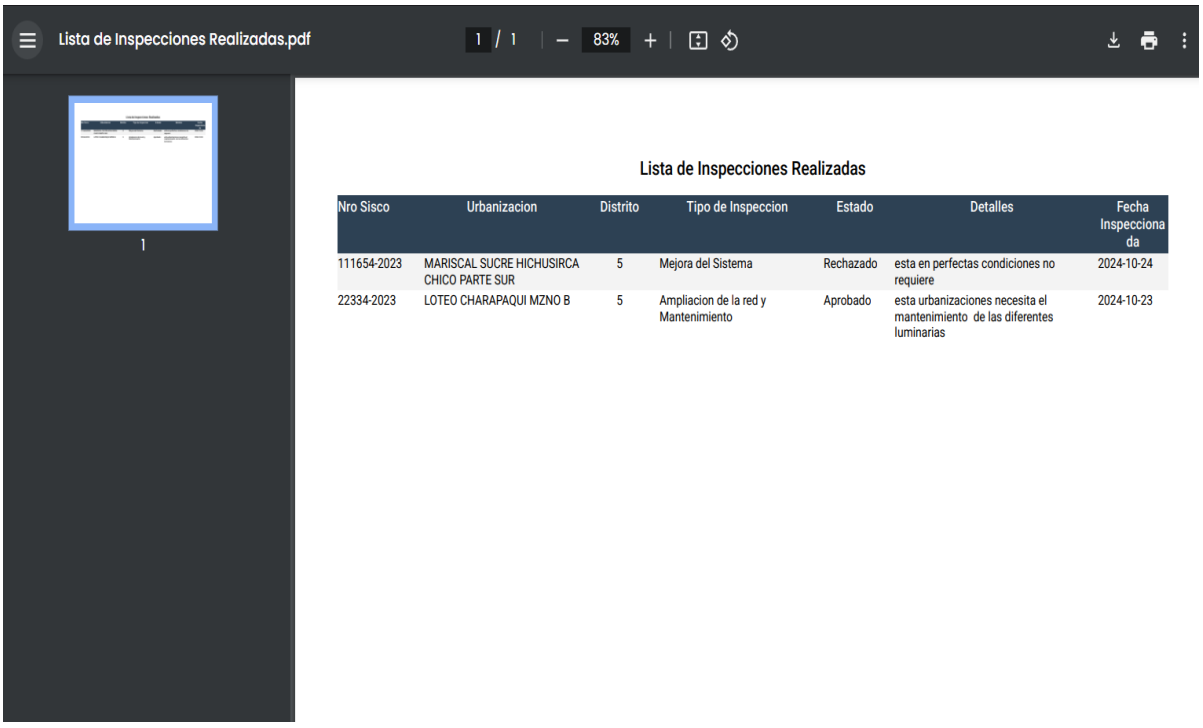


Figura *109.*

Módulo de Equipamientos: Por Distritos

Figura *110.*

Módulo de Equipamientos: Registrar

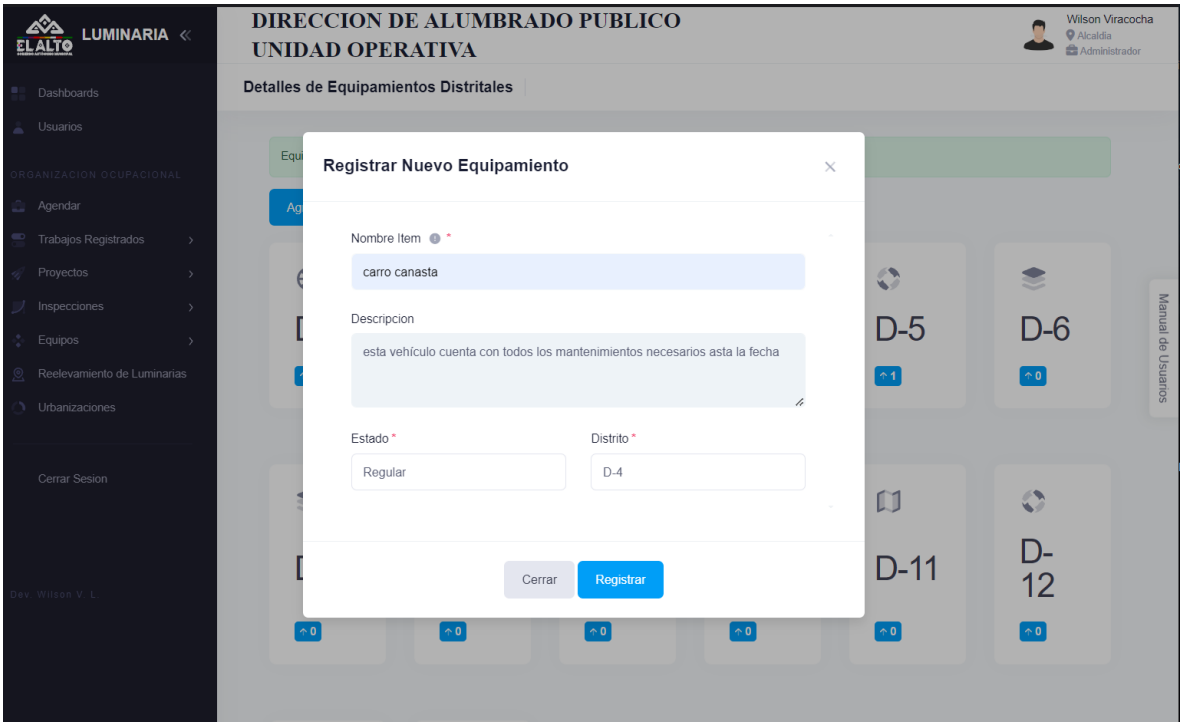


Figura *111.*

Módulo de Equipamientos: Listar

Figura *112.*

Módulo de Equipamientos: Modificar

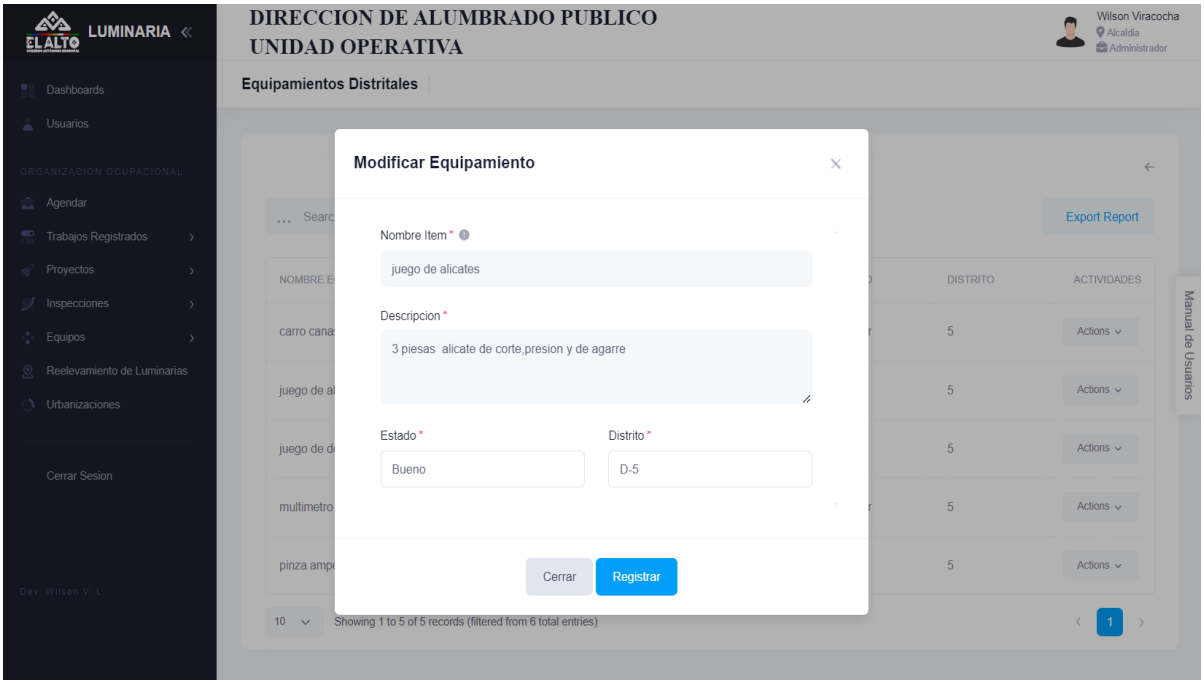


Figura *113.*

Módulo de Equipamientos: Eliminar

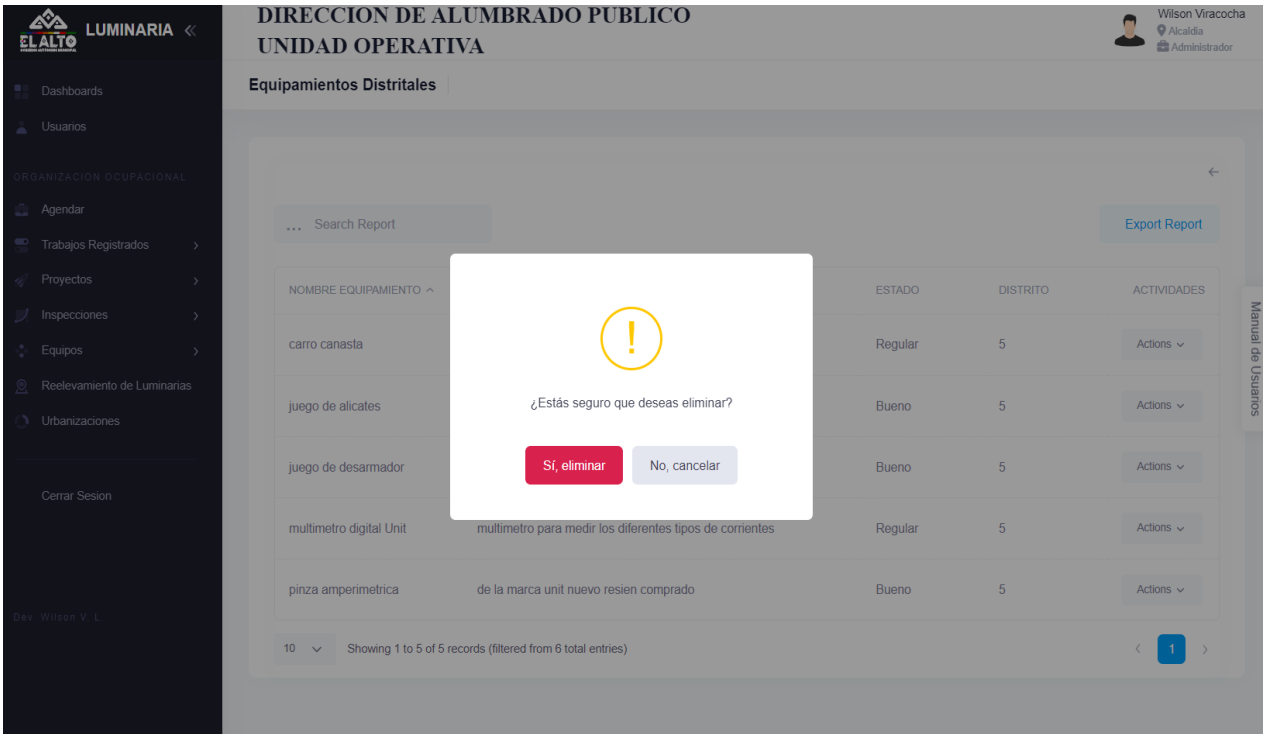


Figura *114.*

Módulo de Equipamientos: Buscar

Figura *115.*

Módulo de Accesorios: Listar

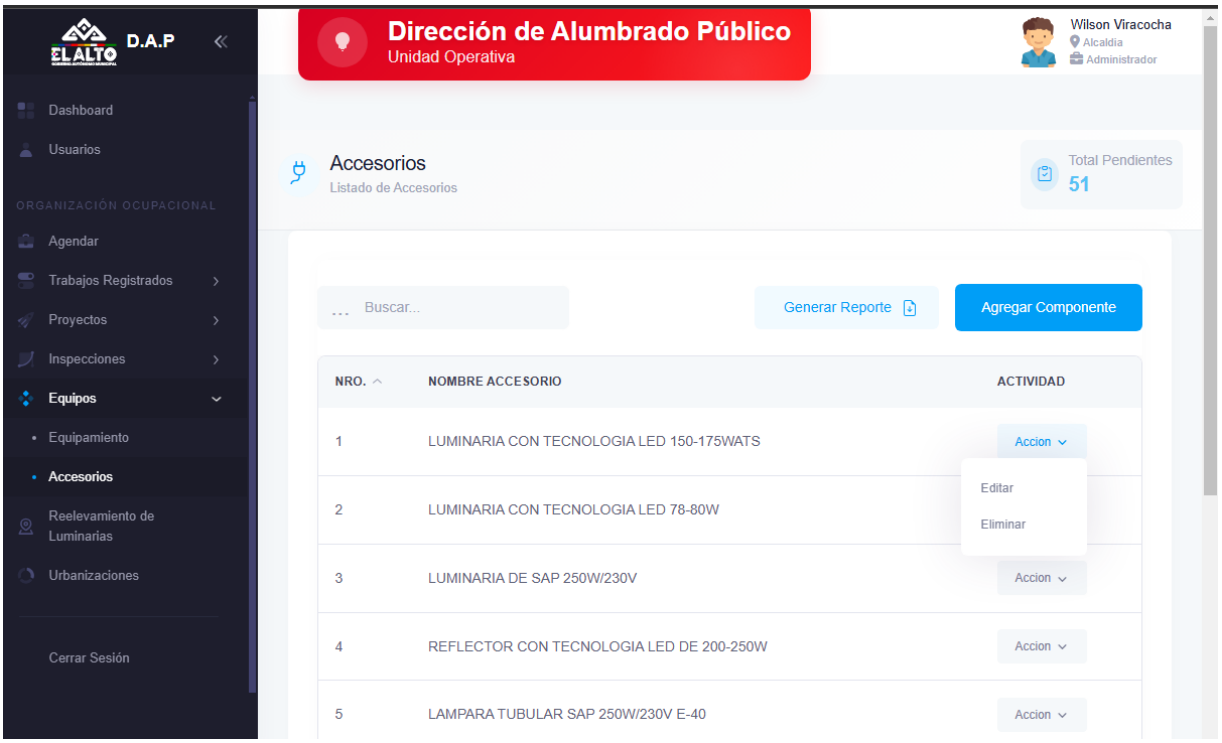


Figura *116.*

Módulo de Accesorios: Registrar

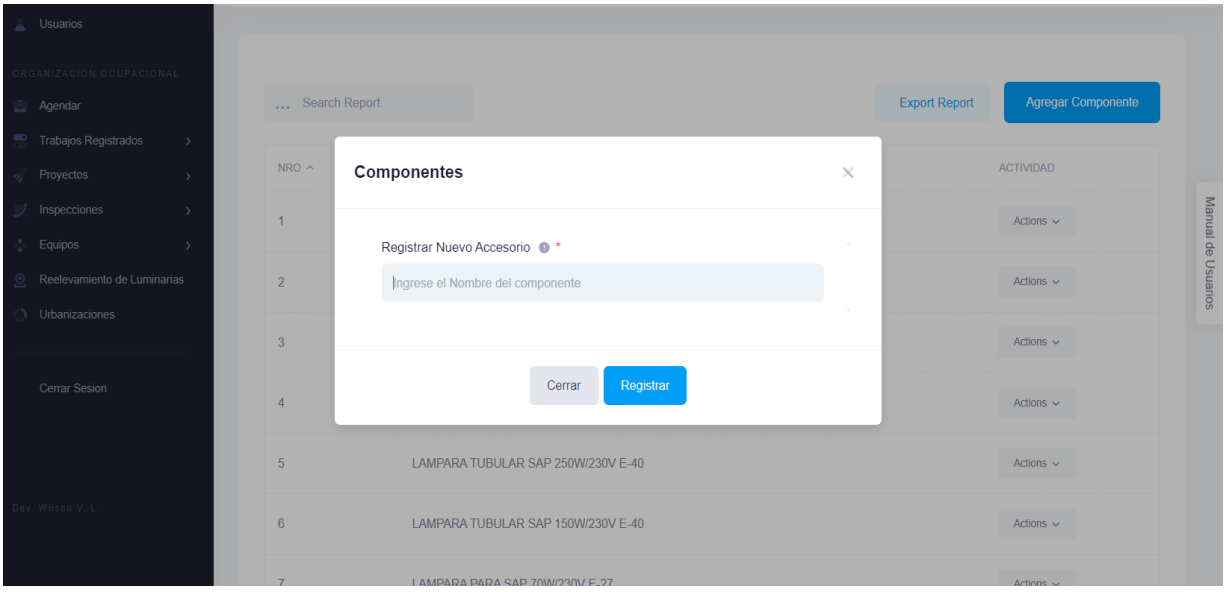


Figura *117.*

Módulo de Accesorios: Modificar

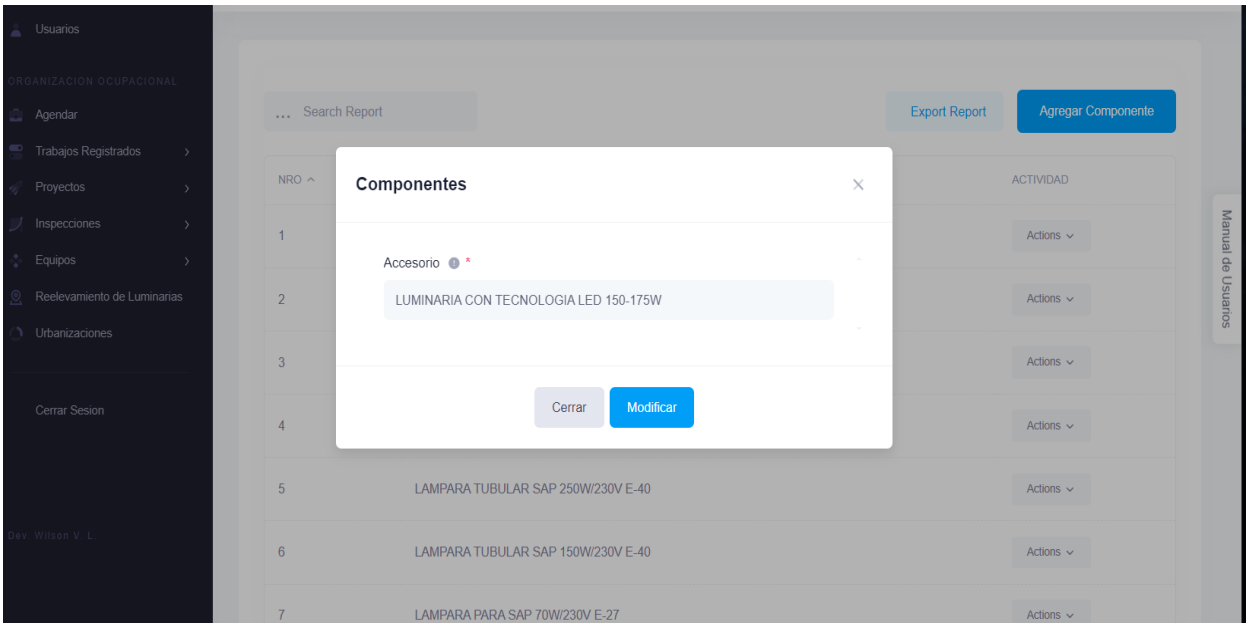


Figura *118.*

Módulo de Accesorios: Eliminar

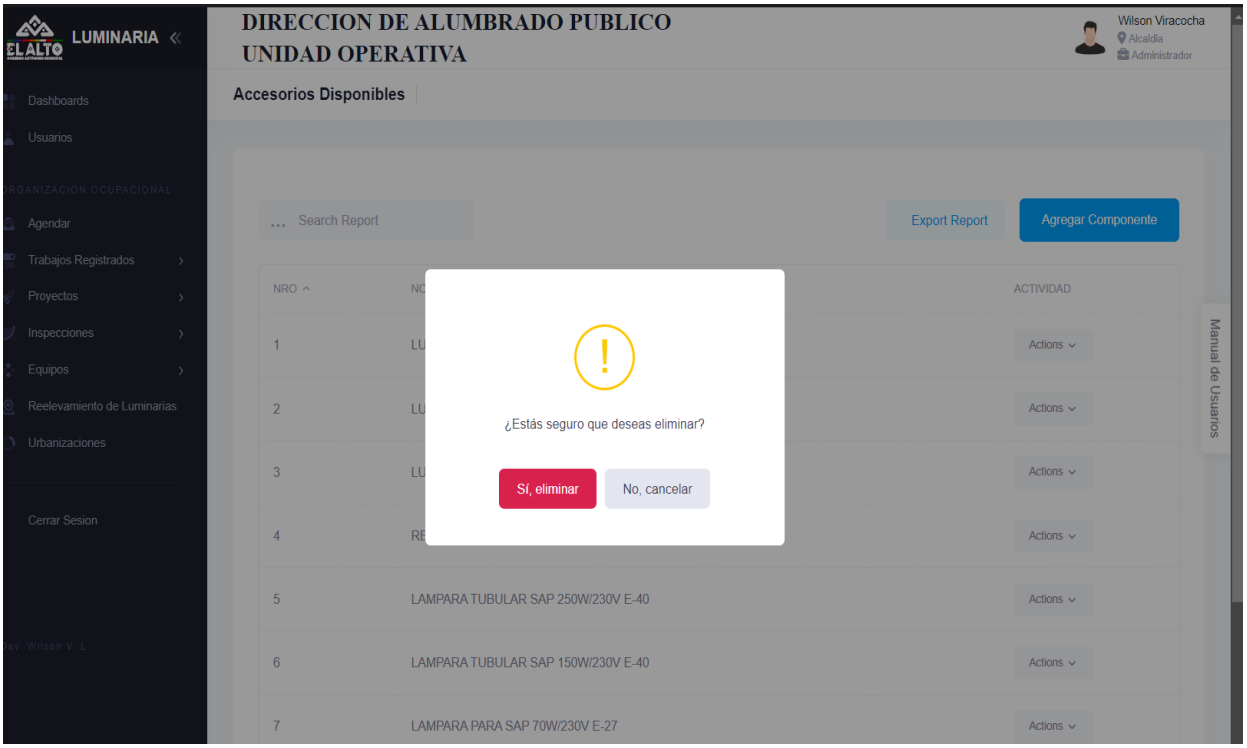
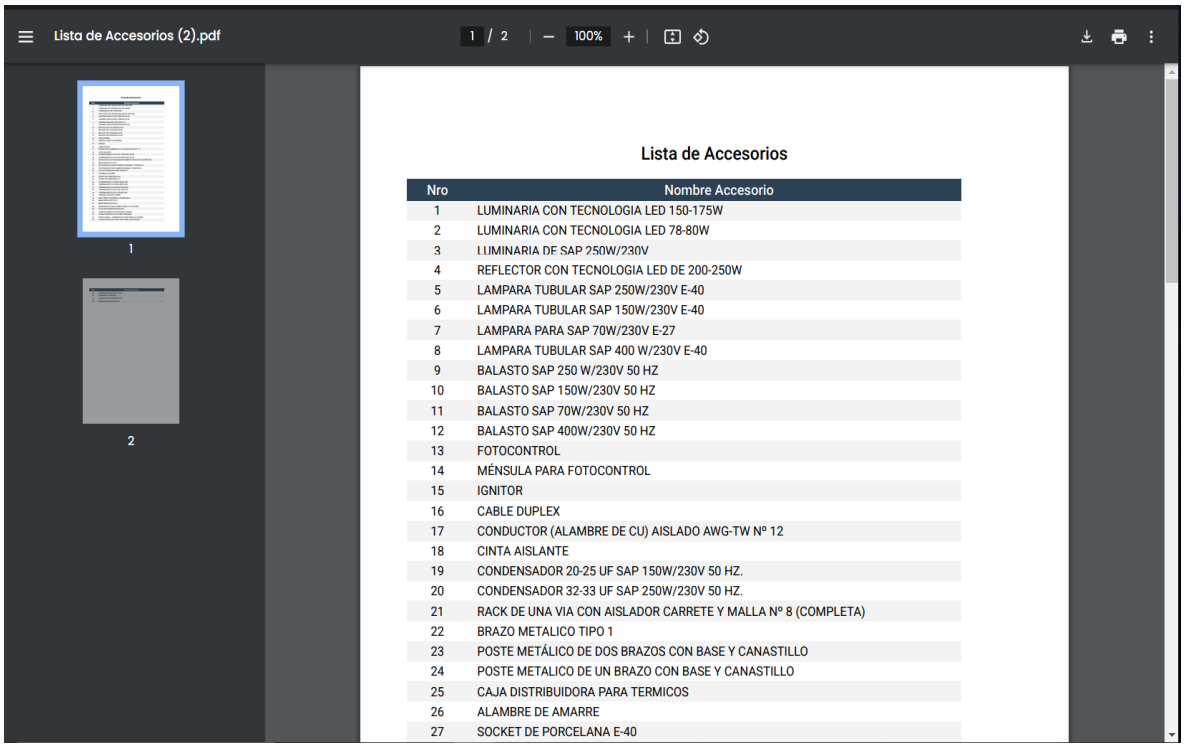


Figura *119.*

Módulo de Accesorios: Reporte General



Lista de Accesorios

Nro	Nombre Accesorio
1	LUMINARIA CON TECNOLOGIA LED 150-175W
2	LUMINARIA CON TECNOLOGIA LED 78-80W
3	LUMINARIA DE SAP 250W/230V
4	REFLECTOR CON TECNOLOGIA LED DE 200-250W
5	LAMPARA TUBULAR SAP 250W/230V E-40
6	LAMPARA TUBULAR SAP 150W/230V E-40
7	LAMPARA PARA SAP 70W/230V E-27
8	LAMPARA TUBULAR SAP 400 W/230V E-40
9	BALASTO SAP 250 W/230V 50 HZ
10	BALASTO SAP 150W/230V 50 HZ
11	BALASTO SAP 70W/230V 50 HZ
12	BALASTO SAP 400W/230V 50 HZ
13	FOTOCONTROL
14	MÉNSULA PARA FOTOCONTROL
15	IGNITOR
16	CABLE DUPLEX
17	CONDUCTOR (ALAMBRE DE CU) AISLADO AWG-TW Nº 12
18	CINTA AISLANTE
19	CONDENSADOR 20-25 UF SAP 150W/230V 50 HZ.
20	CONDENSADOR 32-33 UF SAP 250W/230V 50 HZ.
21	RACK DE UNA VIA CON AISLADOR CARRETE Y MALLA Nº 8 (COMPLETA)
22	BRAZO METALICO TIPO 1
23	POSTE METÁLICO DE DOS BRAZOS CON BASE Y CANASTILLO
24	POSTE METALICO DE UN BRAZO CON BASE Y CANASTILLO
25	CAJA DISTRIBUIDORA PARA TERMICOS
26	ALAMBRE DE AMARRE
27	SOCKET DE PORCELANA E-40

Figura *120.*

Módulo de Accesorios: Buscar

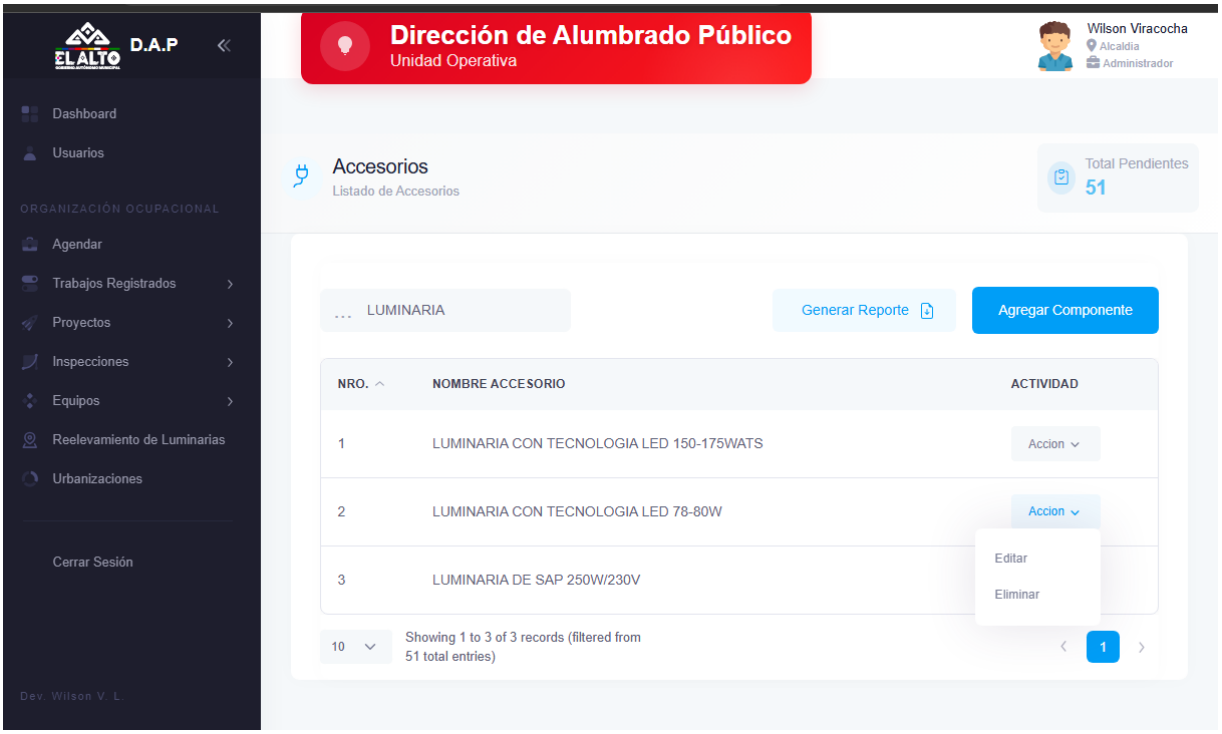


Figura *121.*

Módulo de Reelevamiento: Por Distrito

Figura *122.*

Módulo de Reelevamiento: Registrar

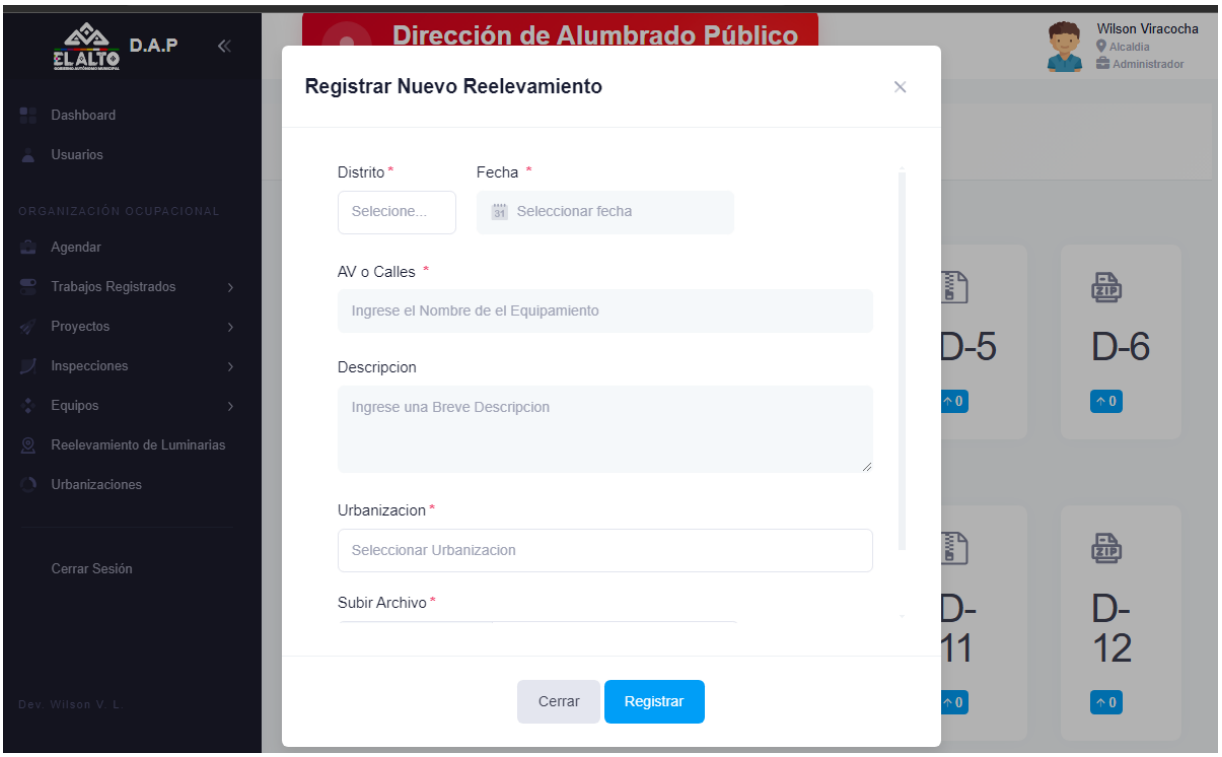


Figura *123.*

Módulo de Reelevamiento: Listar

Figura *124.*

Módulo de Reelevamiento: Modificar

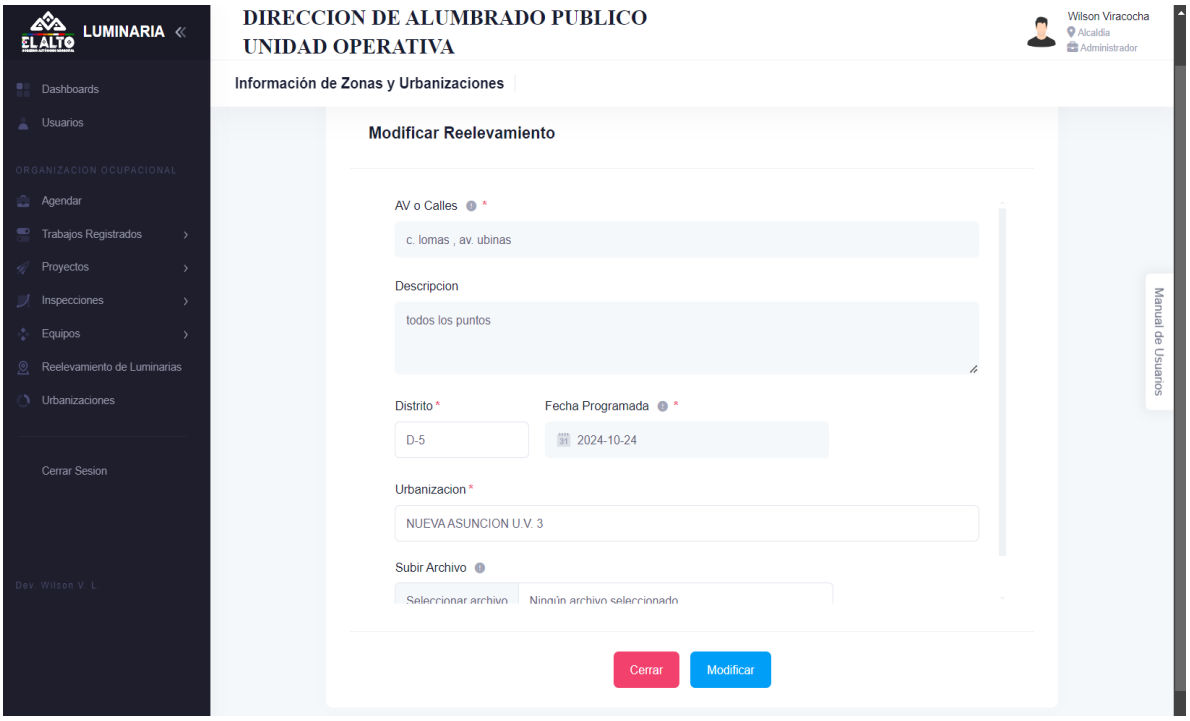


Figura *125.*

Módulo de Reelevamiento: Eliminar

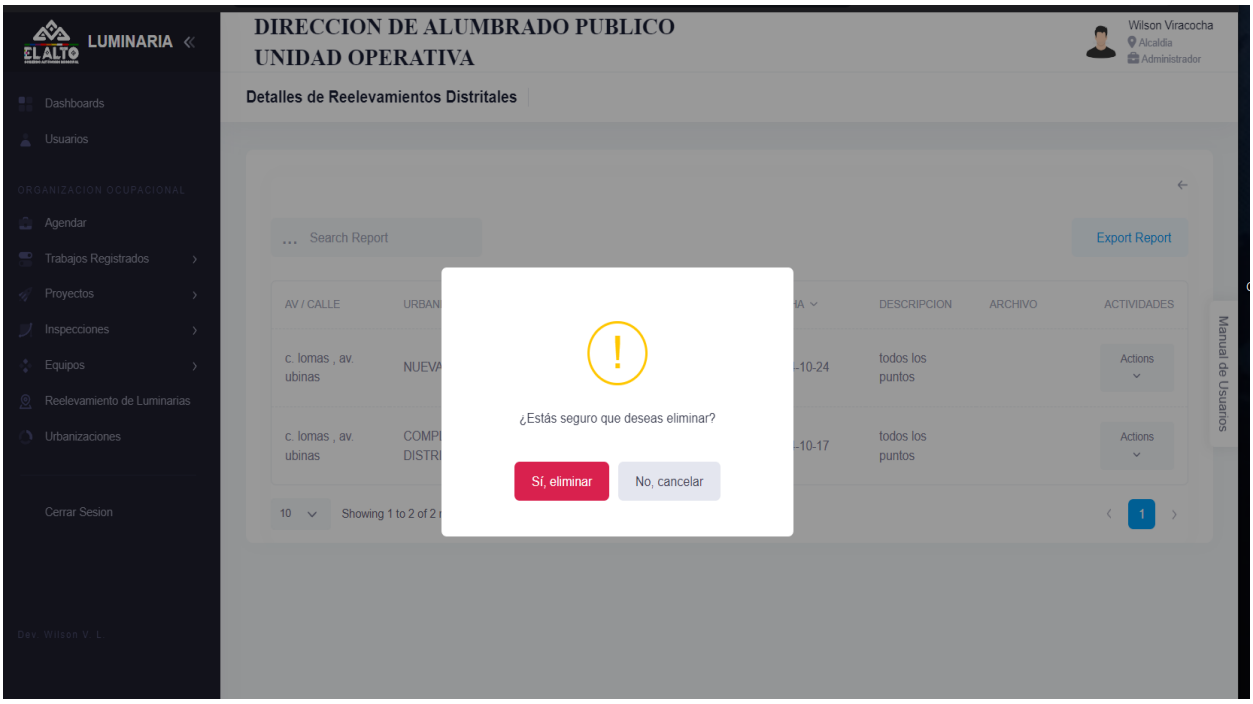


Figura *126.*

Módulo de Reelevamiento: Reporte General Por Distrito

Figura *127.*

Módulo de Reelevamiento: Buscar

Figura *128.*

Módulo de Urbanización: Listar

Figura *129.*

Módulo de Urbanización: Registrar

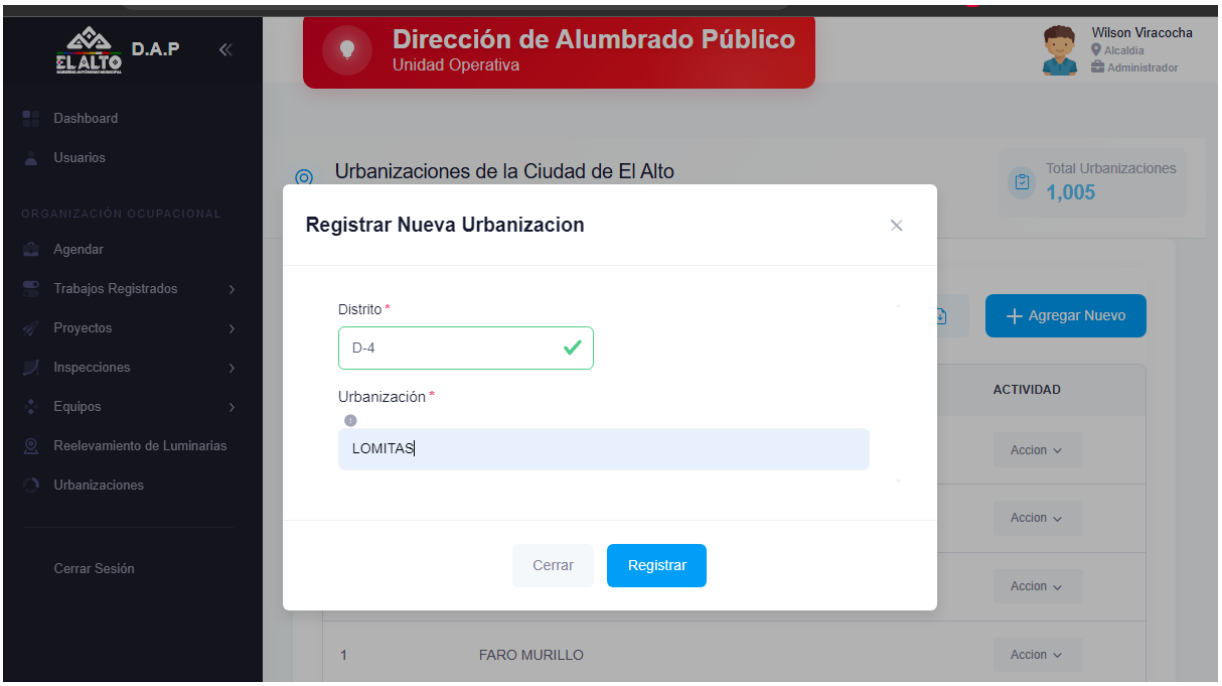


Figura *130.*

Módulo de Urbanización: Modificar

Figura *131.*

Módulo de Urbanización: Eliminar

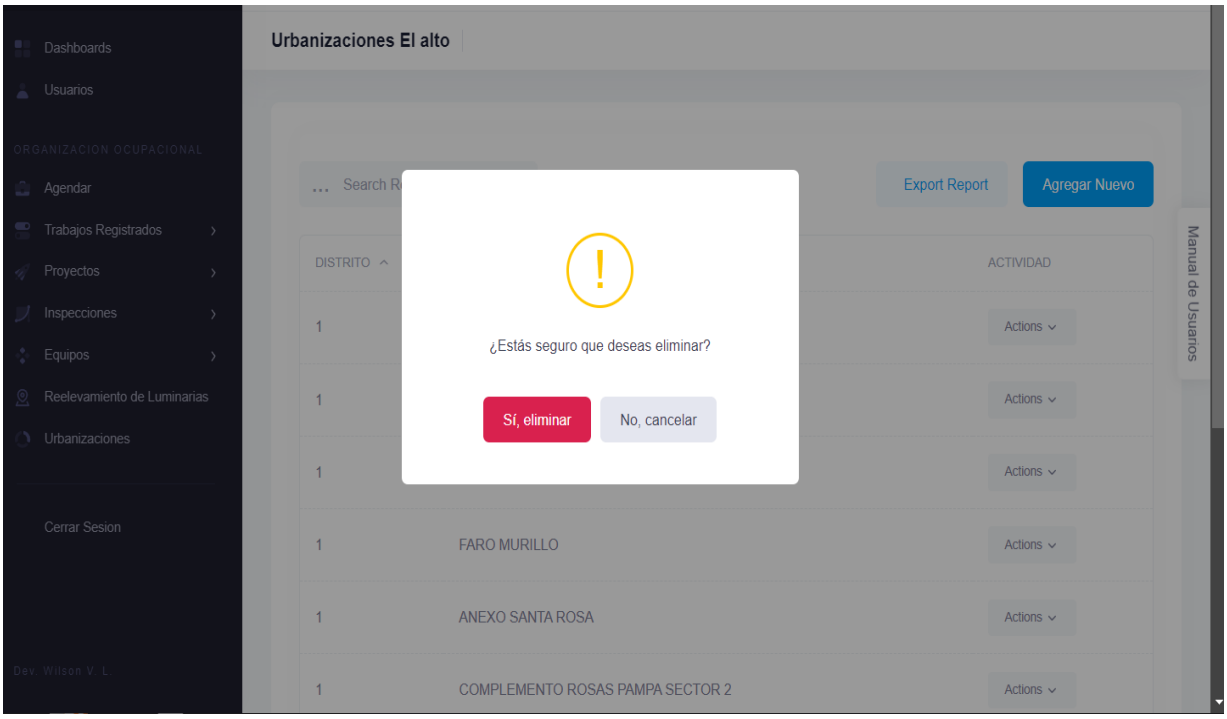


Figura *132.*

Módulo de Urbanización: Reporte General

Figura *133.*

Módulo de Urbanización: Buscar

Figura *134.*

Perfil de Usuario

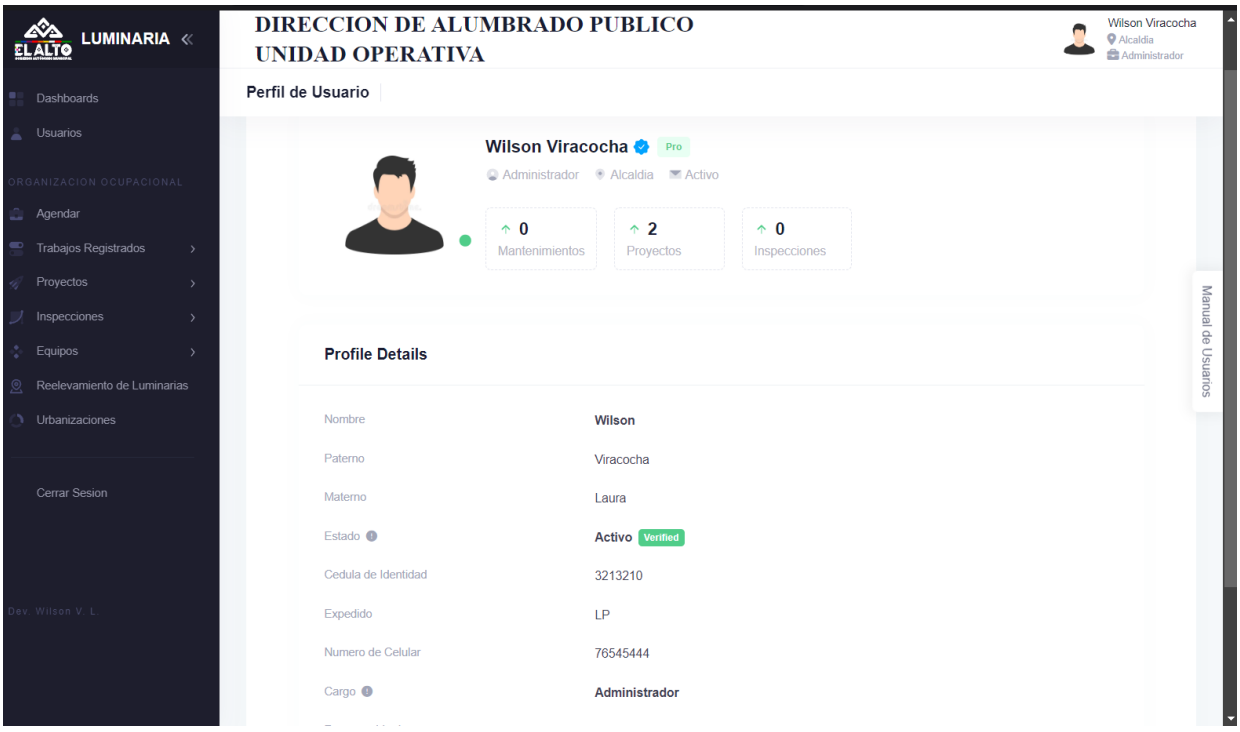
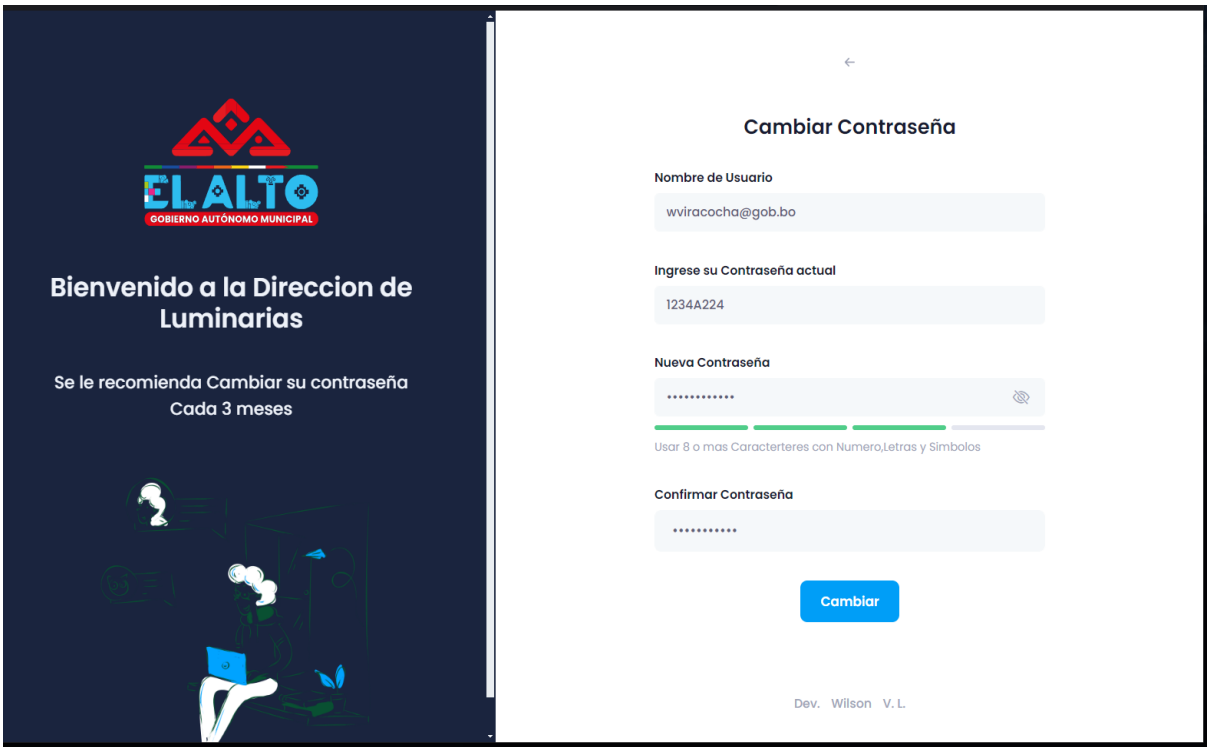


Figura *135.*

Cambiar Contraseña



3.4. PRUEBAS DEL SISTEMA

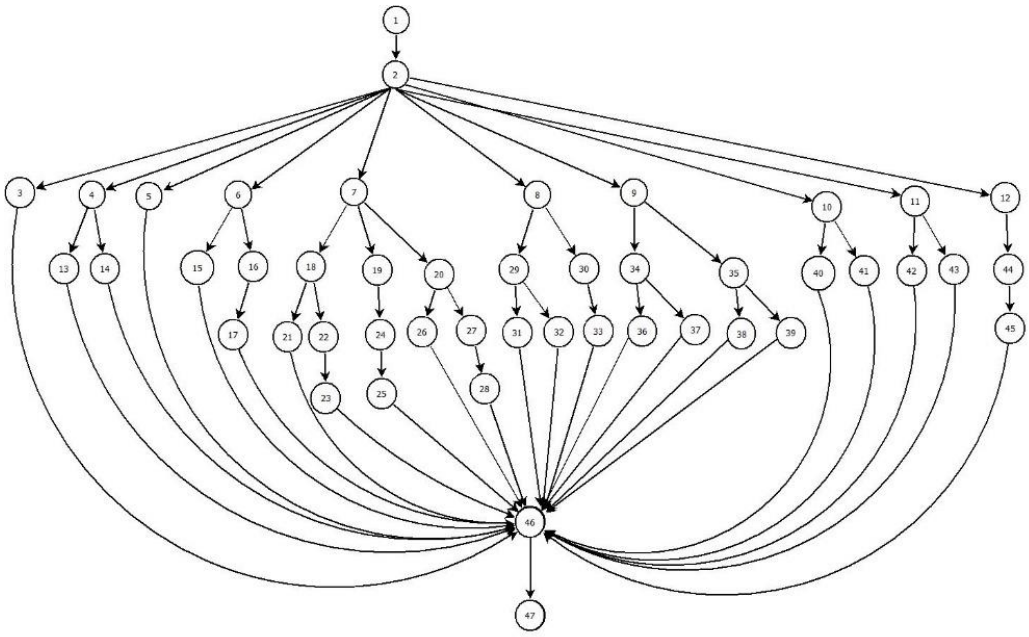
Una vez finalizado el desarrollo del sistema, se procede a llevar a cabo las pruebas de los procedimientos implementados. El objetivo es identificar cualquier posible error durante el proceso, antes de que el sistema se ponga en funcionamiento.

3.4.1. *Pruebas Estructurales*

En las pruebas estructurales, me enfoco en el código interno del sistema. Como tengo acceso al código, reviso su estructura y lógica. Lo que busco es verificar cómo el sistema procesa los datos, explorando el flujo del código, las condiciones y las funciones. Estas pruebas me permiten detectar errores lógicos, vulnerabilidades o código innecesario. Uso técnicas como la cobertura de código para asegurar que todas las partes importantes del sistema se prueben.

Figura *136.*

Diagrama de pruebas estructurales



Donde:

- Inicio del Sistema
- Menú Principal
- Dashboard
- Administración de Usuarios
- Agendar Trabajo
- Detalles Generales
- Proyectos
- Inspecciones
- Equipos
- Reelevamiento de Luminarias
- Urbanizaciones
- Realizar Trabajo

- Listar Usuarios
- Registrar
- Trabajos en Espera
- Trabajos Realizados
- Generar Reporte
- Proyectos en Espera
- Obras Ejecutadas
- Luminarias Retiradas
- Registrar Proyecto
- Listar Proyecto
- Generar Reporte de Proyecto
- Listar Obras Ejecutadas
- Generar Reporte de Obras Ejecutadas
- Registrar Luminaria Retirada
- Listar Luminaria Retirada
- Generar Reporte de Luminaria Retirada
- Inspecciones en Espera
- Inspecciones Realizadas
- Registrar Inspección
- Listar Inspección en Espera
- Listar Inspección Realizada
- Equipamiento
- Accesorios
- Registrar Equipamiento
- Listar Equipamiento

- Registrar Accesorios
- Listar Accesorios
- Registrar Reelevamiento
- Listar Reelevamiento
- Registrar Urbanización
- Listar Urbanización
- Listar Trabajos Pendientes
- Empezar Trabajo
- Fin Ciclo Sistema
- Fin Sistema

Tras realizar el análisis del grafo basado en las características del sistema, el siguiente paso es determinar la complejidad ciclomática del grafo mediante:

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

Donde:

A= 68 (Aristas)

N= 47 (Nodos)

$$V(G) = 68 - 47 + 2 = 23$$

Por lo tanto, la complejidad ciclomática es: V(G) = 23, lo que indica que hay 23 caminos independientes.

- Camino 1: 1-2-3-47-48
- Camino 2: 1-2-4-13-47-48
- Camino 3: 1-2-4-14-47-48
- Camino 4: 1-2-5-47-48

- Camino 5: 1-2-6-15-47-48
- Camino 6: 1-2-6-16-17-47-48
- Camino 7: 1-2-7-18-21-47-48
- Camino 8: 1-2-7-18-22-23-47-48
- Camino 9: 1-2-7-19-24-25-47-48
- Camino 10: 1-2-7-20-26-47-48
- Camino 11: 1-2-7-20-27-28-47-48
- Camino 12: 1-2-8-29-31-47-48
- Camino 13: 1-2-8-29-32-47-48
- Camino 14: 1-2-8-30-33-47-48
- Camino 15: 1-2-9-34-36-47-48
- Camino 16: 1-2-9-34-37-47-48
- Camino 17: 1-2-9-35-38-47-48
- Camino 18: 1-2-9-35-39-47-48
- Camino 19: 1-2-10-40-47-48
- Camino 20: 1-2-10-41-47-48
- Camino 21: 1-2-11-42-47-48
- Camino 22: 1-2-11-43-47-48
- Camino 23: 1-2-12-44-45-47-48

Preparar los casos de prueba para asegurar la ejecución de cada camino. Esta condición final requiere que, para ejecutar algunos caminos, se definan las condiciones necesarias que permitan ejecutar al menos los nodos específicos en ese camino.

Camino 1: Se muestra el Dashboard donde podemos encontrar los datos de detalles de trabajo, inspecciones y proyectos

Camino 2: Se registra los usuarios que van a tener acceso al sistema

Camino 3: Se lista todos los usuarios con acceso al sistema

Camino 4: Se agenda los diferentes trabajos para todos los distritos

Camino 5: Lista todos los trabajos en espera de los diferentes distritos de la ciudad de el alto

Camino 6: Lista todos los trabajos Realizados de los diferentes distritos de la ciudad de el alto como también se genera reporte del trabajo

Camino 7: Se registra los diferentes proyectos de cada distrito y urbanización

Camino 8: Lista todos los proyectos pendientes de cada urbanización como también genera reporte del proyecto

Camino 9: Lista todos los proyectos finalizados como también genera reporte de todos los detalles

Camino 10: Registra luminarias retiradas

Camino 11: Lista todas las luminarias retiradas de los diferentes distritos de la ciudad de el alto

Camino 12: Registra inspeccione

Camino 13: Lista todas las inspecciones pendientes de las diferentes urbanizaciones de la ciudad de el alto

Camino 14: Lista todas las inspecciones realizadas de las diferentes urbanizaciones de la ciudad de el alto

Camino 15: Registra todos los equipamientos de cada distrito

Camino 16: Lista los equipamientos que tiene los 14 distritos

Camino 17: Registra los accesorios necesarios para los mantenimientos

Camino 18: Lista accesorios utilizados en los diferentes trabajos

Camino 19: Registra reelevamiento de cada urbanización

Camino 20: Lista reelevamiento de todos los distritos

Camino 21: Registra Urbanizaciones

Camino 22: Lista urbanizaciones

Camino 23: Realizar trabajo

3.4.2. ***Pruebas Funcionales***

En este caso, no interesa el código, sino el comportamiento externo del sistema. Aquí, trato al sistema como una "caja cerrada", donde solo veo los resultados que genera a partir de las entradas que le doy. Lo que busco es verificar si el sistema responde correctamente a diferentes situaciones, validando que cumpla con los requisitos y que los usuarios finales no se enfrenten a errores. Es útil para identificar fallos funcionales.

Tabla 13.

Prueba Funcionales

	Entrada
Aprobación	Registro de Usuarios Agendar Trabajo Registro de proyectos Registro de Inspecciones Registro de Equipamientos Registro de Accesorios Registro de Reelevamientos Registro de Urbanizaciones
Seguridad	Una vez teniendo usuarios y contraseñas se podrá acceder.
Funciones	
Software	Menú de Administrador

	Menú de Veedor Menú de Coordinador Menú de Técnico
Salida	
Resultado	• Datos estadísticos de Mantenimiento, inspección y proyectos • Reporte de Usuarios Activos • Reporte de trabajos realizados • Reporte de proyectos • Reporte de obras ejecutadas • Reporte de luminarias retiradas • Reporte de inspecciones en espera • Reporte de inspecciones realizadas • Reporte de equipamiento • Reporte de accesorios • Reporte de reelevamiento • Reporte de urbanizaciones • Datos de Usuarios con acceso al sistema • Datos de trabajos de mantenimientos e instalación • Datos de proyectos • Datos de luminarias retiradas • Datos de equipamiento • Datos de accesorios • Datos de urbanizaciones
Seguridad	Son resultados que verán los usuarios al acceder con su usuario y contraseña

CAPÍTULO IV
CALIDAD, COSTO, SEGURIDAD Y PRUEBAS

INGENIERÍA
DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO

4. CAPÍTULO IV

4.1. CICLO DE VIDA DEL SISTEMA

Basado en la información que sea proporcionado y considerando que se desarrolló el sistema "SISTEMA WEB PARA EL REGISTRO, SEGUIMIENTO Y CONTROL EFICIENTE DE LUMINARIAS PÚBLICAS, MATERIALES Y HERRAMIENTAS", se estimó que el ciclo de vida de este sistema será aproximadamente de 5 a 7 años.

Esta estimación se basa en los siguientes factores:

1. Tecnología utilizada:
   - Al ser un sistema web, la tecnología base (frameworks, lenguajes, etc.) suele tener un ciclo de vida más prolongado que soluciones de escritorio o móviles.
   - La adopción de arquitecturas modernas y flexibles, como las empleadas en el framework Laravel 10, permiten una mayor longevidad del sistema.
2. Calidad y mantenibilidad:
   - El cumplimiento de la métrica del 92% en la ISO/IEC 25000 indica que he diseñado y desarrollado el sistema con altos estándares de calidad.
   - La documentación, modularidad y escalabilidad inherente al uso de Laravel 10 facilitan el mantenimiento y la evolución del sistema a lo largo del tiempo.
3. Necesidades del negocio:
   - El sistema de gestión de luminarias públicas, materiales y herramientas es una solución core para la organización, lo que implica una necesidad de mantener su funcionamiento y evolución durante varios años.
   - La naturaleza del negocio y las regulaciones cambiantes pueden requerir actualizaciones y mejoras periódicas al sistema.

Teniendo en cuenta estos factores, mi estimación es que el ciclo de vida útil de este sistema será aproximadamente de 5 a 7 años, siempre y cuando realice las actualizaciones y mantenimiento necesarios durante este período.

Es importante mencionar que esta es una estimación general, y el ciclo de vida real puede variar dependiendo de diversos factores externos e internos que puedan surgir durante la operación del sistema.

4.2. METRICAS DE CALIDAD DE SOFTWARE

En este numeral se verá el desarrollo de la medición de la calidad del software mediante la métrica ISO/IEC 25000 define un modelo general de calidad que ayudara a demostrar que el sistema es confiable.

4.2.1. *Funcional*

A través de la funcionalidad, se puede evaluar la capacidad del software para cumplir y proporcionar funciones que satisfacen tanto las necesidades explícitas como las implícitas del usuario bajo condiciones específicas. Los aspectos a considerar son:

- **Número de entradas de usuario:** se refiere a cada entrada que aporta datos al sistema.
- **Número de salidas de usuario:** se refiere a cada salida que el sistema genera para el usuario, incluyendo informes, pantallas, mensajes de error, etc.
- **Número de peticiones de usuario:** una petición se define como una entrada interactiva que genera una respuesta del software en forma de salidas interactivas.
- **Número de archivos:** se contabilizan archivos maestros lógicos, que pueden ser un grupo lógico de datos o un archivo independiente.

- **Número de interfaces externas:** se cuentan las interfaces legibles por la máquina que se utilizan para intercambiar información con otros sistemas.

La funcionalidad se mide a través de los puntos de función (PF), que ofrecen una medida objetiva, cuantitativa y auditada del tamaño de la aplicación, basada en la percepción del usuario sobre esta (Pressman R., *Ingeniería del Software: Un Enfoque Práctico*, 2010).

Para calcular los puntos de función, se utiliza la siguiente fórmula:

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

Dónde:

PF: Medida de funcionalidad

Cuenta total: es la suma de los siguientes datos (N° De Entradas, N° De Salidas, N° Peticiones, N° De Archivos, N° De Interfaces Externas).

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

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

Σ Fi: Son los valores de ajuste de complejidad, donde (1 ≤ i ≤ 14)

En la siguiente tabla muestra la cuenta de los dominios de información establecidos en el sistema web de acuerdo a los parámetros de medición.

Tabla 14.

Parámetros de Entrada

PARAMETROS DE ENTRADA	CUENTA	FACTOR DE COMPLEJIDAD	RESULTADO
Número de entradas de Usuario	9	6	54
Numero de salidas de Usuario	7	7	49
Número de peticiones de Usuario	7	6	42

PARAMETROS DE ENTRADA	CUENTA	FACTOR DE COMPLEJIDAD	RESULTADO
Numero de archivos	4	15	60
Numero de interfaces externas	2	10	20
Cuenta Total			225

Tabla 15.

Ajuste de Complejidad

FACTOR DE AJUSTE DE VALOR	PESO
¿Requiere el sistema copias de seguridad y de recuperación fiable?	5
¿Se requiere comunicaciones de datos especializadas para transferir información hacia o desde la aplicación?	5
¿Existe funciones de procesamiento distribuidas?	2
¿Es crítico el rendimiento?	5
¿El sistema correrá en un entorno operativo existente enormemente utilizado?	5
¿El sistema requiere entradas de datos en línea?	4
¿La entrada de datos en línea requiere que la transacción de entrada se construya sobre múltiples pantallas u operaciones?	4
¿Los ALI se actualizan en línea?	4
¿Las entradas, salidas, archivos o consultas son complejos?	2
¿El procesamiento interno es complejo?	3
¿El código se diseña para ser reutilizable?	5
¿La conversión y la instalación se incluyen en el diseño?	4
¿El sistema se diseña para instalaciones múltiples en referentes organizaciones?	3
¿La aplicación se diseña para facilitar el cambio y su uso por parte del usuario?	4
Σ Fi	55

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

$$PF = Conteo\ total[0.65 + 0.01 * \Sigma \text{Fi}]$$

Donde:

Σ (Fi)

- Sumatoria de los valores de ajuste

Se tiene:

Σ (Fi)=55

$$PF = 225[0.65 + 0.01 * 55]$$

$$PF = 270$$

A continuación, calculamos el ajuste, con sus pesos máximos siendo este:

Σ (Fi)= 5 * 14

Σ (Fi)= 70

De aquí tenemos:

$$PF\ ajuste = 225 * [0.65 + (0.01 * 70)]$$

$$PF\ ajuste = 303$$

Con esto podemos calcular la funcionalidad del sistema expresándola en porcentaje:

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

$$Funcionalidad = \frac{270}{303} * 100$$

$$\text{Funcionalidad} = 89\%$$

4.2.2. ***Confiabilidad***

ha sido diseñado para mantener un desempeño constante bajo condiciones normales de uso. Me he enfocado en aspectos clave como la madurez, la capacidad de recuperación, y la tolerancia a fallos, asegurando que el sistema pueda recuperarse de posibles errores de manera efectiva. Además, he implementado mecanismos para almacenar los datos de forma segura y para facilitar su acceso, lo que permite una administración estable y confiable. Con estas medidas, mi sistema brinda información confiable y oportuna, garantizando un soporte eficaz para la toma de decisiones.

1. Tiempo Medio Entre Fallos (MTBF)

MTBF = Tiempo Total de Operación / Número de Fallos

MTBF = (14 días × 24 horas) / 2 fallos

MTBF = 336 / 2 = 168 horas entre fallos

2. Disponibilidad del Sistema

Disponibilidad = (Tiempo Total - Tiempo de Inactividad) / Tiempo Total × 100

A = ((336 horas – tiempo inactividad) / 336) × 100

3. Confiabilidad Total del Sistema

R(t) = e^(-t/MTBF) × (A/100) × (RT/RTmax)

Donde:

- t = tiempo de operación continúa deseado (24 horas)

- MTBF = 168 horas

- A = Disponibilidad calculada

- RT = Tiempo de respuesta actual (0.020ms)

- RTmax = Tiempo de respuesta máximo aceptable (0.050ms)

4. Factor de Criticidad para Módulo de Mantenimiento

FC = (PT × IC × D) / 100

Donde:

- PT = Peso de la transacción (70%)

- IC = Índice de criticidad (90%)

- D = Disponibilidad requerida (95%)

5. Confiabilidad Final Ajustada

R final = (R(t) × 0.7) + (FC × 0.3)

Métricas de Monitoreo Continuo

1. Tasa de éxito de transacciones = (Transacciones exitosas / Total de transacciones) × 100
2. Tiempo de respuesta promedio = 0.020ms (Actual) / 0.050ms (Objetivo)
3. Eficiencia de concurrencia = (Usuarios concurrentes sin degradación / Máximo usuarios concurrentes) × 100

Umbrales de Alerta

- Crítico: R final < 95%
- Advertencia: 95% ≤ R final < 97%
- Óptimo: R final ≥ 97%

Meta: R final ≥ 0.95 (95%)

4.2.3. ***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, se utiliza la siguiente formula:

Formula:

$$FU = [\frac{\sum xi}{n} * 100]$$

Donde:

- $\mathrm{xi} = \text{Valores de evaluación de cada pregunta}$
- $\mathrm{n} = \text{Numero de preguntas}$

Se toma en cuenta la siguiente tabla

Tabla 16.

Parámetros de medición de usabilidad

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

Tabla 17.

Preguntas de Usabilidad del Sistema

Nro	**Pregunta**	**Si**	**No**	**Resultado xi**
1	¿Puede Utilizar con facilidad el sistema?	5	0	1
2	¿Puede Controlar operaciones que el sistema solicita?	5	0	1
3	¿Las Respuestas del sistema son complicadas?	0	5	1
4	¿El Sistema permitió la retroalimentación de información?	4	1	0.8
5	¿El sistema cuenta con una interfaz agradable a la vista?	5	0	1
6	¿La respuesta del sistema es satisfactoria?	5	0	1
7	¿Le parece complicadas las funciones del sistema?	0	5	1
8	¿Se hace difícil o dificultoso aprender a manejar el sistema?	0	5	1
9	¿Los resultados que proporciona el sistema facilitan el trabajo?	5	0	0.8
10	¿Durante el uso del sistema se produjo errores?	0	5	1
Total				9.6

Con los resultados obtenidos del cuestionario calculamos la usabilidad:

$$FU = [\frac{9.6}{10} * 100]$$
$$FU = [0.96 * 100]$$
$$FU = 96\%$$

Con esto concluimos que la usabilidad es de 96%

4.2.4. ***Eficiencia***

La eficiencia de un sistema es una medida que determina en qué grado utiliza de manera óptima sus recursos para alcanzar sus objetivos. Para calcular el grado de eficiencia, se emplea una escala que evalúa el desempeño del sistema en función de diversos criterios.

Tabla 18.

Escala de Valores de Eficiencia

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

Para la valoración de la eficiencia del sistema se tiene la siguiente tabla

Tabla 19.

Valoración de la Eficiencia del Sistema

Nro	**Pregunta**	**Porcentaje**
1	¿La distribución y estilo de la interfaz permite que un usuario introduzca con eficiencia las operaciones y la información?	5
2	¿Una secuencia de operaciones (o entrada de datos) puede realizarse con facilidad?	4
3	¿Los datos de salida están presentados de modo que se entienden inmediatamente?	5

Nro	Pregunta	Porcentaje
4	¿Las operaciones jerárquicas están organizadas de manera que minimizan la navegación del usuario para hacer que alguna se ejecute?	5
5	¿Procesa y responde adecuadamente cuando realiza alguna consulta o búsqueda?	5
Total		24

Para calcular la eficiencia:

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

Donde:

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

Remplazando se tiene lo siguiente:

$$E = \frac{24}{5} * \frac{100}{5}$$

$$E = 96\%$$

4.2.5. ***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, que proporciona un indicador de estabilidad del sistema desarrollado.

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

Donde:

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

- Mt: Número de módulos totales 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 20.

Valores para la Mantenibilidad

Descripción	Valor
Mf	9
Fa	0
Fc	1
Fd	0

Remplazando la ecuación se tiene:

$$IMS = \frac{[9 - (0 + 1 + 0)]}{9}$$

$$IMS = 0.88 * 100$$
$$IMS = 88\%$$

4.2.6. ***Portabilidad***

Es la capacidad de un software para ser transferido a un entorno diferente donde se considerarán los siguientes aspectos:

- **Adaptabilidad:** Evalúa la facilidad para adaptar el software a diferentes ambientes sin necesidad de realizar 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 facilidad y el esfuerzo necesario para sustituir el software por otro con funciones similares.

El sistema está diseñado para ser altamente compatible. Funciona en cualquier servidor que cuente con los servicios web Apache, PHP y la base de datos MariaDB. Su desarrollo sobre el framework Laravel agiliza su adaptación a distintos entornos. Al ser una aplicación web, su ejecución es posible en cualquier dispositivo con acceso a internet y un navegador.

Para obtener la portabilidad, se tiene la siguiente formula:

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

Reemplazando la formula se obtiene:

$$Portabilidad = 1 - (\frac{1}{9})$$

$$Portabilidad = 0.888 * 100$$

$$Portabilidad = 89\%$$

Resultados:

Basándonos en los resultados obtenidos, podemos determinar la calidad general del sistema a partir de los indicadores evaluados. Esta calidad está directamente vinculada al nivel de satisfacción que experimenta el usuario al interactuar con el sistema.

Tabla 21.

Resultados

Características	**Resultados**
Funcionalidad	89%
Confiabilidad	95%

Características	Resultados
Usabilidad	96%
Eficiencia	96%
Mantenibilidad	88%
Portabilidad	89%
Evaluación total	92%

4.3. ESTIMACION DE COSTOS

4.3.1. *COCOMO II*

Las estimaciones de costo del sistema han sido desarrolladas bajo las KLDC (Kilo – líneas de código) como se indica a continuación:

El proyecto se implementó con 17.610 líneas de código en el lenguaje de programación PHP.

Como:

$$KLDC = \frac{LDC}{100}$$

$$KLDC = \frac{10.610}{100}$$

$$KLDC = 10{,}61\%$$

Los coefientes que se usarán serán los valores que se detallan en la siguiente

Tabla 22.

Coeficientes del Modelo COCOMO II

Proyecto de Software	a	b	c	D
Orgánico	2.40	1.05	2.50	0.38

Proyecto de Software	a	b	c	D
Semiacoplado	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

Nota. Tabla de coefientes del modelo cocomo II Extraído de: S. Pressman, 2010.

A continuación, se presentan las ecuaciones necesarias para determinar el costo total del software.

Tabla 23.

Ecuaciones del Modelo COCOMO II

Variable	Ecuación	Tipo/Unidad
Esfuerzo requerido por el proyecto	E=a x (KLDC) x FAE	Persona/Meses
Tiempo requerido por el proyecto	T=c x (E)d	Meses
Número de personas requeridas para el proyecto	NP=$\frac{E}{T}$	Persona
Costo Total	CT=Sueldo Mes x NP x T	$us

Nota. Ecuaciones de cocomo II Extrado de: Prentice-Hall, 1981

Tabla 24.

Cálculo de los Atributos FAE

Atributos que afectan al costo	Valor Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Atributos del Software						
Fiabilidad del software	0.75	0.88	1	1.15	1.4	-
Tamaño base de datos	0.94	1	1.08	1.16	-	-
Complejidad del producto	0.7	0.85	1	1.15	1.3	1.65
Atributos del Hardware						

Atributos que afectan al costo	Valor Muy Bajo	Bajo	Nominal	Alto	Muy Alto	Extra Alto
Restricciones de tiempo de ejecución	1	1	1.11	1.3	1.66	-
Restricciones de memoria	1	1.06	1.21	1.56	2.11	-
Volatilidad de máquina virtual	0.87	1	1.15	1.3	-	-
Tiempo de respuesta	0.87	1	1.07	1.15	1.29	-
Atributos de Personas						
Capacidad de análisis	1.46	1.19	1	0.86	1.7	-
Experiencia en la aplicación	1.29	1.13	1	0.91	1.82	-
Capacidad de programadores	1.42	1.17	1	0.86	0.7	-
Experiencia en S.O. usado	1.21	1.1	1	0.9	-	-
Experiencia en el lenguaje de programación	1.14	1.07	1	0.95	-	-
Atributos del Proyecto						
Uso de técnicas actuales de programación	1.24	1.1	1	0.91	0.82	-
Uso de herramientas de software	1.24	1.1	1	0.83	-	-
Restricciones de tiempo de desarrollo	1.23	2.23	1	1.04	1.1	-
Totales			0.395			

De acuerdo con las ecuaciones, coeficientes y exponentes especificados en las tablas Nro 13 y 14, y dado que el tamaño del software (medido en KLDC) es menor a 50, se emplearon los valores correspondientes a tipos orgánicos para calcular los atributos FAE.

- **Calculando el Esfuerzo**

$$\boldsymbol{E = \alpha \times (KLDC)\beta \times FAE}$$

$$\boldsymbol{E = 2,4 \times (10.610)^{1,05}} \times 0.395$$

$$\boldsymbol{E = 11.31 \approx 11\ Personas/Mes}$$

- **Calculando el Tiempo**

$$\boldsymbol{T = c \times (E)d}$$

$$\boldsymbol{T = 2.5\ \times 11.319^{0.38}}$$

$$\boldsymbol{T = 6.28 \approx a\ 6\ meses}$$

- **Calculando el Personal Promedio:**

$$\boldsymbol{NP = \frac{E}{T}}$$

$$\boldsymbol{NP = \frac{11.31}{6.11}}$$

$$\boldsymbol{NP = 1.85 \approx 2\ personas}$$

- **Calculando el Costo Total**

$$\boldsymbol{CT = Sueldo\ Mes\ \times NP\ \times T}$$

$$\boldsymbol{CT = 500\ \times 1\ \times 6}$$

$$\boldsymbol{CT = 3.000\ \$us}$$

$$\boldsymbol{CT = 20.910\ Bs}$$

En resumen, se requiere 1 persona estimando un trabajo de 6 meses y con costo total de 20.910 bs

4.4. SISTEMA DE GESTION DE SEGURIDAD DE LA INFORMACION

la seguridad de la información es como un conjunto de medidas técnicas, organizacionales y legales que permiten a la organización proteger y asegurar la confidencialidad e integridad de la información que genera.

4.4.1. ***Norma ISO 27001***

Para cumplir con la norma ISO 27001, implemento medidas de seguridad en el sistema, como la encriptación y el uso de variables de sesión. La encriptación permite proteger datos sensibles, como las contraseñas, mientras que las variables de sesión ayudan a restringir el acceso. Esto significa que solo los usuarios con credenciales válidas pueden acceder al sistema. Una vez que un usuario se autentica, se crea una variable de sesión que, al finalizar las tareas, es destruida para asegurar la integridad del sistema.

4.4.1.1. **Autenticación, Autorización y Control de Acceso**

La seguridad de la información engloba un conjunto integral de estrategias y controles que abarcan tres pilares fundamentales:

1. El aspecto técnico: implementación de herramientas, sistemas y protocolos tecnológicos
2. El componente organizacional: políticas internas, procedimientos y roles definidos
3. El marco regulatorio: cumplimiento de normativas y estándares legales vigentes

Todos estos elementos trabajan en armonía para garantizar dos objetivos críticos: que la información sensible permanezca protegida de accesos no autorizados (confidencialidad).

Figura *137.*

Autenticación y Validación de Usuarios

```php
public function authenticate(): void
{
    $this->ensureIsNotRateLimited();

    if (! Auth::attempt($this->only('email', 'password'), $this->boolean('remember'))) {
        RateLimiter::hit($this->throttleKey());

        throw ValidationException::withMessages([
            'email' => trans('auth.failed'),
        ]);
    }

    RateLimiter::clear($this->throttleKey());
}
```

4.4.1.2. Encriptación de Contraseña

Laravel utiliza por defecto el algoritmo de hash Bcrypt, que implemento a través del método Hash::make(). Este proceso genera un hash único de 60 caracteres por cada contraseña, aplicando automáticamente un salto aleatorio y ejecutando 12 rondas de encriptación. Lo valioso de este enfoque es que cada vez que un usuario ingresa la misma contraseña, el hash generado es diferente gracias al salto único, lo que fortalece significativamente la seguridad contra ataques de diccionario y fuerza bruta. Además, al ser un hash unidireccional, es matemáticamente imposible revertir el proceso para obtener la contraseña original, garantizando así la confidencialidad de las credenciales de los usuarios.

Figura *138.*

Encriptación de Contraseña

```
$user->email = $usuario . $lash;
$user->Password =  Hash::make($contrase);
$user->assignRole($rol->name);
$user->save();
```

4.4.1.3. Csrf en Formularios

En Laravel, CSRF me proporciona una capa adicional de seguridad de manera muy sencilla. Cada vez que creo un formulario, Laravel genera automáticamente un **token CSRF** único para esa sesión. Este token se incluye en el formulario y actúa como una especie de contraseña. Cuando el formulario se envía, Laravel verifica que el token que llega con la solicitud coincida con el que se generó.

Figura *139.*

Csrf en Formularios

```
<div class="modal-content">
    <form class="form" action="{{route("registro.usuario")}}" id="modadRegistraUsuarios_form" method="POST">
        @csrf
        <div class="modal-header" id="modadRegistraUsuarios_header">
            <h2>Registrar Nuevo Usuario</h2>
```

4.5. PRUEBAS

4.5.1. *Introducción*

Este capítulo documenta las pruebas realizadas al sistema web para el Registro, Seguimiento y Control Eficiente de Luminarias Públicas, Materiales y Herramientas. Las pruebas fueron diseñadas y ejecutadas siguiendo metodologías estándar de la industria y considerando los requerimientos de seguridad de la norma ISO 27001.

4.5.2. *Pruebas de Estructurales*

4.5.3. *Descripción General*

Las pruebas estructurales se centraron en examinar la lógica interna del software, particularmente en los formularios y la seguridad del sistema.

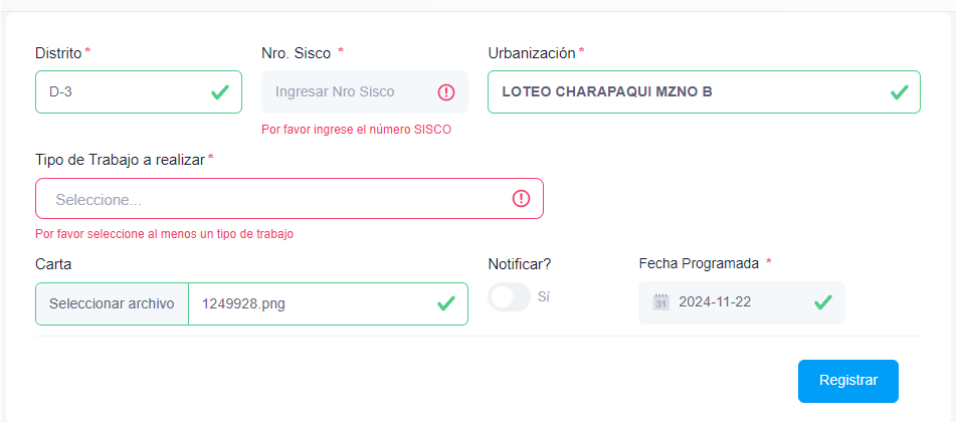
4.5.3.1. Áreas Probadas

a) Validación de Formularios

- Lógica de Validación
- Verificación de reglas de validación
- Prueba de funciones de sanitización
- Comprobación de manejo de errores

Figura *140.*

Validación de Formulario



b) Seguridad

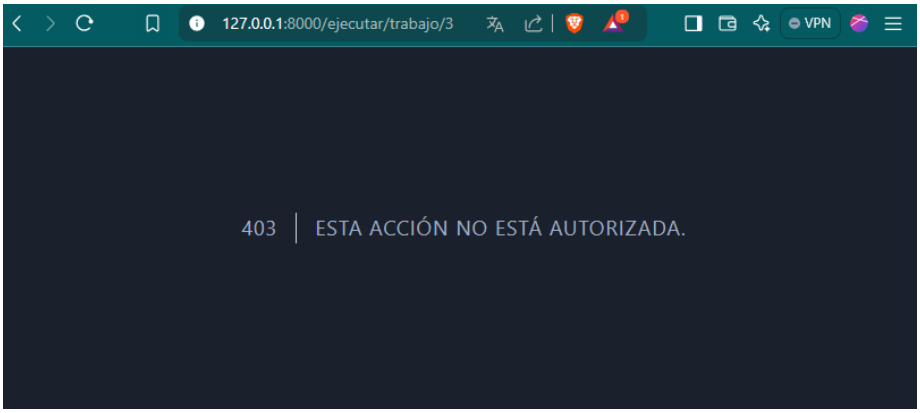
- Implementaciones de Laravel 10
- Verificación de middleware de autenticación
- Prueba de protección contra inyección SQL
- Validación de manejo de sesiones

Figura *141.*

Validación de Inicio de Sesión

Figura *142.*

Verificación de middleware



4.5.3.2. **Resultados**

- Se confirmó el correcto funcionamiento de las validaciones
- Se verificó la efectividad de las medidas de seguridad implementadas
- Se comprobó la integridad de los datos en operaciones CRUD

4.5.4. ***Pruebas Funcionales***

4.5.4.1. **Descripción General**

Las pruebas funcionales se realizaron evaluando el comportamiento del sistema desde una perspectiva externa, sin considerar la estructura interna del código. Se enfocaron en verificar que cada entrada produzca los resultados esperados.

4.5.4.2. **Casos de Prueba Ejecutados**

a) Módulo de Autenticación

- Prueba de Login
- Entrada de credenciales válidas
- Entrada de credenciales inválidas
- Verificación del bloqueo después de 5 intentos fallidos

- Comprobación de encriptación de contraseñas

Figura *143.*

Verificación de bloqueo

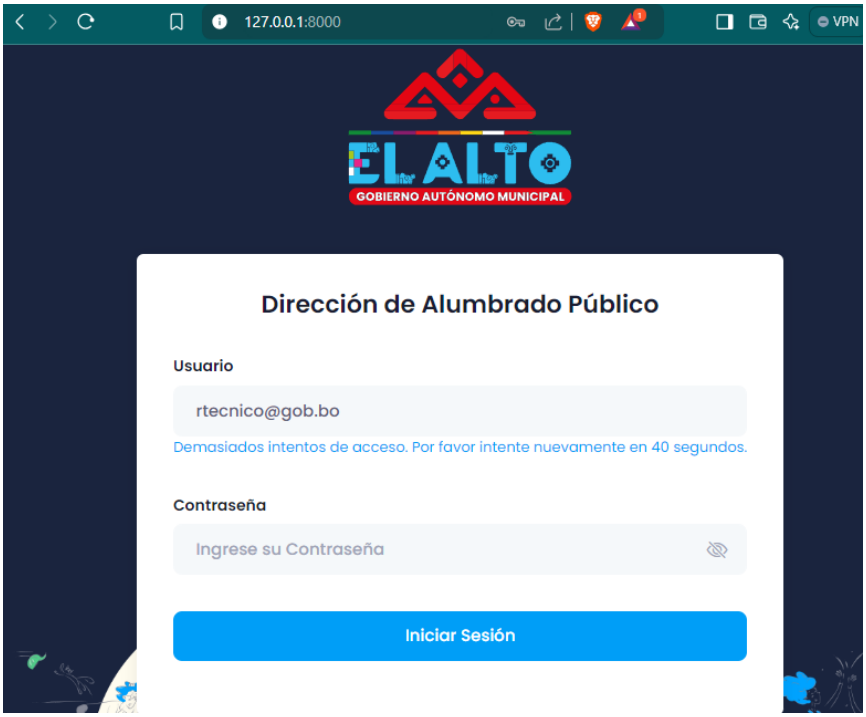


Figura *144.*

Encriptación de contraseñas

password
$2y$12$FucEmGlyNy1q0qUZTylhIOzhnZnV50RGxBk6J/w11rs...
$2y$12$eBFylpAMCKmbql6eka8O4.q8ueyrXkJcdvJ76ja4uG9...
$2y$12$n7nspE78nRBcmyZl6WJjwOSvsTiMGyTTPsR.2MiVlyB...
$2y$12$.Nz8wC8wTe/N2ggqNaev5eseSPdFzrgENgUIDrWEBHs...
$2y$12$V57UmUdBln.vO8ryKo4AGuzUAGMwJPNkJoxRjQj3od1...
$2y$12$TKipLUmUltAeTw6V0Py6S.ivnwhCs0/1DnvCpbcGU71...

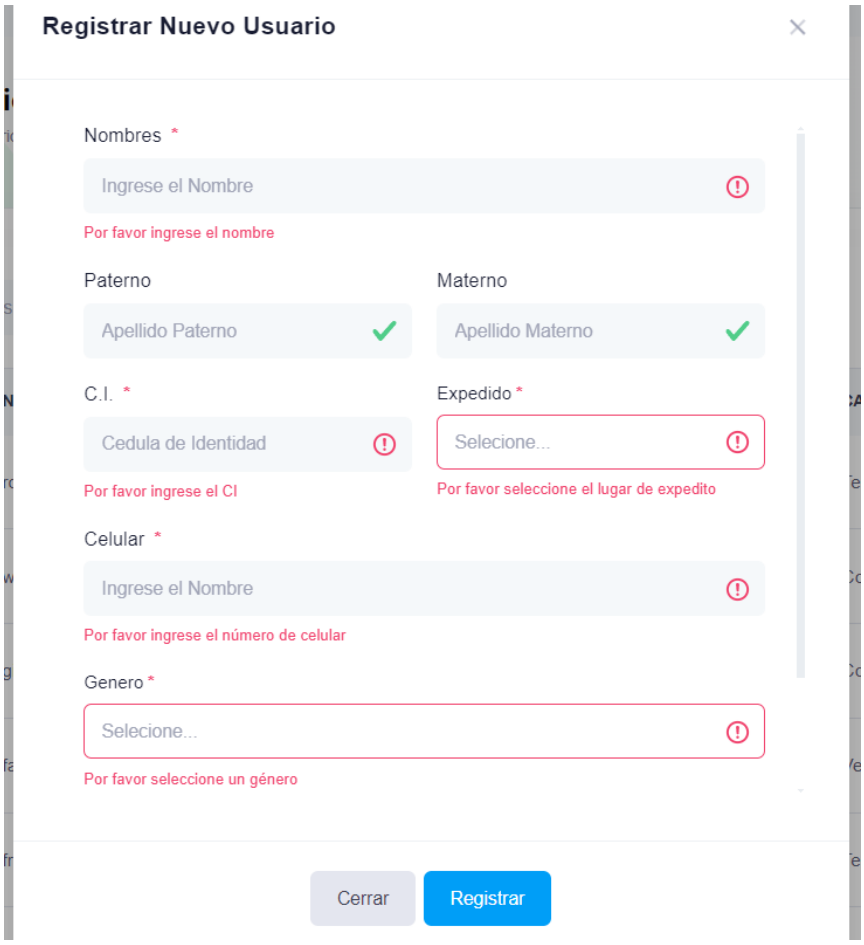
b) Módulo de Usuarios

- Registro de Usuarios
- Validación de campos numéricos

- Validación de campos alfabéticos
- Verificación de campos obligatorios
- Validación de selección de roles
- Prueba de permisos por rol (Administrador, Veedor, Coordinador, Técnico)

Figura *145.*

Validación de Campos Usuarios



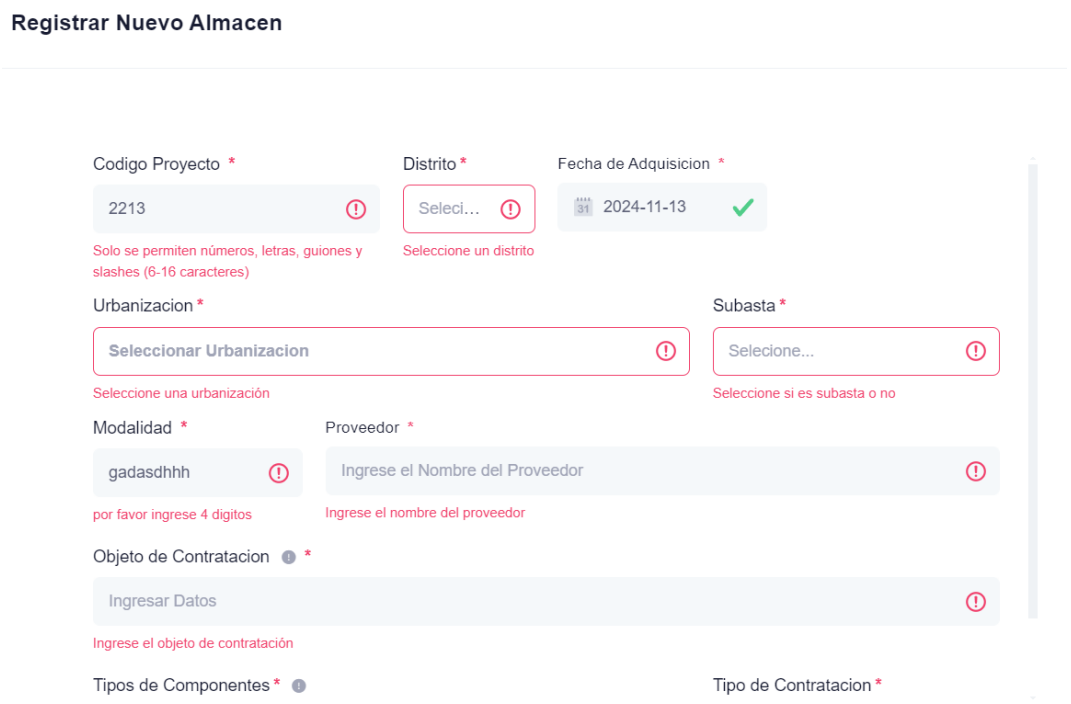
c) Módulo de Proyectos

- Formularios de Registro
- Validación de campos obligatorios
- Verificación de tipos de datos

- Prueba de restricciones de entrada
- Validación de mensajes de error

Figura *146.*

Validación de Campos Proyectos

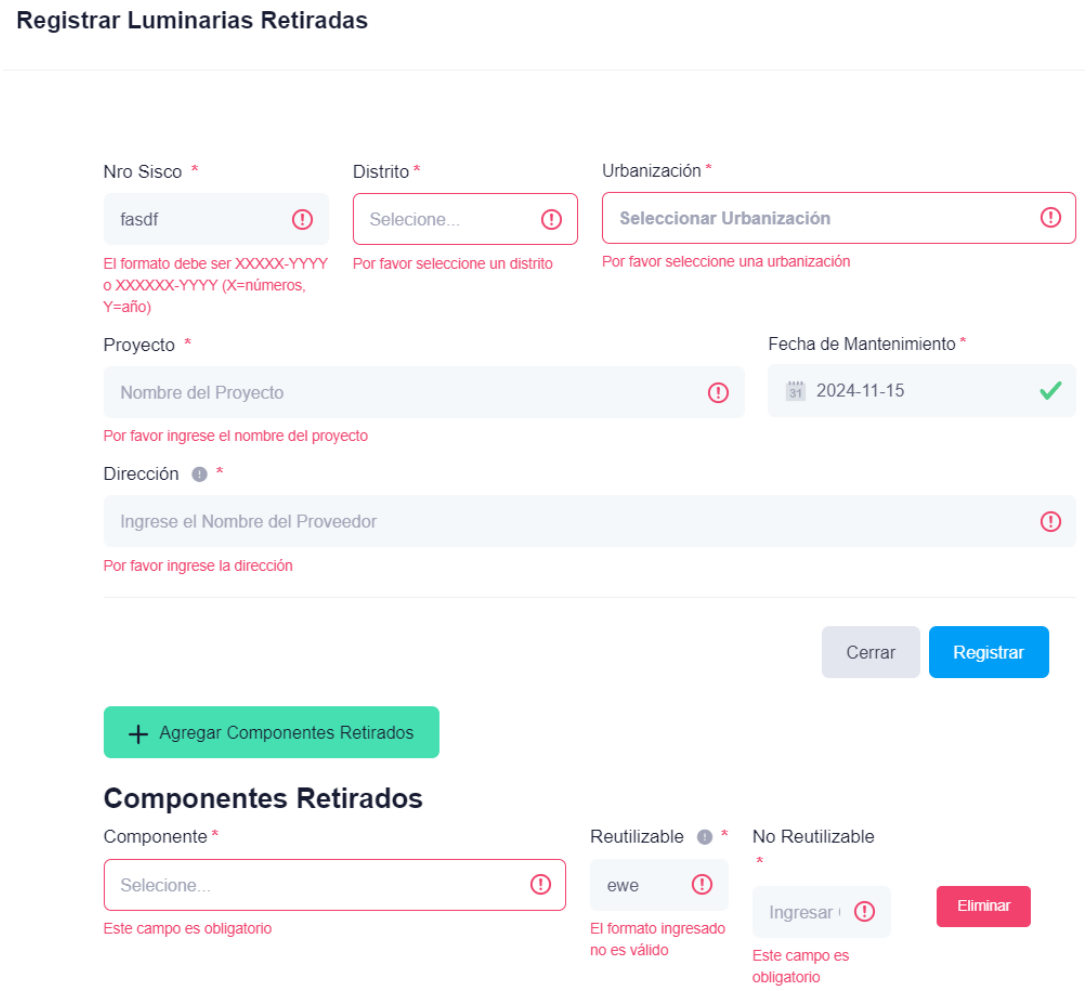


d) Módulo de Luminarias Retiradas

- Gestión de Registros
- Validación de formato de datos
- Prueba de campos obligatorios
- Verificación de relaciones entre registros
- Validación de mensajes de respuesta

Figura *147.*

Validación de Campos Luminarias Retiradas



4.5.4.3. Resultados y Correcciones

- Se identificaron y corrigieron errores en los mensajes de validación
- Se ajustaron las validaciones de entrada en formularios específicos
- Se verificó el correcto funcionamiento después de las correcciones

4.5.5. *Pruebas de Seguridad*

4.5.5.1. Controles Implementados

- Autenticación segura con bloqueo temporal

- Protección contra CSRF
- Encriptación de datos sensibles
- Validación de entradas y salidas

4.5.5.2. **Resultados de Pruebas de Seguridad**

- Se verificó el correcto funcionamiento del bloqueo de cuentas
- Se confirmó la efectividad de la protección CSRF
- Se comprobó la seguridad en el almacenamiento de contraseñas

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

INGENIERÍA
DE SISTEMAS
UNIVERSIDAD PUBLICA DE EL ALTO

5. CAPÍTULO V

5.1. CONCLUSIONES

A través de la implementación el proyecto cumplió satisfactoriamente con el desarrollo del sistema del sistema web, se logró desarrollar un sistema web eficiente para el registro, seguimiento y control de luminarias públicas, materiales y herramientas en la ciudad de El Alto. El sistema cumple con la generación de información confiable y oportuna, lo que ha permitido mejorar la toma de decisiones en la gestión de estos recursos confirma el cumplimiento de este objetivo, obteniendo una evaluación total de 88%. Este resultado indica que el sistema es funcional, confiable, usable, eficiente, mantenible y portátil.

5.1.1. *Conclusiones Objetivos Específicos*

Este objetivo se alcanzó mediante un análisis exhaustivo de los procesos actuales, utilizando entrevistas, revisiones documentales y observaciones directas. facilitó la identificación detallada de los requisitos funcionales y no funcionales del sistema. Gracias a este enfoque, se obtuvo una visión clara y precisa de las necesidades de los usuarios y del entorno, asegurando que el sistema responda de manera óptima a las demandas reales.(Ver anexo c)

Se cumplió este objetivo a través de un análisis profundo utilizando la metodología UWE, lo que permitió definir y estructurar los requerimientos de manera efectiva. Este análisis fue crucial para guiar las etapas de diseño y desarrollo, asegurando que el sistema cubriera las funciones específicas necesarias para la gestión de luminarias públicas, materiales y herramientas.

Con el diseño de una base de datos estructurada y optimizada a través de diagramas ER y casos de uso, y bajo la metodología UWE, se alcanzó este objetivo. La base de datos fue diseñada para soportar de manera efectiva el almacenamiento y la consulta de la

información, asegurando la integridad y accesibilidad de los datos. Esto facilita la administración y consulta de los recursos gestionados, manteniendo la coherencia y disponibilidad de la información.

Este objetivo se logró mediante la creación de reportes automatizados, los cuales incluyen información detallada de cada trabajo, proyecto o luminaria retiradas, así como registros completos de equipamientos y accesorios. Además, se implementó un dashboard que permite visualizar gráficamente la información relevante del sistema, promoviendo la transparencia y facilitando el monitoreo de la gestión municipal. Este enfoque asegura una administración clara y accesible de los datos.

El desarrollo del sistema web, resultó en un incremento notable en la eficiencia y en el mantenimiento, con un puntaje de 96% en eficiencia y 88% en mantenibilidad. Estos resultados reflejan que el sistema es capaz de operar de manera ágil y estable, simplificando las tareas de mantenimiento y optimizando los recursos necesarios para su operación. La evaluación total de 88% en las métricas de calidad asegura que el sistema cumple con los estándares requeridos para su implementación y uso a nivel de la ciudad de El Alto.

5.2. RECOMENDACIONES

A la finalización del presente proyecto se procede a detallar las siguientes recomendaciones

Se recomienda capacitación de usuarios:
Diseñar un plan de capacitación integral que incluya:

- Manuales de usuario paso a paso, tanto en formato digital como impreso.
- Sesiones de entrenamiento presenciales o virtuales para personal administrativo y técnico.
- Materiales de apoyo como guías rápidas de referencia.

- Asegurarse de que todos los usuarios clave, incluyendo personal de campo, reciban la capacitación necesaria.

Copias de seguridad y recuperación de datos:

- Implementar un plan de respaldo de la base de datos con frecuencia mensual.
- Almacenar copias de seguridad en ubicaciones seguras y fuera del sitio principal.

Control de acceso y seguridad:

- Establecer políticas sólidas de gestión de usuarios y contraseñas, como:
    - Longitud mínima, complejidad y renovación periódica.
    - Mantener actualizados los sistemas operativos, frameworks y librerías para corregir vulnerabilidades.

6. **BIBLIOGRÁFIA**

Achour, M., Betz, F., Dovgal, A., & Lopes, N. (2023). PHP Manual. PHP Documentation Group.

Albrecht, A. (1979). Measuring application development productivity. Proceedings of the 1979 International Conference on Software Engineering, 83-92.

Apache Friends. (2023). XAMPP Apache + MariaDB + PHP + Perl. https://www.apachefriends.org/

Bangyal, W. H., Hameed, A., Alosaimi, W., & Alyami, H. (2018). A new initialization approach in particle swarm optimization for global optimization problems. Computational Intelligence and Neuroscience, 2018, 1-15. https://doi.org/10.1155/2018/2048918

Bartholomew, D. (2019). MariaDB cookbook. Packt Publishing.

Boehm, B., Abts, C., & Chulani, S. (2021). Software Cost Estimation with COCOMO II. Addison-Wesley Professional.

Boehm, B., & Turner, R. (2003). Balancing agility and discipline: A guide for the perplexed. Addison-Wesley Professional.

Boyce, P. R. (2014). Human factors in lighting. CRC Press.

Clientify. (2022). Recolección de datos: métodos, técnicas e instrumentos. https://www.clientify.com

Cohn, M. (2022). Agile Estimating and Planning: A Comprehensive Guide to Planning Poker and Other Collaborative Techniques. Addison-Wesley Professional.

Cohn, M. (2023). Agile Estimating and Planning: Innovative Techniques for Software Projects. Addison-Wesley Professional.

Comparasoftware. (2023). 8 Técnicas de Recolección de Datos: Las más utilizadas. https://blog.comparasoftware.com

Daft, R. L., & Marcic, D. (2020). Understanding management (11th ed.). Cengage Learning.

Davenport, T. H. (2018). The AI advantage: How to put the artificial intelligence revolution to work. MIT Press.

Deming, W. E. (2018). The new economics for industry, government, education (3rd ed.). MIT Press.

Djouab, R., & Bari, M. (2020). An ISO/IEC 25010 based quality model for evaluating cloud systems. International Journal of Advanced Computer Science and Applications, 11(7), 741-749.

Duckett, J. (2014). JavaScript and JQuery: Interactive Front-End Web Development. Wiley.

Duckett, J. (2021). HTML and CSS: Design and Build Websites. John Wiley & Sons.

Dvorski, D. D. (2007). Installing, configuring, and developing with XAMPP. Skills Canada.

Erixon, P. (2018). Building custom plugins for WordPress. Apress.

Estdale, J., & Georgiadou, E. (2018). Applying the ISO/IEC 25010 quality models to software product. In Systems, Software and Services Process Improvement (pp. 492-503). Springer.

García-Hernández, R. (2023). Ciberseguridad en la administración pública local. Editorial Tecnos.

García-Mireles, G. A., & Villa-Martínez, H. A. (2023). A metamodel to integrate software quality characteristics based on ISO/IEC 25010. Computer Standards & Interfaces, 92, 103756.

Gauchat, J. D. (2017). HTML5 for masterminds: How to take advantage of HTML5 to create responsive websites and revolutionary applications. Mink Books.

Gobierno Autónomo Municipal de El Alto. (2021). Manual de procedimientos administrativos. El Alto: Alcaldía Municipal.

Grenning, J. (2021). Planning Poker for Improved Project Estimates. Pragmatic Bookshelf.

Griffiths, M. (2022). Agile Project Estimation Techniques: From Planning Poker to Bucket System. O'Reilly Media.

Groover, M. P. (2019). Fundamentals of modern manufacturing: Materials, processes, and systems (7th ed.). John Wiley & Sons.

Heizer, J., Render, B., & Munson, C. (2020). Operations management: Sustainability and supply chain management (13th ed.). Pearson.

International Organization for Standardization. (2019). ISO/IEC 25010:2019 Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model. ISO.

International Organization for Standardization. (2022). ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection. ISO.

Johnson, E. (2022). Mastering Visual Studio Code: A practical guide to streamlining your development workflow. Packt Publishing.

Jorgensen, P. C. (2024). Software Testing: A Craftsman's Approach (5th ed.). CRC Press.

Kua, P. (2023). Practical Estimation Techniques for Agile Projects. O'Reilly Media.

Laudon, K. C., & Laudon, J. P. (2018). Management information systems: Managing the digital firm (15th ed.). Pearson.

Laurie, B., & Laurie, P. (2022). Apache: The definitive guide. O'Reilly Media.

Llaneza, M., Dapozo, G., Greiner, C., & Estayno, M. (2023). Modelo de calidad de producto de software basado en ISO/IEC 25010. In XXV Workshop de Investigadores en Ciencias de la Computación (WICC 2023).

Meyers, S. (2014). Effective C++: 55 specific ways to improve your programs and designs. Addison-Wesley.

Microsoft. (2023). Visual Studio Code Documentation. https://code.visualstudio.com/docs

Myers, G. J., Sandler, C., & Badgett, T. (2021). The art of software testing (3rd ed.). John Wiley & Sons.

Myers, G. J., Sandler, C., & Badgett, T. (2022). The Art of Software Testing. John Wiley & Sons.

Myers, G. J., Sandler, C., & Badgett, T. (2023). The Art of Software Testing (4th ed.). John Wiley & Sons.

Neirotti, P., De Marco, A., Cagliano, A. C., Mangano, G., & Scorrano, F. (2014). Current trends in Smart City initiatives: Some stylised facts. Cities, 38, 25-36.

Oracle Corporation. (2023). MySQL 8.0 Reference Manual. Oracle.

Organización Internacional de Normalización. (2014). ISO/IEC 25000:2014, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE. ISO.

Otto, M., & Thornton, J. (2022). Bootstrap 5 Documentation. Bootstrap Core Team.

Otto, M., & Thornton, J. (2022). Bootstrap 5 Documentation. https://getbootstrap.com/docs/5.0/getting-started/introduction/

Perkovic, L. (2021). Introduction to computing using Python: An application development focus. John Wiley & Sons.

Pressman, R. S., & Maxim, B. R. (2020). Software engineering: A practitioner's approach. McGraw-Hill Education.

Pressman, R. S., & Maxim, B. R. (2024). Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill Education.

Project Management Institute. (2017). A guide to the project management body of knowledge (PMBOK guide) (6th ed.). Project Management Institute.

Robbins, J. N. (2021). Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics. O'Reilly Media.

Robbins, S. P., & Coulter, M. (2021). Management (15th ed.). Pearson.

Rossi, G., Pastor, O., Schwabe, D., & Olsina, L. (2017). Web engineering: Modelling and implementing web applications. Springer.

Rothman, J. (2021). Predicting the Unpredictable: Pragmatic Approaches to Estimating Project Schedule or Cost. Pragmatic Bookshelf.

Rubin, K. S. (2012). Essential Scrum: A practical guide to the most popular agile process. Addison-Wesley Professional.

Schwartz, B., Zaitsev, P., & Tkachenko, V. (2012). High Performance MySQL: Optimization, Backups, and Replication (3rd ed.). O'Reilly Media.

Sikos, L. F. (2020). Web standards: Mastering HTML5, CSS3, and XML. Apress.

Slack, N., Brandon-Jones, A., & Johnston, R. (2019). Operations management (9th ed.). Pearson.

Smyth, N. (2020). Visual Studio Code Distilled: Evolved code editing for Windows, macOS, and Linux. Apress.

Sommerville, I. (2016). Software engineering (10th ed.). Pearson Education Limited.

Spillner, A., Linz, T., & Schaefer, H. (2021). Software Testing Foundations. Rocky Nook.

Spillner, A., Linz, T., & Schaefer, H. (2022). Software Testing Foundations: A Study Guide for the Certified Tester Exam (5th ed.). Rocky Nook.

Stauffer, M. (2019). Laravel: Up & Running: A Framework for Building Modern PHP Apps (2nd ed.). O'Reilly Media.

Stevenson, W. J. (2018). Operations management (13th ed.). McGraw-Hill Education.

Stuttard, D., & Pinto, M. (2021). The web application hacker's handbook: Finding and exploiting security flaws. John Wiley & Sons.

Tatroe, K., MacIntyre, P., & Lerdorf, R. (2020). Programming PHP: Creating dynamic web pages. O'Reilly Media.

Wanyoike, M. (2020). Vue.js 3 by example. Packt Publishing.

Welling, L., & Thomson, L. (2017). PHP and MySQL web development. Addison-Wesley Professional.

Williams, B., Damstra, D., & Stern, H. (2022). Professional WordPress: Design and development. Wrox.

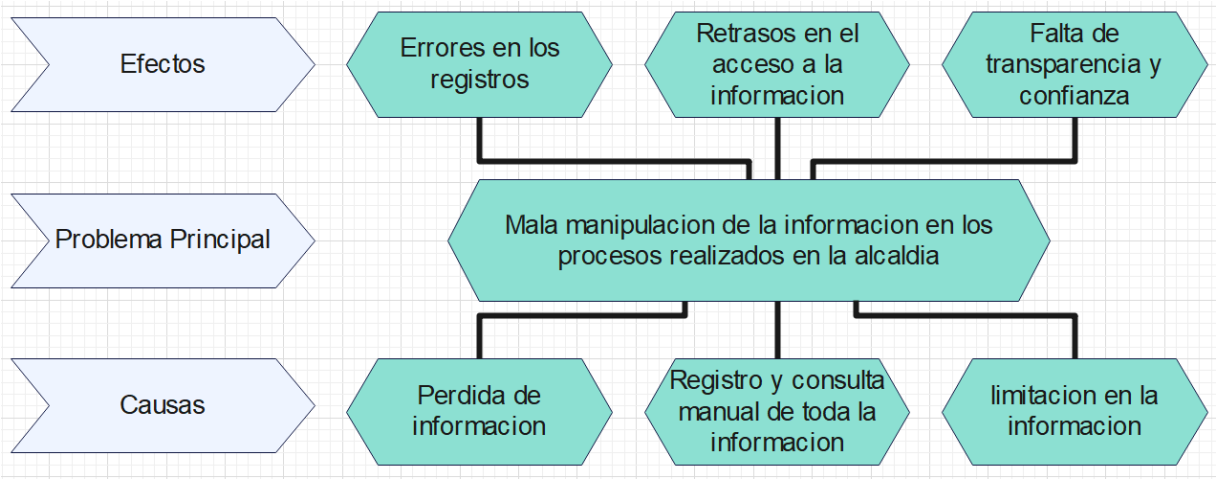
Wilson, J. R. (2021). Node.js design patterns: Design and implement production-grade Node.js applications using proven patterns and techniques. Packt Publishing.

Womack, J. P., & Jones, D. T. (2016). Lean thinking: Banish waste and create wealth in your corporation.

Zakas, N. C. (2019). High performance JavaScript: Build faster web application interfaces. O'Reilly Media.

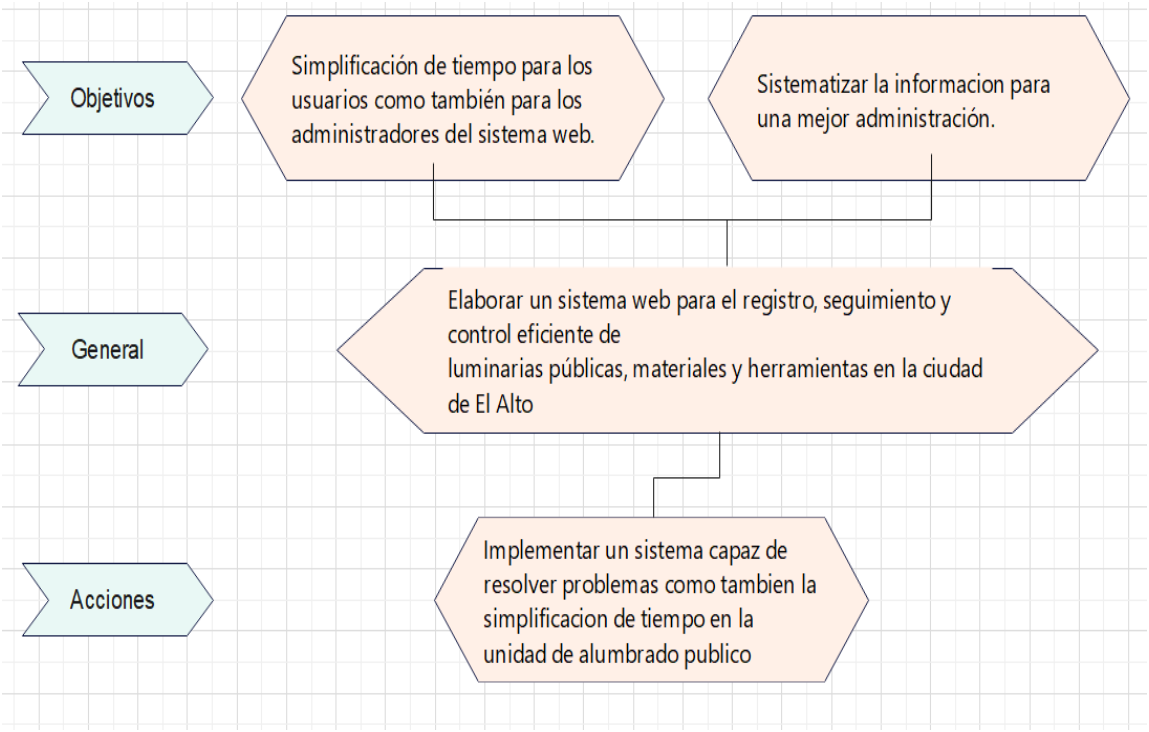
ANEXOS A

Árbol de problemas



ANEXOS B

Árbol de objetivos



ANEXOS C

Diagrama de flujo Administración de Usuario

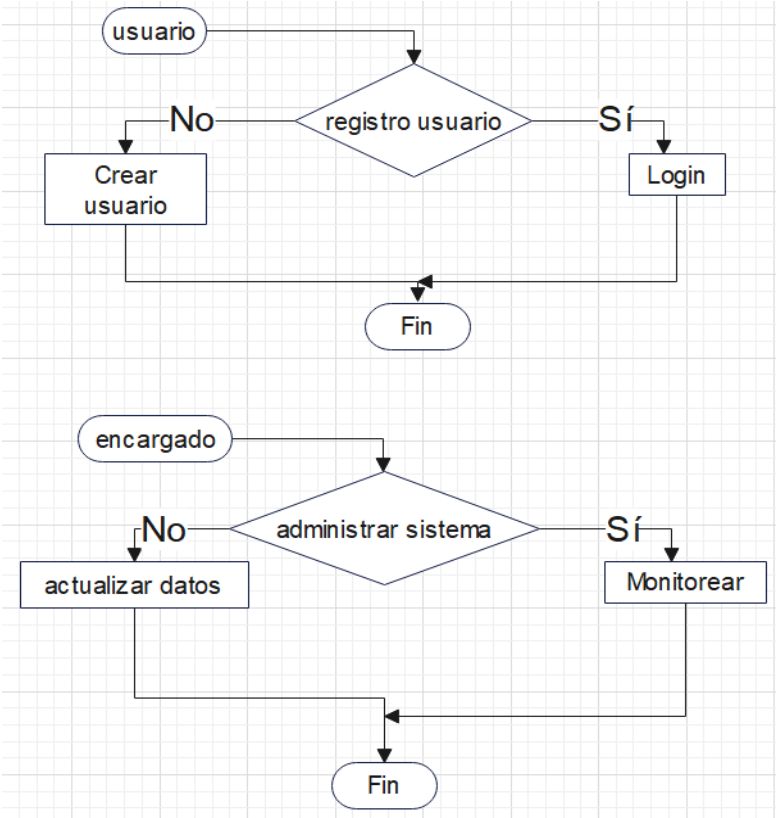
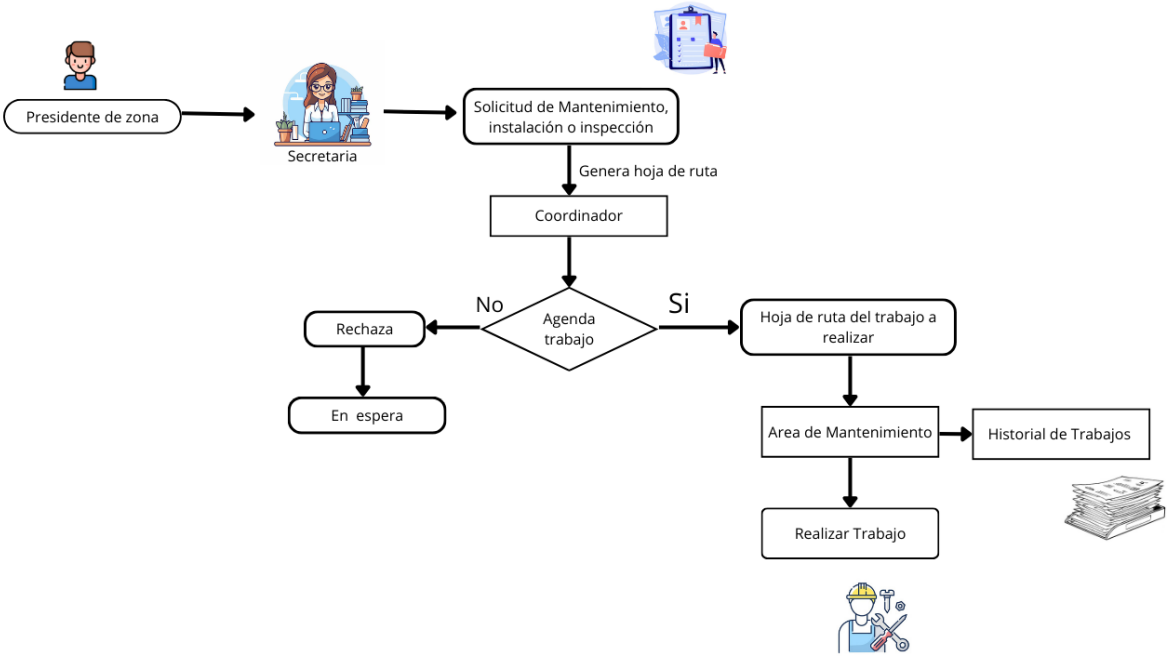
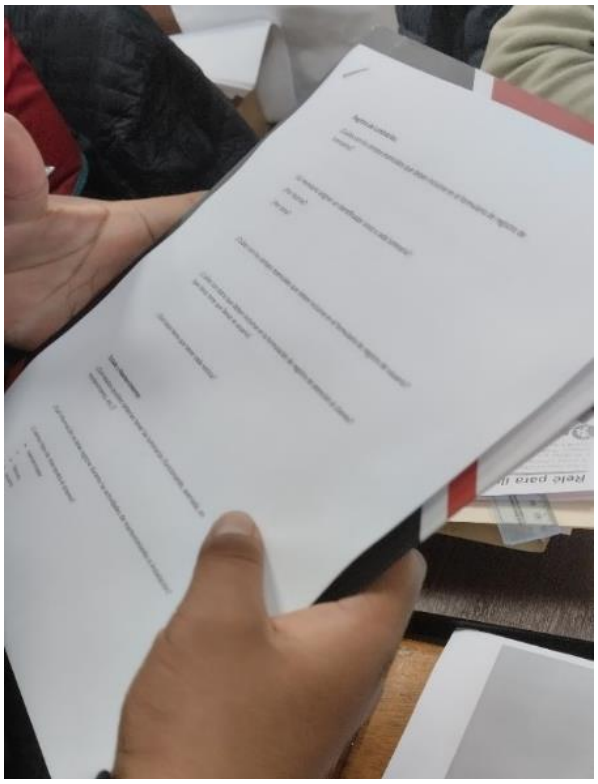


Diagrama de Flujo Trabajos



Encuesta



Nota. Encuesta al jefe de la unidad de alumbrado público para recabar información de los requerimientos necesarios

Recopilación de datos

Nota. Recolección de datos en la dirección *de alumbrado público*

Aval de Conformidad Especialista

AVAL DE CONFORMIDAD

(TUTOR ESPECIALISTA)

El Alto, noviembre de 2024

Señora:
M.Sc. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA WEB PARA EL REGISTRO, SEGUIMIENTO Y CONTROL EFICIENTE DE LUMINARIAS PÚBLICAS, MATERIALES Y HERRAMIENTAS
CASO: GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO G.A.M.E.A
MODALIDAD: PROYECTO DE GRADO
Universitario: Wilson Viracocha Laura
Registro Universitario: 200020708
Cedula de Identidad: 8377140 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al Reglamento Vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,

Lic.Ing. Luis Francisco Páez Rocha
TUTOR ESPECIALISTA

Nota. Aval de conformidad ing. Luis Francisco Páez Rocha

Aval de conformidad Revisor

AVAL DE CONFORMIDAD

(TUTOR REVISOR)

El Alto, noviembre de 2024

Señora:
M.Sc. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: SISTEMA WEB PARA EL REGISTRO, SEGUIMIENTO Y CONTROL EFICIENTE DE LUMINARIAS PÚBLICAS, MATERIALES Y HERRAMIENTAS
CASO: GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO G.A.M.E.A
MODALIDAD: PROYECTO DE GRADO
Universitario: Wilson Viracocha Laura
Registro Universitario: 200020708
Cedula de Identidad: 8377140 LP

Para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al Reglamento Vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,

Lic.Ing. Yuly Ramirez Limachi
TUTOR REVISOR

Aval Institucional

GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO

EL ALTO
GOBIERNO AUTÓNOMO MUNICIPAL

AVAL DE CONFORMIDAD

El Alto, Noviembre de 2024

Señor:
M.Sc.Lic.Ing. Marisol Arguedas Balladares
TUTOR METODOLÓGICO
TALLER DE GRADO II
Presente. –

REF. AVAL DE CONFORMIDAD

Distinguido tutor metodológico:
Mediante la presente tengo a bien comunicarle mi conformidad del Trabajo de Grado:

TITULO: **"SISTEMA WEB PARA EL REGISTRO, SEGUIMIENTO Y CONTROL EFICIENTE DE LUMINARIAS PÚBLICAS, MATERIALES Y HERRAMIENTAS"**
CASO: **GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO "G.A.M.E.A."**
MODALIDAD: **PROYECTO DE GRADO**
Univ. **WILSON VIRACOCHA LAURA**
Registro Universitario: **200020708**
Cedula de Identidad: **8377140 L.P.**

De tal forma cabe recalcar que el **PROYECTO** satisface los requerimientos de la institución, de esta forma se dio cumplimiento de los objetivos del presente.

El presente **PROYECTO** fue **IMPLEMENTADO** en el **GOBIERNO AUTÓNOMO MUNICIPAL DE EL ALTO** satisfactoriamente y se realizó las capacitaciones necesarias en la institución.

En cuanto certifico, en honor a la verdad, para fines consiguientes del interesado para su defensa pública y evaluación correspondiente a la materia de Taller de Grado II, de acuerdo al reglamento vigente de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

Atentamente,

Ing. Gerardo Valdez Balcazar
JEFE
EL ALTO UNIDAD DE SISTEMAS
GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO

DIRECCION ADMINISTRATIVA
UNIDAD DE SISTEMAS
GOBIERNO AUTONOMO MUNICIPAL DE EL ALTO
El Alto - Bolivia