



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E INGENIERÍA

**LICENCIATURA EN CIENCIAS COMPUTACIONALES**

**TESIS**

**SISTEMA DE INVENTARIO PARA EL ÁREA ACADÉMICA  
DE INGENIERÍA Y ARQUITECTURA: AIRONE**

Para obtener el título de

Licenciado en Ciencias Computacionales

**PRESENTA**

Diego Ángel Barajas Pérez

**Directora**

M.I.D. Norma Laura Salazar Viveros

**Comité Tutorial:**

Dra. Theira Irasema Samperio Monroy

Dra. Verónica Martínez Lazcano

Dra. Martha Idalid Rivera González

Pachuca de Soto, Hgo., México., Julio 2025



Mineral de la Reforma, Hgo., a 9 de julio de 2025

Número de control: ICBI-D/1272/2025

Asunto: Autorización de impresión.

**MTRA. OJUKY DEL ROCÍO ISLAS MALDONADO  
 DIRECTORA DE ADMINISTRACIÓN ESCOLAR DE LA UAEH**

Con Título Quinto, Capítulo II, Capítulo V, Artículo 51 Fracción IX del Estatuto General de nuestra Institución, por este medio, le comunico que el Jurado asignado al egresado de la Licenciatura en Ciencias Computacionales **Diego Ángel Barajas Pérez**, quien presenta el trabajo de titulación “**Sistema de inventario para el Área Académica de Ingeniería y Arquitectura: Airone**”, ha decidido, después de revisar fundamento en lo dispuesto en el Título Tercero, Capítulo I, Artículo 18 Fracción IV; dicho trabajo en la reunión de sinodales, **autorizar la impresión del mismo**, una vez realizadas las correcciones acordadas.

A continuación, firman de conformidad los integrantes del Jurado:

**Presidente:** Dra. Verónica Martínez Lazcano

**Secretario:** Dra. Theira Irasema Samperio Monroy

**Vocal:** M.I.D. Norma Laura Salazar Viveros

**Suplente:** Dra. Martha Idalid Rivera González

Sin otro particular por el momento, reciba un cordial saludo.

Atentamente  
 “Amor, Orden y Progreso”

Mtro. Gabriel Vergara Rodríguez  
 Director del ICBI

GVR/YCC



Ciudad del Conocimiento, Carretera Pachuca-Tulancingo Km. 4.5 Colonia Carboneras, Mineral de la Reforma, Hidalgo, México. C.P. 42184  
 Teléfono: 771 71 720 00 Ext. 40001  
 direccion\_icbi@uaeh.edu.mx, vergarar@uaeh.edu.mx

“Amor, Orden y Progreso”



## Agradecimientos

---

Me gustaría iniciar este apartado otorgando mis mayores agradecimientos a mis padres, Durbin Barajas López y Coynta Verenice Pérez Sosa, su apoyo constante e incondicional fue el elemento preponderante para culminar mis estudios universitarios. Jamás olvidaré todo lo que han hecho por mí. *¡Gracias por todo!*

También me gustaría otorgar un agradecimiento a mi hermana, Alejandra Berenice Barajas Pérez, tus consejos, bromas repentinas y constante pendiente a mi bienestar son cosas que atesoraré con absoluto cariño, gracias por estar presente en esta etapa tan importante de mi vida.

Para todas las amistades que hice durante esta etapa, les agradezco por permitirme conocer un poco más acerca de cada uno de ustedes fuera de la rutina, siempre conservaré una parte de ustedes en forma de chistes, frases, platillos, música, libros, lugares, conocimiento, y mucho, pero mucho más.

Finalmente, deseo dedicar un gran agradecimiento a mi asesora de tesis, la maestra Norma Laura Salazar Viveros, por su constante apoyo, retroalimentación y consejo, los cuales, fueron fundamentales para culminación de este trabajo, gracias por ser mi docente, tutora y asesora.

*The Best Is Yet To Come...*

## Resumen

---

El desarrollo de sistemas web ha sido un ámbito que ha adquirido una gran importancia desde sus orígenes en la década de los 90's, reemplazando en gran medida la manera en que consumimos software, la creciente accesibilidad de Internet, así como el desarrollo de herramientas enfocadas al desarrollo web ha facilitado la difusión de esta clase de sistemas dentro de una gran variedad de ámbitos como lo puede ser la educación, gestión empresarial, venta, comunicación e infraestructura, entre otros ámbitos.

El presente trabajo muestra el proceso de desarrollo de un sistema web, cuyo propósito es el de cubrir y facilitar las actividades de gestión y consulta de información de inventario, así como clasificar los principales tipos de elementos y bienes con los que cuenta el Área Académica de Ingeniería y Arquitectura de la Universidad Autónoma del Estado de Hidalgo, haciendo uso del Framework de desarrollo de aplicaciones web basado en PHP, Laravel, en conjunto a otras tecnologías afines.

El proceso de desarrollo siguió los principios de la metodología ágil Programación Extrema (XP), dada la compatibilidad con el flujo de trabajo que se lleva a cabo dentro del Área Académica de Ingeniería y Arquitectura, contando con lapsos de entrega cortos y una comunicación frecuente. El resultado del proceso de desarrollo fue un sistema de inventario polivalente, cubriendo con todos los requisitos solicitados, además de implementar funcionalidades para la difusión de eventos e importación de información proveniente de hojas de cálculo.

De acuerdo a las pruebas de usabilidad, donde se evaluaron distintos parámetros como la facilidad de uso, diseño y accesibilidad del sistema, se puede concluir que este desarrollo es una herramienta eficaz para el objetivo que le fue planteado, siendo capaz de proporcionar una experiencia intuitiva para la mayoría de usuarios, contando con el potencial de ser utilizado y/o adaptado para otros ámbitos y áreas ajenas a donde este desarrollo fue planteado originalmente.

## Abstract

---

The development of web systems is a field which has been acquiring a great importance since its origins on the decade of the 90's, replacing into a large extent the way we consume software, the growing accessibility of internet, and the development of web development focused tools have made easier the diffusion of this sort of systems inside many areas such as education, communications and infrastructure, among other areas.

This document presents the development process of a web system with the purpose of cover and simplify the management and data consulting activities of the “Área Académica de Ingeniería y Arquitectura” of the “Universidad Autónoma del Estado de Hidalgo”, this system does also contain the necessary tools for elements and goods classification of the academic area, making use of the PHP based development Framework Laravel, along with a set of related tools.

The development process followed applied the principles established on the Extreme Programming (XP) software development methodology due to the favorable compatibility with the “Área Académica de Ingeniería y Arquitectura” personal's workflow, short-term progress deliveries and frequent communication. The product of the development process was a polyvalent inventory system, which covers the entirety of the original requirements along with event diffusion and datasheets importing functionalities.

According to the usability tests, where a set of parameters such as ease of use, design and accessibility were evaluated, it can be concluded that the developed system is an efficient tool for its purpose for which it was intended, being capable of provide an intuitive experience for the majority of users, with the potential of being used and/or adopted for other areas and use cases beyond it was originally planned.

# Índice de Contenido

---

<i>Introducción</i> .....	1
<i>Antecedentes</i> .....	2
<i>Planteamiento del problema</i> .....	3
<i>Propuesta de solución</i> .....	4
<i>Justificación</i> .....	5
<i>Objetivo General</i> .....	6
<i>Objetivos Específicos</i> .....	6
<i>Metodologías</i> .....	7
<i>Herramientas de desarrollo</i> .....	10
<i>Alcance</i> .....	11
<b>Capítulo 1. Marco Teórico</b> .....	12
1.1 <i>Familia de protocolos de Internet (TCP/IP) y La World Wide Web</i> .....	12
1.1.1 <i>Modelo de referencia OSI y TCP/IP</i> .....	12
1.1.2 <i>World Wide Web</i> .....	13
1.1.3 <i>Protocolo HTTP y HTTPS</i> .....	14
1.2 <i>Conceptos de desarrollo Web</i> .....	15
1.2.1 <i>Aplicación web</i> .....	15
1.2.2 <i>Patrón de diseño MVC</i> .....	16
1.2.3 <i>Framework</i> .....	17
1.2.4 <i>Gestor de dependencias</i> .....	17
1.2.5 <i>Bases de datos SQL y NoSQL</i> .....	18
1.2.6 <i>LAMP</i> .....	20
1.2.7 <i>API</i> .....	22
1.3 <i>Herramientas de desarrollo Web</i> .....	23
1.4 <i>Marco Metodológico</i> .....	34
1.5 <i>Inventario y administración educativa</i> .....	39
1.5.1 <i>Inventario</i> .....	40
1.5.2 <i>Sistema de inventario</i> .....	40
1.5.3 <i>Administración Educativa</i> .....	41
1.6 <i>Estado del arte</i> .....	42
1.6.1 <i>Holded</i> .....	42
1.6.2 <i>Odo</i> .....	42
1.6.3 <i>Skyware Inventory</i> .....	43
1.6.4 <i>Monstock</i> .....	43
1.6.5 <i>Zoho Inventory</i> .....	44

<b>Capítulo 2 . Metodología</b> .....	46
2.1 Planeación .....	46
2.1.1 Definición de las funcionalidades principales .....	55
2.1.2 Determinación de iteraciones para el desarrollo del Sistema de Inventario Airone.....	56
2.2 Diseño.....	59
2.2.1 Diseño de la base de datos .....	59
2.2.2 Sitemap .....	62
2.2.3 Backend .....	64
2.2.4 Frontend .....	70
2.3 Codificación .....	77
2.4 Pruebas.....	90
<b>Capítulo 3 . Resultados</b> .....	97
3.1 Muestra del sistema final.....	97
<b>Capítulo 4. Manual de usuario</b> .....	121
<b>Capítulo 5. Validación/Pruebas de usabilidad</b> .....	147
Conclusiones.....	155
Trabajo futuro .....	156
Bibliografía y referencias.....	157

# Índice de Figuras

---

## **Capítulo 1. Marco Teórico**

<i>Figura 1. 1. Conjunto de protocolos TCP/IP en relación con el modelo de OSI de referencia</i>	13
<i>Figura 1. 2. Proceso de Comunicación por medio de HTTP</i>	14
<i>Figura 1. 3. Proceso de Encriptación Asimétrica. Adaptado de Algoritmo Asimétrico</i>	15
<i>Figura 1. 4. Organización de la Arquitectura MVC</i>	16
<i>Figura 1. 5. Proceso de comunicación de una API. Adaptado de API</i>	22
<i>Figura 1. 6. Logotipo de HTML</i>	23
<i>Figura 1. 7. Logotipo de CSS</i>	24
<i>Figura 1. 8. Logotipo de JavaScript</i>	25
<i>Figura 1. 9. Logotipo de PHP</i>	26
<i>Figura 1. 10. Logotipo de Laravel</i>	27
<i>Figura 1. 11. Logotipo de Composer</i>	28
<i>Figura 1. 12. Logotipo de Livewire</i>	28
<i>Figura 1. 13. Logotipo de jQuery</i>	29
<i>Figura 1. 14. Logotipo de TailwindCSS</i>	30
<i>Figura 1. 15. Logotipo de Laravel Excel</i>	30
<i>Figura 1. 16. Logotipo de FullCalendar</i>	31
<i>Figura 1. 17. Logotipo de Apache HTTP Server</i>	32
<i>Figura 1. 18. Logotipo de MySQL</i>	32
<i>Figura 1. 19. Logotipo de Git</i>	33
<i>Figura 1. 20. Arquitectura de la Metodología SOHDM</i>	35
<i>Figura 1. 21. Arquitectura de la Metodología en Espiral</i>	36
<i>Figura 1. 22. Arquitectura de la Metodología XP</i>	38

## **Capítulo 2. Metodología**

<i>Figura 2. 1. Diagrama Entidad-Relación del Sistema de Inventario Airone</i>	60
<i>Figura 2. 2. Diagrama Relacional del Sistema de Inventario Airone</i>	61
<i>Figura 2. 3. Sitemap del Sistema de Inventario Airone</i>	63
<i>Figura 2. 4. Diagrama de casos de uso del proceso de Login y Landing del sistema</i>	64
<i>Figura 2. 5. Diagrama de casos de uso del proceso de Login y Landing del sistema</i>	64
<i>Figura 2. 6. Diagrama de casos de uso del Calendario del sistema y Difusión de mensajes</i>	65
<i>Figura 2. 7. Diagrama de casos de uso de la generación de registros para la administración del sistema de Inventario Airone (Categorías, Etiquetas y Fondos)</i>	66
<i>Figura 2. 8. Diagrama de casos de uso del proceso de administración de elementos de inventario, trasposos y mantenimientos</i>	67
<i>Figura 2. 9. Diagrama DFD del proceso de importación de registros de inventario por medio de un archivo XLSX del Sistema de Inventario Airone</i>	69

<i>Figura 2. 10. Prototipo de vista de Login.</i> .....	70
<i>Figura 2. 11. Prototipo de vista de registro.</i> .....	71
<i>Figura 2. 12. Prototipo de vista de recuperación de contraseña.</i> .....	71
<i>Figura 2. 13. Prototipo de vista de configuración de cuenta.</i> .....	72
<i>Figura 2. 14. Prototipo de vista de Landing.</i> .....	72
<i>Figura 2. 15. Prototipo de vista de Dashboard.</i> .....	73
<i>Figura 2. 16. Prototipo de vista de listado de elementos.</i> .....	73
<i>Figura 2. 17. Prototipo de vista de creación y edición de registro de elemento.</i> .....	73
<i>Figura 2. 18. Prototipo de vista de listado de atributos para la administración del sistema.</i> .....	74
<i>Figura 2. 19. Prototipo de vista de creación de nuevo registro de atributo para la administración del sistema.</i> .....	74
<i>Figura 2. 20. Prototipo de vista de listado de usuarios.</i> .....	74
<i>Figura 2. 21. Prototipo de vista de asignación de roles a usuario.</i> .....	75
<i>Figura 2. 22. Prototipo de vista de listado de roles.</i> .....	75
<i>Figura 2. 23. Prototipo de vista de creación y edición de rol.</i> .....	76
<i>Figura 2. 24. Prototipo de calendario de actividades.</i> .....	76
<i>Figura 2. 25. Función para la creación de registro de categorías.</i> .....	77
<i>Figura 2. 26. Función para la actualización de un registro de categorías.</i> .....	78
<i>Figura 2. 27. Función para la eliminación de un registro de categorías.</i> .....	78
<i>Figura 2. 28. Función para la creación de un rol.</i> .....	79
<i>Figura 2. 29. Función para la actualización de un rol.</i> .....	79
<i>Figura 2. 30. Función para la eliminación de un rol.</i> .....	80
<i>Figura 2. 31. Función para la asignación de roles a un usuario.</i> .....	80
<i>Figura 2. 32. Función para creación de registro de elemento.</i> .....	80
<i>Figura 2. 33. Función para actualización de registro de elemento.</i> .....	81
<i>Figura 2. 34. Funciones dedicadas para la actualización de registros de atributos adicionales y generación de registros de traspasos.</i> .....	82
<i>Figura 2. 35. Función para la eliminación de un registro de elemento.</i> .....	82
<i>Figura 2. 36. Lógica para la creación de un registro de traspaso.</i> .....	83
<i>Figura 2. 37. Función para la creación de un registro de mantenimiento.</i> .....	83
<i>Figura 2. 38. Función para la exportación de un registro de mantenimiento a formato PDF.</i> .....	84
<i>Figura 2. 39. Función para la exportación de registros de elementos de inventario por XLSX.</i> .....	85
<i>Figura 2. 40. Función para la importación de registros de elementos de inventario por XLSX.</i> .....	85
<i>Figura 2. 41. Función para la Exportación de la base de datos en formato SQL.</i> .....	86
<i>Figura 2. 42. Función para el envío de notificaciones al canal de Telegram.</i> .....	87
<i>Figura 2. 43. Función para la administración del calendario de eventos.</i> .....	88
<i>Figura 2. 44. Lógica para la generación de CRON Tasks de notificaciones de eventos.</i> ....	89

### **Capítulo 3. Resultados**

<i>Figura 3. 1. Login. Vista Principal.</i>	97
<i>Figura 3. 2. Login. Restauración de contraseña.</i>	98
<i>Figura 3. 3. Login. Registro de cuenta.</i>	98
<i>Figura 3. 4 Landing. Vista Principal.</i>	99
<i>Figura 3. 5. Landing. Configuración de Cuenta, Actualización de datos.</i>	99
<i>Figura 3. 6. Landing. Configuración de Cuenta, Autenticación, Sesiones y Eliminación de cuenta.</i>	100
<i>Figura 3. 7. Landing. Acerca De.</i>	100
<i>Figura 3. 8. Dashboard. Estadísticas y Vista principal.</i>	101
<i>Figura 3. 9. Calendario de Eventos. Creación de Evento.</i>	102
<i>Figura 3. 10. Calendario de Eventos. Actualización de Evento.</i>	102
<i>Figura 3. 11. Calendario de Eventos. Eliminación de Evento.</i>	103
<i>Figura 3. 12. Administración del Sistema. Listado de Categorías.</i>	104
<i>Figura 3. 13. Administración del Sistema. Creación de Categoría.</i>	104
<i>Figura 3. 14. Administración del Sistema. Edición de Categoría.</i>	105
<i>Figura 3. 15. Administración del Sistema. Listado de Etiquetas.</i>	106
<i>Figura 3. 16. Administración del Sistema. Creación de Categoría.</i>	106
<i>Figura 3. 17. Administración del Sistema. Edición de Etiqueta.</i>	107
<i>Figura 3. 18. Administración del sistema. Listado de fondos.</i>	107
<i>Figura 3. 19. Administración del Sistema. Creación de Fondo.</i>	108
<i>Figura 3. 20. Administración del Sistema. Edición de Fondo.</i>	108
<i>Figura 3. 21. Administración de Elementos. Listado y Buscador de elementos.</i>	109
<i>Figura 3. 22. Administración de Elementos. Detalles de Registro de Elemento.</i>	110
<i>Figura 3. 23. Administración de Elementos. Creación de Elemento.</i>	110
<i>Figura 3. 24. Administración de Elementos. Edición de Elemento.</i>	111
<i>Figura 3. 25. Administración de Elementos. Registro de Traspasos.</i>	111
<i>Figura 3. 26. Administración de Elementos. Creación de reporte de Mantenimiento.</i>	112
<i>Figura 3. 27. Administración de Elementos. Listado de reporte de Mantenimiento.</i>	112
<i>Figura 3. 28. Administración de Elementos. Exportación de Registro de Mantenimiento.</i>	113
<i>Figura 3. 29. Usuarios y Roles. Listado de Usuarios.</i>	114
<i>Figura 3. 30. Usuarios y Roles. Asignación de Roles a Usuario.</i>	114
<i>Figura 3. 31. Usuarios y Roles. Listado de Roles.</i>	115
<i>Figura 3. 32. Usuarios y Roles. Creación de Rol.</i>	115
<i>Figura 3. 33. Usuarios y Roles. Edición de Rol.</i>	116
<i>Figura 3. 34. Exportación e Importación de Datos. Vista principal.</i>	117
<i>Figura 3. 35. Exportación e Importación de Datos. Respaldo de la base de datos en formato SQL.</i>	117
<i>Figura 3. 36. Exportación e Importación de Datos. Exportación de Registros de Elementos en formato XLSX.</i>	118

<i>Figura 3. 37. Exportación e Importación de Datos. Vista de Importación de Registros. ..</i>	118
<i>Figura 3. 38. Exportación e Importación de Datos. Importación Exitosa. ....</i>	119
<i>Figura 3. 39. Exportación e Importación de Datos. Error de Formato de documento. ....</i>	119
<i>Figura 3. 40. Exportación e Importación de Datos. Error de Formato de Archivo. ....</i>	119
<i>Figura 3. 41. Centro de Notificaciones. Vista principal. ....</i>	120
<i>Figura 3. 42. Centro de Notificaciones. Envío Exitoso de Mensaje. ....</i>	120
<i>Figura 3. 43. Centro de Notificaciones. Mensaje difundido en Canal de Telegram. ....</i>	120
<b>Capítulo 5. Validación/Pruebas de Usabilidad</b>	
<i>Figura 5. 1. Gráfica de Pastel, Pregunta 1. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	147
<i>Figura 5. 2. Gráfica de Pastel, Pregunta 2. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	148
<i>Figura 5. 3. Gráfica de Pastel, Pregunta 3. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	148
<i>Figura 5. 4. Gráfica de Pastel, Pregunta 4. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	149
<i>Figura 5. 5. Gráfica de Pastel, Pregunta 5. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	149
<i>Figura 5. 6. Gráfica de Pastel, Pregunta 6. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	150
<i>Figura 5. 7. Gráfica de Pastel, Pregunta 7. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	150
<i>Figura 5. 8. Gráfica de Pastel, Pregunta 8. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	151
<i>Figura 5. 9. Gráfica de Pastel, Pregunta 9. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	151
<i>Figura 5. 10. Gráfica de Pastel, Pregunta 10. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	152
<i>Figura 5. 11. Gráfica de Pastel, Pregunta 11. Prueba de Usabilidad. Sistema de Inventario Airone. ....</i>	152

## Índice de Tablas

---

### **Protocolo**

*Tabla 1. Ventajas y desventajas entre las metodologías SOHDM, Espiral y XP..... 8*

### **Capítulo 1. Marco Teórico**

*Tabla 1. 1 Comparativa entre metodologías de desarrollo de Software..... 39*

*Tabla 1. 2. Comparación De Características entre Software de Inventario..... 45*

### **Capítulo 2. Metodología**

*Tabla 2. 1. Permisos y Funciones del Sistema de Inventario Airone ..... 68*

*Tabla 2. 2. Prueba 1. Comprobación de funcionalidad de Administración de Categorías, Fondos y Etiquetas. .... 90*

*Tabla 2. 3. Prueba 2. Comprobación de funcionalidad de Administración de Registros de Elementos de Inventario. .... 91*

*Tabla 2. 4. Prueba 3. Comprobación de funcionalidad de Generación y Exportación de Reportes de Mantenimiento. .... 92*

*Tabla 2. 5. Prueba 4. Comprobación de funcionalidad de Generación automática de Registros de Traspaso. .... 93*

*Tabla 2. 6. Prueba 5. Comprobación de funcionalidad de Exportación e Importación de Registros De Elementos de Inventario a través de un archivo XLSX. .... 94*

*Tabla 2. 7. Prueba 6. Comprobación de funcionalidad de Creación y Asignación de Roles a Usuarios del sistema..... 95*

*Tabla 2. 8. Prueba 7. Comprobación de funcionalidad de Calendario Interactivo de Eventos y Notificación por Canal de Telegram. .... 96*

### **Capítulo 5. Validación/Pruebas de Usabilidad**

*Tabla 5. 1. Cambios de valores, Pregunta 10. Prueba de Usabilidad. Sistema de Inventario Airone ..... 153*

*Tabla 5. 2 Concentrado de respuestas, Sumatorias y Promedios. Prueba de Usabilidad. Sistema de Inventario Airone ..... 153*

*Tabla 5. 3 Concentrado de Sumatorias y Promedios Finales. Prueba de Usabilidad. Sistema de Inventario Airone ..... 154*

# Índice de Tarjetas

---

## **Capítulo 2. Metodología**

<i>Tarjeta 2. 1. Historia de Usuario 0. Sistema de Administración de Elementos, Eventos y Mantenimientos del AAIyA. ....</i>	<i>47</i>
<i>Tarjeta 2. 2 Historia de Usuario 1. Administración de Categorías, Fondos y Etiquetas. ...</i>	<i>48</i>
<i>Tarjeta 2. 3. Historia de Usuario 2. Administración de Registros de Elementos de Inventario. ....</i>	<i>48</i>
<i>Tarjeta 2. 4. Historia de Usuario 3. Creación de Reportes de Mantenimiento Exportables a PDF. ....</i>	<i>49</i>
<i>Tarjeta 2. 5. Historia de Usuario 4. Generación automática de registros de Traspasos. ...</i>	<i>49</i>
<i>Tarjeta 2. 6. Historia de Usuario 5. Exportación e Importación de Registros De Elementos a través de un archivo XLSX. ....</i>	<i>50</i>
<i>Tarjeta 2. 7. Historia de Usuario 6. Creación y Asignación de Roles a Usuarios del sistema. ....</i>	<i>50</i>
<i>Tarjeta 2. 8. Historia de Usuario 7. Calendario Interactivo de Creación de Eventos y Notificación por Canal de Telegram. ....</i>	<i>51</i>
<i>Tarjeta 2. 9. CRC para la clase Elemento.....</i>	<i>52</i>
<i>Tarjeta 2. 10. CRC para la clase Fondo .....</i>	<i>52</i>
<i>Tarjeta 2. 11. CRC para la clase Categoría .....</i>	<i>52</i>
<i>Tarjeta 2. 12. CRC para la clase Etiqueta .....</i>	<i>53</i>
<i>Tarjeta 2. 13. CRC para la clase Traspaso.....</i>	<i>53</i>
<i>Tarjeta 2. 14. CRC para la clase Calendario .....</i>	<i>53</i>
<i>Tarjeta 2. 15. CRC para la clase Notificación .....</i>	<i>53</i>
<i>Tarjeta 2. 16. CRC para la clase Exportación e Importación .....</i>	<i>53</i>
<i>Tarjeta 2. 17. CRC para la clase Mantenimiento .....</i>	<i>54</i>
<i>Tarjeta 2. 18. CRC para la clase Usuario.....</i>	<i>54</i>
<i>Tarjeta 2. 19. CRC para la clase Rol .....</i>	<i>54</i>

# Introducción

---

Internet ha sido una herramienta que ha generado un gran impacto en la sociedad, desde su origen hasta la actualidad (ITU, 2024), su adopción exponencial ha derivado en la generación de una vasta selección de servicios y herramientas con distintos propósitos y funcionalidades, cubriendo las actividades más sencillas como el tiempo, correo electrónico, transferencia de archivos, chat, video y entretenimiento, entre muchas otras más.

Con base a las tendencias y crecimiento natural del ecosistema de Internet, se puede afirmar que uno de los servicios más fundamentales es, sin lugar a dudas, el protocolo *HTTP* (*HyperText Transfer Protocol*, también denominado Protocolo de Transferencia de Hipertexto), en conjunto a todas las tecnologías que se han derivado y/o proliferado en esta (Cruz et al., 2022).

*HTTP* es un protocolo el cual describe la especificación para la comunicación entre un cliente y servidor para la transmisión de documentos de hipermedia, este protocolo en conjunto a otros como *TCP* y *DNS*, entre otros han marcado una base para el desarrollo de lo que es conocido como la *World Wide Web* (*WWW*), (Mozilla, 2025).

Con el lanzamiento y constante evolución de diversas tecnologías web, lenguajes de programación, y servicios, la *World Wide Web* no ha hecho más que expandirse, permitiendo la creación de tecnologías como el *software* como servicio y *Virtual Private Servers*, entre muchas más tecnologías que toman como base la web de manera directa o indirecta.

Esta proliferación ha marcado una tendencia por el desarrollo de aplicaciones web, también denominadas *webapps* o sistemas web, estas son un tipo de aplicación la cual es alojada en un servidor web remoto en Internet o una red local, es accesible a través de cualquier navegador web compatible sin importar el sistema operativo donde se esté ejecutando.

Dado este panorama, las aplicaciones web cada vez se vuelven una mayor constante dentro del ámbito de muchas empresas, incluyendo también instituciones educativas, la Universidad Autónoma del Estado de Hidalgo no es la excepción, contando con diversos tipos de sistemas web, desde su portal web, biblioteca digital, plataforma garza, hasta servicios escolares, además de sistemas monolíticos propios de la administración de cada área académica.

Estos sistemas comúnmente cuentan con el propósito de asistir y complementar distintas actividades que se llevan a cabo dentro del área en la que fue planteado el sistema. Este movimiento de adoptar sistemas complementarios es cada vez más constante dentro de las diversas áreas académicas, particularmente pertenecientes al Instituto de Ciencias Básicas e Ingeniería (ICBI).

Se ha tomado la iniciativa dentro del Área Académica de Ingeniería y Arquitectura (AAIyA) de implementar y automatizar, por medio de un sistema web, una gama de procesos y actividades de administración del Área Académica, haciendo uso de tecnologías, técnicas y metodologías de ingeniería y desarrollo de *software*, al que se le ha denominado Sistema de Inventario *Airone*.

## Antecedentes

---

La Universidad Autónoma del Estado de Hidalgo es la institución de enseñanza más antigua de Hidalgo, contando con 6 institutos, 9 escuelas superiores y 7 preparatorias (Universidad Autónoma del Estado de Hidalgo, 2024). El Área Académica de Ingeniería y Arquitectura es una de las 6 Áreas Académicas que conforman el Instituto de Ciencias Básicas e Ingeniería, a lo largo de sus casi 24 años de creación, ha sido un pilar dentro del Instituto de Ciencias Básicas e Ingeniería.

El Área Académica de Ingeniería y Arquitectura tiene como misión la formación integral de ciudadanos competitivos y capaces para atender la problemática propia y nacional, con ética, solidaridad y responsabilidad, con la generación y aplicación de conocimientos científicos y de tecnología propia (Universidad Autónoma del Estado de Hidalgo, 2024).

Con el fin de lograr su misión y política de calidad, el Área Académica de Ingeniería y Arquitectura cuenta con una gran gama de dispositivos y equipo, capaz de manipular materiales de distintos tipos, todo este equipo es catalogado de acuerdo a su índole, la cual puede ser tanto industrial como computacional en su mayoría, todo este equipo se encuentra repartido entre los distintos edificios y laboratorios que conforman el área.

El procedimiento de la administración del área depende en gran medida de la generación manual de reportes, haciendo uso de plantillas predefinidas y hojas de cálculo, los datos de inventario que se recopilan usualmente están repartidos en varios conjuntos de archivos, cuando se requieren de información nueva y actualizaciones acerca de los distintos elementos de inventario, se recurre a la creación de nuevos formatos y reportes.

Esta práctica puede llegar a ser ineficiente a la hora de hacer actividades como inventario, la razón de esto es la dependencia a la labor del personal prestador en ejercicio de servicio social e información proporcionada por parte de docentes, investigadores y responsables de laboratorios, la naturaleza efímera de los prestadores en ejercicio de servicio social dificulta esta actividad.

Las etiquetas y códigos de inventario proporcionados por la Coordinación de Administración y Finanzas son cruciales para la contabilidad e inventariado de los distintos elementos del Área Académica de Ingeniería y Arquitectura, sin embargo, estas etiquetas poco a poco se están desgastando, haciendo cada vez más ardua la labor de registrar estos elementos conforme pasa el tiempo.

## Planteamiento del problema

---

La dirección administrativa del Área Académica de Ingeniería y Arquitectura realiza diversas actividades para administrar la gran variedad de elementos y actividades que posee el área, las cuales abarcan, pero no se limitan a la administración, contabilidad, mantenimiento, baja de elementos, documentación de los elementos de inventario y adquisición de elementos, entre otras actividades.

La administración y ejecución de estas actividades cuenta con graves deficiencias, la naturaleza descentralizada del proceso de administración es muy ineficiente a la hora de hacer actividades relacionadas a contabilidad e inventariado, la constante generación de documentos y archivos nuevos, el cambio constante de personal en ejercicio de servicio social, desgaste de etiquetas de inventario y las actividades de traspaso no documentadas son las principales brechas dentro del ejercicio de administración de elementos del área.

En consecuencia, se ha ido manifestando gradualmente una serie de problemas derivados estas deficiencias como lo son la gran acumulación archivos y documentos, dificultad para que el personal en ejercicio de servicio social ejerza sus actividades, pérdida/desaparición de elementos, desubicación de elementos derivada del traspaso no documentado, avería constante de equipo industrial y de cómputo e inversión de recursos incorrecta.

Se estima, en el peor de los casos, consecuencias como el robo, pérdida de información crítica, negligencia en el aprendizaje de los alumnos, omisión de la labor por parte del personal en ejercicio de servicio social; retraso y cancelación de eventos en el Área Académica de Ingeniería y Arquitectura, entre otras consecuencias.

Estos fenómenos han afligido de manera directa e indirecta al personal administrativo, docentes, investigadores, personal de mantenimiento y a prestadores en ejercicio de servicio social del área.

## Propuesta de solución

---

La petición del personal administrativo del Área Académica de Ingeniería y Arquitectura consiste en el desarrollo de un sistema donde consultar, registrar y actualizar información acerca de los principales elementos de inventario repartidos en los distintos edificios del área académica, además de un conjunto de características con las que asistir las actividades comunes del personal administrativo del área, derivado la problemática mencionada anteriormente, se propone el desarrollo de una aplicación web de inventario autoadministrable.

De acuerdo a los requisitos solicitados, este sistema contará con una característica dedicada para registrar y editar registros de los distintos elementos pertenecientes al área a través de un formulario dedicado donde, además, es posible atribuir una imagen de referencia.

Con el fin de facilitar la búsqueda de los elementos se implementará una barra de búsqueda dinámica la cual podrá utilizar como parámetros de búsqueda palabras que inserte el usuario en conjunto a atributos que contiene cada registro (Categoría, Etiquetas, Estatus y Ubicación, entre otras).

El sistema contará con un listado predefinido de categorías y fondos que podrán ser asignados a cada elemento, ambos atributos y el apartado de etiquetas serán alterables de acuerdo a la forma de trabajo y clasificación que se lleve a cabo por parte del personal administrativo del Área Académica de Ingeniería y Arquitectura, aumentando de esta manera la flexibilidad del sistema.

Con el fin de asistir de una manera más avanzada a la labor de administración en el Área Académica de Ingeniería y Arquitectura se implementarán características como: (i) Generación de registros de traspasos en el instante en que sea editada la ubicación de un elemento. (ii) Generación de reportes de mantenimiento con la posibilidad de ser exportados en formato PDF. (iii) Programación de anuncios y eventos por medio de un calendario interactivo enlazado a un canal dedicado de *Telegram*. (iv) Exportación e importación de registros de elementos en formato XLSX.

Finalmente, será implementada una característica dedicada para la administración de roles y permisos de usuarios donde se podrán crear nuevos roles con base en una lista de permisos con cada función esencial dentro del sistema, los roles podrán ser asignados directamente a cada usuario, la cantidad de roles que pueden ser asignados van desde 0 o más, con el fin de facilitar la administración y control de las acciones que cada usuario del sistema puede realizar. Todas estas funcionalidades serán intuitivas para cualquier usuario, permitiendo la manipulación del Sistema de Inventario *Airone* sin necesidad de conocimientos avanzados de programación o bases de datos.

## Justificación

---

La creación e implementación de este sistema web otorgará al personal administrativo del Área Académica de Ingeniería y Arquitectura de una serie de herramientas y características efectivas para practicar y complementar las actividades habituales que realiza el personal administrativo.

Entre las actividades que se encontrarán cubiertas o simplificadas por el sistema se encuentran: (i) Contabilizar de elementos. (ii) Inventario de equipo de cómputo, industrial, seguridad, infraestructura, laboratorio e inmobiliario. (iii) Planificación de mantenimientos y eventos. (iv) Recopilación y registro de bajas. (v) Determinación de adquisición de elementos.

Todas estas actividades requieren de un uso extensivo de hojas de cálculo y una recopilación de documentación muy vasta y compleja, limitando de manera importante quiénes y cómo pueden aportar al registro y recopilación de información acerca de los elementos del sistema, todo esto es en el supuesto caso de que se cuente con la información necesaria, es un hecho que existe una pérdida y sesgo en la información recabada a través de los años, derivado de la cantidad de documentos y archivos que se manejan.

Los puntos más críticos de todo el procedimiento de administración básicamente están limitados a dos personas, generando un cuello de botella grave en el flujo de actividades administrativas las cuales no pueden ser solventadas de manera apropiada a pesar de contar con el personal en ejercicio de servicio social.

El romper la brecha de complejidad con las tareas dentro de la administración aportará grandes beneficios en materia de organización, tiempo de respuesta, resolución de problemas y organización de eventos.

Estos beneficios podrán apreciarse en la reducción del tiempo necesario para las actividades relacionadas con los elementos del área, un tiempo menor de capacitación, asignación y coordinación para el personal en ejercicio de servicio social, elaboración de estrategias de administración tomando en base al sistema, efectividad y fiabilidad superior del equipo e inmobiliario de aulas y laboratorios del área, reducción y distribución de la carga de actividades derivadas del apoyo en laboratorios y distintas ubicaciones del área para el personal, y muchos más beneficios potenciales derivados de la implementación del sistema.

El sistema, junto a sus herramientas y características, parten de la premisa de optimizar, centralizar y volver más eficiente el proceso de administración de elementos y actividades dentro del Área Académica de Ingeniería y Arquitectura, establecer las bases para una administración mucho más dinámica y efectiva es la prioridad de este sistema.

## Objetivo General

---

Desarrollar un sistema web que facilite la gestión y consulta de información del inventario del Área Académica de Ingeniería y Arquitectura mediante la implementación de tecnología web para su uso por parte del personal administrativo y prestadores en ejercicio de servicio social de esta área.

## Objetivos Específicos

---

- Identificar las herramientas que serán utilizadas para el desarrollo del Sistema de Inventario *Airone*.
- Adoptar cursos y capacitaciones para manejar de manera eficiente y apropiada las herramientas necesarias para el desarrollo del Sistema de Inventario *Airone*.
- Realizar todas las pruebas pertinentes al Sistema de Inventario *Airone* para corroborar todas sus funcionalidades.
- Validar el Sistema de Inventario *Airone* con el personal que hará uso del sistema y usuarios.
- Implementar el Sistema de Inventario *Airone* dentro de una red de área local (LAN) en el edificio del Área Académica de Ingeniería y Arquitectura.

# Metodologías

---

Durante el análisis del presente desarrollo se han considerado tres posibles metodologías de desarrollo de *software* para el desarrollo del Sistema de Inventario *Airone*, cada una aportando un conjunto de beneficios e impedimentos, estas son:

## **Scenario-based Object-oriented Hypermedia Design Methodology**

La metodología *SOHDM* (*Scenario-based Object-oriented Hypermedia Design Methodology*) es una metodología de desarrollo web, la cual se basa fuertemente en tecnologías orientadas a objetos, esta metodología propone y hace uso de escenarios como principal recurso para la abstracción y definición de los requisitos del usuario (Lee et al., 1998), dentro de esta definición se hace un uso extensivo de diagramas de flujos de datos de acuerdo a la notación de Yourdon DeMarco.

El proceso de la metodología de *SOHDM* se basa en continuar fusionando cada diagrama de clases para conseguir un modelo de clases navegacional del sistema, simplificando la implementación y construcción del sistema. La metodología *SOHDM* se basa en seis fases siendo (i) Análisis del dominio. (ii) Modelado de objetos. (iii) Diseño de Vistas. (iv) Diseño Navegacional. (v) Diseño de la Implementación. (vi) Construcción.

## **Metodología en Espiral**

La metodología en espiral es un enfoque para el desarrollo de *software* creado por Barry Boehm, consistiendo de un modelo iterativo enfocado a la evolución del proceso del *software* mediante el diseño y desarrollo de prototipos, adoptando los aspectos sistémicos de la metodología en cascada (Doshi et al., 2021).

La metodología se basa en 4 fases las cuales serán aplicadas de manera iterativa, consistiendo de (i) Objetivo y determinación alternativa. (ii) Análisis y evaluación de riesgos. (iii) Desarrollo y pruebas y (iv) Planificación del siguiente ciclo.

El uso de esta metodología incita a los desarrolladores a elaborar actividades de análisis acerca del rendimiento y progreso realizado en cada ciclo realizado basándose en esta metodología, enfatizando, además, la identificación de las fortalezas y debilidades dentro de cada proceso y producto generado.

## Programación Extrema (XP)

La programación extrema (XP) es una metodología de desarrollo de *software* desarrollada por Kent Beck, Ward Cunningham y Ron Jeffries a finales de los noventa (Wells, 2009). Los valores de la programación extrema (comunicación, simplicidad, retroalimentación, valentía y respeto) son el motor para las actividades y tareas específicas de XP.

La programación extrema se basa en las relaciones interpersonales, promoviendo el trabajo en equipo, aprendizaje de los desarrolladores y el clima laboral, la programación extrema depende en gran medida de la retroalimentación continua entre el cliente y el equipo de desarrollo.

Esta metodología de desarrollo de *software* es especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, respaldándose de cinco fases, estas fases son (i) Planeación. (ii) Diseño. (iii) Codificación. (iv) Pruebas. (v) Lanzamiento.

La Tabla 1 muestra una comparativa entre las principales ventajas y desventajas de las metodologías de desarrollo de *software* SOHDM, Espiral y Programación Extrema:

	Ventajas	Desventajas
<b>SOHDM</b>	<ul style="list-style-type: none"> <li>• Adopta un paradigma Orientado a objetos y eventos para su diseño.</li> <li>• Hace uso de un flujo de proceso Iterativo y Evolutivo.</li> <li>• Es una metodología enfocada en el desarrollo de sistemas web.</li> </ul>	<ul style="list-style-type: none"> <li>• No cubre la generación de pruebas unitarias para el proyecto.</li> <li>• No cubre la generación de documentación/ procesos de mantenimiento del proyecto.</li> <li>• El proceso iterativo de la metodología es prolongado y redundante, requiriendo de muchos diagramas de diseño.</li> </ul>
<b>Espiral</b>	<ul style="list-style-type: none"> <li>• Gran énfasis en la gestión de riesgos.</li> <li>• Flexibilidad superior en comparación con modelos de desarrollo de software tradicionales.</li> <li>• El cliente puede interceder dentro de cada fase del proceso de la metodología.</li> </ul>	<ul style="list-style-type: none"> <li>• Los periodos de tiempo para cada fase del proceso en espiral son rígidos y con poco margen para ser modificados.</li> <li>• La metodología requiere de un control de riesgos elaborado.</li> <li>• Ineficiente para el desarrollo de proyectos con lapsos de tiempo cortos.</li> </ul>
<b>Programación Extrema</b>	<ul style="list-style-type: none"> <li>• La metodología tiene un alto énfasis en la interacción con el cliente.</li> <li>• Ciclo de Iteraciones corto altamente flexible a los cambios.</li> <li>• Generación de prototipos y entregables en periodos de tiempo breves.</li> <li>• Cubre todas las fases fundamentales para el desarrollo de software.</li> </ul>	<ul style="list-style-type: none"> <li>• Efectiva únicamente en proyectos con lapsos de tiempo cortos.</li> <li>• La corrección de errores es una labor inmediata para el uso efectivo de esta metodología.</li> <li>• No es tan eficiente en proyectos con lapsos de tiempo y requisitos muy rígidos.</li> </ul>

Tabla 1. Ventajas y desventajas entre las metodologías SOHDM, Espiral y XP.

## **Metodología de desarrollo de software elegida**

Dado el contexto, requisitos y opciones con las que se dispone para el desarrollo del Sistema de Inventario *Airone*, la metodología de desarrollo ágil programación extrema es idónea, el constante cambio de requisitos, lapsos de entrega cortos y la comunicación frecuente con el personal interesado son puntos clave para esta decisión. La metodología de desarrollo ágil programación extrema será utilizada para el desarrollo del Sistema de Inventario *Airone*.

El personal administrativo del Área Académica de Ingeniería y Arquitectura cuenta con una alta disposición para describir, observar y desempeñar revisiones constantes al sistema de inventario, las iteraciones de dos semanas son idóneas y altamente compatibles con el flujo de trabajo del personal del área.

El uso de código como documentación es un punto clave para la adopción de esta metodología, debido a que el personal administrativo cuenta con nociones de programación superiores al promedio, el conocimiento vasto del funcionamiento del sistema, en conjunto a la documentación detallada que se ofrecerá junto al sistema finalizado ofrecen una visión general del sistema más completa, otorgando al sistema una mayor accesibilidad, además de una documentación y recopilación de errores mucho más inmediata, detallada y estructurada.

## Herramientas de desarrollo

---

El desarrollo del Sistema de Inventario *Airone* implica el uso de una buena diversidad de herramientas de *software* enfocadas al desarrollo web, cada una con funcionalidades y propósitos específicos. *HTML*, en conjunto de *Apache HTTP server* serán utilizados dentro del Sistema de Inventario *Airone* para integrar las funcionalidades de servidor web para la recuperación de documentos de hipertexto y cualquier recurso utilizado por estos.

El diseño e interfaces del sistema serán generados haciendo uso del lenguaje *CSS*, en conjunto a *Frameworks* como *Bootstrap*, *TailwindCSS*, facilitando la labor de generar estilos a diversos componentes de *HTML*, otorgando de mayor estética a las distintas interfaces del sistema. *Bootstrap* será utilizado en el diseño del panel administrativo y páginas de aterrizaje.

El panel administrativo *AdminLTE 3*, el cuál usa en gran medida el *Framework* de *Frontend*, *Bootstrap*, será implementado en el Sistema de Inventario *Airone*, con el fin de otorgar de las distintas vistas del sistema una interfaz intuitiva, sencilla y dinámica para los usuarios finales, sin saturar de elementos o volver monótona la navegación. El uso de *TailwindCSS* en el sistema web consistirá en complementar toda área en la que *Bootstrap* no es suficiente en términos de diseño, por ejemplo, la implementación de ciertos colores y estilos de elementos y componentes.

La generación de interfaces dinámicas, manipulación del *DOM*, además del tratamiento de imágenes y variables del lado del cliente será proporcionada gracias a herramientas como *JavaScript*, *jQuery* tratará con las transacciones *AJAX*, mientras que *Livewire* será utilizada en gran medida para la implementación de barras de búsqueda responsivas.

El lado del servidor trabajará con *PHP* en conjunto al *Framework* de *Backend*, *Laravel*, ambas herramientas tratarán con la mayoría de peticiones del cliente del lado del servidor, *Laravel* será utilizado para el desarrollo de toda la estructura del sistema, siguiendo con un patrón de diseño *MVC (Model – View – Controller)*, además de las distintas herramientas que ofrece este *Framework*.

El SGBD utilizado para para este desarrollo será *MySQL*, este proporcionará servicios de bases de datos relacionales al Sistema de Inventario *Airone*, gestionando todas las consultas realizadas por el usuario y proporcionándolas al *Backend* y *Frontend* del Sistema de Inventario *Airone*.

Funcionalidades como notificaciones *Pop-Up*, Generación de documentos en formato PDF y XLSX, Envío de notificaciones por Telegram, Gestión de permisos y calendario interactivo serán implementados mediante el uso las librerías *Toastr*, *DOMPDF*, *Laravel Excel*, *Laravel Telegram*, *Laravel Permission*, y *FullCalendar*. El control de versiones y paquetes para el desarrollo del Sistema de Inventario *Airone* fue proporcionado por las herramientas *Composer*, *Vite* y *Git*, interviniendo en el control de las iteraciones propuestas para el sistema, otorgando de un desarrollo y despliegue mucho más consistentes.

## Alcance

---

- El sistema proporcionará una contabilidad fiable de los elementos del inventario del Área Académica de Ingeniería y Arquitectura.
- La clasificación de elementos dentro del sistema ofrecerá la pauta de tener un filtrado consistente de los elementos.
- Se podrá abordar de manera adecuada y detallada cada elemento de inventario a nivel de especificaciones, atributos y localización.
- Las herramientas integradas del sistema ofrecen la posibilidad de alterar y modificar el sistema de acuerdo a parámetros como categorías, fondos y etiquetas, con el fin de apropiar aún más el sistema a la forma de trabajo del personal administrativo del área.
- El sistema contará con las herramientas necesarias para generar reportes individuales de mantenimiento en formato PDF.
- El sistema dispondrá de una herramienta dedicada para exportar e importar datos de registros de elementos en formato XLSX.
- El sistema permitirá generar respaldos de la base de datos *SQL*, con el fin de proporcionar un recurso de asistencia en caso de fallas del servidor.
- El calendario embebido dentro del *Dashboard* del sistema es interactivo, y puede modificarse de acuerdo a los eventos que sean agregados, dichos eventos serán notificados por medio de un canal dedicado de *Telegram* bajo una hora concreta.
- La administración de permisos y roles dentro del sistema será altamente flexible de acuerdo a las necesidades del personal administrativo del Área Académica de Ingeniería y Arquitectura.
- Cada usuario contará con la posibilidad de recuperar su contraseña por medio de correo electrónico además de habilitar la autenticación de dos pasos para el acceso al sistema.

# Capítulo 1 . Marco Teórico

---

En este capítulo se abordará un conjunto de conceptos básicos acerca del entorno donde se desenvuelve el Sistema de Inventario *Airone*, como lo es Internet, desarrollo web y sus herramientas, con el propósito de establecer y contextualizar al lector acerca de los diversos fundamentos detrás del desarrollo de sistemas web.

Este capítulo cuenta con un marco metodológico, donde se evaluarán un conjunto de metodologías de desarrollo de *software*, mostrando sus premisas, fases y consideraciones, concentrándolas por medio de un cuadro comparativo.

Una vez abordados los principales conceptos técnicos de desarrollo web, se indagarán conceptos más cercanos al ámbito en el que el Sistema de Inventario *Airone* se encuentra embebido, como lo es el inventario y la administración educativa.

## 1.1 Familia de protocolos de Internet (TCP/IP) y La World Wide Web

---

La familia de protocolos de Internet es un conjunto de protocolos de comunicación que componen la arquitectura de Internet, fueron implementados por primera vez en el año 1972 por parte del Departamento de Defensa de los Estados Unidos, ejecutándose en *ARPANET*, una red de dicho departamento.

Esta familia de protocolos se conforma de más de 100 normas o protocolos que no dependen de ningún fabricante, estos protocolos han llegado al grado de ser estándares, estos estándares actúan en conjunto para habilitar e interconectar computadoras, independientemente de su sistema operativo, hardware o presentación, la familia de protocolos de Internet ha sido vital para establecer lo que conocemos actualmente como Internet (Oracle, s.f).

### 1.1.1 Modelo de referencia OSI y TCP/IP

El modelo de referencia *OSI* y *TCP/IP* son un modelo conceptual y de referencia creado por la Organización Internacional para la Estandarización y Vinton Cerf junto a Robert E. Kahn respectivamente.

Ambos modelos de referencia describen y definen una estructura para las distintas actividades de red e interconexión, el modelo de referencia *OSI* consta de siete capas mientras que el modelo de referencia *TCP/IP* consta de 4 capas, cada capa de ambos modelos está asociada a uno o más protocolos de la familia de protocolos *TCP/IP*.

Las capas representan las operaciones de transferencia de datos comunes a todos los tipos de transferencias de datos, desde medios físicos hasta una aplicación receptora (Universidad Autónoma del Estado de Hidalgo, s.f). La Figura 1.1 muestra las 7 capas del modelo de referencia OSI en conjunto a las 4 capas del modelo de referencia TCP/IP, resaltando sus diferencias y equivalencias entre sí.

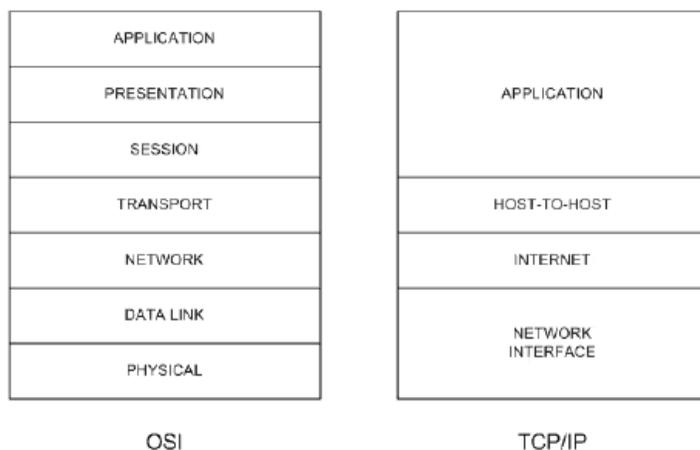


Figura 1. 1. Conjunto de protocolos TCP/IP en relación con el modelo de OSI de referencia. Información general de TCP/IP. Por CISCO. (2022). CISCO ([http://cisco.com/c/es\\_mx/support/docs/ip/routing-information-protocol-rip/13769-5.html](http://cisco.com/c/es_mx/support/docs/ip/routing-information-protocol-rip/13769-5.html)).

### 1.1.2 World Wide Web

La *World Wide Web* es un sistema de hipermedios interconectados entre sí accesibles mediante una conexión a Internet en conjunto a un *software* especializado, un navegador web, por ejemplo. La *World Wide Web* no es lo mismo que el Internet, esta es solo una de las muchas aplicaciones construidas sobre Internet (Mozilla, 2024).

El núcleo de la *World Wide Web* son tres protocolos que trabajan en conjunto, el primero de ellos es un sistema universal para referenciar recursos en la web, como páginas, este es denominado *URI*, este sistema establece las pautas para el siguiente protocolo, *HTTP*, el protocolo *HTTP* se encarga de la comunicación entre el navegador y el servidor web, finalmente, *HTML* define la estructura y contenido de una página web por medio de documentos de hipertexto (Mozilla, 2024).

Esta base de protocolos ha establecido las pautas para el desarrollo de Internet tal y como lo conocemos, la proliferación e integración de otras tecnologías a esta pila de protocolos y su uso masivo es prueba del impacto que ha tenido la *World Wide Web*.

### 1.1.3 Protocolo HTTP y HTTPS

#### **HTTP**

*HTTP* es un protocolo diseñado para transferir información entre dispositivos conectados de la red, este protocolo está orientado a transacciones, siguiendo un principio de conexión cliente-servidor, *HTTP* se ejecuta sobre otros protocolos, por ejemplo, *TCP*, *IP*, *DNS*, entre otros. El protocolo *HTTP* es capaz de transmitir documentos de hipertexto *HTML*, imágenes, vídeos, contenido, entre otros datos. *HTTP* puede incluso ser utilizado para transmitir partes de documentos, y actualizar páginas web de manera dinámica.

*HTTP* se basa en el uso de peticiones y respuestas, una petición de *HTTP* se conforma de (i) Un método *HTTP*, este método define la operación que el cliente quiere realizar. (ii) La dirección del recurso pedido (del acrónimo en inglés, *URL*). (iii) La versión del protocolo *HTTP*. (iv) Cabeceras *HTTP* opcionales, estas cabeceras pueden suplir de información adicional a los servidores.

Una respuesta *HTTP* se conforma de cuatro elementos. (i) La versión del protocolo *HTTP* que se está usando. (ii) Un código de estado. (ii) Un mensaje de estado. (iii) Cabeceras *HTTP*, como las de las peticiones. (iv) El recurso que se ha pedido, este recurso puede ser código *HTML*, *CSS*, *JavaScript*, entre otros recursos. La Figura 1.2 resume el proceso de comunicación entre el usuario, navegador y servidor web mediante el protocolo *HTTP*.



Figura 1. 2. Proceso de Comunicación por medio de HTTP. Adaptado de El proceso de comunicación HTTP. Por TechWebPe. (2021). TechWebPe. (<https://techwebpe.com/dominios/http-que-es-como-funciona/>).

#### **HTTPS**

El protocolo *HTTPS* es un protocolo basado en *HTTP*, con el objetivo de realizar comunicación y transferencias de datos de manera segura. *HTTPS* es particularmente útil cuando se trabaja con servicios que manipulan información delicada por medio de la web, por ejemplo, bancos en línea, servicios de alojamiento en la nube, correo electrónico y sistemas médicos, entre otros usos (Cloudflare, 2024).

*HTTPS* asegura las comunicaciones mediante encriptación asimétrica, por medio del protocolo *TLS/SSL* (*Transport Layer Security* y *Secure Sockets Layer* respectivamente). La encriptación asimétrica utiliza dos claves diferentes para encriptar las comunicaciones entre dos partes, los datos encriptados con la clave pública sólo se pueden desencriptar con la clave privada. La figura 1.3 muestra cómo funciona este método de encriptación.

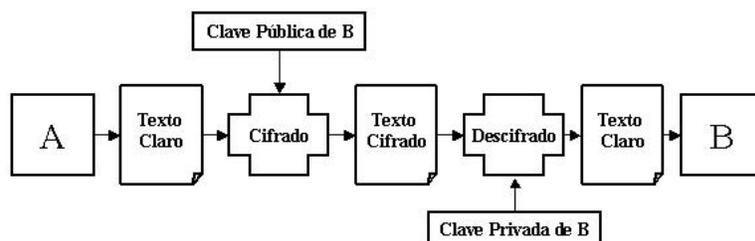


Figura 1. 3. Proceso de Encriptación Asimétrica. Adaptado de Algoritmo Asimétrico. Por Durán, D. (2013). Ecured.cu. (<https://www.ecured.cu/Archivo:Algoritmo Asimetrico.jpeg>)

## 1.2 Conceptos de desarrollo Web

---

El desarrollo de aplicaciones web es un ámbito muy enriquecido de conceptos, los cuales proponen en su mayoría definiciones clave, modelos de desarrollo, arquitectura, herramientas de programación y *software* relacionado, a continuación, se presentarán un conjunto de conceptos.

### 1.2.1 Aplicación web

Una aplicación web, también denominada, sistema web o *webapp*, es una categoría de *software* centrado en el uso extensivo de un navegador web, agrupa una amplia gama de aplicaciones.

La constante evolución de las tecnologías web ha permitido la creación de aplicaciones web más sofisticadas, más allá de solo características aisladas y contenido estático, esta evolución se plasma en integraciones con bases de datos y aplicaciones de negocios, contando con la generación e implementación de contenido dinámico (Pressman, S., 2010).

La principal premisa de las aplicaciones web es su acceso, el cual no requiere de una instalación como un *software* de escritorio común, únicamente un navegador web y una conexión hacia el sistema web.

### 1.2.2 Patrón de diseño MVC

El patrón de diseño *MVC* es un patrón de diseño de *software* introducido por Trygve Reenskaug en el año 1979 (Reenskaug, T., s.f.) utilizado para la implementación y estructuración de interfaces de usuario, datos y lógica de control con el propósito de modularizar un sistema de acuerdo a tres componentes, los componentes de *MVC* son:

- **Modelo:** Este componente se encarga de todas la manipulación y definición de los datos, generalmente esto implica el uso de consultas a una base de datos.
- **Controlador:** Administra la lógica necesaria para actualizar el modelo y/o vista de acuerdo a las entradas/acciones del usuario de la aplicación, por lo tanto, actúa como un intermediario entre el modelo y la vista, esta lógica puede ser desde una solicitud a los datos del modelo, manejo de datos hacia las vistas, entre otras actividades.
- **Vista:** Define cómo se deben mostrar los datos de la aplicación, además de solicitar actualizaciones al modelo por medio del controlador, es el medio por el cual el usuario puede manipular el controlador.

Este patrón de diseño es el soporte y base de muchos sistemas basados en la web, tal es su influencia que gran parte del desarrollo web, en conjunto a diversos *Frameworks*, se basan alrededor del concepto de *MVC* (*Laravel*, *Django* y *Ruby on Rails*, entre otros *Frameworks*). La Figura 1.4 muestra cómo es que interactúan los componentes de un sistema de acuerdo a las pautas de *MVC*.

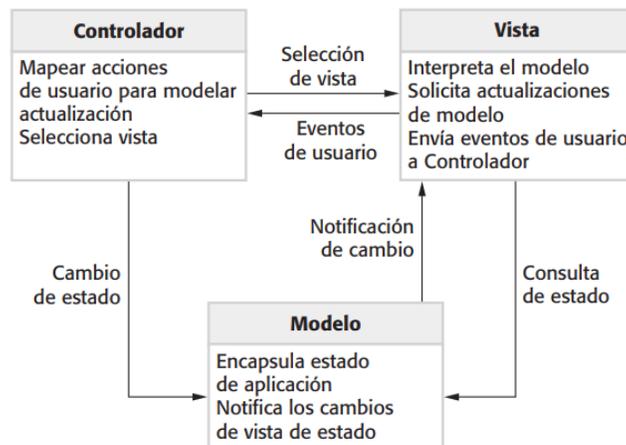


Figura 1. 4. Organización de la Arquitectura MVC (p.156). Adaptado de *La organización del MVC. Por Sommerville. I* (2011). *Ingeniería del Software, Un enfoque Practico, Novena Edición. Pearson Educación.*

### 1.2.3 Framework

Un *Framework* es un marco de trabajo que ofrece y plantea un conjunto de prácticas estandarizadas y componentes reutilizables que aceleran el proceso de desarrollo y organización de aplicaciones, tomando como referencia un patrón de desarrollo de *software*, actúa como una plantilla para la organización y desarrollo de *software*.

Los *Frameworks* ofrecen un conjunto de recursos desarrollados por otros programadores, estos recursos incluyen documentación, código, pautas y convenciones que pueden utilizarse durante el proceso de desarrollo de la aplicación, evitando de esta manera la reinención de componentes comunes.

El uso de *Frameworks* conlleva un conjunto de ventajas importantes como lo es la mejora una mayor eficiencia, productividad y tiempos de desarrollo más cortos, pero también implica consideraciones muy importantes, la principal de ellas es la curva de aprendizaje requerida para utilizar un *Framework*, la sobrecarga de recursos e inflexibilidad de la arquitectura son puntos muy importantes a considerar a la hora de adoptar un *Framework*. Dentro del contexto de la programación de aplicaciones web, los *Frameworks* destacan por su gran variedad y enfoques que pueden adoptar, ya sea dentro de contextos o ámbitos como el *Frontend*, *Backend* o incluso una combinación de ambos, denominada *Full-Stack* (Assembler Institute, 2022).

### 1.2.4 Gestor de dependencias

También denominado Sistema de Gestión de Paquetes, es una herramienta que automatiza la instalación, actualización, y gestión de bibliotecas y paquetes de *software* que un proyecto o sistema operativo necesita para funcionar.

Estos paquetes, también conocidos como dependencias, pueden ser bibliotecas de código, *Frameworks*, módulos o cualquier otro componente que sea requerido por un proyecto para ser ejecutado correctamente, los paquetes pueden ser creados por terceros verificados (openSUSE, 2011).

Los paquetes que puede manejar e instalar el Gestor de Dependencias usualmente son definidos mediante un repositorio dedicado. Este repositorio es el medio por el cuál las dependencias de un proyecto pueden ser actualizadas, verificadas e incluso documentadas de acuerdo a su desarrollo.

## 1.2.5 Bases de datos SQL y NoSQL

### *Base de datos SQL*

Una base de datos *SQL*, es un tipo de base de datos relacional, la cuál es descrita por medio de un lenguaje de consulta, su principal premisa es el tratamiento de los datos, estos son almacenados de manera estructurada por medio de tablas y están relacionados entre sí.

Las bases de datos *SQL* permiten la relación de las tablas y los datos que almacenan por medio del uso de claves de registros únicas para cada tabla, el uso de claves es una actividad común para la administración de los registros de las tablas.

La definición y manipulación de las bases de datos *SQL* es por medio del lenguaje *SQL* (*Structured Query Language*), este lenguaje de consulta se divide en 5 subconjuntos de lenguajes, cada uno con un propósito diferente dentro de la administración y uso de una base de datos *SQL*. Esos subconjuntos son (Amazon, 2023):

- Lenguaje de definición de datos (*Data Definition Language*). Contiene un conjunto de comandos *SQL* que definen la estructura de la base de datos, esta estructura puede consistir de bases de datos, tablas, columnas de atributos y sus relaciones.
- Lenguaje de consulta de datos (*Data Query Language*). Son un conjunto de comandos enfocados en la recuperación de datos almacenados en la base de datos, este lenguaje también se encarga de las operaciones de unión (*Join*) y vistas (*Views*).
- Lenguaje de manipulación de datos (*Data Manipulation Language*). Se encarga de añadir información nueva, además de modificar registros existentes en una base de datos relacional, esta agregación puede ser hacia una o más tablas pertenecientes a una base de datos en común.
- Lenguaje de control de datos (*Data Control Language*). Administra o autoriza el acceso a la base de datos por medio de permisos y roles, los cuales pueden ser atribuidos de manera individual o en grupos de usuarios.
- Lenguaje de control de transacciones (*Transaction Control Language, TCL*). El motor relacional utiliza el lenguaje de control de transacciones para hacer cambios en la base de datos de manera automática, este lenguaje es vital para la creación de transacciones con *rollback* y *commit*, añadiendo una capa de atomicidad a las consultas *SQL*.

Las bases de datos *SQL* cuentan con restricciones de integridad que aportan a la consistencia, atomicidad de los datos y consultas en varios niveles, estas restricciones implican una integridad referencial, de dominio, formato, rango y de llave primaria (Franco, K., 2020, Diapositivas 4 y 5).

### ***Base de datos NoSQL***

Una base de datos *NoSQL* o *Not Only SQL* es un tipo de base de datos no relacional que no cuenta con un identificador que relacione un conjunto de datos con otro, por lo tanto, no requiere que los datos presenten una estructura definida para poder manipularlos.

Las bases de datos *NoSQL* albergan datos dentro de una estructura de datos, este acercamiento al almacenamiento de datos ofrece una rápida escalabilidad para gestionar conjuntos de datos grandes y normalmente no estructurados (IBM, 2024). Las bases de datos *NoSQL* pueden adoptar muchas formas, estas pueden clasificarse en cuatro tipos de bases de datos *NoSQL* (Oracle, s.f), las cuales son:

- **Valor-clave.** Es el tipo más flexible de base de datos *NoSQL*, la aplicación tiene un control completo sobre lo que se almacena en el campo de valor sin restricción alguna. La clave de la que hace uso la aplicación puede ser sintética o autogenerada, así como tener diferentes formatos, esta clave es única.
- **Orientadas a documentos.** También denominadas bases de datos orientadas a documentos o de almacenamiento de documentos, se utilizan para almacenar, recuperar y gestionar datos semiestructurados.

No es necesario especificar los campos que debe contener cada documento, la especificación del formato de los datos depende en gran medida de la naturaleza del tipo de documento, *JSON (JavaScript Object Notation)* es comúnmente utilizado dentro de este tipo de base de datos *NoSQL*.

- **Grafos.** Organiza los datos como nodos y relaciones, que muestran las conexiones entre nodos, esta conexión por nodos da la posibilidad de establecer relaciones al estilo de una base de datos relacional.
- **Columna ancha.** Almacena y gestiona datos en forma de tablas, filas y columnas. Se despliegan comúnmente en situaciones donde una aplicación requiere de un formato de columna para la captura de datos sin un tipo de esquema estandarizado. La adopción de estructuras de tablas da apertura a procesos de lectura de datos eficientes en conjunto a un mayor rendimiento.

## **ACID**

Es un acrónimo en inglés que refiere a cuatro palabras que hacen referencia a un conjunto de características que debe poseer una base de datos con el fin de mantener la integridad de los datos, estas características son:

- **Atomicidad (*Atomicity*)**. Una transacción debe de ocurrir en su totalidad, o no ocurrir en lo absoluto, esta característica debe de garantizar que todas las sentencias *SQL* de una sola transacción hacia la base de datos se completen o, en su defecto, se reviertan a su estado original.
- **Consistencia/Coherencia (*Consistency*)**. Todas las transacciones dentro de la base de datos deben respetar todas las reglas de integridad definidas en el contexto de la base de datos, si alguna transacción infringe una o más reglas, se revertirá la transacción. También hace referencia a mantener una coherencia conjunta para todos los usuarios, a pesar de que estos realicen diversas transacciones en simultaneo.
- **Aislamiento (*Isolation*)**. Se debe asegurar que cualquier transacción en plena ejecución sea invisible para todas las transacciones en proceso hasta el instante en que sea completada, esta característica asegura que cualquier nueva transacción espere hasta que finalice una transacción anterior antes de comenzar a funcionar.
- **Durabilidad (*Durability*)**. Garantiza que los cambios realizados por cada transacción completada con éxito sean persistentes, a pesar de que ocurra una falla en el sistema donde es almacenada la base de datos.

### **1.2.6 LAMP**

*LAMP* es el acrónimo de una pila de cuatro *software* utilizados en conjunto, con el propósito de generar aplicaciones web, el acrónimo hace referencias al uso de un sistema operativo *Linux*, un servidor web *Apache*, un servidor de base de datos *MySQL* y el lenguaje de programación *PHP*.

El acrónimo *LAMP* fue presentado por Michael Kunze en un artículo publicado en la revista *Computertechnik* en 1998, con la premisa de presentar una alternativa rentable a los paquetes comerciales de distintas empresas para el despliegue y uso de tecnologías web (Michael, K., 1998). Las tecnologías utilizadas en la pila *LAMP* son:

#### ***Linux***

Es una familia de sistemas operativos basados en *UNIX*, propuesto por Linus Torvalds en el año 1991, *Linux* adopta un modelo de desarrollo de *software* libre, por lo tanto, el núcleo Linux cuenta con una gran variedad de distribuciones, cada una con propósitos distintos, desde el uso de escritorio, ciberseguridad, servidores, multimedia, entre otros.

Cada distribución de *Linux* cuenta con un conjunto de herramientas *GNU*, las cuales permiten administrar los recursos que brinda el *kernel* (Lenovo, s.f), además de instalar módulos de *software*, entre otras propiedades más.

### ***Apache***

Se refiere a un servidor *HTTP* de código abierto creado y mantenido por la *Apache Software Foundation* desde el año de 1995, este *software* tiene el propósito de procesar solicitudes *HTTP* y páginas web, facilitando la actividad de establecer una conexión entre un servidor y los navegadores de los visitantes del sitio web mediante una arquitectura Cliente - Servidor, la naturaleza de los servidores web permiten que esta conexión sea multiplataforma.

*Apache* cuenta con una estructura basada en módulos, cada módulo permite implementar distintas funcionalidades adicionales, estos módulos pueden implementar herramientas de seguridad, implementación de caché, entre otras más.

### ***MySQL***

*MySQL* es un sistema gestor de bases de datos relacionales desarrollado por David Axmark, Allan Larsson y Michael Widenius, *MySQL* fue adquirido por *Sun Microsystems* en 2008 y subsecuentemente por *Oracle* en el 2010.

*MySQL* utiliza un modelo de base de datos relacional, este modelo enfatiza el almacenamiento de los datos en tablas con un conjunto de relaciones, el uso de este modelo implica el uso de un lenguaje de consulta *SQL* con el fin de poder realizar transacciones dentro de la base de datos.

### ***PHP***

*PHP* es un lenguaje de programación enfocado para el desarrollo web, su código es ejecutado del lado del servidor, generando *HTML* y enviándolo al cliente, *PHP* es compatible con una gran gama de sistemas operativos y servidores web, siendo compatible con paradigmas de programación como la programación por procedimientos u orientada a objetos.

El servidor web, de acuerdo a su configuración, puede procesar todos los ficheros *HTML* mediante *PHP*, este comportamiento es posible gracias a que *PHP* trabaja con un intérprete, otorgando de este lenguaje de una gran flexibilidad y dinamismo. (PHP Documentation Group, 2024).

*PHP* fue creado inicialmente por Rasmus Lerdorf en el año 1994, la implementación de referencia de *PHP* es producida por *The PHP Group* (PHP Documentation Group, 2024).

### 1.2.7 API

Una *API*, también denominada, *Application Programming Interface* o interfaz de programación de aplicaciones, es el medio por el cuál a través de un conjunto de protocolos y reglas establecidas, se establece la comunicación entre distintas aplicaciones de *software*, comúnmente para el intercambio de datos de manera consistente y segura (IBM, 2024).

La principal premisa de las *API* es su capacidad de ocultar los detalles internos de cada sistema implicado en el proceso de comunicación de la *API*, realizando exclusivamente el intercambio de información relevante para las solicitudes realizadas dentro del contexto de la *API*, actuando como un *ad hoc* de tipo Cliente-Servidor. La Figura 1.5 muestra el proceso de comunicación entre un cliente, la *API* y el proveedor de la *API*.

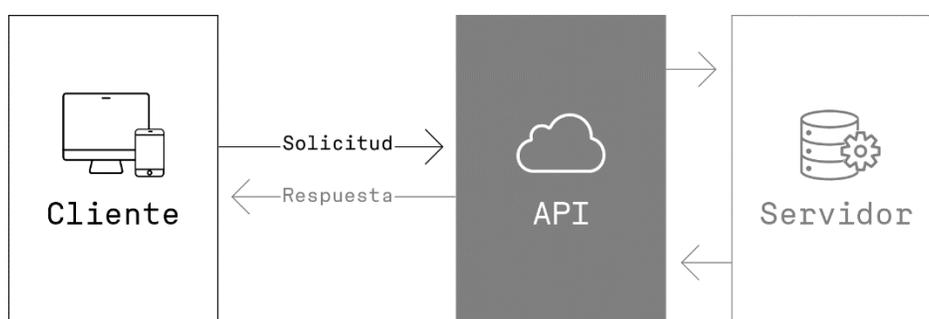


Figura 1. 5. Proceso de comunicación de una API. Adaptado de API, Comunicación C-S. Por Arizbé, K. (2023). Gluo. (<https://www.gluo.mx/blog/que-es-una-api-y-para-que-sirve>)

Las *APIs* se clasifican de acuerdo a los casos de uso a los que pertenecen, su clasificación es (IBM, 2024):

- *API* de bases de datos. Establecen la conexión entre las aplicaciones y uno o más sistemas de gestión de bases de datos, con propósito de realizar consultas de índole *SQL* o *NoSQL*.
- *API* del Sistema Operativo. Definen las pautas de utilización de servicios y recursos del sistema operativo para una o más aplicaciones en ejecución.
- *API* Remotas. Definen las interacciones posibles para distintas aplicaciones de acuerdo al dispositivo en donde se esté utilizando.
- *API* Web. Realizan la transferencia de datos e información relacionada a funcionalidades o servicios particulares, haciendo uso del protocolo *HTTP* y/o la infraestructura de Internet para el proceso de comunicación.

## 1.3 Herramientas de desarrollo Web

---

El desarrollo del Sistema de Inventario *Airone* implica el uso de las siguientes herramientas de *software*:

### HTML

El *HyperText Markup Language*, o en español, lenguaje de marcado de hipertexto, es un formato de datos utilizado para la creación de documentos de hipertexto portables multiplataforma (Connoly, 1999), este lenguaje ha evolucionado a lo largo de los años desde su liberación y estandarización en el año 1995, su última versión estable, *HTML5*, convierte a *HTML* de un simple formato de marcado para estructurar documentos, en una plataforma completa de desarrollo de aplicaciones (Mozilla, 2024), *HTML5* es un estándar vivo, es decir, que sigue siendo desarrollado, manteniendo y expandiendo por la *W3C*, también denominada *World Wide Web Consortium*.

La especificación de *HTML* fue creada por Tim Berners-Lee y Robert Cailliau en conjunto a un grupo de trabajo en el año 1989, la especificación de *HTML* en conjunto a otras especificaciones conforma la *World Wide Web* (World Wide Web Consortium, 2020).

Es importante destacar que *HTML* no es un lenguaje de programación sino un lenguaje de marcado, este marcado consiste de etiquetas *HTML*, las etiquetas definen la estructura del contenido de la página. *HTML* ofrece una serie de elementos que permiten que diversas partes del contenido luzcan o se comporten de una determinada manera. La Figura 1.6 Muestra el logotipo de *HTML*.



Figura 1. 6. Logotipo de HTML. Adaptado de HTML5 logo and wordmark. Por Wikipedia contributors. (s.f.). Wikipedia, The Free Encyclopedia. ([https://es.m.wikipedia.org/wiki/Archivo:HTML5\\_logo\\_and\\_wordmark.svg](https://es.m.wikipedia.org/wiki/Archivo:HTML5_logo_and_wordmark.svg)).

## CSS

*Cascading Style Sheets*, en español, Hojas de estilo en cascada, es un lenguaje de hojas de estilo que permite separar el contenido de la presentación, proporcionando una manera eficiente y flexible de controlar el diseño y el estilo de las páginas web. Al igual que *HTML*, *CSS3* es un estándar vivo, es decir, que sigue siendo desarrollado, mantenido y expandido por la *World Wide Web Consortium*.

CSS fue propuesto y diseñado por Håkon Wium Lie en conjunto a Bert Bos el 10 de octubre de 1994 (Håkon, W., S.F.). El desarrollo de CSS inició durante la estadía de Lie en el *CERN* (Organización Europea para la Investigación Nuclear) mientras trabajaba con Tim Berners-Lee.

En la década de los noventa, derivado de una convocatoria por parte la *W3C* para la creación de un lenguaje de hojas de estilos específico para el lenguaje *HTML*, diversas propuestas de lenguajes de hojas de estilos fueron presentadas, la *W3C* terminó por adoptar y generar la primera Recomendación CSS en 1996, esta recomendación fue denominada *CSS1* (López, J., 2009).

CSS fue diseñado con el objetivo de marcar la separación del contenido del documento y su presentación, el objetivo de esta separación es mejorar, simplificar y añadir accesibilidad al documento. Este lenguaje permite, además, una mayor flexibilidad y control en la especificación de características presentacionales, además de reducir la redundancia de código entre páginas *HTML*, haciendo uso de un solo archivo de hoja de estilos. La Figura 1.7 Muestra el logotipo de CSS.



Figura 1. 7. Logotipo de CSS. Adaptado de *CSS3 logo and wordmark*. Por *Wikipedia contributors*. (s.f.). *Wikipedia, The Free Encyclopedia*. ([https://es.m.wikipedia.org/wiki/Archivo:CSS3\\_logo\\_and\\_wordmark.svg](https://es.m.wikipedia.org/wiki/Archivo:CSS3_logo_and_wordmark.svg)).

## JavaScript

*JavaScript* es un lenguaje de programación creado por Brendan Eich durante su estadía en *Netscape Communications Corporation* durante 1995, es un lenguaje interpretado, dialecto del estándar *ECMAScript*, el uso más común de *JavaScript* es el de escribir funciones embebidas que interactúan con el *Document Object Model (DOM)* de la página.

Con el fin de estandarizar *JavaScript*, *Netscape*, envió especificaciones del lenguaje a *ECMA*, *ECMA* es una organización europea encargada de hacer estándares de tecnologías. Para el año 1997, *ECMA* lanzó la primera versión de su estándar, llamado *ECMAScript 1* (López, Y., 2023).

*JavaScript* es utilizado para crear interfaces y páginas interactivas, crear aplicaciones del lado del servidor, conectar bases de datos y mucho más. Su código de programación se ha convertido en una parte esencial del desarrollo web moderno, derivando a que cualquier navegador moderno cuente con una implementación de *JavaScript*, puede encontrarse en prácticamente cualquier aplicación web. La Figura 1.8 Muestra el logotipo de *JavaScript*.



Figura 1. 8. Logotipo de JavaScript. Adaptado de Javascript badge. Por Wikimedia Commons. (s.f.). Wikimedia.org. ([https://commons.wikimedia.org/wiki/File:Javascript\\_badge.svg](https://commons.wikimedia.org/wiki/File:Javascript_badge.svg)).

## PHP

*PHP* es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web debido a que puede ser incrustado en *HTML*. *PHP* funciona del lado del servidor, a diferencia de otros lenguajes como *JavaScript*, *PHP* genera código *HTML* y es enviado al cliente, el cliente solo observará el resultado del código *PHP*, no el código *PHP* en sí.

*PHP* fue creado inicialmente por Rasmus Lerdorf en el año 1994. En la actualidad, la implementación de referencia de *PHP* es producida por *The PHP Group* (PHP Documentation Group, 2024).

El lenguaje *PHP* se integra fácilmente con *HTML*, lo que permite crear y diseñar aplicaciones web dinámicas. Se pueden incluir scripts *PHP* directamente en el código *HTML*, facilitando la generación de contenido dinámico en las páginas web, de esta manera siendo una alternativa de *JavaScript* o, también, pueden ser integrados de manera conjunta. La Figura 1.9 Muestra el logotipo de *PHP*.



Figura 1. 9. Logotipo de PHP. Adaptado de PHP-logo. Por Wikipedia contributors. (s.f.). Wikipedia, The Free Encyclopedia. (<https://es.m.wikipedia.org/wiki/Archivo:PHP-logo.svg>).

## Laravel

*Laravel* es un *Framework* de desarrollo de aplicaciones web basado en *PHP*, contando como objetivo proporcionar de una sintaxis elegante y expresiva para crear código de forma sencilla. (Laravel Holdings Inc., 2024)

*Laravel* se basa en el patrón de arquitectura de *software MVC*, este atributo en conjunto a las herramientas integradas del *Framework* facilita la tarea de generar código bien estructurado y fácil de mantener, el cual en consecuencia hace del proceso de desarrollo de aplicaciones web una actividad mucho más rápida.

*Laravel* enfatiza la generación de funcionalidades y escalabilidad, aprovechando todas las características de las versiones más recientes de *PHP*, cada lanzamiento anual de *Laravel* implementa estas características (Laravel Holdings Inc., 2023).

*Laravel* cuenta con diversas herramientas y funcionalidades entre las más destacadas se encuentran:

- *Eloquent. ORM (Object Relational Mapping)*, el cual permite una manipulación mucho más flexible de la base de datos adaptándola a una abstracción de objetos.
- *Blade*. Sistema de plantillas que permiten modularizar y reutilizar componentes como código *HTML*, *CSS*, *JavaScript* y *PHP* de páginas web.
- Migraciones. un sistema de control estructura de la base de datos.
- Validaciones. Funcionalidad integrada para validar y sanitizar la información recibida por parte del usuario.
- *APIs* integrados de correo, notificaciones y seguridad.
- *Facades*. Herramienta para la invocación rápida de clases.

La Figura 1.10 Muestra el logotipo de *Laravel*.

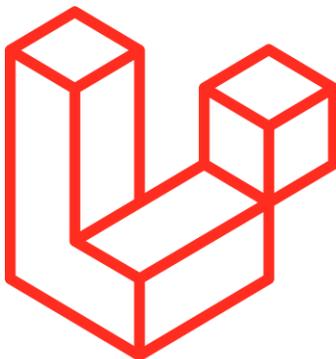


Figura 1. 10. Logotipo de Laravel. Adaptado de Laravel Logotipo Vector. Por Worldvectorlogo (s.f.) Worldvectorlogo.com (<https://worldvectorlogo.com/es/logo/laravel-2>).

## Composer

*Composer* es un gestor de dependencias y paquetes dedicado al lenguaje *PHP*, fue creado por Nils Adermann y Jordi Boggiano en año 2012 (Boggiano, J., 2024), permite la instalación de dependencias desde una línea de comandos, permite la instalación completa de aplicaciones *PHP*.

*Composer* es compatible con diversos *Frameworks* de *PHP*, incluido *Laravel*, esta compatibilidad permite trabajar ambas herramientas en conjunto, el uso de *Composer* permite una gestión más eficiente de los paquetes y librerías que pueden utilizarse durante el desarrollo de una aplicación web.

La selección de librerías y paquetes que pueden instalarse desde *Composer* es vasta y actualizada, gracias al uso de *Packagist*, un repositorio de librerías de *PHP* dedicado para *Composer*. La Figura 1.11 Muestra el logotipo de *Composer*.



Figura 1. 11. Logotipo de Composer. Adaptado de Logo-composer-transparent. Por Wikipedia contributors. (s.f.). Wikipedia, The Free Encyclopedia. (<https://es.m.wikipedia.org/wiki/Archivo:Logo-composer-transparent.png>).

## Livewire

*Livewire* es un *Framework* para el desarrollo de componentes dinámicos con alta compatibilidad para *Laravel*, permite la generación rápida de componentes dinámicos para las vistas de un sistema, trabajando de manera conjunta con el controlador de un proyecto de *Laravel*, manipulando cuantas vistas se deseen, gracias a su implementación conjunta con las plantillas *Blade* de *Laravel*. La Figura 1.12 Muestra el logotipo de *Livewire*.



Figura 1. 12. Logotipo de Livewire. Adaptado de Laravel Livewire Logo. Por Logowik. (2024). Logowik.com (<https://logowik.com/laravel-livewire-logo-vector-65669.html>).

## **jQuery**

*jQuery* es una librería de *JavaScript* creada por John Resig en el año 2006 la cuál proporciona de herramientas para agilizar el desarrollo de código dinámico, facilitando la manipulación del *DOM* de las vistas, manejar eventos, crear animaciones y agregar *AJAX* (*Asynchronous JavaScript And XML*), *jQuery* evita el proceso de reescribir código y/o funciones de *JavaScript* desde cero. La Figura 1.13 Muestra el logotipo de *jQuery*.



*Figura 1. 13. Logotipo de jQuery. Adaptado de JQuery-logo. Por Wikipedia contributors. (s.f.). Wikipedia, The Free Encyclopedia. (<https://es.m.wikipedia.org/wiki/Archivo:jQuery-Logo.svg>).*

## **Toastr**

*Toastr* es una librería para crear mensajes de notificación o alerta al usuario final, las notificaciones que puede desplegar son de carácter informativo, éxito, advertencia y error, cada contando con un color representativo como azul, verde, amarillo y rojo, estas notificaciones pueden ser desplegadas en distintas partes de la pantalla además de contar con la opción de utilizar distintas animaciones y funciones a las notificaciones.

En el sistema esta librería será utilizada en conjunto al calendario para desplegar notificaciones de acuerdo a las acciones realizadas por el usuario, estas notificaciones son de carácter temporal, volviéndose idóneas en el contexto de consultas *AJAX* del calendario.

## **Bootstrap**

*Bootstrap* es un framework de frontend basado fuertemente en *HTML* y *CSS* puro, incluye una amplia gama de elementos de diseño, como formularios, botones y menús que se adaptan a diferentes formatos de navegación. Los componentes de *Bootstrap* son fáciles de implementar y son altamente compatibles con otros *Frameworks*, adaptándose de manera eficaz a ambientes de navegadores móviles.

*Bootstrap* será utilizado en el diseño del panel administrativo y páginas de aterrizaje, el panel administrativo *AdminLTE 3*, será implementado, con el fin de otorgar de las distintas vistas del sistema una interfaz intuitiva, sencilla y dinámica para los usuarios finales.

## TailwindCSS

*TailwindCSS* es un *Framework* de frontend creado por Adam Wathan y Steve Schoger en el año 2017, *TailwindCSS* ofrece una gran variedad de utilidades para diseñar y generar componentes e interfaces dinámicas con una mayor libertad y flexibilidad.

El enfoque de esta librería es ofrecer una gran gama de clase de utilidad y no componentes predefinidos, como lo es el caso del framework *Bootstrap*, esto es posible gracias a un uso intensivo de *JavaScript* para crear cada una de las clases dedicadas. La Figura 1.14 Muestra el logotipo de *TailwindCSS*.



Figura 1. 14. Logotipo de TailwindCSS. Adaptado de Tailwind\_CSS\_logo. Por Wikipedia contributors. (s.f.). Wikipedia, The Free Encyclopedia. ([https://en.m.wikipedia.org/wiki/File:Tailwind\\_CSS\\_logo.svg](https://en.m.wikipedia.org/wiki/File:Tailwind_CSS_logo.svg)).

## DOMPDF

*DOMPDF* es una librería de *PHP* la cual adapta código *HTML* y *CSS* a un formato *PDF*, en conjunto a esto, *DOMPDF* puede manejar imágenes en formatos como *JPEG* y *PNG*, además de ajustar el texto a distintos tipos de página y orientación, siendo idónea para exportar código *HTML* y *CSS* estructurado en documentos.

## Laravel Excel

*Laravel Excel* es una librería de *Laravel* que permite generar y administrar de manera rápida archivos *Excel*, la principal cualidad de *Laravel Excel* es su capacidad de importar y exportar archivos de *Excel* sin importar si estos cuentan con estilos y/o formatos personalizados particulares. La Figura 1.15 Muestra el logotipo de *Laravel Excel*.



Figura 1. 15. Logotipo de Laravel Excel. Adaptado de Laravel\_Excel\_Spartner. Por Spartner. (s.f.). Spartner Supercharged Excel exports and imports in Laravel. (<https://laravel-excel.com>).

## Laravel Telegram

*Laravel Telegram* es una librería que permite la generación rápida de notificaciones por medio de un *Bot* de *Telegram* hacia un usuario o grupo en específico únicamente requiriendo de un ID del chat y credenciales del *Bot*.

Esta librería es utilizada para la generación de notificaciones desde el sistema web, estas notificaciones podrán ser generadas por medio de eventos generados en el calendario o el envío directo de notificaciones con el que cuenta el sistema.

## Spatie Laravel Permission

*Laravel Permission* es paquete el cual proporciona de un conjunto de herramientas para administrar los permisos con lo que cuenta un usuario dentro del sistema, tomando de base la generación y asignación de roles para estas acciones, el paquete ofrece sus propias tablas de bases de datos y modelos dedicados para su funcionamiento, otorgando de este paquete de una gran flexibilidad y fácil integración en un cualquier proyecto basado en *Laravel*.

El paquete es implementado con el fin de integrar una administración eficiente de permisos y roles disponibles para cada usuario al Sistema de Inventario *Airone*, sin necesidad de realizar configuraciones avanzadas al sistema web, gracias a *Laravel Permission* un administrador del sistema podrá gestionar todo esto con solo un par de *clicks*.

## FullCalendar

*FullCalendar* es una librería basada en *jQuery* creada por Adam Shaw la cual despliega un calendario que puede adoptar una forma de agenda, es altamente compatible con una implementación *AJAX* por parte de *jQuery*, otorgando del calendario de una interactividad superior. La Figura 1.16 Muestra el logotipo de *FullCalendar*.



Figura 1. 16. Logotipo de FullCalendar. Adaptado de FullCalendar, Por Drupal (s.f.). Drupal.org. (<https://www.drupal.org/files/project-images/fullcalendar.png>).

## Apache HTTP Server

*Apache HTTP Server* es un servidor HTTP de código abierto creado y mantenido por la *Apache Software Foundation* desde el año de 1995, este *software* tiene el propósito de procesar solicitudes *HTTP* y páginas web, es una herramienta necesaria para desplegar servidores web, es idóneo para el desarrollo de aplicaciones web y pruebas locales.

El uso de este *software* en este desarrollo será principalmente el de alojar al sistema web, volviéndolo accesible dentro de una red de área local, siendo indispensable para el correcto despliegue del sistema web. La Figura 1.17 Muestra el logotipo de *Apache HTTP Server*.



Figura 1. 17. Logotipo de Apache HTTP Server. Adaptado de *Apache\_HTTP\_server\_logo(2019-Present)*. Por *Wikimedia Commons*. (s.f.). *Wikimedia.org*. ([https://commons.wikimedia.org/wiki/File:Apache\\_HTTP\\_server\\_logo\\_\(2019-present\).svg](https://commons.wikimedia.org/wiki/File:Apache_HTTP_server_logo_(2019-present).svg)).

## MySQL

*MySQL* es un sistema gestor de bases de datos relacionales, esta permite la administración de bases de datos, tablas, relaciones, vistas, usuarios y tuplas, además de una conexión sencilla con diversos lenguajes de programación, incluidos aquellos dedicados a la programación web.

Su uso en el sistema web radica en que es el núcleo de toda la información que será almacenada para el sistema, desde aquella que puede registrar cada usuario, los usuarios en sí, registros de elementos, eventos, configuración de los usuarios, registros necesarios para las librerías, entre muchos otros datos más, este *software* es imprescindible para el correcto funcionamiento del sistema. La Figura 1.18 Muestra el logotipo de *MySQL*.



Figura 1. 18. Logotipo de MySQL. Adaptado de *MySQL Logo*. Por *Buytaert, D.* (2010). *The history of MySQL AB*. (<https://dri.es/the-history-of-mysql-ab>).

## AdminLTE 3

*AdminLTE 3* es un panel de administración diseñado para su uso conjunto con el *Framework Bootstrap* y *jQuery*, esta librería ofrece diversos menús de navegación y plantillas que pueden implementarse a un sistema web, además de facilidades como *Widgets*, estilos, iconos, complementos de formularios e integraciones de diversas librerías y *plugins* de *JavaScript*.

## Git

*Git* es un sistema de control de versiones distribuido, creado por Linus Torvalds, este sistema permite el desarrollo de un *software* en forma de un repositorio distribuido, otorgando de cada contribuidor del proyecto de una copia del repositorio, permitiendo ser modificado incluso sin conexión.

Las modificaciones individuales de cada contribuidor se pueden añadir de manera no centralizada, este tipo de acercamiento permite que las modificaciones puedan ser fusionadas a un repositorio principal, donde todos los colaboradores podrán observar los cambios realizados, facilitando el rastreo y generación de un historial de cambios al *software*. La Figura 1.19 Muestra el logotipo de *Git*.



Figura 1. 19. Logotipo de Git. Adaptado de Full color Git logo for light backgrounds. Por Long. J. (s.f.). Git-logo downloads. (<https://git-scm.com/downloads/logos>).

## 1.4 Marco Metodológico

---

Durante el análisis del presente desarrollo se han considerado tres posibles metodologías de desarrollo de *software* para el desarrollo del Sistema de Inventario *Airone*, cada una aportando un conjunto de beneficios e impedimentos, éstas son:

### SOHDM

La metodología *SOHDM* (*Scenario-based Object-oriented Hypermedia Design Methodology*) es una metodología de desarrollo de *software* web, la cual se basa fuertemente en tecnologías orientadas a objetos, esta metodología propone y hace uso de escenarios como principal recurso para la abstracción y definición de los requisitos del usuario (Lee et al., 1998), dentro de esta definición se hace un uso extensivo de diagramas de flujos de datos de acuerdo a la notación de Yourdon DeMarco.

A partir de este conjunto de escenarios se obtiene como producto final un modelo conceptual del sistema en un diagrama de clases. El proceso de *SOHDM* continúa fusionando estos diagramas de clases para conseguir un modelo de clases navegacionales del sistema, otorgando de una definición clara de los enlaces entre páginas, simplificando la implementación y construcción del sistema.

Todo este procedimiento y arquitectura de la metodología *SOHDM* consiste de seis fases:

1. *Análisis del dominio*. Se realiza un estudio de las necesidades del usuario, condiciones de la aplicación, el entorno de trabajo y las actividades de los usuarios y/o entidades externas. La finalidad principal de esta fase es determinar, a través de Diagramas de contexto y listas de eventos, los escenarios que representarán las actividades que se pueden realizar dentro con el sistema.
2. *Modelado de objetos*. Con el fin de modelar adecuadamente cada clase implicada en el sistema se genera un diagrama de clases, en forma de tarjetas *CRC* (*Class-Responsibility-Collaborator*), que representará una estructura al nivel del modelado del sistema.
3. *Diseño de Vistas*. Se reorganizan los objetos en unidades navegacionales como páginas web que representan una vista de los objetos del sistema.
4. *Diseño Navegacional*. Se enlazan las vistas definiendo hiperenlaces navegables dentro del sistema.
5. *Diseño de la Implementación*. Se realiza todo el boceto de páginas, interfaces y base de datos del sistema.
6. *Construcción*. Se construye la base de datos relacional y, a continuación, es implementada dentro del sistema.

La Figura 1.20 muestra las fases de la metodología *SOHDM*, además de las interacciones entre sus fases, en conjunto a los productos generados por parte de cada una de estas.

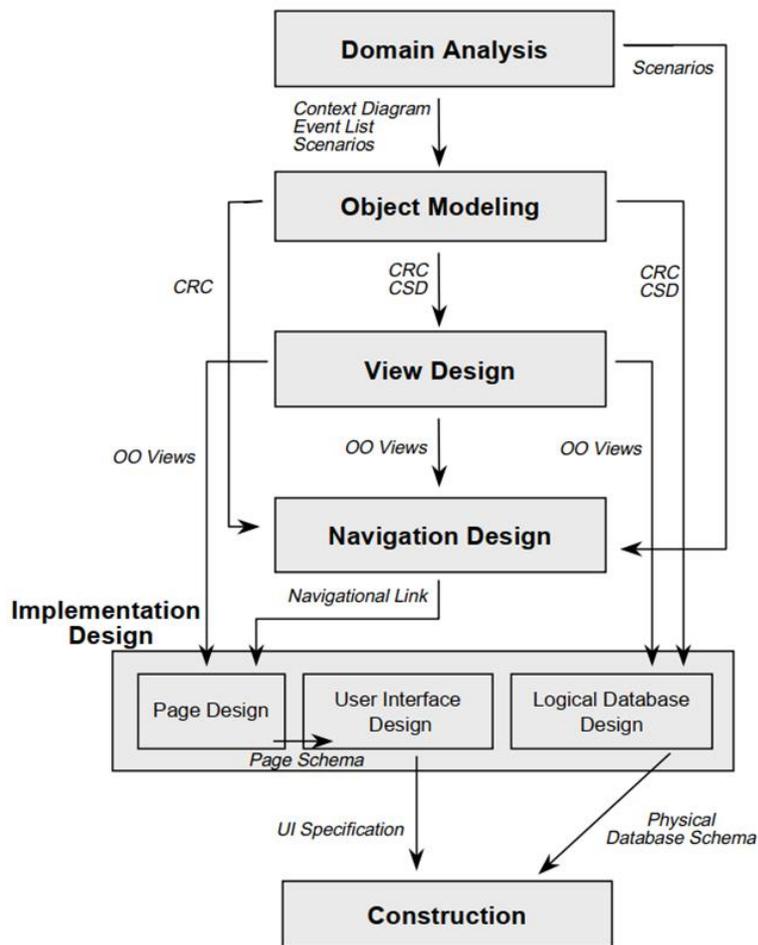


Figura 1. 20. Arquitectura de la Metodología SOHDM. Adaptado de *Architecture of SOHDM* (p.48). Lee, H., Lee, C. y Yoo, C. (1998). *Scenario-based object-oriented methodology for developing hypermedia information systems. Proceedings of the Hawaii International Conference on System Sciences* (p. 47-56).

## Metodología en Espiral

La metodología en espiral es un enfoque de desarrollo de *software* desarrollado por Barry Boehm, la metodología en espiral consiste de un modelo iterativo enfocado a la evolución del proceso del *software* mediante la el diseño y desarrollo de prototipos, tomando con los aspectos sistémicos de la metodología en cascada. Tiene el potencial para hacer un desarrollo rápido de versiones incrementalmente más completas. Las etapas de la metodología incremental son:

1. *Objetivo y determinación alternativa (Planeación y Comunicación)*. Los objetivos son determinados y negociados conjuntamente con el cliente; También se discuten posibles alternativas, es común dentro de esta también especificar las condiciones del entorno y desarrollo.
2. *Análisis y evaluación de riesgos (Modelado)*. Se identifican y evalúan los riesgos potenciales. También se evalúan las alternativas existentes. Los riesgos son registrados, evaluados y luego reducidos a través de prototipos como plantillas de diseño o componentes funcionales, simulaciones y *software* de análisis.
3. *Desarrollo y pruebas (Construcción)*. Los prototipos son desarrollados a fondo, añadiendo e implementando funcionalidades. El código es probado en un entorno de prueba por varias iteraciones, hasta que el *software* pueda ser validado e implementado en un entorno de producción.
4. *Planificación del siguiente ciclo (Despliegue)*. El cliente evalúa si los objetivos propuestos en el ciclo presente se han cumplido, conjuntamente, el siguiente ciclo se planifica en esta etapa. Si se producen errores, se buscan soluciones, si una alternativa otorga de una mejor solución, se adoptará en esta una mayor preferencia en el siguiente ciclo.

La Figura 1.21 muestra las fases de la metodología en espiral y su naturaleza iterativa.

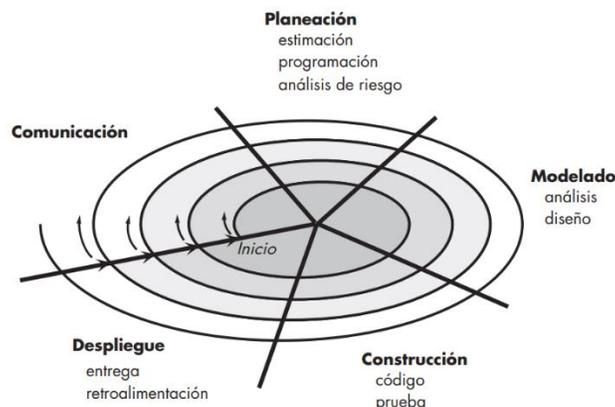


Figura 1. 21. Arquitectura de la Metodología en Espiral. Adaptado de Modelo de Espiral Común (p.39). De Pressman, S. (2010). Ingeniería del Software, Un enfoque Practico, Séptima Edición. McGraw Hill.

## Programación Extrema (XP)

La programación extrema (XP) es una metodología de desarrollo de *software* desarrollada por Kent Beck, Ward Cunningham y Ron Jeffries a finales de los noventa (Wells, 2009). Los valores de la programación extrema (comunicación, simplicidad, retroalimentación, valentía y respeto) son el motor para las actividades y tareas específicas de XP. El nombre de esta metodología se origina por la acción de la metodología de llevar a niveles extremos las prácticas reconocidas, como el desarrollo iterativo (Sommerville, 2011, p. 64-65).

La programación extrema se basa en las relaciones interpersonales, promoviendo el trabajo en equipo, aprendizaje de los desarrolladores y el clima laboral, la programación extrema depende en gran medida de la retroalimentación continua entre el cliente y el equipo de desarrollo, la sencillez en las soluciones y coraje para afrontar cambios son vitales para el desempeño efectivo en las actividades de esta metodología.

Esta metodología de desarrollo de *software* es especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Maida, 2015, p.58). La programación extrema se respalda de 5 fases, las cuales se describen a continuación (Melendez, et al., 2016):

1. *Planeación*. Dentro del ámbito de la programación extrema la planificación consta de un diálogo continuo entre las partes involucradas en el proyecto, estas partes se conforman del cliente, coordinadores y programadores respectivamente.

Un proyecto que hace uso de la programación extrema comienza recopilando las historias de usuarios, estas historias se conforman principalmente de casos de uso. La recopilación de este conjunto de historias permite a los programadores evaluar de manera efectiva el tiempo de desarrollo de cada una de las historias proporcionadas por el cliente. Durante la planeación es común generar productos como Planes de entregas (*Release Plan*) e iteraciones (*Iteration Plan*) así como una calendarización de las reuniones de seguimiento conjuntas al cliente, denominadas *Stand-Up Meeting*.

2. *Diseño*. Esta fase consta de la creación y prototipado de estructura del sistema, la utilización de artefactos visuales es vital para el desarrollo de esta estructura, estos artefactos incluyen como mínimo, tarjetas *CRC* diagramas orientados a objetos, diagramas de flujo, entre otros artefactos visuales (Sinnaps, 2020).

Existe la posibilidad de que el diseño de algunas historias del usuario pueda llegar a ser complejo o derivar en ambigüedades, ante esta problemática, la programación extrema sugiere la creación inmediata de un prototipo a nivel operativo, este prototipo actuará como un recurso para disminuir el riesgo derivado de la complejidad de la historia, esta situación se denomina puntualmente *Spike* o solución en punta (Pressman, S., 2010, p.63).

3. *Codificación.* La fase de codificación consiste primero de la generación de un conjunto de pruebas unitarias, estas pruebas tienen como fin el evaluar el producto que generará durante esta fase, el fin de estas pruebas es el de generar una retroalimentación inmediata y de disponer de herramientas adecuadas para la generación de una futura prueba unitaria universal.

Durante la labor de codificación, se incita a la programación en parejas, las parejas de programadores se mantienen en constante cambio y rotación, esta forma de programar promueve la propiedad colectiva del código, revisión informal constante del código y la mejora constante del *software* (Sommerville, I., 2011, p. 71).

4. *Pruebas.* Esta fase parte de la premisa de pruebas unitarias de la fase de codificación, las pruebas unitarias recopiladas son adaptadas y elaboradas para permitir su automatización, con el fin de que sean fáciles de ser ejecutadas repetidas veces, facilitando su iteración.

Dentro de estas pruebas unitarias destacan las pruebas de aceptación XP, o también denominadas pruebas del cliente, son un conjunto de pruebas y requisitos especificados por parte del cliente, este tipo de pruebas normalmente se centran en aspectos como las características y/o funcionalidades del sistema, dando énfasis en todos los aspectos que son visibles y revisables por parte del cliente. (Pressman, S., 2010, p.64-65).

5. *Lanzamiento.* Una vez implementado y validado con éxito todas las historias del usuario, el sistema es integrado al entorno solicitado por el cliente, en esta fase es posible realizar una labor de inspección, mantenimiento y análisis *a posteriori* del sistema.

En la Figura 1.22 se presentan todas las fases del proceso de programación extrema.



Figura 1. 22. Arquitectura de la Metodología XP. Adaptado de *El proceso de la Programación extrema* (p.62). De Pressman, S. (2010). *Ingeniería del Software, Un enfoque Practico, Séptima Edición.* McGraw Hill.

La Tabla 1.1 muestra una comparativa entre las principales características de las metodologías de desarrollo de *software* SOHDM, Espiral y Programación Extrema:

<b>Aspecto</b>	<b>SOHDM</b>	<b>Espiral</b>	<b>Programación Extrema</b>
<b>Enfoque</b>	Enfoque híbrido, orientado a sistemas web.	Iterativo con énfasis en la gestión de riesgos.	Desarrollo ágil con énfasis en la colaboración del equipo, la retroalimentación rápida y la mejora continua.
<b>Gestión de Riesgos</b>	Incorporación de elementos de gestión de riesgos de manera integrada por parte de la metodología.	Cada ciclo de la espiral comienza con la identificación y mitigación de riesgos.	Menor enfoque explícito en la gestión de riesgos, confiando en la retroalimentación continua para mitigar problemas.
<b>Tipo de Iteración de fases</b>	Iteraciones ajustadas según las necesidades del proyecto; flexibilidad en la adaptación del modelo.	Iteraciones cíclicas, donde cada ciclo agrega componentes y refina el sistema.	Iteraciones cortas (con una duración aproximada de 2 semanas), con entregas continuas y ajustes basados en retroalimentación.
<b>Flexibilidad</b>	Alta, la naturaleza híbrida de la metodología permite seleccionar y combinar enfoques según el proyecto.	Moderada, su capacidad de adaptar las fases de cada ciclo depende en gran medida del nivel de riesgo.	Muy alta, XP cuenta con la capacidad de responder rápidamente a cambios en requisitos y entorno.
<b>Documentación</b>	Extensa y estructurada para garantizar los objetivos, se apoya en gran medida en el diseño orientado a objetos.	Moderada, la documentación se enfoca en la gestión de riesgos y en la planificación de cada ciclo.	Menos documentación formal, con énfasis en la comunicación cara a cara y el código como principal fuente de documentación.
<b>Uso</b>	Proyectos complejos donde se requieren múltiples enfoques y una integración detallada de subsistemas.	Proyectos donde los riesgos son altos y deben ser gestionados continuamente.	Proyectos con alta incertidumbre en los requisitos, donde la velocidad de entrega y flexibilidad son claves.

Tabla 1. 1 Comparativa entre metodologías de desarrollo de Software.

## 1.5 Inventario y administración educativa

---

El abordar la definición de un sistema de inventario es necesario para adquirir una noción efectiva de las funcionalidades que cubrirá el Sistema de Inventario *Airone*, en conjunto a esta definición se cubrirán definiciones afines a la administración educativa, ambos ámbitos se entrelazan en el desarrollo del sistema en mayor o menor medida, a continuación, se describirán ambos términos.

### 1.5.1 Inventario

Un inventario es una lista o registro de bienes y recursos que una empresa tiene en existencia en un periodo dado, este periodo puede ser un lapso de tiempo definido o que integre información reciente de manera incremental. El inventario efectivo debe de ser explícito con las características y valor de cada uno de los elementos que integran el patrimonio de una empresa además de presentar un orden y clasificación apropiados para cada uno de los elementos que conforman el inventario.

Los inventarios pueden documentar información de bienes finales (Listos para ser utilizados), intermedios (En desarrollo o necesarios para el proceso productivo de otros bienes) o capitales (Necesarios para sustentar, mejorar o acelerar la producción de otros bienes) (Editorial Etecé, 2023). Los inventarios permiten y otorgan a las empresas las herramientas necesarias para conocer la cantidad y valor de sus existencias, ofreciendo las pautas necesarias para gestionar de manera eficiente el flujo de bienes y recursos.

Una gestión eficiente del inventario otorga muchos beneficios, los más importantes son minimizar la pérdida o deterioro de productos además de la toma de decisiones basadas en datos, gracias a la consulta y análisis de datos organizados del inventario (Aprende-logística, 2021).

### 1.5.2 Sistema de inventario

Un sistema de inventario es una herramienta, comúnmente de *software* con el propósito de automatizar los procedimientos de gestión y seguimiento de bienes, productos y materiales de una organización (Simpliroute, 2023). Cuenta con herramientas dedicadas para rastrear transacciones, generar reportes detallados sobre el inventario de manera individual para cada elemento o en conjuntos. Los sistemas de inventario se dividen en 3 tipos principales (Hearson, J., 2024).

- *Sistema de inventario manual*. Se basa en el recuento físico de artículos, el registro y documentación de detalles se realiza con el uso de papel y/o *software* de hojas de cálculo como *Excel*.

- *Sistema de inventario periódico.* Este sistema de gestión del inventario implica el recuento físico del inventario en lapsos de tiempo programados, los datos se introducen en un *software* de seguimiento del inventario dedicado. Este sistema es idóneo en empresas no comerciales, como las pertenecientes al sector salud o las administraciones públicas, por ejemplo, instituciones educativas o departamentos gubernamentales.
- *Sistema de inventario perpetuo.* Este sistema utiliza una solución de *software* automatizada con el fin de proporcionar datos en tiempo real, la empresa realiza el seguimiento y actualiza el recuento del inventario mediante números de seguimiento de envíos, códigos de barras, etiquetas *RFID* y otros métodos automatizados, este proceso puede incluir aparatos como escáneres y cintas transportadoras.

### **1.5.3 Administración Educativa**

La administración educativa se encarga de la gestión de los centros educativos en función de los recursos humanos, financieros, pedagógicos y tecnológicos con el propósito de elevar cumplir con los objetivos de la institución educativa, así como promover la calidad académica y mejora de sus procesos. (Universidad Internacional de La Rioja. 2023)

La administración educativa, de acuerdo a su enfoque y particularidades sociopolíticas de su entorno, puede dividirse en cuatro categorías distintas (Universidad Alfonso X El Sabio, 2024), las cuales son:

- *Directiva.* Adopta un enfoque en las necesidades del centro escolar, haciendo uso de funciones administrativas y competencias derivadas del liderazgo, impulsando a todo un equipo de trabajo para satisfacer y solventar las necesidades, se basa fuertemente en el cumplimiento de metas.
- *Pedagógica.* Está plenamente enfocada en la actualización y ajuste de la oferta curricular con el fin de asegurar que todo plan educativo propuesto en la institución adopte un enfoque hacia una educación de calidad, haciendo uso de aportaciones del profesorado y la comunidad educativa.
- *Financiera y administrativa.* Está centrado en mantener una buena salud económica, priorizando la eficiencia de los procesos de organización dentro del instituto educativo. A través de la misma se gestionan todos los documentos vinculados con los estudiantes y transacciones.
- *Comunitaria.* Está altamente centrada en un desarrollo conjunto del ambiente estudiantil, laboral y colectivo, la contigüidad del instituto educativo con la comunidad educativa es crucial, y con el fin de promover la participación ciudadana, programas de intercambio y todos aquellos aspectos sociales derivados del instituto educativo.

## 1.6 Estado del arte

---

Los sistemas de inventario son un área común dentro del desarrollo de *software*, contando con una gran diversidad de estos, cada uno con herramientas generales hasta aquellos enfocados a ámbitos más específicos, cada sistema cuenta con fortalezas o funcionalidades que las hacen destacar, la adopción de uno u otro sistema depende en gran medida de las necesidades del usuario.

### 1.6.1 Holded

Es un *software* como servicio enfocado en la monitorización y administración de negocios ideal para su aplicación en ámbitos como lo son ventas, recursos humanos, inventario y contabilidad, análisis de negocio, entre otras funciones más.

Cada una de estas funcionalidades pueden ser adquiridas por un costo mensual, este costo depende de las funcionalidades que se desean utilizar, cada funcionalidad se divide en módulos denominados por *Holded* como *Gems*, el módulo dedicado a inventario tiene un costo de 25 EUR al mes.

El módulo de inventario permite realizar una gama de tareas que incluye la categorización de productos, supervisión de los elementos del inventario, gestión de productos por almacenes, generación de códigos QR y de barras, entre otras funciones.

*Holded* cuenta con una documentación muy vasta acerca de cómo utilizar el *software*, esta documentación incluye vídeos, foros y tutoriales, sin embargo, si se requiere de una asistencia más personalizada, es necesaria una inversión de 50 EUR mensuales.

### 1.6.2 Odoo

Consiste de un *software* como servicio y de escritorio de inventario de código abierto con alto énfasis en la implementación de diversas funcionalidades y aplicaciones que abarcan distintas áreas de una empresa.

*Odoo* cuenta con aplicaciones dedicadas para ventas, firmas electrónicas, documentos, alquiler, recursos humanos, VOIP, programación de citas, inventario, entre otras funciones.

El *software* permite utilizar de manera gratuita una de las distintas aplicaciones con las que cuenta, una cantidad superior sólo puede ser probada por un periodo de 15 días, después de este periodo de tiempo se puede adquirir una de las dos suscripciones mensuales disponibles. La primera suscripción, denominada “Estándar”, la cual tiene un costo de 180 MXN ofrece acceso a todas las aplicaciones disponibles de *Odoo*, listas para ser integradas en el ambiente de administración requerida por el usuario.

La segunda suscripción, denominada “Personalizada”, tiene un costo de 274 MXN, este tipo de suscripción cuenta con todos los beneficios de una suscripción “estándar”, además de herramientas para integrar la API de *Odoo* a aplicaciones externas, además de instalar el *software* de escritorio de *Odoo*, este *software On-premise* permite una integración más cercana y directa a equipos finales.

### **1.6.3 Skyware Inventory**

Es un entorno de inventario basado en la web, implementa una base de datos para el control de inventario dedicado, el *software* es flexible con distintos ámbitos los cuales se incluyen la Electrónica, Agricultura, Medicina, Transporte, Alimentos, entre muchos otros ámbitos.

*Skyware Inventory* se adapta de acuerdo al estilo de almacenamiento y clasificación de los elementos de inventario, permitiendo a los usuarios detallar si existen variantes de un mismo producto, el *software* también puede adaptarse y trabajar como un inventario aislado, es decir, no se implementan funcionalidades de venta de manera a menos que el usuario lo requiera.

Cada elemento de inventario puede definirse de acuerdo a una serie de atributos específicos de acuerdo al contexto donde se utiliza el sistema, incluyendo un identificador único, descripción, nombres, variantes, códigos de barras, *UPC*, *SKU* e incluso secciones personalizadas.

*Skyware Inventory* cuenta con cuatro tipos de suscripciones de acuerdo a las necesidades del usuario, el *software* puede utilizarse de manera personal de manera gratuita, si se desean opciones multiusuario y/o funcionalidades específicas pueden adoptarse licencias con costos desde los 3 USD, 6 USD y hasta 10 USD por usuario, es una elección idónea si se desea una contabilidad y gestión rápida de elementos de inventario.

### **1.6.4 Monstock**

Es una plataforma de *software* con énfasis en la administración de inventarios, en conjunto a labores de logística, contando con funcionalidades denominadas “características”, cada una de estas permite integrar en el sistema la gestión de actividades como productos, generación de recibos de compras, suministros, rentas, reposición inteligente, visualización de datos, inventario, movilidad, entre otras funcionalidades integradas a la plataforma.

*Monstock* permite adaptarse en ámbitos como la entrega a domicilio, Industria, informática, artesanías, servicios, química, salud, *E-commerce*, arquitectura, y más, la administración de elementos permite la generación de códigos QR y de barras, además de una imagen dedicada.

Los usuarios cuentan con una prueba gratuita de 14 días, una vez caducada esta prueba se puede adquirir una de las 4 suscripciones, las suscripciones con las que se cuentan son: (i) Esencial, con un costo de 15 EUR más 10 por usuario. (ii) Profesional, con un costo de 25 EUR más 20 por usuario. (iii) Empresa, con un costo de 40 EUR más 30 por usuario. (iv) Sin límites, la cuál requiere de consultar un presupuesto con *Monstock*.

### **1.6.5 Zoho Inventory**

Es un *software* de inventario basado en la web con alta flexibilidad a la elaboración de inventarios, principalmente tomando enfoque a ámbitos como inmobiliarias, diseño, consultorías, tecnología, educación, publicidad, diseño web, entre otros ámbitos más.

*Zoho Inventory* permite el registro de cualquier tipo de elemento de inventario, incluido bienes no materiales, cada artículo puede poseer de atributos pertenecientes a distintos ámbitos y contextos de una organización, por ejemplo, atributos físicos, marca del producto, identificadores, información dedicada para un contexto de venta, información de seguimiento, entre otros atributos más.

El *software* puede utilizarse de manera gratuita, pero contará con limitaciones, entre esas limitaciones se encuentran las de limitar el uso del sistema a un único usuario, se encuentra limitado el inventario a un “almacén” y solo se podrán generar 50 pedidos y facturas al mes.

*Zoho Inventory* tiene 4 tipos de suscripción anuales, *Standard*, *Professional*, *Premium* y *Enterprise* con costos de 579 MXN, 1399 MXN, 2299 MXN y 4399 MXN respectivamente, cada paquete expande de las capacidades del *software* acorde a la suscripción contratada.

En la Tabla 1.2 se realiza una comparativa entre los 5 distintos *software* abordados en este apartado y el Sistema de Inventario *Airone*.

Aspecto	Holded	Odoo	Skyware Inventory	Monstock	Zoho Inventory	Sistema de Inventario <i>Airone</i>
<b>Enfoque</b>	Uso sencillo, para empresas pequeñas.	Uso conjunto con otras aplicaciones como Shopify.	Uso estricto de gestión de inventario e informes.	Uso como ERP ( <i>Enterprise Resource Planning</i> ).	Uso conjunto en empresas pequeñas y medianas.	Solución de administración de inventario y actividades dedicada para el AAIyA
<b>¿Es gratuito?</b>	No.	Si.	No.	No.	Si.	Si.
<b>Características sobresalientes</b>	Facilidad de contabilidad, uso en módulos.	Altamente modular, compatible con tareas de recursos humanos.	Control avanzado de elementos de inventario, alertas automáticas.	Optimización de stock en tiempo real, función de traslado de elementos.	Generación de Kits de elementos, administración de múltiples almacenes.	Sistema de calendario, notificaciones, alto control de la actividad del usuario.
<b>Límite de usuarios</b>	15	Sin límite.	Definido por el usuario.	Definido por el usuario.	3 (Gratis), Definido por el usuario.	Definido por el AAIyA.
<b>Sistema de Almacenes</b>	Si.	Si.	No.	Si.	Si.	No.
<b>Funcionalidades con cargo extra</b>	Si.	Si.	Si.	Si.	Si.	No.
<b>Periodo de Trial</b>	14 días	14 días	14 días	14 días	14 días	-
<b>Plataformas</b>	Web, iOS, Android	Web, iOS, Android, Windows	Web	Web, iOS, Android	Web, iOS, Android	Web
<b>Capacidad de automatización</b>	Moderada	Alta	Baja	Alta	Alta	Media
<b>Escalabilidad</b>	Pequeñas empresas	Grandes y Medianas Empresas	Pequeñas y medianas empresas	Pequeñas y medianas empresas	Pequeñas y medianas empresas	Áreas Académicas de la UAEH

Tabla 1. 2. Comparación De Características entre Software de Inventario.

## Capítulo 2 . Metodología

---

Tras la determinación de una metodología de desarrollo durante el apartado de metodologías de desarrollo de *software* de este documento, se determinó que el desarrollo del Sistema de inventario *Airone* será de acuerdo a las fases, valores y principios de la programación extrema, durante este capítulo se mostrará la aplicación de esta metodología durante el desarrollo del Sistema de Inventario *Airone*.

Este capítulo se encontrará dividido de acuerdo a las fases de la metodología XP, iniciando por la planeación, apartado donde se describirán y analizarán los requisitos funcionales y no funcionales del usuario, apoyándose de Tarjetas de historia de usuario y *CRC*. El apartado de diseño contiene todos los diagramas y diseños relacionados a la estructura del sistema, base de datos y descripción de algunas funcionalidades principales para el sistema.

En el apartado de codificación se abordará la implementación de la lógica que será utilizada por el Sistema de Inventario *Airone*, para proporcionar las funcionalidades descritas durante las fases anteriores de la metodología.

Finalmente, el apartado de pruebas describe el conjunto de funcionalidades que serán probadas mediante un conjunto de pruebas, la descripción de estas pruebas en conjunto a sus resultados será desplegada en un formato.

### 2.1 Planeación

---

La etapa de planificación consistió de la recolección de las distintas historias de usuario, las historias de usuario son el recurso por el cual se hará análisis de toda la gama de condiciones, expectativas y necesidades del usuario, cada una necesaria para determinar con exactitud las actividades de este desarrollo, y así, generar un producto capaz de satisfacer todos los requerimientos del usuario.

El análisis de las historias del usuario implica realizar una selección de estas historias, definiendo un alcance claro de las funcionalidades del sistema, el producto de la selección será un marco de trabajo consistente para el desarrollo del sistema, gracias a la generación de tiempos de entrega e iteraciones. Las historias del usuario fueron generadas en base a una historia de usuario 0, esta historia de usuario es el producto del análisis, negociación y descripción de todas las necesidades del personal del Área Académica de Ingeniería y Arquitectura. Las Historias de Usuario restantes adoptan de base los requisitos recopilados en la Historia de Usuario 0, cada una de estas aborda a mayor detalle los requisitos descritos en esta, en conjunto a detalles más técnicos, referentes a su integración en la aplicación de la metodología XP. Las Tarjetas 2.1 a 2.8 muestran todas las historias de usuario recopiladas.

Historia de Usuario 0	
<b>Nombre:</b> Sistema de Administración de Elementos, Eventos y Mantenimientos del AAIyA	
<b>Usuario:</b> Personal Administrativo del AAIyA, Prestadores de Servicio Social del AAIyA.	<b>Tipo de Actividad:</b> Nueva
<p><b>Prueba Aceptación:</b> Generación de Sistema de inventario para el Área Académica de Ingeniería y Arquitectura, este debe de ser Accesible para el personal administrativo del AAIyA, así como prestadores de Servicio Social, debe de ser un sistema basado en la web.</p> <p>Los registros de los elementos deben contener atributos carácter general (Descripción, Fecha de adquisición, mantenimiento, categorías, fondos, etc.), a excepción de 6 categorías predefinidas en el sistema, estas categorías predefinidas podrán ser manejadas automáticamente por el sistema con registros dedicados para los atributos adicionales. El sistema no administrará bienes de papelería a petición del cliente. Se debe contar con una barra de búsqueda dedicada a la búsqueda de elementos.</p> <p>Integrar sistema de generación de reportes de mantenimiento con archivos exportables a formato PDF.</p> <p>Generación de Anotaciones de traspasos (Si se transfirió el objeto hacia otro edificio u otra área académica externa de la universidad) de manera automática.</p> <p>Posibilidad de corroborar el Estado del objeto (Si esta de alta o de baja). Calendario interactivo de eventos con notificaciones por medio de un canal dedicado de Telegram.</p>	<b>Referencia a Historia Previa:</b> N/A
<b>Programador Responsable:</b> Diego Ángel Barajas Pérez	<b>Iteración Asignada:</b> N/A
<b>Prioridad:</b> N/A	<b>Riesgo de desarrollo:</b> N/A
<b>Elementos a Terminar:</b> Sistema de Inventario <i>Airone</i> .	

Tarjeta 2. 1. Historia de Usuario 0. Sistema de Administración de Elementos, Eventos y Mantenimientos del AAIyA.

<b>Historia de Usuario 1</b>	
<b>Nombre:</b> Administración de Categorías, Fondos y Etiquetas	
<b>Usuario:</b> Personal Administrativo del AAIyA, Prestadores de Servicio Social del AAIyA.	<b>Tipo de Actividad:</b> Nueva
<b>Prueba Aceptación:</b> El usuario puede Crear, Editar, Visualizar y Eliminar Registros de Categorías, Fondos y Etiquetas que poseerán los elementos.	<b>Referencia a Historia Previa:</b> 0
<b>Programador Responsable:</b> Diego Ángel Barajas Pérez	<b>Iteración Asignada:</b> 1
<b>Prioridad:</b> Alta	<b>Riesgo de desarrollo:</b> Baja
<b>Elementos a Terminar:</b> Base de datos (Tablas de Categorías, Fondos y Etiquetas) Vistas de Administración de Categorías, Fondos y Etiquetas, Lógica para Administrar Registros.	

*Tarjeta 2. 2 Historia de Usuario 1. Administración de Categorías, Fondos y Etiquetas.*

<b>Historia de Usuario 2</b>	
<b>Nombre:</b> Administración de Registros de Elementos de Inventario	
<b>Usuario:</b> Personal Administrativo del AAIyA, Prestadores de Servicio Social del AAIyA.	<b>Tipo de Actividad:</b> Nueva
<b>Prueba Aceptación:</b> El usuario puede Crear, Editar, Visualizar y Eliminar Registros de Elementos de Inventario, Adjuntando una imagen de cada registro, el usuario puede integrar atributos adicionales a 6 Categorías predefinidas.	<b>Referencia a Historia Previa:</b> 1
<b>Programador Responsable:</b> Diego Ángel Barajas Pérez	<b>Iteración Asignada:</b> 2, 6
<b>Prioridad:</b> Alta	<b>Riesgo de desarrollo:</b> Media
<b>Elementos a Terminar:</b> Base de datos (Tabla de Elementos, Atributos Adicionales, Relaciones con otras Tablas) Vistas de Administración de Registros de Elementos Barra de Búsqueda para Registros de Elementos Lógica para Administrar Registros de Atributos Adicionales.	

*Tarjeta 2. 3. Historia de Usuario 2. Administración de Registros de Elementos de Inventario.*

<b>Historia de Usuario 3</b>	
<b>Nombre:</b> Creación de Reportes de Mantenimiento Exportables a PDF	
<b>Usuario:</b> Personal Administrativo del AAIyA, Prestadores de Servicio Social del AAIyA.	<b>Tipo de Actividad:</b> Nueva
<b>Prueba Aceptación:</b> El usuario puede crear registros de mantenimiento parada cualquier elemento de inventario registrado en el sistema, una vez creados, estos pueden exportarse a un archivo en formato PDF.	<b>Referencia a Historia Previa:</b> 2
<b>Programador Responsable:</b> Diego Ángel Barajas Pérez	<b>Iteración Asignada:</b> 3
<b>Prioridad:</b> Alta	<b>Riesgo de desarrollo:</b> Media
<b>Elementos a Terminar:</b> Base de datos (Tablas de Mantenimientos, Relaciones con otras Tablas) Vistas de Creación de Reporte y Exportación de Reporte Generación y exportación de archivos PDF	

*Tarjeta 2. 4. Historia de Usuario 3. Creación de Reportes de Mantenimiento Exportables a PDF.*

<b>Historia de Usuario 4</b>	
<b>Nombre:</b> Generación automática de registros de Traspasos	
<b>Usuario:</b> Personal Administrativo del AAIyA, Prestadores de Servicio Social del AAIyA.	<b>Tipo de Actividad:</b> Nueva
<b>Prueba Aceptación:</b> El usuario al instante en el que edita la ubicación o edificio de un registro de elemento se generará de manera automática un registro de traspaso, el cual contiene la ubicación anterior y actual del elemento.	<b>Referencia a Historia Previa:</b> 2
<b>Programador Responsable:</b> Diego Ángel Barajas Pérez	<b>Iteración Asignada:</b> 3
<b>Prioridad:</b> Media	<b>Riesgo de desarrollo:</b> Baja
<b>Elementos a Terminar:</b> Base de datos (Tablas de Traspasos, Relaciones con otras Tablas) Vistas de Listado de Traspasos Lógica para Generar Registros de Traspasos	

*Tarjeta 2. 5. Historia de Usuario 4. Generación automática de registros de Traspasos.*

Historia de Usuario 5	
<b>Nombre:</b> Exportación e Importación de Registros De Elementos a través de un archivo XLSX	
<b>Usuario:</b> Personal Administrativo del AAIyA, Prestadores de Servicio Social del AAIyA.	<b>Tipo de Actividad:</b> Nueva
<b>Prueba Aceptación:</b> El usuario podrá exportar e importar registros de elementos de inventario a un archivo XLSX, la importación seguirá un formato predefinido para importar registros de elementos.	<b>Referencia a Historia Previa:</b> -
<b>Programador Responsable:</b> Diego Ángel Barajas Pérez	<b>Iteración Asignada:</b> 4
<b>Prioridad:</b> Media	<b>Riesgo de desarrollo:</b> Alta
<b>Elementos a Terminar:</b> Vistas de Exportación e Importación de Registros de Elementos Lógica para Exportación e Importación de Registros de Elementos por medio de un archivo XLSX	

*Tarjeta 2. 6. Historia de Usuario 5. Exportación e Importación de Registros De Elementos a través de un archivo XLSX.*

Historia de Usuario 6	
<b>Nombre:</b> Creación y Asignación de Roles a Usuarios del sistema.	
<b>Usuario:</b> Personal Administrativo del AAIyA, Prestadores de Servicio Social del AAIyA.	<b>Tipo de Actividad:</b> Nueva
<b>Prueba Aceptación:</b> El usuario podrá crear y asignar distintos roles a otros usuarios, limitando de esta manera las acciones que puede realizar.	<b>Referencia a Historia Previa:</b> -
<b>Programador Responsable:</b> Diego Ángel Barajas Pérez	<b>Iteración Asignada:</b> 2
<b>Prioridad:</b> Media	<b>Riesgo de desarrollo:</b> Baja
<b>Elementos a Terminar:</b> Base de datos (Tablas de Permisos y Roles) Vistas de Administración de Usuarios y Roles Lógica para Administración de Roles y asignación de roles a usuario	

*Tarjeta 2. 7. Historia de Usuario 6. Creación y Asignación de Roles a Usuarios del sistema.*

<b>Historia de Usuario 7</b>	
<b>Nombre:</b> Calendario Interactivo de Creación de Eventos y Notificación por Canal de Telegram	
<b>Usuario:</b> Personal Administrativo del AAIyA, Prestadores de Servicio Social del AAIyA.	<b>Tipo de Actividad:</b> Nueva
<b>Prueba Aceptación:</b> El usuario podrá crear, modificar y eliminar eventos por medio de un calendario interactivo, este calendario alojará eventos futuros a la fecha presente, estos eventos serán anunciados a través de un canal de notificaciones dedicado de Telegram, el usuario podrá difundir mensajes a este grupo de manera directa.	<b>Referencia a Historia Previa:</b> -
<b>Programador Responsable:</b> Diego Ángel Barajas Pérez	<b>Iteración Asignada:</b> 5
<b>Prioridad:</b> Alta	<b>Riesgo de desarrollo:</b> Media
<b>Elementos a Terminar:</b> Base de datos (Tablas de Eventos) Vistas de Administración de Eventos Lógica para Administración de Eventos y Difusión de eventos a canal de Telegram Lógica para Difusión de mensajes creados por el usuario	

*Tarjeta 2. 8. Historia de Usuario 7. Calendario Interactivo de Creación de Eventos y Notificación por Canal de Telegram.*

Tras la recopilación de requisitos del usuario se procedió a la generación de tarjetas *CRC*, cada una de estas tarjetas contiene un conjunto de responsabilidades y colaboraciones que se verán reflejadas en un modelo, controlador y vista del sistema, este acercamiento proporcionará de una definición de las acciones/funciones más concisas dentro del sistema, particularmente, gracias al seguimiento de las pautas de *MVC*, a continuación, las Tarjetas 2.9 a 2.19 muestran las tarjetas *CRC* generadas.

<b>Elemento</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>	<b>Clases:</b>	<b>Relación:</b>
Mostrar Listado Elementos	Mantenimiento	*...1
Registrar Elemento	Traspaso	*...1
Actualizar Elemento	Usuario	1...*
Editar Elemento	Fondos	1...*
Remover Elemento	Categorías	1...*
Mostrar Detalles	Etiquetas	*...*
Administrar atributos adicionales		

*Tarjeta 2. 9. CRC para la clase Elemento*

<b>Fondo</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>	<b>Clases:</b>	<b>Relación:</b>
Mostrar Listado Fondos	Elemento	*...1
Registrar Atributo		
Actualizar Atributo		
Editar Atributo		
Remover Atributo		

*Tarjeta 2. 10. CRC para la clase Fondo*

<b>Categoría</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>	<b>Clases:</b>	<b>Relación:</b>
Mostrar Listado Categorías	Elemento	*...1
Registrar Categoría		
Actualizar Categoría		
Editar Categoría		
Remover Categoría		

*Tarjeta 2. 11. CRC para la clase Categoría*

<b>Etiqueta</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>  Mostrar Listado Etiquetas Registrar Etiqueta Actualizar Etiqueta Editar Etiqueta Remover Etiqueta	<b>Clases:</b>  Elemento	<b>Relación:</b>  *...*

*Tarjeta 2. 12. CRC para la clase Etiqueta*

<b>Traspaso</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>  Generar Registro Mostrar Registro	<b>Clases:</b>  Elemento Ubicación Edificio	<b>Relación:</b>  1...* 1...* 1...*

*Tarjeta 2. 13. CRC para la clase Traspaso*

<b>Calendario</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>  Desplegar Eventos Crear Evento Editar Evento Eliminar Evento Anunciar Evento	<b>Clases:</b>  -	<b>Relación:</b>  -

*Tarjeta 2. 14. CRC para la clase Calendario*

<b>Notificación</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>  Enviar Notificación	<b>Clases:</b>  -	<b>Relación:</b>  -

*Tarjeta 2. 15. CRC para la clase Notificación*

<b>Exportación e Importación</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>  Exportar Base de Datos Exportar Registros de Elementos Importar Registros de Elementos	<b>Clases:</b>  -	<b>Relación:</b>  -

*Tarjeta 2. 16. CRC para la clase Exportación e Importación*

<b>Mantenimiento</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>  Mostrar Listado de Mantenimientos Crear Reporte de Mantenimiento Exportar Reporte de mantenimiento	<b>Clases:</b>  Elemento	<b>Relación:</b>  1...*

*Tarjeta 2. 17. CRC para la clase Mantenimiento*

<b>Usuario</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>  Registrar Usuario Eliminar Usuario Actualizar Usuario	<b>Clases:</b>  Elementos	<b>Relación:</b>  1...*

*Tarjeta 2. 18. CRC para la clase Usuario*

<b>Rol</b>		
<b>Responsabilidades</b>	<b>Colaboraciones</b>	
<b>Acciones:</b>  Mostrar Listado de Roles Crear Rol Eliminar Rol Editar Rol Asignar Roles	<b>Clases:</b>  Usuario	<b>Relación:</b>  1...*

*Tarjeta 2. 19. CRC para la clase Rol*

### **2.1.1 Definición de las funcionalidades principales**

La recopilación y análisis de las historias del usuario, en conjunto a la creación de tarjetas *CRC*, otorgó las pautas para definir adecuadamente un conjunto de funcionalidades que tendrá el sistema web, a continuación, se describirán las funcionalidades recopiladas.

#### ***Registro, Edición y Eliminación de registros de inventario***

El sistema cuenta con la capacidad de administrar registros individuales de elementos de inventario del Área Académica de Ingeniería y Arquitectura, cada registro cuenta con los siguientes atributos: (i) Nombre del elemento, (ii) Descripción del elemento, (iii) Fecha de adquisición, (iv) Fecha de su último mantenimiento, (v) Status de inventario, (vi) Ubicación particular del elemento, (vii) Edificio donde se encuentra el inventario, (viii) Categoría del elemento, (ix) Fondo con el que fue adquirido el elemento, (x) Etiquetas del elemento.

Esta capacidad del sistema es el núcleo de todo el sistema web, pues de esta funcionalidad parte la mayoría de funciones implementadas para el sistema.

#### ***Administración de Categorías, Fondos y Etiquetas***

El Sistema de Inventario *Airone* permitirá al usuario crear sus propias Categorías, Tipos de Fondos y Etiquetas, este apartado añadiendo una gran flexibilidad al sistema, adaptándose a las necesidades del usuario, sin forzarlo a recurrir a las opciones predeterminadas con las que cuenta el sistema en cada uno de estos apartados.

#### ***Atributos adicionales a las categorías predeterminadas del sistema***

El sistema contará con la capacidad de generar registros de atributos adicionales para las seis categorías predeterminadas del sistema de inventario, estas categorías son: (i) Computación, (ii) Inmobiliario, (iii) Infraestructura, (iv) Equipo de Laboratorio, (v) Maquinaria, (vi) Equipo de seguridad.

El propósito de estos atributos adicionales es el de corresponder una mayor cantidad de información a los tipos de elementos más comunes dentro del área, enriqueciendo con datos particulares de estos tipos de elementos.

#### ***Generación de reportes de elementos individuales en formato PDF***

El sistema dispondrá de un método de generación de reportes de mantenimiento individuales para los registros de elementos, estos reportes se encontrarán presentes en el sistema, disponibles para su revisión y conteo, estos registros de mantenimiento pueden ser exportados en formato PDF para su impresión, con el fin de complementar actividades del área académica involucradas con la generación de documentación de elementos.

### ***Exportación e Importación de registros en formato XLSX***

La funcionalidad de Exportación e Importación de registros es vital para el proceso de migración y adopción del Sistema de Inventario *Airone*, la recopilación de registros de inventario solía ser a través de diversas hojas de cálculo, dada esta situación, se tiene muy presente esta funcionalidad.

### ***Exportación la base de datos en formato SQL***

Además de la exportación de registros de elementos de inventario en formato XLSX, el sistema contará con la capacidad de exportar la base de datos completa del sistema, con el fin de proporcionar un medio para respaldar la información completa del sistema.

### ***Calendario Interactivo de actividades***

El Sistema de Inventario *Airone* proporcionará un calendario de actividades interactivo, el cual permite el registro de eventos en el sistema, cuando se cumple la fecha de un evento, se notificará en un canal de *Telegram* dedicado en el horario determinado por el usuario (11:00 AM).

### ***Administración de permisos y roles del sistema***

El Sistema de Inventario *Airone* contará con una herramienta dedicada a la administración de permisos y roles dentro del sistema, esta será altamente flexible de acuerdo a las necesidades del usuario, permitiendo la administración de roles en base a un conjunto de permisos, además de asignarlos a usuarios en específico, esta herramienta otorgará un grado de control sobre las acciones que puede realizar cada usuario.

## **2.1.2 Determinación de iteraciones para el desarrollo del Sistema de Inventario *Airone*.**

Una vez que se cuenta con un panorama claro acerca de las actividades, clases y requerimientos del usuario, se determinarán todas las iteraciones con las que contará el desarrollo del sistema, cada iteración tendrá una duración aproximada de 1 a 2 semanas de acuerdo a las funcionalidades que se implementarán, así como su complejidad técnica.

La disposición de las iteraciones será de la siguiente manera:

### ***Iteración 1 - Creación del proyecto y desarrollo de componentes base del sistema***

Esta primera iteración constara de tareas como la generación de un proyecto nuevo de Laravel, implementado un ambiente de desarrollo adecuado para el sistema, así como la generación de un repositorio en *GitLab* para alojar el sistema web.

La iteración tiene contemplada la Esquematación y generación de la base de datos del sistema, la implementación de un panel administrativo, un sistema de *Login* por medio de *Laravel Jetstream* y un *CRUD* de Elementos de Inventario, Etiquetas y Categorías.

- Creación del proyecto en *Laravel*
- Generación de un repositorio para el sistema en *GitLab*.
- Esquematación y generación de base de datos.
- Implementación de panel administrativo.
- Implementación de *Login* de *Laravel Jetstream*.
- *CRUD* de etiquetas, fondos y categorías.

### ***Iteración 2 - Implementación de sistema de roles y permisos, y atributos adicionales***

Esta iteración está centrada en la implementación del sistema de roles y permisos, haciendo uso de la Librería *Spatie Laravel-Permission*, además de la generación de toda la lógica y funciones dedicados para la implementación de los registros de atributos adicionales para los elementos con una categoría compatible.

- Implementación de librería *Spatie Laravel-Permission*.
- Agregar un *CRUD* de Roles.
- Asignación de Roles a usuarios.
- *CRUD* de elementos de inventario
- Implementación de la asignación de atributos adicionales a registros de elementos.

### ***Iteración 3 – Generación automática de registros de trasposos, exportación y generación de registros de mantenimiento***

La iteración tiene contemplada la automatización de la generación de una bitácora de trasposos, además de la integración apropiada de la creación y exportación de registros de mantenimiento.

- Introducir la lógica necesaria para automatizar la generación de registros de traspaso de acuerdo a la edición de los elementos.
- Opción para generar un registro de mantenimiento al momento de interactuar con las opciones de un registro de elemento en lista.
- Exportación de Registros de mantenimiento individuales a un documento en formato PDF.

#### ***Iteración 4 – Exportación e Importación de Registros de elementos***

Durante esta iteración se hará uso de *Laravel Excel*, con el propósito de Implementar un método para Exportar e Importar registros de elementos de inventario de manera rápida para cualquier usuario del sistema.

En conjunto a esta labor se diseñó la lógica necesaria para exportar un archivo en formato *SQL* con toda la base de datos del sistema.

- Uso de *Laravel Excel* para la Exportación e importación de registros de elementos de inventario.
- Implementación de Funcionalidad para importar registros de atributos adicionales.
- Exportación de la base de datos del sistema.

#### ***Iteración 5 – Notificaciones por medio de Telegram y Calendario Interactivo***

Esta Iteración está enfocada en la implementación de todo el sistema de notificaciones necesario para ser usado en el sistema, en conjunto, mediante el uso de *FullCalendar*, se hará la integración de un calendario interactivo compatible con el sistema de notificaciones.

- Generación de un canal de Notificaciones de Usuario en *Telegram*.
- Implementación de un sistema de envío de notificaciones a un canal de *Telegram*.
- Integración de un Calendario Interactivo de eventos.
- Envío automatizado de notificaciones de eventos de calendario a un canal de *Telegram*.

#### ***Iteración 6 - Cambios de interfaz, generación de barras de búsqueda e integración de página de aterrizaje.***

Esta última iteración tendrá un enfoque en la mejora de todas las interfaces del sistema, además de volverla mucho más accesible e intuitiva para cualquier usuario, volviendo mucho más sencilla su navegación, se realizará una corrección de errores identificados durante el desarrollo del sistema.

- Mejoras de experiencia de usuario (UX).
- Generación de interfaces más intuitivas.
- Cambios en la estética de las interfaces del sistema.
- Barras de búsqueda para usuario y elementos.
- Integración de página de *Landing* para el usuario.
- Generación de apartado de estadísticas en *Dashboard*.
- Corrección de errores identificados

## 2.2 Diseño

---

Durante esta fase de la metodología se desarrollarán un conjunto de interfaces y diseños de la arquitectura de la aplicación, tomando un enfoque en el patrón de diseño *MVC*, el uso del *Framework Laravel* como *Backend* en el sistema facilitará la implementación de esta arquitectura en el sistema, la integración y diseño de rutas del sistema también será definido en conjunto a otros componentes necesarios para el funcionamiento del sistema.

Finalmente, en esta fase se generarán un conjunto de prototipos de interfaz de usuario para cada una de las funcionalidades y navegación del sistema.

### 2.2.1 Diseño de la base de datos

El diseño de la base de datos del sistema implica la generación de dos tipos de diagrama, el primero de ellos es el diagrama Entidad-Relación, este tipo de diagrama fue propuesto por primera vez en el año 1976 por Peter Chen (Pressman, S., 2010, p. 142), con el propósito de diseñar bases de datos relacionales, este tipo de diagrama es ideal para representar todos los objetos de datos y relaciones dentro de un sistema, facilitando el proceso de abstracción

El diagrama Entidad-Relación del sistema estableció las pautas necesarias para generar un diagrama relacional del sistema, este diagrama describe las tablas principales del sistema. Así como sus relaciones, descartando aquellas generadas por el ambiente de desarrollo del *Framework Laravel*, particularmente, del paquete *Laravel Jetstream*. Cada una de estas tablas contiene los atributos que serán implementados en la base de datos de producción del sistema. La Figura 2.2 muestra el diagrama relacional del Sistema de Inventario *Airone*.

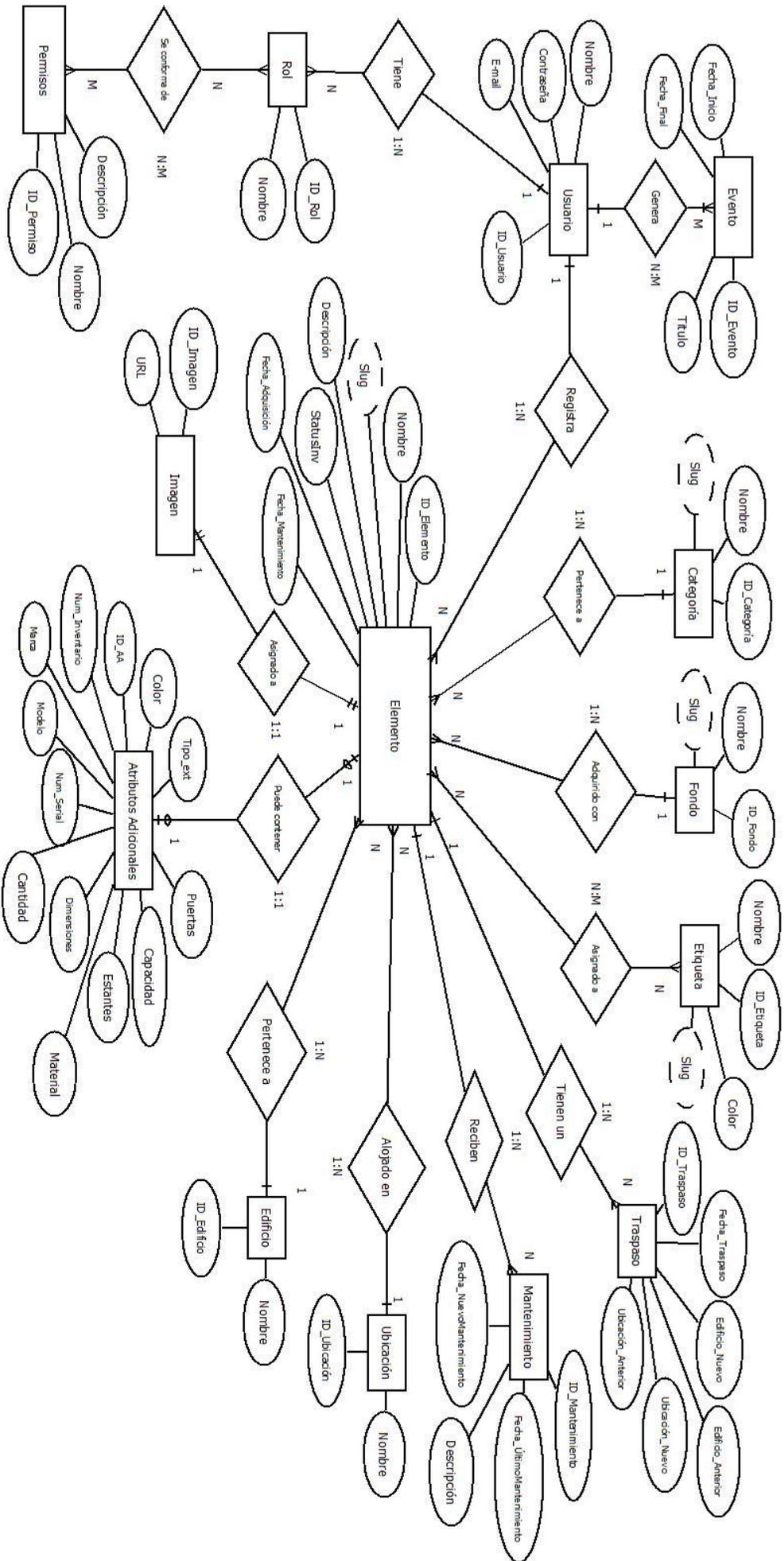


Figura 2. 1. Diagrama Entidad-Relación del Sistema de Inventario Airon.

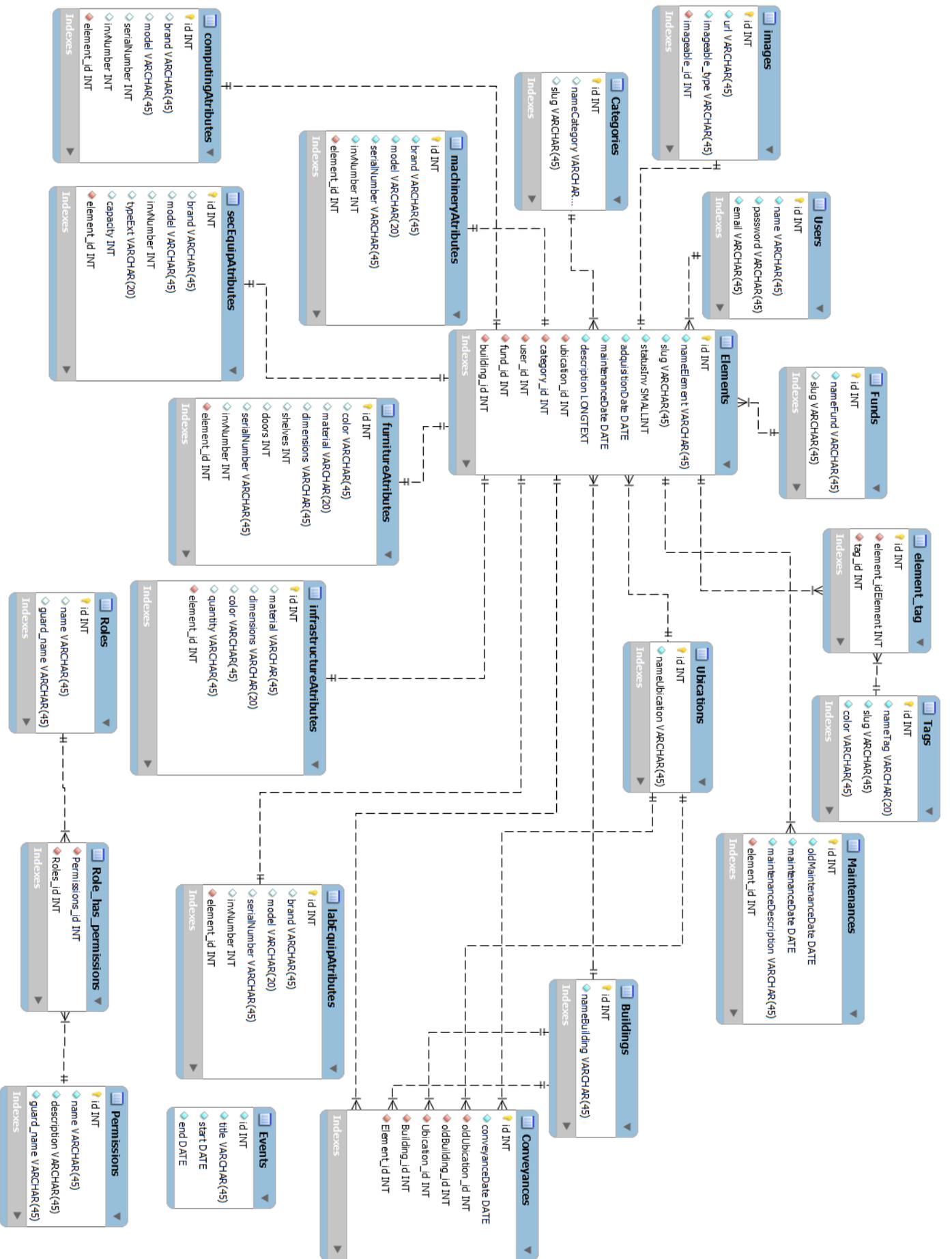


Figura 2. 2. Diagrama Relacional del Sistema de Inventario Airore.

El diagrama relacional será implementado en el sistema gestor de bases de datos relacionales (del acrónimo en inglés, *RDBMS*) *MySQL*, haciendo uso del motor de almacenamiento *InnoDB* integrado, este motor cuenta con una gran accesibilidad de uso, permitiendo realizar acciones como transacciones (*Commit – Rollback*), asegurando de esta manera el conjunto de características *ACID* en la base de datos del Sistema de Inventario *Airone*.

La manipulación de la base de datos será en conjunto al *ORM* de *Laravel Eloquent*, el *ORM* permite manipular las tablas de una base de datos relacional como si se tratase de un objeto dentro del contexto de una clase de un lenguaje orientado a objetos, esta técnica de manipulación de base de datos es compatible con *ACID* y todas las características de una base de datos relacional.

### **2.2.2 Sitemap**

Un *Sitemap* es un diagrama de referencia para representar disposición de las distintas vistas/páginas web que conforman al sistema, el uso de un *Sitemap* permitirá establecer una referencia para las rutas y vistas a implementar durante la codificación del sistema, así como su acceso e interacción entre sí. La Figura 2.3. contiene el *Sitemap* generado para el Sistema de Inventario *Airone*.

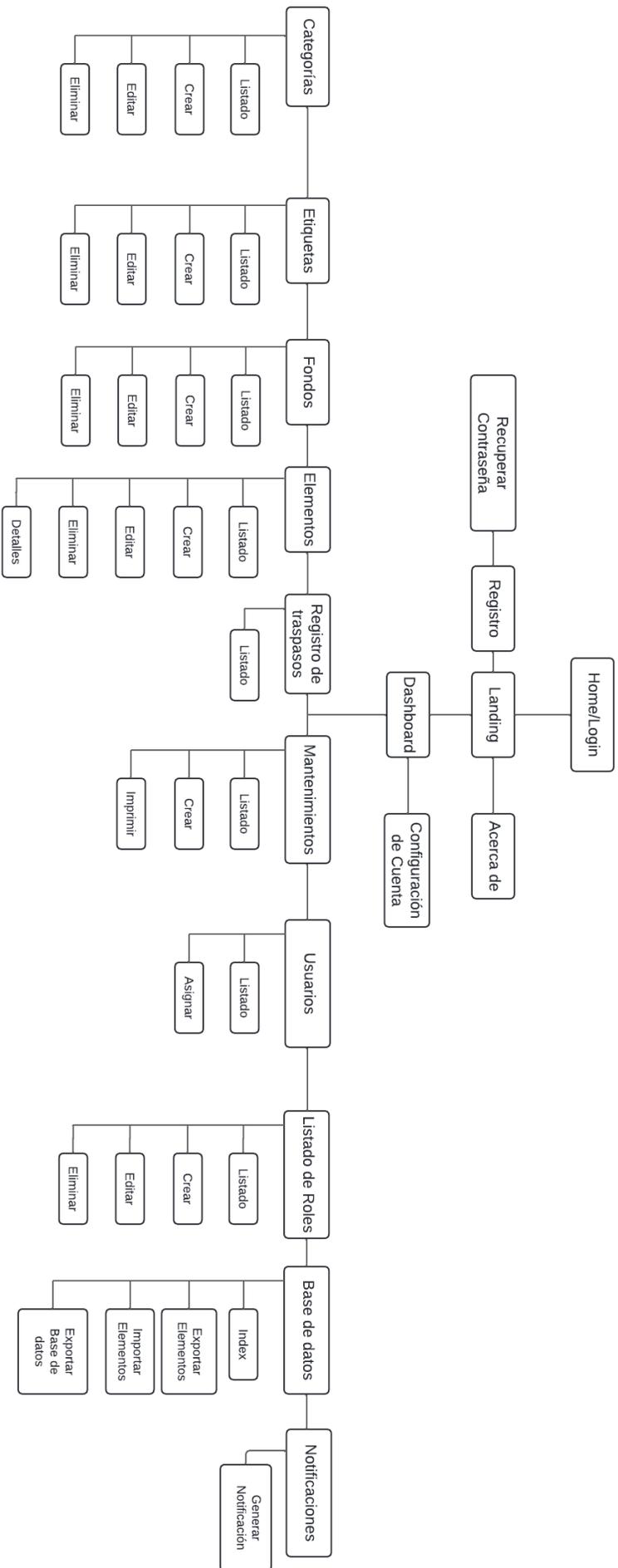


Figura 2. 3. Siemap del Sistema de Inventario Airone.

### 2.2.3 Backend

El desarrollo del *Backend* del sistema será modelado de acuerdo a las convenciones de *Laravel*, este *Framework* enfatiza el uso del patrón de diseño *MVC*, contando con archivos de configuración y rutas dedicadas para cada uno de los componentes del *Framework*, se hará uso de modelos para la manipulación de la base de datos utilizando el *ORM Eloquent*, controladores para las dividir y aislar las principales funcionalidades del sistema, y, finalmente, vistas, las cuales conformarán toda la parte que el usuario visualizará del sistema, es decir, el *Frontend*.

El acceso hacia las vistas y funciones embebidas en los controladores del sistema serán administrados por *Laravel*, mediante uno o más archivos de *Routing*, estos archivos determinarán el conjunto de rutas accesibles del sistema, la generación de *URL's* amigables es vital para la navegación del sistema, esto implica el uso de un sistema de *Slugs*, estos son generados durante la creación y edición de los distintos registros del sistema, la extensión de *jQuery stringToSlug* asegura la creación de estos *Slugs*.

El *Login* del sistema fue implementado haciendo uso del paquete *Laravel Jetstream*, este paquete proporciona de un *Login* funcional al sistema en conjunto de características como Recuperación de contraseña, Autenticación de dos factores y control de sesiones, la integración de *Laravel Jetstream* al sistema será conjunto a la de un *Landing*, esta será la vista que conectará al *Dashboard* del sistema, Configuración de cuenta y cierre de sesión entre sí. La Figura 2.5 muestra un diagrama de casos de uso del proceso de *Login* del sistema, en conjunto a las acciones que pueden realizarse en el *Landing* del sistema.

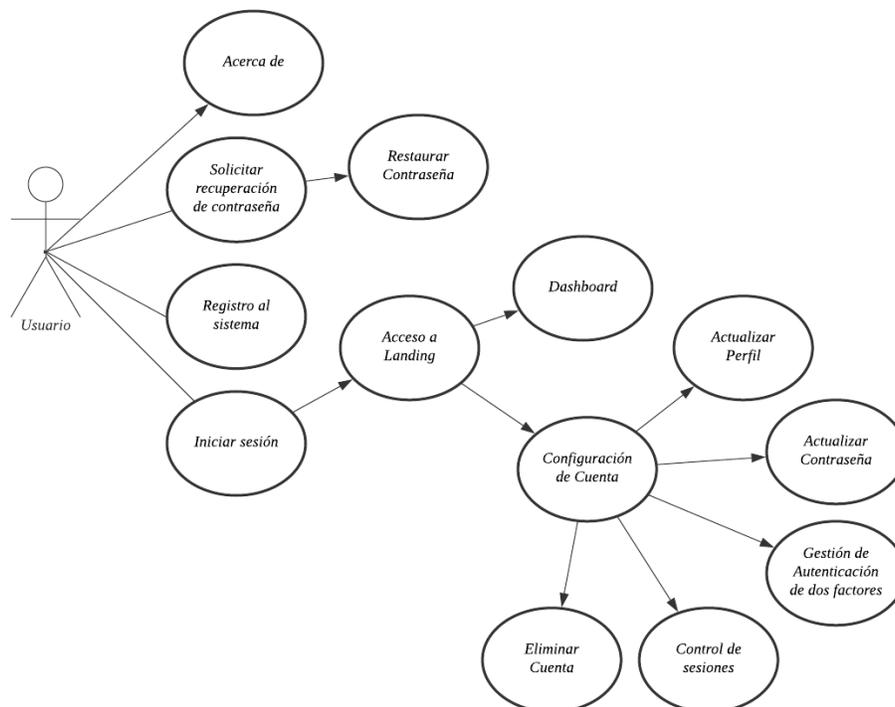


Figura 2. 5. Diagrama de casos de uso del proceso de *Login* y *Landing* del sistema.

Para el despliegue y manipulación del calendario de eventos se hará uso de la librería *FullCalendar*, el uso conjunto de esta librería, *jQuery* y *Toastr* otorgará de un comportamiento dinámico al calendario, gracias al uso de *AJAX*, los eventos creados por el usuario serán enviados al controlador en formato de una petición *POST* con datos en formato *JSON*, el resultado del proceso de estas transacciones se reflejará en las notificaciones generadas por *Toastr*.

El envío de notificaciones del sistema será implementado a través del uso de la librería *Laravel Telegram*, la librería proporciona las herramientas necesarias para que el usuario pueda difundir un mensaje a el canal de notificaciones dedicado para el Sistema de Inventario *Airone* en *Telegram* de manera sencilla. El calendario del Sistema de Inventario *Airone* anunciará cualquier evento en el calendario de forma automática al canal de notificaciones de *Telegram*, gracias a la integración de *CRON Tasks*, esta herramienta de programación permite la generación y aplicación de tareas programadas, de acuerdo a una fecha específica.

Los eventos generados en el calendario serán anunciados a una hora preestablecida (11:00 AM) en la fecha acordada del evento, una vez un evento ha sido anunciado este será eliminado de manera automática. La siguiente figura muestra el proceso de uso del calendario del sistema, además de la difusión de mensajes a través del canal de notificaciones de *Telegram*.

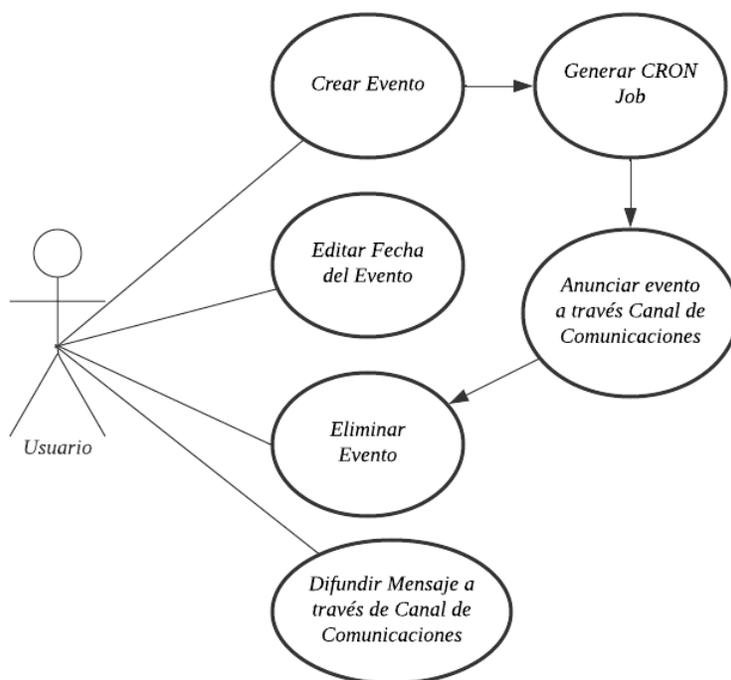


Figura 2. 6. Diagrama de casos de uso del Calendario del sistema y Difusión de mensajes.

La gestión de registros de atributos como Categorías, Etiquetas y Fondos disponen de la misma lógica para su administración, el usuario podrá crear un registro de cualquiera de estos atributos y utilizarlo en la creación de registros de elementos, además de reflejar estos nuevos registros y sus cambios en otras vistas donde se haga uso de estos datos. A continuación, se muestra el proceso de administración de estos registros por parte del usuario.

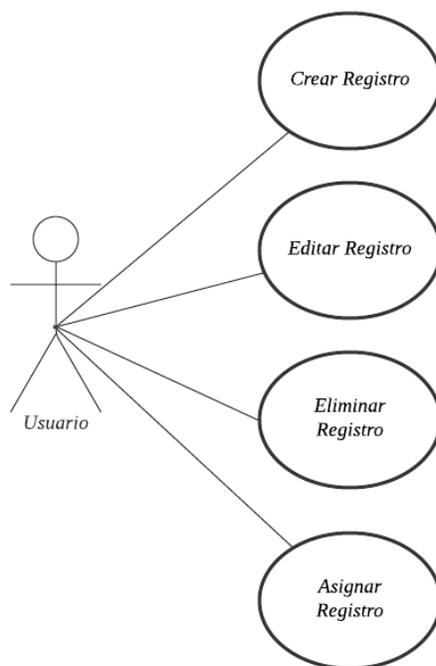


Figura 2. 7. Diagrama de casos de uso de la generación de registros para la administración del sistema de Inventario Airone (Categorías, Etiquetas y Fondos).

La administración de elementos la creación y manipulación de este tipo de registros será realizado por medio de un formulario, el control, validación y tratamiento de los datos introducidos por el usuario se apoyan en gran medida de las herramientas de validación de datos integradas de *Laravel*, además de la librería *Carbon*, con el propósito de administrar los atributos relacionados al tiempo, el uso de relaciones polimórficas es vital para correcto tratamiento y asignación de imágenes a los registros.

La generación de registros de trasпасos del sistema ocurrirá en el instante en el que un usuario actualice el registro de algún elemento, particularmente atributos como el Edificio o Ubicación, la fecha de cada registro será obtenida mediante el uso de la librería *Carbon*. El proceso de administración de los elementos se muestra en el siguiente diagrama.

El proceso de generación y exportación de reportes de mantenimiento del sistema será generado con el uso de la librería *DOMPDF*, cada reporte de mantenimiento se limita a describir únicamente un elemento registrado en el sistema, el usuario podrá consultar un listado con todos los reportes de mantenimiento generados. La siguiente figura muestra el proceso de administración de elementos, generación de registros de traspaso y las actividades de creación y exportación de reportes de mantenimiento.

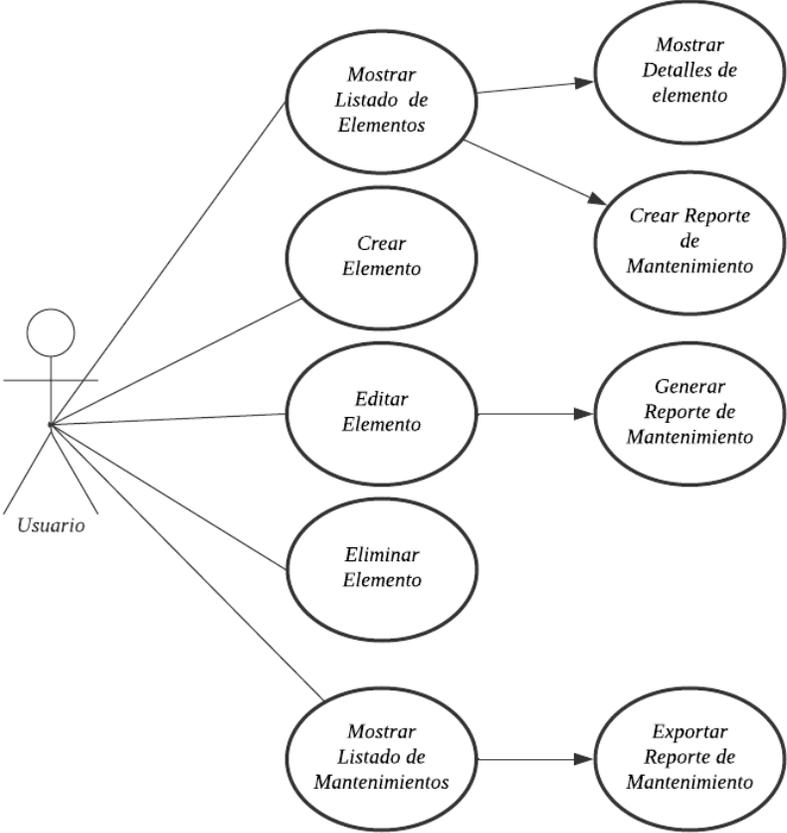


Figura 2. 8. Diagrama de casos de uso del proceso de administración de elementos de inventario, traspasos y mantenimientos.

La asignación de roles y permisos de usuario permitirá al usuario diseñar y generar sus propios roles, de acuerdo a un conjunto de permisos diseñados tomando de referencia las distintas funcionalidades que poseerá el apartado del *Dashboard* del sistema, la librería *Spatie Laravel Permission* proporciona las herramientas necesarias para administrar los roles y permisos del sistema de inventario *Airone*. La Tabla 2.1 muestra los permisos que pueden asignarse a un rol y la funcionalidad del sistema a la que corresponden.

Función	Permisos
<i>Dashboard</i>	Acceder al <i>Dashboard</i>
Administración de Elementos	Acceder a Listado de Elementos Crear Elementos Editar Elementos Eliminar Elementos
Administración de Categorías	Acceder a Listado de Categorías Crear Categorías Editar Categorías Eliminar Categorías
Administración de Etiquetas	Acceder a Listado de Etiquetas Crear Etiquetas Editar Etiquetas Eliminar Etiquetas
Administración de Fondos	Acceder a Listado de Fondos Crear Fondos Editar Fondos Eliminar Fondos
Administración de Roles	Acceder a Listado de Roles Crear Rol Editar Permisos de Rol Eliminar Rol
Administración de Usuarios	Asignar a un Rol Ver listado de Usuarios
Trasposos	Ver listado de Trasposos
Importación y Exportación	Acceder a Listado de Roles Crear Rol Editar Permisos de Rol Eliminar Rol
Calendario	Manipulación del calendario de eventos
Notificaciones	Envío de notificaciones a canal de <i>Telegram</i>

Tabla 2. 1. Permisos y Funciones del Sistema de Inventario Airone

La funcionalidad de exportación e importación de datos del sistema hará uso de la librería *Laravel Excel*, la importación de datos consiste de la subida de un archivo XLSX con un formato determinado, este archivo será puesto a diversas validaciones, con el fin de verificar el formato e integridad del archivo enviado, así como la validez de su contenido.

El Sistema de Inventario *Airone* cuenta con la capacidad de exportar todos los registros de elementos añadidos al sistema en conjunto a sus atributos adicionales en un archivo en formato XLSX, además de esta exportación, el sistema permitirá realizar un respaldo de la base de datos del sistema, mediante la exportación de un archivo *SQL* con todo lo necesario para restaurar la base de datos del sistema. La Figura 2.9 muestra el proceso de importación de registros de inventario por medio de un diagrama de flujo.

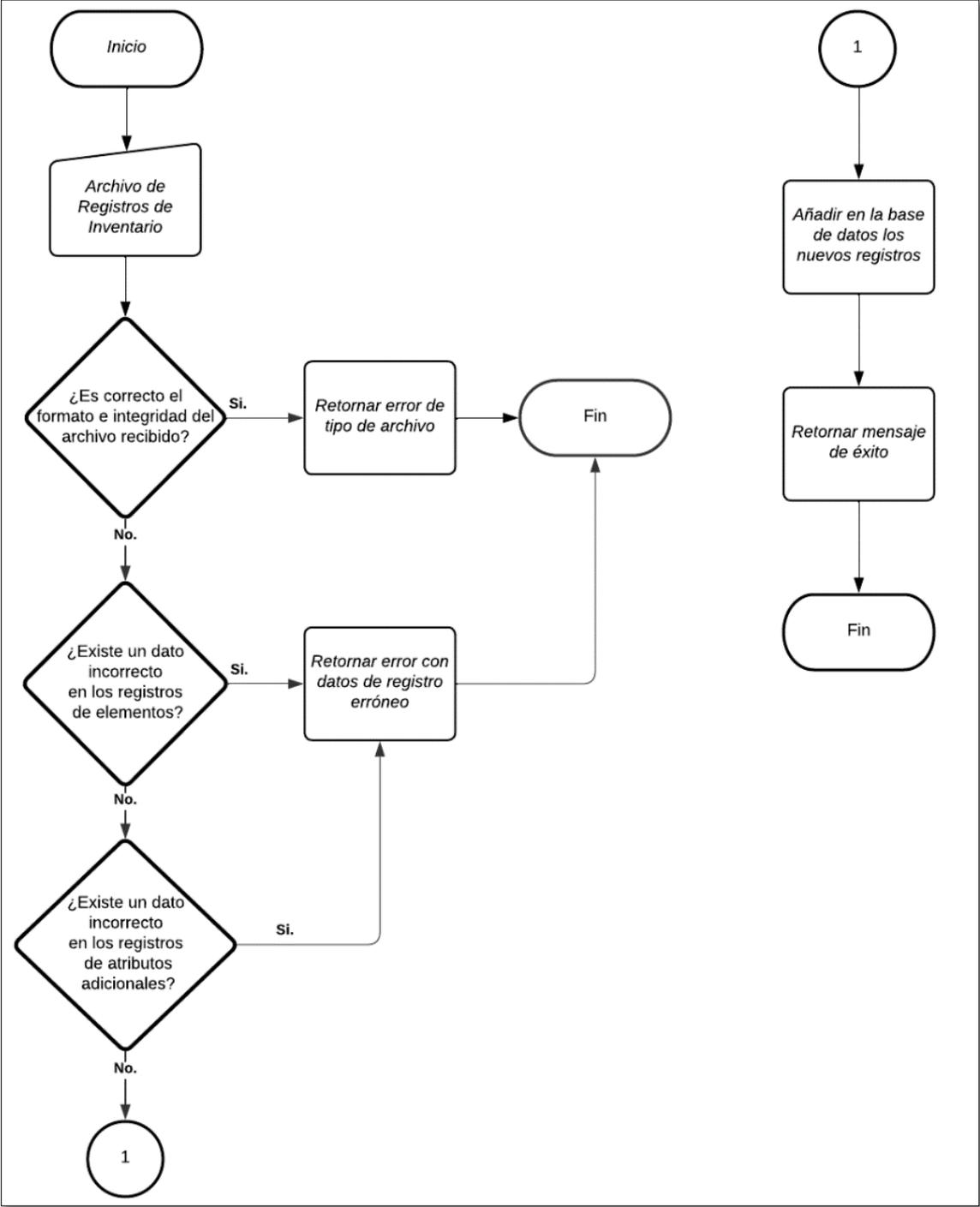


Figura 2. 9. Diagrama DFD del proceso de importación de registros de inventario por medio de un archivo XLSX del Sistema de Inventario Airone.

## 2.2.4 Frontend

El *Frontend* del sistema será diseñado con el uso de diversas herramientas integradas de *Laravel*, el sistema de plantillas *Blade* otorga de diversos componentes, los cuales permitirán generar y reutilizar elementos de las vistas del sistema, los componentes pueden basar su comportamiento de acuerdo a un conjunto de parámetros preestablecidos, otorgando de un grado de dinamismo a las vistas.

*Frameworks* de *Frontend* como *Bootstrap* y *TailwindCSS* conforman la estética del sistema de inventario *Airone*, proporcionando de una interfaz accesible e intuitiva, su uso conjunto permite el diseño de vistas del sistema más enriquecidas de elementos visuales.

El *Dashboard* integrado al sistema será generado haciendo uso de la librería *Laravel-AdminLTE*, la librería proporciona de un *Dashboard* altamente personalizable e intuitivo, las vistas que proporcionan de las funcionalidades clave del sistema, excluyendo todas aquellas relacionadas al *Login* y *Landing*, serán alojadas en el *Dashboard*.

Los formularios que utilizará el sistema en conjunto a otros componentes *HTML* derivados de esta, como títulos, atributos, identificadores de *JavaScript*, clases de *CSS* y asignación de métodos *HTTP* como *POST*, *PUT* y *DELETE*, serán implementados haciendo uso de la librería *Spatie Laravel-HTML*.

Los prototipos de vistas del sistema fueron generados tomando de referencia un prototipo funcional del sistema de inventario *Airone*, este prototipo integra las herramientas, paquetes y librerías más cruciales en el desarrollo del sistema final, proporcionando de las principales vistas y componentes de *Frontend* con las que contará el sistema de inventario *Airone*. Las Figuras 2.10. a 2.25. muestran las distintas vistas diseñadas para este prototipo.

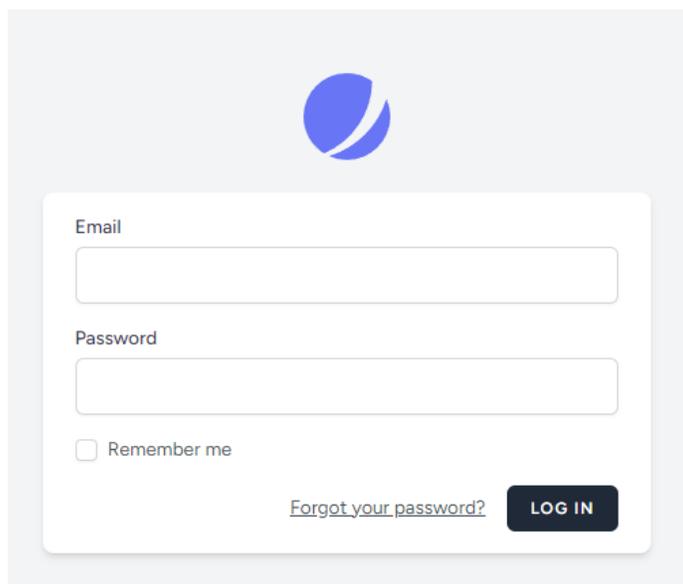
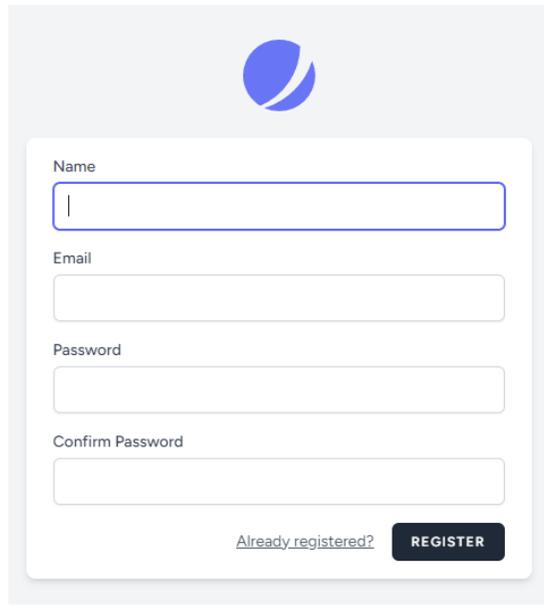
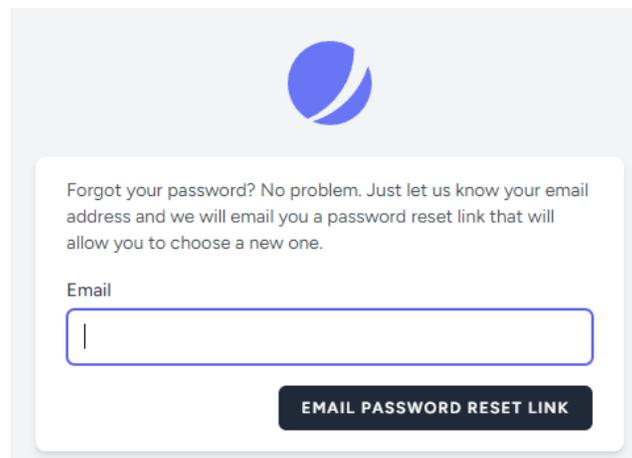


Figura 2. 10. Prototipo de vista de Login.



The image shows a registration form prototype. At the top center is a blue circular logo with a white swoosh. Below the logo is a white rounded rectangle containing the form fields. The fields are labeled 'Name', 'Email', 'Password', and 'Confirm Password'. Each label is followed by a white input box with a blue border. At the bottom right of the form is a dark blue button with the text 'REGISTER' in white. To the left of the button is a link that says 'Already registered?'.

*Figura 2. 11. Prototipo de vista de registro.*



The image shows a password recovery form prototype. At the top center is a blue circular logo with a white swoosh. Below the logo is a white rounded rectangle containing the form. The text reads: 'Forgot your password? No problem. Just let us know your email address and we will email you a password reset link that will allow you to choose a new one.' Below this text is a label 'Email' followed by a white input box with a blue border. At the bottom right of the form is a dark blue button with the text 'EMAIL PASSWORD RESET LINK' in white.

*Figura 2. 12. Prototipo de vista de recuperación de contraseña.*

Figura 2. 13. Prototipo de vista de configuración de cuenta.

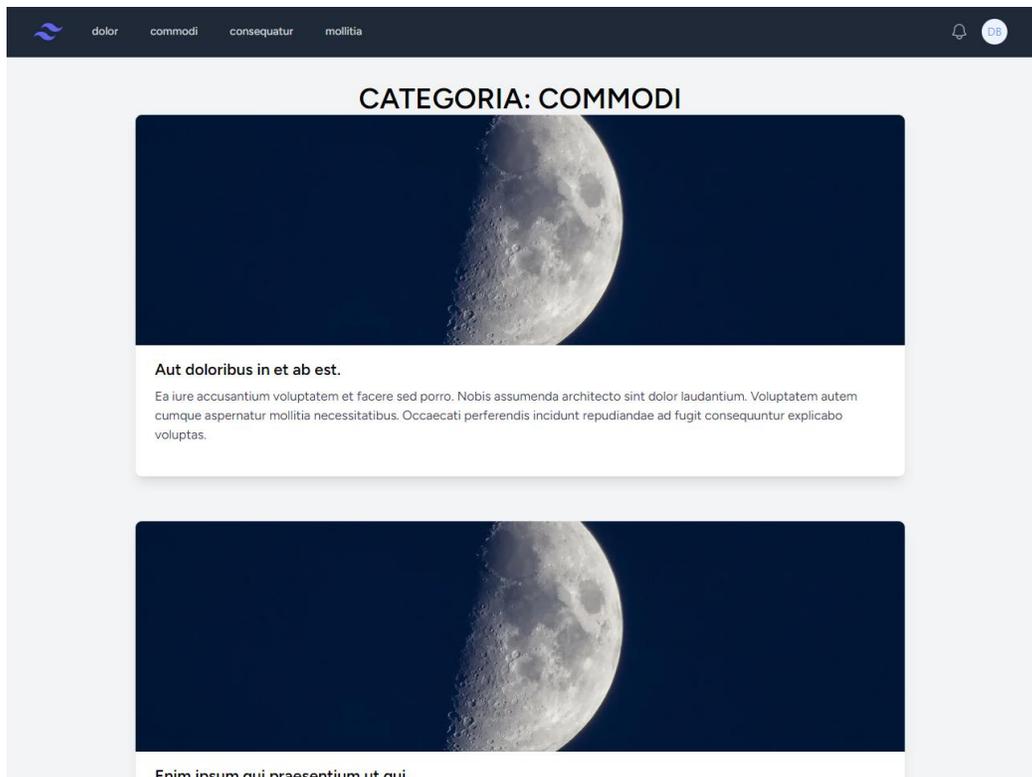


Figura 2. 14. Prototipo de vista de Landing.

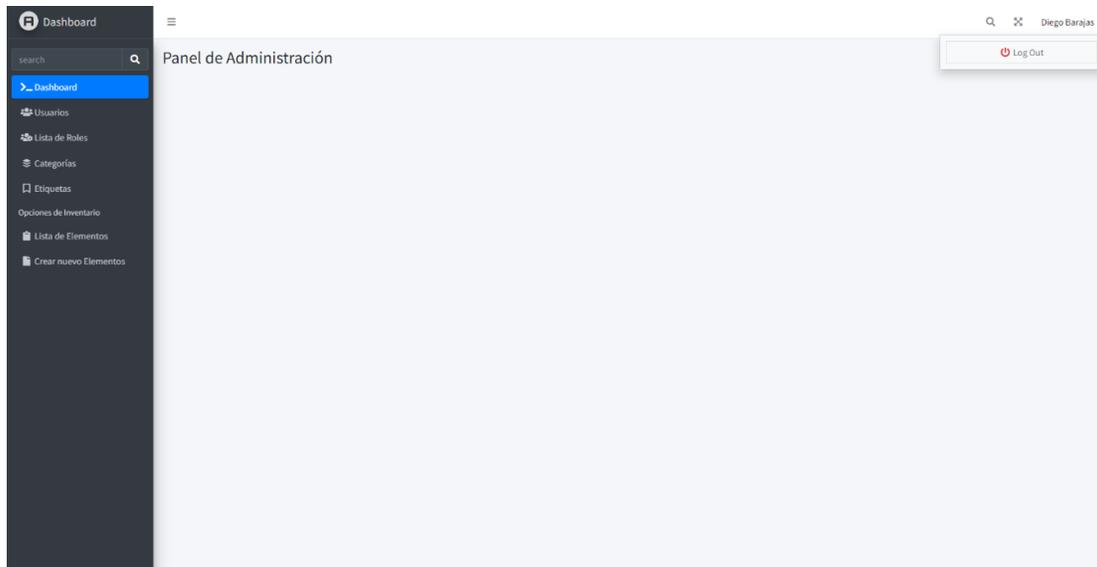


Figura 2. 15. Prototipo de vista de Dashboard.

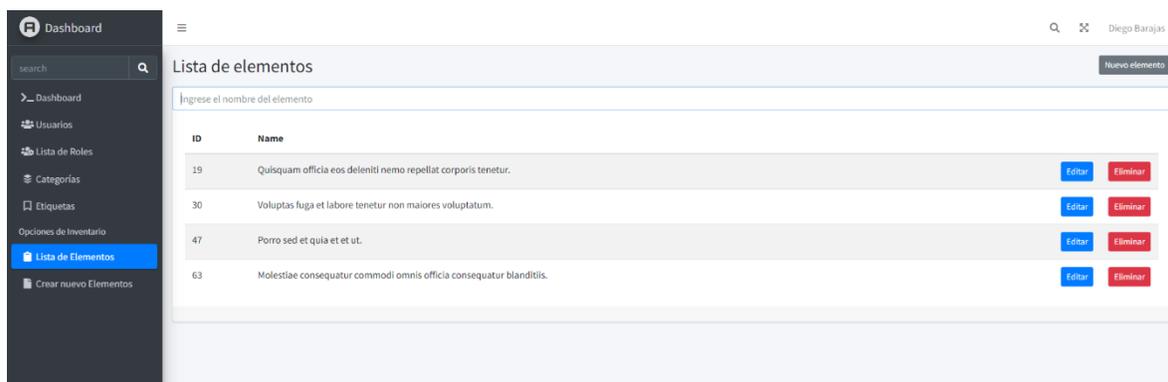


Figura 2. 16. Prototipo de vista de listado de elementos.

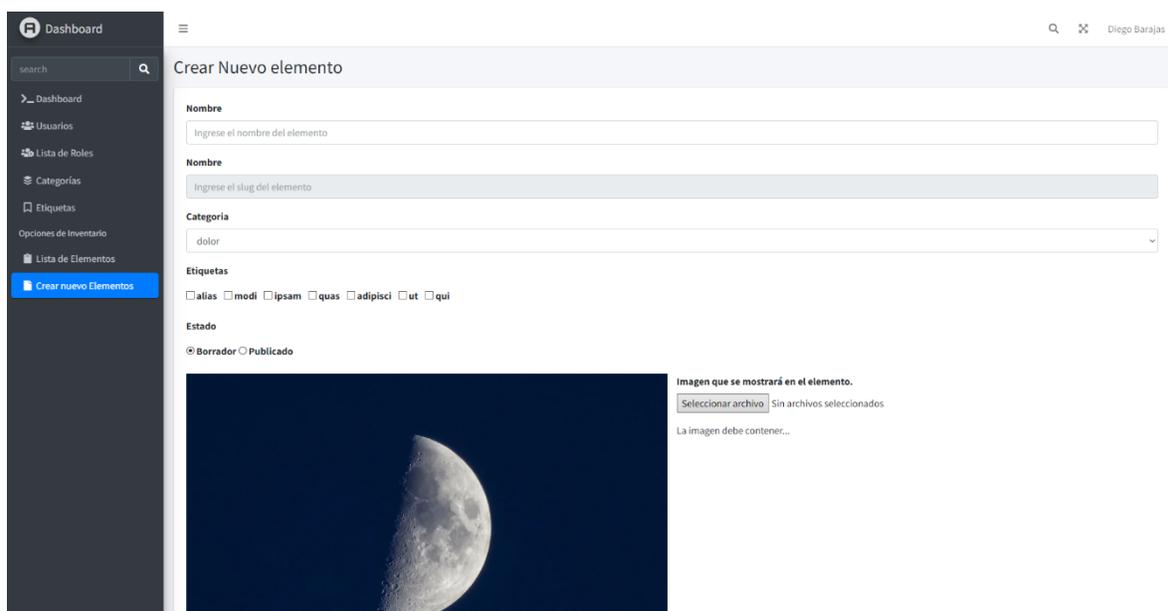


Figura 2. 17. Prototipo de vista de creación y edición de registro de elemento.

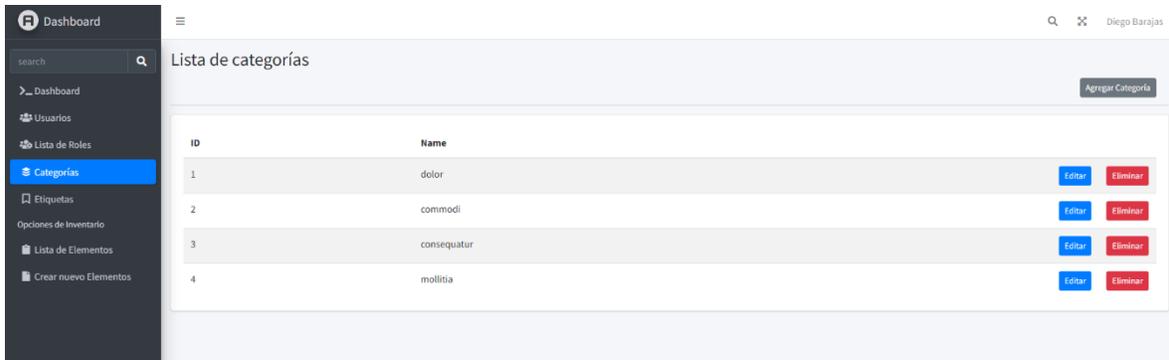


Figura 2. 18. Prototipo de vista de listado de atributos para la administración del sistema.

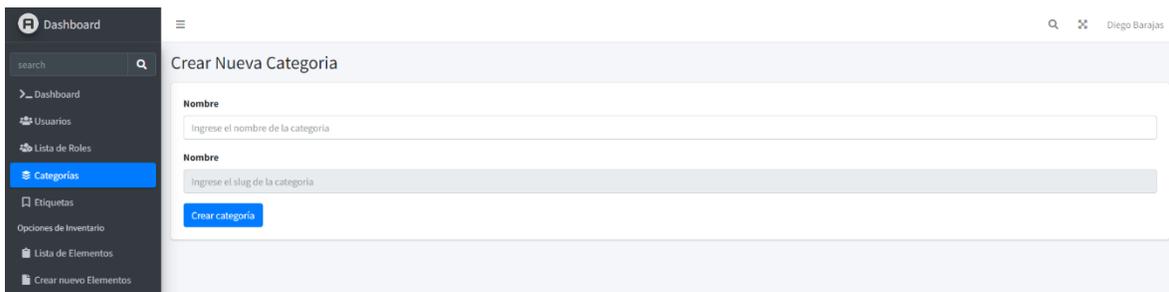


Figura 2. 19. Prototipo de vista de creación de nuevo registro de atributo para la administración del sistema.

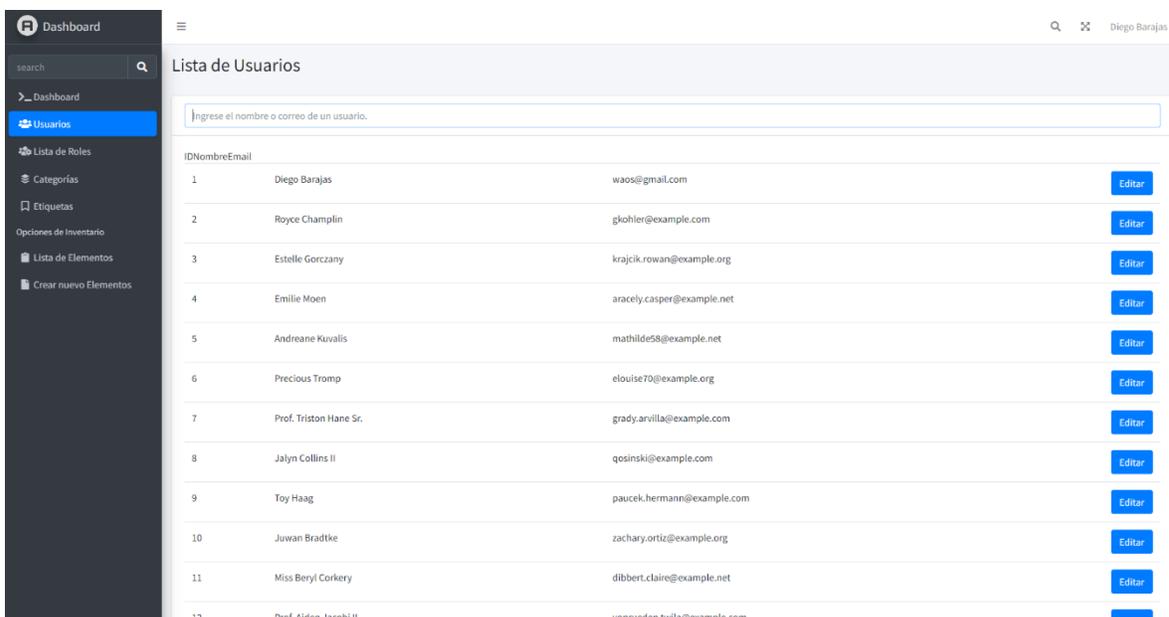


Figura 2. 20. Prototipo de vista de listado de usuarios.

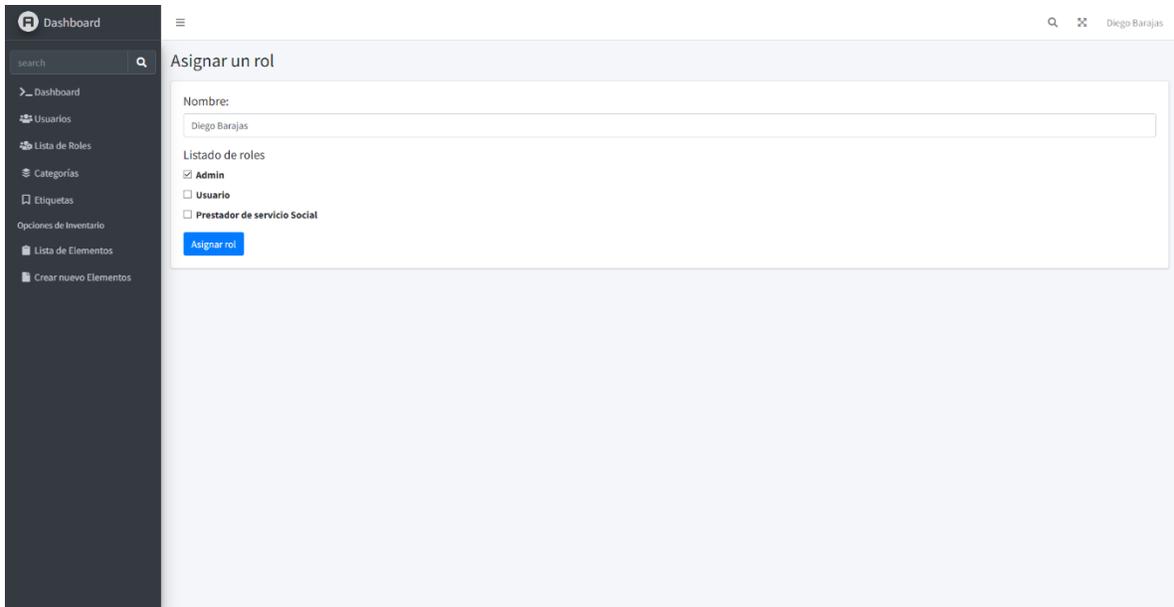


Figura 2. 21. Prototipo de vista de asignación de roles a usuario.

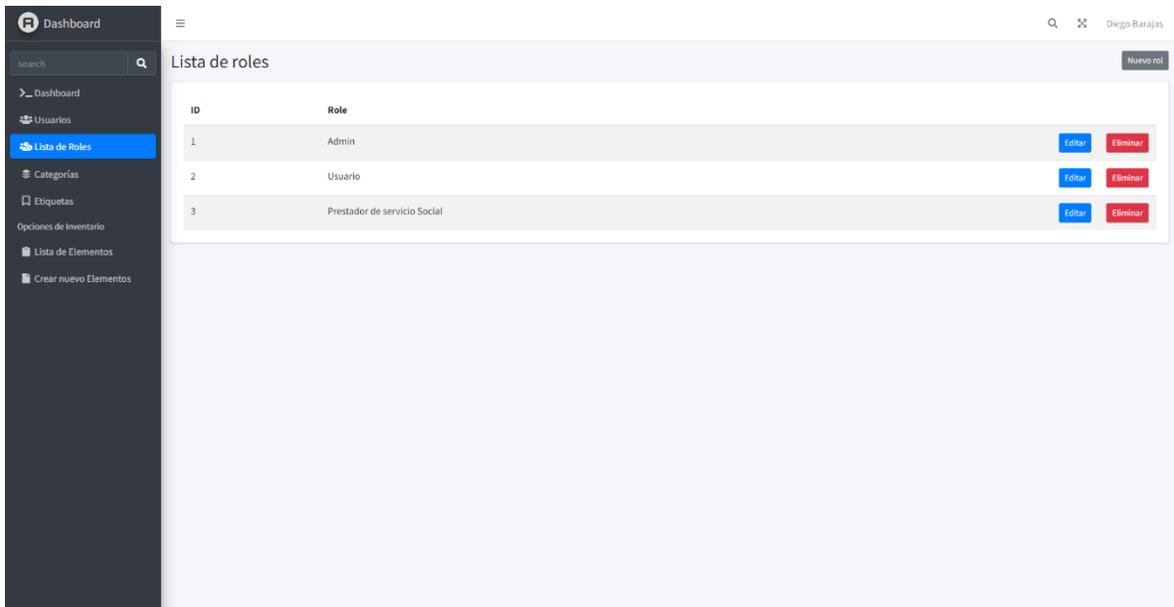


Figura 2. 22. Prototipo de vista de listado de roles.

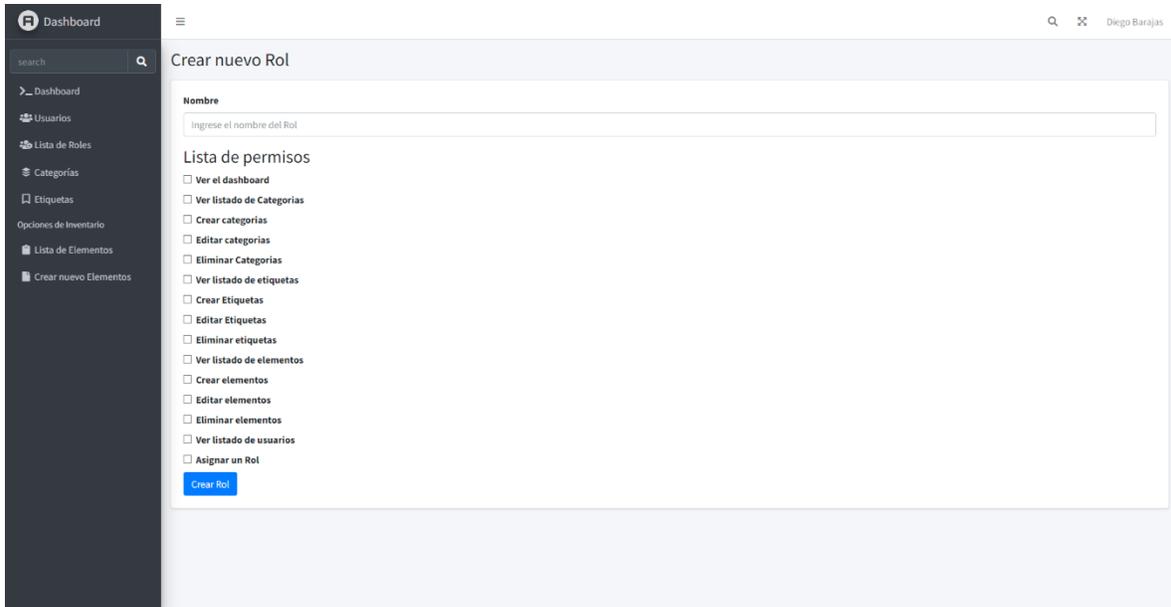


Figura 2. 23. Prototipo de vista de creación y edición de rol.

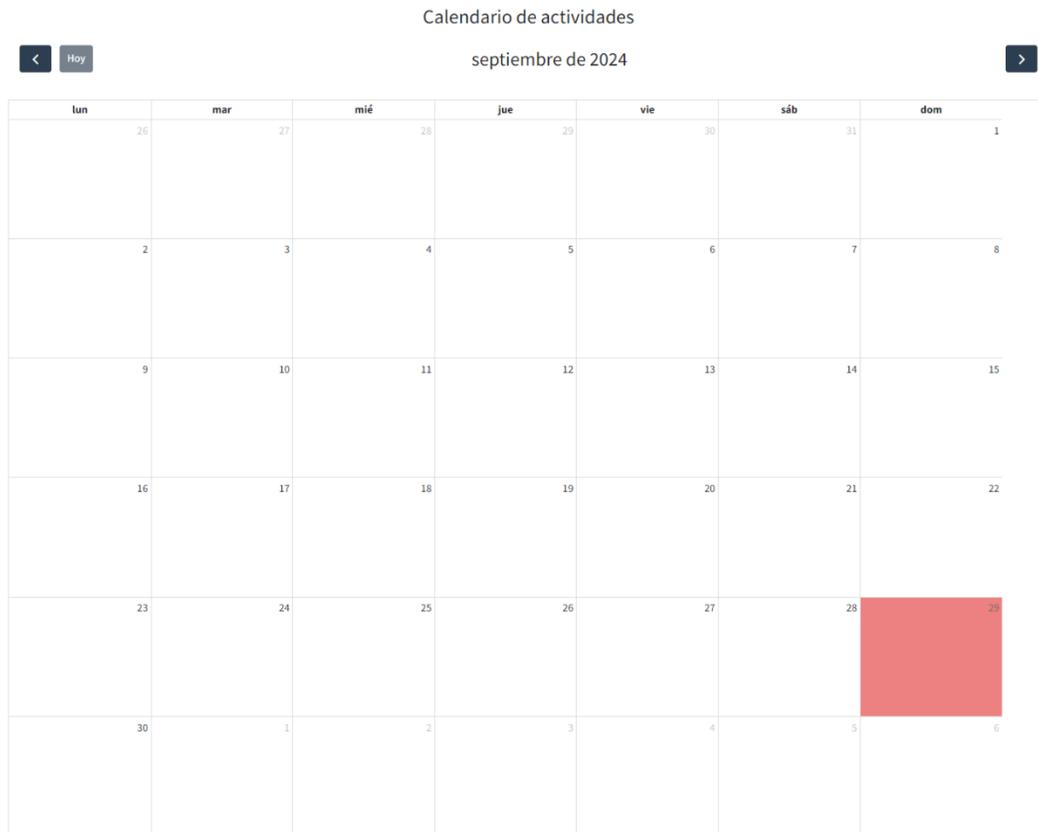


Figura 2. 24. Prototipo de calendario de actividades.

## 2.3 Codificación

---

La definición, diseño y estructura del sistema obtenida durante la fase de diseño ha establecido las pautas necesarias para iniciar el proceso de codificación, durante este apartado se describirá la codificación de las funcionalidades más importantes del sistema, siguiendo el orden de las iteraciones definidas, también se describirán las funcionalidades más destacadas de los distintos códigos implementados.

### 2.3.1 Iteración 1 - Creación del proyecto y desarrollo de componentes base del sistema

Durante la primera iteración se realizó una implementación conjunta de tres *CRUD*, cada uno correspondiendo a la administración de etiquetas, categorías y fondos dentro del Sistema de Inventario *Airone*.

Las operaciones de Creación, Actualización y Eliminación para cada una de las tablas siguen un patrón en común respecto a su lógica y funcionamiento, por lo tanto, en las Figuras 2.25 a 2.27 se mostrará el código utilizado para la administración de categorías en el Sistema de Inventario *Airone*.

```
public function store(Request $request)
{
    //Esta función se encarga de almacenar un nuevo registro para la tabla categories

    //En este apartado la variable $request (La cuál contiene los datos de la petición/categoría capturada por el Form
    $request->validate([
        'nameCategory' => 'required|string|max:50', //En este campo se dice que el apartado nameCategory debe ser oblig
        'slug' => 'required|string|unique:categories' //En este campo se dice que el slug debe de ser requerido y su n
    ]);

    /*Una vez la petición ha pasado por todas las validaciones se creará una variable denominada $category la cual con
    Además, en el apartado Category::create($request->all()); se añade la tupla de la petición a la base de datos. */
    $category = Category::create($request->all());

    //Finalmente una vez finalizado este procedimiento se redirecciona a la vista edit del CRUD de categorías (resource
    return redirect()->route('admin.categories.edit', $category)->with('info', 'La categoría se ha creado con éxito');
    //El método with() se encarga de retornar un token de retorno el cuál en nuestro código de página se verificará y
}
}
```

Figura 2. 25. Función para la creación de registro de categorías.

- Función *validate()*: Esta función se encarga de comprobar y validar los datos generados por el usuario almacenados en la variable *Request*, la validación implica comprobar aspectos como la longitud, tipo, entre otras reglas.
- Función *redirect()*: Una vez se haya ejecutado todo el código que corresponde a la acción del usuario entrada del usuario se redireccionará a una vista definida, este retorno puede contar con una variable y mensaje de notificación al usuario, por ejemplo, uno de éxito.

```

public function update(Request $request, Category $category)
{
    // return $request;
    //Esta función tiene la función de borrar una tupla/categoría de la base de datos.
    $request->validate([
        'nameCategory' => 'required|string|max:50',
        'slug' => "required|string|unique:categories,slug,$category->id"
        //En este campo se dice que slug debe de ser requerido y también sus campos únicos, pero únicamente en la tabla categoría
        //Además con el ,slug,$category->id y el uso de comillas dobles ignorará el slug de la categoría que vamos a actualizar,
    ]);

    //Condicional con el propósito de evaluar si la categoría a editar no es una de las categorías principales del sistema.
    if($category->id <= 6){
        return redirect()->route('admin.categories.index', $category)->with('error', 'Acción Prohibida.');
```

Figura 2. 26. Función para la actualización de un registro de categorías.

- Función *update()*: Tomando de base los datos de un modelo almacenado en la variable donde se invoca esta función, se actualiza un registro de la tabla haciendo uso de otra variable que almacena un modelo de la misma tabla de base de datos como parámetro de esta función.

```

public function destroy(Category $category)
{
    //Condicional con el propósito de evaluar si la categoría a eliminar no es una de las categorías principales del sistema.
    if($category->id <= 6){
        return redirect()->route('admin.categories.index')->with('error', 'Acción Prohibida.');
```

Figura 2. 27. Función para la eliminación de un registro de categorías.

- Función *delete()*: Se eliminará el registro al que hace la referencia la variable y modelo en la que se invoca esta función, la acción es permanente y afecta a la base de datos.

### 2.3.2 Iteración 2 - Implementación de sistema de roles y permisos, y atributos adicionales

Una vez que el sistema cuenta con las funcionalidades necesarias para la administración de los distintos parámetros de elementos, se realizó la implementación del *CRUD* de roles y elementos, el *CRUD* de elementos cuenta también con la lógica necesaria para administrar los atributos adicionales de los registros de elementos de acuerdo a las categorías correspondientes, además de las medidas necesarias para administrar elementos que fueron importados por medio de un archivo XLSX, las Figuras 2.28 a 2.35 muestran el código utilizado para la implementación de las funcionalidades mencionadas.

```
public function store(Request $request)
{
    $request->validate([
        'name' => 'required|string|max:50', //Validacion de campo name, este debe ser obligatorio
    ]);

    $role = Role::create($request->all()); //Creacion del rol en sí de acuerdo a los datos proporcionados
    $role->permissions()->sync($request->permissions); //Asignación de permisos al rol recién creado
    return redirect()->route('admin.roles.edit', $role)->with('info', 'El rol se creó con éxito.');
```

Figura 2. 28. Función para la creación de un rol.

- Función *sync()*: Sincroniza toda tabla intermedia del registro con una lista de identificadores o colección de instancias de modelos, con el fin de efectuar de manera exitosa cualquier cambio generado por una tabla que forma parte de una relación Muchos a Muchos, en este caso se le están otorgando un conjunto de permisos a un rol.

```
public function update(Request $request, Role $role)
{
    //Lógica para retornar a index si es un permiso no modificable.
    if ($role->id <= 2) {
        return redirect()->route('admin.roles.index', $role)->with('info2', 'Acción Prohibida.');
```

Figura 2. 29. Función para la actualización de un rol.

```

public function destroy(Role $role)
{
    //Lógica para retornar a index si es un permiso no modificable.
    if ($role->id <= 2) {
        return redirect()->route('admin.roles.index', $role)->with('info2', 'Acción Prohibida.');
```

Figura 2. 30. Función para la eliminación de un rol.

```

public function update(Request $request, User $user)
{
    $user->roles()->sync($request->roles); //llamadas a la tabla roles, se realizará una sincroniza
    return redirect()->route('admin.users.edit', $user)->with('info', 'Se asignaron los roles corr
}
```

Figura 2. 31. Función para la asignación de roles a un usuario.

```

$element->save();
//Una vez generado el id con la primer llamada del metodo save se sobres
//Este nuevo slug tiene el propósito de evitar que haya un conflicto por
$element->slug = ($element->slug)."-" . $element->id;
$element->save();

/* //// El proposito de este switch es de acuerdo al category_id de la p
un registro de atributos especificos, si no se pertenece alguno
la ejecución normalmente, sin crear una tupla/registro de atribu
switch ($request->category_id) {
    case "1":
        $computingAttribute = new computingAttribute();
        $computingAttribute->brand = $request->brand;
        $computingAttribute->model = $request->model;
        $computingAttribute->serialNumber = $request->serialNumber;
        $computingAttribute->invNumber = $request->invNumber;
        $computingAttribute->element_id = $element->id;
        $computingAttribute->save();
        break;
}
// Este if se encarga de verificar si se encuentra una imagen dentro del en
if ($request->file('file')){
    $url = Storage::put('public/elements', $request->file('file'));

    //Creación de registro con relación polimorfica a partir del atributo d
    $element->image()->create([
        'url' => $url
    ]);
    // Llamamos a element con la relacion polimórfica image y la creacion d
}

if($request->tags){ //Condional la cual verifica si estamos mandando info
    $element->tags()->attach($request->tags);
}

return redirect()->route('admin.elements.edit', $element)->with('info',
});
```

Figura 2. 32. Función para creación de registro de elemento.

```

if($element->slug == $request->slug){
    $element->save();
}
else{
    //En este caso donde los Slugs son diferentes se generará uno nuevo con
    $element->slug = $request->slug;
    $element->slug = ($element->slug)."-" . $element->id;
    $element->save();
}

if($request->file('file')){
    $url = Storage::put('public/elements', $request->file('file'));

    //Este if comprueba si ya existe una imagen creada en el element, si
    if($element->image){
        Storage::delete($element->image->url);

        $element->image->update([
            'url' => $url
        ]);
    }else{
        $element->image()->create([
            'url' => $url
        ]);
    }
}

if($request->tags){ //Condional la cual verifica si estamos mandando in
    $element->tags()->sync($request->tags); //Aqui con tags() recuperamos
    //Se usa el metodo sync para evitar etiquetas duplicadas, ademas de e
}

return redirect()->route('admin.elements.edit', $element)->with('info',

```

Figura 2. 33. Función para actualización de registro de elemento.

- Función *save()*: Esta función es utilizada si, durante la creación de un nuevo registro por medio del *ORM* de *Eloquent*, se desea modificar el valor de algún atributo de la instancia a guardar en la base de datos.
- Función *create()*: Genera una relación polimórfica 1 a 1 entre el registro del elemento y su imagen, la función *imagen* indica la existencia de una instancia de una imagen para la variable con los datos de la variable donde fue invocado.
- Función *attach()*: Añade las etiquetas a las que pertenece el elemento, es decir, aplica de una relación muchos a muchos para el registro de elementos recién generado, método alternativo para añadir registros a tablas derivadas de relaciones muchos a muchos.

```

switch ($request->category_id) {
    case "1":
        /* el if dentro de este case se encarga de corroborar si el category_id de la petición es el mismo que el del registro de la BD de la tabla de elementos, si no es así se ejecutará la función deleteAttribute */
        if ($element->category_id != $request->category_id){
            //En la función únicamente se realiza la tarea de eliminar el registro de la BD de la tabla de elementos
            deleteAttribute($element);
            //Una vez eliminado el registro se creará el nuevo registro de atributos específicos
            $computingAttribute = new computingAttribute();
            $computingAttribute->brand = $request->brand;
            $computingAttribute->model = $request->model;
            $computingAttribute->serialNumber = $request->serialNumber;
            $computingAttribute->invNumber = $request->invNumber;
            $computingAttribute->element_id = $element->id;
            $computingAttribute->save();

            //Finalmente estas 2 líneas del if se asegura de actualizar el registro de la tabla de elementos
            $element->category_id = $request->category_id;
            $element->save();
        }
        else{
            //Este if existe con el fin de comprobar que el elemento existe, en caso contrario se ejecutará la función deleteAttribute
            $computingAttribute = computingAttribute::where('element_id', $element->id)->first();
            // return $computingAttribute;
            if($computingAttribute == null){
                deleteAttribute($element);
            }
            $computingAttribute = new computingAttribute();
            $computingAttribute->brand = $request->brand;
            $computingAttribute->model = $request->model;
            $computingAttribute->serialNumber = $request->serialNumber;
            $computingAttribute->invNumber = $request->invNumber;
            $computingAttribute->element_id = $element->id;
            $computingAttribute->save();
        }
    }
}

```

Figura 2. 34. Funciones dedicadas para la actualización de registros de atributos adicionales y generación de registros de traspasos.

- Función *deleteAttribute()*: Función creada con el propósito de eliminar de manera rápida un registro de atributos adicionales relacionado correspondiente a un elemento, con el propósito de simplificar la lógica de actualización de registros de elementos de inventario, particularmente, el apartado de atributos adicionales.
- Método *Carbon::now()*: Invocación de la librería *Carbon*, recupera la fecha y hora en el instante que fue invocada esta función.

```

public function destroy(Element $element)
{
    //Invocación de nuestro metodo creado en ElementPolicy, va a recibir a la variable element
    $this->authorize('author', $element);

    //La invocación de este switch tiene como proposito eliminar la tabla de atributos extra del elemento en caso de que exista
    switch($element->category_id){
        case "1":
            /*De acuerdo a la category_id del elemento (NO el de la petición, sino el del registro recopilado en la BD de la tabla de elementos) se utilizará ese valor en un switch el cual de acuerdo a la categoría a la que pertenece eliminará el registro de la BD de la tabla de atributos extra del elemento */
            $deleted = computingAttribute::where('element_id', $element->id)->delete();
            break;
        case "2":
            $deleted = furnitureAttribute::where('element_id', $element->id)->delete();
            break;
        case "3":
            $deleted = infrastructureAttribute::where('element_id', $element->id)->delete();
            break;
        case "4":
            $deleted = labEquipAttribute::where('element_id', $element->id)->delete();
            break;
        case "5":
            $deleted = machineryAttribute::where('element_id', $element->id)->delete();
            break;
        case "6":
            $deleted = secEquipAttribute::where('element_id', $element->id)->delete();
            break;
        default:
            //En este caso el default existe dentro de este switch para cuando se cumple la condición de ejecución de la función deleteAttribute
            break;
    }

    $element->delete();
    return redirect()->route('admin.elements.index')->with('info', 'El elemento se ha eliminado con éxito');
}

```

Figura 2. 35. Función para la eliminación de un registro de elemento.

### 2.3.3 Iteración 3 – Generación automática de registros de traspasos, exportación y generación de registros de mantenimiento

Durante esta iteración se implementó dentro de la lógica de actualización de registros de elementos, la lógica necesaria para que, en el instante que un usuario actualice el edificio o ubicación de un registro de elemento, se genere un registro de traspaso de manera automática, además de esta integración se codificó las funciones necesarias para la generación y exportación de registros de mantenimiento. Las Figuras 2.36 a 2.38 muestran la codificación de la lógica para cada una de estas funcionalidades.

```
//El if se encarga de comprobar si se ha actualizado el id de edificio o ubicación, en caso de que sea diferente
if ($element->building_id != $request->building_id || $element->ubication_id != $request->ubication_id ) {
    $conveyance = new Conveyance();
    $conveyance->conveyanceDate = Carbon::now();
    $conveyance->oldBuilding_id = $element->building_id;
    $conveyance->oldUbication_id = $element->ubication_id;
    $conveyance->ubication_id = $request->ubication_id;
    $conveyance->building_id = $request->building_id;
    $conveyance->element_id = $element->id;
    $conveyance->save();
}
////
```

Figura 2. 36. Lógica para la creación de un registro de traspaso.

- Método `Carbon::now()`: Se invoca una instancia de la herramienta integrada de *Laravel*, *Carbon*, en conjunto a la función `now()`, la cual retorna la fecha y hora actual en el instante en el que fue invocado.

```
public function store(Request $request)
{
    //Reglas de validación para el registro de mantenimiento
    $request->validate([
        'maintenanceDescription' => 'required|string|max:2000',
        'maintenanceDate' => 'required|date|after_or_equal:oldMaintenanceDate', //la regla after_or_equal:oldMaintenanceDate
        'oldMaintenanceDate' => 'required|date', //before_or_equal
        'element_id' => 'required',
    ]);

    $element = Element::find($request->element_id);
    $element->maintenanceDate = $request->maintenanceDate;
    $element->save();

    //creación del nuevo registro de mantenimiento
    $maintenance = new Maintenance();
    $maintenance->maintenanceDescription = $request->maintenanceDescription;
    $maintenance->maintenanceDate = $request->maintenanceDate;
    $maintenance->oldMaintenanceDate = $request->oldMaintenanceDate;
    $maintenance->element_id = $request->element_id;
    $maintenance->save();

    // return $maintenance;

    //Retorno a vista index de mantenimientos con mensaje de éxito.
    return redirect()->route('admin.maintenances.index')->with('info', 'El registro se ha creado con éxito');
```

Figura 2. 37. Función para la creación de un registro de mantenimiento.

```

public function print(Maintenance $maintenance)
{
    //Recuperación de registro en tabla elementos con el ID
    $element = Element::find($maintenance->element_id);

    //Recopilación de datos de ubicaciones y edificios.
    $buildings = Building::all();
    $ubications = Ubication::all();

    //Switch case encargado de recopilar atributos adicionales en caso de existir.
    switch ($element->category_id){
        case "1":
            $atributes = computingAttribute::where('element_id', $element->id)->first();
            break;
        case "2":
            $atributes = furnitureAttribute::where('element_id', $element->id)->first();
            break;
        case "3":
            $atributes = infrastructureAttribute::where('element_id', $element->id)->first();
            break;
        case "4":
            $atributes = labEquipAttribute::where('element_id', $element->id)->first();
            break;
        case "5":
            $atributes = machineryAttribute::where('element_id', $element->id)->first();
            break;
        case "6":
            $atributes = secEquipAttribute::where('element_id', $element->id)->first();
            break;
        default:
            break;
    }

    setlocale(LC_TIME, 'es_ES');
    //Recopilación de la fecha actual y del mes (En letras y español, para el documento).
    $month = Carbon::now()->locale('es_ES')->monthName;

    $pdf = PDF::loadView('admin.maintenances.print', ['element'=>$element, 'maintenance'=>$maintenance,
    return $pdf->stream('Reporte_Elemento_'.$element->id.'.pdf');
    // return view('admin.maintenances.print', compact('element', 'maintenance'));
}

```

Figura 2. 38. Función para la exportación de un registro de mantenimiento a formato PDF.

- Método `PDF::loadView()`: Esta instancia invoca a la librería *DOMPDF*, particularmente para generar un documento en formato PDF, tomando de base una vista dada, la cual mostrará todos los datos, etiquetas y estilos contenidos dentro de la vista.

### 2.3.4 Iteración 4 – Exportación e Importación de Registros de elementos

Durante esta iteración se realizó la codificación de la lógica necesaria para la exportación e importación de registros en formato XLSX, se hizo un uso extensivo de la librería *Laravel Excel* para realizar ambas funcionalidades, también se incluye la lógica necesaria para exportar en formato *SQL* todos las tablas y registros de la base de datos. Las Figuras 2.39 a 2.41 Muestran las funciones utilizadas para realizar estas funcionalidades.

```
public function export(){
    //Generación de fecha actual
    $fecha = Carbon::now()->format('Y-m-d');
    //Generación del documento con el título que incluye la fecha actual.
    return Excel::download(new DatabaseExport, 'Listado_Elementos_'.$fecha.'.xlsx');
}
```

Figura 2. 39. Función para la exportación de registros de elementos de inventario por XLSX.

- Método *Excel::download()*: Se invoca a la librería *Laravel Excel*, el método *download* perteneciente a esta librería recibirá 2 parámetros, el primero es una instancia de *Exportación*, esta instancia contiene el formato y condiciones de los datos que serán exportados, el segundo parámetro es el nombre del archivo exportado.

```
public function import(Request $request){
    $request->validate([
        //Regla de validación para comprobar que le archivo subido es un .xlsx
        'import' => 'required|mimes:xlsx',
    ]);
    // return $request;
    $request->file('import')->storeAs('public/uploads', 'upload.xlsx');
    // Define el rango de IDs que se eliminarán en caso de que falle la transacción
    $startId = Element::latest()->first()->id;

    //Try Catch con el proposito de capturar errores durante el proceso de importación.
    try {
        Excel::import(new DatabaseImportStep1, storage_path('app/public/uploads/upload.xlsx'));
    } catch (\Maatwebsite\Excel\Validators\ValidationException $e) {
        $failures = $e->failures();
        foreach ($failures as $failure) {

            //Retorno del primer error detectado en la excepción.
            return redirect()->route('admin.database.upload', compact('failures'))->with('infoError', 'S');
        }
    }
    try {
        Excel::import(new DatabaseImportStep2, storage_path('app/public/uploads/upload.xlsx'));

        return redirect()->route('admin.database.upload')->with('info', 'Hoja de cálculo importada con');
    } catch (\Maatwebsite\Excel\Validators\ValidationException $e) {
        $failures = $e->failures();
        //Captura de todos los errores
        // return $failures; //Depuración errores completos

        //Si se detecta una falla al momento de importar registros de atributos se eliminarán todos los
        //Lógica para eliminar registros de elementos.
        foreach ($failures as $failure) {
            $endId = Element::latest()->first()->id;
            if($startId != $endId){
                // Elimina los registros dentro del rango especificado
                Element::whereBetween('id', [($startId + 1), $endId])->delete();
                return redirect()->route('admin.database.upload', compact('failures'))->with('infoError', 'S');
            }
        }
        return redirect()->route('admin.database.upload', compact('failures'))->with('infoError', 'S');
    }
}
```

Figura 2. 40. Función para la importación de registros de elementos de inventario por XLSX.

- Método *Excel::import*: Se invoca a la librería *Laravel Excel*, la función *import* perteneciente a esta librería recibirá 2 parámetros, una instancia de Importación, esta instancia contiene el formato, condiciones, validaciones y reglas necesarias para que los datos sean importados de manera consistente, el segundo parámetro es un archivo en formato XLSX proporcionado por el usuario, este archivo se encuentra almacenado en una ubicación predefinida dentro de la estructura del sistema

```

public function dump(){
    // Bucle a través de cada tabla seleccionada.
    foreach ($tables as $table) {
        $show_table_query = "SHOW CREATE TABLE " . $table;
        $statement = $connect->prepare($show_table_query);
        $statement->execute();
        $show_table_result = $statement->fetchAll();

        // Añadir la declaración CREATE TABLE al output.
        foreach ($show_table_result as $show_table_row) {
            $output .= "\n\n" . $show_table_row["Create Table"] . "\n\n";
        }

        // Evitar la exportación de registros para las tablas cache, sessions y cache_locks.
        if (!in_array($table, ['cache', 'cache_locks', 'sessions', 'failed_jobs'])) {
            $select_query = "SELECT * FROM " . $table;
            $statement = $connect->prepare($select_query);
            $statement->execute();
            $total_row = $statement->rowCount();

            // Bucle a través de cada registro de la tabla actual.
            for ($count = 0; $count < $total_row; $count++) {
                // Obtener un registro como un array asociativo.
                $single_result = $statement->fetch(\PDO::FETCH_ASSOC);
                // Obtener las claves (nombres de columnas) del registro.
                $table_column_array = array_keys($single_result);
                // Obtener los valores del registro y escapar caracteres especiales.
                $table_value_array = array_map(function($value) {
                    // Escapar barras invertidas y otros caracteres especiales
                    return addslashes($value);
                }, array_values($single_result));

                // Construir la consulta INSERT INTO para el registro actual.
                $output .= "\nINSERT INTO $table (";
                $output .= "" . implode(", ", $table_column_array) . ") VALUES (";
                $output .= "" . implode("'", $table_value_array) . "');\n";
            }
        }
    }

    // Nombre del archivo de respaldo.
    $file_name = 'Respaldo_BDD_' . date('y-m-d') . '.sql';
}

```

Figura 2. 41. Función para la Exportación de la base de datos en formato SQL.

- Función *execute()*: Ejecuta una sentencia de *SQL*, esta sentencia puede ser proporcionada por una variable o introducida directamente dentro de los parámetros de la función.
- Función *rowCount()*: Esta función retorna el número de coincidencias dada una sentencia *SQL*.
- Función *fetch()*: Retorna las coincidencias de una sentencia *SQL* en formato de diccionario.
- Función *array\_keys()*: Retorna las llaves de un arreglo de tipo diccionario dado.
- Función *array\_map()*: Mapeo Arreglo retorna un arreglo/diccionario que contiene el producto de la ejecución de una función sobre un arreglo/diccionario.

### 2.3.5 Iteración 5 - Notificaciones por medio de Telegram y Calendario Interactivo

La quinta iteración es la última en proporcionar de funcionalidades clave al sistema, particularmente, las notificaciones por medio de *Telegram* y el calendario interactivo, para la integración de ambas funcionalidades se recurrió al uso de *Laravel Telegram* y *FullCalendar*, particularmente una integración de ambos por parte del calendario interactivo, con el fin de proporcionar de notificaciones de eventos de manera automática, de acuerdo a la fecha en que ocurrirá este evento. Las Figuras 2.42 a 2.44 muestran el código utilizado para implementar estas funcionalidades.

```
public function send(Request $request){

    //Recupera de la petición el ID del Chat del grupo y el mensaje que mandará el BOT.
    $chatId = $request->chat_id; // ID del chat del grupo
    //Recuperación de variable con datos del usuario.
    $user = Auth::user();

    //Generación del mensaje final que se enviará al grupo.
    $message = "El usuario ".$user->name." ha enviado la siguiente notificación: ". $request->message;
    //Se crea una instancia de stdClass, que es una clase genérica en PHP. Los objetos de tipo stdClass son simples y no
    // Se crea el objeto genérico para actuar como el destinatario de la notificación
    $notifiable = new stdClass();
    // return $message;

    // Se define el método send para devolver el chat ID del grupo (El atributo send contiene esa función)
    $notifiable->send = function() use ($chatId) {
        | return $chatId;
    };

    // Envía la notificación usando el sistema de notificaciones de Laravel
    Notification::send($notifiable, new TelegramNotification($message, $chatId));
    //Redirección a la página de notificaciones con mensaje de éxito.
    return redirect()->route('admin.notifications.index')->with('info', 'La notificación fue enviada con éxito.');
```

Figura 2. 42. Función para el envío de notificaciones al canal de Telegram.

- Función *user()*: Esta función retorna el nombre del usuario que actualmente se encuentra en sesión y ejecutó esta acción.
- Función *send()*: Esta función embebida de *Laravel* para el envío de una notificación por medio de un canal externo al de la aplicación (comúnmente un *API*), utilizando una instancia de la Librería de *Laravel Telegram* para indicar que se enviará por este medio la notificación, haciendo uso de las credenciales y contenido correspondientes.

```

public function ajax(Request $request)
{
    $today = Carbon::today()->format('Y-m-d');
    $formattedStart = Carbon::parse($request->start)->format('Y-m-d');
    $formattedEnd = Carbon::parse($request->end)->format('Y-m-d');

    $request->validate([
        'type' => 'required|string|max:255|in:add,update,delete', //El valor
        'title' => 'required_if:type,add,update|string|max:255', //Esta es
        'start' => 'required_if:type,add,update|date|after:' . $today, //E
        'end' => 'required_if:type,add,update|date|after_or_equal:start',
    ]);

    switch ($request->type) {
        case 'add': //Si la petición contiene en type el valor "add" se
            $event = Event::create([
                'title' => $request->title,
                'start' => $request->start,
                'end' => $request->end,
            ]);
            // Se retornará una respuesta en formato JSON con una variab
            return response()->json($event);
            break;
        case 'update':
            //En el caso de que la solicitud AJAX tenga el valor en type
            $event = Event::findOrFail($request->id)->update([
                'title' => $request->title,
                'start' => $formattedStart,
                'end' => $formattedEnd,
            ]);

            return response()->json($event);
            break;
        case 'delete':
            //Se realiza la búsqueda del registro, una vez encontrado se e
            $event = Event::findOrFail($request->id)->delete();
            // $event = Event::find($request->id)->delete();
            return response()->json($event);
            break;
        default:
            //Y no pasa nada...
            break;
    }
}

```

Figura 2. 43. Función para la administración del calendario de eventos.

- Clase *Carbon*: Convierte una fecha dada a un *String*, la función se encarga de proporcionar de un formato de fecha específico al *String* recién generado, en este caso se adoptó de un formato ‘Año-Mes-Día’.
- Función `json()`: Retorna en formato *JSON* un conjunto de Atributos, esto serán utilizados por la librería *FullCalendar*, particularmente, en las transacciones *AJAX*.

```

$notifications = Event::where('start', '>=', Carbon::today()->toDateString()->get());

foreach($notifications as $notification)
{
    $formattedDate = Carbon::parse($notification->start);
    $notificationDay = $formattedDate->format('d');
    $notificationMonth = $formattedDate->format('m');
    Schedule::call(function () use ($notification) {
        $formattedDateStart = Carbon::parse($notification->start);
        $formattedDateEnd = Carbon::parse($notification->end);
        $daysDifference = $formattedDateStart->diffInDays($formattedDateEnd);
        $year = Carbon::now()->format('Y');
        $notificationYear = $formattedDateStart->format('Y');
        $notifiable = new \stdClass();
        if($year == $notificationYear){
            //Este ifi else tiene la función de en base al valor de la variable $daysDifference (Variable con los días de diferenc
            if($daysDifference <= 1){
                //Envío de la notificación al canal de telegram
                Notification::send($notifiable, new TelegramNotification("El evento ".$notification->title." transcurrirá el día " . $notificationDay . " de " . $notificationMonth . " de " . $notificationYear));
            }
            else{
                //Generación de variables que almacenan los valores de días y meses de inicio y fin del evento
                $startDay = $formattedDateStart->format('d');
                $startMonth = $formattedDateStart->format('m');
                $endDay = $formattedDateEnd->format('d');
                $endMonth = $formattedDateEnd->format('m');
                //Envío de la notificación al canal de telegram
                Notification::send($notifiable, new TelegramNotification("El evento ".$notification->title." transcurrirá del día " . $startDay . " de " . $startMonth . " de " . $notificationYear . " al día " . $endDay . " de " . $endMonth . " de " . $notificationYear));
            }

            //Una vez generada la notificación se eliminará el registro de notificación actual.
            $notification->delete();
        }
    });
    //La notificación será programada a la hora establecida (Hora asignada (11:00 AM)) con el día y mes asignado por el registro
    $notification->cron('00 11 ' . $notificationDay . ' ' . $notificationMonth . ' *');
}
}

```

Figura 2. 44. Lógica para la generación de CRON Tasks de notificaciones de eventos.

- Clase *Schedule::call*: La instancia de *Schedule* se encarga de realizar una acción, de acuerdo a un parámetro de hora definido a través de variables o datos estáticos, particularmente la función CRON asiste en determinar cuándo se realizará esta acción.

En este caso se definió, mediante el uso de una función anónima, la difusión de la notificación, de acuerdo a la duración del evento, además de su eliminación una vez realizado el evento, esta función, en conjunto de una estructura de iteración *ForEach*, se ha automatizado la generación de notificaciones.

## 2.4 Pruebas

Con el propósito de evaluar y comprobar el funcionamiento del Sistema de Inventario *Airone* se han generado un conjunto de pruebas unitarias y de aceptación en base a las historias de usuario recopiladas, cada una evaluará distintas funcionalidades del sistema, definiendo los comportamientos esperados de acuerdo a un conjunto de parámetros dados. Las tablas 2.2 a 2.8 describirán las pruebas creadas para la evaluación del sistema en conjunto los resultados obtenidos sus resultados, producto de estas pruebas.

<b>Prueba 1</b>
<b>Historia de Usuario:</b> 1.- Administración de Categorías, Fondos y Etiquetas
<b>Nombre:</b> Comprobación de funcionalidad de Administración de Categorías, Fondos y Etiquetas.
<b>Descripción:</b> Se comprobará que el sistema pueda hacer acciones de un CRUD para las tablas Categorías, Fondos y Etiquetas.
<b>Condiciones de Ejecución:</b> La entrada del usuario puede duplicarse.
<b>Entrada/Pasos de Ejecución:</b> El usuario realizará los siguientes pasos. <ol style="list-style-type: none"><li>1. Accede a la vista de Creación de Registro de administración (Categorías, Fondos o Etiquetas).</li><li>2. Crea un Registro Nuevo.</li><li>3. Guarda la información.</li><li>4. Edita el Registro con el mismo nombre.</li><li>5. Edita el Registro con un nuevo nombre.</li><li>6. Elimina el Registro Nuevo.</li></ol>
<b>Resultado Esperado:</b> Se almacenará de manera correcta el registro de administración, El editar el elemento con el mismo nombre retornará un mensaje de error, notificando la redundancia del contenido, después se actualizará de manera correcta.
<b>Evaluación de la Prueba:</b> El sistema crea con éxito cada registro de administración, la actualización del elemento con el mismo nombre divulga un mensaje con esta situación al usuario, la actualización con un nombre nuevo actualizó el registro con éxito, la eliminación del registro de administración se realizó con éxito.

Tabla 2. 2. Prueba 1. Comprobación de funcionalidad de Administración de Categorías, Fondos y Etiquetas.

<b>Prueba 2</b>
<b>Historia de Usuario:</b> 2.- Administración de Registros de Elementos de Inventario
<b>Nombre:</b> Comprobación de funcionalidad de Administración de Registros de Elementos de Inventario.
<b>Descripción:</b> Se comprobará que el sistema pueda administrar de manera apropiada los registros de elementos de inventario, poniendo a prueba diversas validaciones, a nivel de <i>Frontend</i> como de <i>Backend</i> , dando tratamiento de manera apropiada a la imagen y atributos adicionales proporcionados por el usuario.
<b>Condiciones de Ejecución:</b> La entrada del usuario puede ser incorrecta, la imagen que inserte el usuario puede ser de distintos formatos compatibles, el registro puede no pertenecer a una categoría con atributos adicionales.
<p><b>Entrada/Pasos de Ejecución:</b> El usuario realizará los siguientes pasos.</p> <ol style="list-style-type: none"> <li>1. Accede a la vista de Creación de Registro de Elemento de Inventario.</li> <li>2. Crea un Registro con datos Incompletos.</li> <li>3. Crea un Registro con datos correctos perteneciendo a una categoría con atributos adicionales, la imagen es de formato PNG.</li> <li>4. Comprueba el registro en el listado de elementos.</li> <li>5. Revisa los detalles del registro, corroborando los atributos adicionales insertados.</li> <li>6. Vuelve a editar el registro que creó el usuario, el registro ahora pertenecerá a una categoría sin atributos adicionales, la imagen que añadió el usuario es en formato JPG.</li> <li>7. Vuelve a editar el registro que creó el usuario, se adjunta un archivo XLSX.</li> <li>8. Comprueba el registro en el listado de elementos.</li> <li>9. Revisa los detalles del registro, corroborando la ausencia de atributos adicionales.</li> <li>10. Elimina el registro creado del elemento de inventario.</li> </ol>
<b>Resultado Esperado:</b> Se almacenará de manera correcta el registro de elemento de inventario además del registro de atributos adicionales si es necesario, adjuntar un archivo de diferente a una imagen o enviar un formulario incompleto retornará un mensaje de error, El registro aparecerá en la lista de registros de elementos de inventario, se podrá consultar sus atributos de manera apropiada.
<b>Evaluación de la Prueba:</b> El sistema manejó de manera apropiada los formularios incompletos o con archivos diferentes a una imagen, se generó de manera exitosa los registros de elementos y de atributos adicionales en los casos donde se requiriera, se desplegó con éxito el registro de elemento de inventario en la lista dedicada, los resultados se presentaron de manera apropiada y consistente a la categoría del elemento.

*Tabla 2. 3. Prueba 2. Comprobación de funcionalidad de Administración de Registros de Elementos de Inventario.*

<b>Prueba 3</b>
<b>Historia de Usuario:</b> 3.- Creación de Reportes de Mantenimiento Exportables a PDF
<b>Nombre:</b> Comprobación de funcionalidad de Generación y Exportación de Reportes de Mantenimiento.
<b>Descripción:</b> Se comprobará que el sistema pueda generar reportes de mantenimiento individuales, además de exportación a un archivo de formato PDF.
<b>Condiciones de Ejecución:</b> La entrada del usuario puede contener una discrepancia con las fechas del Ultimo mantenimiento y el mantenimiento actual.
<p><b>Entrada/Pasos de Ejecución:</b> El usuario realizará los siguientes pasos.</p> <ol style="list-style-type: none"> <li>1. Acceder a la vista de Listado de Elementos de Inventario.</li> <li>2. Generar un reporte de mantenimiento del elemento, insertando una fecha discrepante en la fecha del mantenimiento.</li> <li>3. Generar un reporte de mantenimiento del elemento, con datos válidos.</li> <li>4. Acceder a la vista de Listado de Mantenimientos</li> <li>5. Exportar el documento en formato PDF.</li> <li>6. Descargar el documento PDF Generado.</li> </ol>
<b>Resultado Esperado:</b> Se generará de manera correcta un registro de mantenimiento, lo datos de este registro podrán ser utilizados para generar un documento en formato PDF con información del elemento y el mantenimiento que recibió. Si al instante de generar un registro hay una discrepancia entre la fecha del último mantenimiento que recibió el elemento y la fecha del mantenimiento actual se retornará un mensaje de informando de la discrepancia.
<b>Evaluación de la Prueba:</b> El sistema valida de manera apropiada el registro, divulgando mensaje de notificando la discrepancia de fechas, se crea con éxito el registro de mantenimiento, Se logró exportar con éxito un documento PDF descargable con la descripción del reporte de mantenimiento.

*Tabla 2. 4. Prueba 3. Comprobación de funcionalidad de Generación y Exportación de Reportes de Mantenimiento.*

<b>Prueba 4</b>
<b>Historia de Usuario:</b> 4.- Generación automática de registros de Traspasos
<b>Nombre:</b> Comprobación de funcionalidad de Generación automática de Registros de Traspaso.
<b>Descripción:</b> Se comprobará que el sistema pueda generar de manera automática registros de traspasos de elementos, particularmente al editar un registro de elemento de inventario.
<b>Condiciones de Ejecución:</b> No aplica.
<b>Entrada/Pasos de Ejecución:</b> El usuario realizará los siguientes pasos. <ol style="list-style-type: none"> <li>1. Acceder a la vista de Edición de Registro de Elemento de Inventario.</li> <li>2. Editar atributos de Edificio o Ubicación del registro.</li> <li>3. Acceder a la vista de Listado de Traspasos.</li> </ol>
<b>Resultado Esperado:</b> Se generará de manera automática un registro de traspaso del sistema, conteniendo la fecha de la ubicación/Edificio anterior del elemento, así como de su nueva Ubicación/Edificio.
<b>Evaluación de la Prueba:</b> El sistema genera de manera exitosa el registro de traspaso, respondiendo de manera apropiada a la edición de uno o ambos atributos de localización del elemento, el listado muestra datos consistentes a los generados a los registros de elemento de inventario con la ubicación actualizada.

*Tabla 2. 5. Prueba 4. Comprobación de funcionalidad de Generación automática de Registros de Traspaso.*

<b>Prueba 5</b>
<b>Historia de Usuario:</b> 5.- Exportación e Importación de Registros De Elementos de Inventario a través de un archivo XLSX.
<b>Nombre:</b> Comprobación de funcionalidad de Exportación e Importación de Registros De Elementos de Inventario a través de un archivo XLSX
<b>Descripción:</b> El usuario podrá exportar e importar registros de elementos de inventario a un archivo XLSX, la importación seguirá un formato predefinido para importar registros de elementos.
<b>Condiciones de Ejecución:</b> El archivo puede contener datos incorrectos o no seguir el formato preestablecido para importar de manera apropiada los registros de elementos.
<b>Entrada/Pasos de Ejecución:</b> El usuario realizará los siguientes pasos. <ol style="list-style-type: none"> <li>1. Acceder a la vista de Exportación de Registros de Elementos.</li> <li>2. Descargar archivo en formato XLSX con Registros de elementos.</li> <li>3. Acceder a la vista de Importación de Registros de Elementos.</li> <li>4. Subir Archivo de Importación con Formato Incorrecto.</li> <li>5. Subir Archivo de Importación con Información Incorrecto.</li> <li>6. Subir Archivo de Importación con Información Correcta.</li> <li>7. Acceder a la vista de Listado de Traspasos.</li> </ol>
<b>Resultado Esperado:</b> Se generará de manera correcta un archivo XLSX que contiene todos registros de elementos de inventario y atributos adicionales correspondientes, la importación de elementos serpa consistente y generará registros de elementos de inventario completos, excluyendo atributos como etiquetas e imagen. Se rechazará cualquier archivo adjunto si no es de formato XLSX, si el archivo adjunto no posee de datos correctos o válidos se retornará un mensaje especificando donde se encuentra el dato incorrecto el documento XLSX.
<b>Evaluación de la Prueba:</b> El sistema generó de manera exitosa el archivo XLSX con los registros de elementos de inventario, el sistema respondió apropiadamente a cada problemática existente a la hora de importar archivos XLSX con registros de elementos de inventario, desde un formato de archivos incorrecto, hasta datos incorrectos, señaló de manera apropiada la ubicación de cada error el sistema.

*Tabla 2. 6. Prueba 5. Comprobación de funcionalidad de Exportación e Importación de Registros De Elementos de Inventario a través de un archivo XLSX.*

<b>Prueba 6</b>
<b>Historia de Usuario:</b> 6.- Creación y Asignación de Roles a Usuarios del sistema.
<b>Nombre:</b> Comprobación de funcionalidad de Creación y Asignación de Roles a Usuarios del sistema.
<b>Descripción:</b> Se comprobará que un administrador o usuario privilegiado pueda crear y asignar distintos roles a otros usuarios, limitando de esta manera las acciones que puede realizar.
<b>Condiciones de Ejecución:</b> No aplica.
<b>Entrada/Pasos de Ejecución:</b> El usuario realizará los siguientes pasos. <ol style="list-style-type: none"> <li>1. Acceder a la vista de Creación de Roles.</li> <li>2. Crear un Rol asignando un conjunto de permisos.</li> <li>3. Acceder a la vista de Edición de Roles de usuario.</li> <li>4. Asignar Rol/Roles a Usuario.</li> </ol>
<b>Resultado Esperado:</b> podrá crear, modificar y eliminar eventos por medio de un calendario interactivo, este calendario alojará eventos futuros a la fecha presenta, estos eventos serán anunciados a través de un canal de notificaciones dedicado de Telegram, el usuario podrá difundir mensajes a este grupo de manera directa.
<b>Evaluación de la Prueba:</b> El sistema genera de manera exitosa los roles del sistema, además de asígnalos de manera exitosa a un usuario. Los roles se reflejan en cada instancia de un usuario del sistema de manera acorde a los permisos atribuidos a cada usuario.

*Tabla 2. 7. Prueba 6. Comprobación de funcionalidad de Creación y Asignación de Roles a Usuarios del sistema.*

<b>Prueba 7</b>
<b>Historia de Usuario:</b> 7.- Calendario Interactivo de Creación de Eventos y Notificación por Canal de Telegram
<b>Nombre:</b> Comprobación de funcionalidad de Calendario Interactivo de Eventos y Notificación por Canal de Telegram.
<b>Descripción:</b> Se comprobará que el usuario crear, modificar y eliminar eventos por medio de un calendario interactivo, además de difundir mensajes a un canal dedicado de Telegram de manera directa o por eventos.
<b>Condiciones de Ejecución:</b> -
<b>Entrada/Pasos de Ejecución:</b> El usuario realizará los siguientes pasos. <ol style="list-style-type: none"> <li>1. Acceder a la vista de Dashboard.</li> <li>2. Crear un Evento.</li> <li>3. Arrastrar Evento a fecha equivalente o anterior a la actual.</li> <li>4. Arrastrar Evento a fecha futura.</li> <li>5. Acceder a la vista de Notificaciones.</li> <li>6. Difundir mensaje a canal de Telegram.</li> </ol>
<b>Resultado Esperado:</b> El usuario podrá crear, modificar y eliminar eventos por medio de un calendario interactivo, el calendario alojará eventos futuros a la fecha presente, estos eventos serán anunciados a través de un canal de notificaciones dedicado de Telegram, el usuario podrá difundir mensajes a este grupo de manera directa.
<b>Evaluación de la Prueba:</b> El sistema genera de manera exitosa cada evento del calendario, el calendario es responsivo e intuitivo para el usuario. El calendario invalida la acción derivada de arrastrar un evento a una fecha no válida para el calendario, los eventos generados en el calendario se difunden de manera exitosa en el canal de Telegram dedicado al sistema a una hora establecida (11:00 AM), La difusión de mensajes al canal de Telegram fue exitosa.

*Tabla 2. 8. Prueba 7. Comprobación de funcionalidad de Calendario Interactivo de Eventos y Notificación por Canal de Telegram.*

## Capítulo 3 . Resultados

---

Este apartado abordará la muestra de la versión final del Sistema de Inventario *Airone*, mostrando cada funcionalidad y vista con las que cuenta el sistema, en conjunto a una breve descripción de cada una de estas.

### 3.1 Muestra del sistema final

---

El Sistema de Inventario *Airone* ha finalizado su proceso de desarrollo, por lo tanto, se cuenta con un producto terminado, a continuación, se mostrará el sistema final generado en base a la aplicación de las fases y actividades propuestas por la metodología XP, describiendo y mostrando cada una de las interfaces, vistas y funcionalidades que contiene el sistema final.

#### 3.1.1 Login del sistema

La figura 3.1 muestra la vista de *Login* del sistema, esta será la primera vista que el usuario tendrá al acceder al Sistema de Inventario *Airone*, esta vista incluye botones para iniciar sesión, Recuperar una cuenta, Registrarse y Observar un apartado de Acerca de del sistema.

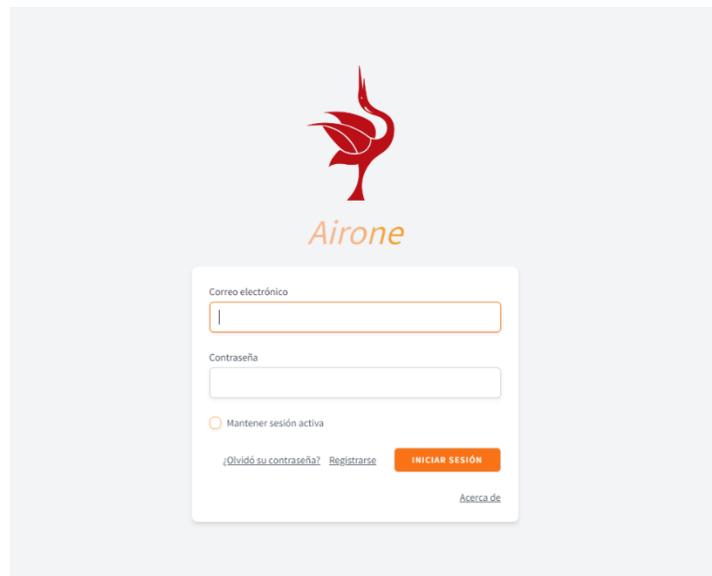
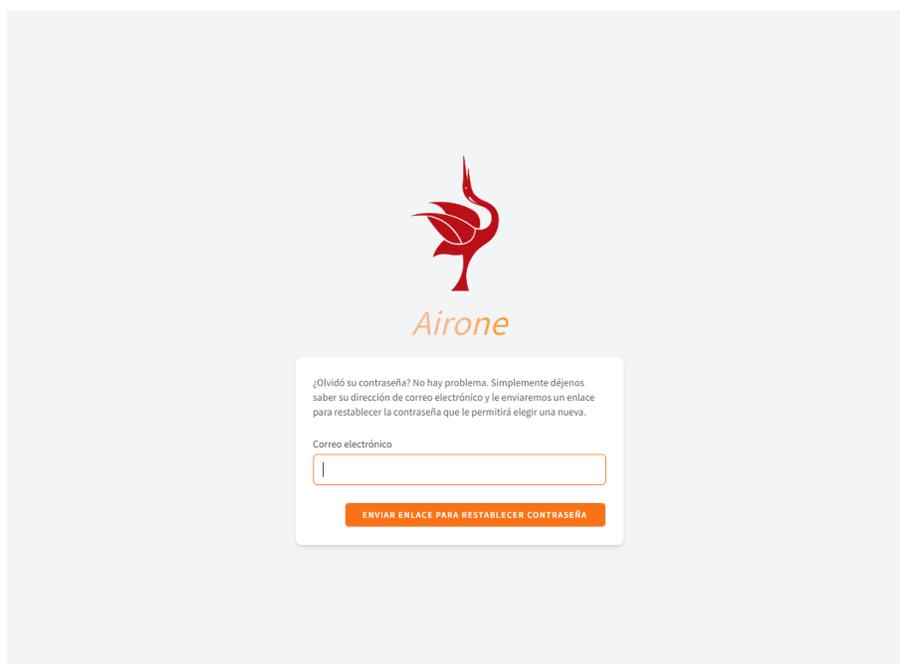


Figura 3. 1. Login. Vista Principal.

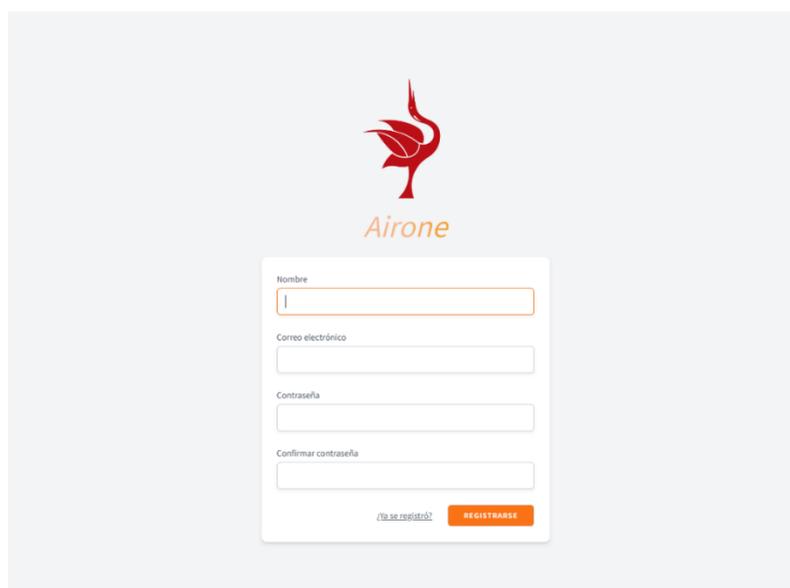
La Figura 3.2 muestra la vista de recuperación de contraseña del sistema, esta vista es el medio por el cual un usuario puede restaurar la contraseña de su cuenta en caso de olvidarla, requiriendo únicamente el correo electrónico de la cuenta que se desea recuperar.



The screenshot shows the Airone logo at the top center. Below it, a text box contains the message: "¿Olvidó su contraseña? No hay problema. Simplemente déjenos saber su dirección de correo electrónico y le enviaremos un enlace para restablecer la contraseña que le permitirá elegir una nueva." Below this text is a single input field labeled "Correo electrónico" with a cursor inside. At the bottom of the form is an orange button with the text "ENVIAR ENLACE PARA RESTABLECER CONTRASEÑA".

*Figura 3. 2. Login. Restauración de contraseña.*

La Figura 3.3 muestra la vista de registro de usuario, la creación de una nueva cuenta requiere de rellenar campos obligatorios, estos campos albergarán el nombre de usuario, correo electrónico y contraseña, cada uno de ellos, siendo parámetros obligatorios para crear una cuenta nueva.



The screenshot shows the Airone logo at the top center. Below it, a registration form is displayed with four input fields: "Nombre", "Correo electrónico", "Contraseña", and "Confirmar contraseña". Each field has a cursor. At the bottom of the form, there is a link that says "¿Ya se registró?" and an orange button labeled "REGISTRARSE".

*Figura 3. 3. Login. Registro de cuenta.*

### 3.1.2 Landing del sistema

La Figura 3.4 muestra la vista de *Landing*, la función de esta vista es proporcionar al usuario de una página intermedia donde acceder a vistas y opciones como la Configuración de la cuenta, *Dashboard*, Cierre de Sesión y Acerca De.

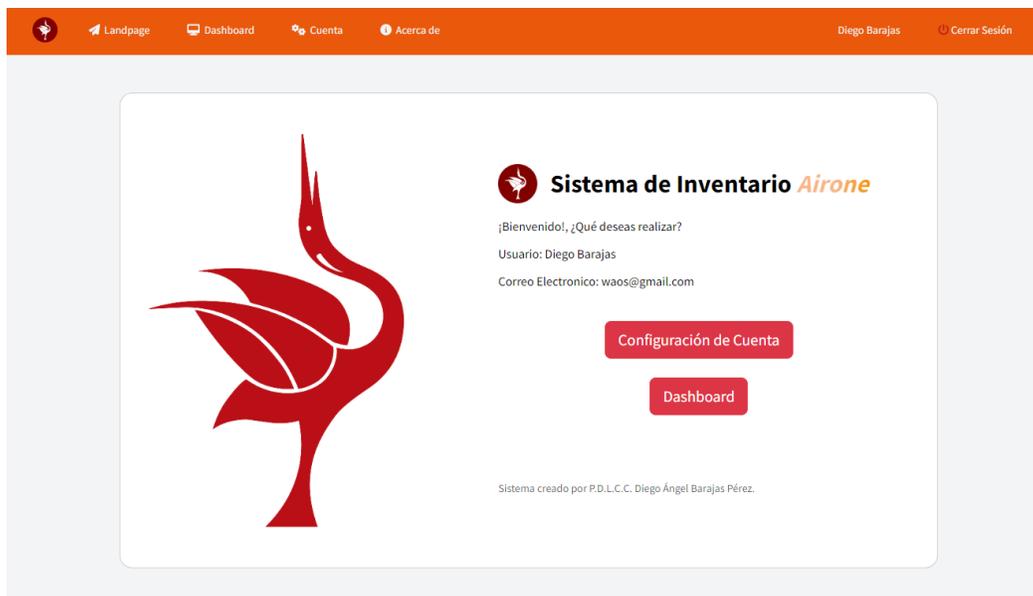


Figura 3. 4 Landing. Vista Principal.

Las Figuras 3.5 y 3.6 muestran la vista de Configuración de cuenta, este es el medio por el cual se podrán actualizar datos de la cuenta, como lo es el nombre del usuario, correo electrónico o contraseña, además de habilitar una autenticación de dos factores, control de sesiones y eliminar de cuenta.

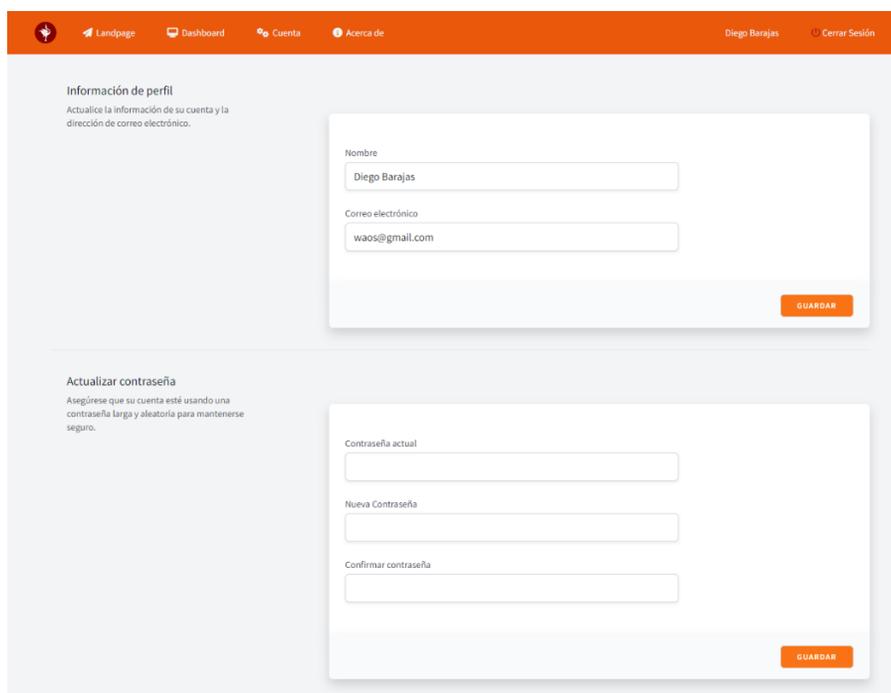


Figura 3. 5. Landing. Configuración de Cuenta, Actualización de datos.

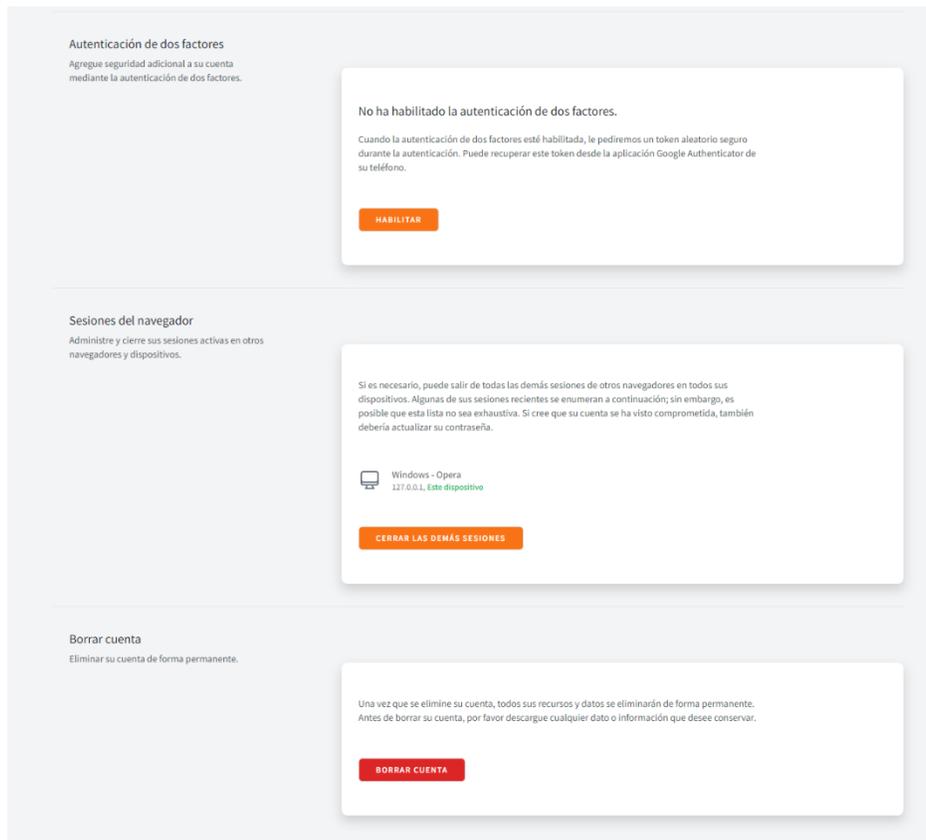


Figura 3. 6. Landing. Configuración de Cuenta, Autenticación, Sesiones y Eliminación de cuenta.

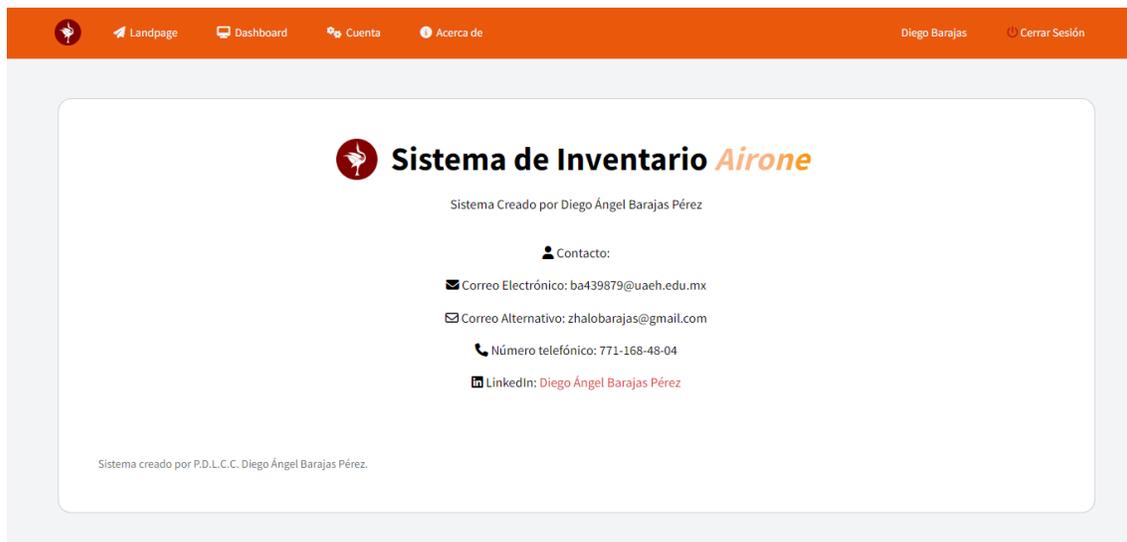


Figura 3. 7. Landing. Acerca De.

### 3.1.3 Dashboard del sistema

La Figura 3.8 muestra el *Dashboard* del sistema, esta vista contiene el formato estándar de las vistas que alojan las principales funcionalidades del sistema, el *Dashboard* se conforma de un *Sidebar* con todas las funcionalidades que puede realizar el usuario, estas se encuentran divididas en seis grupos, los cuales consisten de (i) Administración del sistema, (ii) Administración de elementos, (iii) Usuarios y Roles, (iv) Exportación e Importación de Datos, y (v) Centro de Notificaciones, la cabecera del sistema cuenta con un icono de modo oscuro persistente para este apartado del sistema.

La vista principal del *Dashboard* contiene un apartado de estadísticas mensuales y globales del sistema, particularmente, cuenta con datos como usuarios registrados, Elementos registrados, mantenimientos realizados y traspasos realizados.

Además de las estadísticas, esta vista cuenta con el calendario de eventos, el calendario es interactivo, permitiendo a un usuario, generar, editar y eliminar eventos, estos eventos serán anunciados de manera automática en un canal dedicado de Telegram a una hora definida (11:00 AM) de acuerdo a la fecha del evento. Las Figuras 3.9 a 3.11 muestran el funcionamiento del calendario.



Figura 3. 8. Dashboard. Estadísticas y Vista principal.

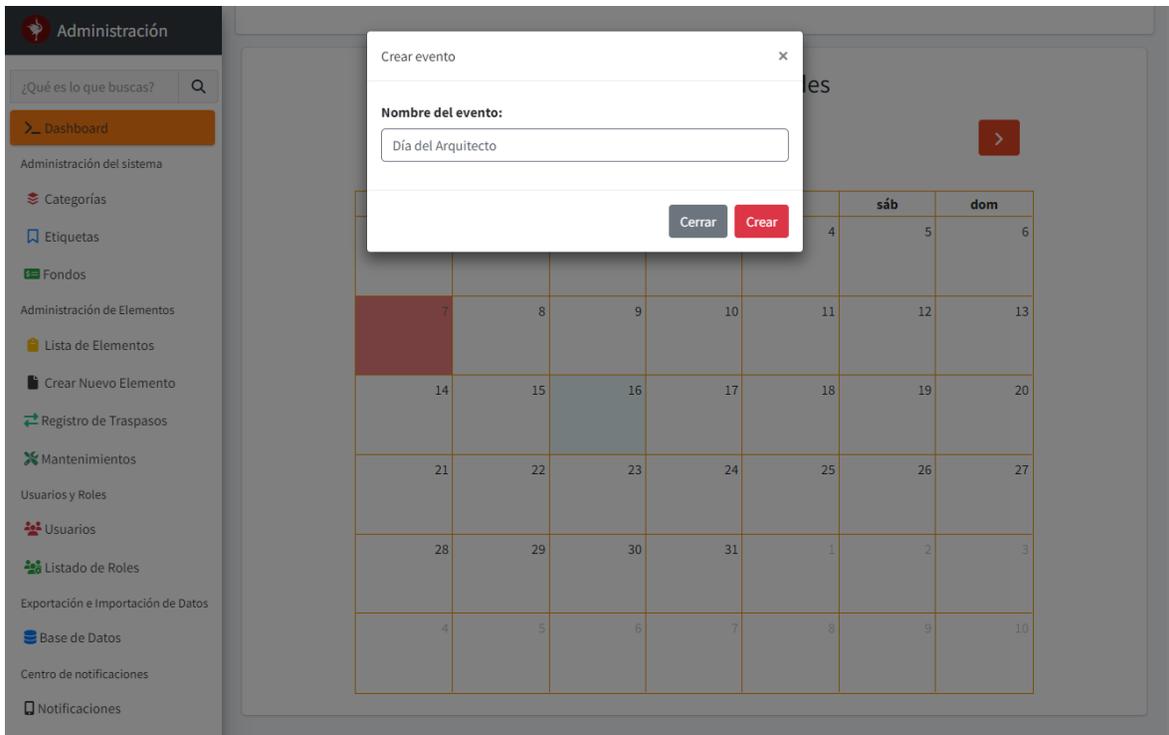


Figura 3. 9. Calendario de Eventos. Creación de Evento.

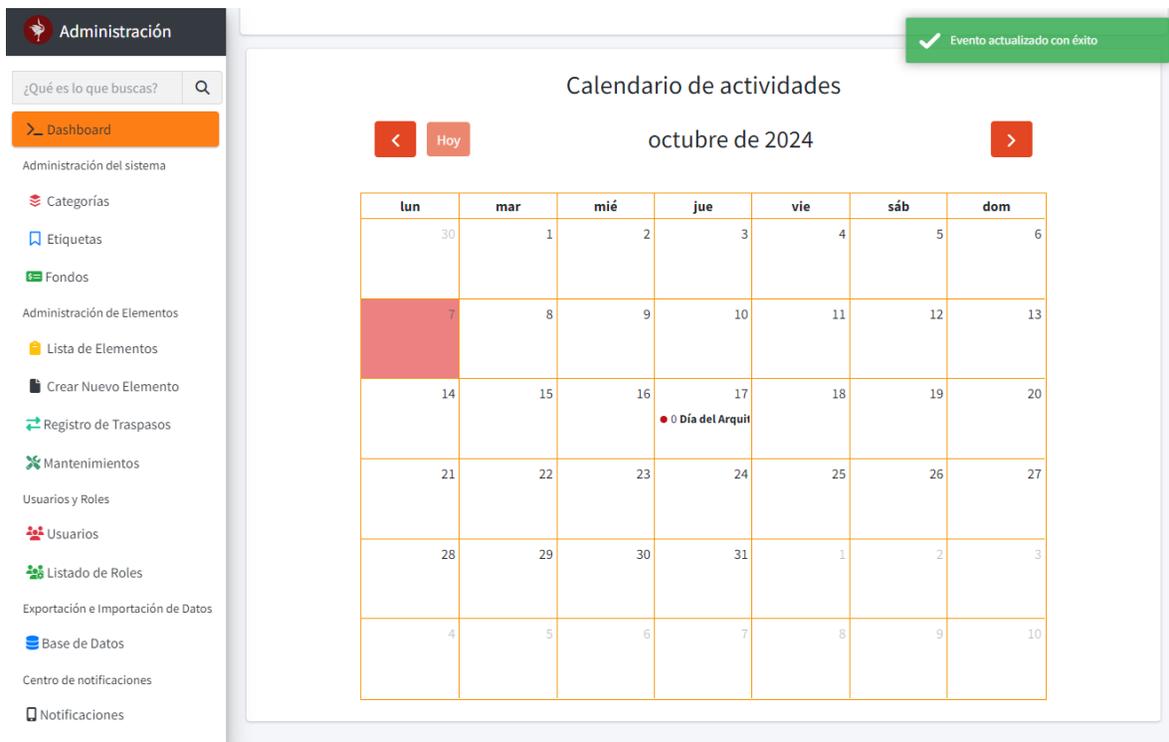


Figura 3. 10. Calendario de Eventos. Actualización de Evento.

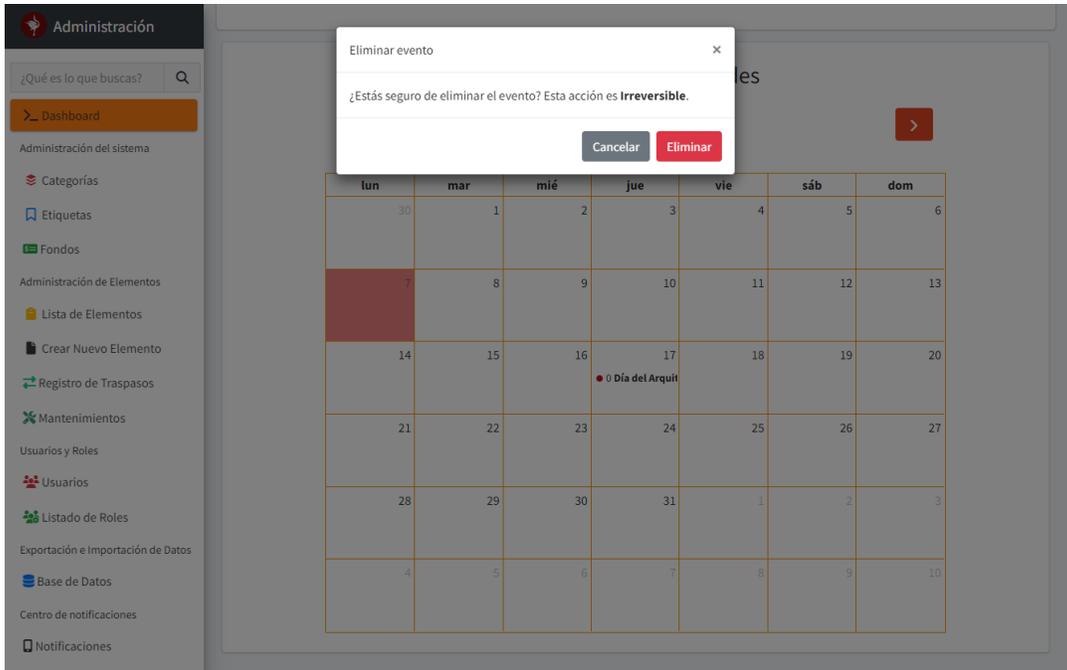


Figura 3. 11. Calendario de Eventos. Eliminación de Evento.

### 3.1.4 Administración del Sistema

La administración del sistema es un apartado compuesto de un conjunto de vistas para la creación, edición y eliminación de registros de los atributos de Categoría, Fondo y Etiquetas, cada uno de estos atributos forman parte de los registros de elementos. La administración de cualquiera de estos atributos se compone de una vista para el listado, creación y edición del registro. Las Figuras 3.12 a 3.20 muestran las vistas para la administración de cada uno de estos atributos.

El listado de categorías cuenta con seis categorías predefinidas, las cuales no pueden ser editadas, estas corresponden a aquellas a las que se les atribuye un registro de atributos adicionales correspondientes a un registro de elemento perteneciente a cualquiera de estas categorías, este es el medio por el cual el usuario puede realizar acciones como acceder a la vista de creación, edición o eliminar una categoría.

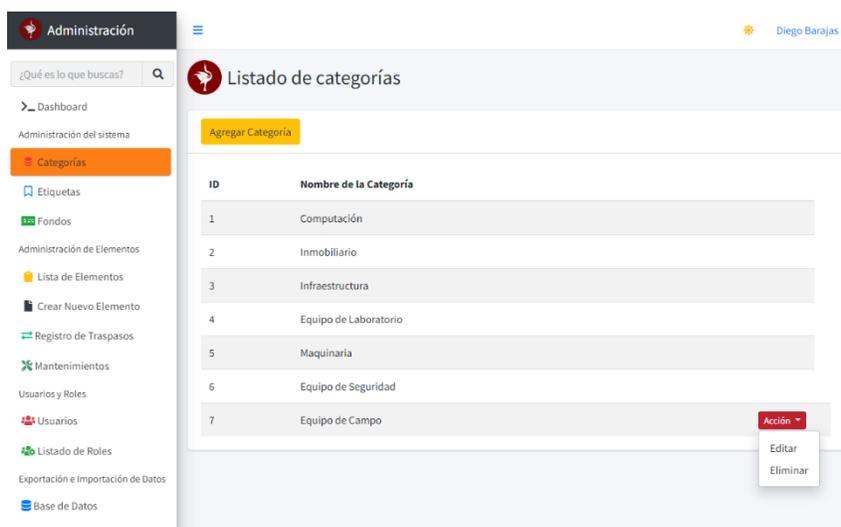


Figura 3. 12. Administración del Sistema. Listado de Categorías.

La creación de categorías consiste únicamente de un formulario para insertar el nombre de la categoría a crear, como se muestra en la Figura 3.13.

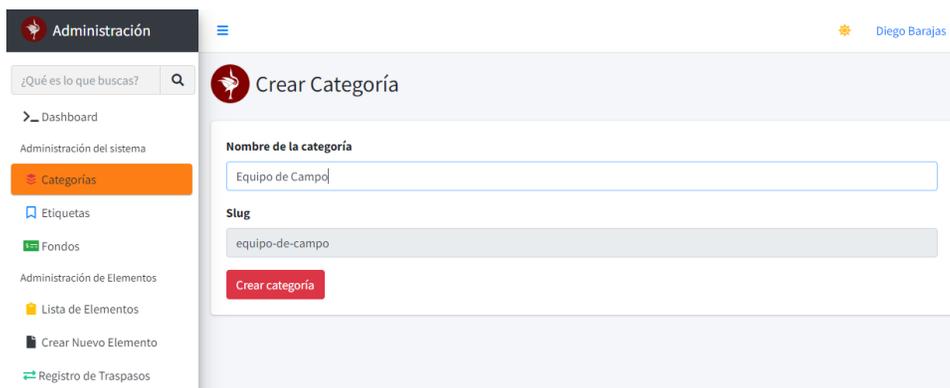


Figura 3. 13. Administración del Sistema. Creación de Categoría.

La edición de categorías contiene una vista muy similar a aquella de creación de categorías, una vez el usuario haya creado una categoría será redirigido a la vista respectiva de edición de categoría que ha creado, esta redirección viene acompañada de una notificación. Como muestra la Figura 3.14.

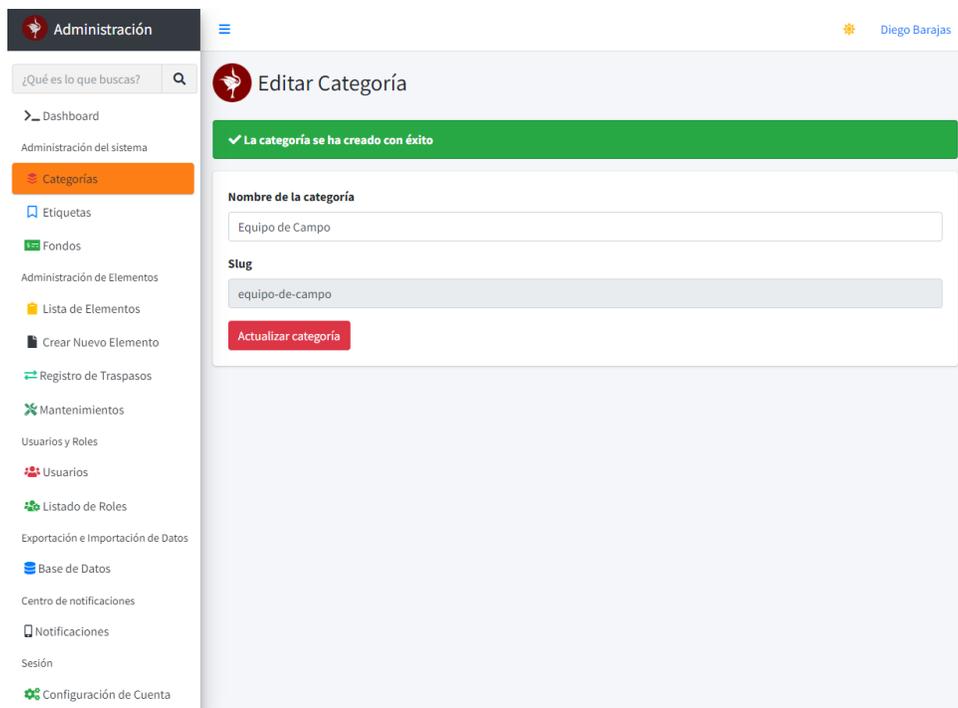


Figura 3. 14. Administración del Sistema. Edición de Categoría.

El listado de etiquetas cuenta con una interfaz similar a la del listado de categorías, contando con un patrón de diseño e interacciones similares, no se cuenta con un conjunto predeterminado de etiquetas, dando al usuario libertad de generar las etiquetas que desee, como se muestra en la Figura 3.15.

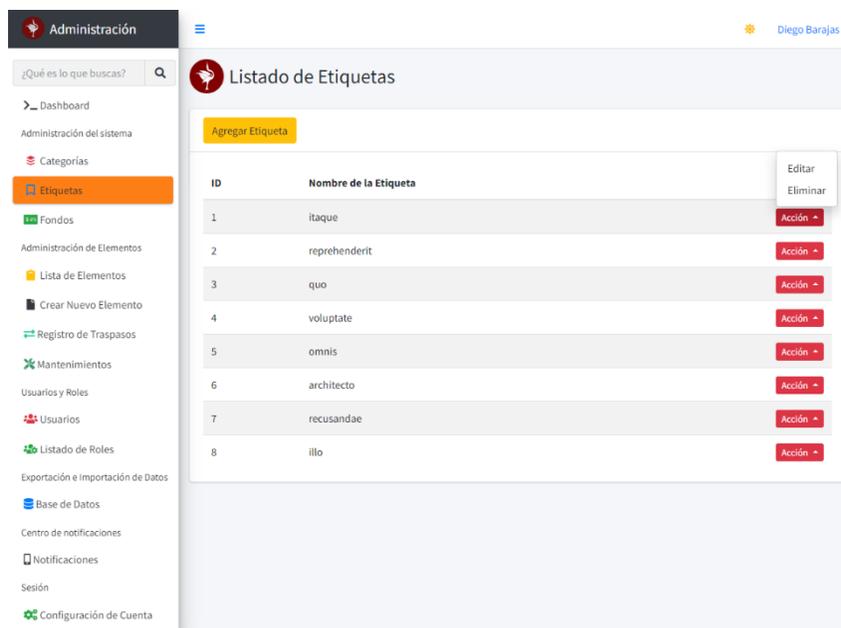


Figura 3. 15. Administración del Sistema. Listado de Etiquetas.

La creación y edición de etiquetas están conformadas de dos apartados, el primero de ellos para dar un título a la etiqueta, el otro para asignar un color a esa etiqueta, el color seleccionado podrá apreciarse en vistas como el listado de elementos y detalles del registro de elemento, las figuras 3.16 y 3.17 muestran ambas vistas.

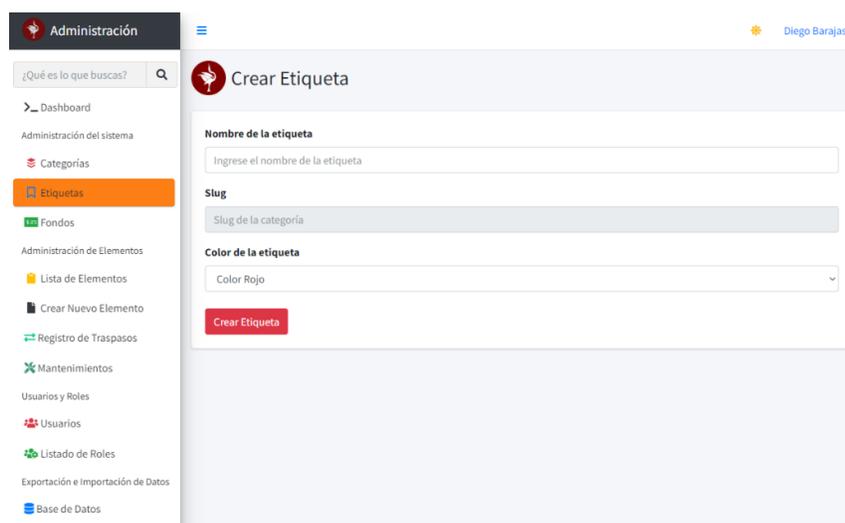


Figura 3. 16. Administración del Sistema. Creación de Categoría.

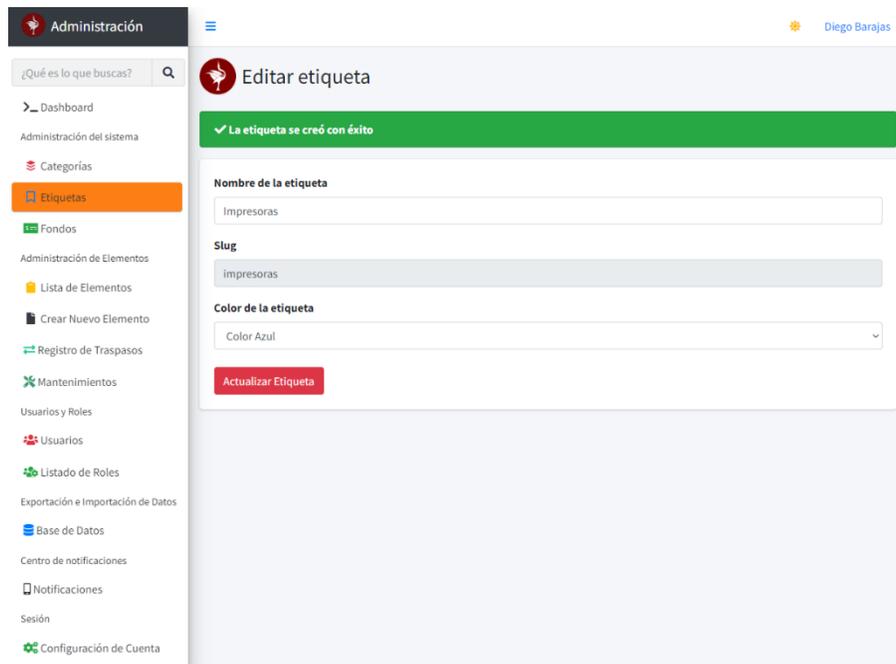


Figura 3. 17. Administración del Sistema. Edición de Etiqueta.

El listado de fondos cuenta con siete registros predeterminados, cubriendo de manera general los principales fondos a los que puede pertenecer un elemento de inventario del área, si se requiriera de más atributos, estos pueden ser añadidos y asignados, de acuerdo al modo de uso del usuario. La Figura 3.18 muestra la vista con el listado.

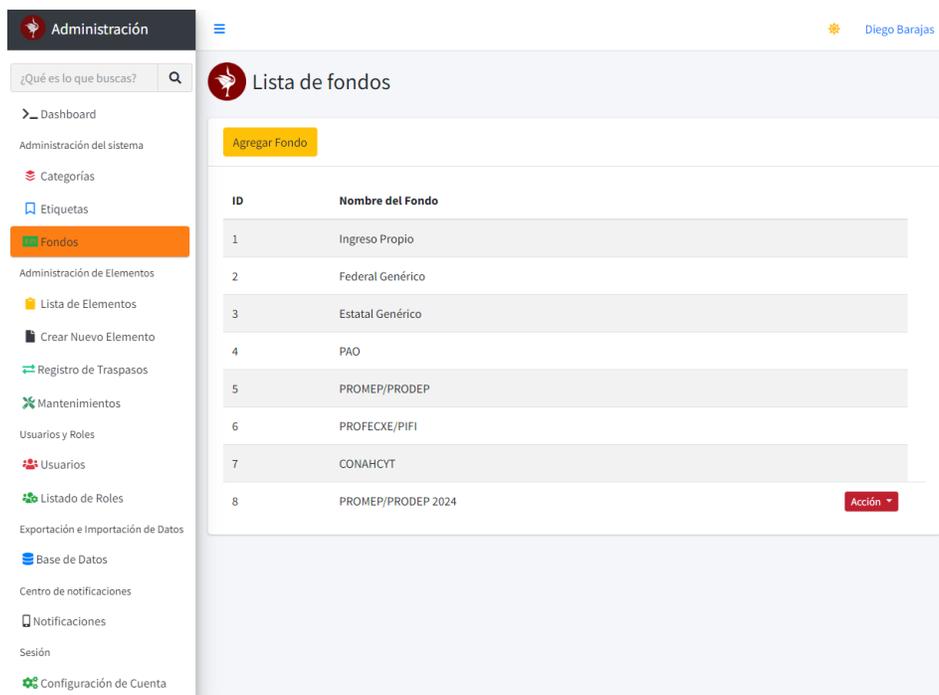


Figura 3. 18. Administración del sistema. Listado de fondos.

La creación y edición de registros de fondos cuenta con la misma lógica que con, por ejemplo, categorías, siguiendo un mismo patrón de diseño, como puede observarse en las figuras 2.19 y 3.20, en esta última puede apreciarse un ejemplo de mensaje de error generado, derivado de actualizar un registro sin cambio alguno.

The screenshot shows the 'Crear Nuevo Fondo' (Create New Fund) form. On the left is a sidebar menu with 'Fondos' highlighted. The main form has two input fields: 'Nombre del fondo' (Fund Name) containing 'PROMEP/PRODEP 2024' and 'Slug' containing 'promep-prodep-2024'. A red 'Crear fondo' button is at the bottom.

Figura 3. 19. Administración del Sistema. Creación de Fondo.

The screenshot shows the 'Actualizar Fondo' (Update Fund) form. A teal error banner at the top reads '¡El nuevo nombre del fondo es idéntico, prueba uno diferente.' (The new fund name is identical, try a different one). The input fields for 'Nombre del fondo' and 'Slug' both contain 'PROMEP/PRODEP 2024'. A red 'Actualizar fondo' button is at the bottom.

Figura 3. 20. Administración del Sistema. Edición de Fondo.

### 3.1.4 Administración del Elementos

La administración de Elementos se conforma de un conjunto de vistas enfocadas en la creación, actualización y eliminación de registros de elementos, además de un listado de registros de traspasos, así como la generación de reportes de mantenimiento, esta última cuenta con un medio por el cual los registros de mantenimiento podrán ser exportados a formato PDF. Las Figuras 3.21 a 3.40 muestran las vistas para de cada una de estas funcionalidades.

La Figura 3.21 muestra el listado de elementos, esta cuenta con un buscador integrado, de acuerdo a los parámetros proporcionados por el usuario, se filtrarán todos los registros y desplegará aquellos equivalentes. Bastará con hacer *click* al nombre de un registro de elemento para desplegar más detalles acerca del este, la Figura 3.22 muestra la vista de detalles de registro.

The screenshot shows the 'Lista de Elementos' page. On the left is a sidebar with navigation options like 'Dashboard', 'Categorías', 'Etiquetas', 'Fondos', 'Crear Nuevo Elemento', 'Registro de Traspasos', 'Mantenimientos', 'Usuarios y Roles', 'Usuarios', 'Listado de Roles', 'Exportación e Importación de Datos', 'Base de Datos', 'Centro de notificaciones', 'Notificaciones', 'Sesión', and 'Configuración de Cuenta'. The main content area has a search bar and filters for 'Nombre del Elemento', 'Categoría', 'Fondo', 'Etiqueta', 'Edificio', 'Ubicación', and 'Status'. Below the filters is a table with columns: ID, Nombre del Elemento, Edificio, Ubicación, Fondo, Categoría, Status, Fecha de Registro, and Etiquetas. A dropdown menu is open over the second row, showing options: 'Acción', 'Editar', 'Eliminar', and 'Mantenimiento'.

ID	Nombre del Elemento	Edificio	Ubicación	Fondo	Categoría	Status	Fecha de Registro	Etiquetas
7	Test_GIF	Edificio H	Cubículos de Investigadores	Ingreso Propio	Computación	Alta	2024-10-02	quo, arquitecto
1	Computadora HP Slimline	Edificio A	Oficinas de Coordinadores	Ingreso Propio	Computación	Alta	2024-07-25	
	Organizador	Edificio A	Oficinas de Coordinadores	Ingreso Propio	Inmobiliario	Alta	2024-07-25	quo, voluptate
3	Tablas de madera	Edificio A	Oficinas de Coordinadores	Ingreso Propio	Infraestructura	Alta	2024-07-25	

Figura 3. 21. Administración de Elementos. Listado y Buscador de elementos.

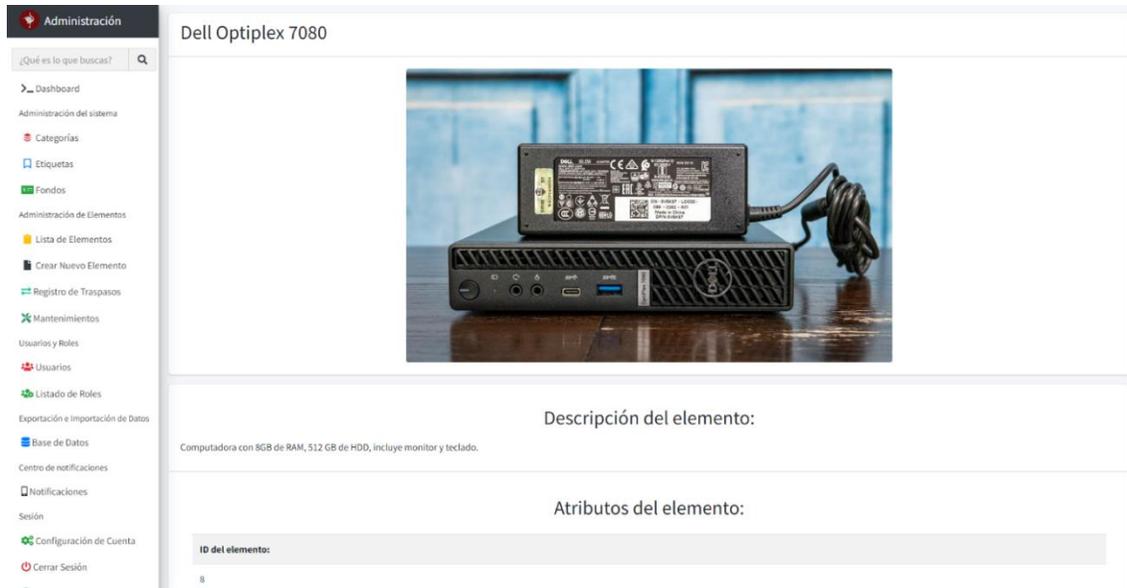


Figura 3. 22. Administración de Elementos. Detalles de Registro de Elemento.

La vista de registro de elementos es el medio donde un usuario puede generar registros de elementos, integrando atributos e incluso, una imagen representativa del elemento, de acuerdo a la categoría que seleccione el usuario, la vista puede desplegar casillas para atributos adicionales, la edición de elementos utiliza el mismo diseño, como se observa en las Figuras 3.23 y 3.24.

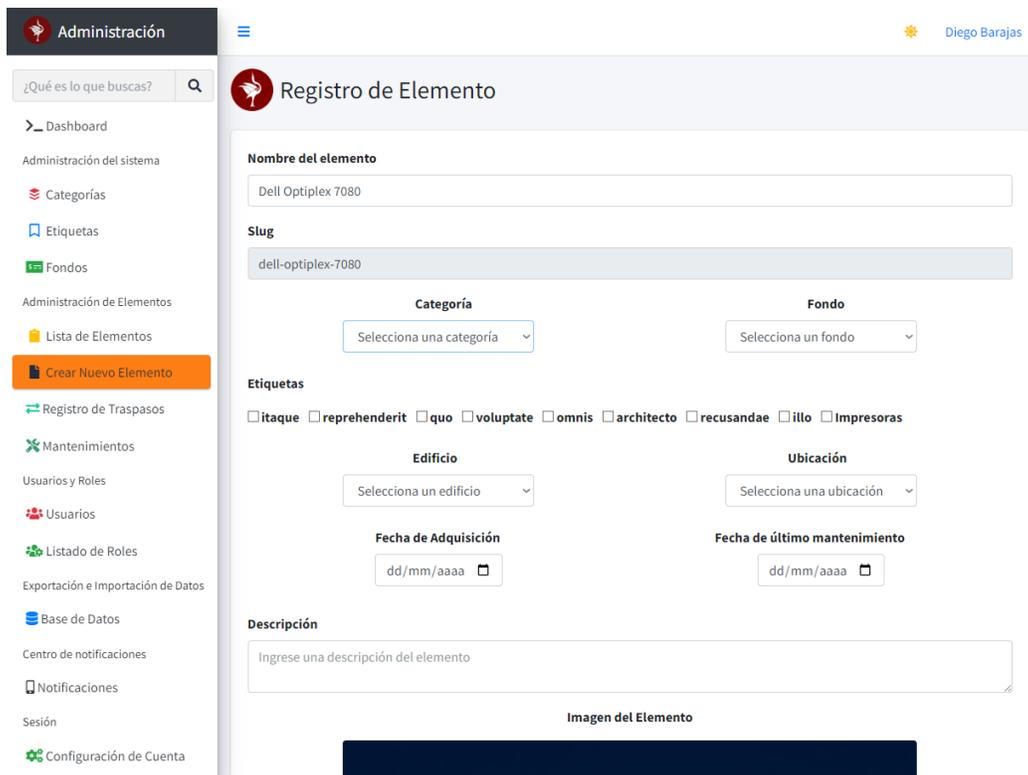


Figura 3. 23. Administración de Elementos. Creación de Elemento.

**Administración** Diego Barajas

¿Qué es lo que buscas?

**Editar Registro de Elemento**

✓ El elemento se ha creado con éxito

**Nombre del elemento**  
Dell Optiplex 7080

**Slug**  
dell-optiplex-7080-8

**Categoría** Computación **Fondo** Federal Genérico

**Etiquetas**  
 itaque  reprehenderit  quo  voluptate  omnis  architecto  recusandae  illo  Impresoras

**Edificio** Edificio H **Ubicación** Cubículos de Investigadore

**Fecha de Adquisición** 03/09/2024 **Fecha de último mantenimiento** 15/01/2025

**Descripción**  
Computadora con 8GB de RAM, 512 GB de HDD, incluye monitor y teclado.

**Marca**

Figura 3. 24. Administración de Elementos. Edición de Elemento.

El listado de registros de traspasos muestra, en orden del más reciente, los traspasos generados a los distintos elementos, cada registro contiene la ubicación y edificio anterior donde se encontraba el elemento de inventario, además de su nueva ubicación y edificio, en conjunto a la fecha del traspaso, como puede observarse en la Figura 3.25.

**Administración** Diego Barajas

¿Qué es lo que buscas?

**Registro de Traspasos**

ID	Elemento	Ubicación Anterior	Edificio Anterior	Nueva Ubicación	Nuevo Edificio	Fecha Traspaso
1	Computadora HP Slimline	Edificio H	Aulas	Edificio AAlyA	Cubículos de Investigadores	2024-05-03

Figura 3. 25. Administración de Elementos. Registro de Traspasos.

La generación de registros de mantenimiento consiste de actualizar la fecha de mantenimiento del elemento a aquella cuando recibió su ultimo mantenimiento, además de añadir una descripción del mantenimiento realizado, todos estos serán utilizados para el documento que podrá ser exportado en formato PDF, en la vista de listado de mantenimientos realizados, las figuras 3.26 a 3.29 muestran cada vista dentro de este proceso.

**Administración** Diego Barajas

¿Qué es lo que buscas?

**Registro de mantenimiento**

**Nombre del elemento**  
Dell Optiplex 7080

**Fecha de Último Mantenimiento**  
06/02/2025

**Fecha de Mantenimiento**  
dd/mm/aaaa

**Descripción del mantenimiento**  
Ingrese la descripción del mantenimiento

**Crear Registro**

Figura 3. 26. Administración de Elementos. Creación de reporte de Mantenimiento.

**Administración** Diego Barajas

¿Qué es lo que buscas?

**Listado de mantenimientos**

✓ El registro se ha creado con éxito

ID Registro	ID Elemento	Elemento	Fecha de Mantenimiento Anterior	Fecha de Mantenimiento	Generar Documento
1	1	Computadora HP Slimline	2024-03-13	2024-08-07	
2	8	Dell Optiplex 7080	2025-01-15	2025-02-06	

Figura 3. 27. Administración de Elementos. Listado de reporte de Mantenimiento.

Reporte\_Elemento\_8 1 / 1 100%



1



**UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO**  
Pachuca, Hidalgo a 07 de octubre de 2024.

Registro de mantenimiento de Equipo.

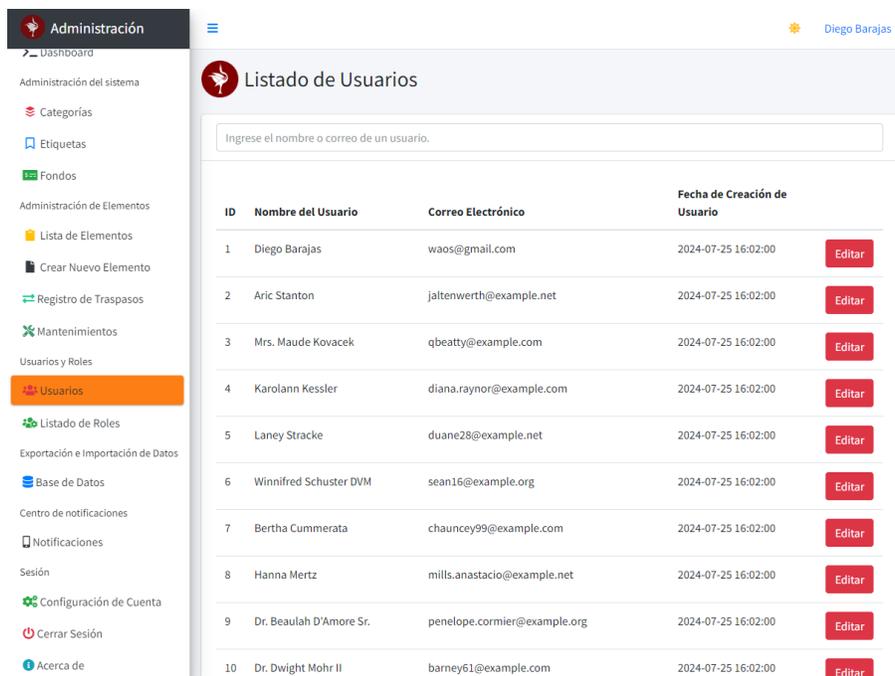
<b>Elemento/Equipo:</b>	Dell Optiplex 7080
<b>Marca:</b>	DELL
<b>Modelo:</b>	Optiplex 7080
<b>ID Elemento:</b>	8
<b>Número de serie:</b>	32478AA3T-G
<b>Número de inventario:</b>	54067045
<b>Área de adscripción:</b>	Edificio H , Cubículos de Investigadores
<b>Fecha de último mantenimiento:</b>	2025-01-15
<b>Fecha de mantenimiento:</b>	2025-02-06

**Descripción del mantenimiento:**  
Cambio de pasta térmica, reemplazo de memorias RAM, actualizada a 16GB, Sistema Operativo Actualizado a Windows 11, Puerto DisplayPort Reparado.

Figura 3. 28. Administración de Elementos. Exportación de Registro de Mantenimiento.

### 3.1.5 Usuarios y Roles

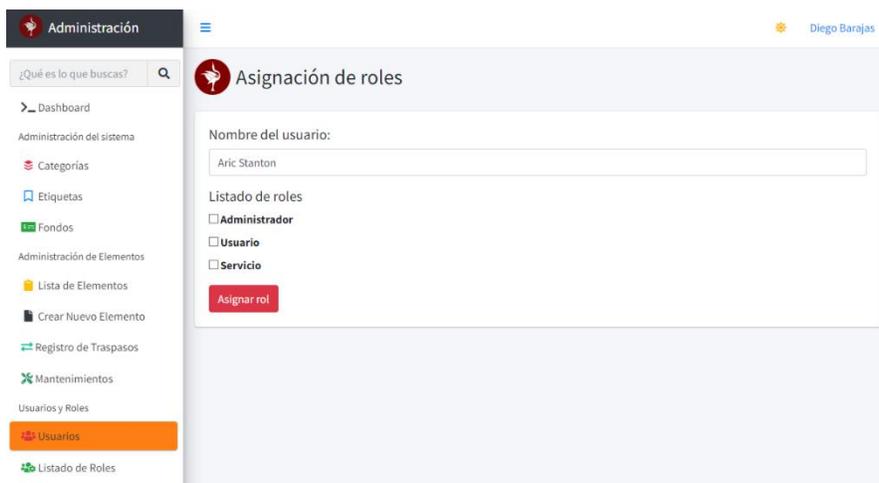
El apartado de Usuario y Roles del *Dashboard* se conforma de un conjunto de vistas dedicadas para la administración y asignación de roles, la opción de usuarios consiste de un listado de usuarios del sistema, el cual cuenta con una barra de búsqueda que utilizará como parámetros de búsqueda un nombre o correo electrónico, retornando todas las coincidencias de acuerdo a la entrada del usuario, este listado también es el medio por el cual podremos a la asignación de roles, en base a los roles creados para el sistema, se podrán asignar un conjunto de roles a un usuario, las Figuras 3.29 y 3.30 muestran ambas vistas.



The screenshot shows the 'Listado de Usuarios' page. On the left is a sidebar with navigation options. The main content area has a search bar and a table of users. The table has the following data:

ID	Nombre del Usuario	Correo Electrónico	Fecha de Creación de Usuario	
1	Diego Barajas	waos@gmail.com	2024-07-25 16:02:00	Editar
2	Aric Stanton	jaltenwerth@example.net	2024-07-25 16:02:00	Editar
3	Mrs. Maude Kovacek	qbeatty@example.com	2024-07-25 16:02:00	Editar
4	Karolann Kessler	diana.raynor@example.com	2024-07-25 16:02:00	Editar
5	Laney Stracke	duane28@example.net	2024-07-25 16:02:00	Editar
6	Winnifred Schuster DVM	sean16@example.org	2024-07-25 16:02:00	Editar
7	Bertha Cummerata	chauncey99@example.com	2024-07-25 16:02:00	Editar
8	Hanna Mertz	mills.anastacio@example.net	2024-07-25 16:02:00	Editar
9	Dr. Beulah D'Amore Sr.	penelope.cormier@example.org	2024-07-25 16:02:00	Editar
10	Dr. Dwight Mohr II	barney61@example.com	2024-07-25 16:02:00	Editar

Figura 3. 29. Usuarios y Roles. Listado de Usuarios.



The screenshot shows the 'Asignación de roles' page. It features a search bar, a text input for the user name (Aric Stanton), and a list of roles with checkboxes. The roles listed are Administrador, Usuario, and Servicio. An 'Asignar rol' button is located below the list.

Figura 3. 30. Usuarios y Roles. Asignación de Roles a Usuario.

La administración de roles se compone de un listado, dentro de este se podrán comprobar los roles creados en el sistema, Existen dos roles predefinidos, los cuales no pueden ser modificados, el primero de ellos es Administrador, este rol cuenta con todos los permisos del sistema, Usuario, por otra parte, solo cuenta con aquellos necesarios para modificar elementos y acceder al *Dashboard*. La figura 3.31 muestra este listado.

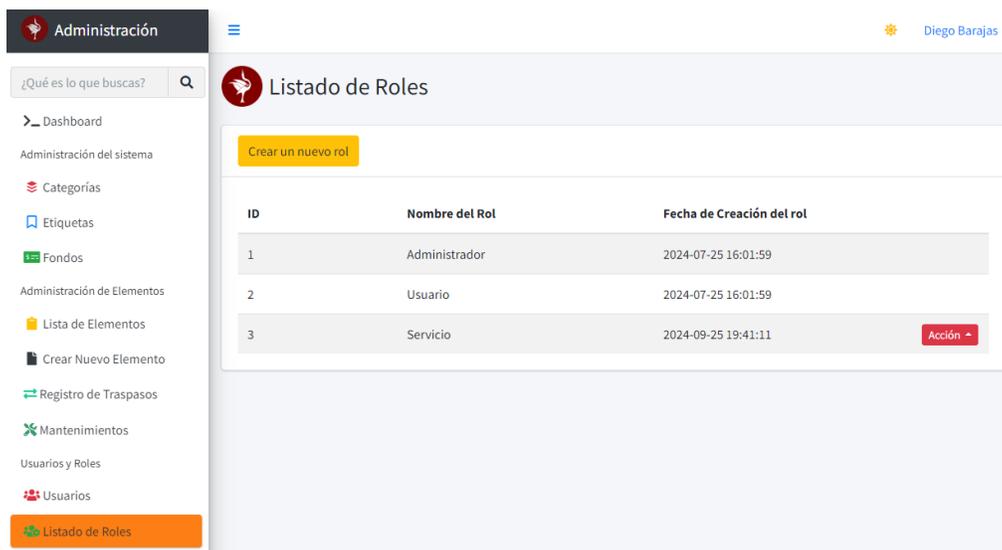


Figura 3. 31. Usuarios y Roles. Listado de Roles.

La creación y edición de roles del sistema se conforman de dos vistas donde se podrán ingresar el nombre del rol, además seleccionar un conjunto de permisos asignables a de acuerdo a las necesidades del usuario, como se aprecia en las Figuras 3.32 y 3.33.

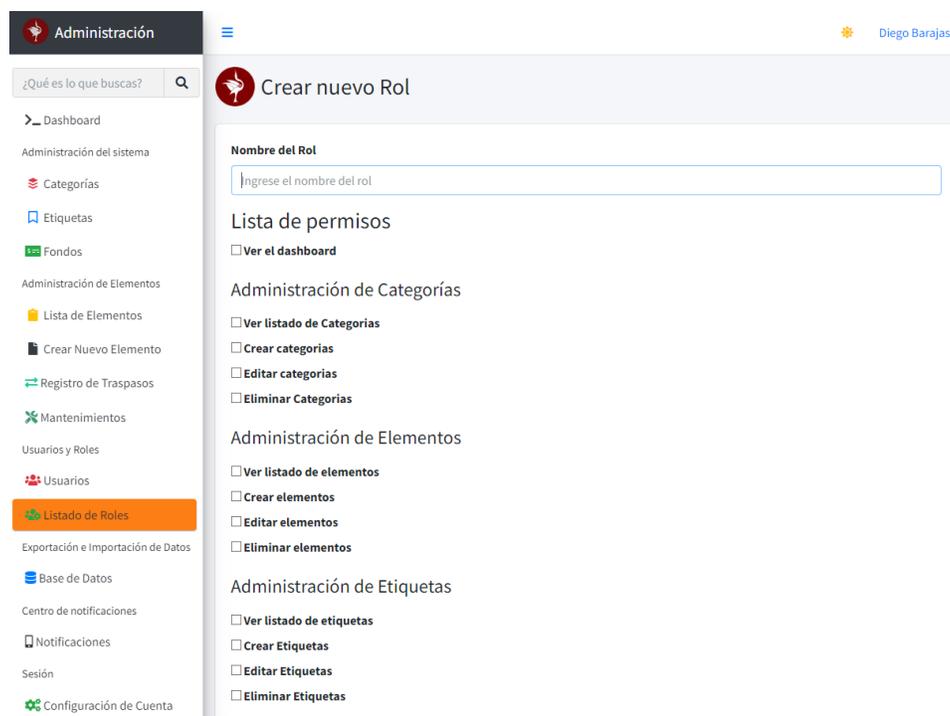


Figura 3. 32. Usuarios y Roles. Creación de Rol.

The image shows a web application interface for editing a role. On the left is a sidebar with a search bar and a list of navigation items. The main content area is titled 'Editar Rol' and contains a form for the role name and a section for permissions.

**Administración**

¿Qué es lo que buscas?

- Dashboard
- Administración del sistema
- Categorías
- Etiquetas
- Fondos
- Administración de Elementos
  - Lista de Elementos
  - Crear Nuevo Elemento
  - Registro de Traspasos
  - Mantenimientos
- Usuarios y Roles
  - Usuarios
  - Listado de Roles**
- Exportación e Importación de Datos
  - Base de Datos
- Centro de notificaciones
- Notificaciones
- Sesión
- Configuración de Cuenta

**Editar Rol**

Nombre del Rol

Servicio

Lista de permisos

Acceso al Panel de administración

- Ver el dashboard

Administración de Categorías

- Ver listado de Categorías
- Crear categorías
- Editar categorías
- Eliminar Categorías

Administración de Elementos

- Ver listado de elementos
- Crear elementos
- Editar elementos
- Eliminar elementos

Administración de Etiquetas

- Ver listado de etiquetas
- Crear Etiquetas

Figura 3. 33. Usuarios y Roles. Edición de Rol.

### 3.1.6 Exportación e Importación de Datos

La Exportación e Importación de Datos es un apartado del sistema dedicado al manejo y tratamiento de volúmenes de información grandes, particularmente, registros de elementos de inventario por medio de archivos de hojas de cálculo en formato XLSX, contando además con la posibilidad de exportar la base datos del sistema, en un archivo de formato *SQL*.

La Figura 3.34 muestra la vista principal de este apartado, esta vista es el medio por el cuál un usuario puede exportar tanto registros de elementos como la base de datos, además de acceder a la vista de importación de registros, las figuras 3.35 y 3.36 muestran los archivos exportados mediante esta vista.

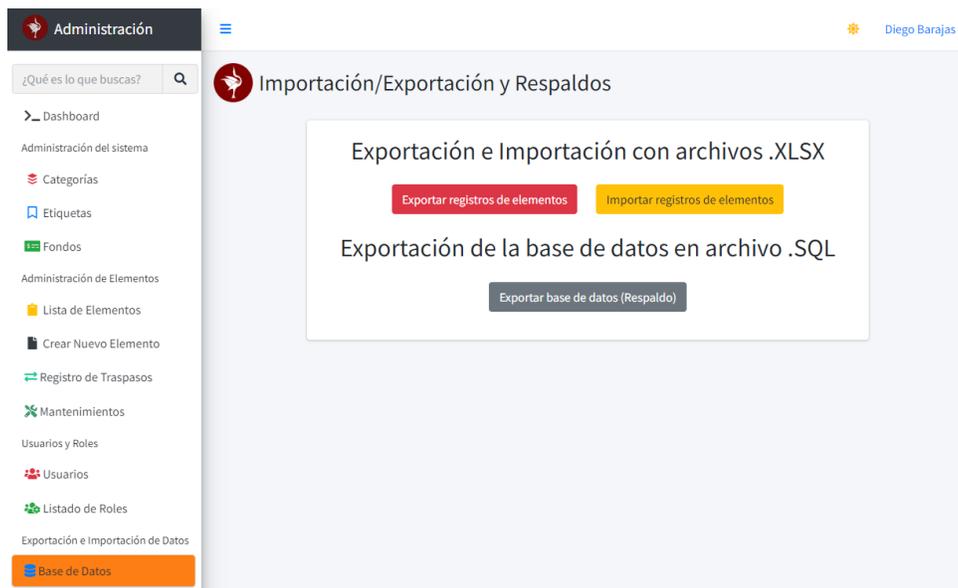


Figura 3. 34. Exportación e Importación de Datos. Vista principal.

```
1
2
3 CREATE TABLE `users` (
4   `id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
5   `name` varchar(255) NOT NULL,
6   `email` varchar(255) NOT NULL,
7   `email_verified_at` timestamp NULL DEFAULT NULL,
8   `password` varchar(255) NOT NULL,
9   `two_factor_secret` text DEFAULT NULL,
10  `two_factor_recovery_codes` text DEFAULT NULL,
11  `two_factor_confirmed_at` timestamp NULL DEFAULT NULL,
12  `remember_token` varchar(100) DEFAULT NULL,
13  `current_team_id` bigint(20) unsigned DEFAULT NULL,
14  `profile_photo_path` varchar(2048) DEFAULT NULL,
15  `created_at` timestamp NULL DEFAULT NULL,
16  `updated_at` timestamp NULL DEFAULT NULL,
17  PRIMARY KEY (`id`),
18  UNIQUE KEY `users_email_unique` (`email`)
19 ) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
20
21
22 INSERT INTO users (id, name, email, email_verified_at, password, two_factor_secret, two_factor_recovery_codes, two_factor_confirmed_at, remember_token) VALUES (1, 'John Doe', 'john.doe@example.com', NULL, 'password123', NULL, NULL, NULL, NULL);
23
24 INSERT INTO users (id, name, email, email_verified_at, password, two_factor_secret, two_factor_recovery_codes, two_factor_confirmed_at, remember_token) VALUES (2, 'Jane Smith', 'jane.smith@example.com', NULL, 'password456', NULL, NULL, NULL, NULL);
25
26 INSERT INTO users (id, name, email, email_verified_at, password, two_factor_secret, two_factor_recovery_codes, two_factor_confirmed_at, remember_token) VALUES (3, 'Bob Johnson', 'bob.johnson@example.com', NULL, 'password789', NULL, NULL, NULL, NULL);
27
28 INSERT INTO users (id, name, email, email_verified_at, password, two_factor_secret, two_factor_recovery_codes, two_factor_confirmed_at, remember_token) VALUES (4, 'Alice Brown', 'alice.brown@example.com', NULL, 'password012', NULL, NULL, NULL, NULL);
29
30 INSERT INTO users (id, name, email, email_verified_at, password, two_factor_secret, two_factor_recovery_codes, two_factor_confirmed_at, remember_token) VALUES (5, 'Charlie Davis', 'charlie.davis@example.com', NULL, 'password345', NULL, NULL, NULL, NULL);
31
```

Figura 3. 35. Exportación e Importación de Datos. Respaldo de la base de datos en formato SQL.

A	B	C	D	E	F	G	H
Nombre Elemento	Status de Inventario	Fecha de Adquisición	Fecha de Mantenimiento	Descripción	ID_Ubicación	ID_Categoría	ID_Usuario
1	Computadora HP Slimline	2 13/03/2011	07/08/2024	Pc Slimline con 4Gb de Ram y Disco Duro de 500GB	1	1	1
2	Organizador	2 13/03/2019	13/03/2024	Organizador de 3/4	1	2	1
3	Tablas de madera	2 13/03/2019	13/03/2024	Organizador de 3/4	1	3	1
4	Electrodo	2 13/03/2019	13/03/2024	Organizador de 3/4	1	4	1
5	Sierra Truper	2 13/03/2019	13/03/2024	Organizador de 3/4	1	5	1
6	Extintor32	2 13/03/2019	13/03/2024	Extintor 32	1	6	1
7	Test_GIF	2 12/07/2024	09/01/2025	Test	3	1	1
8	Dell Optiplex 7080	2 03/09/2024	06/02/2025	Computadora con 8GB de RAM, 512 GB de HDD, incluye monitor y teclado.	3	1	1
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							

Figura 3. 36. Exportación e Importación de Datos. Exportación de Registros de Elementos en formato XLSX.

La importación de datos consiste de una vista donde se podrá adjuntar un archivo XLSX con un formato definido para la importación de registro de elementos, este formato es idéntico al generado por la exportación en formato XLSX de registros del sistema, de acuerdo a las validaciones realizadas al archivo adjunto, la vista retornará un mensaje de éxito, o un mensaje de error, de acuerdo a la naturaleza de este, las figuras 3.37 a 3.40 muestran el comportamiento de esta vista.

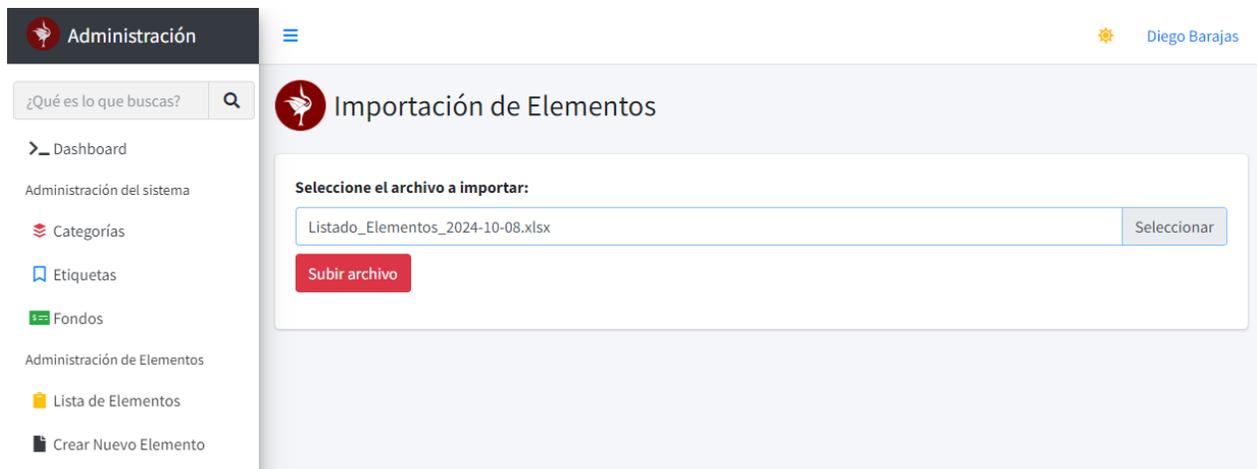


Figura 3. 37. Exportación e Importación de Datos. Vista de Importación de Registros.

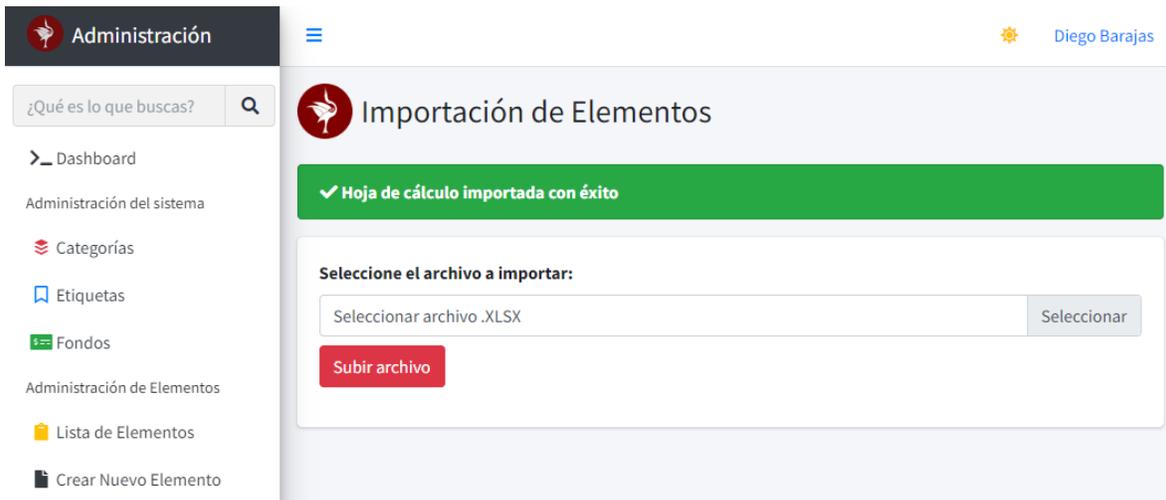


Figura 3. 38. Exportación e Importación de Datos. Importación Exitosa.

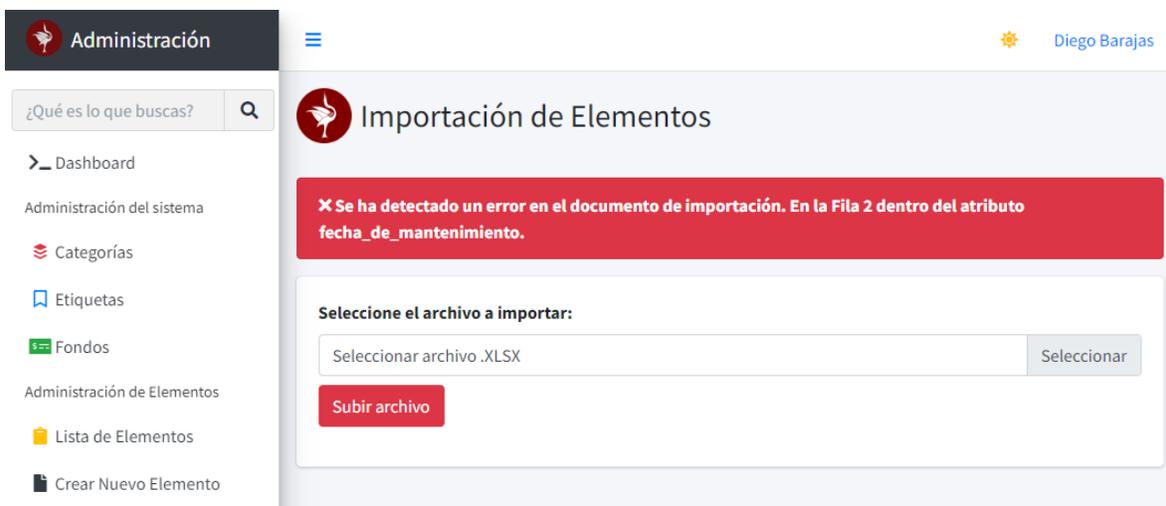


Figura 3. 39. Exportación e Importación de Datos. Error de Formato de documento.

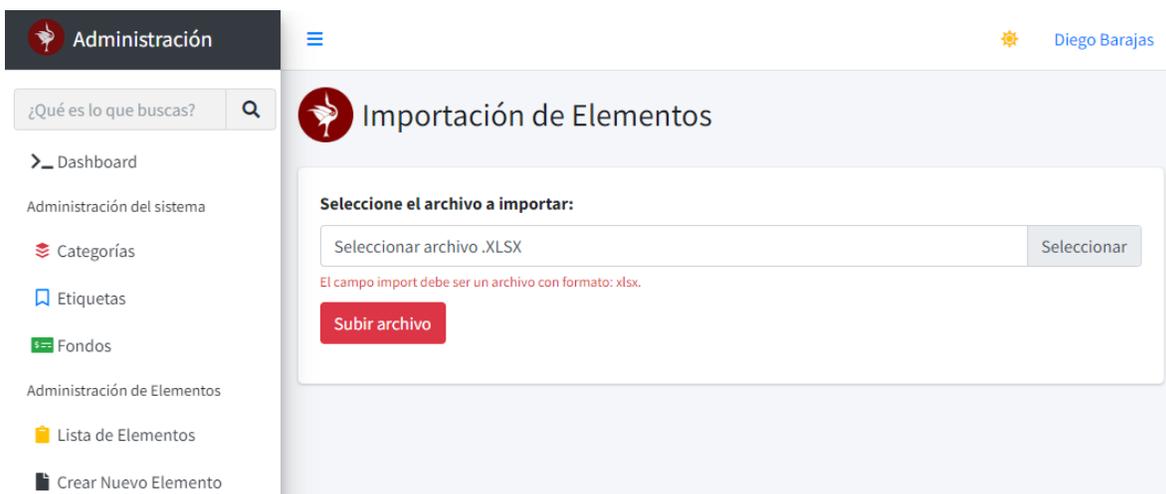


Figura 3. 40. Exportación e Importación de Datos. Error de Formato de Archivo.

El centro de notificaciones del sistema consiste de una vista donde el usuario podrá difundir un mensaje al grupo de notificaciones dedicado de *Telegram* para el Sistema de Inventario *Airone*, este mensaje tiene un límite de dos mil caracteres, al instante nel que se difunde este mensaje se desplegará también el usuario que difundió el mensaje, todo este proceso puede observarse en las figuras 3.41 a 3.43.

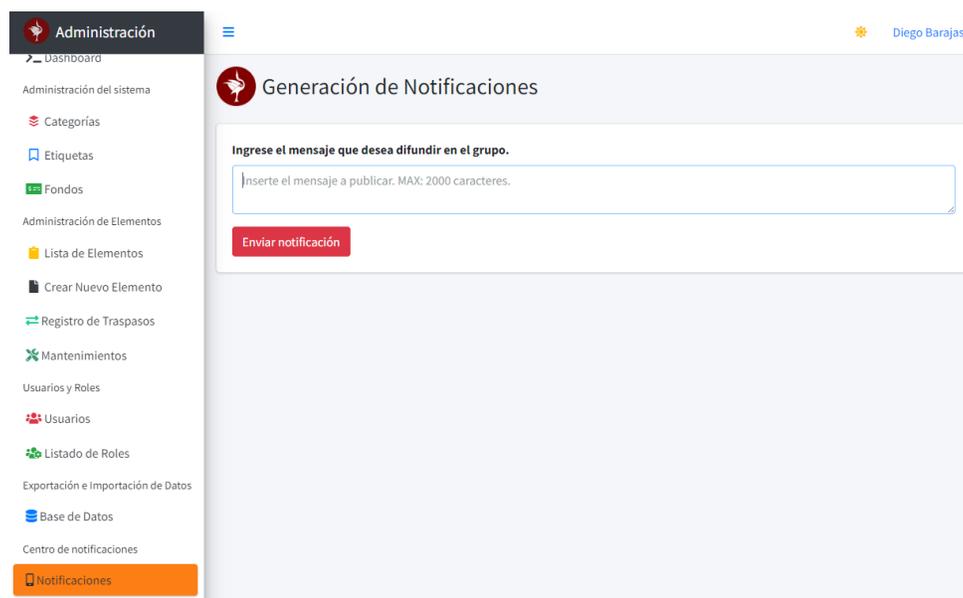


Figura 3. 41. Centro de Notificaciones. Vista principal.

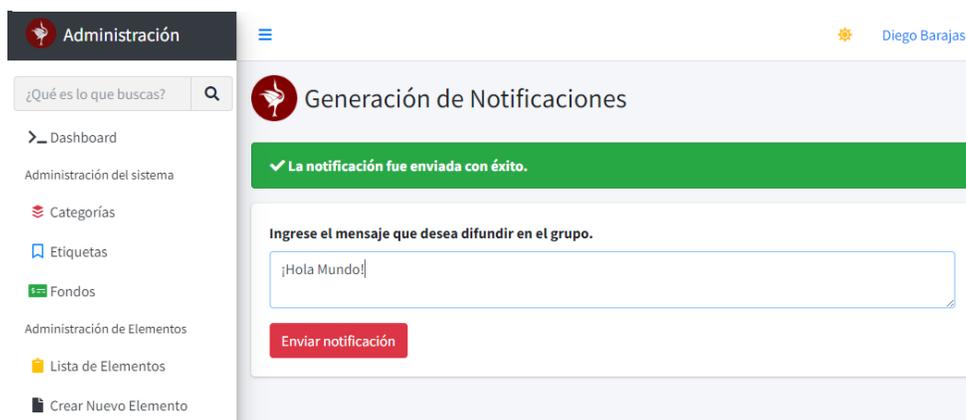


Figura 3. 42. Centro de Notificaciones. Envío Exitoso de Mensaje.

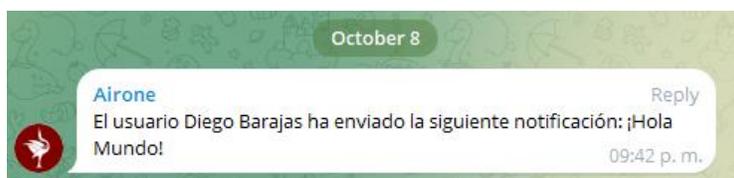


Figura 3. 43. Centro de Notificaciones. Mensaje difundido en Canal de Telegram.

## Capítulo 4. Manual de usuario

---

El Sistema de Inventario *Airone*, una vez implementado, debe de contar con la documentación pertinente para todos los usuarios finales. Esta documentación consistirá de un manual de usuario.

El propósito de este manual es proporcionar un documento de referencia con el cual los usuarios finales podrán guiarse a través de las distintas funcionalidades con las que cuenta el Sistema de Inventario *Airone*, además de describir un conjunto de soluciones a problemas comunes dentro del uso de este sistema. A continuación, se mostrará el manual de usuario generado:



*Airone*

Sistema de Inventario *Airone*  
Manual de Usuario

# I. Introducción al Manual de Usuario

---

## A. Objetivo del Sistema de Inventario Airone

El objetivo del Sistema de Inventario Airone es otorgar al personal administrativo y prestadores en ejercicio de servicio social del Área Académica de Ingeniería y Arquitectura un sistema web que facilite la gestión y consulta de información del inventario del Área Académica de Ingeniería y Arquitectura, cubriendo actividades como el registro y administración de elementos de inventario pertenecientes al área, soluciones para la administración de eventos, contabilidad, reportes de mantenimiento, traspasos y difusión de eventos por medio de Telegram.

## B. Requisitos

Para utilizar el Sistema de Inventario *Airone* se requiere de los siguientes requisitos:

- Navegador web
  - Microsoft Edge (última versión)
  - Google Chrome (última versión)
  - Firefox (última versión)
  - Safari (última versión)
  - Opera (última versión)
  - Cualquier navegador basado en Chromium en su última versión
- Sistema Operativo
  - Android (6 – última versión)
  - iOS (8 – última versión)
  - Windows (7 – 11)
  - Linux (Cualquier distribución basada en Debian, SUSE o RPM con un Desktop Environment habilitado, última versión)
  - MacOS (11 – última versión)
- Conexión a la misma red de área local (LAN), o en su defecto, una VPN, donde se encuentra alojado el servidor con el Sistema de Inventario Airone (De manera inalámbrica o alámbrica).

## II. Estructura del Sistema

---

### A. Login y Registro

#### i. Login

La interfaz de Login será la primera vista que el usuario tendrá del sistema, si el usuario ya está registrado en el sistema puede insertar sus credenciales en los recuadros correspondientes e iniciar sesión, esta puede ser una sesión mucho más duradera si se marca el recuadro “Mantener sesión activa”. Si el usuario desea crear una cuenta puede hacer “click” en el enlace “Registrarse”. En caso de que el usuario haya olvidado su contraseña puede hacer “click” en el enlace “¿Olvidó su contraseña?” se enviará un correo a la persona encargada de administrar esta parte del sistema, finalmente, si desea obtener información de contacto del creador del sistema puede hacer “click” en el enlace “Acerca de”.

The image shows two screenshots of the Airone login and password recovery interface. The left screenshot shows the login form with fields for 'Correo electrónico' (doejohn@gmail.com) and 'Contraseña' (masked with dots). Below the password field is a checkbox for 'Mantener sesión activa'. At the bottom, there are links for '¿Olvidó su contraseña?', 'Registrarse', and a button for 'INICIAR SESIÓN'. A red circle highlights these bottom elements, with red arrows pointing to them. The right screenshot shows the password recovery form with the same email field and a button for 'ENVIAR ENLACE PARA RESTABLECER CONTRASEÑA'.

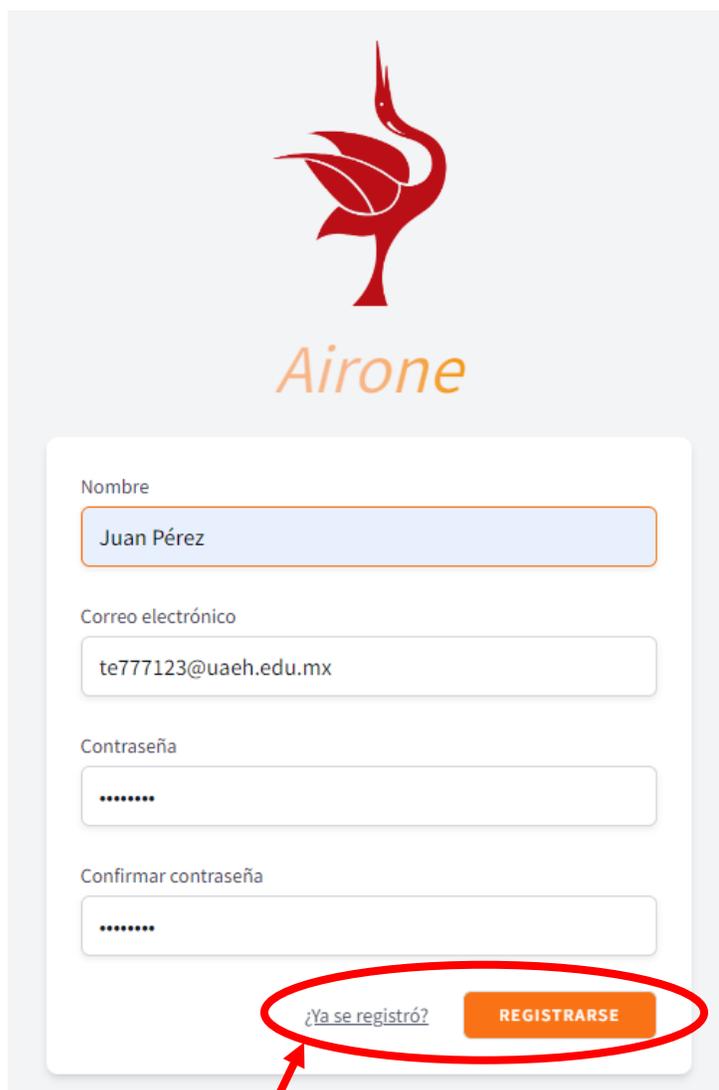
Botones de acción para recuperación de contraseña, registro e inicio de sesión.

Botón de acerca de.

## ii. Registro de usuario

Para crear un nuevo usuario dentro del sistema el usuario debe de proporcionar los siguientes datos:

- Nombre de usuario. **Una combinación de tu primer nombre y apellido paterno es idónea como nombre de usuario.**
- Correo Electrónico. **Utiliza de preferencia un correo institucional de la universidad.**
- Contraseña. **Debe contener mínimo 8 caracteres.**



Airone

Nombre  
Juan Pérez

Correo electrónico  
te777123@uaeh.edu.mx

Contraseña  
.....

Confirmar contraseña  
.....

[¿Ya se registró?](#) REGISTRARSE

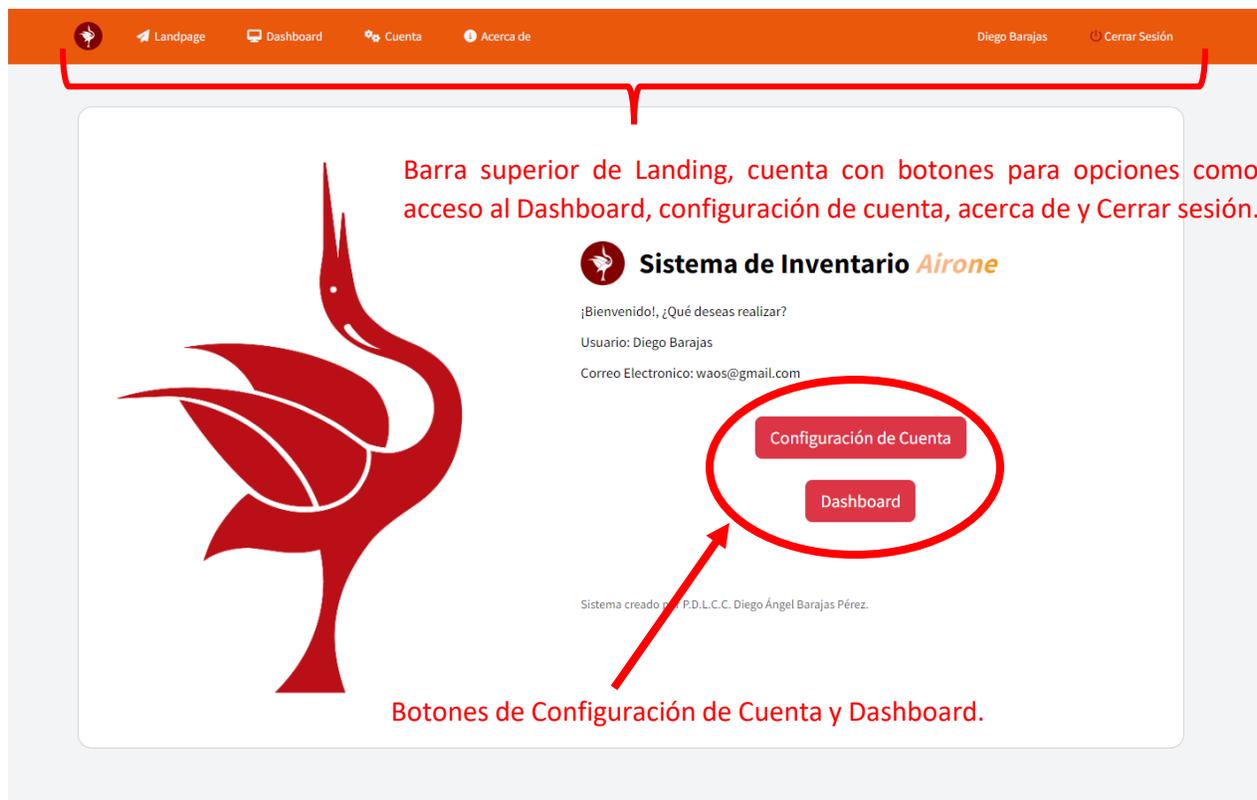
Botones de redirección a inicio de sesión y culminación de registro.

## B. Landpage

### i. Premisas de la Landpage

Una vez el usuario haya accedido con sus credenciales accederá al Landpage, en esta interfaz se podrá acceder a diversas funcionalidades, estas funcionalidades son:

- *Configuración de cuenta*
- *Dashboard*
- *Acerca de*
- *Cerrar Sesión*



## C. Configuración de cuenta

- En esta interfaz puede el usuario puede realizar varios cambios dentro de su cuenta, entre los cambios se encuentra:
- Actualización de datos de perfil. En este apartado se pueden actualizar datos como lo son el nombre de usuario y correo electrónico.
- Actualización de contraseña. Se requiere de tu contraseña actual para actualizarla.
- Autenticación de dos Factores. La autenticación de dos factores implica utilizar un dispositivo móvil con una aplicación autenticadora (Microsoft Authenticator o Google Authenticator) el cual generará una serie de 6 números que se requerirán en conjunto a tu contraseña cada vez que hagas Login. Necesario para aumentar la seguridad de la cuenta.
- Administración de sesiones. Si existen una o más sesiones abiertas en distintos navegadores o sesiones el usuario puede cerrarlas desde esta opción.
- Borrar cuenta. Si el usuario desea eliminar su cuenta se solicitará su contraseña. **ADVERTENCIA:** Si se elimina la cuenta de usuario también se eliminarán todos los registros del sistema que impliquen el ID del usuario, por ejemplo, registros de elemento, mantenimiento, traspasos, entre otros.

Información de perfil  
Actualice la información de su cuenta y la dirección de correo electrónico.

Nombre  
Diego Barajas

Correo electrónico  
waos@gmail.com

Casillas para la actualización de datos de la cuenta.

GUARDAR

Autenticación de dos factores  
Agregue seguridad adicional a su cuenta mediante la autenticación de dos factores.

No ha habilitado la autenticación de dos factores.

Cuando la autenticación de dos factores esté habilitada, le pediremos un token aleatorio seguro durante la autenticación. Puede recuperar este token desde la aplicación Google Authenticator de su teléfono.

HABILITAR



Aplicaciones como Microsoft Authenticator o Google Authenticator pueden trabajar de manera efectiva con la autenticación de dos factores del Sistema de Inventario Airone.

#### Autenticación de dos factores

Agregue seguridad adicional a su cuenta mediante la autenticación de dos factores.

##### Termine de habilitar la autenticación de dos factores.

Cuando la autenticación de dos factores esté habilitada, le pediremos un token aleatorio seguro durante la autenticación. Puede recuperar este token desde la aplicación Google Authenticator de su teléfono.

Para terminar de habilitar la autenticación de dos factores, escanee el siguiente código QR usando la aplicación de autenticación de su teléfono o ingrese la clave de configuración y proporcione el código OTP generado.



Código QR para enlazar aplicación autenticadora.

Casilla para la entrada inicial de código de 6 dígitos generado por aplicación autenticadora.

Clave de configuración

Código

CONFIRMAR

CANCELAR

#### Borrar cuenta

Eliminar su cuenta de forma permanente.

Una vez que se elimine su cuenta, todos sus recursos y datos se eliminarán de forma permanente. Antes de borrar su cuenta, por favor descargue cualquier dato o información que desee conservar.

BORRAR CUENTA

Botón de eliminación de cuenta, requiere de contraseña para confirmar acción.

# III. Dashboard del Sistema

## A. Premisas del Dashboard

La interfaz del Dashboard consiste de los siguientes componentes:

- *Switch Modo Oscuro*. Es un botón con un icono en forma de sol/luna, su función es cambiar el tema del Dashboard, este cambio es persistente en todas las vistas del Dashboard.
- *Plegable*. Su función es minimizar o desplegar el Sidebar.
- *Sidebar*. El Sidebar es el área donde se pueden acceder a distintas partes del sistema por medio de los elementos del Sidebar.
- *Cuerpo de la interfaz*: Se desplegará el contenido de la opción/vista seleccionada en este apartado de la página.



## B. Landpage

La interfaz principal de la vista contiene un panel de estadísticas en la cual, se muestran datos como registros creados en total/mensuales, traspasos realizados en total/mensualmente. El segundo apartado de esta vista es un calendario de eventos donde se desplegarán todos los eventos arreglados, si un usuario tiene los permisos para modificar el calendario podrá arrastrar, crear y eliminar eventos dentro del calendario.

Dashboard

Estadísticas Globales

30 Elementos registrados en total.	12 Usuarios registrados en total.	2 Mantenimientos realizados.	1 Traspasos realizados.
---------------------------------------	--------------------------------------	---------------------------------	----------------------------

Estadísticas Mensuales

30 Elementos registrados en este mes.	2 Mantenimientos realizados este mes.	1 Traspasos realizados este mes.
--	--	-------------------------------------

Calendario de actividades

junio de 2024

lun	mar	mié	jue	vie	sáb	dom
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

● 0 Test

Evento muy largo.

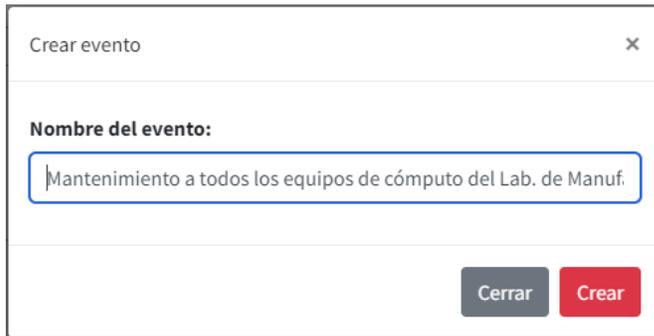
Botón para desplazarse un mes atrás en el calendario, el botón "Hoy" redirigirá el calendario a la fecha actual.

Botón para desplazarse un mes adelante en el calendario.

## C. Administración de Eventos

### i. Creación de eventos

Para crear un evento debes hacer “click” en la fecha donde deseas crear el evento, se desplegará una ventana donde podrás insertar el nombre del evento.

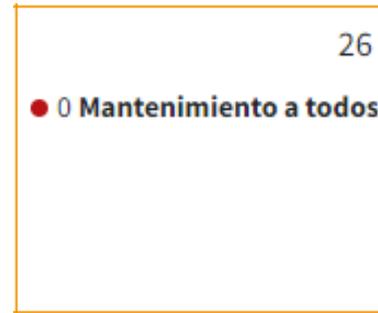


Crear evento

Nombre del evento:

Mantenimiento a todos los equipos de cómputo del Lab. de Manuf.

Cerrar Crear

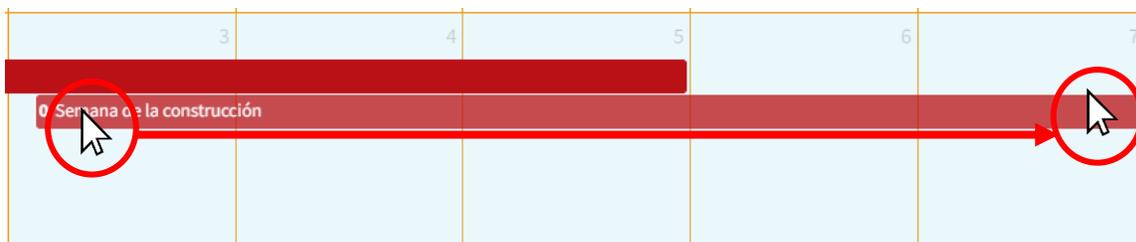


Si deseas crear un evento con un periodo de tiempo mayor a un día debes hacer mantener “click” en fecha donde inicial o final de tu evento, a continuación, arrastra el ratón mientras se mantiene el “click” y suéltalo hasta donde desees que sea el periodo del evento, finalmente agrega el nombre del evento.



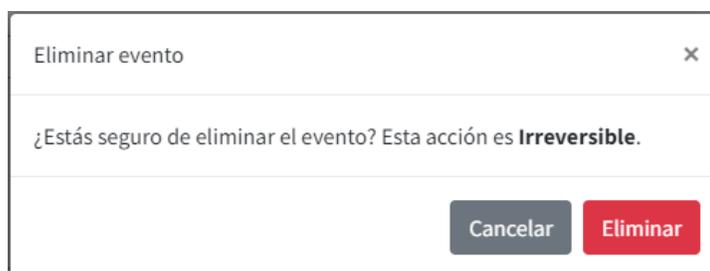
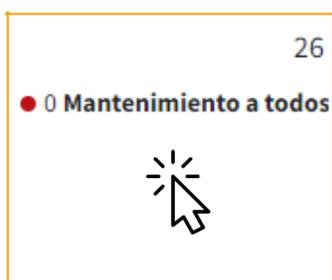
## ii. Actualización de eventos

Si deseas actualizar un evento del calendario basta con arrastrar el título al día donde se desea cambiar la fecha.



## iii. Eliminación de eventos

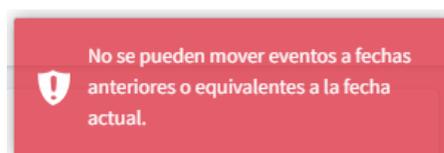
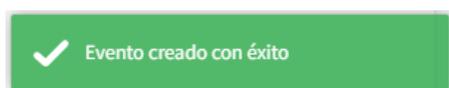
Para eliminar un evento basta con hacer "click" en el título del evento, se desplegará un dialogo para confirmar la acción.



## iv. Consideraciones del calendario de Eventos

No se pueden crear eventos o actualizar eventos en una fecha anterior o equivalente a la fecha actual. Cuando la fecha del evento sea equivalente a la fecha actual se enviará una notificación al canal de [Telegram](#) del sistema, todas las notificaciones se enviarán a las **11:00 AM**.

Al momento de crear, actualizar o eliminar un evento se te notificará en la parte superior derecha si la acción fue realizada con éxito u ocurrió algún error con la acción.



## D. Administración del Sistema

La administración del sistema es conjunto de vistas donde el usuario puede crear, editar y eliminar registros de tres tipos, los cuales son categorías, etiquetas y fondos, estos registros serán utilizados en el registro de elementos de inventario.

Las vistas de administración del sistema se conforman de las siguientes vistas:

### *i. Listado de Registros*

Es la primera página que se desplegará al momento de administrar uno de los tres tipos de registros; se conforma de un botón de creación, listado de registros y botón de acción.

- Listado de registros. Es el área donde se desplegarán todos los registros del del respectivo tipo seleccionado.
- Botón de creación. Este botón redirigirá a la vista de creación de registro.
- Botón de acción. Este botón desplegará opciones para editar o eliminar el registro.

ID	Nombre de la Etiqueta	
1	dolore	Editar Eliminar
2	aliquid	Acción ▾
3	qui	Acción ▾
4	sit	Acción ▾
5	dignissimos	Acción ▾
6	deleniti	Acción ▾
7	voluptates	Acción ▾
8	voluptas	Acción ▾

### *ii. Creación y edición de Registros del Sistema*

En esta interfaz el usuario puede nombrar su nuevo registro y nombrarlo, al momento de crear su registro será redirigido a la interfaz de edición. **Nota: Si el registro es de tipo etiqueta aquí también se puede editar el color de la etiqueta.**

Nombre de la categoría

Slug

**Crear categoría** ← Botón de creación registro.

✓ La categoría se ha creado con éxito

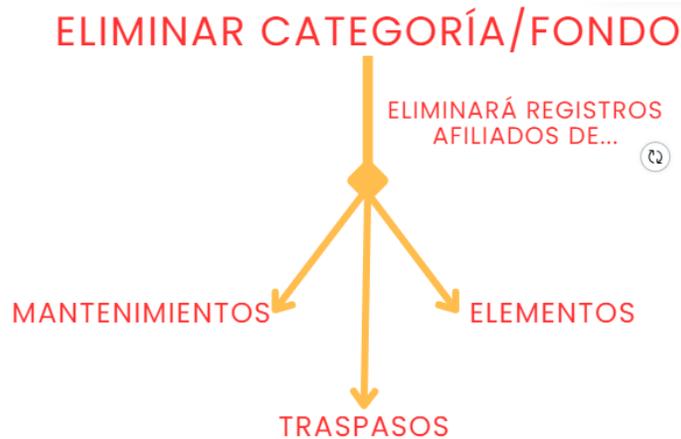
Nombre de la categoría

Slug

**Actualizar categoría** ← Botón de actualización de registro.

### iii. Eliminación de Registros del Sistema

En la interfaz índice de la administración de cada tipo de registro en el recuadro acción esta la opción “Eliminar”, al hacer “click” en esta opción se desplegará una advertencia junto al botón definitivo para eliminar el registro deseado. **Nota: Si se elimina un registro de tipo categoría o fondo se eliminarán todos los registros de elementos, traspasos y mantenimientos que hagan referencia a este registro eliminado, esta acción es PERMANENTE. Si se elimina una etiqueta esta será removida de todos los elementos que hacen uso de ella.**



Advertencia. ×

---

¿Estás seguro de eliminar la categoría Equipo de manufactura? Esta acción también eliminará todos los elementos registrados pertenecientes a esta categoría.

Cancelar
Eliminar

## E. Administración de Elementos

La administración de elementos es el medio por el cual el usuario puede buscar, revisar detalles, crear, editar y eliminar registros de elementos. En este apartado del sistema también es posible generar un registro de traspasos y mantenimientos. La administración de elementos consiste de las siguientes vistas:

### i. Lista de Elementos

La lista de elementos consiste de una interfaz con un buscador y listado de elementos; el buscador es una herramienta que permite realizar la recuperación de elementos basándose en la combinación de las siguientes condiciones:

- Nombre del elemento
- Categoría
- Fondo
- Etiqueta
- Edificio
- Ubicación
- Status

Buscador

Nombre del Elemento

Ingrese el nombre del elemento a buscar

Categoría: Selecciona una categoría

Fondo: Selecciona un fondo

Etiqueta: Selecciona una etiqueta

Edificio: Selecciona un edificio

Ubicación: Selecciona una ubicación

Status: N/A

Barra de búsqueda de registros de elementos.

Filtros para los registros de elementos.

Esta vista es también el medio por el cual se puede acceder a opciones como editar, eliminar, examinar detalles de cualquier elemento (Haciendo “click” al nombre del elemento) y generar un reporte de mantenimiento.

Nombre del elemento con enlace a descripción.

ID	Nombre del Elemento	Edificio	Ubicación	Fondo	Categoría	Status	Fecha de Registro	Etiquetas
Acción	C1 test6	Edificio H	Areas de Apoyo	Federal Genérico	Computación	Alta	2024-06-20	deleniti
	test3	Edificio H	Laboratorios	Federal Genérico	Computación	Alta	2024-06-19	dignissimos

Acciones del elemento.

## ii. Crear Nuevo Registro de Elemento

Esta vista consiste de un formulario en el cual el usuario podrá registrar los datos de un nuevo elemento. Cada registro de elementos contiene el siguiente conjunto de atributos:

- Nombre del Elemento
- Categoría y Fondo
- Etiquetas
- Edificio y Ubicación
- Fecha de Adquisición
- Fecha de Último Mantenimiento
- Descripción
- Estado de Inventario
- Imagen del elemento
- **Atributos Adicionales**

**Nombre del elemento**

**Slug**

**Categoría**

**Fondo**

**Etiquetas**

dolore  aliquid  qui  sit  dignissimos  deleniti  voluptates  voluptas

**Edificio**

**Ubicación**

**Fecha de Adquisición**

**Fecha de último mantenimiento**

**Descripción**

**Imagen del Elemento**



Sin archivos seleccionados

*La imagen debe mostrar las principales características del elemento.*

**Estado de Inventario**

Baja  Alta

El Sistema de Inventario Airone cuenta con 6 categorías predefinidas:

- 1) **Computación**
- 2) **Inmobiliario**
- 3) **Infraestructura**
- 4) **Equipo de Laboratorio**
- 5) **Maquinaria**
- 6) **Equipo de Seguridad**

El formulario de registro de elemento desplegará un conjunto de atributos adicionales de acuerdo a la categoría seleccionada, los atributos adicionales de acuerdo a cada categoría son:

**1) Computación.**

- a) Marca
- b) Modelo
- c) Número de Inventario
- d) Número de serie

**2) Inmobiliario.**

- a) Número de Inventario
- b) Número de serie
- c) Color
- d) Material
- e) Dimensiones
- f) Estantes
- g) Puertas

**3) Infraestructura.**

- a) Color
- b) Material
- c) Dimensiones
- d) Cantidad

**4) Equipo de Laboratorio.**

- a) Marca
- b) Modelo
- c) Número de Inventario
- d) Número de serie

**5) Maquinaria.**

- a) Marca
- b) Modelo
- c) Número de Inventario
- d) Número de serie

**6) Equipo de Seguridad.**

- a) Marca
- b) Modelo
- c) Número de Inventario
- d) Tipo de Extintor (Opcional)
- e) Capacidad (Opcional)

### iii. Registro de Traspasos

El registro de traspasos es una vista donde se almacenan todos los registros de traspasos. Un registro de traspaso se generará de manera automática en el instante que un elemento sea editado, particularmente en los atributos “Edificio” y/o “Ubicación”.

**Edificio**
**Ubicación**

Seleccione un edificio

←

→

Seleccione una ubicación

Ambos atributos al ser alterados generarán un registro de traspaso.

ID	Elemento	Ubicación Anterior	Edificio Anterior	Nueva Ubicación	Nuevo Edificio	Fecha Traspaso
4	PC HP	Edificio F	Auditorio	Edificio AAlyA	Auditorio	2024-06-19

### iv. Mantenimientos

Un registro de mantenimiento es el medio donde se puede actualizar la fecha de mantenimiento de un elemento, un reporte de mantenimiento consiste de añadir la nueva fecha de mantenimiento del elemento en conjunto a una descripción del mantenimiento. Un registro de reporte consta de los datos del elemento, nueva fecha de mantenimiento y una descripción del mantenimiento, puede imprimirse este reporte en un archivo de formato PDF mediante esta vista.

ID Registro	ID Elemento	Elemento	Fecha de Mantenimiento Anterior	Fecha de Mantenimiento	Generar Documento
5	1	Computadora HP Slimline	2024-06-05	2024-08-08	<div style="background-color: #e74c3c; color: white; padding: 5px; border-radius: 3px;">  Imprimir         </div>
6	10	Test móvil	2024-06-21	2024-09-12	<div style="background-color: #e74c3c; color: white; padding: 5px; border-radius: 3px;">  Imprimir         </div>



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO  
Pachuca, Hidalgo a 25 de febrero de 2025.

#### Registro de mantenimiento de Equipo.

##### Descripción del elemento.

<b>Elemento/Equipo:</b>	Computadora HP Slimline
<b>Marca:</b>	HP
<b>Modelo:</b>	HP Pavilion Slimline s5000 series
<b>ID Elemento:</b>	1
<b>Número de serie:</b>	MXX0450Z5Z
<b>Número de inventario:</b>	00210487
<b>Área de adscripción:</b>	Edificio A , Oficinas de Coordinadores
<b>Fecha de último mantenimiento:</b>	2024-03-13
<b>Fecha de mantenimiento:</b>	2024-08-07

##### Descripción del mantenimiento:

Pasta térmica

## F. Usuarios y Roles

El apartado de usuarios y roles es el medio por el cual el usuario puede crear, editar, eliminar y asignar roles a usuarios, los roles permiten a un administrador o un usuario con los permisos necesarios controlar las acciones que puede realizar distintos usuarios dentro del Sistema de Inventario Airone.

### *i. Listado de Usuarios*

El listado de usuarios contiene una barra de búsqueda para filtrar por medio de nombres y correos los usuarios registrados, el botón editar redirige a la vista de asignación de roles a ese usuario en particular.

ID	Nombre del Usuario	Correo Electrónico	Fecha de Creación de Usuario	
1	Diego Barajas	waos@gmail.com	2024-06-14 12:13:19	 <b>Botón de edición de roles a usuario.</b> 
2	Prof. Louie Kuhic DDS	hjast@example.org	2024-06-14 12:13:19	
3	Dr. Samson Collier	araceli93@example.org	2024-06-14 12:13:20	
4	General Kivalis IV	precious.steuber@example.org	2024-06-14 12:13:20	

Nombre del usuario:

Listado de roles

- Administrador**  **Roles asignables al usuario.**
- Usuario
- Prestador Servicio Social



## ii. Listado de Roles

El listado de roles es la vista por la cual se pueden comprobar todos los roles creados, además de también crearse, editar y eliminar, si se desea crear un rol nuevo para el sistema, es necesario entrar a esta vista.

Crear un nuevo rol

ID	Nombre del Rol	Fecha de Creación del rol
1	Administrador	2024-06-14 12:13:18
2	Usuario	2024-06-14 12:13:18
3	Prestador Servicio Social	2024-06-17 21:35:40

Editar  
Eliminar  
Acción ^

La lista de permisos consiste de todos los permisos disponibles en el sistema excluyendo la Landpage, si se desea crear roles para acciones puntuales dentro del sistema, este es la función idónea. **Nota. Si se elimina un rol este será removido de todos los usuarios que lo tengan, no eliminará al usuario.**

**Nombre del Rol**

**Lista de permisos**

- Ver el dashboard.
- Administración de Categorías**
  - Ver listado de Categorías.
  - Crear categorías.
  - Editar categorías.
  - Eliminar Categorías.
- Administración de Elementos**
  - Ver listado de elementos.
  - Crear elementos.
  - Editar elementos.
  - Eliminar elementos.
- Administración de Etiquetas**
  - Ver listado de etiquetas.
  - Crear Etiquetas.
  - Editar Etiquetas.

Campo para insertar el nombre del Rol.

Permisos con los que puede contar el rol.

## G. Exportación e Importación de Datos

Este apartado es dedicado a la exportación de datos en formato .XLSX (Registros de elementos) y .SQL (Respaldo de tablas de la base de datos) así como importación de registros de elementos por medio de archivos .XLSX. **Nota.** Se recomienda generar 1 respaldo de la base de datos en periodos de 2 a 4 semanas o cuando se realicen actividades que impliquen cambios sustanciales a la base de datos del sistema, por ejemplo, la importación de una gran cantidad de elementos o eliminar registros de tipo categoría/fondo.

### Exportación e Importación con archivos .XLSX

Exportar registros de elementosImportar registros de elementos

### Exportación de la base de datos en archivo .SQL

Exportar base de datos (Respaldo)

### i. Importación de registros de elementos

La importación de registros de elementos consiste de importar en la base de datos del sistema registros de la tabla elementos y/o registros de atributos adicionales a partir de un archivo .XLSX con un formato predefinido, este formato puede ser recuperado realizando una exportación de registros de elementos en .XLSX.

**Seleccione el archivo a importar:**

Seleccionar

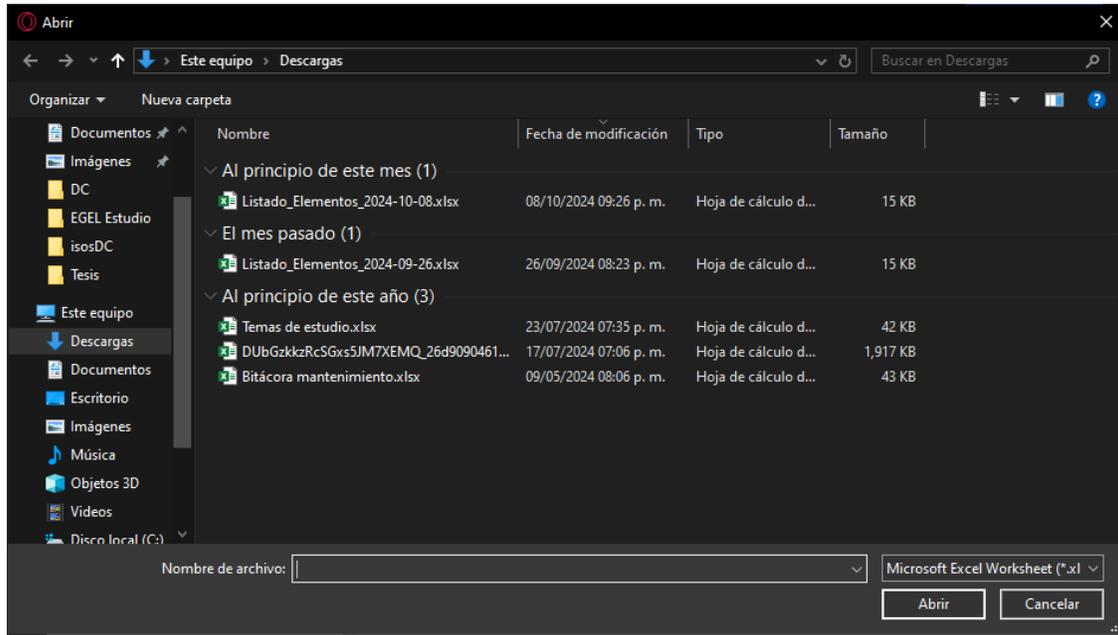
Subir archivo

**Seleccione el archivo a importar:**

Seleccionar

Subir archivo

El buscador mostrará únicamente archivos en formato .XLSX, como se observa en la siguiente imagen:



El archivo .XLSX para importar registros se conforma de 7 hojas de Excel, 1 para registros de elementos y 6 para registros de atributos adicionales (Para elementos que pertenecen a 1 de las 6 categorías con atributos adicionales).



Hoja 1: Elementos. La hoja “Elementos” contiene espacios para los siguientes atributos:

- Nombre\_Elemento. Se incluye el nombre del elemento a registrar.
- Status\_de\_Inventario. Este espacio únicamente puede tener 2 valores, 1 si el elemento está de baja, 2 si el elemento está de alta.
- Fecha\_de\_Adquisición. Se incluye la fecha de adquisición del elemento, esta fecha debe estar en formato “DD/MM/AAAA”.
- Fecha\_de\_Mantenimiento. Se incluye la fecha de mantenimiento del elemento, esta fecha debe estar en formato “DD/MM/AAAA”.
- Descripción: Se incluye la descripción del elemento.

Nombre_Elemento	Status_de_Inventario	Fecha_de_Adquisición	Fecha_de_Mantenimiento	Descripción
Computadora HP Slimline	2	13/03/2011	08/08/2024	Pc Slimline con 4Gb de Ram y Disco Duro de 500GB

ID_Ubicación	ID_Categoría	ID_Usuario	ID_Fondo	ID_Edificio
1	1	1	1	1

- **ID\_Ubicación:** En este apartado se incluye el ID de la ubicación del elemento. La disposición de las ubicaciones son las siguientes:

<b>ID</b>	<b>Ubicación</b>
1	Oficinas de Coordinadores
2	Aulas
3	Cubículos de Investigadores
4	Auditorio
5	Jefatura
6	Áreas de Apoyo
7	Laboratorios
8	Bodegas
9	Laboratorios de Cómputo

- **ID\_Categoría:** En este apartado se incluye el ID de la categoría del elemento. La disposición de las categorías son las siguientes:

<b>ID</b>	<b>Categoría</b>
1	Computación
2	Inmobiliario
3	Infraestructura
4	Equipo de Laboratorio
5	Maquinaria
6	Equipo de Seguridad

Si se desea agregar una categoría ajena a 1 de estas 6 es posible, únicamente se debe de obtener el identificador de la categoría e insertarlo en el registro.

- **ID\_Usuario:** En este apartado se incluye el ID de usuario que registró el elemento, este dato puede consultarse en la lista de usuarios.

- **ID\_Fondo:** En este apartado se incluye el ID del fondo del elemento. La disposición de los fondos son las siguientes:

<b>ID</b>	<b>Fondo</b>
2	Federal Genérico
3	Estatad Genérico
4	PAO
5	PROMEPI/PRODEP
6	PROFECXE/PIFI
7	CONAHCYT

Si se desea agregar un fondo ajeno a 1 de estos 7 es posible, únicamente se debe de obtener el identificador del fondo e insertarlo en el registro respectivo.

- **ID\_Edificio:** En este apartado se incluye el ID del edificio del elemento. La disposición de los edificios son las siguientes:

<b>ID</b>	<b>Edificio</b>
1	Edificio A
2	Edificio F
3	Edificio H
4	Edificio AAlyA

Hoja 2 – 6: Atributos adicionales. Las hojas de atributos adicionales contienen espacios para insertar los respectivos datos. El apartado ID\_Elemento debe contener el ID de un elemento ya creado perteneciente a una categoría con atributos adicionales. Si se desea añadir de esta manera registros de atributos adicionales se debe tener en cuenta los siguientes puntos:

- El elemento debe de estar registrado.
- Si se desea añadir el registro de un elemento nuevo a ser importado por medio del archivo de importación este el ID de elemento debe ser el ID seguido al último registro, o el que indique el recuadro de error, si es que se ha realizado una importación de elementos y registros de atributos adicionales sin éxito, como se muestra en la siguiente imagen.

**X Se ha detectado un error en el documento de importación. En la Fila 2 dentro del atributo id\_elemento. No han sido importados los registros de elementos. El Contador de ID de elemento es: 66, por lo tanto, tu siguiente registro de atributos adicionales debe iniciar con el ID 67.**

### ***Importación. Aspectos a considerar.***

- Los registros de elementos importados por este método no contendrán etiquetas o imagen del elemento de manera predefinida, pueden ser agregados de manera convencional en el sistema.
- Es posible importar únicamente registros de elementos o de atributos adicionales (Únicamente si son a elementos ya existentes).
- La importación de elementos rechazará cualquier otro archivo de hojas de cálculo que no sea de formato .XLSX.
- La importación de elementos rechazará cualquier fecha que no se encuentre formateada en DD/MM/AAAA.

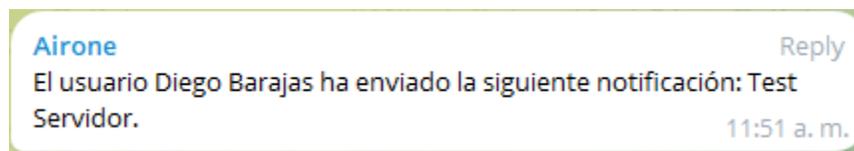
## H. Notificaciones

Si se desea transmitir un mensaje por medio del canal de notificaciones de Telegram sin necesidad de generar un evento es posible por medio de esta opción, se enviará la notificación de manera inmediata.

Ingrese el mensaje que desea difundir en el grupo.

Enviar notificación

Al momento de enviar una notificación por este medio el mensaje será generado de la siguiente manera:



## I. Sesión

Este último apartado del Dashboard cuenta con 3 opciones:

- Configuración de cuenta. Esta opción redirige a la vista de configuración de cuenta.
- Cerrar Sesión. Esta opción cierra la sesión actual y redirige a la vista de Login.
- Acerca de. Esta opción redirige a la vista de acerca de.

 Configuración de Cuenta

 Cerrar Sesión

 Acerca de

## IV. Contacto

---

Sistema Creado por Diego Ángel Barajas Pérez

Correo Electrónico: [ba439879@uaeh.edu.mx](mailto:ba439879@uaeh.edu.mx)

Correo Alternativo: [zhalobarajas@gmail.com](mailto:zhalobarajas@gmail.com)

Número telefónico: 771-168-48-04

LinkedIn: [Diego Ángel Barajas Pérez](#)



*Airones*

## Capítulo 5. Validación/Pruebas de usabilidad

Con el propósito de evaluar la usabilidad, facilidad de uso y diseño del Sistema de Inventario *Airone* se ha propuesto una prueba de usabilidad, esta prueba emplea la escala de *Likert* de cinco puntos. La escala de *Likert* permite medir el grado de conformidad o satisfacción de una persona hacia un producto de, en este caso, *Software*, por medio de una oración la cual puede relatar un encuestado de manera afirmativa o negativa.

La prueba cuenta con 11 preguntas de elaboración propia las cuales siguen una escala de *Likert*, este tipo de escala permite definir una escala la cual puede describir una valoración del 0 al 10, o en algunas variantes, 0 a 5 a un conjunto de preguntas dadas (Landaluce C., 2024). El conjunto de preguntas elaboradas evaluará los distintos aspectos del Sistema de Inventario *Airone*. Esta prueba fue respondida por 16 usuarios, de los cuales el 37.6% usarán el sistema de Inventario *Airone* de forma activa. Todos los encuestados cuentan con una noción clara del Sistema de Inventario *Airone*, así como un tiempo de uso considerable con este.

Las Figuras 5.1. a 5.11 presentarán respectivamente una gráfica de pastel con las respuestas obtenidas por cada pregunta de la prueba:

1.- Fue simple usar el Sistema de Inventario *Airone*.

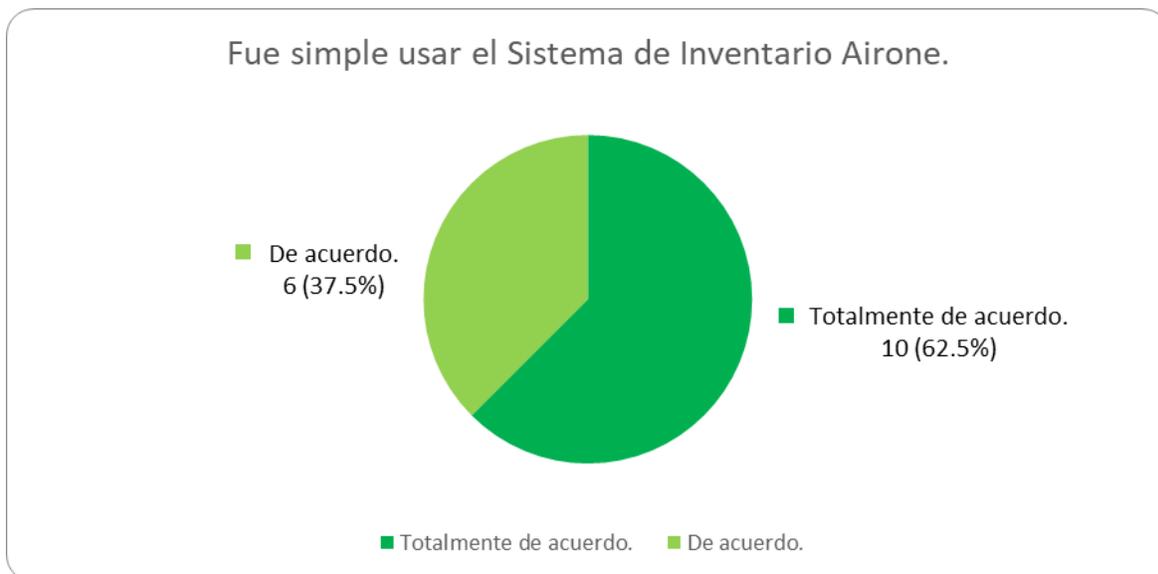
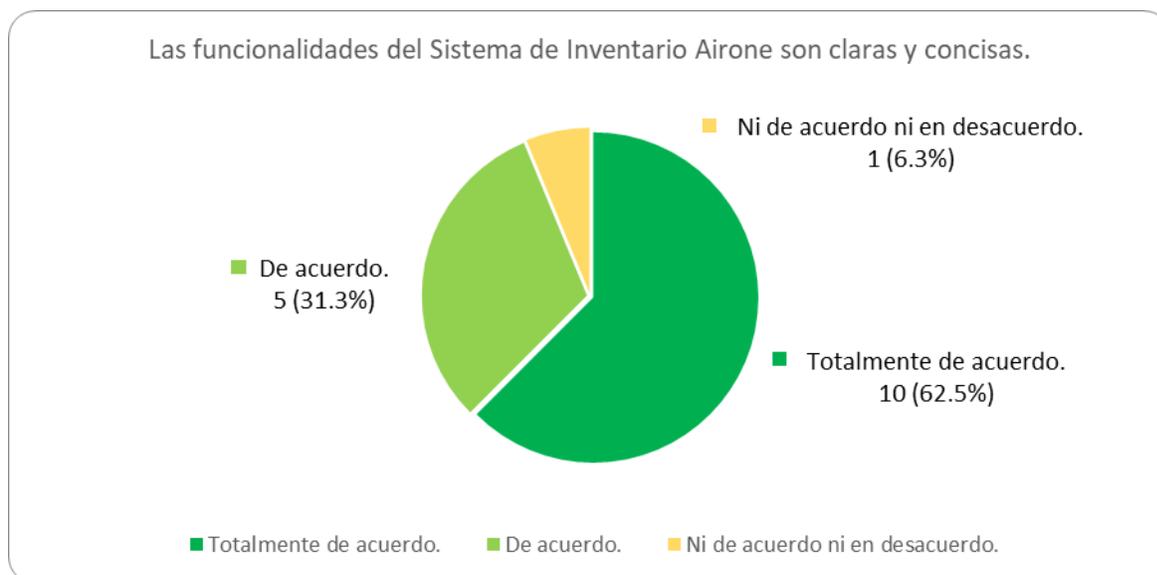


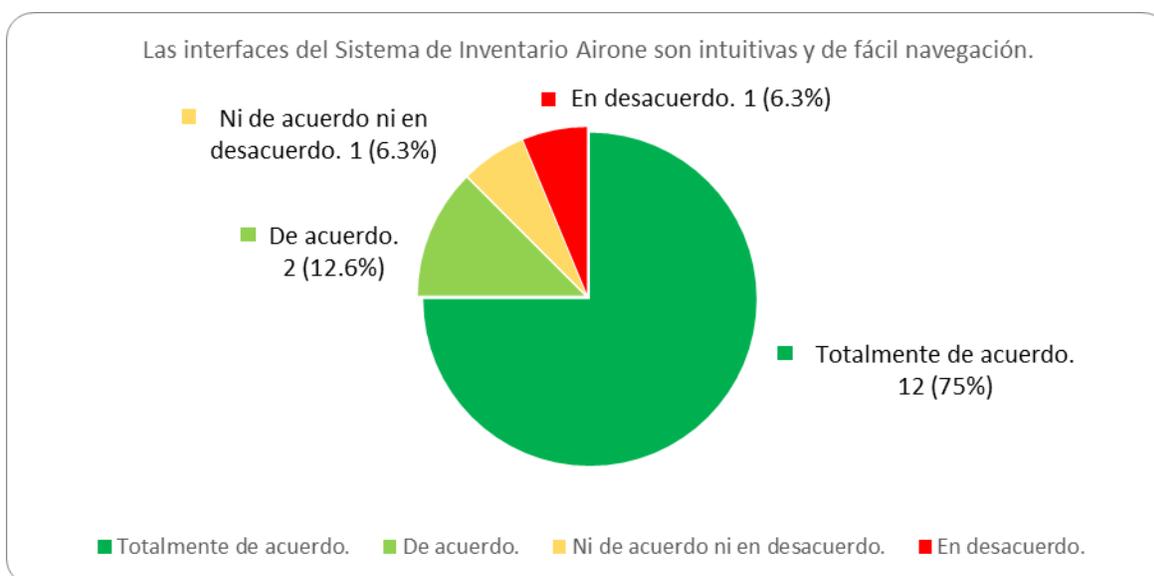
Figura 5. 1. Gráfica de Pastel, Pregunta 1. Prueba de Usabilidad.  
Sistema de Inventario *Airone*.

2.- Las funcionalidades del Sistema de Inventario *Airone* son claras y concisas.



*Figura 5. 2. Gráfica de Pastel, Pregunta 2. Prueba de Usabilidad. Sistema de Inventario Airone.*

3.- Las interfaces del Sistema de Inventario *Airone* son intuitivas y de fácil navegación.



*Figura 5. 3. Gráfica de Pastel, Pregunta 3. Prueba de Usabilidad. Sistema de Inventario Airone.*

4.- El Sistema de Inventario *Airone* muestra mensajes de error explícitos cuando cometo un error.

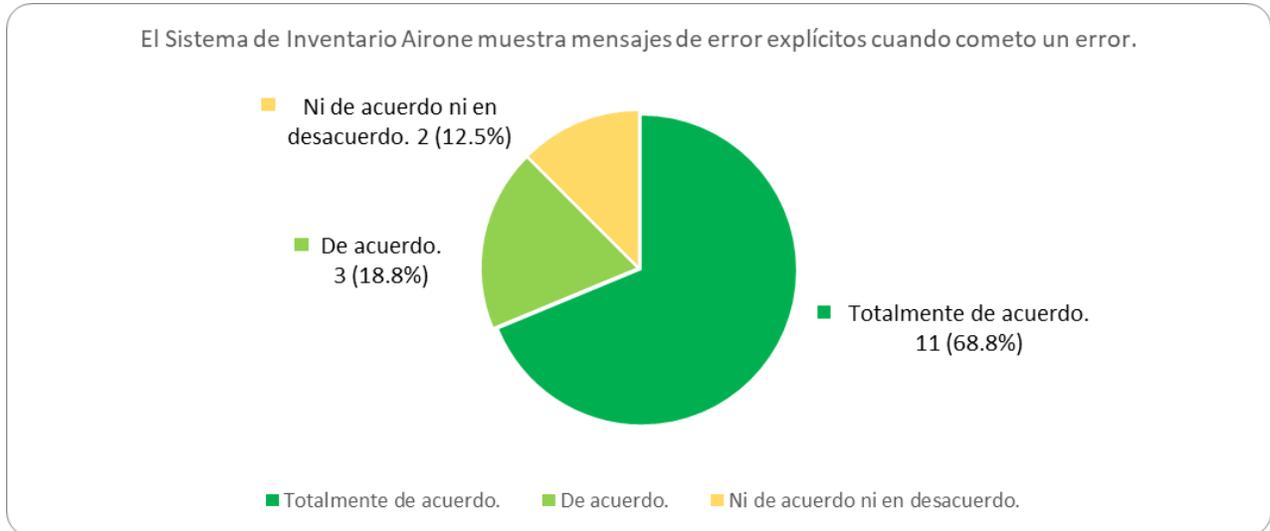


Figura 5. 4. Gráfica de Pastel, Pregunta 4. Prueba de Usabilidad.  
Sistema de Inventario *Airone*

5.- Cada vez que cometo un error en el Sistema de Inventario *Airone* puedo resolverlo fácil y rápidamente.

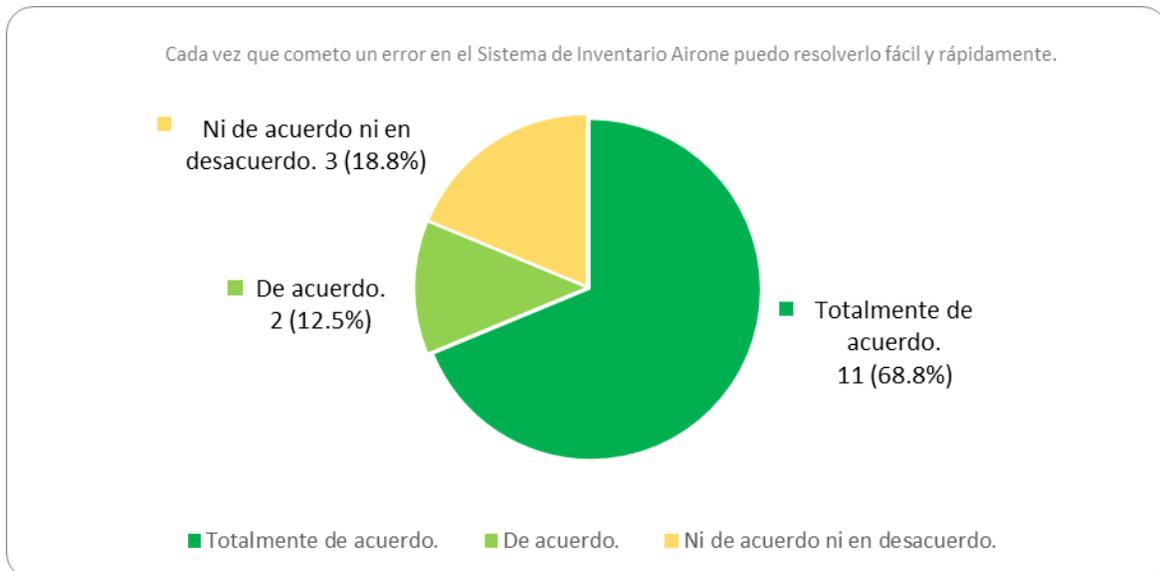


Figura 5. 5. Gráfica de Pastel, Pregunta 5. Prueba de Usabilidad.  
Sistema de Inventario *Airone*

6.- Es fácil encontrar la información que necesito en el Sistema de Inventario *Airone*.

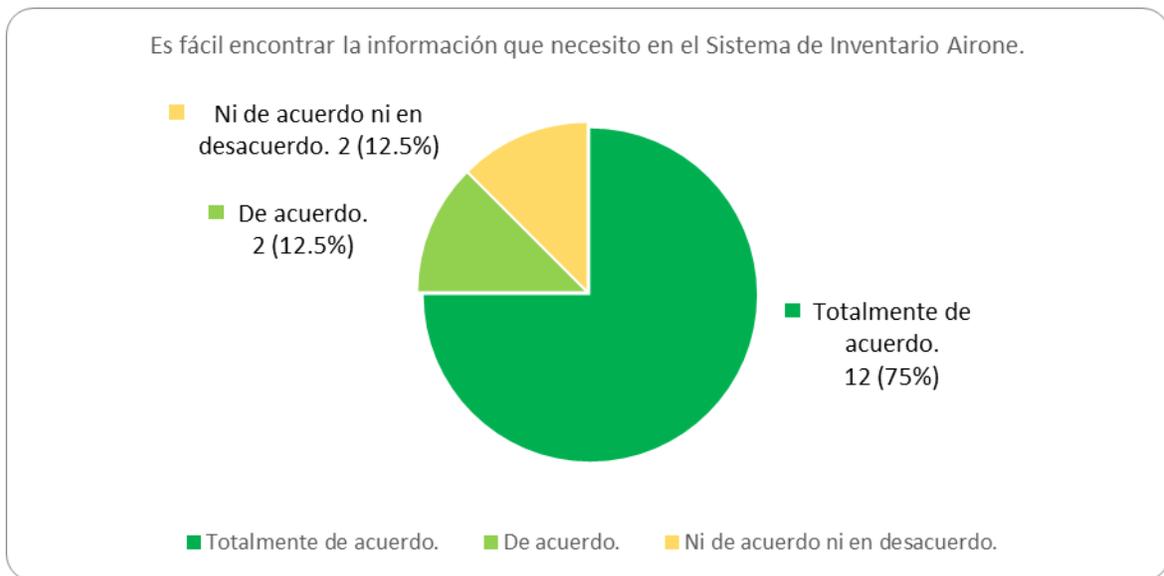


Figura 5. 6. Gráfica de Pastel, Pregunta 6. Prueba de Usabilidad. Sistema de Inventario *Airone*

7.- Me siento muy cómodo/a utilizando el Sistema de Inventario *Airone*.

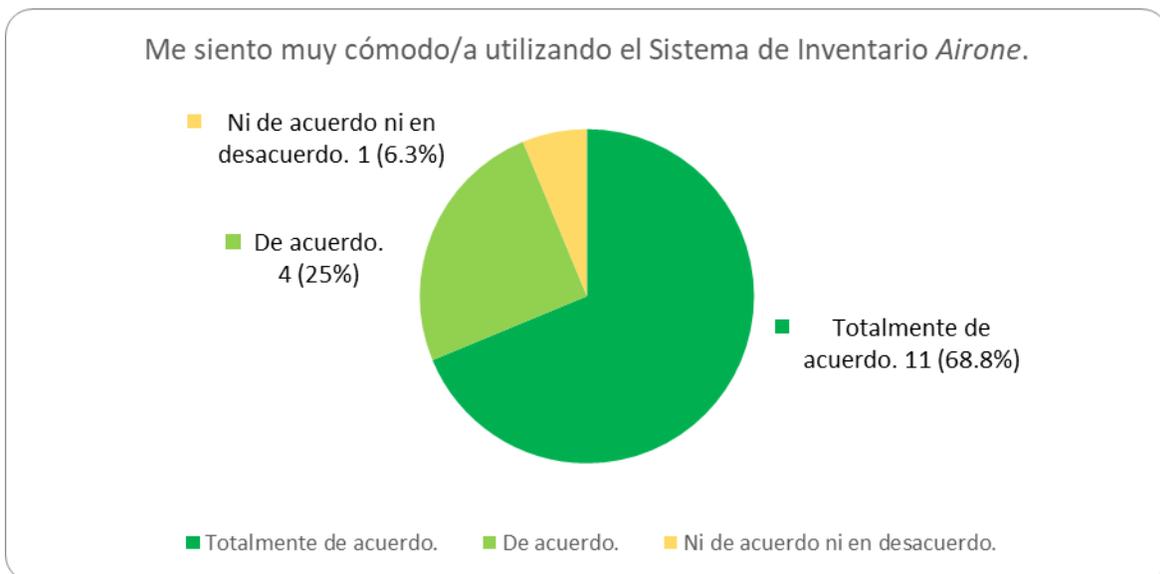


Figura 5. 7. Gráfica de Pastel, Pregunta 7. Prueba de Usabilidad. Sistema de Inventario *Airone*

8.- Me siento muy seguro/a utilizando el Sistema de Inventario *Airone*.

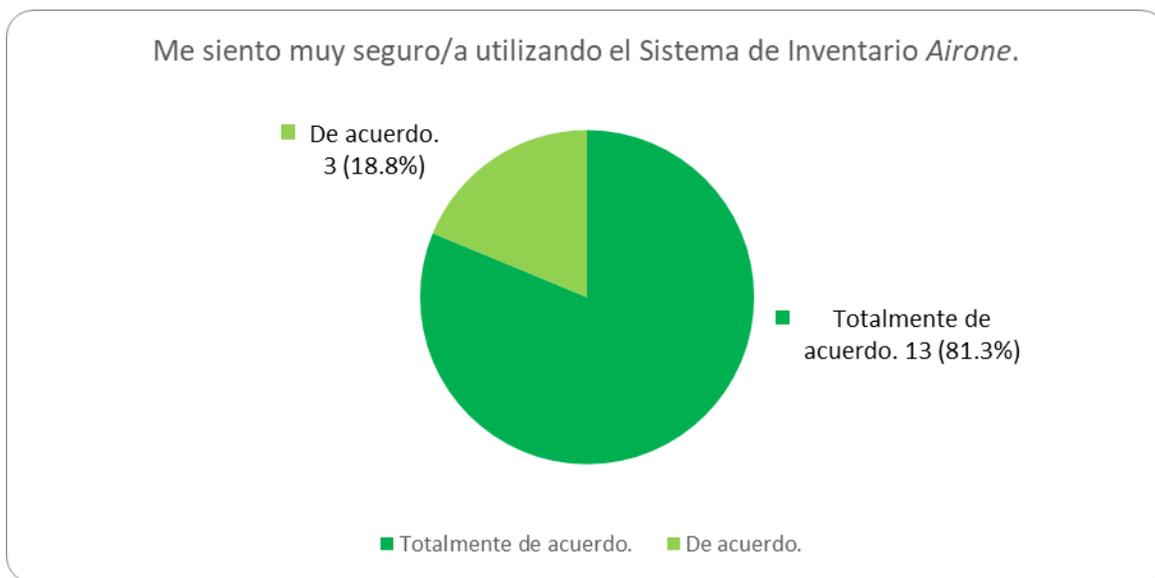


Figura 5. 8. Gráfica de Pastel, Pregunta 8. Prueba de Usabilidad. Sistema de Inventario *Airone*

9.- El Sistema de Inventario *Airone* cuenta con todas las herramientas que esperaba que tuviera.

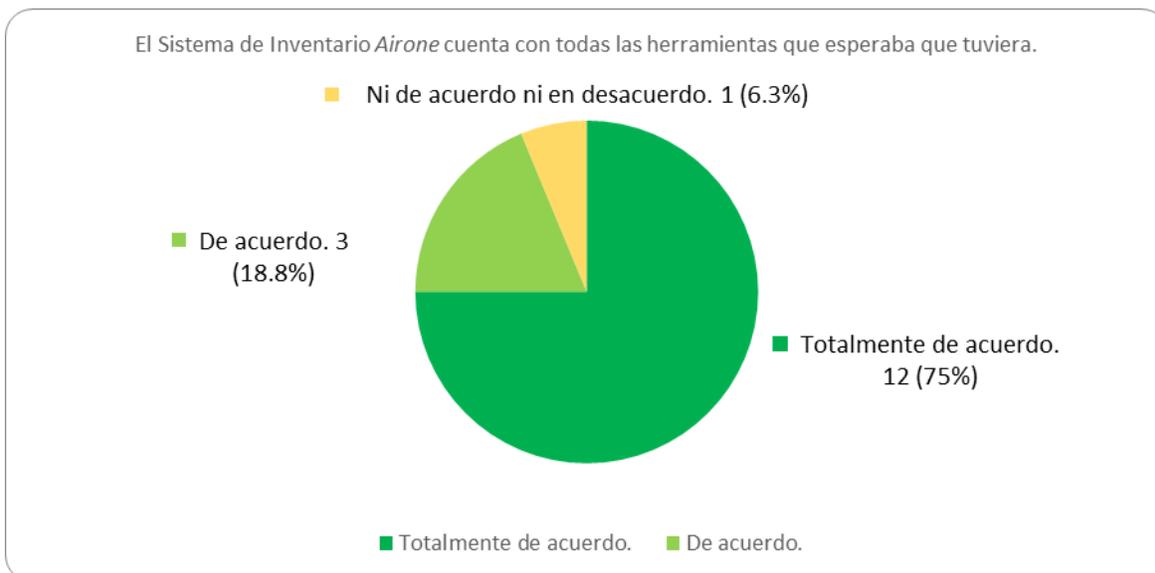


Figura 5. 9. Gráfica de Pastel, Pregunta 9. Prueba de Usabilidad. Sistema de Inventario *Airone*

10.- Requerí de aprender muchas cosas antes de poder utilizar apropiadamente el Sistema de Inventario Airone.

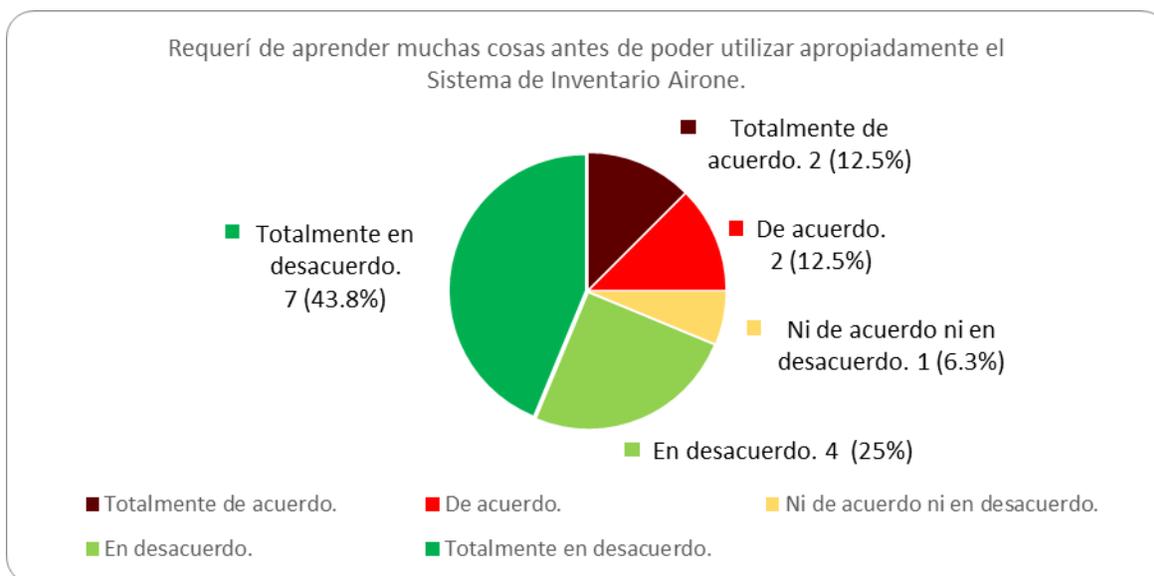


Figura 5. 10. Gráfica de Pastel, Pregunta 10. Prueba de Usabilidad. Sistema de Inventario Airone

11.- En general, estuve satisfecho con el Sistema de Inventario Airone.

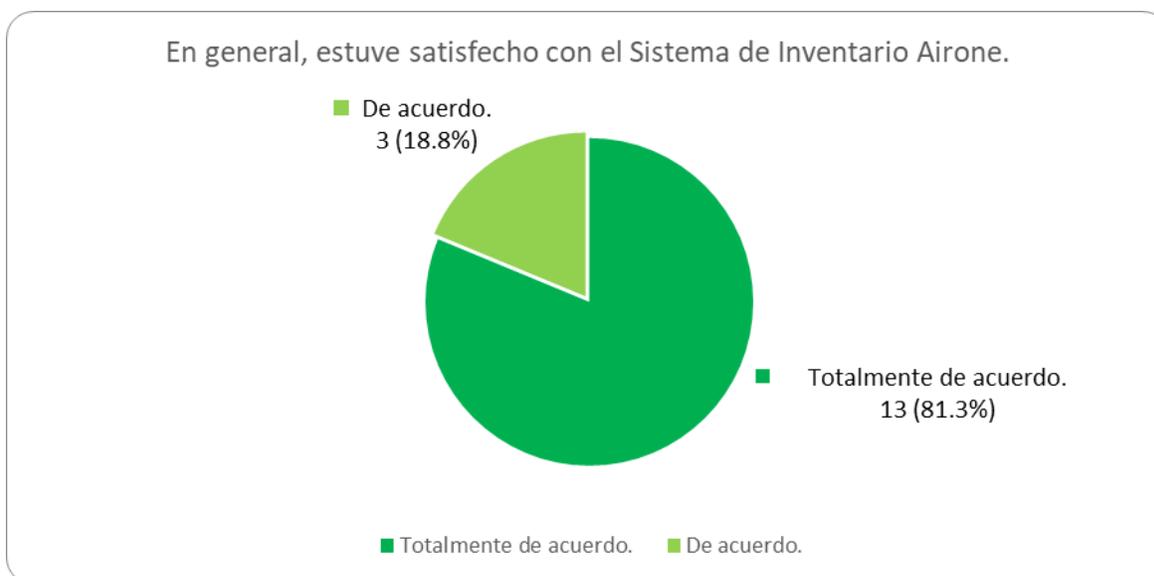


Figura 5. 11. Gráfica de Pastel, Pregunta 11. Prueba de Usabilidad. Sistema de Inventario Airone

Tras la recopilación de las respuestas de cada participante se realizó el cálculo correspondiente para obtener un promedio de puntaje individual para cada una de las preguntas y, finalmente, traducir este promedio a un porcentaje de aceptación del sistema en relación con los aspectos evaluados con la prueba proporcionada.

Antes explicar la metodología realizada es importante rectificar que la pregunta número diez cuenta con la particularidad de que una respuesta negativa en realidad representa una respuesta favorable, se tratará dentro de la metodología de manera pertinente este fenómeno invirtiendo el valor de cada respuesta. La Tabla 5.1 muestra los cambios realizados para las respuestas de esta pregunta:

Respuesta	Valor Original	Valor Sustituido
Totalmente de Acuerdo.	5	1
De Acuerdo.	4	2
Ni de Acuerdo ni en desacuerdo.	3	3
En desacuerdo.	2	4
Totalmente en Desacuerdo.	1	5

*Tabla 5. 1. Cambios de valores, Pregunta 10. Prueba de Usabilidad. Sistema de Inventario Airone*

La metodología aplicada fue la siguiente:

1.- Para cada pregunta, se calculó la sumatoria de los puntos de todas las respuestas (puntaje total) a esa pregunta, una vez obtenida esta sumatoria, a continuación, se divide sobre dieciséis, es decir, el número de personas que ha respondido esa pregunta. De esta manera se obtiene un promedio de puntaje por pregunta.

La Tabla 5.2 muestra el concentrado de datos utilizados en conjunto a los resultados de la aplicación del primer paso de la metodología:

	Pregunta 1	Pregunta 2	Pregunta 3	Pregunta 4	Pregunta 5	Pregunta 6	Pregunta 7	Pregunta 8	Pregunta 9	Pregunta 10	Pregunta 11
Encuestado 1	5	5	5	5	5	5	5	5	5	3	5
Encuestado 2	5	5	5	5	5	5	5	5	5	5	5
Encuestado 3	4	4	4	5	5	5	5	5	5	5	5
Encuestado 4	5	5	5	5	5	5	5	5	5	5	5
Encuestado 5	5	5	5	5	5	5	4	5	5	5	5
Encuestado 6	4	4	5	5	4	5	5	5	5	4	5
Encuestado 7	5	5	5	5	5	5	5	5	5	1	5
Encuestado 8	5	4	5	3	3	3	4	4	4	4	4
Encuestado 9	5	5	4	4	5	5	5	5	4	5	4
Encuestado 10	4	4	5	3	5	5	5	5	4	2	5
Encuestado 11	4	3	5	4	4	4	4	4	5	5	5
Encuestado 12	5	5	5	5	5	5	5	5	5	4	5
Encuestado 13	4	4	3	4	3	3	3	4	3	4	4
Encuestado 14	5	5	5	5	5	5	5	5	5	1	5
Encuestado 15	5	5	5	5	5	4	5	5	5	2	5
Encuestado 16	4	5	2	5	3	5	4	5	5	5	5
Sumatoria	74	73	73	73	72	74	74	77	75	60	77
Promedio	4.63	4.56	4.56	4.56	4.50	4.63	4.63	4.81	4.69	3.75	4.81

*Tabla 5. 2 Concentrado de respuestas, Sumatorias y Promedios. Prueba de Usabilidad. Sistema de Inventario Airone*

2.- A continuación, se realizó la sumatoria de los promedios obtenidos para cada pregunta, el producto de esta sumatoria fue dividido por el número de preguntas con los que cuenta este cuestionario, es decir, once. La Tabla 5.3 muestra este procedimiento:

	Promedio de la Pregunta
Pregunta 1	4.63
Pregunta 2	4.56
Pregunta 3	4.56
Pregunta 4	4.56
Pregunta 5	4.50
Pregunta 6	4.63
Pregunta 7	4.63
Pregunta 8	4.81
Pregunta 9	4.69
Pregunta 10	3.75
Pregunta 11	4.81
Sumatoria	50.125
Promedio Final	4.56

*Tabla 5. 3 Concentrado de Sumatorias y Promedios Finales. Prueba de Usabilidad. Sistema de Inventario Airone*

3.- Finalmente, se tradujo el promedio final de las encuestas a un porcentaje aplicando una regla de tres:

$$\left(\frac{4.56}{5}\right) \times 100 = 91.2\%$$

Una vez aplicada esta metodología se puede afirmar que el porcentaje obtenido representa una evaluación positiva general en relación con las preguntas realizadas a los usuarios encuestados del Sistema de Inventario *Airone*.

## Conclusiones

---

Los sistemas web han ganado una gran relevancia en el panorama actual del desarrollo de aplicaciones y *software* en general, siendo cada vez más común y demandado este tipo de sistemas. Este fenómeno es producto de la mayor accesibilidad que poseen los sistemas web, en comparación con cualquier *software* o sistema de escritorio convencional, además del auge del desarrollo de tecnologías para el desarrollo web, así como navegadores web.

El desarrollo del Sistema de Inventario *Airone* implicó el aprendizaje de nuevas tecnologías de desarrollo web, así como la practica constante de competencias relacionadas a este ámbito, particularmente, el uso de lenguajes dedicados al desarrollo web, así como todas las tecnologías derivadas de estos lenguajes, de acuerdo al papel que desenvuelven dentro del panorama del desarrollo de aplicaciones web.

La aplicación de *Frameworks* de desarrollo web, tanto para *Backend*, como *Frontend*, fue vital durante el desarrollo del Sistema de Inventario *Airone*, debido a que estos proporcionaron un conjunto de recursos estandarizados y documentación, los cuales facilitaron actividades como definir una estructura consistente para el *Backbone* del sistema, así como sus vistas e interfaces que lo componen, integrando *APIs*, gracias al uso del patrón de arquitectura de *software MVC*, proporcionado particularmente, por el *Framework Laravel*.

El Sistema de Inventario *Airone* simplificará una gran gama de actividades administrativas dentro del Área Académica de Ingeniería y Arquitectura de la Universidad Autónoma del Estado de Hidalgo, particularmente aquellas relacionadas a la gestión de inventario, generación de reportes de mantenimiento, planificación de eventos y asistencia en la planeación de adquisición de nuevos recursos para esta área académica.

Finalmente, puede afirmarse que el Sistema de Inventario *Airone* ha demostrado ser una herramienta eficaz, capaz de proporcionar una experiencia intuitiva y enriquecida a nivel de diseño para los usuarios, contando con un gran potencial de desarrollo para ser aplicado en ámbitos ajenos a aquellos donde fue originalmente planeado el sistema.

## Trabajo futuro

---

El Sistema de Inventario *Airone* cuenta con un gran potencial para seguir siendo desarrollado y mejorado, con el fin de cubrir nuevas necesidades o cambios dentro de la administración del Área Académica de Arquitectura e Ingeniería, así como para ser utilizado fuera del ámbito donde este sistema fue propuesto, integrando funcionalidades, las cuales, permitirán al sistema ser una solución mucho más completa y competente frente a otras soluciones de inventario.

Dado este panorama para el Sistema de Inventario *Airone*, se ha identificado un conjunto de mejoras y funcionalidades que pueden ser implementadas.

**Generación de Códigos QR y de barras para cada elemento.** Generación de Códigos de Barras y QR para la rápida identificación de los elementos registrados dentro del Sistema de inventario *Airone*, proporcionando de un medio tangible por el cuál identificar cada uno de los elementos de inventario registrados.

**Mejoras al calendario de eventos.** Registro de eventos persistentes dentro del sistema, programación de eventos los cuales pueden repetirse de acuerdo a una fecha, mes o semana. Creación de resumen de eventos generados en un formato de lista. Habilitado de acciones automatizadas dentro del sistema por medio de eventos programados.

**Sistema de permisos simplificado.** Reducción de listado de permisos asignables dentro del Sistema de Inventario *Airone*, asignación de roles a varios usuarios de manera simultánea.

**Graficas con datos y estadísticas generadas.** Integración de distintas gráficas dentro del Sistema de Inventario *Airone*, las cuales mostrarán información de interés a los usuarios, asistiendo de esta manera a la toma de decisiones para la adquisición de recursos.

**Mejoras a la exportación de reportes.** Posibilidad de adjuntar conjuntos de información de acuerdo a las necesidades del usuario en un único documento de formato PDF. Posibilidad de generar y exportar un reporte general, este reporte describirá diversas estadísticas e información de las actividades y uso del Sistema de Inventario *Airone*.

**Mejoras a la generación de reportes de mantenimiento.** Generación de reportes de mantenimiento por conjuntos de elementos definidos por el usuario o tomando de referencia uno o más atributos por los cuales los registros de elementos se clasifican (Categorías, fondos, etiquetas, entre otros atributos).

## Bibliografía y referencias

---

- Amazon. (2023). *What is SQL?*. <https://aws.amazon.com/es/what-is/sql/>
- Aprende-logística. (2021). *¿Qué es un Inventario? Definición, concepto y tipos* Aprende-logistica.com. <https://aprende-logistica.com/almacen/inventario/>
- Arizbé, K. (2023). *¿Qué es una API y para qué sirve?*. Gluo.mx. Recuperado el 28 de septiembre de 2024, de <https://www.gluo.mx/blog/que-es-una-api-y-para-que-sirve>
- Assembler Institute. (2022). *Que es un Framework en programación y sus principales usos*. <https://assemblerinstitute.com/blog/framework-programacion/>
- Boehm, B. (Mayo, 1988). *A Spiral Model of Software Development and Enhancement*. IEEE Computer. 21 (p. 61-72).
- Boggiano, J. (2024). *Packagist. The PHP Package Repository*. Packagist.org. Recuperado el 3 de septiembre de 2024, de <https://packagist.org>
- Cloudflare. (2024). *¿Qué es HTTPS?*. Recuperado el 25 de agosto de 2024, de <https://www.cloudflare.com/es-es/learning/ssl/what-is-https/>
- Connolly (1999). *HTML 2.0 Materials*. Wwww.w3.org. <https://www.w3.org/MarkUp/html-spec/>
- Cruz G., Galarza, R., Delgado, R., Marcillo, M. (2022). Aplicación de protocolos SSL y TSL para el envío de información. *Journal TechInnovation*, 1(2), 4–9. <https://doi.org/10.47230/Journal.TechInnovation.v1.n2.2022.4-9>
- Doshi D., Jain L., Gala K. (2021). REVIEW OF THE SPIRAL MODEL AND ITS APPLICATIONS. *International Journal of Engineering Applied Sciences and Technology*, Volumen 5, 311 – 316. <https://www.ijeast.com/papers/311-316,Tesma512,IJEAST.pdf>
- Durán, D. (2013). *Encriptación Asimétrica*. Ecured.cu. Recuperado el 25 de agosto de 2024, de [https://www.ecured.cu/Algoritmo\\_asimétrico](https://www.ecured.cu/Algoritmo_asimétrico)
- Editorial Etecé. (2023). *Bienes - Definición, concepto, tipos y características*. Concepto.de <https://concepto.de/bienes/>
- Franco, K. (2020), *Tema: Restricciones de Integridad*. [Diapositiva de PowerPoint]. Universidad Autónoma del Estado de Hidalgo.
- Håkon, W. (s.f.). *Håkon Wium Lie*. <https://www.wiumlie.no/en>

- Hearson, J. (2024). ¿Qué es la gestión de inventario? Oracle.com. <https://www.oracle.com/mx/scm/inventory-management/what-is-inventory-management/>
- IBM. (2024) ¿Qué es una base de datos NoSQL?. Ibm.com. <https://www.ibm.com/mx-es/topics/nosql-databases>
- IBM. (2024). ¿Qué es una API (interfaz de programación de aplicaciones)?. Ibm.com. <https://www.ibm.com/mx-es/topics/api>
- International Telecommunication Union. (2024). *Facts and Figures - Internet use*. itu.int. Recuperado el 25 de mayo de 2025, de <https://www.itu.int/itu-d/reports/statistics/2024/11/10/ff24-internet-use/>
- Kunze, M. (1998), “Let There be Light” LAMP: Freeware Web Publishing System with Database Support. Archive.org. Recuperado el 3 de septiembre de 2024, de <https://web.archive.org/web/19990203093819/http://www.heise.de/ct/english/98/12/230/>
- Kurose, J. F., Ross, K. W., Mañoso Hierro, C., Pérez de Madrid, A., & Martín Romo, M. (2010). *Redes de computadoras: un enfoque descendente*. Addison-Wesley.
- Landaluce C. (2024). RECODIFICACIÓN DE ESCALAS TIPO LIKERT A TRAVÉS DE LA CLASIFICACIÓN NO SUPERVISADA. LAS IMPLICACIONES DE LAS RELACIONES POR INTERNET RESPECTO A LAS RELACIONES PRESENCIALES COMO ESTUDIO DE CASO. *Revista Internacional de Sociología*, Volumen 8, 1 – 18. <https://revintsociologia.revistas.csic.es/index.php/revintsociologia/article/view/1419/1736>
- Laravel Holdings Inc. (2023). *Introduction - Documentation Laravel PHP Framework*. Recuperado el 26 de agosto de 2024, de <http://laravel.com/docs/introduction>
- Laravel Holdings Inc. (2024). *Laravel - the PHP framework for web artisans*. Laravel.com. <https://laravel.com>
- Lee, H., Lee, C. y Yoo, C. (1998). *Scenario-based object-oriented methodology for developing hypermedia information systems*. Proceedings of the Hawaii International Conference on System Sciences (p. 47-56).
- Lenovo. (s.f). ¿Qué es Linux?. Recuperado el 3 de septiembre de 2024, de <https://www.lenovo.com/mx/es/faqs/sistemas-operativos/que-es-linux/?srsltid=AfmBOorkU1poeJcWD1W0qaUHXgF3HPVXS22aQynXjeoXnIsgIJDt5jWK>

- López, J. (2009). *Breve historia de CSS*. Universidad de Murcia. [http://dis.um.es/~lopezquesada/documentos/IES\\_1213/LMSGI/curso/UT5/libroswebcss/www.librosweb.es/css/capitulo1/breve\\_historia\\_de\\_css.html](http://dis.um.es/~lopezquesada/documentos/IES_1213/LMSGI/curso/UT5/libroswebcss/www.librosweb.es/css/capitulo1/breve_historia_de_css.html)
- López, Y. (2023). *La historia completa de JavaScript, el único lenguaje que entienden los navegadores*. Ed.team. Recuperado el 25 de agosto de 2024 de <https://ed.team/blog/la-historia-completa-de-javascript-el-unico-lenguaje-que-entienden-los-navegadores>
- Maida, E. y Pacienza, J. (2015). *Metodologías de desarrollo de software* [Tesis de licenciatura], Universidad Católica Argentina <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Melendez, S., Gaitan, M., Pérez, N. (2016). *METODOLOGÍA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACIÓN EXTREMA*. <https://repositorio.unan.edu.ni/id/eprint/1365/1/62161.pdf>
- Mozilla (2024). *HTML5*. MDN Web Docs. <https://developer.mozilla.org/es/docs/Glossary/HTML5>
- Mozilla (2024). *World Wide Web*. MDN Web Docs. [https://developer.mozilla.org/es/docs/Glossary/World\\_Wide\\_Web](https://developer.mozilla.org/es/docs/Glossary/World_Wide_Web)
- Mozilla (2025). *An overview of HTTP*. MDN Web Docs. <https://developer.mozilla.org/es/docs/Web/HTTP/Guides/Overview>
- openSUSE. (2011). *Conceptos Gestión de paquetes*. [http://es.opensuse.org/Conceptos\\_Gestión\\_de\\_paquetes](http://es.opensuse.org/Conceptos_Gestión_de_paquetes)
- Oracle. (s.f.) *Introducción al conjunto de protocolos TCP/IP*. Oracle.com. Recuperado el 25 de agosto de 2024, de <https://docs.oracle.com/cd/E19957-01/820-2981/6nei0r0r9/index.html>
- Oracle. (s.f.). *¿Qué es NoSQL?* Oracle.com. Recuperado el 25 de agosto de 2024, de <https://www.oracle.com/mx/database/nosql/what-is-nosql/>
- PHP Documentation Group (2024). *¿Qué es PHP?*. PHP.Net. <https://www.w3.org/MarkUp/html-spec/>
- Pressman, S. (2010). *Ingeniería del Software, Un enfoque Practico, Séptima Edición*. McGraw Hill.
- Reenskaug, T. (s.f.). *Trygve MVC*. Universitetetioslo.no. <https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html>

- Simpliroute. (2023). *Sistema de Inventario: Qué es, Tipos y Cómo Implementarlo*.  
<https://simpliroute.com/es/blog/sistema-de-inventario>
- Sinnaps. (2020). *METODOLOGÍA XP O PROGRAMACIÓN EXTREMA*.  
<https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp>
- Sommerville, I. (2011). *Ingeniería de Software, Novena Edición*. PEARSON EDUCACIÓN.
- TechWebPe. (2021). *HTTP: ¿Qué es? ¿Qué función cumple? ¿Cómo funciona?* [Imagen].  
<https://techwebpe.com/dominios/http-que-es-como-funciona/>
- Universidad Alfonso X El Sabio. (2024). *¿Qué es la administración educativa y para qué sirve?*. Uax.com. Recuperado el 27 de agosto de 2024, de  
<https://www.uax.com/blog/educacion/administracion-educativa-que-es>
- Universidad Autónoma del Estado de Hidalgo. (2024). *Escuelas e Institutos — UAEH*.  
<https://www.uaeh.edu.mx/campus/>
- Universidad Autónoma del Estado de Hidalgo. (s.f.). *Comparación de los modelos OSI y TCP/IP*. Recuperado el 27 de agosto de 2024, de  
<https://www.uaeh.edu.mx/scige/boletin/huejutla/n10/r1.html>
- Universidad Internacional de La Rioja (2023). *Administración educativa: qué es, tipos y ejemplos*. Unir.net. Recuperado el 26 de agosto de 2024, de  
<https://mexico.unir.net/noticias/educacion/administracion-educativa/>
- Wells, D. (2009). *Extreme Programming: A gentle introduction*.  
<http://www.extremeprogramming.org/>
- World Wide Web Consortium. (2024). *Tim Berners-Lee*.  
<https://www.w3.org/People/Berners-Lee/>