



Universidad
Nacional
de Loja

Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos

Naturales no Renovables

Carrera de Ingeniería en Sistemas

Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja

Web application for management of the representativeness of urban trees into the Dynamic research project of growth and ecosystem services of urban grove in the city of Loja

Trabajo de Titulación
previo a la obtención del
título de Ingeniero en
Sistemas

AUTOR:

Johnny Fabián González Guamán

DIRECTOR:

Ing. Wilman Patricio Chamba Zaragocín, Mg. Sc

Loja – Ecuador

2023

Certificación

Loja, 17 de febrero del 2023

Ing. Wilman Patricio Chamba Zaragocín, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

CERTIFICO:

Que he revisado y orientado todo el proceso de elaboración del Trabajo Titulación denominado: **Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja**, previo a la obtención del título de **Ingeniero en Sistemas**, de la autoría del estudiante **Johnny Fabián González Guamán**, con cédula de identidad Nro.**1150036125**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.

Ing. Wilman Patricio Chamba Zaragocín, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

Autoría

Yo, **Johnny Fabián González Guamán**, declaro ser autor del presente Trabajo de Titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos, de posibles reclamos y acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja la publicación de mi Trabajo de Titulación, en el Repositorio Digital Institucional – Biblioteca Virtual.

Firma:

A handwritten signature in blue ink, appearing to read 'Johnny Fabián González Guamán', with a horizontal line extending to the right.

Cédula de identidad: 1150036125

Fecha: 28 de marzo de 2023

Correo electrónico: jfgonzalezg@unl.edu.ec

Teléfono: +593 99-217-3117

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Titulación

Yo, **Johnny Fabián González Guamán**, declaro ser el autor del Trabajo de Titulación denominado: **Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que, con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior, con los cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del Trabajo de Titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los veintiocho días del mes de marzo de dos mil veintitrés.

Firma:



Autor: Johnny Fabián González Guamán

Cédula: 1150036125

Dirección: Loja (Cdla. del Maestro Segunda Etapa)

Correo Electrónico: jfgonzalezg@unl.edu.ec

Teléfono: +593 99-217-3117

DATOS COMPLEMENTARIOS:

Director del Trabajo de Titulación: Ing. Wilman Patricio Chamba Zaragocín, Mg. Sc.

Dedicatoria

A Dios, por permitirme llegar a este momento tan importante en mi vida. Por los triunfos y los momentos difíciles que me ha enseñado a valorarlo cada día más.

A mis padres Alonso y Neli quienes han sido y siguen siendo un pilar fundamental de apoyo para conseguir mis metas, gracias a su ejemplo, enseñanza de perseverancia y buenos valores me han permitido cumplir una de las metas valiosas en mi vida.

Johnny Fabián González Guamán

Agradecimiento

Mi especial agradecimiento a Dios por darme esa fortaleza para continuar día a día cumpliendo los propósitos que me he planteado. Por la salud y bendición que da a mí y mi familia para apoyarnos en todo momento, permitirnos, proponernos y alcanzar nuestras metas.

A mis padres, que han estado siempre apoyándome para que pueda cumplir mis objetivos de vida y llegar a ser un buen profesional y una excelente persona.

De manera especial a la Carrera de Ingeniería en Sistemas de la Universidad Nacional de Loja y a sus docentes, quienes han sido una pieza fundamental en mi formación profesional a lo largo de todos mis años de estudio. Gracias por su esfuerzo y sus conocimientos impartidos para formar buenos profesionales.

Al Ing. Wilman Patricio Chamba Zaragocín, Mg. Sc. director de Trabajo de Titulación, quien gracias a sus conocimientos ha impulsado a seguir y culminar este trabajo de titulación.

Johnny Fabián González Guamán

Índice de Contenidos

Portada	i
Autoría	iii
Carta de autorización	iv
Dedicatoria	v
Agradecimiento	vi
Índice de Contenidos	vii
Índice de tablas	x
Índice de figuras	xiii
Índice de anexos	xvi
1. Título	1
2. Resumen	2
2.1. Abstract.....	3
3. Introducción	4
4. Marco Teórico	6
4.1. Proyecto de vinculación con la sociedad.....	6
4.1.1. Contexto	6
4.1.2. Proyecto de investigación forestal.....	7
4.2. Metodología de desarrollo de software	8
4.2.1. Metodología ICONIX.....	8
4.3. Tecnologías de desarrollo de software.....	10
4.3.1. JavaScript.....	10
4.3.2. React	10
4.3.3. Node (Express).....	10
4.3.4. ORM (Object-Relacional Mapping)	11
4.4. Servidor de mapas.....	12
4.4.1. Mapbox.....	12
4.5. Trabajos relacionados.....	14

5. Metodología	16
5.1. Contexto	17
5.2. Procedimiento.....	18
5.2.1. Objetivo 1: Establecer el diseño de la Aplicación Web en base a la metodología ICONIX.	18
5.2.2. Objetivo 2: Implementar el diseño de la aplicación mediante el framework ReactJS y NodeJS.....	18
5.2.3. Objetivo 3: Ejecutar pruebas de funcionalidad para la validación de la Aplicación Web	19
5.3. Recursos.....	19
5.3.1. Recursos Científicos	19
5.3.2. Recursos Técnicos	19
5.3.3. Recursos humanos	20
5.3.4. Recursos de Hardware y Software.....	20
6. Resultados	21
6.1. Objetivo 1: Establecer el diseño de la Aplicación Web en base a la metodología ICONIX.	21
6.1.1. Elaborar el modelo de requerimientos del sistema.....	21
6.1.2. Elaborar el diseño lógico y físico de la aplicación.	23
6.2. Objetivo 2: Implementar el diseño de la aplicación mediante el framework ReactJS y NodeJS.....	35
6.2.1. Codificar la Aplicación Web	35
6.2.2. Documentar la arquitectura de la aplicación web	40
6.3. Objetivo 3: Ejecutar pruebas de funcionalidad para la validación de la Aplicación Web	43
6.3.1. Pruebas unitarias al servicio web API-REST.....	43
6.3.2. Pruebas de Caja Negra	44
6.3.3. Pruebas funcionales	45
6.3.4. Pruebas no funcionales	45
7. Discusión	49
8. Conclusiones	52

9. Recomendaciones	53
10. Bibliografía.....	55
11. Anexo	58

Índice de tablas:

Tabla 1. Servidores de mapas web	12
Tabla 2. Resumen de la Metodología ICONIX utilizada en el TT.....	16
Tabla 3. Requerimientos Funcionales	21
Tabla 4. Requerimiento No Funcionales	22
Tabla 5. Fichas de caso de uso Visualizar árboles.....	25
Tabla 6. Ficha de caso de uso Visualizar reportes	26
Tabla 7. Ficha de caso de uso Editar árbol	26
Tabla 8. Ficha de caso de uso Descargar reporte.....	27
Tabla 9. Principales endpoints del servicio web de la aplicación WebGISTree	43
Tabla 10. Caso de prueba de caja negra - Editar árbol.	44
Tabla 11. Aprobación de funcionalidad final de requerimientos funcionales.....	45
Tabla 12. Requerimientos Funcionales	69
Tabla 13. Requerimiento No Funcionales	70
Tabla 14. Fichas de caso de uso Visualizar árboles.....	74
Tabla 15. Ficha de caso de uso Visualizar reportes	74
Tabla 16. Ficha de caso de uso Autenticar usuario.....	75
Tabla 17. Ficha de caso de uso Crear usuario	76
Tabla 18. Ficha de caso de uso Registrar árbol	76
Tabla 19. Ficha de caso de uso Eliminar árbol.....	77
Tabla 20. Ficha de caso de uso Descargar reporte.....	77
Tabla 21. Ficha de caso de uso Registrar cuenta	78
Tabla 22. Ficha de caso de uso Validar Cuenta	78
Tabla 23. Petición para obtener todos los registros de árboles.	111
Tabla 24. Petición para obtener un registros de árbol por su ID.....	112
Tabla 25. Petición para conocer si un árbol está activo.....	112
Tabla 26. Petición para registrar datos en tabla árbol	113
Tabla 27. Petición para iniciar sesión de usuario	113
Tabla 28. Petición para actualizar datos en tabla árbol	114
Tabla 29. Petición para habilitar estado de árbol.....	114
Tabla 30. Petición para obtener los roles de usuarios	115
Tabla 31. Petición para obtener lista de usuarios.....	115
Tabla 32. Petición para obtener información de usuarios logueado	116
Tabla 33. Petición para traer datos de usuario por su Id	116
Tabla 34. Petición para registrar un usuario.....	117
Tabla 35. Petición para habilitar un usuario	117

Tabla 36. Petición para actualizar usuario.....	118
Tabla 37. Petición para deshabilitar usuario.....	118
Tabla 38. Petición para verificar token.	119
Tabla 39. Petición para verificar si existe el email	119
Tabla 40. Petición para enviar link de recuperar contraseña	120
Tabla 41. Petición para obtener las meta-características.....	120
Tabla 42. Petición para obtener especies	121
Tabla 43. Petición para filtrar especies por familia	121
Tabla 44. Petición para registrar especie	122
Tabla 45. Petición para actualizar especie	122
Tabla 46. Petición para obtener lista de nombre comunes.....	123
Tabla 47. Petición para registrar familia	123
Tabla 48. Petición para obtener sitios	124
Tabla 49. Petición para registrar nuevo sitio	124
Tabla 50. Petición para registrar nuevo sitio	125
Tabla 51. Petición para obtener localizaciones de árboles.....	125
Tabla 52. Petición para obtener propiedades para Id de árbol	126
Tabla 53. Petición para actualizar lista de características para un árbol	127
Tabla 54. Petición para filtrar árboles por características.....	128
Tabla 55. Petición para actualizar datos en tabla inconveniente	128
Tabla 56. Petición para obtener parroquias.....	129
Tabla 57. Petición para obtener categorías.....	129
Tabla 58. Petición para obtener tipos de categorías	130
Tabla 59. Petición para obtener columnas para tabla reportes.....	130
Tabla 60. Petición para obtener filas para tabla reporte	131
Tabla 61. Caso de prueba - Visualizar Árboles.....	137
Tabla 62. Caso de prueba – Mostrar mapas	137
Tabla 63. Caso de prueba – Filtrar árboles	138
Tabla 64. Caso de prueba – Visualizar reportes.....	138
Tabla 65. Caso de prueba – Autenticar usuario	138
Tabla 66. Caso de prueba – Resetear contraseña.....	139
Tabla 67. Caso de prueba – Cambiar contraseña	139
Tabla 68. Caso de prueba – Registrar usuario.....	139
Tabla 69. Caso de prueba – Validar cuenta	140
Tabla 70. Caso de prueba – Editar árbol.....	140
Tabla 71. Caso de prueba – Registrar Árbol	141
Tabla 72. Caso de prueba – Eliminar árbol	142

Tabla 73. Caso de prueba – Descargar reporte de árboles 142

Índice de figuras:

Figura 1. Descripción del proceso de Iconix.	9
Figura 2. Mapas de Mapbox [23].	13
Figura 3. Modelo de dominio del negocio	23
Figura 4. Prototipo rápido de GUI en Figma.	24
Figura 5. Diagramas de caso de uso de la Aplicación Web.	25
Figura 6. Diagrama de secuencia para el caso de uso Visualizar árboles.....	28
Figura 7. Diagrama de secuencia para el caso de uso Visualizar reportes.	29
Figura 8. Diagrama de secuencia para el caso de uso Autenticar usuario.	29
Figura 9. Diagrama de secuencia para el caso de uso Recuperar/Actualizar contraseña. ..	30
Figura 10. Diagrama de secuencia para el caso de uso Editar árbol.	30
Figura 11. Diagrama de secuencia para el caso de uso Registrar árbol	31
Figura 12. Diagrama de secuencia para el caso de uso Eliminar árbol.....	31
Figura 13. Diagrama de secuencia para el caso de uso Descargar reporte.....	32
Figura 14. Diagrama de clases	33
Figura 15. Diagrama de datos del aplicativo web obtenidos de MySQL Workbench.....	34
Figura 16. Estructura de archivos de un servicio API-REST	35
Figura 17. Estructura de carpetas y paquetes del backend.	36
Figura 18. Estructura de carpetas y paquetes del frontend.....	36
Figura 19. Especificación de rutas para el servicio usuario.....	37
Figura 20. Especificación del modelo especie.	37
Figura 21. Controlador actualizar especie por si Id.	38
Figura 22. Middlewares para autorización y verificar usuario administrador.	38
Figura 23. Schemas para la validación de entradas.	39
Figura 24. Interfaz inicial de la aplicación web, árbol seleccionado.....	40
Figura 25. Interfaz para administrar registro de árboles.....	40
Figura 26. Diagrama de componentes de la aplicación web.	41
Figura 27. Diagrama de despliegue en servidores Railway y Netlify.	42
Figura 28. Interfaz con funcionalidades de acuerdo al rol.....	46
Figura 29. Encriptación de contraseña en base de datos.	47
Figura 30. Controlador para crear registros de árboles por lotes	47
Figura 31. Envío de registros de árboles en formato JSON	48
Figura 32. Configuración del límite del cuerpo de la solicitud en Express.....	48
Figura 33. Solicitud de tesista a Director de CIS.....	58
Figura 34. Inventario de árboles en formato xlsx.	63
Figura 35. Inventario de especies de árboles	64

Figura 36. Ficha de campo para evaluación fitosanitaria de árboles urbanos.	65
Figura 37. Modelo de dominio del negocio	71
Figura 38. Modelo de dominio de seguridad	71
Figura 39. Caso de Uso para el Sistema de Información Geográfica de árboles.	72
Figura 40. Diseño de prototipo inicial de GUI para la aplicación web.....	73
Figura 41. Diagrama de secuencia para el caso de uso Visualizar árboles.....	79
Figura 42. Diagrama de secuencia para el caso de uso Visualizar reportes.	79
Figura 43. Diagrama de secuencia para el caso de uso Autenticar usuario.	80
Figura 44. Diagrama de secuencia para el caso de uso Recuperar/Actualizar contraseña.	80
Figura 45. Diagrama de secuencia para el caso de uso Editar árbol.	81
Figura 46. Diagrama de secuencia para el caso de uso Registrar árbol.	81
Figura 47. Diagrama de secuencia para el caso de uso Eliminar árbol.....	82
Figura 48. Diagrama de secuencia para el caso de uso Descargar reporte.	82
Figura 49. Diagrama de secuencia para el caso de uso Registrar usuario.....	83
Figura 50. Diagrama de secuencia para el caso de uso Validar Cuenta usuario.....	83
Figura 51. Diagrama de clase inicial basado en el aplicativo móvil.	84
Figura 52. Diagrama de clase adapta a los requerimientos del aplicativo web.	85
Figura 53. Diagrama de datos del aplicativo web obtenidos del MySQL Workbench.	86
Figura 54. Estructura interna de los servicios en apiServices.	88
Figura 55. Definición de las rutas para el servicio usuario	88
Figura 56. Estructura de un modelo con Sequelize ORM.	89
Figura 57. Estructura de un controlador para la actualización de datos.	89
Figura 58. Middleware de autorización de token.....	90
Figura 59. Middleware de autorización para identificar usuario administrador.	91
Figura 60. Métodos más utilizados del archivo utils.	91
Figura 61. Repositorio GitHub de frontend y backend.....	92
Figura 62. Repositorio GitHub de base de datos.	92
Figura 63. Esquema de validación de entradas con Yup.	93
Figura 64. Estructura de carpetas y paquetes del backend.	94
Figura 65. Estructura de carpetas y paquetes del frontend.....	94
Figura 66. Pantalla principal de la aplicación web.....	95
Figura 67. Aplicar filtros por lugar de ubicación.	96
Figura 68. Aplicar filtros por características de árboles.....	96
Figura 69. Aplicar filtros por nombre científico del árbol.....	96
Figura 70. Alternativas de mapas base de la aplicación web.	97
Figura 71. Reporte preliminar del árbol y observación de código QR	98
Figura 72. Reporte completo de un árbol seleccionado.	98

Figura 73. Pantallas para la autenticación de usuario.....	99
Figura 74. Proceso de recuperación de contraseña.....	100
Figura 75. Proceso de registro de usuario, envió de email al usuario administrador.....	100
Figura 76. Vista Actualizar perfil de usuario.....	101
Figura 77. Vista Administración de árboles, acciones para usuarios con rol investigador.	101
Figura 78. Acción Editar árboles en página Administración.	102
Figura 79. Acción Desplazar ubicación de árbol en mapa.	102
Figura 80. Campos para la Edición o Creación de árbol.	103
Figura 81. Vista de pantalla para Registrar árbol.	103
Figura 82. Acción Eliminar habilitado para usuario con rol Administrador.	104
Figura 83. Vista de Administración de árboles, acciones para usuario con rol administrador.	104
Figura 84. Vista de la interfaz Reporte adaptado a un tamaño de pantalla menor a 900px de ancho.	105
Figura 85. Mostrar u ocultar las columnas de la tabla.	106
Figura 86. Aplicar filtros por columnas.....	106
Figura 87. Archivo descargado en formato csv.	106
Figura 88. Administrar registros de familias.	107
Figura 89. Administrar registros de especies.	107
Figura 90. Administrar registros de usuarios.....	107
Figura 91. Diagrama de componentes de la aplicación web.	108
Figura 92. Diagrama de despliegue.	109
Figura 93. Prueba de navegador Chrome versión 109.0.5414.119.....	133
Figura 94. Prueba de navegador Firefox versión 109.0.1	133
Figura 95. Prueba de navegador Safari versión 16.2.....	134
Figura 96. Prueba de navegador Opera versión 56.03051.104.....	134
Figura 97. Prueba de navegador Edge versión 109.0.1518.70	135
Figura 98. Prueba de requerimientos funcionales.....	143
Figura 99. Diseño de Poster Científico para Casa Abierta, Ingeniería Forestal, UNL.....	145
Figura 100. Exposición de poster en Casa Abierta, Ingeniería Forestal-UNL.....	146
Figura 101. Socialización de aspectos técnicos de la aplicación web.....	146

Índice de anexos:

Anexo 1. Solicitud de Tesista para el desarrollo de la Aplicación Web.....	57
Anexo 2. Entrevista al Director del Proyecto de investigación Forestal.....	58
Anexo 3. Revisión de documentación física y digital.....	61
Anexo 4. Desarrollo de la metodología ICONIX.....	65
Anexo 5. Pruebas al servicio Web API-REST	109
Anexo 6. Pruebas de navegador	131
Anexo 7. Pruebas de Caja Negra	135
Anexo 8. Pruebas Funcionales	142
Anexo 9. Presentación de Poster del Trabajo de Titulación en Casa Abierta, Ingeniería Forestal – UNL	143
Anexo 10. Acuerdo de confidencialidad y no divulgación de la base de datos del proyecto 17-DI-FARNR-2021: Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja	146
Anexo 11. Certificado de traducción del resumen.....	151

1. Título

Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja.

Web application for management of the representativeness of urban trees into the Dynamic research project of growth and ecosystem services of urban grove in the city of Loja

2. Resumen

Actualmente, la gestión sostenible de los árboles urbanos es una preocupación constante en varias ciudades del país, por ende, muchas instituciones del sector público, formulan e implementan políticas destinadas a la conservación, uso y manejo de los árboles en zonas urbanas. En este contexto, el proyecto de investigación forestal “*Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja*” busca elaborar y mantener un inventario de los árboles dentro de sus respectivos límites, para ello, se requiere una aplicación que gestione los datos científicos y técnicos recopilados por los investigadores de forma manual, con la finalidad que sirva a futuras investigaciones, dando origen a la pregunta de investigación: ¿Cómo se podrá ayudar a los investigadores del proyecto forestal a obtener la información relacionada a los registros de árboles urbanos de la ciudad de Loja?. Para resolver esta pregunta el Trabajo de Titulación se desarrolló con la metodología ICONIX, estableciendo los modelos de diseño arquitectónico lógico y físico de la aplicación representados en diagramas UML, en base a los requerimientos del sistema validados por el director del proyecto forestal. Posteriormente, se implementó estos modelos aplicando el estilo n-tier, desarrollando el backend y frontend bajo las tecnologías JavaScript, React, Mapbox, Node, Express, Sequelize y MySQL, obteniendo un producto de software acordado por el investigador. Finalmente, se realizaron pruebas de caja negra y funcionales, al director del proyecto de investigación forestal quien validó funcionalmente toda la aplicación, y las pruebas de los servicios API-REST para las futuras integraciones de aplicaciones.

Así se dio contestación a la pregunta de investigación, creando una aplicación web que permite al usuario según su rol gestionar las características dasométricas, morfológicas, ecológicas y servicios ecosistémicos del arbolado de la ciudad de Loja a través de mapas interactivos en el menor tiempo posible.

Palabras claves: Gestión forestal urbana, ICONIX, Software, React, Node, MySQL.

2.1. Abstract

Nowadays, the sustainable management of urban trees is a constant concern in various cities in the country, therefore many public institutions, formulate and implement policies destined to conservation, use and management of trees in urban areas. In this context, the forestry research project. "Growth dynamics and ecosystem services of urban grove in the city of Loja" seeks to elaborate and keep an inventory of trees within their respective limits, for this is required an application to manage the scientific and technical data collected by researchers manually, with the purpose of helping future research, giving rise to the research question: How could be help the researchers of the forestry project to obtain information related to records of trees in urban areas of the city of Loja? To solve this question, the Degree Project was developed with the ICONIX methodology, establishing the logical and physical architectural design models of the application represented in UML diagrams, based on the system requirements validated by the director of the forest project. Subsequently, these models were implemented applying the n-tier style, developing the backend and frontend under the JavaScript, React, Mapbox, Node, Express, Sequelize and MySQL technologies, obtaining a software product agreed by the researcher. Finally, various validation, black box and functional tests were carried out, to the director of the forest research project who functionally validated the entire application, and the tests of the API-REST services for future application integrations.

Thus, the research question was reply, creating a web application that allows the user according to his role to manage the dasometric characteristics, morphological, ecological and ecosystem services of the grove of the city of Loja through interactive maps in the shortest possible time.

Keywords: Urban forest management, ICONIX, Software, React, Node, MySQL.

3. Introducción

En el Ecuador el Ministerio del Ambiente ampara los árboles de las ciudades a través de directrices¹ nacionales para la conservación, uso y manejo de los árboles en zonas urbanas, puesto que, es un elemento integrante del Patrimonio Natural del país, dichas directrices establecen que los municipios deben garantizar el mantenimiento y prevención del recurso forestal urbano, involucrando a la ciudadanía en su cuidado [1].

En las Instituciones de Educación Superior los proyectos de vinculación con la sociedad constituyen unas de las maneras de contribuir con soluciones a diferentes problemáticas sociales desde el ámbito académico e investigativo. En la Universidad Nacional de Loja, la vinculación con la sociedad en el marco de la investigación científica considera cualquier actividad científica o tecnológica dirigida a la sociedad que aporte a la mejora y actualización de planes de desarrollo local, regional y nacional, y la transferencia de conocimiento y tecnología [2]. En virtud de ello, en la Facultad Agropecuaria y de Recursos Naturales Renovables, Carrera de Ingeniería Forestal, nace el proyecto de investigación: “*Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja*” periodo 2021-2023, dirigido por el Dr. Darwin Alexander Pucha Cofrep, que busca en términos generales, fomentar la gestión elaborando y manteniendo actualizado el inventario de árboles dentro de sus respectivos límites, y promover la educación y sensibilización ambiental en los ciudadanos. Por ello, a través de su gestión se solicita a la Carrera de Ingeniería en Sistemas/Computación, la colaboración de un tesista que soluciones la problemática presente a través de una aplicación web.

Por lo anterior, el presente Trabajo de Titulación (TT) fue propuesto con el fin de dar respuesta a la pregunta de investigación: “**¿Cómo se podrá ayudar a los investigadores del Proyecto Forestal a obtener la información relacionada a los registros de árboles urbanos de la ciudad de Loja?**”. Para esto se estableció el objetivo principal “*Desarrollar una aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación, Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja*”. Y para conseguirlo, se desarrollaron tres objetivos específicos: “*Establecer el diseño de la Aplicación Web en base a la metodología ICONIX*”, “*Implementar el diseño de la aplicación mediante el framework ReactJS y NodeJS*”; y al final, “*Ejecutar pruebas de funcionalidad para la validación de la Aplicación Web*”.

¹ <https://www.ambiente.gob.ec/el-ministerio-del-ambiente-ampara-los-arboles-de-las-ciudades-del-ecuador-con-un-acuerdo/>

Como en [3], [4], existen propuestas de diseño o planes de trabajo que intentan resolver problemas similares, este TT aborda un enfoque único al trabajar con 4460 registros de árboles con 94 características individuales, entre ellas, dasométricas, morfológicas y ecológicas, que son gestionadas por la base de datos y la aplicación, para representar además de datos, sus puntos geográficos en mapas, pues otros trabajos relacionados al tema muestran menor esfuerzo al contener reducidas variables, sin analizar la posibilidad de almacenar datos de árboles a lo largo del tiempo, siendo ideal esta propuesta para valorar a futuro los registros individuales e impulsar futuras investigaciones.

En cuanto al desarrollo del Trabajo de Titulación se encuentra organizado de la siguiente manera:

La **sección Marco Teórico**, se encuentran los antecedentes referentes al tema y trabajos relacionados; la **sección Metodología** se detalla el contexto en donde se realizó el proyecto, el procedimiento y los recursos que fueron utilizados; la **sección Resultados** se compone de los tres objetivos específicos planteados para TT donde se evidencian: los requerimientos funcionales y no funcionales, el desarrollo de la metodología Iconix, donde se diseña y documenta la arquitectura lógica y física de la aplicación a través de los modelos de dominio, casos de uso, fichas de casos de uso, diagrama de secuencia, diagrama de clases, diagrama de componentes y diagrama de despliegue, la codificación aplicando los estilos arquitectónico n-tier tanto en el backend como en el frontend; y finalmente la ejecución de los pruebas de integración, caja negra y funcionales que comprueba el cumplimiento de los requerimientos funcionales y no funcionales de aplicación web; la **sección Discusión**, se analiza los resultados obtenidos de aplicar las fases de la metodología Iconix, el proceso ordenado que se llevó, las tecnologías que se adaptaron en el desarrollo de la aplicación y las pruebas ejecutadas; la **sección Conclusiones** abarca los acontecimientos más sobresalientes del proyecto, entre ellos se concluye que la Aplicación Web permitió consultar a través de mapas interactivos de Mapbox información incrustada de los árboles urbanos de la ciudad de Loja y gestionar los registros dasométricos, morfológicos y ecológicos; y finalmente en la **sección Recomendaciones** posee las sugerencias y posibles trabajos futuros aplicables para la continuidad del proyecto.

4. Marco Teórico

En esta sección se recoge la información bibliográfica que será la base teórica para la comprensión de los aspectos que se aborda en este Trabajo de Titulación.

4.1. Proyecto de vinculación con la sociedad

4.1.1. Contexto

Hace miles de años que los árboles han colonizando los espacios a lo largo y ancho del mundo. En el proceso han evolucionado para satisfacer sus necesidades y nos han acompañado desde el principio de nuestra corta historia coexistiendo y siendo fuente de recurso importante para desarrollo de las civilizaciones. Pero solo hace unas décadas se ha comenzado a mirarlos como una posibilidad de salvarnos de nosotros mismos. Que nuestra salud depende en cierta medida de la cantidad de árboles que tenemos a nuestro alrededor. De modo que se han visto obligados a repensar nuestras ciudades y apoyarse para ello de conceptos como arbolado urbano, silvicultura urbana, forestaría urbana y los bosques urbanos [5]. Pues estos temas son de interés público en el Ecuador, tanto que el Estado a través del Código Orgánico del Ambiente establece entre sus regulaciones, mecanismos que promuevan y fomenten la generación de información ambiental, así como actividades que fortalezcan la sostenibilidad, conservación, protección y restauración del ambiente, además de las regulaciones de los deberes y derechos que garanticen los fines expuestos [6].

Para la conservación de árboles en zonas urbanas, corresponde a la Autoridad Ambiental Nacional (MAE) y a las alcaldías la responsabilidad de formular e implementar las políticas públicas destinadas a la conservación, uso y manejo de los árboles en zonas urbanas. Así lo establece el Acuerdo 018 aprobado por el Ministerio del Ambiente que, con el Sistema Nacional Descentralizado de Gestión Ambiental, busca el ejercicio de gestión ambiental, para ello requieren de datos científicos y técnicos sobre la biodiversidad y el ambiente, que deberán ser recolectados y actualizados permanentemente.

Para ello es importante la participación de las instituciones de educación superior públicas, privadas y mixtas, al igual que otras instituciones de investigación, que se articulen a los organismos de control, para recopilar y compilar datos científicos y técnicos sobre la biodiversidad y el ambiente local.

4.1.2. Proyecto de investigación forestal

Actualmente en la Universidad Nacional de Loja se encuentra aprobado el proyecto de vinculación con la sociedad UNL-17-DI-FARNR-2021 denominado “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”, periodo 2021-2023, dirigido por el Dr. Darwin Alexander Pucha Cofrep, que busca en términos generales fomentar conciencia ambiental (la educación y sensibilización), específicamente, generar y facilitar el conocimiento sobre la dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja. Por ello, la propuesta del proyecto busca finalmente tener un inventario del arbolado urbano dentro de los límites establecidos, con datos generales y técnicos muy específicos como la dinámica de crecimiento de cada especie forestal, sus servicios ecológicos y su estado fitosanitario. Finalmente, disponer de la información no solo en los repositorios académicos, sino también que sea accesible a la ciudadanía en general.

Por este motivo el proyecto puntualiza tres objetivos específicos:

- Evaluar la dinámica de crecimiento anual e interanual de especies forestales urbanas de la ciudad de Loja, y su relación con la estructura física y química de la madera.
- Evaluar el estado fitosanitario y valoración ecosistémica del secuestro de carbono, belleza escénica y purificación de oxígeno de los árboles urbanos de la ciudad de Loja.
- Crear una aplicación móvil/web para consultar características dasométricas, morfológicas, y ecológicas de cada árbol urbano de la ciudad de Loja.

Y trabajará juntamente con los siguientes grupos de interés:

- MUNICIPIO DE LOJA:
 - Director del proyecto de Regeneración Urbana del Ilustre Municipio de Loja.
 - Coordinadora de patrimonio natural del Ilustre Municipio de Loja.
 - Jefa de Ambiente del Ilustre Municipio de Loja.
- UNIVERSIDAD NACIONAL DE LOJA
 - Dirección de Investigación
 - Carrera de Forestal
 - Carrera de Computación
 - Carrera de Agronomía
 - Carrera de Turismo
- CÁMARA DE TURISMO DE LOJA
- CIUDADANÍA DE LOJA

4.2. Metodología de desarrollo de software

Las metodologías de diseño y desarrollo de software son usadas para estructurar, planificar y controlar el proceso de desarrollo de los sistemas informáticos. Ayuda también a que el seguimiento de los proyectos sea ordenado, pues los lineamientos contribuyen a tener un control sobre los entregables, desde el análisis de requerimiento hasta las pruebas, haciendo más fácil el seguimiento de los proyectos [7].

Existen muchas metodologías de desarrollo de software, en este Trabajo de Titulación se utilizó una metodología de desarrollo ICONIX, considerada como una metodología pesada-ligera que se halla a medio camino entre un RUP (Rational Unified Process) y XP (eXtreme Programming).

4.2.1. Metodología ICONIX

Iconix maneja casos de uso como RUP, pero sin llegar a ese nivel de exhaustividad y, puede llegar a ser pequeño y firme como XP sin quitar el análisis y diseño que realiza en XP [8]. Utiliza un enfoque flexible y abierto, un proceso profundo en el seguimiento de requisitos que incluyen valiosas técnicas, y un juego mínimo de diagramas UML para codificar rápida y eficazmente [9].

Características de ICONIX

- Interactivo e incremental, se organiza en varias iteraciones las mini-tareas con el fin de refinar el modelo estático en modelos dinámicos.
- Trazabilidad, capacidad de seguir una relación entre los diferentes artefactos de software producidos.
- Dinámica del UML, ofrece un uso dinámico de diagramas UML, utiliza algunos diagramas sin exigir la utilización de todos.

Fases de ICONIX

Los procesos de la metodología Iconix, presentan las actividades de cada etapa y exhibe una secuencia de pasos que deben ser seguidos, constan de cuatro fases (Ver Figura 1): *Análisis de requisitos, Análisis y diseño preliminar, Diseño e Implementación* [10].

En cada fase implica un número de iteraciones de los cuales se obtienen entregables:

Fase 1: Modelo de dominio → Prototipos rápidos GUI → Modelo de casos de uso.

Fase 2: Fichas de casos de uso → Diagramas de robustez.

Fase 3: Diagramas de secuencias → Diagramas de clases → Diagramas de base de datos.

Fase 4: Codificación → Diagramas de despliegue → diagrama de componentes.

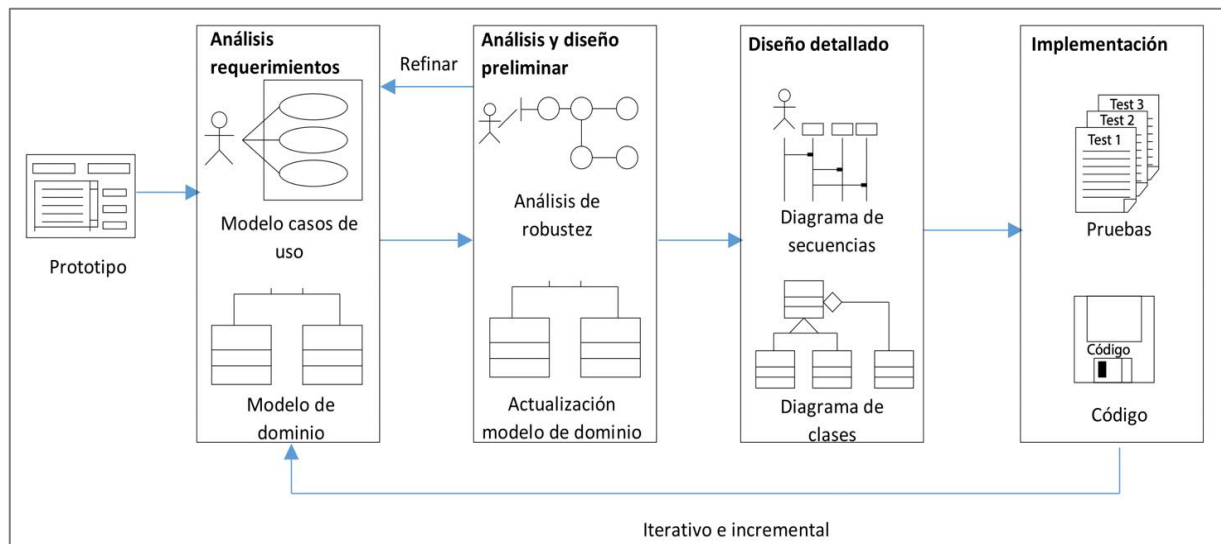


Figura 1. Descripción del proceso de Iconix.

Análisis de requisitos: Esta primera fase, los requerimientos del sistema proporcionados por el cliente se grafican en un prototipado inicial, de esto se identifican los objetos reales que interactúan con sistema para posterior representarlos en un modelado inicial o modelo de dominio de negocio, que de forma iterativa se presenta al cliente para refinarlos y comprender mejor los requerimientos del sistema que finalmente se diagrama en el modelo de casos de uso [11].

Análisis y diseño preliminar: Es la segunda fase, partiendo de los casos de uso se describen cada uno de ellos en las fichas de casos de uso. No se hace énfasis en especificar a profundidad, sino a que sea con brevedad, y en cada iteración se describa con sus precondiciones y postcondiciones [11]. Opcionalmente se realiza el diagrama de robustez para visualizar de mejor manera la interacción entre objetos y el sistema. Dicha ficha no corresponde al UML.

Diseño: Es la tercera fase, se realizan los diagramas de secuencia obtenidos desde cada ficha de caso de uso, modelando la interrelación entre usuarios, objetos, controladores y entidades. También se lleva a cabo el diseño del diagrama de clase y se define el modelo de datos que se implementará en el sistema [12].

Implementación: Es la cuarta fase, se creará el software escribiendo y generando el código fuente de acuerdo a las especificaciones realizadas en anteriores fases. Se sugiere realizar el diagrama de componentes y despliegue [12]. Finalmente, se realizan las pruebas teniendo en cuenta las exigencias planteadas en la primera fase para lograr la calidad del software que será entregado al cliente.

La metodología ICONIX resuelve el 80% del desarrollo de software utilizando solo un 20% de los modelos definidos en UML. No descarta la utilización de todos los modelos en caso de ser necesario, sino que sugiere la utilización de conjunto mínimo de modelos y un proceso dinámico para desarrollarlos. Además, promueve la participación de los usuarios con prototipos tempranos, descripción de casos de uso y en las pruebas del sistema [12].

4.3. Tecnologías de desarrollo de software

Existen una variedad de lenguajes de programación, frameworks y base de datos que facilitan el trabajo a los desarrolladores de software, entre estos se mencionan JavaScript, React, Node entre las combinaciones más populares [13], [14], [15].

4.3.1. JavaScript

*JavaScript*² es un lenguaje de programación ligero, interpretado, orientado del lado del cliente y también del servidor, usado por parte del cliente para hacer web más interactivas y para entornos fuera del navegador, en Node.js para crear backend [16]. JavaScript se ejecuta utilizando el motor V8 desarrollado por Google para su navegador Google Chrome. El uso de este mecanismo proporciona un entorno de ejecución del lado del servidor. Con esto, JavaScript se ejecuta y compila de manera más rápida y dinámica, y si se usa como único lenguaje mejora de la productividad de programación [17].

4.3.2. React

Considerada como una herramienta de JavaScript, *ReactJS*³ es una librería para el desarrollo de aplicaciones frontend utilizada para crear interfaces de usuario (UI) para grandes y pequeños proyectos. Representa la capa de vista en el patrón MVC. Se caracteriza por ser declarativa, basada en componentes, por lo que permite reutilizar elementos pasando parámetros de diferentes tipos de datos, funciones de forma encapsulada y encargándose de cambiar estados individualmente [18].

4.3.3. Node (Express)

*NodeJS*⁴ es un entorno que soporta el desarrollo de JavaScript del lado del servidor para desarrollar backend. Este entorno de ejecución se orienta a eventos asíncronos por lo que permite construir aplicaciones en red escalables, es decir, tiene la capacidad de realizar muchas conexiones de manera simultánea sin que tenga que leer el código línea a línea, ni abrir múltiples procesos [17]. No solo permite crear servidores web, sino que los hace más

² <https://developer.mozilla.org/es/docs/Web/JavaScript>

³ <https://reactjs.org>

⁴ <https://nodejs.org>

ágiles y capaces de trabajar con otros lenguajes e instalar bibliotecas y dependencias adicionales como Express.

*ExpressJS*⁵ es un framework que facilita y hace más cómodo el desarrollo de APIs de Node, aunque se considera minimalista, tiene mucha flexibilidad y alta capacidad de personalización, permite la libertad para crear paquetes y organizar la funcionalidad de la aplicación mediante middleware con el fin de resolver los problemas que surgen durante el desarrollo de una aplicación y enrutamiento para la organización de rutas y las cabeceras de respuesta adecuadas [17].

4.3.4. ORM (Object-Relational Mapping)

Un ORM es un modelo de programación que permite mapear las estructuras de una base de datos relacional, es decir, ayuda a gestionar la persistencia de datos en el RDBMS mediante el mapeo de las características de la base de datos a los objetos de la aplicación [19]. Esto permite simplificar y acelerar el desarrollo de aplicaciones ya que las acciones CRUD (Create, Read, Update, Delete) a ejecutar sobre la base de datos física se realizan de forma indirecta. ORM incluye todo, desde cómo se mapea el objeto a las columnas de la base de datos hasta cómo hacer consultas entre los objetos. Muchos desarrolladores prefieren tratar con datos persistentes al programar en lugar de tratar con sentencias SQL para recuperar datos en la base de datos. Esto ha llevado a que aumente la popularidad de los ORM, siendo Sequelize uno de los más populares en Node.

*Sequelize*⁶ es un ORM basado en promesas, compatible con varios dialectos: PostgreSQL, MySQL, MariaDB, SQLite y MSSQL. Se considera a sí mismo como Framework que permite realizar operaciones utilizando métodos JavaScript.

Al utilizar un ORM como Sequelize, dispondrá de la siguiente función sin necesidad de código adicional [19]:

- Gestión de transacciones
- Agrupación de conexiones
- Validación de modelos/datos
- Integración de datos (fuera del ámbito del SGBD de claves externas (FK), restricciones únicas, etc.)
- Carga dinámica
- Migración de esquemas y cascada

⁵ <https://expressjs.com>

⁶ <https://sequelize.org/>

- Bloqueo optimista

MySQL⁷ es un SGBDR o Sistema de Gestión de Base de Datos Relacional y está soportado por lenguajes como JavaScript a través de ORM Sequelize.

4.4. Servidor de mapas

Tradicionalmente, los mapas digitales se renderizaban en el servidor y se envían como imágenes al cliente, pero la nueva tecnología web abre posibilidades de renderización dinámica de mapas en un navegador web (cliente) [20]. Este renderizado permite realizar tareas de representación de coordenadas, permitir a los usuarios sobreponer sus propios datos espaciales, como vectores (puntos, líneas o polígonos) y ráster (malla de píxeles como imágenes de satélites) sobre los mapas base. Para ello, existen distintos proveedores de mapas (ver Tabla 1) en línea que ofrecen su API para la integración con otras aplicaciones o software [21].

Tabla 1. Servidores de mapas web

Servidores de mapas	Descripción	Es	Cualidad
OpenStreet Map	Crea y ofrece datos geográficos libre al mundo.	Código abierto	Mapa vectorial (puntos, líneas y polígonos más atributos asociados a descripciones o etiquetas)
ESRI	Unos de los más completos, administra varios tipos de mapas.	Sin costo mientras no exceda el límite de transacciones al mes.	Mapa vectorial, ráster, rutinas de geoprocetos, geoconsultas, análisis espacial, grabar en la nube las coberturas generadas por los usuarios.
Google Maps	Ofrece imágenes de satelitales y de terreno.	Sin costo mientras no exceda el crédito mensual.	Mapas de tráfico en tiempo real, rutas de transporte público, calculador de rutas, imágenes a pie de calles y navegación con GPS.
Mapbox	Contiene ejemplos y tutoriales para usar utilizar todas sus herramientas.	Sin costo mientras no exceda el límite de peticiones mensual.	Mapas dinámicos vectoriales, mapas ráster, navegación inteligente basado en el tráfico en tiempo real, desarrollo de atlas personalizado y búsqueda de sitios.

4.4.1. Mapbox

Mapbox es un servidor que proporciona herramientas de cartografía digital que permite personalizar mapas en las aplicaciones de forma rápida y sencilla [20]. *Mapbox GL JS*⁸ es una librería JavaScript del lado del cliente (renderizado del lado del usuario) que utiliza la tecnología cartográfica de Mapbox.

⁷ <https://mysql.com/>

⁸ <https://mapbox.com/mapbox-gljs>

En términos de formato de los datos, el usuario puede importar en Mapbox tilesets o bien crea datasets. Los tilesets son datos vectoriales o normalizados que el usuario puede editar en cuanto a su estilo, es decir, colores, transparencia, nomenclatura, simbología y lo dataset son los datos que el usuario desea cargar y entran en la base de datos en la nube de Mapbox, para convertirse automáticamente en conjuntos de mosaicos. Los tilesets vectoriales se derivan de muchos tipos de archivos como GeoJSON, Shapefiles.

Mapbox GL JS puede modificar los controles, disposiciones de elementos de navegación, zoom en el mapa, añadir y personalizar marcadores, implementar ventanas emergentes (pop-ups) y demás. Asimismo, se pueden añadir capas de información geográfica vectoriales en formatos GeoJSON [22]. Cada capa proporciona normas sobre cómo el renderizador debe dibujar ciertos datos en el navegador para mostrar en el resultado en los mapas de Mapbox.

Mapas de Mapbox

Mapbox destaca el diseño de cinco mapas de ámbito profesional para su uso: Luz, Oscuro, Calles, Al aire libre e Imágenes de satélite. En la Figura 2, se observa los mapas diseñados por Mapbox.

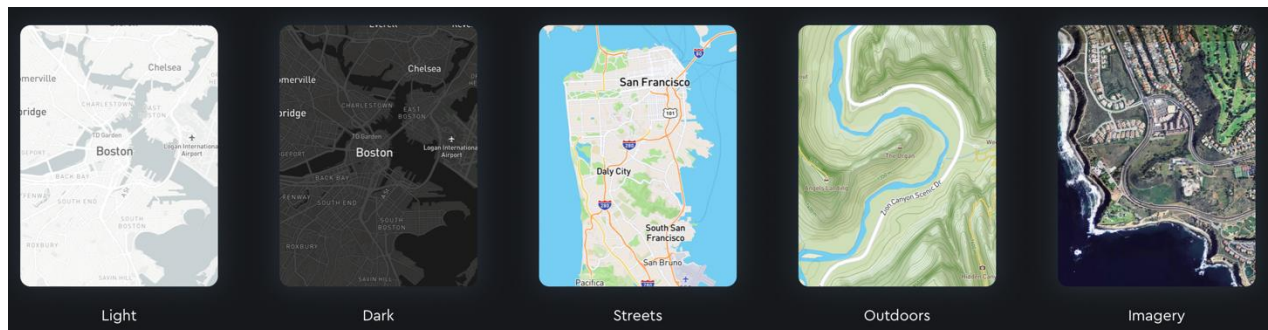


Figura 2. Mapas de Mapbox [23].

Costos Mapbox GL JS

La potencia de Mapbox y su conjunto de soluciones para web mapping tiene un coste. Se mantiene políticas de planes y precios para el uso de la plataforma de manera escalable. Para cualquier aplicación web y móvil, el límite de pago se establece a partir de las 50.000 peticiones por mes. Se cuenta como una carga de mapa cada vez que Mapbox GL JS se inicializa en una página o aplicación web. Una carga de mapa incluye solicitudes ilimitadas de API de mosaicos vectoriales y API de mosaicos ráster. Mapbox establece un cobro de 0,50\$ por cada 1.000 peticiones de cualquier tipo a los distintos servicios [23].

Para usar este servicio es necesario obtener un token de acceso para usarlo en cualquiera de las herramientas, API, o SDK de Mapbox. Mapbox utiliza tokens para asociar una cuenta con solicitudes a los recursos de la API y evaluar la cantidad de peticiones que se realiza. Los tokens de acceso tienen ámbitos públicos o secretos. Su cuenta tendrá al menos un token de acceso público.

En general, cuando se pide al diseñador que implemente un diseño para una aplicación sencilla, la biblioteca Mapbox GL JS puede resultar bastante especializado (es decir, el proyecto podría más fácilmente utilizando Google Maps, OpenStreetMaps). Sin embargo, entre más compleja se hace la gestión de los datos que es procesado por el usuario y más personalizada tiene que ser la solución a implementar por el diseñador, más apropiada será la biblioteca Mapbox GL JS [20]. A menudo, el uso de Mapbox GL JS puede resultar más complicado, pero suele ser ideal para crear aplicaciones de visualización de datos desde cero.

4.5. Trabajos relacionados

En esta sección se resume los trabajos relacionados a las aplicaciones web para la gestión arbórea, y otras propuestas respecto a la gestión forestal con el enfoque principal diseñar una solución informática.

“Desarrollo de una página web para el control del arbolado del municipio de Fusagasugá”

Este trabajo de titulación surgió de la necesidad por parte de la alcaldía de Fusagasugá de registrar los árboles del sector de Fusagasugá, para llevar un censo de ellos y mostrar a la comunidad por medio de un mapa donde está ubicado el árbol y una ficha técnica que evidencie las características principales. El objetivo principal del proyecto es desarrollar una página Web para el control del arbolado del Municipio de Fusagasugá. Para el desarrollo de la solución informática se utilizó la metodología XP (eXtreme Programming). Se utilizó el motor de mapa MapBox el framework de Laravel, para ello se codificó la conexión por medio del lenguaje de JavaScript, se utilizó la base de datos PostgreSQL y el Frontend fueron diseñadas con Balsamiq Mockups 3. Para la arquitectura del sistema trabajó con los controladores y servicios en el lenguaje de PHP y como gestor de base de datos PostgreSQL. El trabajo concluye con el desarrollo de una aplicación web que cumple con 14 requerimientos funcionales, entre los cuales incluyen la gestión de familia, especies, gestión de algunas características dasométricas, riesgos y todos estos se resumen (a través de una tabla) en una sección (Arbolado) que muestra los registros actuales, registrando los nuevos árboles a través de varios formularios [4].

“Aplicación web para la gestión de inventario de la Estación Biológica Pindo Mirador del Gobierno Autónomo Descentralizado Provincial de Pastaza”

El proyecto presenta en el contexto el vivero donde llegan temporalmente semillas, plántulas extraídos de los bosques para su cuidado y que una vez que se han desarrollado son introducidos en los bosques. No cuenta con método de registro de la información en el cual garantice la fiabilidad y facilidad de manejo de los datos. Este proyecto se plantea resolver la pregunta de investigación ¿Cómo mejorar la gestión de Inventario en la Estación Biológica Pindo Mirador del Gobierno Autónomo Descentralizado Provincial de Pastaza? Y para ello utiliza la metodología de trabajo Scrum, con tres fases: en la primera, referente al Análisis, realiza la identificación de los requisitos funcionales y no funcionales, modela los casos de uso, describe las fichas de casos de uso, modela los diagrama de secuencia; en la segunda, referente al Diseño, realiza el diagrama de clases, modelo de datos y base de datos; Finalmente, aplicando la arquitectura cliente-servidor y patrón de diseño MVC se desarrolla la aplicación web en PHP con Bootstrap y utiliza como base de datos PostgreSQL. La pruebas funcionales realizadas validaron que se puede gestionar el inventario arbóreo para esto se usó la aplicación para registrar el ingreso de 100 plantas, editar sus campos, realizar el egreso de 37 plantas, visualizar información en una tabla [24].

“Aplicación web para la gestión de la información del inventario forestal a ser implementado en un servidor web dentro de la intranet de la Dirección Forestal del Ministerio del Ambiente”

El presente trabajo de titulación tiene como objetivo principal crear una aplicación web que facilite la gestión de la información del Inventario Nacional Forestal dentro de una base de datos, crear una aplicación móvil bajo plataforma Android - Java, que implemente las interfaces de ingreso de datos con los formularios utilizados para el levantamiento del Inventario Nacional Forestal y presentar reportes con la información referente a biodiversidad (especies). Para alcanzar este objetivo se emplea la metodología XP, con ella se desarrolló la aplicación web utilizando como lenguaje de programación PHP, servidor Apache, gestor de base datos PostgreSQL y para la parte móvil se utilizó Java con SQLite. También utiliza servicios web de Google Map para representar los datos de vegetación. Entre las pruebas realizadas en el proyecto, fueron de carga y estrés, de caja blanca, de caja negra que demostraron cumplir con los requerimientos para los dos módulos especificados [25].

5. Metodología

El presente Trabajo de Titulación (TT) pertenece al tipo de Investigación Aplicada, debido a que los recursos utilizados para llevar a cabo este proceso permiten aplicar más allá de conocimientos informáticos teóricos y centrarse en una solución informática práctica que tenga como resultado un producto de software, como es el caso de este TT, para la gestión del inventario de árboles urbanos de la ciudad de Loja.

La metodología de desarrollo de software ejecutada en el TT es ICONIX, esta sugiere artefactos que pueden ser utilizados en cada una de sus fases para el modelado de un sistema.

A continuación, un resumen de las fase ejecutadas para el TT:

Tabla 2. Resumen de la Metodología ICONIX utilizada en el TT

Fase	Actividad	Artefactos	Técnica	Herramienta
PREVIA	Identificación de los requerimientos	Requerimientos Funcionales Requerimientos no funcionales	Entrevista al Director del Proyecto Forestal. Análisis de la documentación.	Grabadora de voz, Microsoft Word 2019
FASE 1: Análisis de Requisitos	Identificación de los objetos y todas las relaciones de agregación y generalización.	Modelo de Dominio	Análisis de la documentación.	StarUML v5.0.2
	Prototipado rápido de las interfaces de la aplicación	Prototipo de interfaz de usuario	Análisis de la documentación. Diseño del prototipo.	Figma Desktop v116.5.188
	Identificación de los casos de uso y actores involucrados	Modelo de Casos de uso	Análisis de la documentación. Revisión y validación del modelado.	StarUML v5.0.2, Microsoft Word 2019
FASE 2: Análisis y Diseño Preliminar	Descripción de las Casos de uso	Ficha de Casos de uso		
FASE 3 Diseño	Especificación del comportamiento	Diagrama de secuencia	Análisis de la documentación. Revisión y validación del modelado.	StarUML v5.0.2, Microsoft Word 2019
	Identificación de relaciones	Diagrama de clases		

	Generación del modelo de datos	Diagrama de Datos		MySQL Workbench v8.029
FASE 4 Implementación	Generación del código de la aplicación	Código	<p>Análisis de la documentación.</p> <p>Desarrollo del backend y frontend.</p> <p>Repositorio del Sistemas de control de versiones (GitHub).</p>	<p>Visual Studio Code v1.73.1, Postman v10.5.2, MySQL Workbench v8.029, GitHub.</p> <ul style="list-style-type: none"> • JavaScript • NodeJs • ExpressJs • ORM Sequelize • ReactJS • MaterialUI • Mapbox.
	Diseño de arquitectura, descripción de los elementos físicos y sus relaciones.	Diagrama de componentes	Análisis de la documentación. Revisión y validación del modelado.	StarUML v5.0.2
		Diagrama de despliegue		
Pruebas de funcionalidad	Informe de pruebas	Ejecución de pruebas unitarias, caja negra y funcionales la aplicación.	Browser, Postman.	

En la sección 5.1 se introduce al contexto y área de estudio donde se ejecutó del trabajo de titulación; en la sección 5.2 se describe los procedimientos llevado a cabo para el cumplimiento de los objetivos planeados; finalmente se especifica en la sección 5.3 los diferentes recursos que fueron empleados.

5.1. Contexto

El trabajo de titulación se llevó a cabo en el cantón Loja, en la Carrera de Ingeniería en Sistemas/Computación de la Facultad de Energía, las Industrias, y los Recursos Naturales No Renovables, conjuntamente con la Carrera de Ingeniería Forestal de la Facultad de Agropecuaria y de Recursos Naturales Renovables, pertenecientes ambas a la Universidad Nacional de Loja. Lugar donde se dio a conocer sobre el proyecto de investigación “Dinámica de crecimiento y servicios ecosistémico del arbolado urbano de la ciudad de Loja” dirigido por el Director Darwin Pucha Cofrep y su equipo de investigadores, quienes a través del oficio Of.No.189-DAPC-FARNR-UNL solicitan un tesista para cubrir con las necesidades de gestión del inventario de árboles de la zona urbana de la ciudad de Loja (ver Anexo 1).

En este contexto se realizó el acercamiento al proyecto y se planteó el presente TT, cuyo objetivo general es “Desarrollar una aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”, donde se formuló tres objetivos específicos los mismo que se detallan en la siguiente sección.

5.2. Procedimiento

El proceso de la investigación se detalla a continuación, mencionando cada uno de los objetivos con sus actividades:

5.2.1. Objetivo 1. Establecer el diseño de la Aplicación Web en base a la metodología ICONIX.

- Entrevista al Director de Proyecto Forestal Ing. Darwin Pucha Cofrep (**ver Anexo 2**).
- Revisión de la documentación física y digital del inventario de árboles (**ver Anexo 3**).
- Identificación de los requisitos funcionales y no funcionales (**ver Sección Resultados, Objetivo 1, Ítem 6.1.1**).
- Elaboración de las tres primeras fases de la metodología ICONIX (**ver Sección Resultados, Objetivo 1, Ítem 6.1.2**)

5.2.2. Objetivo 2: Implementar el diseño de la aplicación mediante el framework ReactJS y NodeJS.

Desarrollo de la fase de Implementación correspondiente a la cuarta fase de la metodología ICONIX. El proceso realizado para alcanzar este objetivo es el siguiente:

- Selección de herramientas, levantamiento para el entorno de desarrollo y utilización del repositorio de control de versiones Git (**ver Sección Resultados, Objetivo 2, Ítem 6.2.1**).
- Codificación del backend y frontend de acuerdo a los modelos y prototipos obtenidos de las tres primeras fases Iconix. (**ver Sección Resultados, Objetivo 2, Ítem 6.2.1.1**).
- Documentación de la arquitectura de la aplicación web (**ver Sección Resultados, Objetivo 2, Ítem 6.2.2**)

5.2.3. Objetivo 3: Ejecutar pruebas de funcionalidad para la validación de la Aplicación Web

- Se desarrolla las pruebas unitarias de API-REST de la aplicación web con la herramienta Postman (ver **Sección Resultados, Objetivo 3, Ítem 6.3.1**).
- Se evaluó la aplicación en un ambiente de control, donde se realizaron las pruebas unitarias, pruebas de caja negra y pruebas de funcionalidad (ver **Sección Resultados, Objetivo 3, Ítems 0, 6.3.4 y 6.3.4**).

5.3. Recursos

Para cumplir con los objetivos planeados se utilizaron los siguientes recursos:

5.3.1. Recursos Científicos

5.3.1.1. Experimentación

Permitió evaluar la aplicación en un ambiente simulado, con el fin de determinar si el desarrollo de la solución informática cumple con los requisitos funcionales y no funcionales.

5.3.2. Recursos Técnicos

5.3.2.1. Entrevista

La entrevista permitió la recolección de información sobre las necesidades del proyecto forestal y sobre la gestión actual del inventario de árboles urbanos en la ciudad, de esta manera se pudo obtener la especificación de requerimientos funcionales y no funcionales para la aplicación web (ver entrevista, **Anexo 2**).

5.3.2.2. Metodología de desarrollo de software

La metodología empleada para el desarrollo de la aplicación web fue ICONIX, misma que cuenta con cuatro fases que van desde el análisis hasta la implementación, construyendo en camino la documentación de los modelos empleado para la construcción del producto de software. Para conocer más detalle sobre metodología ICONIX, visitar **Marco Teórico, Metodología ICONIX**.

5.3.2.3. Revisión Bibliográfica

Permitió la recolección de información para la validez y sustentación científica del fundamento teórico del presente TT, las fuentes bibliográficas se basaron en libros, revistas y documentos científicos.

5.3.3. Recursos humanos

En el presente Trabajo de Titulación (TT) estuvieron presentes los siguientes participantes:

- Johnny Fabián González Guamán como investigador del TT.
- Ing. Wilman Patricio Chamba Zaragocín como director de este TT y revisor durante todo el proceso, desde el inicio del anteproyecto hasta su culminación.
- Ing. Darwin Pucha Cofrep como Director del proyecto de investigación, Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja y testeador de la aplicación web.
- Otros investigadores del proyecto forestal para la fase levantamiento de información.

5.3.4. Recursos de Hardware y Software

5.3.4.1. Hardware

- **Laptop MacBook Pro:** Dispositivo para el desarrollo de TT.

5.3.4.2. Software

- **Visual Studio Code:** Como entorno para el desarrollo, usado para la codificación e implementación de la aplicación.
- **Postman:** Para el desarrollo, pruebas y documentación de APIs.
- **GitHub:** Repositorio de control de versiones en el desarrollo del presente TT.
- **MySQL Workbench:** Para la administración de tablas en la base de datos.
- **Mendelay:** Para la gestión bibliográfica.
- **StartUML:** Para el modelado de diagramas especificados en la metodología ICONIX.
- **Figma:** Para el diseño del prototipo inicial de la aplicación.

5.3.4.3. Tecnologías

- **MySQL:** Como gestor de base de datos relacional para la aplicación.
- **JavaScript:** Como lenguaje de programación para el frontend.
- **NodeJS:** Como entorno de ejecución multiplataforma.
- **React:** Como librería/framework para el desarrollo de la aplicación web frontend.
- **ExpressJS:** Como framework para el desarrollo del servidor web o backend.
- **ORM Sequelize:** Como mapeador relacional de objetos para NodeJS y conexión con la base de datos MySQL.
- **Railway:** Como plataforma utilizada para el alojamiento gratuito del backend y la base de datos.
- **Netlify:** Como plataforma utilizada para el alojamiento gratuito del frontend.

6. Resultados

En este apartado se muestran las fases, actividades y tareas que se llevaron a cabo para desarrollar el presente Trabajo de Titulación.

En la primera fase se obtuvo el diseño lógico y físico de la aplicación mediante el desarrollo de la metodología de software ICONIX, en la segunda fase se implementó los diseños aplicados en la primera fase y se desarrolló la infraestructura del backend y frontend de la aplicación web, finalmente, en la tercera fase se realizó pruebas al software para validar la aplicación web.

6.1. Objetivo 1: Establecer el diseño de la Aplicación Web en base a la metodología ICONIX.

Se realizó dos tareas, en primer lugar, se elaboró el modelo de requerimientos del sistema y, en segundo lugar, se realizó el modelo lógico y físico de la aplicación web, ambas tareas corresponden a la primera, segunda y tercera etapa en la ejecución de la metodología de desarrollo de software ICONIX, los cuales se obtuvo los siguientes resultados.

6.1.1. Elaborar el modelo de requerimientos del sistema.

Se realizó una entrevista al director de proyecto forestal Ing. Darwin Pucha Cofrep, el cual respondió inicialmente a tres preguntas abiertas. La entrevista fue grabada haciendo uso del dispositivo celular y posteriormente transcrita para su análisis (ver **Anexo 2**).

Se analizó la documentación transcrita y se identificaron los requisitos funcionales (ver **Tabla 3**) y los no funcionales (ver **Tabla 4**).

Tabla 3. Requerimientos Funcionales

ID	Requisito	Descripción
RF01	Visualizar árboles	El sistema debe permitir al usuario visualizar la ubicación de los registros de árboles en el aplicativo a través de un mapa web de Mapbox.
RF02	Mostrar tipos de mapas base	El sistema debe permitir al usuario cambiar entre diferentes vistas de mapas base del servidor Mapbox: Satelital y de Calles.
RF03	Visualizar reportes	El sistema debe permitir al usuario visualizar dos tipos de reportes: a) Un reporte preliminar con los detalles mínimos del árbol seleccionado en el mapa. b) Un reporte completo para el mismo árbol seleccionado. El sistema debe permitir mostrar el identificador del árbol en un código QR.
RF04	Filtra consultas	El sistema debe permitir al usuario filtrar las búsquedas de árboles por especies, sitio y características, considerada como las variables más relevantes.

RF05	Autenticar usuario	El sistema debe permitir a los usuarios registrados ingresar a la aplicación a través de las credenciales como: nombre de usuario (email) y contraseña.
RF06	Actualizar contraseña	El sistema debe permitir al usuario modificar su contraseña desde su cuenta de usuario y se enviará la contraseña modificada al email.
RF07	Recuperar contraseña	En caso de olvido de contraseña, el sistema debe ayudar al usuario a recuperar el acceso a su cuenta, para ello el sistema validará el email ingresado y se enviará un correo electrónico para el cambio de contraseña.
RF08	Editar árboles	El sistema debe permitir al usuario registrado editar las características de un árbol específico. Para los campos de latitud y longitud, además de digitar manualmente, se debe contar con la opción de desplazar el punto de interés a la nueva posición.
RF09	Registrar árbol	El sistema debe permitir al usuario registrado agregar nuevos árboles y llegar los campos en un formulario de registro.
RF10	Eliminar árbol	El sistema debe permitir al usuario Administrador eliminar un árbol del registro forestal (se espera que la información vinculada con el registro del árbol no se muestre al usuario final).
RF11	Descargar reportes	El sistema debe permitir al usuario Administrador descargar el reporte de uno o varios árboles seleccionados en un archivo CSV.
RF12	Registrar cuenta	El sistema debe permitir al usuario registrarse para crear una cuenta en la aplicación web.
RF13	Validar cuenta	El sistema debe permitir al administrador validar cuenta de usuario registrado.

Tabla 4. Requerimiento No Funcionales

ID	Requisitos	Descripción
RNF01	Seguridad	La información manejada por el sistema estará protegida del acceso no autorizado a través de roles. Un usuario con un rol específico no podrá realizar acciones que no correspondan al rol asignado.
RNF02	Usabilidad	La aplicación mantendrá una interfaz intuitiva, de fácil uso para el óptimo manejo del usuario. El tiempo de aprendizaje del sistema por usuario será menor a 2 horas.
RNF03	Rendimiento o Calidad	La funcionalidad del sistema, informes y consultas que presenten una complejidad mediana, sus tiempos de respuesta hacia el usuario debe responder en menos de 7 segundos.
RNF04	Fiabilidad	El sistema debe garantizar la tolerancia a fallos de conexión proporcionando mensajes informativos y orientados a los usuarios.
RNF05	Persistencia	El sistema debe garantizar la recuperación y guardado de los datos persistentes utilizados por la aplicación en la base de datos relacional MySQL.
RNF06	Sincronización	Los datos modificados o agregados en el sistema deben ser actualizados por todos los usuarios que accedan a ellos en menos de 3 segundos.
RNF07	Disponibilidad	El sistema debe estar activo 24 horas, los 7 días de la semana, los 365 días del año, garantizando toda la funcionalidad del sistema.
RNF08	Mantenibilidad	Será desarrollado de forma modular para mantener una estructura de código consistente y predecible.
RNF09	Desarrollo	Debe funcionar a través de respuestas consumiendo servicios web API REST bajo el formato de datos JSON.
RNF10	Interfaz GUI	La interfaz se ejecutará en entornos web, se utilizará un browser habilitado con el estándar WebGL para el despliegue de gráficos 2D/3D, como por ejemplo Chrome 96, Edge 96, Firefox 97, Firefox 91 (ESR), iOS Safari 14, Safari 14, o versiones posteriores a las especificadas.

6.1.2. Elaborar el diseño lógico y físico de la aplicación.

Se realizaron las 3 primeras fases de la metodología ICONIX:

En Análisis de Requisitos, se diseñó el Diagrama conceptual del modelo de dominio de negocio, Prototipo de interfaz de usuario y Diagrama de casos de uso.

En Análisis y Diseño Preliminar, se elaboró la descripción de las Fichas de casos de uso.

En la fase Diseño, se elaboró el Diagrama de secuencia, el Diagrama de clases y se realizó el Diagramas de Datos de la aplicación.

Se proporcionó documentación como fichas técnicas para el levantamiento de árboles en campo utilizados por ellos (formato físico) y el inventario de árboles levantados hasta fecha de corte 15 de Julio de 2022 (formato digital Excel), el mismo que fue base el desarrollo de esta sección.

A. Análisis de requisitos

Se obtuvo como resultado (ver **Anexo 4, FASE 1: Análisis de Requisitos**) los siguientes artefactos:

- Diagrama Conceptuales del Modelo de dominio del negocio (ver **Figura 3**).
- Prototipo de interfaz de usuario (ver **Figura 4**, y más detalles en **Anexo 4**, Ítem **2.3, FASE 4**).
- Diagrama de Casos de Uso (ver **Figura 5**)

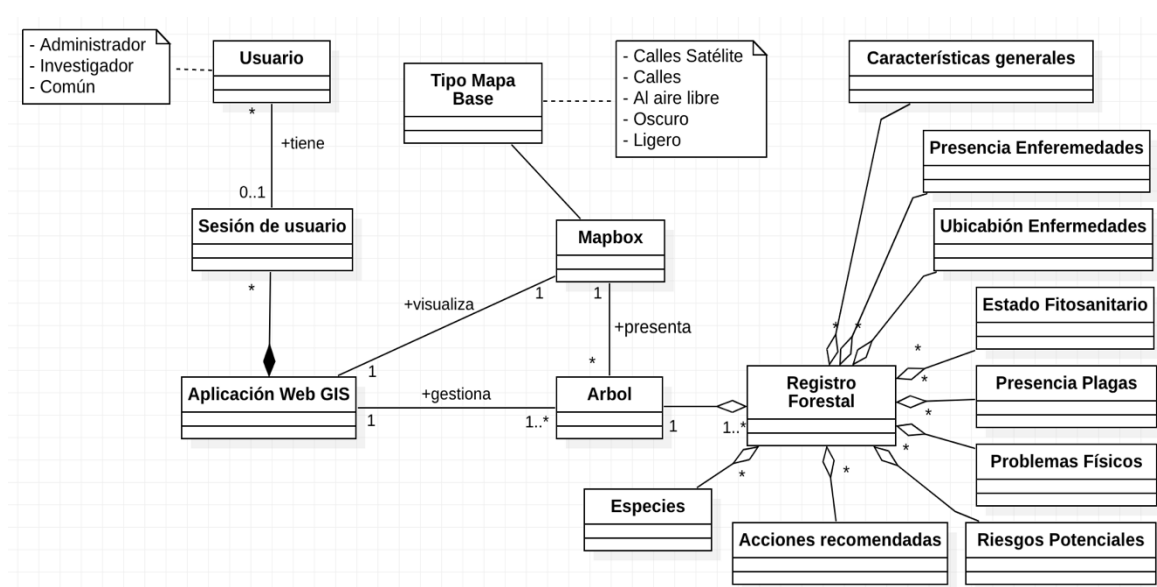


Figura 3. Modelo de dominio del negocio

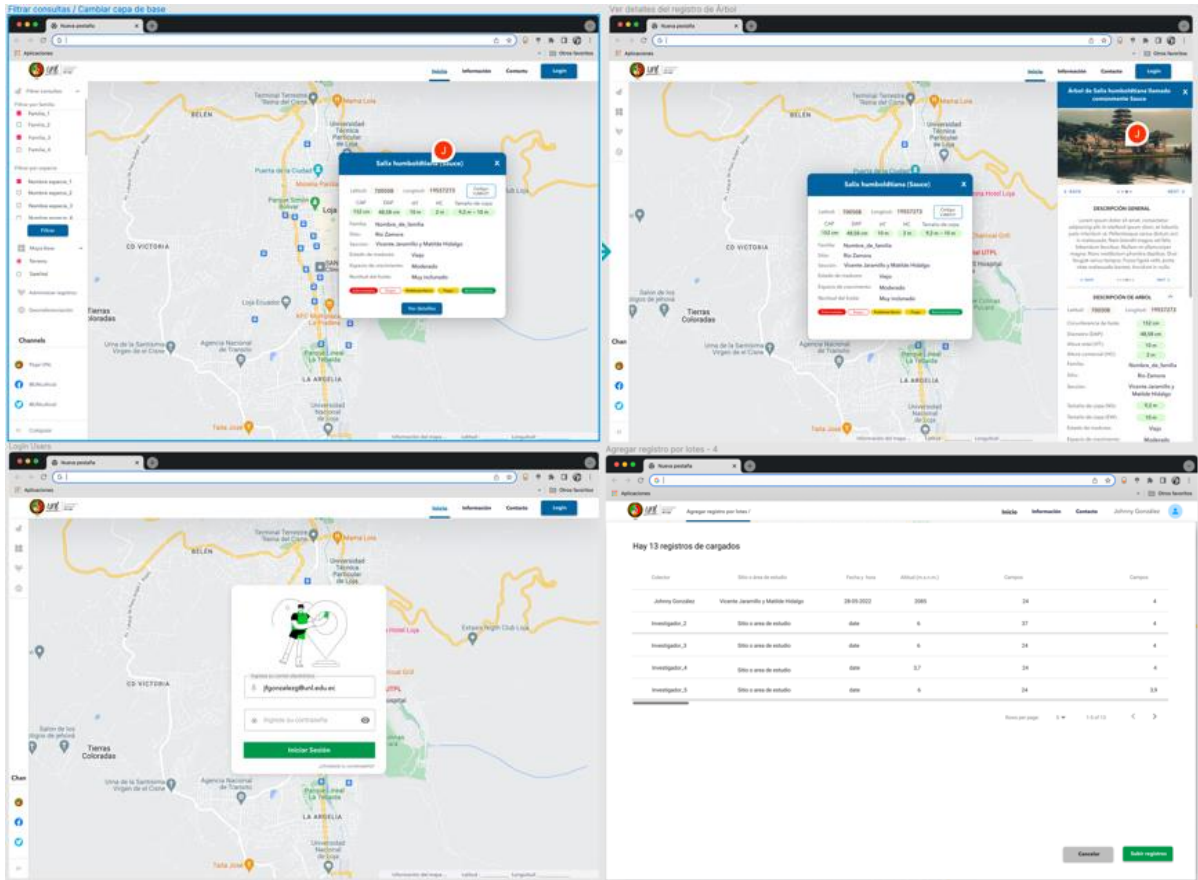


Figura 4. Prototipo rápido de GUI en Figma.

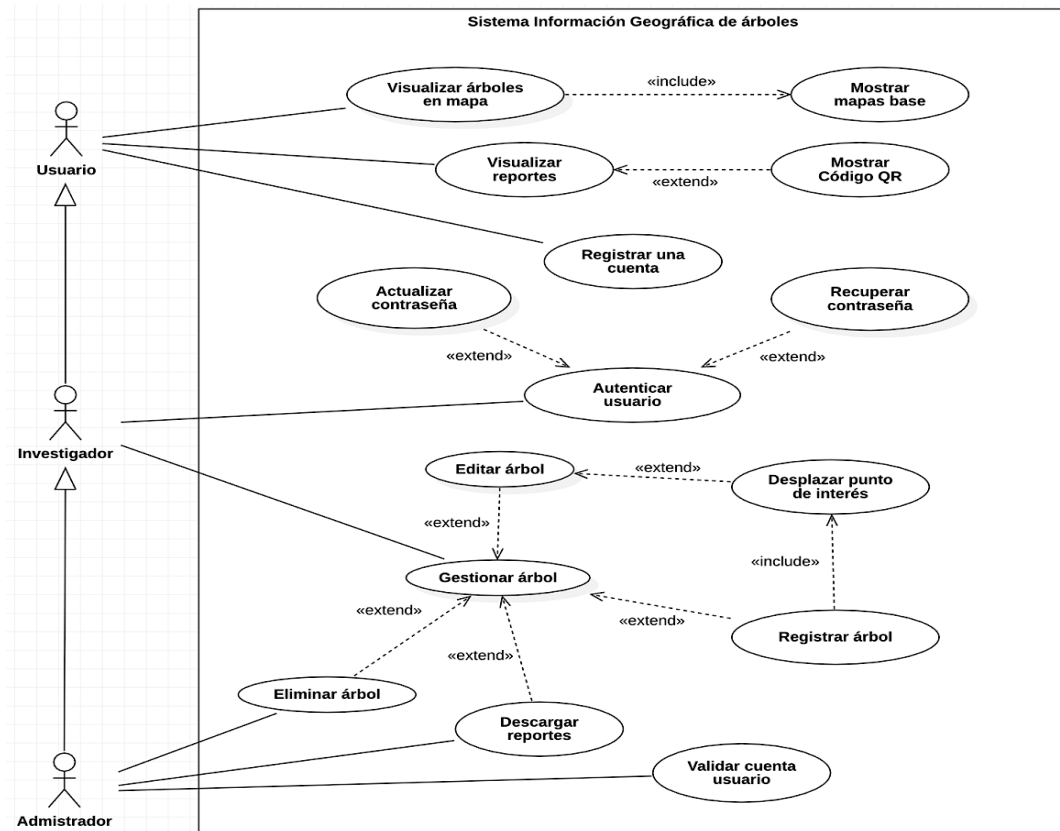


Figura 5. Diagramas de caso de uso de la Aplicación Web.

B. Análisis y Diseño Preliminar

Se realizó la descripción de las fichas de casos de uso, las mismas puede ver en **Anexo 4, FASE 2: Análisis y Diseño Preliminar**. A continuación, se muestran cuatro casos principales:

- Visualizar árboles (ver **Tabla 5**)
- Visualizar reportes (ver **Tabla 6**)
- Gestionar árbol – Editar (ver **Tabla 7**)
- Gestionar árbol – Descargar Reporte (ver **Tabla 8**)

Tabla 5. Fichas de caso de uso Visualizar árboles

CU01	Actores: Usuario/Administrador/Investigador	Prioridad: Alta
Caso de Uso: Visualizar árboles en mapa		
Descripción: El usuario visualiza los registros de árboles como puntos de interés en el mapa de Mapbox y cambia el mapa base de terreno (calles) a satelital.		
Referencias: RF01, RF02.		
Pre-Condiciones:		
- El usuario se encuentra en la página principal del aplicativo web.		
Post-Condiciones:		
- Se alterna dinámicamente el estilo del mapa a visualizarse.		

- El sistema representa las ubicaciones de árboles como puntos de interés en el mapa con un color identificativo.

Flujo normal de los eventos:

Usuario no autenticado

1. El usuario visualizar un mapa web de Mapbox.
 2. El sistema presenta grupos de árboles (puntos de interés) en el mapa.
 3. El usuario hace zoom en el mapa.
 4. El sistema carga más puntos de interés en el mapa.
 5. El usuario selecciona el mapa base que desea utilizar: Terreno o Satelital.
 6. El sistema actualiza el estilo del mapa.
-

Flujo Alternativo de eventos:

Tabla 6. Ficha de caso de uso Visualizar reportes

CU02	Actores: Usuario/Administrador/ Investigador	Prioridad: Alta
Caso de Uso: Visualizar reportes		
Descripción: El usuario desea obtener información de un árbol específico (punto de interés).		
Referencias: RF03, RF04.		
Pre-Condiciones:		
<ul style="list-style-type: none"> - El usuario debe aplicar la búsqueda de árboles según el tipo de información que desea visualizar. - El usuario debe seleccionar un registro de árbol en el mapa interactivo. 		
Post-Condiciones: El sistema realiza consultas y presenta reportes con diferentes niveles de detalles.		
Flujo normal de los eventos:		
Visualizar reporte preliminar		
<ol style="list-style-type: none"> 1. El usuario debe selecciona el árbol de interés en el mapa. 2. El sistema presenta un reporte preliminar con información básica del árbol. 3. El usuario selecciona mostrar código QR. 4. El sistema presenta el código QR con la identificación del árbol. 5. El usuario selecciona el formato imagen para descargar el código QR. 6. El sistema descarga en el formato imagen en el equipo. 		
Visualizar reporte completo		
<ol style="list-style-type: none"> 7. El usuario selecciona el botón Ver más. 8. El sistema presenta un reporte individual con información completa del árbol. 		
Flujo Alternativo de eventos:		

Tabla 7. Ficha de caso de uso Editar árbol

CU04	Actores: Administrador/ Investigador	Prioridad: Alta
Caso de Uso: Gestionar árbol – Editar árbol		
Descripción: El usuario puede editar las características de un árbol, los campos de latitud y longitud, se puede desplazar el punto de interés a una nueva ubicación en el mapa.		
Referencias: RF8.		
Pre-Condiciones:		
<ul style="list-style-type: none"> - El usuario Administrador e Investigador deben iniciar sesión y en la sección del menú Administrar registro. - El usuario debe haber buscado el árbol específico para visualizar las acciones disponibles. 		

Post-Condiciones: Se actualiza el árbol seleccionado con los nuevos valores dentro de la aplicación y el usuario visualiza los campos actualizados.

Flujo normal de los eventos:

1. El usuario selecciona de la lista de acciones, Editar.
 2. El sistema obtiene la información correspondiente al árbol seleccionado y lo muestra en un formulario para su edición con los campos bloqueados.
 3. El usuario hace clic en "Habilitar edición".
 4. El sistema habilita los campos para editar y muestra los botones de guardado.
 5. El usuario actualiza uno o varios campos.
 6. El sistema verifica que el formato del dato ingresado es correcto.
 7. El usuario presiona Guardar.
 8. El sistema almacena los datos modificados de forma permanente y presenta un mensaje de guardado correctamente.
-

Flujo Alternativo de eventos:

A. Formato de dato ingresado inválido

[6.1.] El sistema muestra un mensaje de validación indicando que el formato está incorrecto.

B. Usuario Cancela la acción editar

[7.1.] El sistema no almacena los datos editados y redirecciona al inicio.

C. Actualizar campo latitud y longitud

[5.1.] El usuario selecciona la acción Desplazar ubicación para latitud y longitud.

[5.2.] El sistema habilita el desplazamiento del punto de ubicación.

[5.3.] El usuario desplaza el punto a la ubicación deseada.

[5.4.] El sistema cambia las coordenadas de ubicación a la posición actual.

[5.5.] El caso de uso continúa en el punto (6) del flujo normal de eventos.

Tabla 8. Ficha de caso de uso Descargar reporte

CU07	Actores: Administrador	Prioridad: Media
Caso de Uso: Gestionar árbol – Descargar reporte		
Descripción: El administrador descarga uno o varios registros de árboles en un archivo CSV.		
Referencias: RF11.		
Pre-Condiciones:		
<ul style="list-style-type: none">- El usuario Administrador haya iniciado sesión.- El usuario se encuentra en la sección Reporte general de árboles.		
Post-Condiciones: Se genera un archivo CSV con información de los árboles seleccionados.		
Flujo normal de los eventos:		
<ol style="list-style-type: none">1. El usuario aplica filtros en la tabla para traer los registros deseados.2. El sistema muestra los resultados filtrados en la tabla.3. El usuario selecciona las columnas de la tabla que serán visibles.4. El sistema oculta/muestran las columnas habilitadas.5. El usuario presiona Descargar CSV.6. El sistema genera un archivo CSV con los registros de la tabla.7. El sistema descarga automáticamente el archivo.		
Flujo Alternativo de eventos:		
A. Usuario no filtra resultados		
[1.1.] El sistema muestra por defectos todos los resultados paginados en la tabla.		
[1.2.] El caso de uso continúa en el punto (3) del flujo normal de eventos.		
B. Usuario no selecciona columnas		
[2.1.] El sistema muestra por defectos todas las columnas.		
[2.2.] El caso de uso continúa en el punto (5) del flujo normal de eventos.		

C. Diseño

Se usó la herramienta de modelado StarUML para diseñar los diagramas de secuencia (ver **Anexo 4, FASE 3: Diseño**):

- Se diseño los modelos de secuencia (ver desde **Figura 6** hasta **Figura 13**)
- Se diseño el modelo de clases (ver **Figura 14**)
- Se diseño el modelo de datos (ver **Figura 15**)

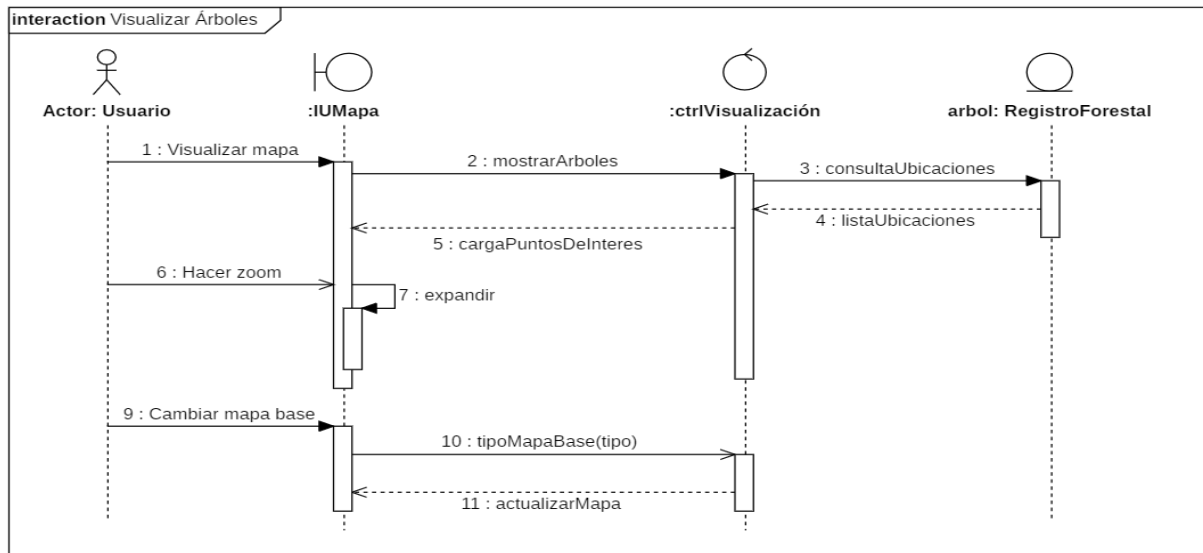


Figura 6. Diagrama de secuencia para el caso de uso Visualizar árboles.

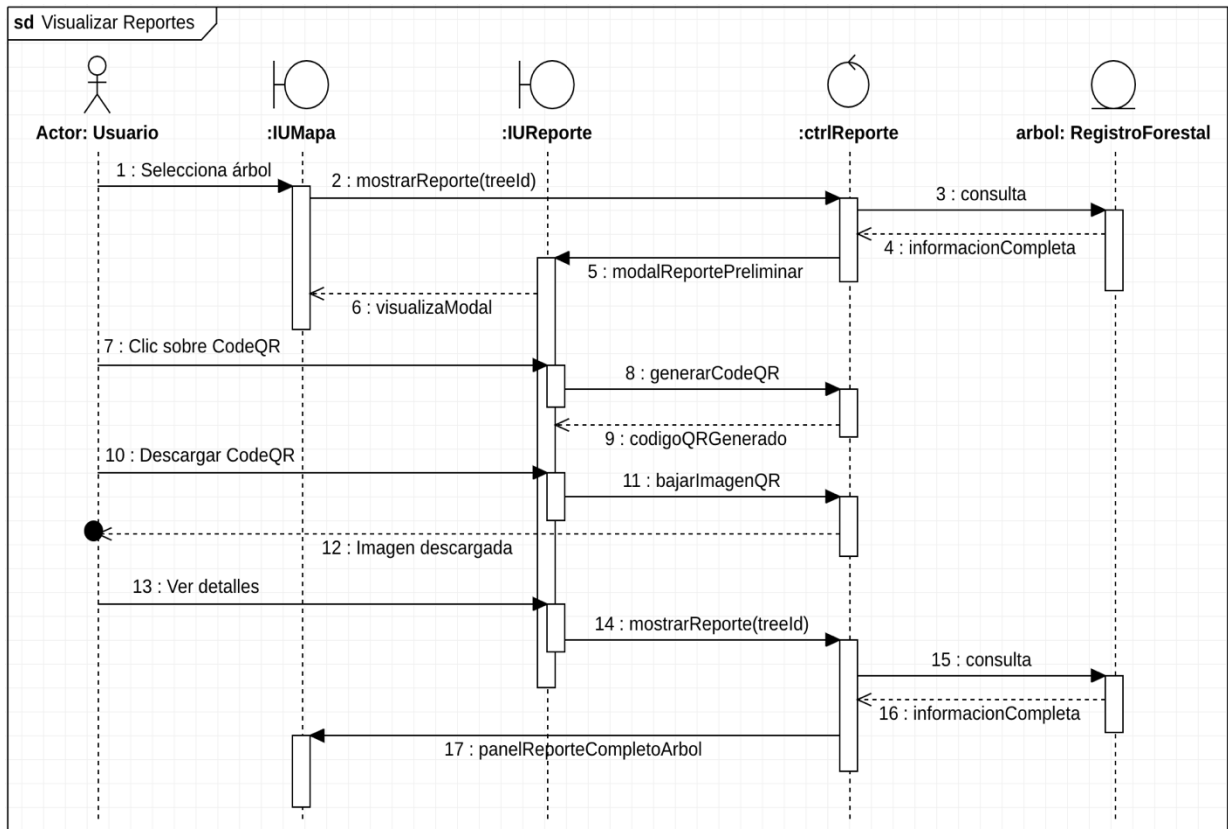


Figura 7. Diagrama de secuencia para el caso de uso Visualizar reportes.

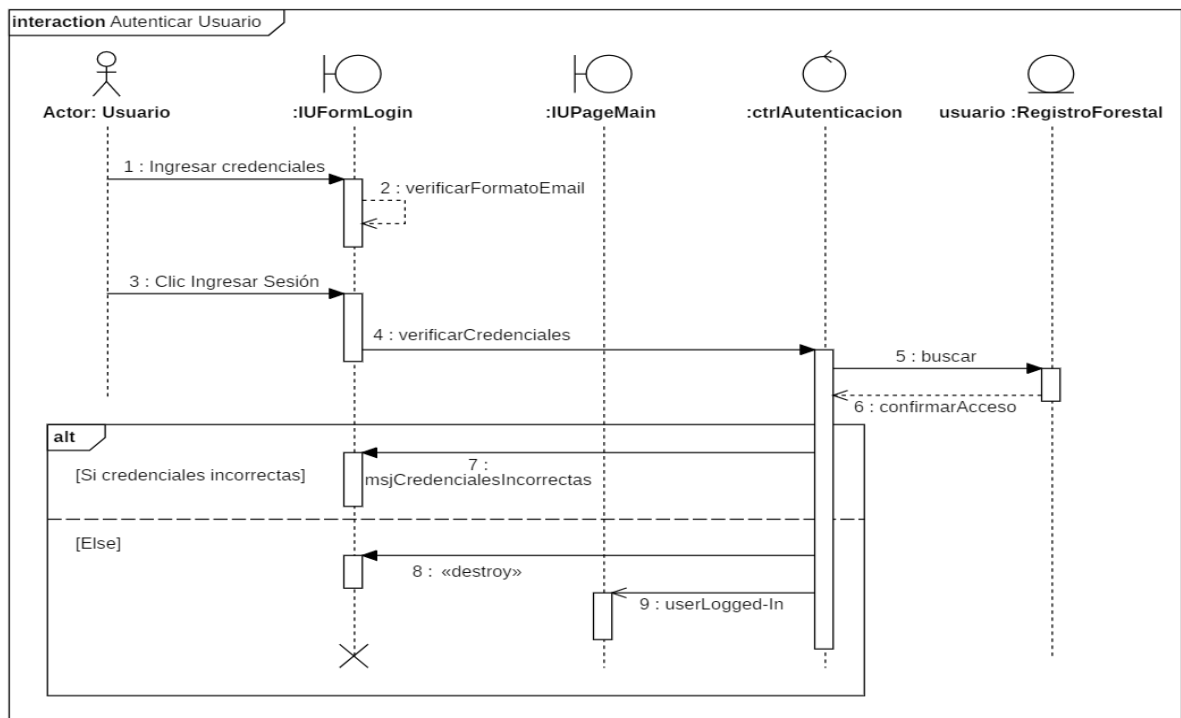


Figura 8. Diagrama de secuencia para el caso de uso Autenticar usuario.

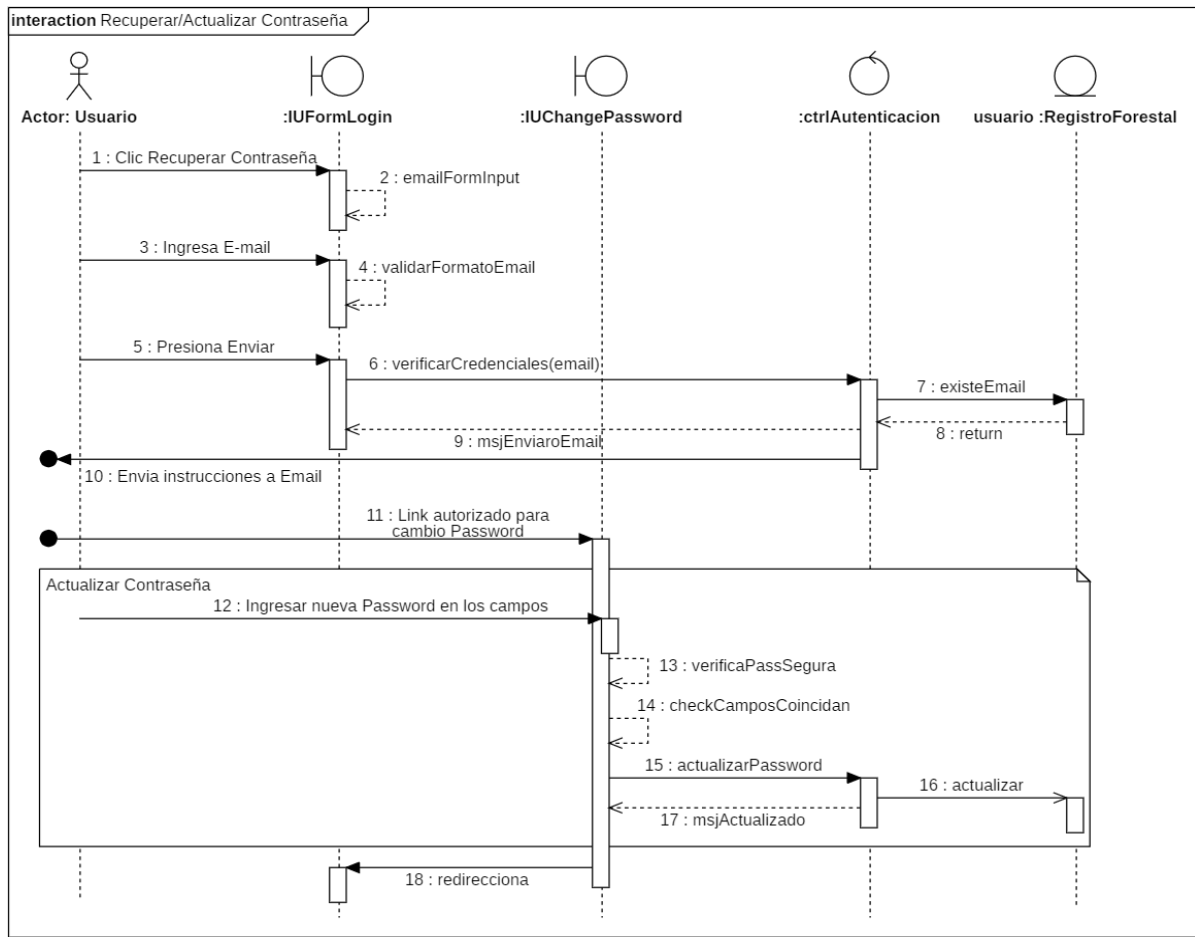


Figura 9. Diagrama de secuencia para el caso de uso Recuperar/Actualizar contraseña.

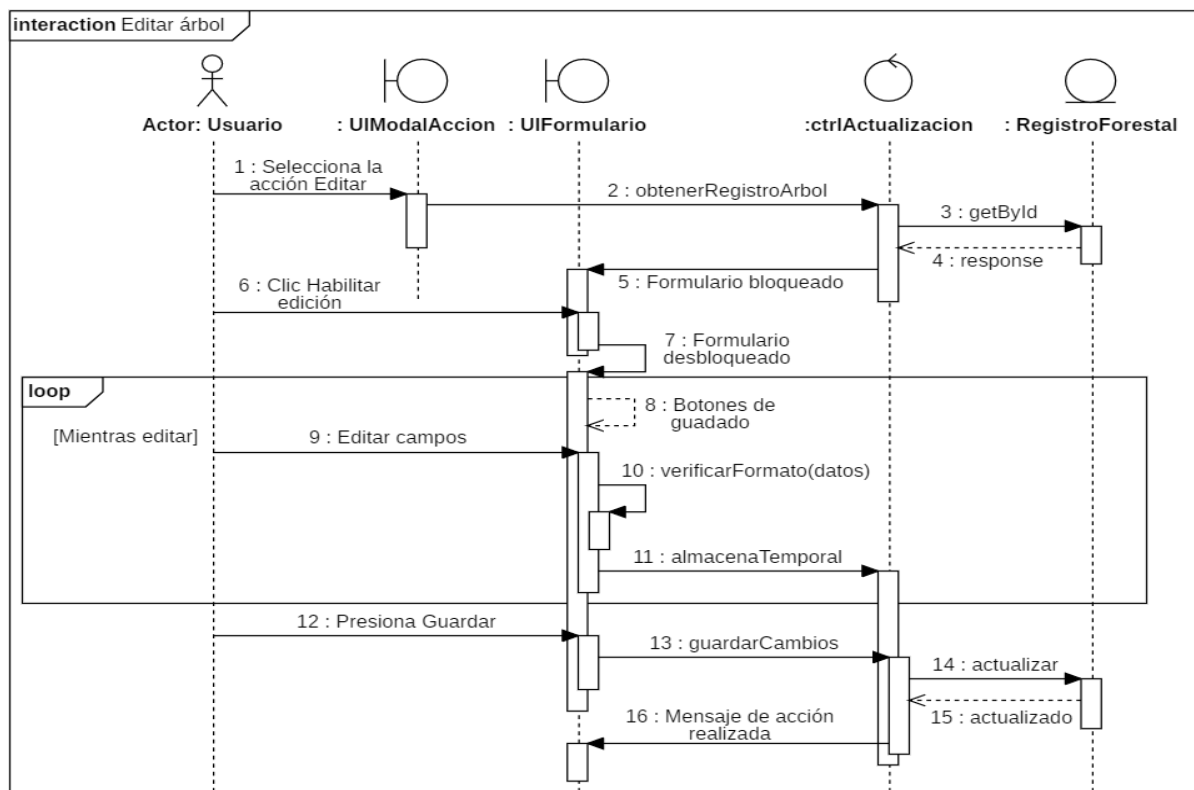


Figura 10. Diagrama de secuencia para el caso de uso Editar árbol.

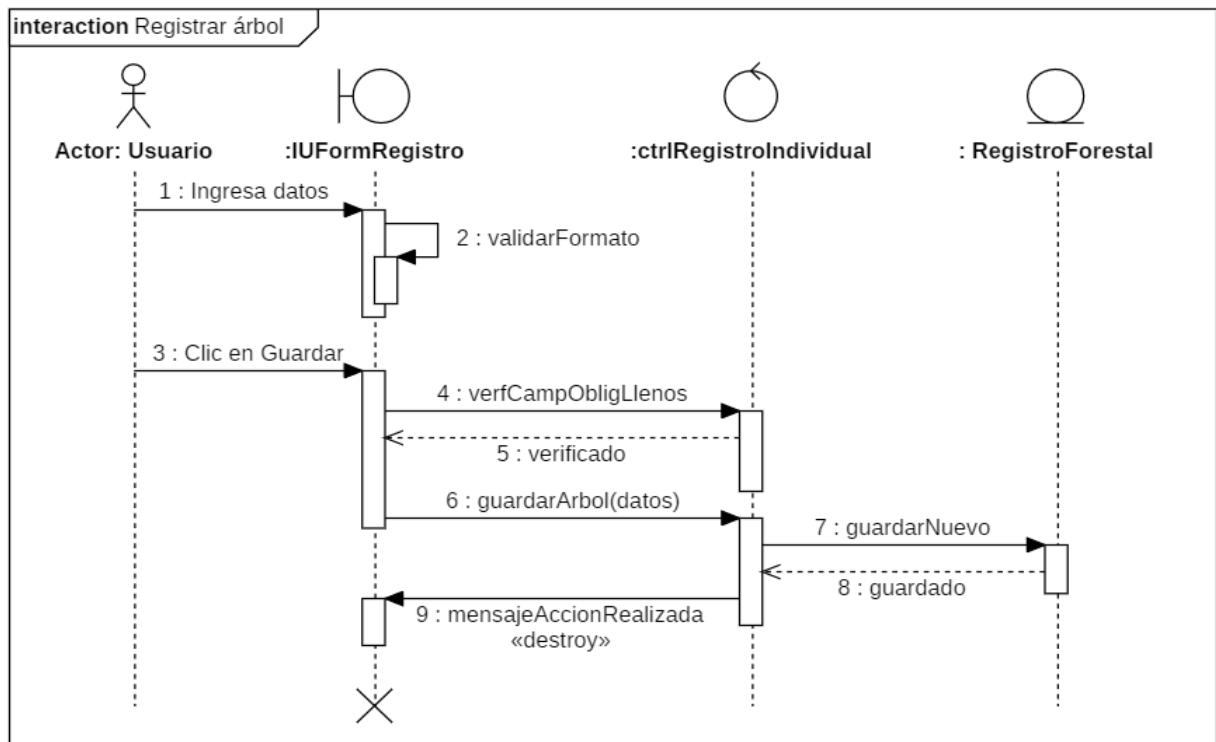


Figura 11. Diagrama de secuencia para el caso de uso Registrar árbol

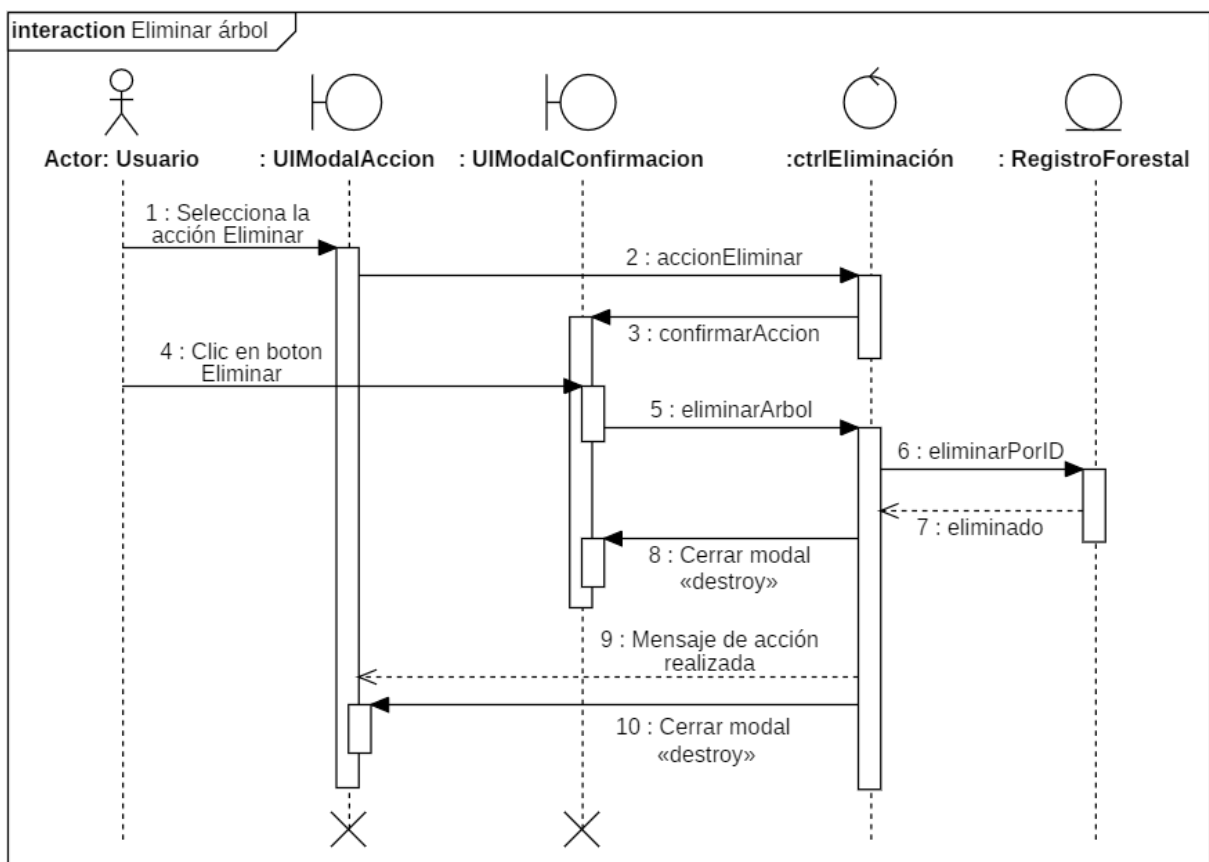


Figura 12. Diagrama de secuencia para el caso de uso Eliminar árbol.

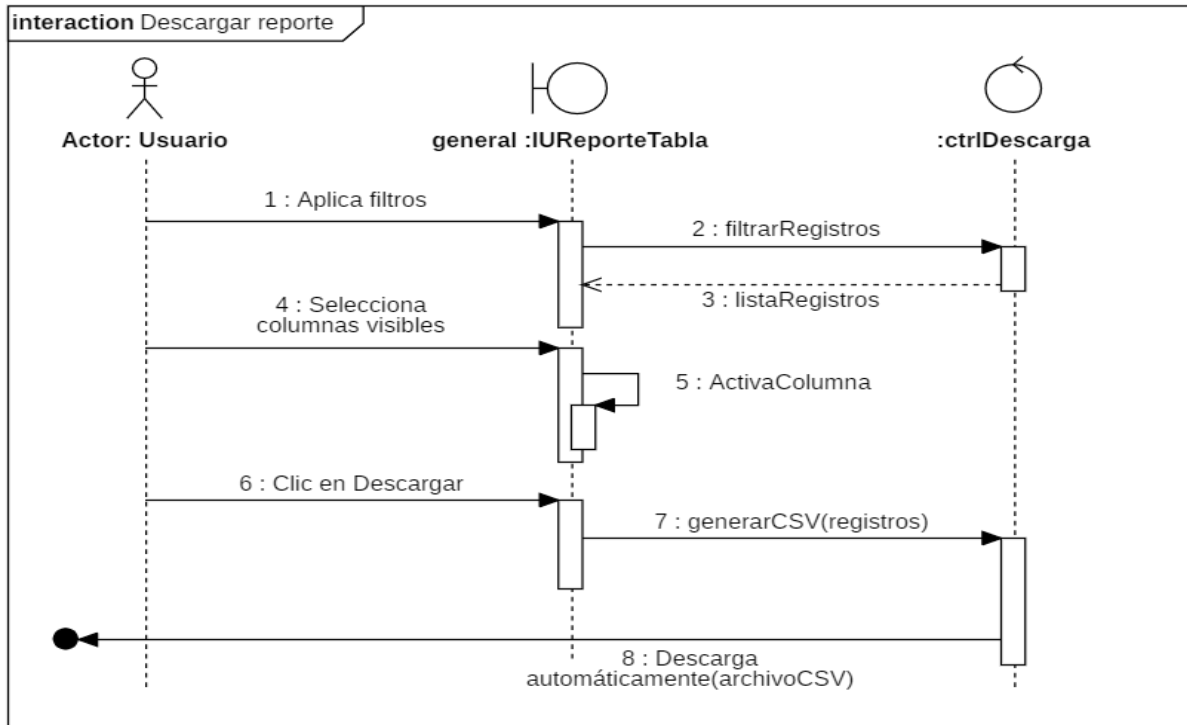


Figura 13. Diagrama de secuencia para el caso de uso Descargar reporte

En la **Figura 14**, el artefacto corresponde al diagrama de clases se obtiene en base al análisis de requerimientos, estableciendo las relaciones entre diferentes clases que intervienen en el modelo de negocio del proyecto de investigación. De acuerdo a un requerimiento no funcional en la que se establece que la persistencia debe hacerse en una base de datos MySQL, que es una base de datos con modelos relacionales, es necesario realizar un mapeo objeto-relacional (MOR), se obtiene el modelo de entidad relación para dicha base de datos en la **Figura 15**, con la finalidad de sincronizar el enfoque relacional y orientado a objeto.

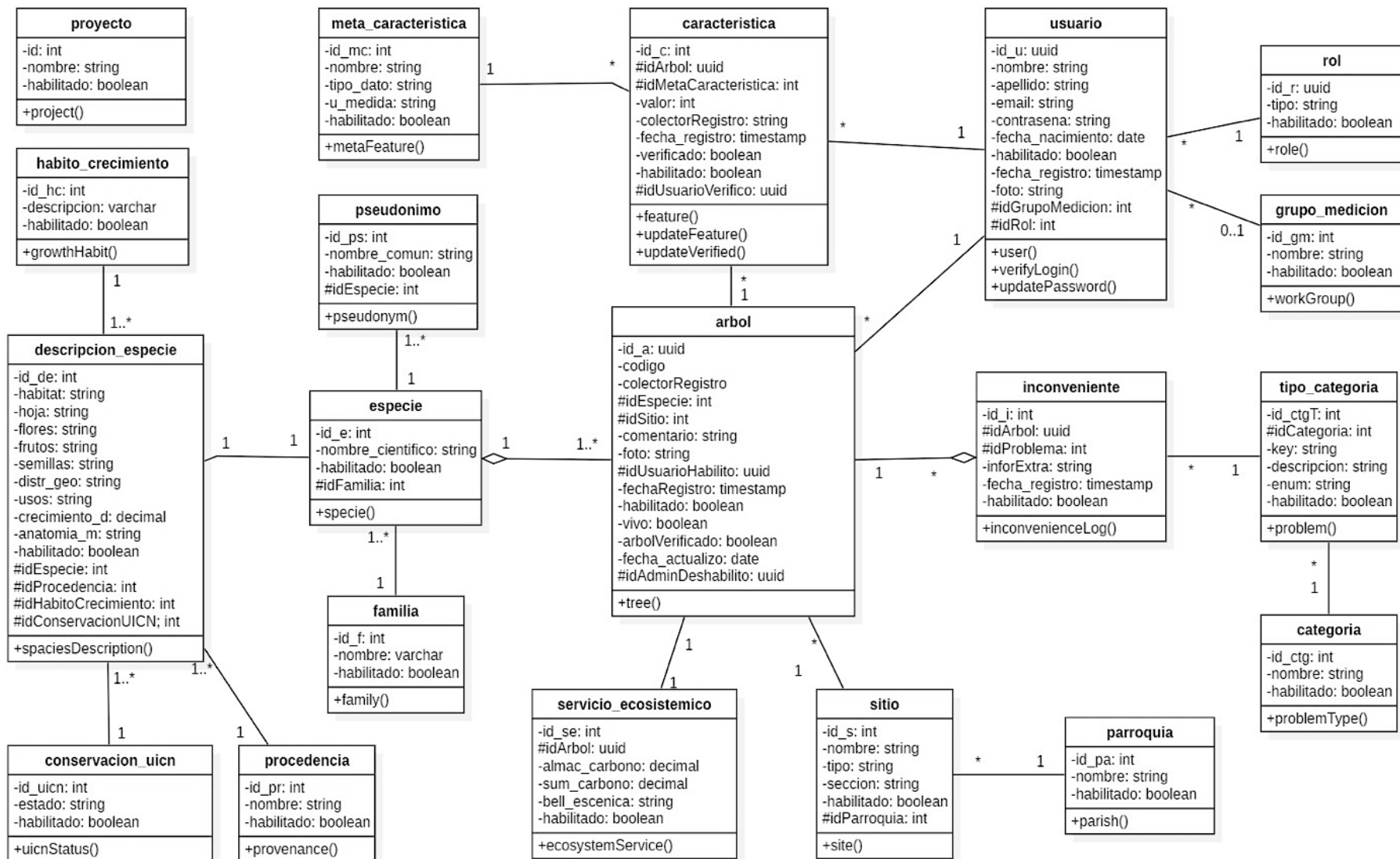


Figura 14. Diagrama de clases

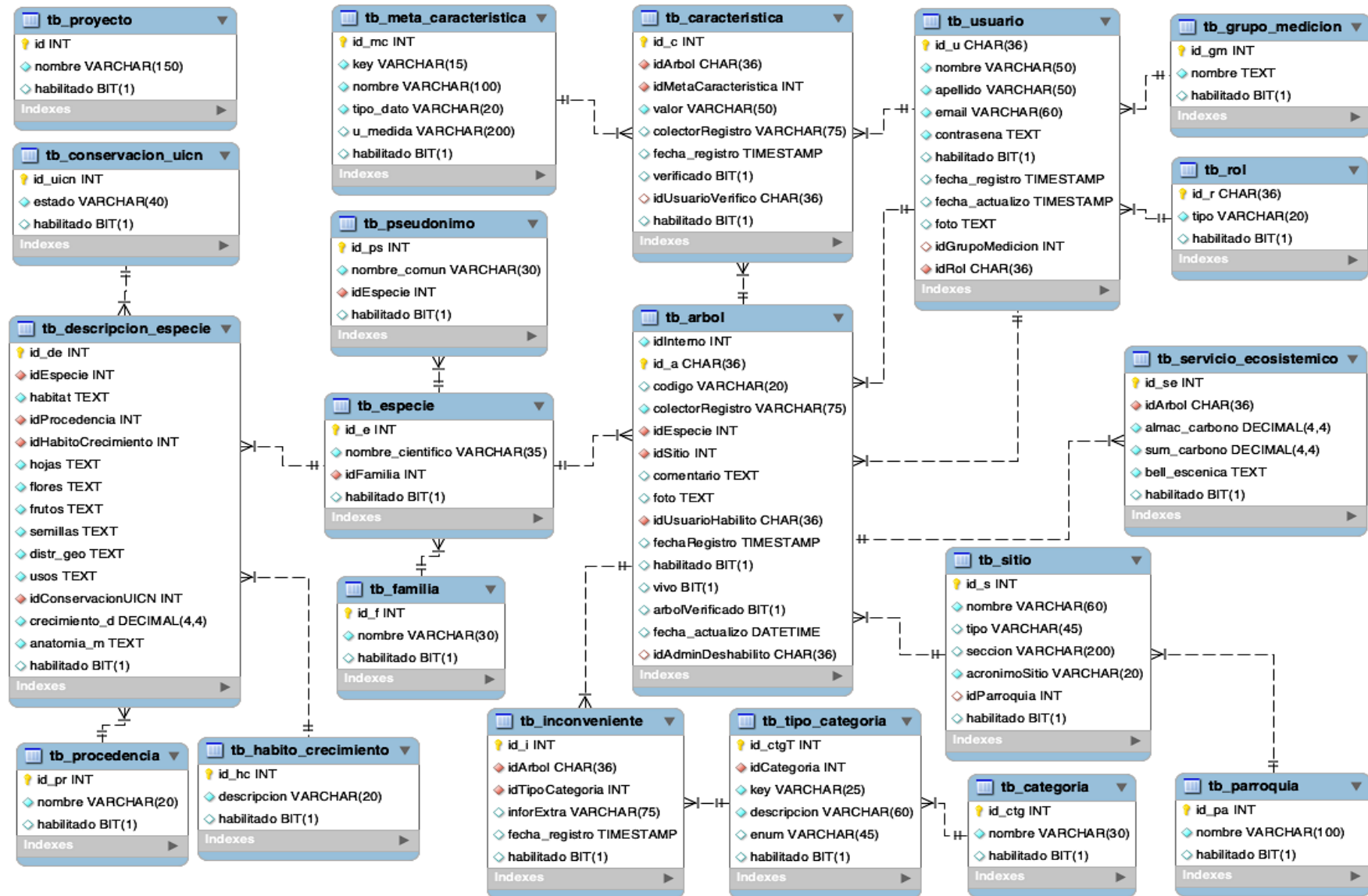


Figura 15. Diagrama de datos del aplicativo web obtenidos de MySQL Workbench

6.2. Objetivo 2: Implementar el diseño de la aplicación mediante el framework ReactJS y NodeJS

Se realizó dos tareas, en primer lugar, la codificación de la aplicación web y el segundo lugar, la documentación correspondiente al diseño de la arquitectura web, que se detallan a continuación:

6.2.1. Codificar la Aplicación Web

La codificación se realizó bajo el entorno de ejecución multiplataforma NodeJS, donde se instaló el framework para backend ExpressJS, y para el frontend ReactJS (ver **Anexo 4, FASE 4, Ítem 5.1**). Se utilizó las herramientas de Visual Studio Code como IDE para la codificación, MySQL Workbench como herramienta de administración de base de datos y Postman como plataforma para el diseño y pruebas del API Server en el backend.

La documentación del código fuente se respaldó en tres repositorios de GitHub (Privados):

- webGISTree-backend
- webGISTree-frontend
- arbol_QR_db

6.2.1.1. Codificación del backend y frontend

La estructura del directorio del backend se muestra en la **Figura 17** y la de frontend en la **Figura 18**, los paquetes utilizados se describen en los archivos package.json. (ver detalles en **Anexo 4, FASE 4, Ítems 5.1.1 - 5.1.2**).

- Se estructuró internamente los servicios del backend en archivos que contienen rutas, modelos y controladores (ver **Figura 16**).



Figura 16. Estructura de archivos de un servicio API-REST

The screenshot shows the VS Code Explorer on the left with a file tree for the backend project. The tree includes folders like 'build', 'node_modules', 'src', 'apiServices', 'config', 'environments', 'mailerConfig', 'mysql', 'middleware', 'routes', 'server', 'templates', and 'utils'. Files include 'app.js', '.babelrc', '.env', '.env.example', '.gitignore', 'package-lock.json', 'package.json', and 'README.md'. The main editor shows the 'package.json' file with the following content:

```

1  {
2    "name": "webgistree-backend",
3    "version": "1.0.0",
4    "description": "ArbolQR App WEB",
5    "main": "index.js",
6    "scripts": {
7      "babel-node": "babel-node --presets=@babel/preset-env",
8      "dev": "nodemon --exec npm run babel-node src/server/index.js",
9      "build": "babel src -d build",
10     "start": "npm run babel-node src/server/index.js"
11   },
12   "keywords": [],
13   "author": "Johnny González",
14   "license": "ISC",
15   "dependencies": {
16     "bcryptjs": "^2.4.3",
17     "consola": "^2.15.3",
18     "cors": "^2.8.5",
19     "dotenv": "^16.0.1",
20     "express": "^4.18.1",
21     "http-errors": "^2.0.0",
22     "jsonwebtoken": "^8.5.1",
23     "lodash": "^4.17.21",
24     "mysql2": "^2.3.3",
25     "nodemailer": "^6.8.0",
26     "sequelize": "^6.21.2",
27     "sequelize-cli": "^6.4.1"
28   },
29   "devDependencies": {
30     "@babel/cli": "^7.18.6",
31     "@babel/core": "^7.18.6",
32     "@babel/node": "^7.18.6",
33     "@babel/preset-env": "^7.18.6",
34     "morgan": "^1.10.0",
35     "nodemon": "^2.0.19"
36   }
37 }
38

```

Figura 17. Estructura de carpetas y paquetes del backend.

The screenshot shows the VS Code Explorer on the left with a file tree for the frontend project. The tree includes folders like 'build', 'node_modules', 'public', 'src', 'actions', 'assets', 'components', 'config', 'containers', 'PrivateRoute', 'redux', 'services', 'theme', and 'utils'. Files include 'App.jsx', 'index.css', 'index.jsx', '.env', '.eslintignore', '.gitignore', 'package-lock.json', 'package.json', 'postcss.config.js', and 'README.md'. The main editor shows the 'package.json' file with the following content:

```

4   "private": true,
5   "dependencies": {
6     "@emotion/react": "^11.10.5",
7     "@emotion/styled": "^11.10.5",
8     "@mui/icons-material": "^5.11.0",
9     "@mui/material": "^5.11.1",
10    "@mui/styled-engine-sc": "^5.8.0",
11    "@mui/x-data-grid": "^5.17.17",
12    "@mui/x-date-pickers": "^5.0.6",
13    "@reduxjs/toolkit": "^1.8.3",
14    "@testing-library/jest-dom": "^5.16.4",
15    "@testing-library/react": "^13.3.0",
16    "@testing-library/user-event": "^14.3.0",
17    "axios": "^0.27.2",
18    "date-fns": "^2.29.3",
19    "formik": "^2.2.9",
20    "lodash": "^4.17.21",
21    "mapbox-gl": "^2.9.2",
22    "moment": "^2.29.4",
23    "mui-datatables": "^4.2.2",
24    "qr-code-styling": "^1.6.0-rc.1",
25    "react": "^18.2.0",
26    "react-detect-offline": "^2.4.5",
27    "react-dom": "^18.2.0",
28    "react-redux": "^8.0.2",
29    "react-router-dom": "^6.3.0",
30    "react-scripts": "^5.0.1",
31    "styled-components": "^5.3.5",
32    "web-vitals": "^2.1.4",
33    "xlsx": "^0.18.5",
34    "yup": "^0.32.11"
35   },
36   "scripts": {
37     "start": "react-scripts start",
38     "build": "react-scripts build",
39     "test": "react-scripts test",
40     "eject": "react-scripts eject"
41   },

```

Figura 18. Estructura de carpetas y paquetes del frontend

***routes**: La estructura de rutas definidas para un servicio API REST por donde se accederán a los diferentes recursos muestra en la **Figura 19**:

```
src > apiServices > user > JS routes.js > ...
1  import { Router } from 'express'
2  import { methods as controller } from './controller'
3  import { authorization, isAdmin, authorization_alt } from '../middleware/authorization'
4  import { validateInfo } from '../middleware/validateInfo'
5  const routes = Router()
6
7  routes.get('/me', authorization, controller.me)
8  routes.get('/', authorization, controller.getUserAll)
9  routes.get('/:id', authorization, controller.getUserById)
10 routes.post('/register-user', validateInfo, controller.createUser)
11 routes.put('/update-account', authorization, controller.updateUserAccount)
12 routes.put('/update-password', authorization_alt, controller.updatePassword)
13 routes.put('/:userId', authorization, controller.updateUserById)
14 routes.put('/disable/:id', authorization, isAdmin, controller.disableUserById)
15 routes.put('/enable/:id', authorization, isAdmin, controller.enableUserById)
16 export default routes
17
```

Figura 19. Especificación de rutas para el servicio usuario.

***models**: Para los modelos se utilizó el ORM Sequelize que permite el mapeo relacional de objetos, crear la base de datos MySQL con sus relaciones, su contenido es similar a la **Figura 20**:

```
JS model.js
src > apiServices > specie > JS model.js > ...
1  import { DataTypes, Model } from 'sequelize'
2  import { sequelize } from '../config/mysql'
3
4  class SpecieModel extends Model {
5    static associate(models) {
6      SpecieModel.belongsTo(models.FamilyModel, { targetKey: 'idFamilia' })
7      SpecieModel.hasMany(models.PseudonymModel, { foreignKey: 'id_e' })
8      SpecieModel.hasMany(models.TreeModel, { foreignKey: 'id_e' })
9    }
10 }
11 SpecieModel.init(
12   {
13     id_e: {
14       type: DataTypes.INTEGER, autoIncrement: true, primaryKey: true,
15     },
16     nombre_cientifico: {
17       type: DataTypes.STRING(35), allowNull: false, unique: true,
18     },
19     idFamilia: {
20       type: DataTypes.INTEGER, allowNull: false, references: {
21         model: 'tb_familia',
22         key: 'id_f',
23       },
24     },
25     habilitado: { type: 'BIT', defaultValue: true },
26   },
27   { sequelize, timestamps: false, modelName: 'tb_especie' }
28 )
29
30 export default SpecieModel
```

Figura 20. Especificación del modelo especie.

* **controller**. Se utilizó los controladores para el manejo de los diferentes aspectos de cada petición, uno de ellos se muestra en la **Figura 21**:

```
175 const updateSpecie = (req, res) => {
176   const { especieId } = req.params
177   const { rol } = req.payload
178   const { nombre_cientifico, idFamilia, habilitado } = req.body
179
180   SpecieModel.update(
181     {
182       nombre_cientifico,
183       idFamilia,
184       habilitado,
185     },
186     {
187       where: { id_e: especieId },
188     }
189   )
190   .then(specie => {
191     if (!Boolean(specie[0])) return res.status(200).json({ update: false, res: especie })
192     res.status(200).json({ update: true, res: especie })
193   })
194   .catch(err => {
195     console.error(err.message)
196     res.status(500).json({ error: 'Server Error', message: err.message })
197   })
198 }
```

Figura 21. Controlador actualizar especie por si Id.

- Se utilizó middlewares para el manejo de la autorización, validar si es administrador:

```
src > middleware > JS authorization.js > isAdmin
1 import 'dotenv/config'
2 import jwt from 'jsonwebtoken'
3 import UserModel from '../apiServices/user/model'
4 import RoleModel from '../apiServices/role/model'
5
6 // Authorization: Bearer <token>
7 export const authorization = (req, res, next) => {
8   try {
9     const accessToken = req.headers['authorization']
10
11     if (!accessToken)
12       return res.status(401).json({
13         auth: false,
14         message: 'Acceso no autorizado. Proporcione JWT.',
15       })
16
17     req.token = accessToken.split(' ')[1]
18     jwt.verify(req.token, process.env.JWT_SECRET, (err, payload) => {
19       if (err)
20         return res.status(403).json({
21           auth: false,
22           message: 'Acceso no autorizado. Token incorrecto o vencido.',
23         })
24       req.payload = payload.user // { id: user_id_u, rol: ['administrador'] }
25       next()
26     })
27   } catch (error) {
28     return res.status(500).json({ Error: error })
29   }
30 }

src > middleware > JS authorization.js > isAdmin
71
72 export const isAdmin = async (req, res, next) => {
73   try {
74     const user = await UserModel.findById(req.payload.id)
75     if (!user)
76       return res.status(403).json({
77         message: 'El idUsuario no existe.',
78       })
79
80     const role = await RoleModel.findById(user.idRol)
81
82     if (role.tipo !== 'administrador')
83       return res.status(403).json({
84         isAdmin: false,
85         message: 'Acceso no autorizado. Requiere rol Admin',
86       })
87     next()
88   } catch (error) {
89     return res.status(500).json({ Error: error })
90   }
91 }
92 }
```

Figura 22. Middlewares para autorización y verificar usuario administrador.

- En el frontend, se utilizó **Yup** conjuntamente con **Formik** para la validación de entradas en los formularios.

```
src > utils > JS schemaValidation.js > ...
1  import * as Yup from 'yup'
2
3  const EMAIL = {
4    email: Yup.string().email('Email Inválido.').required('Obligatorio'),
5  }
6
7  const USERNAME = {
8    username: Yup.string()
9      .strict(true)
10     .trim('No inicie o termine con espacios en blanco.')
11     .min(3, 'Nombre demasiado corto, digite más de 3 caracteres.')
12     .max(50, 'Nombre demasiado largo, máx 50 caracteres.')
13     .required('Obligatorio'),
14  }
15
16  const LASTNAME = {
17    lastname: Yup.string()
18      .strict(true)
19      .trim('No inicie o termine con espacios en blanco.')
20      .min(3, 'Apellido demasiado corto, digite más de 3 caracteres.')
21      .max(50, 'Apellido demasiado largo, max. 50 caracteres.')
22      .required('Obligatorio'),
23  }
24
25  const PASSWORD_LOGIN = {
26    password_login: Yup.string()
27      .strict(true)
28      .trim('Evite espacios en blanco.')
29      .required('Obligatorio'),
30  }
```

Figura 23. Schemas para la validación de entradas.

6.2.1.2. Interfaces de usuario.

En la **Figura 24**, se muestra la interfaz de inicio de la aplicación, que hace énfasis en la visualización de los puntos de ubicaciones de árboles en el mapa. El menú lateral izquierdo, las opciones para filtrar árboles, cambiar entre los diferentes tipos de mapas, administrar árboles, registrar árboles. Estos dos últimos, el acceso solo para usuarios registrado cuya cuenta se encuentra habilitada. Además, se muestra información del árbol seleccionado dentro de un cuadro de información (*popup*) al interior del mapa y un reporte completo, en el panel lateral derecho.

El **Anexo 4, FASE 4**, Ítem **5.1.3** se encuentran todas las interfaces con sus respectivas observaciones.

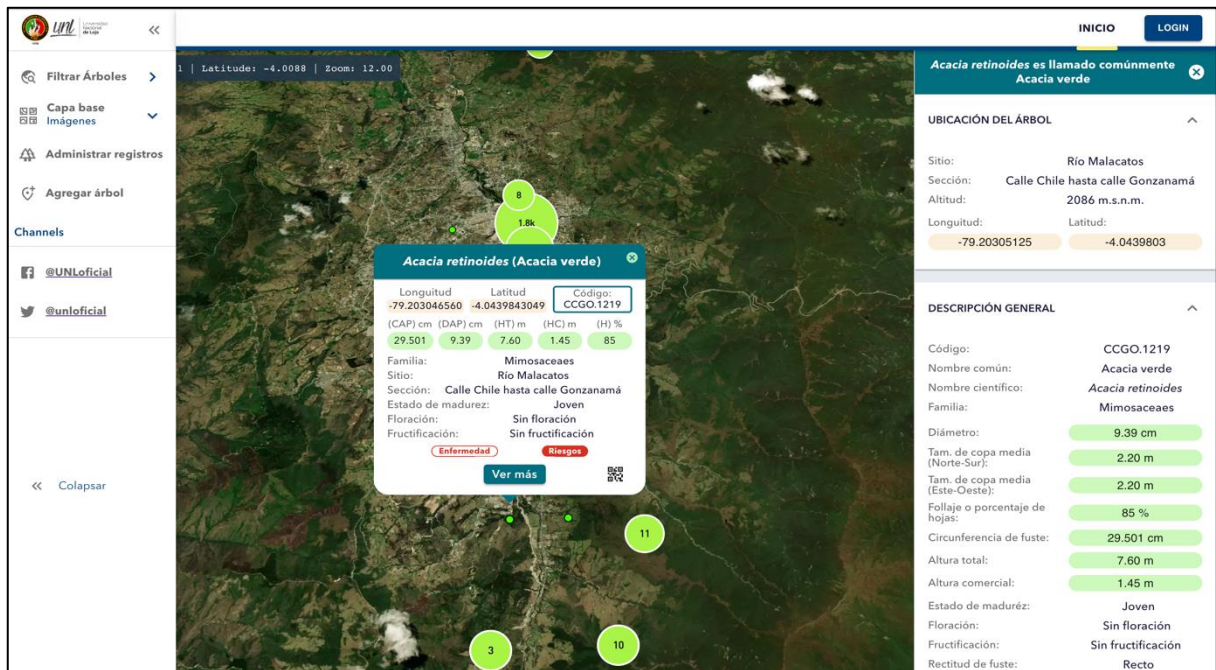


Figura 24. Interfaz inicial de la aplicación web, árbol seleccionado.

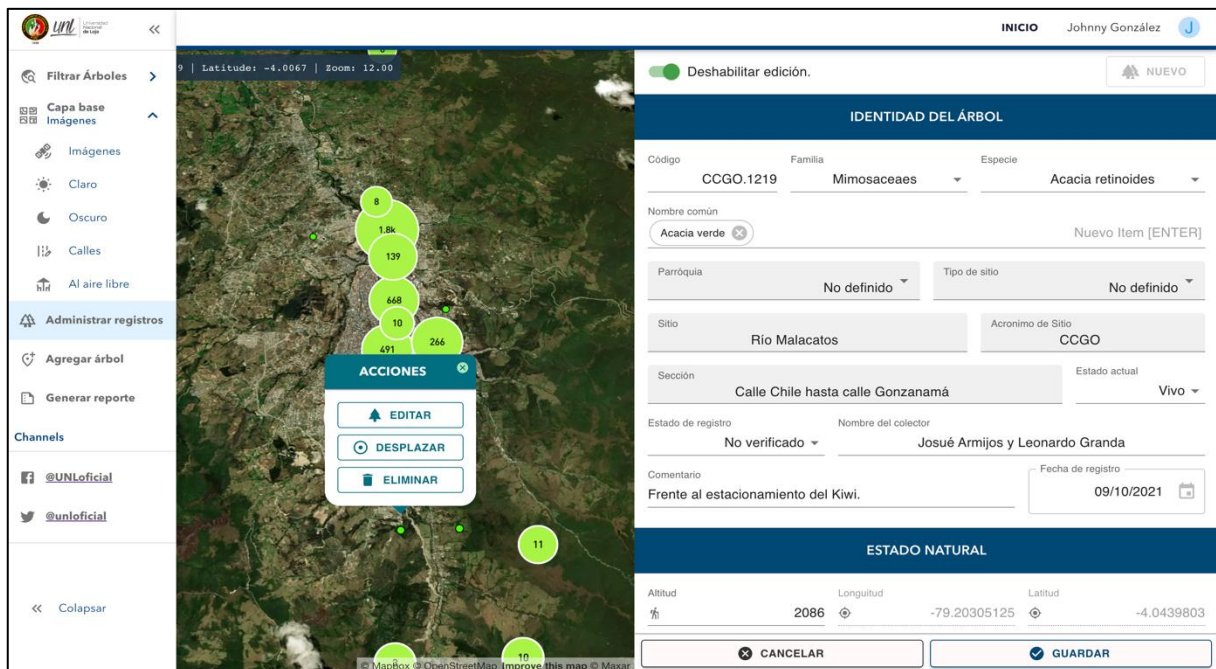


Figura 25. Interfaz para administrar registro de árboles.

6.2.2. Documentar la arquitectura de la aplicación web

Se construyó usando el modelo de arquitectura Cliente – Servidor con el patrón de diseño Modelo Vista Controlador (MVC), la división del backend y del frontend por separado, para los modelos y controladores en el lado del Servidor y la vista en lado del Cliente.

Se elaboró el diagrama de componentes en base al diseño de la arquitectura empleada (ver **Figura 26**). Finalmente, se elaboró el diagrama de despliegue de la aplicación (ver **Figura 27**).

6.2.2.1. Diagrama de componentes

Se diseñó el diagrama de componentes, la arquitectura de la aplicación WebGISTree (ver **Figura 26**), el cual indica la organización de los componentes aplicando la arquitectura n-tier (n-capas).

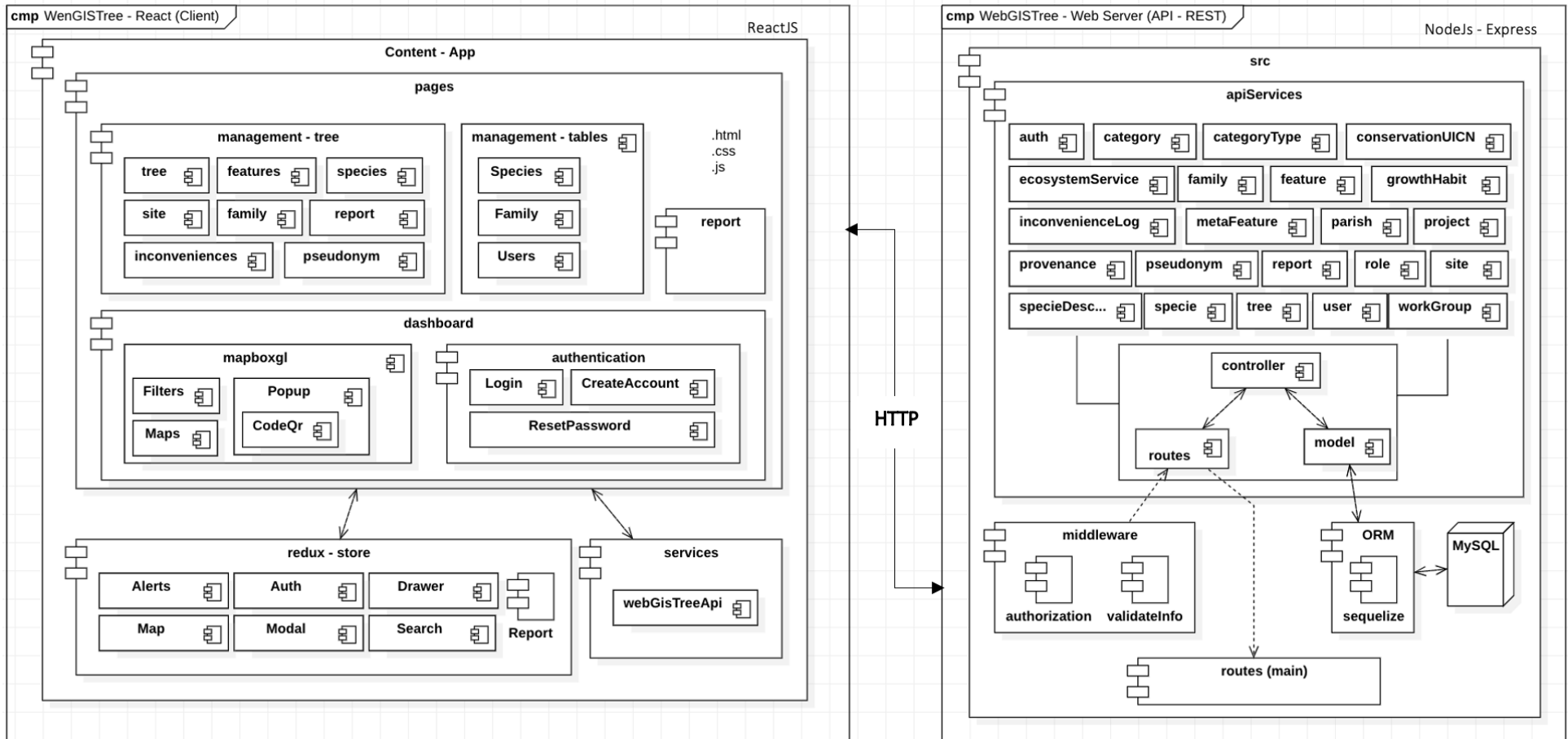


Figura 26. Diagrama de componentes de la aplicación web.

6.2.2.2. Diagrama de despliegue

El diagrama de despliegue ilustra el diseño de la arquitectura física de la aplicación (ver **Figura 27**), este consta de un Cliente web React que se comunica con el Servidor Node, donde se encuentra el servicio web (API-REST) WebGISTree-backend, que interactúa a través de un Mapeador Relacional de Objetos (ORM) con el servidor de base de datos relacional MySQL.

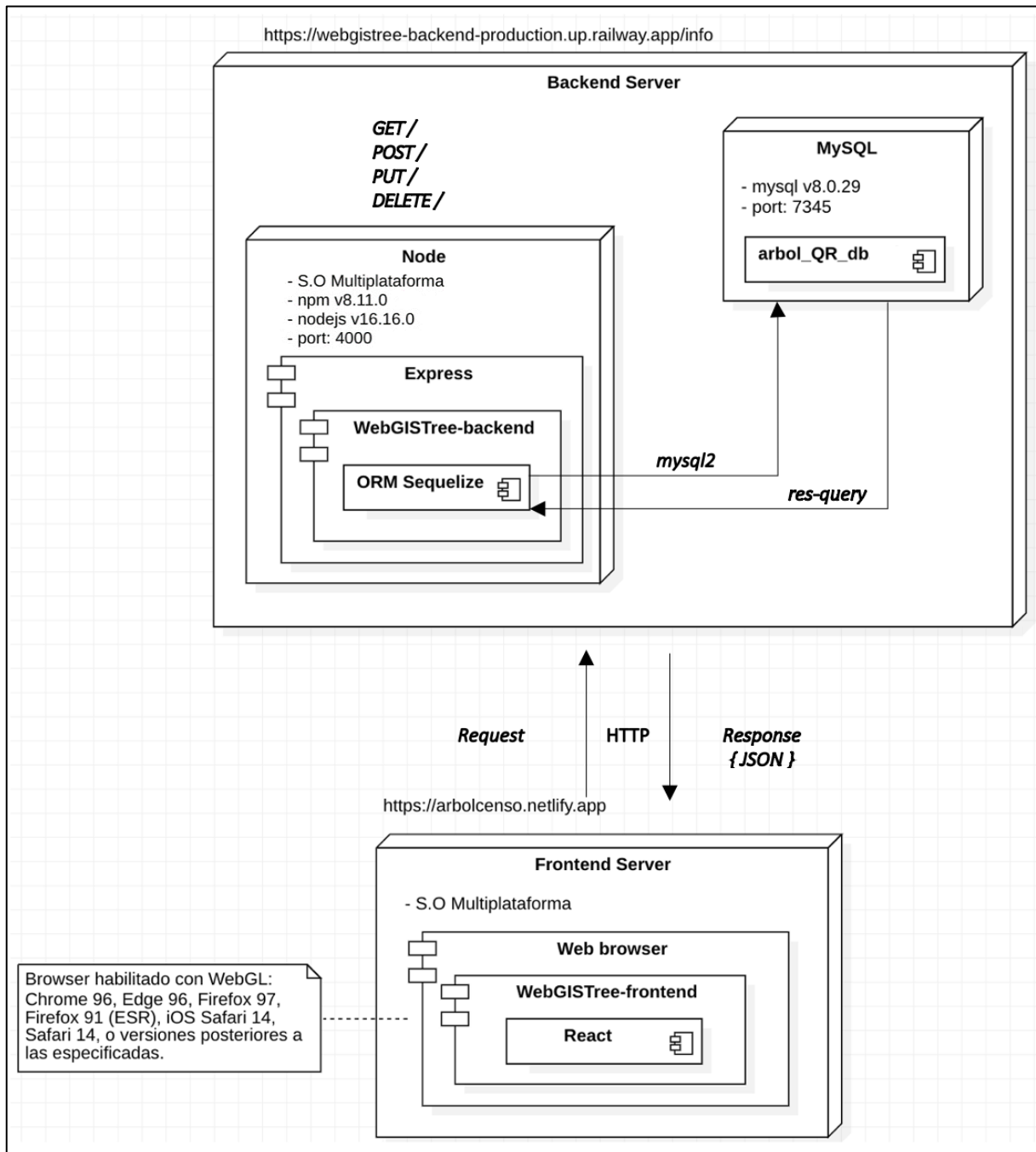


Figura 27. Diagrama de despliegue en servidores Railway y Netlify.

6.3. Objetivo 3: Ejecutar pruebas de funcionalidad para la validación de la Aplicación Web

En esta sección se desarrolló la etapa de pruebas, se comenzó por las pruebas unitarias al servicio web, pruebas de caja negra y pruebas de funcionalidad para garantizar que cumple con los requisitos establecidos en la inicialmente en el desarrollo de solución informática.

6.3.1. Pruebas unitarias al servicio web API-REST

Se realizó las pruebas al servicio web API-REST con la herramienta Postman⁹ y se probó los servicios creados mediante peticiones HTTPS validando sus respectivas respuestas en formato JSON.

Para ello se realizó el despliegue del servicio web WebGISTree en el servidor gratuito Railway¹⁰, que simuló un ambiente de control real, donde se evaluó el correcto funcionamiento de las peticiones y se obtuvo los tiempos de respuestas (ver **Anexo 5**), que se resumen en la siguiente tabla:

Tabla 9. Principales endpoints del servicio web de la aplicación WebGISTree

URL	Token (JWT)	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/feature/locations	No	GET	Obtener los puntos de localización de árboles	8.56s
https://webgistree-backend-production.up.railway.app/auth/send-email-reset-password	No	POST	Enviar email con el link para recuperar contraseña	198ms
https://webgistree-backend-production.up.railway.app/tree/:treeld	No	GET	Obtener un registro de árbol por su ID	331ms
https://webgistree-backend-production.up.railway.app/auth/login	No	POST	Iniciar sesión usuarios	1056ms
https://webgistree-backend-production.up.railway.app/tree/register-tree	Sí	POST	Registrar datos en tabla árbol.	30ms
https://webgistree-backend-production.up.railway.app/feature/update-feature/:treeld	Sí	POST	Registrar características para un árbol	707ms
https://webgistree-backend-production.up.railway.app/inconvenience/update-inconvenience/:treeld	Sí	POST	Registrar inconveniente para un árbol	915ms
https://webgistree-backend-production.up.railway.app/specie/register-specie	Sí	POST	Registrar especies	329ms

⁹ Postman: <https://www.postman.com>

¹⁰ Railway: <https://www.railway.app>

6.3.2. Pruebas de Caja Negra

Se realizó las pruebas de caja negra en base los casos de uso, tomando las descripciones de las fichas de casos uso para realizar las pruebas y validar si cumple o no con los requisitos del software. En la **Tabla 10**, se muestra un ejemplo de caso de prueba de caja negra para editar un registro de árbol.

Tabla 10. Caso de prueba de caja negra - Editar árbol.

PCN10	Casos de uso: Editar árbol	Aprobado		Observación
		Sí	No	
	Descripción de la prueba			
	En la página de "Administrar registros", al seleccionar un árbol del mapa, mostrar una ventana emergente con las acciones disponibles según el tipo del rol del usuario.	X		
	Mostrar el formulario de Edición, al seleccionar el botón "Editar".	X		
	Al seleccionar la acción "Desplazar", si el formulario de edición no ha sido mostrado con anterioridad, direccionar al formulario de edición de árbol, caso contrario, mostrar un marcador para el desplazamiento de ubicación en el mapa.	X		
	El switch "Habilitar edición" o "Deshabilitar edición" permite o no, editar todos campos del formulario. A excepción de código de árbol, fecha de registro, latitud y longitud.	X		
	Validar que los campos sean obligatorios en la sección de "Identidad del árbol" del formulario, a excepción del campo familia.	X		
	Al seleccionar una familia, mostrar en el campo "Especie", la lista filtrada de los nombres científicos correspondiente a esa familia seleccionada.	X		
	Al seleccionar una especie, mostrar en el campo "Nombre Común", la lista filtrada de los seudónimos correspondientes a la especie seleccionada.	X		
	En el campo "Nombre común" seleccionar de la lista o agregar nuevos nombres al hacer 'Enter'.	X		
	Si el campo "Acrónimo de ubicación" no se encuentra registrado o no corresponden a una ubicación registrada para los campos: ("Parroquia", "Sitio", "Tipo de sitio" y "Sección") mostrar un mensaje (Crear nuevo acrónimo) al momento de Guardar.	X		
	Validar el formato de los campos en la sección de "Estado Natural" del formulario. Los campos no son obligatorios, a excepción de "Longitud" y "Latitud" que se encuentran deshabilitados para la edición textual.	X		
	Si la acción "Desplazar" fue habilitada, desplazar el marcador del mapa hacia una nueva ubicación.	X		
	Capturar ubicación (longitud, latitud) del marcador en el formulario, al presionar el botón "Capturar ubicación".	X		
	En las pestañas 1 y 7 del apartado "Condición General", los campos son de tipo selección simple.	X		
	En las pestañas 2, 3, 4, 5, 6 y 8 (correspondientes a las Opciones múltiples) fijar los elementos antes de cambiar de pestaña o antes de presionar el botón guardar.	X		
	Al presionar "Guardar", presentar un mensaje de guardado y actualizar el formulario, caso contrario presentar un mensaje del estado de validación.	X		

Las pruebas fueron ejecutadas bajo la supervisión del director del proyecto forestal Ing. Darwin Pucha Cofrep, quien en previas socializaciones interactuado con la aplicación web,

descubrió comportamiento no esperado y fallas de validación que fueron resueltos para la aprobación de los casos de prueba (ver **Anexo 7**).

6.3.3. Pruebas funcionales

Las pruebas funcionales fueron realizadas a cada uno de los módulos del software para verificar que se cumpla con las necesidades con respecto a los requisitos funcionales de la aplicación web. La validación de las distintas funcionalidades se la realizó en un ambiente controlado con la aplicación web subida en un servidor gratuito, se dio acceso al sistema al director encargado del proyecto de investigación forestal, para que use la aplicación agregando nuevos usuarios, validando usuarios, registrando árboles, editando árboles, entre otras actividades.

En la **Tabla 11** se muestra los requerimientos funcionales con su respectivo estado de cumplimiento que fue aprobado por el director del proyecto forestal (ver **Anexo 8**).

Tabla 11. Aprobación de funcionalidad final de requerimientos funcionales

Id	Requisito	Estado
RF01	Visualizar árboles	Realizado
RF02	Mostrar tipos de mapas base	Realizado
RF03	Visualizar reportes	Realizado
RF04	Filtra consultas	Realizado
RF05	Autenticar usuario	Realizado
RF06	Actualizar contraseña	Realizado
RF07	Recuperar contraseña	Realizado
RF08	Editar árboles	Realizado
RF09	Registrar árbol	Realizado
RF10	Eliminar árbol	Realizado
RF11	Descargar reportes	Realizado
RF12	Registrar cuenta	Realizado
RF13	Validar cuenta	Realizado

6.3.4. Pruebas no funcionales

Para la verificación de los requerimientos no funcionales se seleccionaron dos de mayor importancia, las mismas que se muestran las pruebas realizadas a continuación:

Requerimiento no funcional - Seguridad (RNF01), en las peticiones realizadas al servidor se utilizó el protocolo HTTPS y se usó token (JWT) para la autorización y acceso a recursos privados para usuarios con diferentes roles. Para ello, se crearon dos roles, **administrador** cuyas funcionalidades son completas, e **investigador** cuyas funciones están limitadas (ver **Figura 28**). Cada uno se les asigna diferentes accesos y funcionalidades al momento de iniciar sesión en la aplicación (ver **Anexo 5, FASE 4, Ítem 5.1.3**). El usuario común no requiere de un rol en la aplicación puesto que su acceso se limita a solo lectura de información pública.

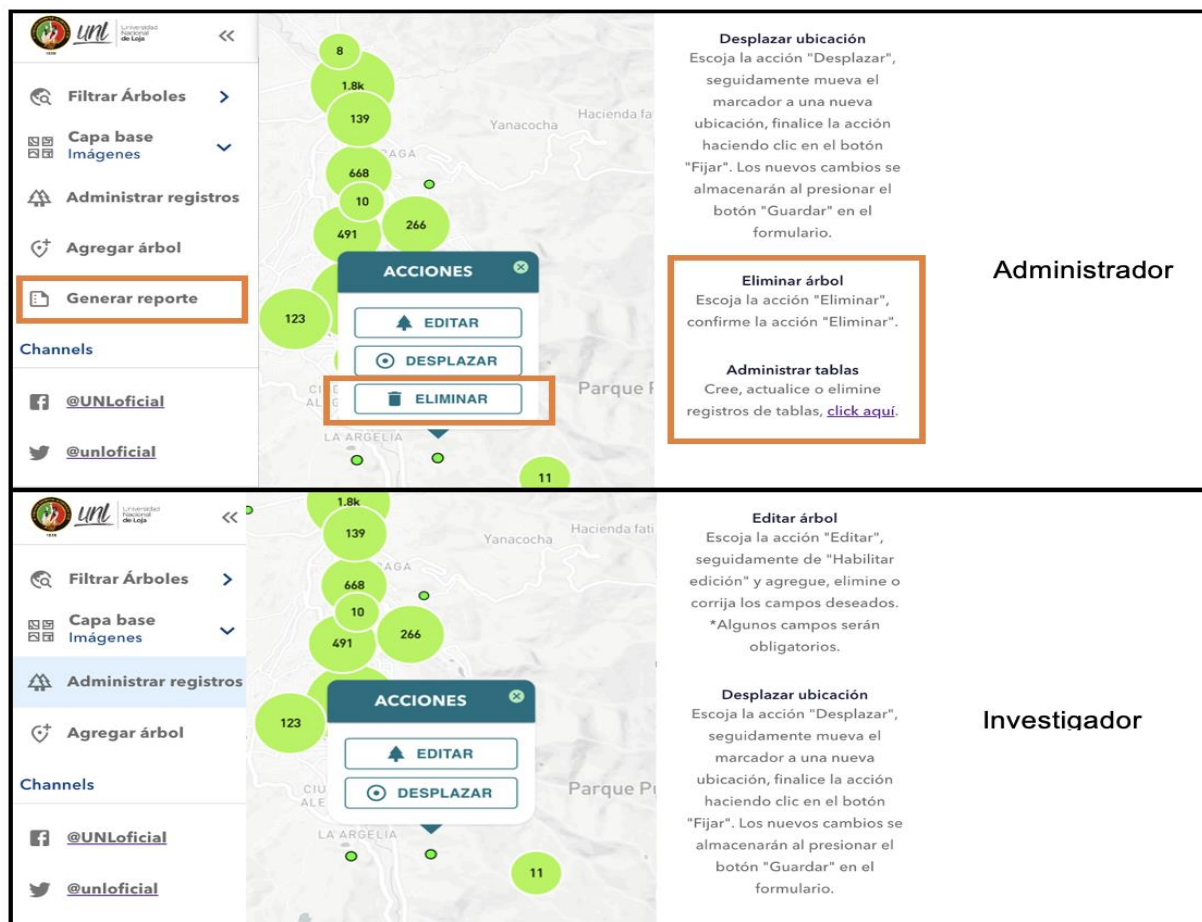


Figura 28. Interfaz con funcionalidades de acuerdo al rol.

Además, se utilizó la función de hashing de contraseña bcrypt (ver **Figura 29**), para asegurar la encriptación de las contraseñas que son almacenadas en la base de datos. De esta manera se cumple este apartado de seguridad.

```

13 export const encryptPassword = async password => {
14   const salt = await bcrypt.genSalt(saltRound)
15   return bcrypt.hash(password, salt)
16 }

```

Función hashing.

Result Grid Filter Rows: Search Edit: Export/Import:

nombre	apellido	email	contrasena	habilitado	fecha_registro
Johnny	González	jfgonzalezg@unl.edu.ec	\$2a\$10\$whUA8dkyWHXM0zS82JozCelPmSginvN85W46exT3wEOL6x...	1	2022-07-19 00:33:41
NULL	NULL	NULL	NULL	NULL	NULL

Figura 29. Encriptación de contraseña en base de datos.

Requerimiento no funcional – Desarrollo (RNF09),

Para validar este requerimiento no funcional se cargaron 4460 registros de árboles a través de Postman (ver **Figura 31**), utilizando servicios creados para la carga de datos (ver **Figura 30**) y estableciendo el límite del tamaño del cuerpo en las solicitudes a 50mb en Express (ver **Figura 32**). Esto sirvió para almacenar los registros en la base de datos que posteriormente fueron consumidos por la aplicación web.

Como se comprueba en las **Pruebas unitarias al servicio web API-REST**, se utiliza el formato JSON para el envío de peticiones y recepción de respuesta del servicio API REST.

```

148 const createTreeBatch = async (req, res) => {
149   try {
150     const { data } = req.body
151     formattedFields(data)
152
153     const actions = data.map(async record => getActions(record, req.payload.id))
154     const resolverActionsAll = Promise.all(actions)
155     resolverActionsAll.then(actions => {
156       const flaw = actions.filter(obj => obj.typeAction === 'validation')
157       if (flaw.length)
158         return res.status(400).json({
159           message: 'Se detectaron problemas al crear los registros!',
160           flaw,
161           actionsAll: actions,
162         })
163       return res
164         .status(200)
165         .json({ message: 'No se detectaron problemas', actionsAll: actions })
166     })
167   } catch (err) {
168     consola.error('Error:', err)
169     res.status(err.status || 500).json({ message: err.message })
170   }
171 }

```

Figura 30. Controlador para crear registros de árboles por lotes

```
1  {
2    "data": [
3      {
4        "code": "AVPJ02",
5        "recorded": "Cristina Jackeline Arias Godoy",
6        "especieId": 2,
7        "sitioId": 1,
8        "commit": "Árbol muy inclinado. ramas sobre la vía.",
9        "dateRegistration": "2021-04-19"
10     },
11     {
12       "code": "AVPJ03",
13       "recorded": "Cristina Jackeline Arias Godoy",
14       "especieId": 2,
15       "sitioId": 1,
16       "commit": "Árbol muy inclinado. ramas sobre la vía.",
17       "dateRegistration": "2021-04-19"
18     },
19     {
20       "code": "AVPJ04",
21       "recorded": "Cristina Jackeline Arias Godoy",
22       "especieId": 2,
23       "sitioId": 1,
24       "commit": "Árbol muy inclinado. ramas sobre la vía.",
25       "dateRegistration": "2021-04-19"
26     }
27   ]
28 }
```

Figura 31. Envío de registros de árboles en formato JSON

```
15 app.use(morgan('dev'))
16 app.use(express.json({ limit: '50mb' }))
17 app.use(cors())
```

Figura 32. Configuración del límite del cuerpo de la solicitud en Express.

Requerimiento no funcional – Interfaz GUI (RNF08), este requerimiento hace referencia a la portabilidad de la aplicación en los distintos navegadores, pues los mapas de Mapbox trabaja con navegadores que soporten el estándar WebGL (para gráficos 2D/3D), por lo que se probó cinco navegadores popularmente conocidos (ver **Anexo 6**): Chrome, Firefox, Safari, Opera y Edge, verificando que los navegadores en sus versiones actuales son aptos para el correcto funcionamiento de la aplicación.

7. Discusión

La realización de la Aplicación web para la gestión de la representatividad de árboles de la ciudad de Loja, se realizó trabajando de la mano del director e investigadores del proyecto “***Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja***”, quienes han realizado un arduo trabajo en el levantamiento de aproximadamente 5000 árboles y facilitaron los datos necesarios para el desarrollo del presente Trabajo de Titulación.

El proceso de desarrollo se ejecutó en base a los tres objetivos planteado previamente, los cuales se discuten a continuación:

Objetivo 1: Establecer el diseño de la Aplicación Web en base a la metodología ICONIX.

De acuerdo con el análisis de los documentos [9], [10], [26], se ha reflejado que la metodología ICONIX es efectivamente una metodología iterativa e incremental, y en este TT lo comprueba, pues al modelar la arquitectura lógica y física, también conllevó mejorar continuamente a los modelos hasta refinar los requerimientos del cliente de acuerdo al producto de software que necesitaba. A través de esta metodología también se permitió llevar de manera ordenada las actividades que se desarrollan en este objetivo, pues como se especifica en [27], cada fase implica pequeñas tareas que se ejecutan de manera ordenada aplicando técnicas, involucrando a los responsables así obtener los artefactos de cada fase. Este proyecto al aplicar esta metodología se logró adaptar con éxito a las necesidades del cliente y alcanzar los objetivos propuestos. Para esto, se involucró al cliente en la descripción de los requerimientos utilizando la técnica de la entrevista; para luego representar el modelo de dominio de negocio, a partir de este realizar un diseño pequeño del prototipo en Figma (**ver Anexo 4, Prototipo inicial de GUI**), que como menciona [9] lo hace con el fin ser presentado al cliente para refinar los requerimientos iniciales, seguidamente diagramar los casos de uso asociados a cada requisitos, los mismos que fueron detallados en las fichas de casos de uso e implementadas en los diagramas de secuencia, con ello refinar nuevamente nuestros modelos hasta completar en el diseño del diagrama de clases todas las propiedades y relaciones existentes; finalmente representarlas en un modelo entidad-relación que se utilizará en posterior en una base de datos relacional.

Es importante recalcar que ningún trabajo revisado en donde se emplee la metodología ICONIX se logra obtener los modelos finales en una primera versión, por lo que en el proceso siempre involucra al cliente y este retroalimentará en cada iteración, por ende, los modelos se

refinarán hasta llegar a uno donde finalmente se documente y sea válido para continuar con la fase de implementación.

Objetivo 2: Implementar el diseño de la aplicación mediante el framework ReactJS y NodeJS.

En la fase de codificación, las tecnologías de JavaScript permitieron el desarrollo del backend y frontend (**ver sección Marco Teórico, Ítem 4.3**). NodeJS y ExpressJS construyeron las diferentes rutas del API-REST y con Sequelize se integraron los modelos y conexiones con la base de datos MySQL. En cambio, en el frontend se utilizó React, Mapbox y Material UI para las diferentes interfaces de usuario. En [15] se menciona que la capacidad de React de aprovechar el manejo del renderizado de componentes a través de los Hooks (estados de React) y Redux facilitan la actualización de los componentes de forma independiente, y esta solución se aplicó al desarrollo; Redux-Toolkit¹¹ manejó el estado global de la aplicación, conteniendo entre otros estados, la estructura del formato GeoJSON utilizado para incrustar información en los mapas Mapbox y con los hooks useEffect controlar el renderizado del componente. Al igual que [28], en este proyecto Mapbox se acopló muy bien en React por soporte de Mapbox GL JS y amplia documentación proporcionada para su integración, como estilos de mapas, marcadores, ventanas emergentes que se aprovecharon en la aplicación. Finalmente, las interfaces de usuario se integraron utilizando el diseño Material UI que facilitó la creación de los componentes reutilizables que en conjunto hicieron posible una construcción dinámica y escalable de la aplicación. La experiencia previa en React facilitó la comprensión de concepto que maneja Mapbox, Material UI, sin embargo, tuvo su grado de complejidad inicial al incluir tecnologías en el backend antes no utilizadas, pero que, una vez superada este desafío, el proceso de codificación fue más fluida. Por otra parte, estas tecnologías utilizadas en conjunto siguieron el modelo de arquitectura Cliente-Servidor con el patrón de diseño está basada MVC especificados (**ver Figura 27**). El término de este objetivo, se alcanzó debido al proceso de análisis y diseño realizadas en las fases anteriores y la comprensión de modelo lógico y físico para la codificación de los módulos que componen la solución informática.

Objetivo 3: Ejecutar pruebas de funcionalidad para la validación de la Aplicación Web.

Este último objetivo que se desarrolló pertenece a la cuarta fase de la metodología Iconix, que también comprende la realización de pruebas al producto de software buscando la calidad del mismo. Para esto, se tomó los resultados de la etapa de codificación para realizar tres

¹¹ <https://redux-toolkit.js.org/>

pruebas en la aplicación: Pruebas unitarias, Pruebas de caja negra y Pruebas de funcionales (**ver sección Resultados, Objetivo 3**). Las pruebas de integración a los servicios del API-REST se aplicaron con el servicio subido a Railway Server, de esta manera probar con la herramienta Postman en un ambiente de ejecución lo más real (**ver Anexo 5. Pruebas al servicio Web API-REST**), realizando peticiones a los servicios de la aplicación, verificando que las respuestas de cada uno de las peticiones estén en formato legible para el desarrollado (JSON) y midiendo sus tiempos de respuesta. Las pruebas de caja negra lograron detectar comportamientos no esperados como: no cerrar las ventanas emergentes al navegar de página de Inicio a otras páginas, capturar la ubicación al mover un árbol, pero no guardarlo correctamente, algunos problemas a nivel de validaciones mal configuradas. Todos estas issues fueron depurados posteriormente para la aprobación de los casos de prueba (**ver Anexo 7**). Las pruebas funcionales se realizaron con el propósito de descartar errores al usar la aplicación, probar que el software funciona correctamente y cumple con las especificaciones de requerimientos, por ello se evaluó de acuerdo a los roles (Administrador, Investigador), que puedan registrar usuarios, crear nuevos árboles, visualizar reportes, actualizar datos y demás actividades de gestión. Y para la validar los requerimientos no funcionales se probaron en tres aspectos (**ver sección Resultados, Objetivo 3, Ítem 6.3.4**): Seguridad, Desarrollo, Interfaz GUI, donde se verificó que el software, encripte las contraseñas de acceso de los usuarios del sistema, proteja y responda a las peticiones a la API-REST en formato JSON a través de JWT, y funcione en los navegadores web más populares.

Tras culminar el proceso de desarrollo y pruebas de la Aplicación Web se puede responder a la pregunta de investigación:

“¿Cómo se podrá ayudar a los investigadores del Proyecto Forestal a obtener la información relacionada a los registros de árboles urbanos de la ciudad de Loja?”

Se ayudó a los investigadores creando una aplicación web que administra las características dasométricas, morfológicas, ecológicas y de servicios ecosistémicos a través de mapas interactivos de Mapbox, el cual opera bajo el formato de archivo geográfico GEOJSON, que son obtenidos (cargados) de la base de datos relacional MySQL e incrustados en interfaces gráficas la información y acciones de acuerdo al rol de usuario, permitiendo así contar con una herramienta para la gestión del arbolado urbano, que sirve tanto a usuarios, investigadores y administradores para sus futuras investigaciones.

8. Conclusiones

Una vez culminado el Trabajo de Titulación, se puede llegar a las siguientes conclusiones:

- La metodología ICONIX ayudó a organizar todas las actividades llevadas a cabo en la aplicación web, de manera que permitió documentar la arquitectura lógica y física de la aplicación, la misma que es n-tier (cliente-servidor) con el patrón de diseño modelo-vista-controlador (MVC).
- Con la técnica de la entrevista, observación y la interacción continua con los involucrados, se permitió obtener los requerimientos iniciales y llevar a cabo un proceso iterativo e incremental.
- Utilizar JavaScript como lenguaje de programación permitió la integración ORM Sequelize con Express para construir el backend sin ejecutar sentencias SQL, además utilizar Mapbox con React para insertar geográficamente los puntos de interés con los registros de cada árbol en un mapa interactivo.
- La fase de pruebas ayudó a validar que la integración a los servicios API-REST tengan un correcto funcionamiento, además detectar errores que posteriormente fueron corregidos y verificar que aplicación cumple con los requisitos especificados.

Finalmente, se concluye que se desarrolló la versión 1.0 de la Aplicación Web denominada "ÁrbolQR", que permite a través de mapas interactivos de Mapbox la consulta de información científica y técnica de los árboles urbanos de la ciudad de Loja y gestionar los registros dasométricos, morfológicos y ecológicos, dando así contestación a nuestra pregunta de investigación.

9. Recomendaciones

Una vez culminado el Trabajo de Titulación, se puede llegar a las siguientes recomendaciones:

- Efectuar en medida de lo posible reuniones frecuentes con el cliente, como mínimo dos por cada fase de la metodología Iconix, a fin de garantizar que los modelos sean refinados en cada iteración y minimizar cambios post-producción.
- Iniciar la metodología Iconix lo más pronto con un prototipado rápido, que ilustre los conceptos e ideas principales que requiere del cliente, de manera que se oriente las necesidades del cliente al tipo de solución informática que en realidad necesita.
- Tener un nivel de conocimiento básico en tecnologías de JavaScript y Node, con cierto grado de experiencia manejando conceptos React, como los Hooks, puesto que esto le ayudará a encontrar soluciones que solo podrán resolverse con este concepto.
- Utilizar Mapbox, con mayor énfasis, para proyectos cuyo desarrollo esté previsto gestionar complejos datos o crear aplicaciones desde cero, teniendo siempre en cuenta que el servicio es gratuito hasta las 50.000 recargas mensuales, suficiente para entornos de desarrollo o producción con un índice de visitas media a baja.
- Ejecutar las pruebas en un ambiente de control lo más parecido a uno real, debido a que nos dará una idea del tiempo de respuesta promedio que puede tardar nuestra aplicación.
- Utilizar la Metodología de desarrollo ICONIX para proyectos de baja o media complejidad y equipos de tamaño pequeño a mediano, en donde se conozca bien el tipo de solución informática a desarrollar y se requiera documentación detallada que sirva para su mantenimiento.

Trabajos Futuros

- Crear una aplicación móvil para que los investigadores realicen el levantamiento de la información y registro de nuevos árboles en campo y que se sincronice con la aplicación web.
- Integrar un módulo de galería de imágenes para cada árbol, que permita a los usuarios autorizados gestionar las imágenes y a los no autorizados solo visualizar.
- Integrar módulo para la historial de registros de los árboles, para obtener reportes de cuantas veces se ha llevado el registro de las características de los árboles.

10. Bibliografía

- [1] *Código Orgánico del Ambiente*. Quito, 2017. Accessed: Feb. 09, 2023. [Online]. Available: https://www.ambiente.gob.ec/wp-content/uploads/downloads/2018/01/CODIGO_ORGANICO_AMBIENTE.pdf
- [2] UNL, “Vinculación con la Sociedad,” *Universidad Nacional de Loja*, 2021. <https://unl.edu.ec/vinculacion-con-la-sociedad> (accessed Feb. 21, 2023).
- [3] P. Segovia Toscano and A. Berbey-Álvarez, “Plan de desarrollo de una aplicación Web para el control y gestión de deforestación en el Ecuador – ‘Web-CONGESDEFO,’” *Revista de I+D Tecnológico*, vol. 16, no. 2, Jul. 2020, Accessed: Feb. 16, 2023. [Online]. Available: <http://portal.amelica.org/ameli/journal/339/3391369005/html/>
- [4] Y. J. Barriga Garcia and V. Medina Polanco, “Desarrollo de una página web para el control del arbolado del municipio de Fusagasugá,” *Universidad de Cundinamarca, Fusagasugá*, 2022. [Online]. Available: <http://www.>
- [5] C. P. González De Canales, “Beneficios del Arbolado Urbano,” 2002.
- [6] “El Ministerio del Ambiente ampara los árboles de las ciudades del Ecuador con un Acuerdo,” *Ministerio del Ambiente*, 2016. <https://www.ambiente.gob.ec/el-ministerio-del-ambiente-ampara-los-arboles-de-las-ciudades-del-ecuador-con-un-acuerdo/> (accessed Feb. 14, 2023).
- [7] B. Corona Tirado, M. Muñoz Mata, J. Miramontes Sandoval, J. A. Calvo-Manzano Villalón, and T. San Feliu Gilabert, “Estado de arte sobre métodos de evaluación ágiles en las pymes,” *RECIBE: Revista Electrónica de Computación, Informática, Biomédica y Electrónica*, vol. 5, no. 1, pp. 1–18, Apr. 2016, Accessed: Feb. 11, 2023. [Online]. Available: <http://recibe.cucei.udg.mx/Recibe/index.php/Recibe>
- [8] R. G. Figueroa, C. J. Solís, and A. A. Cabrera, “Metodología Tradicionales vs. Metodología Ágiles,” *Loja*. [Online]. Available: <http://www.gpicr.com/msf.aspx>
- [9] E. Porras, “Metodología ágil Iconix en la calidad del producto software, Lima, 2017,” *Universidad Nacional Federico Villarreal, Lima*, 2019.
- [10] B. E. Beatriz-Ospina, “Adecuación de la normativa ISO/IEC 29110 e IEEE 829 para gestión de proyectos de desarrollo con metodología ICONIX,” *Revista Sapientía*, vol. 12, no. 24, pp. 6–15, Nov. 2020.
- [11] L. Jumbo-Flores, P. Quezada-Sarmiento, S. Bustamante-Ordoñez, and E. López-Alama, “Desarrollo de Aplicación Web para la Gestión de Producción de Camarón,” *Revista Espacios*, vol. 39, no. 4, p. 28, 2018, Accessed: May 11, 2022. [Online]. Available: <https://www.revistaespacios.com/a18v39n04/18390428.html>

- [12] L. Olivia *et al.*, “Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software: implementación y publicación de un sitio WEB para una empresa SPIN-OFF en el Sur de Sonora, México.”
- [13] Stackoverflow, “Encuesta para desarrolladores de Stack Overflow 2021,” *2021 Developer Survey*, 2021. <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language> (accessed May 11, 2022).
- [14] C. Martín, A. Unquía, and M. Rubio, *Lenguajes de programación*. Madrid: Editorial UNED, 2021. Accessed: May 09, 2022. [Online]. Available: <https://books.google.es/books?hl=es&lr=&id=qms4EAAAQBAJ&oi=fnd&pg=PA1&dq=lenguajes+de+programaci%C3%B3n+populares+lenguajes+%22lenguajes+de+programaci%C3%B3n%22&ots=pPHSavwP91&sig=FM6bYwxchoqPJS20M3W83gJ6fbw#v=onepage&q=lenguajes%20de%20programaci%C3%B3n%20populares%20lenguajes%20%22lenguajes%20de%20programaci%C3%B3n%22&f=false>
- [15] S. Aggarwal, “Modern Web-Development using ReactJS,” *International Journal of Recent Research Aspects*, vol. 5, no. 1, pp. 133–137, 2018.
- [16] S. Tilkov and S. Vinoski, “Node.js: Using JavaScript to build high-performance network programs,” *IEEE Internet Comput*, vol. 14, no. 6, pp. 80–83, Nov. 2010, doi: 10.1109/MIC.2010.145.
- [17] N. Barsoti and D. Gibertoni, “IMPACTO QUE O SEQUELIZE TRAZ PARA O DESENVOLVIMENTO DE UMA API CONSTRUÍDA EM NODE.JS COM EXPRESS.JS,” *Revista Interface Tecnológica*, vol. 17, no. 2, pp. 231–243, Dec. 2020, doi: 10.31510/infa.v17i2.964.
- [18] S. Hoffmann, L. Alves Pinto, and L. Ricardo Uriarte, “ANÁLISE COMPARATIVA ENTRE AS TECNOLOGIAS DE FRONT-END REACT, ANGULAR E VUE COMPARATIVE ANALYSIS BETWEEN THE FRONT-END TECHNOLOGIES REACT, ANGULAR AND VUE.”
- [19] D. Duarte, *Supercharging Node.js Applications with Sequelize: Create high-quality Node ... - Daniel Durante, Sascha Depold - Google Libros*, 1st ed. Birmingham: Packt Publishing, 2022. Accessed: Feb. 11, 2023. [Online]. Available: <https://books.google.com.ec/books?id=b02OEAAAQBAJ&printsec=frontcover#v=onepage&q&f=false>
- [20] S. Stampolidis, “Bibliotecas de visualización de datos. Análisis y comparación de las librerías: D3, MAPBOX GL y PYGAL,” Universidad de Macedonia, Thessaloniki, 2022. Accessed: Feb. 11, 2023. [Online]. Available: <https://dspace.lib.uom.gr/bitstream/2159/27890/1/StampolidisSpirosMsc2022.pdf>
- [21] J. Arista-Cortes, “APIs en JavaScript para el desarrollo de mapas dinámicos en línea y aplicaciones en agricultura.” 2021, Accessed: May 13, 2022. [Online]. Available:

- <https://www.riego.mx/congresos/comeii2021/files/ponencias/extenso/COMEII-21044.pdf>
- [22] Mapbox, "Documentation | Mapbox," 2022. <https://docs.mapbox.com/> (accessed May 14, 2022).
- [23] Mapbox, "Dynamic Maps for Web and Mobile Apps | Mapbox," 2023. <https://www.mapbox.com/maps> (accessed Mar. 24, 2023).
- [24] F. M. Andrade Gavilanes, "Aplicación web para la gestión de inventario de la estación biológica Pindo Mirador del Gobierno Autónomo Descentralizado Provincial de Pastaza," 2018. Accessed: Feb. 13, 2023. [Online]. Available: <https://dspace.uniandes.edu.ec/bitstream/123456789/9599/1/TUAEXCOMSIS001-2019.pdf>
- [25] E. Vega, "Aplicación web para la gestión de la información del inventario forestal a ser implementado en un servidor web dentro de la intranet de la dirección forestal del Ministerio del Ambiente," Universidad Politécnica Salesiana, Quito, 2016.
- [26] "Metodología ICONIX," *Carla Rebeca Patricia de San Martin Oliva*, 2009. <http://www.portalhuarpe.com.ar/seminario09/archivos/metodologiaiconix.pdf> (accessed Dec. 08, 2022).
- [27] E. Porras, "Análisis de los artefactos del proceso ágil Iconix para evaluar la calidad del producto software, 2017," *Rev. Inv. UNSCH*, pp. 185–194, 2018.
- [28] M. Miller, "Mapbox.js: an engaging open-source web mapping tool for teaching data visualization theory," *SPRING*, vol. 165, pp. 32–37, 2020.
- [29] C. Zapata and G. Garcés, "Generación del Diagrama de Secuencia de UML 2.1.1 desde esquemas preconceptuales," *Revista EIA*, vol. 10, Jul. 2008, Accessed: Jun. 21, 2022. [Online]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1794-12372008000200008

11. Anexo

Anexo 1: Solicitud de Tesista para el desarrollo de la Aplicación Web

 	Universidad Nacional de Loja	Facultad Agropecuaria y de Recursos Naturales Renovables
Of.No.189-DAPC-FARNR-UNL		Docente investigador Ing. Darwin Pucha Cofrep Ph.D. CARRERA DE INGENIERIA FORESTAL Correo: darwin.pucha@unl.edu.ec
		Loja, 06 de junio de 2022
Ingeniero Pablo Fernando Ordoñez Ordoñez Mg. Sc. DIRECTOR DE LA CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN Ciudad. –		
ASUNTO: Solicitud de tesista para el desarrollo de Aplicación Web para Proyecto de Investigación "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja"		
De mi consideración,		
Con un atento saludo me permito exponerle y solicitarle lo siguiente:		
<ul style="list-style-type: none">• Que mediante el Oficio Nro. 362-DI-UNL-2021 la Dirección de Investigación de la Universidad Nacional de Loja aprueba el proyecto "Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja", para ser desarrollado en el periodo 2021-2023, y cuyo director es el PhD. Darwin Alexander Pucha Cofrep.• Que mediante el Oficio Nro. 170-DAPC-FARNR-UNL fue solicitado a la Carrera de Ingeniería en Sistemas y Computación, un tesista para el desarrollo una aplicación móvil, y esta fue culminada en el periodo académico octubre 2021-marzo 2022, como parte de un objetivo principal del proyecto.• Que una de las limitaciones de los dispositivos móviles es el tamaño y capacidad de procesamiento, por ello, se requiere una adecuada administración de los registros digitales y la cómoda representación visual así como administración y registro de datos.		
Con estos antecedentes me permito solicitar a usted muy comedidamente, se tome en consideración la designación de un estudiante de la Carrera de Computación para el desarrollo de la <i>Aplicación Web de Arboles Urbanos de Loja</i> como tema de tesis. El estudiante designado trabajará directamente con el grupo de investigación del proyecto y los lineamientos técnicos para el desarrollo de la aplicación los dará el Ing. Wilmar Chamba.		
Por su favorable atención desde ya le anticipo mis sinceros agradecimientos.		
Atentamente,		
Dr. Darwin Pucha Cofrep DIRECTOR DEL PROYECTO UNL-17-DI-FARNR-2021		 Firmado Digitalmente por: DARWIN ALEXANDER PUCHA COFREP Hora oficial Ecuador: 06/06/2022 12:10
#EducamosParaTransformar		

Figura 33. Solicitud de tesista a Director de CIS

Anexo 2: Entrevista al Director del Proyecto de investigación Forestal



1859



Universidad
Nacional
de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Entrevista al director del proyecto de investigación

“Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”

Proyecto de Investigación	Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja
Nombre del entrevistado	Ing. Darwin Pucha Cofrep
Cargo	Director del Proyecto
Fecha	Viernes 20 de mayo de 2022 (10pm – 11pm)
Objetivo	Comprender los requerimientos de la aplicación web.
Entrevistador	Johnny Fabián González Guamán
Firma Entrevistado	
Firma Investigador	

Entrevista

Objetivo:

Establece los requerimientos a través de la descripción de las necesidades, actividades, el alcance, los límites y problemas que se buscan resolver con la aplicación web.

Desarrollo:

Se procedió a la entrevista presencial de acuerdo la planificación y cronograma de trabajo, el primer acercamiento al Proyecto Forestal se realizó el jueves 5 de mayo en donde se introdujo a la problemática al nivel general del proyecto. La entrevista realizada el viernes 20 de mayo responde a las siguientes preguntas abiertas:

Preguntas:

1. ¿A quiénes está enfocado la aplicación?
2. ¿Qué necesidades o actividades se realizará dentro de la Aplicación Web?
3. ¿Qué funciones considera importantes?

De las entrevistas realizadas se obtuvo la siguiente información:

El Ing. Darwin Pucha manifiesta que el desarrollo de una aplicación web es una necesidad, pues complementa al desarrollo de la aplicación móvil que fue desarrollada para la recolección de datos y otras actividades del proyecto investigación forestal.

La aplicación web debe permitir a los usuarios mostrar la ubicación de los árboles en el mapa, filtrar por especies, sitios, características, seleccionar un árbol en específico y que se muestre un reporte con la información detallada del árbol.

Cada registro de árboles debe permitir validar la información, corregidos de ser necesarios, además, se debe cargar nuevos registros y almacenarlos en el sistema, cuyos campos contienen alrededor de 57 variables y más de 4000 mil registros que aún no se encuentran en base de datos de la aplicación.

Necesitan visualizar los registros de árboles en al menos dos vistas importantes en el mapa, una vista de terreno y otra de vista satelital, debe ser dinámica para moverse por el mapa de la ciudad de Loja (zona de estudio) y mostrar la información del registro o árbol al seleccionarlo. Además, se debe generar reporte de las búsquedas realizadas por medio de filtros.

Los usuarios de la aplicación web serán los mismos definidos desde la aplicación móvil: Usuario común, Investigador y Administrador. El usuario común no necesitará de acceso mediante una cuenta, dispondrá de la aplicación web solamente para consultas información de los registros de árboles, aplicando filtros. El usuario Investigador, tendrá acceso mediante

una cuenta de usuario para realizar actividades de verificación de datos, así como añadir nuevos registros de forma individual, también podrá visualizar información referente al árbol. El administrador podrá realizar todas las tareas de los usuarios anteriormente especificados, además, gestionará todos los registros de árboles como: crear, leer, actualizar y eliminar registros.

Las actividades más importantes que se realizarán en la aplicación web serán: Visualizar en mapa los registros de árboles, filtrar búsquedas, mostrar reportes, actualizar los campos correspondientes al registro del árbol seleccionado, validar registros, crear nuevos registros, descargar reporte y eliminar registros.

Anexo 3: Revisión de documentación física y digital.



1859



Universidad
Nacional
de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Documentación física y digital

Proyecto de Investigación	Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja
Investigador	Ing. Darwin Pucha Cofrep
Cargo	Director del Proyecto
Fecha	Viernes 20 de mayo de 2022
Objetivo	Comprender los requerimientos de la aplicación web a través de los documentos proporcionados de forma física y digital.
Encargado de revisión	Johnny Fabián González Guamán
Firma Encargado	
Firma Investigador	

Base_datos_unificadaG1-G6_15jul2022 (1)

Inicio Insertar Dibujar Disposición de página Fórmulas Datos Revisar Vista

J2115 Molle

	B	C	D	E	F	G	H	I	J	K
	N° único	1. Nombre del colector de datos:	2. Sitio o área de estudio:	3. Sección dentro del sitio:	4. Fecha y hora:	5. Altitud (m s. n. m.)	6 Árbol nro. Por colector	7. Código árbol (4 iniciales del sitio y # de árbol. EJ. AVOZ01)	8. Nombre común	9. Nombre científico
2923	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi	06-10-2021 14:00	2065	67	PQJ600	Aliso	Alnus acuminata	
2924	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi	06-10-2021 14:00	2065	68	PQJ601	Arabisco	Jacaranda mimosifolia	
2927	2925	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi	07-10-2021 8:00	2065	69	PQJ602	Cedro	Cedrela montana
2928	2926	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi	07-10-2021 8:43	2065	70	PQJ603	Cedro colorado	Cedrela odorata
2929	2927	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi	07-10-2021 8:43	2065	71	PQJ604	Sauce	Salix humboldtiana
2930	2928	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi	07-10-2021 12:00	2065	72	PQJ605	Cedro colorado	Cedrela odorata
2931	2929	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi	07-10-2021 12:00	2065	73	PQJ606	Sauce	Salix humboldtiana
2932	2930	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi	07-10-2021 12:00	2065	74	PQJ607	Sauce	Salix humboldtiana
2933	2931	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi	07-10-2021 12:00	2065	75	PQJ608	Arabisco	Jacaranda mimosifolia
2934	2932	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	07-10-2021 9:00	2065	76	PQJ609	Cedro	Cedrela montana
2935	2933	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	07-10-2021 9:00	2065	77	PQJ610	Molle	Schinus molle
2936	2934	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	07-10-2021 9:00	2065	78	PQJ611	Cedro	Cedrela montana
2937	2935	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	08-10-2021 9:00	2065	79	PQJ612	Aliso	Alnus acuminata
2938	2936	Karina Ramón	Parque recreacional Jipiro	Pista de Bicicletas	10-10-2021 8:00	2065	82	PQJ615	Arabisco	Jacaranda mimosifolia
2939	2937	Karina Ramón	Parque recreacional Jipiro	Pista de Bicicletas	10-10-2021 9:00	2065	83	PQJ616	Sauce	Salix humboldtiana
2940	2938	Karina Ramón	Parque recreacional Jipiro	Laguna	16-11-2021 9:00	2065	84	PQJ617	Arabisco	Jacaranda mimosifolia
2941	2939	Karina Ramón	Parque recreacional Jipiro	Frente al Castillo Eurolatino	10-10-2021 10:00	2065	85	PQJ618	Ciprés	Cupressus macrocarpa
2942	2940	Karina Ramón	Parque recreacional Jipiro	Frente al Castillo Eurolatino	10-10-2021 10:00	2065	86	PQJ619	Ciprés	Cupressus macrocarpa
2943	2941	Karina Ramón	Parque recreacional Jipiro	Frente al Castillo Eurolatino	10-10-2021 10:00	2065	87	PQJ620	Ciprés	Cupressus macrocarpa
2944	2942	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	10-10-2021 10:00	2065	88	PQJ621	Molle	Schinus molle
2945	2943	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	08-10-2021 9:00	2065	89	PQJ622	Pomarrosa	Syzygium jambos
2946	2944	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	08-10-2021 9:00	2065	90	PQJ623	Sauce	Salix humboldtiana
2947	2945	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	08-10-2021 9:00	2065	91	PQJ624	Aliso	Alnus acuminata
2948	2946	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	08-10-2021 9:00	2065	92	PQJ625	Arabisco	Jacaranda mimosifolia
2949	2947	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	08-10-2021 9:00	2065	93	PQJ626	Pomarrosa	Syzygium jambos
2950	2948	Karina Ramón	Parque recreacional Jipiro	Av. Salvador Bustamante Celi (Baños	08-10-2021 9:00	2065	94	PQJ627	Sauce	Salix humboldtiana
2951	2949	Karina Ramón	Parque recreacional Jipiro	media del parque (Catedral de San B	08-10-2021 9:00	2065	95	PQJ628	Morera	Morus alba
2952	2950	Karina Ramón	Parque recreacional Jipiro	medi del parque (Catedral de San B	08-10-2021 9:00	2065	97	PQJ630	Cedro	Cedrela montana
2953	2951	Karina Ramón	Parque recreacional Jipiro	Parte media del parque (Torre Eifel)	12-10-2021 8:00	2065	98	PQJ631	Arabisco	Jacaranda mimosifolia
2954	2952	Karina Ramón	Parque recreacional Jipiro	Parte media del parque (Torre Eifel)	12-10-2021 8:00	2065	99	PQJ632	Cepillo chino	Callistemon lanceolatus
2955	2953	Karina Ramón	Parque recreacional Jipiro	Parte media del parque (Torre Eifel)	12-10-2021 8:00	2065	100	PQJ633	Cepillo chino	Callistemon lanceolatus
2956	2954	Karina Ramón	Parque recreacional Jipiro	orilla de la Laguna	12-10-2021 8:00	2065	102	PQJ635	Sauce	Salix humboldtiana
2957	2955	Karina Ramón	Parque recreacional Jipiro	Pistas de patinaje	12-10-2021 11:00	2065	103	PQJ636	Sauce	Salix humboldtiana
2958	2956	Karina Ramón	Parque recreacional Jipiro	Pistas de patinaje	12-10-2021 11:00	2065	104	PQJ637	Pomarrosa	Syzygium jambos
2959	2957	Karina Ramón	Parque recreacional Jipiro	Pistas de patinaje	12-10-2021 11:00	2065	105	PQJ638	Guabo	Inga insignis
2960	2958	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	106	PQJ639	Sauce	Salix humboldtiana
2961	2959	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	107	PQJ640	Aliso	Alnus acuminata
2962	2960	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	108	PQJ641	Morera	Morus alba
2963	2961	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	109	PQJ642	Sauce	Salix humboldtiana
2964	2962	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	110	PQJ643	Sauce	Salix humboldtiana
2965	2963	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	111	PQJ644	Aliso	Alnus acuminata
2966	2964	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	112	PQJ645	Luma	Pouteria locuma
2967	2965	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	113	PQJ646	Aliso	Alnus acuminata
2968	2966	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	114	PQJ647	Aliso	Alnus acuminata
2969	2967	Karina Ramón	Parque recreacional Jipiro	Rio zamora	12-10-2021 11:00	2065	115	PQJ648	Aliso	Alnus acuminata

TOTAL-CORREGIDA Hoja8 Hoja9 +

Listo 100%

Figura 34. Inventario de árboles en formato xlsx.

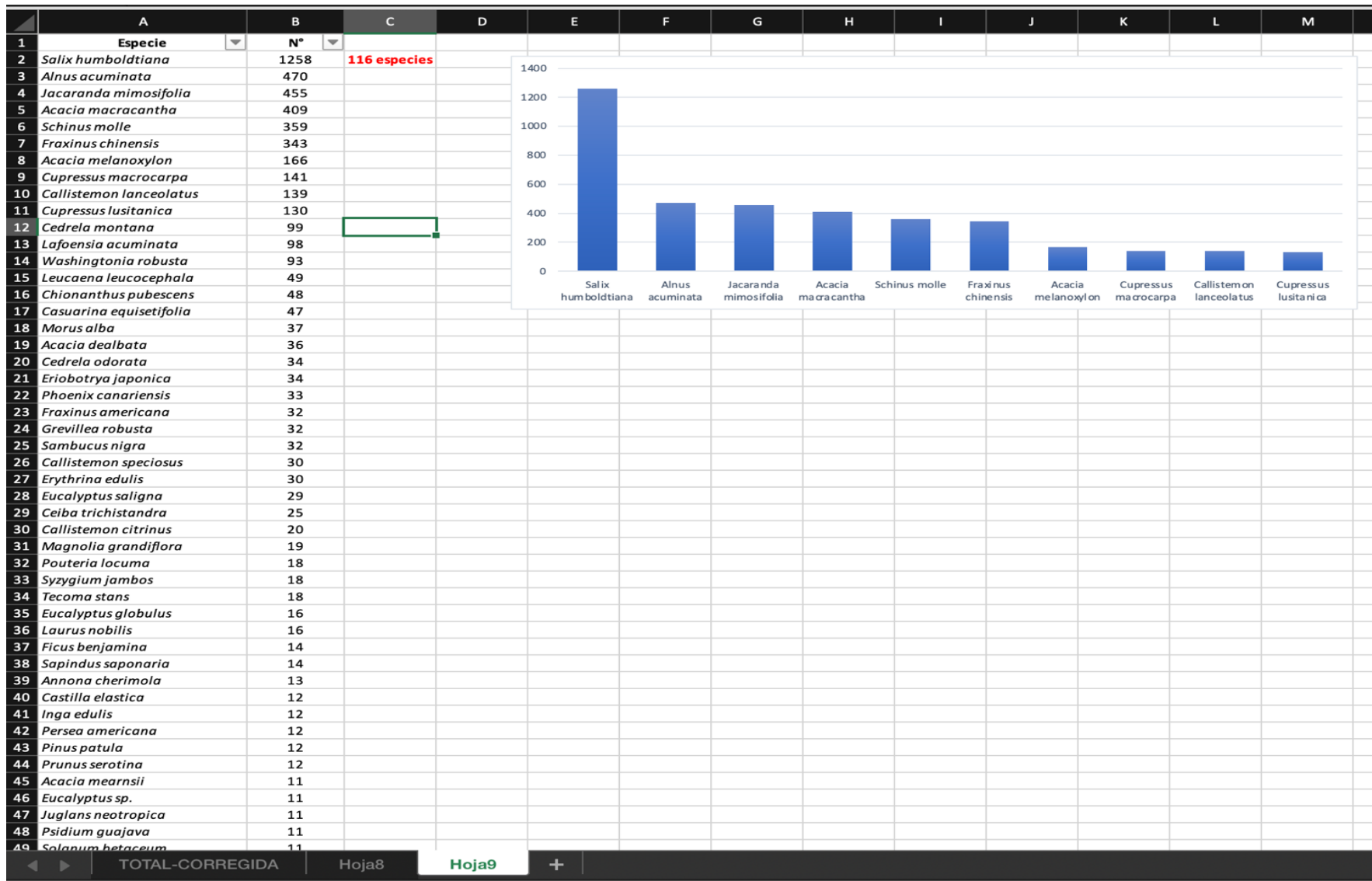


Figura 35. Inventario de especies de árboles

FICHA DE CAMPO PARA EVALUACIÓN FITOSANITARIA DE ÁRBOLES URBANOS

Ficha No.:

1. Nombre del colector de datos:
2. Sitio o área de estudio:
- 2.1. Tipo de sitio: (tipo de dato: lista)
- Parque
 - Avenida
 - Otro [defina].....
- 2.2. Parroquia: ...(tipo de dato: lista).....
3. Sección dentro del sitio
4. Fecha – hora: 5. Altitud (m s. n. m.):

6.	Árbol nro.	1	2	3
	INFORMACIÓN GENERAL			
7.	Código árbol (Iniciales del sitio y # de árbol. EJ. Av02001)			
8.	Nombre común			
9.	Nombre científico			
9.1.	FAMILIA (texto)			
10.	Coordenadas X (long) :			
11.	Coordenadas Y (lat) :			
12.	CAP (cm):			
13.	DAP (cm):			
14.	HT (m):			
15.	HC (m):			
16. 1	Tamaño de copa promedio (m): Norte – Sur: (tipo Double)			
16.2	Tamaño de copa promedio (m): Este – Oeste: (tipo Double)			
17.	Follaje o porcentaje de hojas (%):			
18.	Estado de madurez (joven, adulto, viejo)			
19.	Floración (Terminada, en floración, iniciada, en floración)			
20.	Fructificación (Terminada, en fructificación, iniciada, en fruct.)			
21.	Rectitud del fuste (muy inclinado, inclinado, recto, torcido)			
22.	Espacio de crecimiento (Amplio, moderado, estrecho)			
	ESTADO FITOSANITARIO			
23.	Raíz (Buena, regular, mala):			
24.	Fuste (Buena, regular, mala):			
25.	Corteza (Buena, regular, mala):			
26.	Ramas (Buena, regular, mala):			
27.	Hojas (Buena, regular, mala):			
28.	Cima del árbol (Yema apical): (Buena, regular, mala):			
29.	Copa (Buena, regular, mala)			
30.	Presencia de enfermedades (Si/No)			
	- Escudados			
	- Tumores			

Figura 36. Ficha de campo para evaluación fitosanitaria de árboles urbanos.

Anexo 4: Desarrollo de la metodología ICONIX



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Desarrollo de la metodología ICONIX

Proyecto: “Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”.

Autor:

Johnny Fabián González Guamán

Desarrollo de la metodología ICONIX

Para desarrollar la metodología ICONIX destacan cuatro principales Fases [10]:

- a) Análisis de requisitos
- b) Análisis y Diseño Preliminar
- c) Diseño
- d) Implementación

Previa a la realización de la primera Fase se realizó la especificación de requerimientos del sistema mediante una entrevista al Director del Proyecto Forestal, el cual se detalla a continuación:

1. FASE INICIAL: Especificación de requerimientos del sistema

1.1. Propósito del sistema

El director del Proyecto Forestal “Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”, ha solicitado que se desarrolle un aplicativo web para la consulta y representación de árboles de la ciudad de forma libre.

Durante la reunión de levantamiento de requerimientos los investigadores del proyecto forestal han indicado que dicho aplicativo debe permitir a todos los usuarios ver la ubicación de los árboles en un mapa de puntos de interés. Por defecto, el sistema presentará todos los árboles registrados y la opción de mostrar una información general de dichos árboles.

El aplicativo permitirá al usuario filtrar las búsquedas de árboles por especies, familias; como las características más relevantes. El usuario podrá seleccionar un punto de interés específico del mapa (representación de árbol) y el sistema mostrará información preliminar del árbol, si el usuario decide obtener más información, el aplicativo presentará el detalle completo del árbol.

Los solicitantes han indicado que el mapa debe permitir el cambio entre dos tipos de vistas, vista satelital y vista de terreno para su mejor visualización en el mapa.

El aplicativo debe permitir el acceso mediante una cuenta de usuario a dos grupos: al usuario con rol de Investigador y al usuario con rol de Administrador. El sistema debe validar el acceso mediante correo y contraseña, y contará con un método de recuperación de cuenta, que consistirá en el cambio de contraseña a través de la validación de su credencial por correo electrónico.

Al usuario con rol de Investigador el aplicativo debe permitir, previo acceso a su cuenta, editar las características de un árbol seleccionado y validar si el árbol ha sido registrado correctamente modificando su estado: verificado y no verificado. Para editar la geoubicación del árbol el sistema debe permitir ingresar manualmente la posición y también la opción de

desplazar el punto de interés a la nueva ubicación. Se podrá agregar nuevos árboles de forma individual.

También han indicado que, además de las acciones anteriores, el usuario con rol de Administrador, debe poder seleccionar un árbol específico y eliminarlo, descargar el reporte de uno o varios árboles en un archivo csv.

1.2. Reconocimiento de los Stakeholder (Actores)

- Administrador
- Investigador
- Usuario

1.3. Restricciones técnicas y Alcances

Entorno de ejecución. - La aplicación va ser ejecutada desde cualquier entorno web, por tanto, se requiere de un browser actualizado con acceso a internet para las diferentes plataformas de escritorio (Windows, Mac, Linux). Se recomienda que el browser esté habilitado con el estándar WebGL para el despliegue de gráficos 2D/3D, como por ejemplo Chrome 96, Edge 96, Firefox 97, Firefox 91 (ESR), iOS Safari 14, Safari 14, o versiones posteriores a las especificadas y al menos 4GB de memoria RAM DDR3 para las plataformas de escritorio. Con respecto a la visualización de árboles se empleará MapBox como servidor de mapas web, además de la API Mapbox Style para definir la apariencia visual del mapa: Satellite y Streets, y proporcionar al renderizado de mapas los valores iniciales. Finalmente, para consumir los servicios web se utilizará la infraestructura API RESTful con el formato de datos JSON y las peticiones que se ejecutarán a la base de datos de la aplicación serán las siguientes:

- **GET:** Consulta de atributos o características de los árboles como los registros forestales: riesgos potenciales, problemas físicos, presencia de enfermedades, ubicación de enfermedades, presencias de plagas, estado fitosanitario, acciones recomendadas; además de consulta a usuarios.
- **POST:** Creación de nuevos registros forestales de árboles.
- **PUT:** Actualización de los registros forestales; y actualización de contraseñas.
- **DELETE:** Eliminación de los registros de árboles.

Entorno de desarrollo. – El frontend del aplicativo web estará desarrollado con el framework ReactJS el cual utiliza el lenguaje de programación JavaScript y para el backend con librerías de NodeJS. El API REST estará desarrollado con ExpressJS y se conectará con la base de datos relacional MySQL para la integración de datos con el aplicativo móvil del proyecto. Se utilizará también Mapbox GL JS que utilizan WebGL para desarrollar y representar los mapas interactivos.

1.4. Requerimientos Funcionales

En la **Tabla 12** se presentan los requerimientos funcionales del sistema obtenidos en la entrevista con el director del proyecto forestal y su equipo de trabajo.

Tabla 12. Requerimientos Funcionales

ID	Requisito	Descripción
RF01	Visualizar árboles	El sistema debe permitir al usuario visualizar la ubicación de los registros de árboles en el aplicativo a través de un mapa web de Mapbox.
RF02	Mostrar tipos de mapas base	El sistema debe permitir al usuario cambiar entre diferentes vistas de mapas base del servidor Mapbox como: Satelital y de Calles.
RF03	Visualizar reportes	El sistema debe permitir al usuario visualizar dos tipos de reportes: a) Un reporte preliminar con los detalles mínimos del árbol seleccionado en el mapa. b) Un reporte completo para el mismo árbol seleccionado. El sistema debe permitir mostrar el identificador del árbol en un código QR.
RF04	Filtra consultas	El sistema debe permitir al usuario filtrar las búsquedas de árboles por especies, sitio y características consideradas como las variables más relevantes.
RF05	Autenticar usuario	El sistema debe permitir a los usuarios registrados ingresar a la aplicación a través de las credenciales como: nombre de usuario (email) y contraseña.
RF06	Actualizar contraseña	El sistema debe permitir al usuario modificar su contraseña desde su cuenta de usuario y se enviará la contraseña modificada al email.
RF07	Recuperar contraseña	En caso de olvido de contraseña, el sistema debe ayudar al usuario a recuperar el acceso a su cuenta, para ello el sistema validará el email ingresado y se enviará un correo electrónico para el cambio de contraseña.
RF8	Editar árboles	El sistema debe permitir al usuario registrado editar las características de un árbol específico. Para los campos de latitud y longitud, además de digitar manualmente, se debe contar con la opción de desplazar el punto de interés a la nueva posición.
RF9	Registrar árbol	El sistema debe permitir al usuario registrado agregar nuevos árboles llegando los campos de un formulario de registro.
RF10	Eliminar árbol	El sistema debe permitir al usuario Administrador eliminar un árbol del registro forestal (se espera que la información vinculada con el registro del árbol no se muestre al usuario final).
RF11	Descargar reportes	El sistema debe permitir al usuario Administrador descargar el reporte de uno o varios árboles seleccionados en un archivo CSV.
RF12	Registrar cuenta	El sistema debe permitir al usuario registrarse en para crear una cuenta en la aplicación web.
RF13	Validar cuenta	El sistema debe permitir al administrador validar cuenta de usuario registrado.

1.5. Requerimientos no Funcionales

En **Tabla 13** se presentan los requerimientos no funcionales de acuerdo a los requerimientos y necesidades del proyecto.

Tabla 13. Requerimiento No Funcionales

ID	Requisitos	Descripción
RNF01	Seguridad	La información manejada por el sistema estará protegida del acceso no autorizado y divulgación. El sistema utilizará un método de autenticación JWT para el acceso a la cuenta de usuario.
RNF02	Usabilidad	La aplicación mantendrá una interfaz intuitiva, de fácil uso para el óptimo manejo del usuario. El tiempo de aprendizaje del sistema por usuario será menor a 2 horas.
RNF03	Rendimiento o Calidad	La funcionalidad del sistema, informes y consultas que presenten una complejidad mediana, sus tiempos de respuesta hacia el usuario debe responder en menos de 7 segundos.
RNF04	Fiabilidad	El sistema debe garantizar la tolerancia a fallos de conexión proporcionando mensajes informativos y orientados a los usuarios.
RNF05	Persistencia	El sistema debe garantizar la recuperación y guardado de los datos persistentes utilizados por la aplicación en la base de datos relacional MySQL.
RNF06	Sincronización	Los datos modificados o agregados en el sistema deben ser actualizados por todos los usuarios que accedan a ellos en menos de 3 segundos.
RNF07	Disponibilidad	El sistema debe estar activo 24 horas, los 7 días de la semana, los 365 días del año, garantizando toda la funcionalidad del sistema.
RNF08	Mantenibilidad	Será desarrollado de forma modular para mantener una estructura de código consistente y predecible.
RNF09	Desarrollo	Debe funcionar a través de respuestas consumiendo servicios web API REST bajo el formato de datos JSON.
RNF10	Interfaz GUI	La interfaz se ejecutará en entornos web, se utilizará un browser habilitado con el estándar WebGL para el despliegue de gráficos 2D/3D, como por ejemplo Chrome 96, Edge 96, Firefox 97, Firefox 91 (ESR), iOS Safari 14, Safari 14, o versiones posteriores a las especificadas.

2. FASE 1: Análisis de Requisitos

2.1. Modelo de Dominio

Este diagrama es obtenido y adaptado de acuerdo a la base datos de la aplicación móvil del proyecto forestal. En la **Figura 37** se muestra la representación del modelo de dominio del negocio. También, se muestra en la **Figura 38** el modelo de dominio de seguridad.

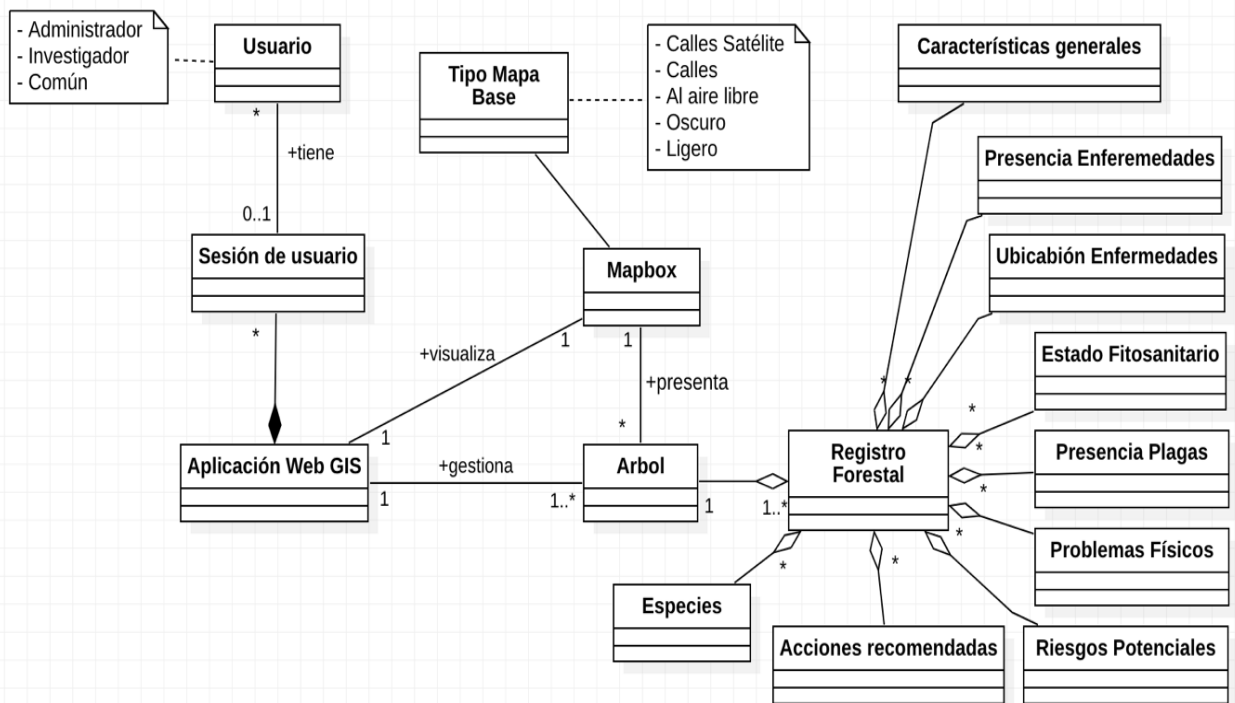


Figura 37. Modelo de dominio del negocio

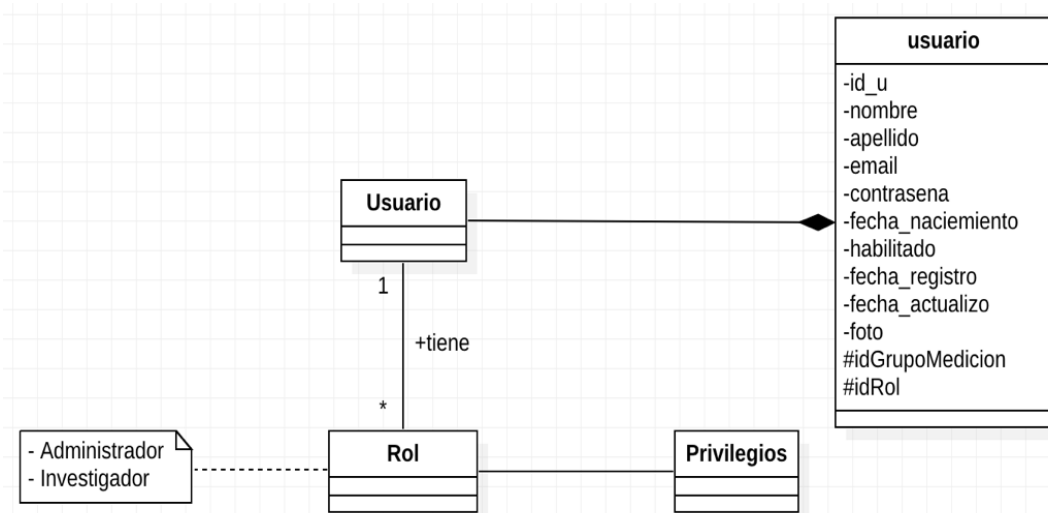


Figura 38. Modelo de dominio de seguridad

2.2. Casos de Uso

En la **Figura 39** se muestra el diseño de Caso de Uso obtenidos de los requerimientos funcionales del sistema.

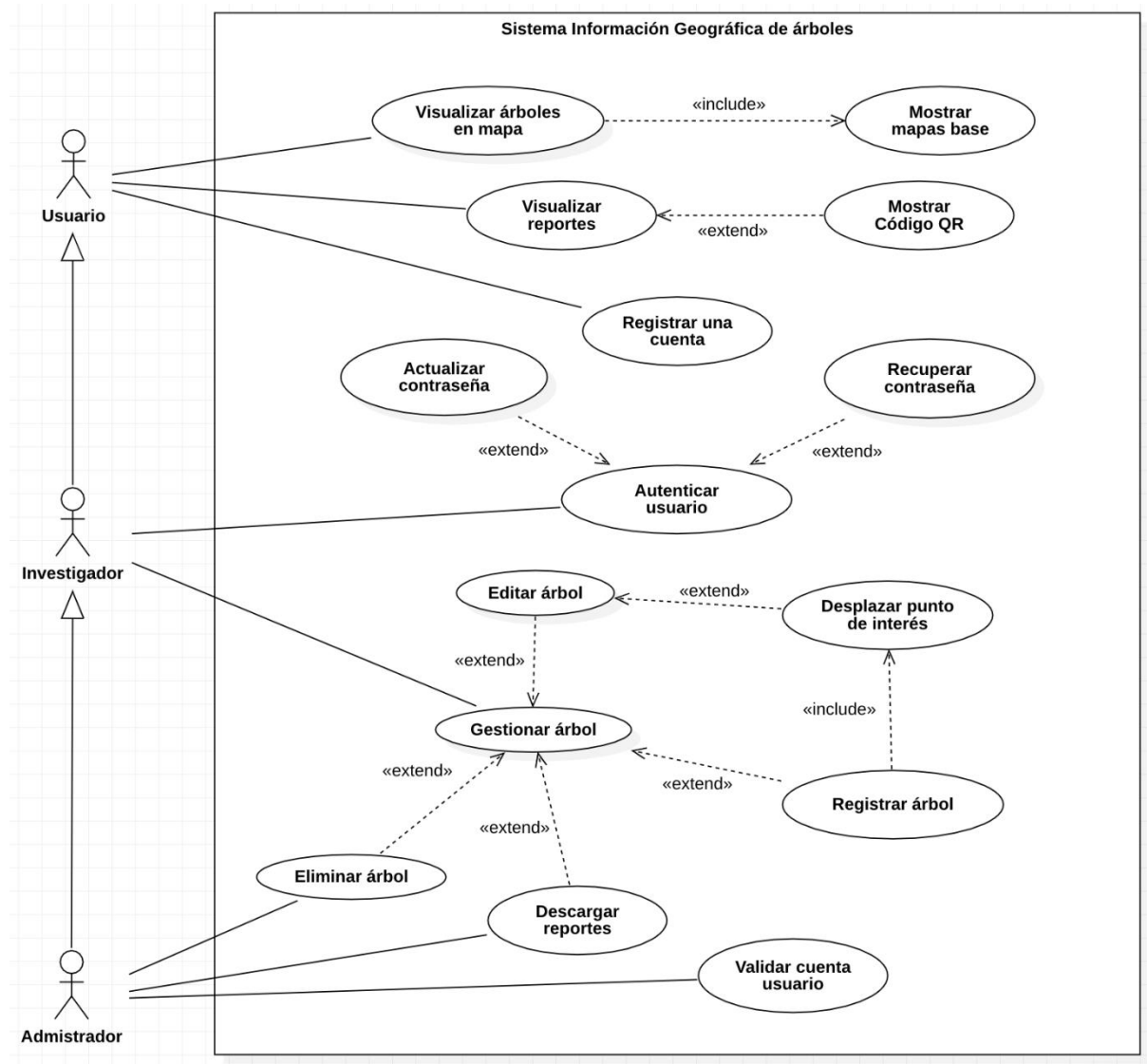


Figura 39. Caso de Uso para el Sistema de Información Geográfica de árboles.

2.3. Prototipo inicial de GUI

Se utilizó la herramienta de diseño Figma para realizar un prototipo de interfaz de usuario inicial, el diseño y concepto se toma de aplicaciones desarrolladas con ArgGIS y se combinan con componentes de Material-UI. En la **Figura 40** se muestran diferentes escenarios de prototipos, para mayor detalle visite el siguiente enlace: [Prototipo App Web](#).

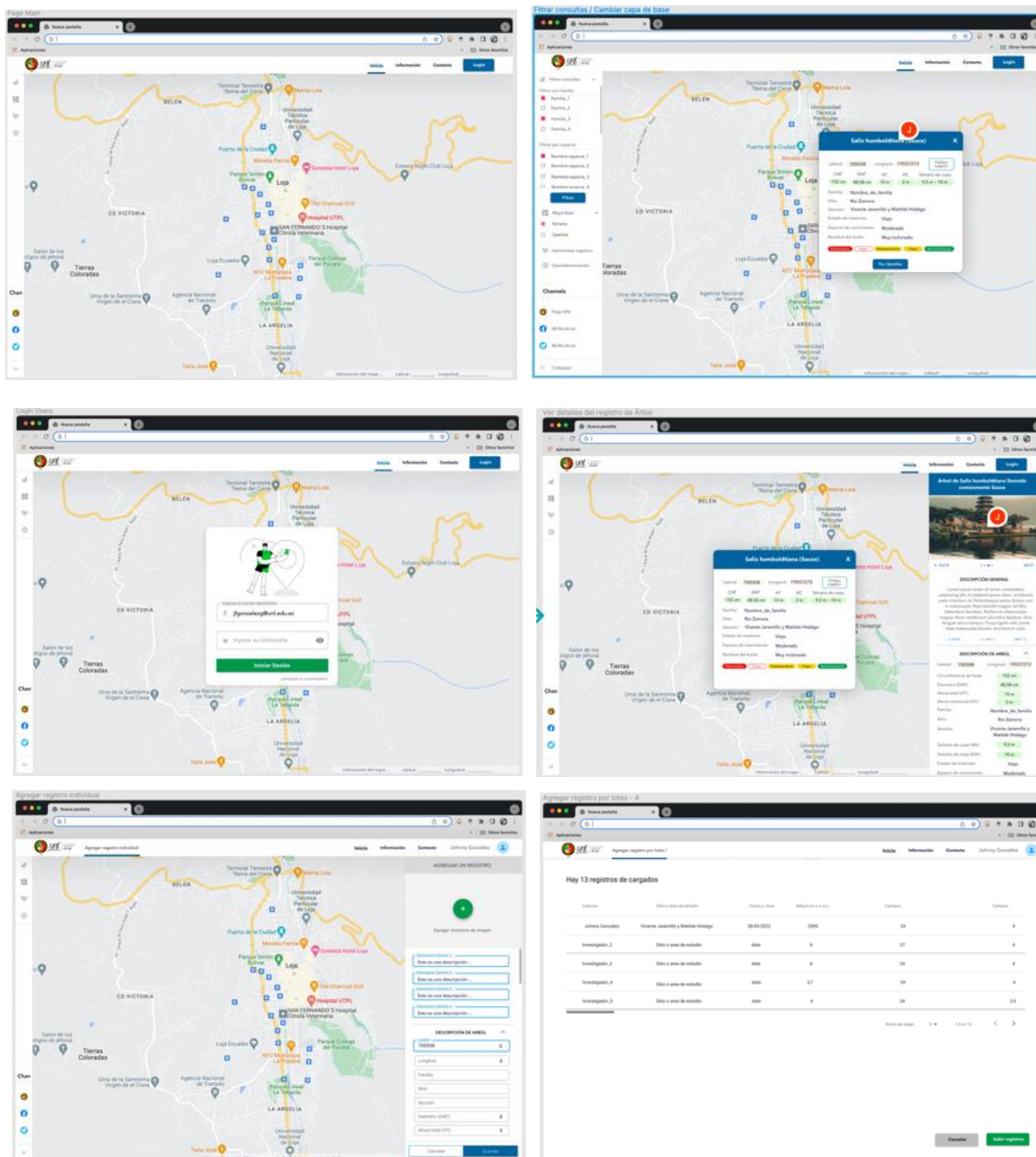


Figura 40. Diseño de prototipo inicial de GUI para la aplicación web.

3. FASE 2: Análisis y Diseño Preliminar

3.1. Ficha de casos de uso

Los casos de uso describen mediante las acciones y reacciones el comportamiento de un sistema desde el punto de vista de un usuario, para definir los límites del sistema y relaciones entre el sistema y entorno. A continuación, se describen los casos de uso en las siguientes tablas.

Tabla 14. Fichas de caso de uso Visualizar árboles

CU01	Actores: Usuario/Administrador/ Investigador	Prioridad: Alta
Caso de Uso: Visualizar árboles en mapa		
Descripción: El usuario visualiza los registros de árboles como puntos de interés en el mapa de Mapbox y cambia el mapa base de terreno (calles) a satelital.		
Referencias: RF01, RF02.		
Pre-Condiciones:		
<ul style="list-style-type: none"> - El usuario se encuentra en la página principal del aplicativo web. 		
Post-Condiciones:		
<ul style="list-style-type: none"> - Se alterna dinámicamente el estilo del mapa a visualizarse. - El sistema representa las ubicaciones de árboles como puntos de interés en el mapa con un color identificativo. 		
Flujo normal de los eventos:		
Usuario no autenticado		
<ol style="list-style-type: none"> 1. El usuario visualizar un mapa web de Mapbox. 2. El sistema presenta grupos de árboles (puntos de interés) en el mapa. 3. El usuario hace zoom en el mapa. 4. El sistema carga más puntos de interés en el mapa. 5. El usuario selecciona el mapa base que desea utilizar: Terreno o Satelital. 6. El sistema actualiza el mapa de Mapbox. 		
Flujo Alternativo de eventos:		

Tabla 15. Ficha de caso de uso Visualizar reportes

CU02	Actores: Usuario/Administrador/Investigador	Prioridad: Alta
Caso de Uso: Visualizar reportes		
Descripción: El usuario desea obtener información de un árbol específico (punto de interés).		
Referencias: RF03, RF04.		
Pre-Condiciones:		
<ul style="list-style-type: none"> - El usuario debe aplicar la búsqueda de árboles según el tipo de información que desea visualizar. - El usuario debe seleccionar un registro de árbol en el mapa interactivo. 		
Post-Condiciones: El sistema realiza consultas y presenta reportes con diferentes niveles de detalles.		
Flujo normal de los eventos:		
Visualizar reporte preliminar		
<ol style="list-style-type: none"> 1. El usuario debe selecciona el árbol de interés en el mapa. 2. El sistema presenta un reporte preliminar con información básica del árbol. 3. El usuario selecciona mostrar código QR. 4. El sistema presenta el código QR con la identificación del árbol. 5. El usuario selecciona el formato imagen para descargar el código QR. 6. El sistema descarga en el formato imagen en el equipo. 		
Visualizar reporte completo		
<ol style="list-style-type: none"> 7. El usuario selecciona el botón <i>Ver más</i>. 8. El sistema presenta un reporte individual con información completa del árbol. 		
Flujo Alternativo de eventos:		

Tabla 16. Ficha de caso de uso Autenticar usuario

CU03	Actores: Administrador/Investigador	Prioridad: Alta
Caso de Uso: Autenticar usuario		
Descripción: Se inicia cuando el usuario desea acceder a su sesión del usuario.		
Referencias: RF05, RF06, RF07.		
Pre-Condiciones:		
<ul style="list-style-type: none">- El usuario debe ser previamente registrado.- El usuario haya ingresado al formulario de autenticación (Login).		
Post-Condiciones:		
<ul style="list-style-type: none">- Crea la sesión de usuario y logueado en el sistema.- Actualiza la contraseña actual y permite iniciar sesión con la credencial actualizada.		
Flujo normal de los eventos:		
Autenticar usuario		
<ol style="list-style-type: none">1. El usuario ingresa sus credenciales de ingreso (email y contraseña).2. El sistema verifica que el formato de correo electrónico ingresado esté correcto.3. El usuario presiona Iniciar Sesión.4. El sistema verifica que el usuario exista a través de las credenciales de acceso.5. El sistema muestra la página principal de la aplicación con su sesión de usuario.		
Recuperar contraseña		
<ol style="list-style-type: none">6. El usuario hace clic en recuperar contraseña en el formulario autenticación.7. El sistema presenta un formulario para ingresar correo electrónico.8. El usuario ingresa su correo electrónico.9. El sistema verifica que el formato de correo electrónico ingresado esté correcto.10. El usuario presiona Enviar Correo.11. El sistema verifica que la credencial ingresada exista.12. El sistema envía un correo electrónico al email ingresado con instrucciones.13. El usuario cambia a una nueva contraseña.14. El sistema verifica que los campos coincidan.15. El sistema actualiza la contraseña actual y retorna a la página principal.		
Actualizar contraseña		
<ol style="list-style-type: none">16. El usuario selecciona el perfil de su cuenta en la parte superior.17. El sistema presenta información del usuario autenticado.18. El usuario ingresa los campos para cambiar contraseña (contraseña nueva, repetir contraseña nueva).19. El sistema verifica que los campos de contraseñas.20. El sistema actualiza la nueva contraseña.		
Flujo Alternativo de eventos:		
A. Formato de correo electrónico inválido		
[2.1.] – [9.1.] El sistema muestra un mensaje de validación indicando que el formato está incorrecto.		
B. Credenciales no coinciden o no existen		
[4.1.] – [11.1] El sistema muestra un mensaje indicando que sus credenciales no coinciden o no existen.		
C. Contraseñas nuevas no coinciden		
[14.1] – [19.1.] El sistema muestra un mensaje de no coincidencia. Los campos Contraseña nueva y Repetir contraseña nueva no coinciden.		

Tabla 17. Ficha de caso de uso Crear usuario

CU04	Actores: Administrador/Investigador	Prioridad: Alta
Caso de Uso: Gestionar árbol – Editar árbol		
Descripción: El usuario puede editar las características de un árbol, los campos de latitud y longitud, se puede desplazar el punto de interés a una nueva ubicación en el mapa.		
Referencias: RF8.		
Pre-Condiciones:		
<ul style="list-style-type: none"> - El usuario Administrador e Investigador deben iniciar sesión y en la sección del menú Administrar registro. - El usuario debe haber buscado el árbol específico para visualizar las acciones disponibles. 		
Post-Condiciones: Se actualiza el árbol seleccionado con los nuevos valores dentro de la aplicación y el usuario visualiza los campos actualizados.		
Flujo normal de los eventos:		
<ol style="list-style-type: none"> 1. El usuario selecciona de la lista de acciones, Editar. 2. El sistema obtiene la información correspondiente al árbol seleccionado y lo muestra en un formulario para su edición con los campos bloqueados. 3. El usuario hace clic en “Habilitar edición”. 4. El sistema habilita los campos para editar y muestra los botones de guardado. 5. El usuario actualiza uno o varios campos. 6. El sistema verifica que el formato del dato ingresado es correcto. 7. El usuario presiona Guardar. 8. El sistema almacena los datos modificados de forma permanente y presenta un mensaje de guardado correctamente. 		
Flujo Alternativo de eventos:		
A. Formato de dato ingresado inválido		
[6.1.] El sistema muestra un mensaje de validación indicando que el formato está incorrecto.		
B. Usuario Cancela la acción editar		
[7.1.] El sistema no almacena los datos editados y redirecciona al inicio.		
C. Actualizar campo latitud y longitud		
[5.1.] El usuario selecciona la acción Desplazar ubicación para latitud y longitud.		
[5.2.] El sistema habilita el desplazamiento del punto de ubicación.		
[5.3.] El usuario desplaza el punto a la ubicación deseada.		
[5.4.] El sistema cambia las coordenadas de ubicación a la posición actual.		
[5.5.] El caso de uso continúa en el punto (6) del flujo normal de eventos.		

Tabla 18. Ficha de caso de uso Registrar árbol

CU05	Actores: Administrador/Investigador	Prioridad: Alta
Caso de Uso: Gestionar árbol – Registrar árbol		
Descripción: El usuario desea ingresar nuevos registros de árboles en el sistema.		
Referencias: RF9		
Pre-Condiciones:		
<ul style="list-style-type: none"> - El usuario Administrador e Investigador deben iniciar sesión y estar logueado. - El usuario haya ingresado al menú Registrar nuevo. 		
Post-Condiciones: Se registra un árbol en el sistema.		
Flujo normal de los eventos:		
<ol style="list-style-type: none"> 1. El usuario completa los campos obligatorios 2. El sistema verifica que el formato en los campos ingresados esté correcto. 3. El usuario presiona Guardar. 4. El sistema verifica que los campos obligatorios estén llenos. 5. El sistema almacena la información en el registro forestal del aplicativo. 		

-
6. El sistema muestra un mensaje de registro agregado correctamente.
-

Flujo Alternativo de eventos:

- A. Formato de dato ingresado incorrecto**
[2.1.] El sistema muestra un mensaje que el formato ingresado no es correcto.
- B. Usuario descarta acción de registrar.**
[3.1.] El sistema mostrará la página principal.
- C. Campos obligatorios no llenos**
[4.1.] El sistema muestra un mensaje que los campos son obligatorios.
-

Tabla 19. Ficha de caso de uso Eliminar árbol

CU06	Actores: Administrador	Prioridad: Media
-------------	-------------------------------	-------------------------

Caso de Uso: Gestionar árbol – Eliminar árbol

Descripción: El administrador elimina un árbol del registro forestal del aplicativo.

Referencias: RF10.

Pre-Condiciones:

- El usuario Administrador haya iniciado sesión.
- El usuario haya generado un reporte general de árboles.

Post-Condiciones: El registro forestal del árbol es eliminado del sistema.

Flujo normal de los eventos:

1. El usuario selecciona un registro de la tabla del reporte general
2. El sistema habilita la opción Eliminar registro.
3. El usuario presiona Eliminar.
4. El sistema muestra un mensaje de confirmación para eliminar el árbol.
5. El usuario confirma la acción de eliminación.
6. El sistema elimina internamente el registro y muestra un mensaje de registro eliminado.

Flujo Alternativo de eventos:

A. Eliminar registro
[4.1.] El usuario cancela su acción de eliminar registro, el sistema cierra el mensaje de confirmación.

Tabla 20. Ficha de caso de uso Descargar reporte

CU07	Actores: Administrador	Prioridad: Media
-------------	-------------------------------	-------------------------

Caso de Uso: Gestionar árbol – Descargar reporte

Descripción: El administrador descarga uno o varios registros de árboles en un archivo CSV.

Referencias: RF11.

Pre-Condiciones:

- El usuario Administrador haya iniciado sesión.
- El usuario se encuentra en la sección Reporte general de árboles.

Post-Condiciones: Se genera un archivo CSV con información de los árboles seleccionados.

Flujo normal de los eventos:

1. El usuario aplica filtros en la tabla para traer los registros deseados.
2. El sistema muestra los resultados filtrados en la tabla.
3. El usuario selecciona las columnas de la tabla que serán visibles.
4. El sistema oculta/muestran las columnas habilitadas.
5. El usuario presiona Descargar CSV.
6. El sistema genera un archivo CSV con los registros de la tabla.
7. El sistema descarga automáticamente el archivo.

Flujo Alternativo de eventos:

A. Usuario no filtra resultados

[1.1.] El sistema muestra por defectos todos los resultados paginados en la tabla.
[1.2.] El caso de uso continúa en el punto (3) del flujo normal de eventos.

B. Usuario no selecciona columnas

[2.1.] El sistema muestra por defectos todas las columnas.

[2.2.] El caso de uso continúa en el punto (5) del flujo normal de eventos.

Tabla 21. Ficha de caso de uso Registrar cuenta

CU08	Actores: Usuario no registrado	Prioridad: Media
Caso de Uso: Registrar cuenta		
Descripción: El usuario crea una cuenta desde la aplicación web.		
Referencias: RF12.		
Pre-Condiciones:		
<ul style="list-style-type: none">- El usuario no tenga una cuenta registrada en la aplicación o se encuentre logueado.- El usuario se encuentre en la página de crear cuenta.		
Post-Condiciones: Se crea un usuario con estado de cuenta no habilitado.		
Flujo normal de los eventos:		
<ol style="list-style-type: none">1. El usuario llena los campos para el registro.2. El sistema verifica que el formato de los campos ingresados sea el correcto.3. El usuario hace clic en el botón <i>Registrarse</i>.4. El sistema registra al usuario con estado deshabilitado.5. El sistema envía un correo electrónico al administrador para validación de cuenta.6. El sistema muestra un mensaje de cuenta en estado pendiente hasta su validación.		
Flujo Alternativo de eventos:		
A. Formato de correo electrónico inválido		
[2.1.] El sistema muestra un mensaje de validación indicando que el formato está incorrecto.		
B. Contraseñas nuevas no coinciden		
[2.2.] El sistema muestra un mensaje de no coincidencia. Los campos Contraseña nueva y Repetir contraseña nueva no coinciden.		

Tabla 22. Ficha de caso de uso Validar Cuenta

CU08	Actores: Administrador	Prioridad: Media
Caso de Uso: Validar Cuenta		
Descripción: El administrador habilita los usuarios que se han registrado en la aplicación web.		
Referencias: RF12.		
Pre-Condiciones:		
<ul style="list-style-type: none">- El administrador haya recibido el correo electrónico del nuevo usuario registrado.- El administrador haya hecho clic en el link para habilitarlo.		
Post-Condiciones: Se crea un usuario con estado de cuenta no habilitado.		
Flujo normal de los eventos:		
<ol style="list-style-type: none">1. El sistema direcciona el enlace a la página de validación de nueva cuenta.2. El administrador hace clic en el botón habilitar.3. El sistema habilita al usuario y se activa la cuenta.		
Flujo Alternativo de eventos:		
A. Enlace no válido, token vencido.		
[2.1.] El sistema muestra un mensaje que el enlace ya ha vencido.		

4. FASE 3: Diseño

4.1. Diagramas de Secuencia

El diagrama de secuencia describe los aspectos dinámicos de un sistema a través de un esquema conceptual que muestra las interacciones entre objetos y los mensajes enviados entre ellos, con el fin de llevar a cabo una transacción del sistema [29]. En las siguientes figuras se muestran las interacciones correspondientes a cada caso de uso.

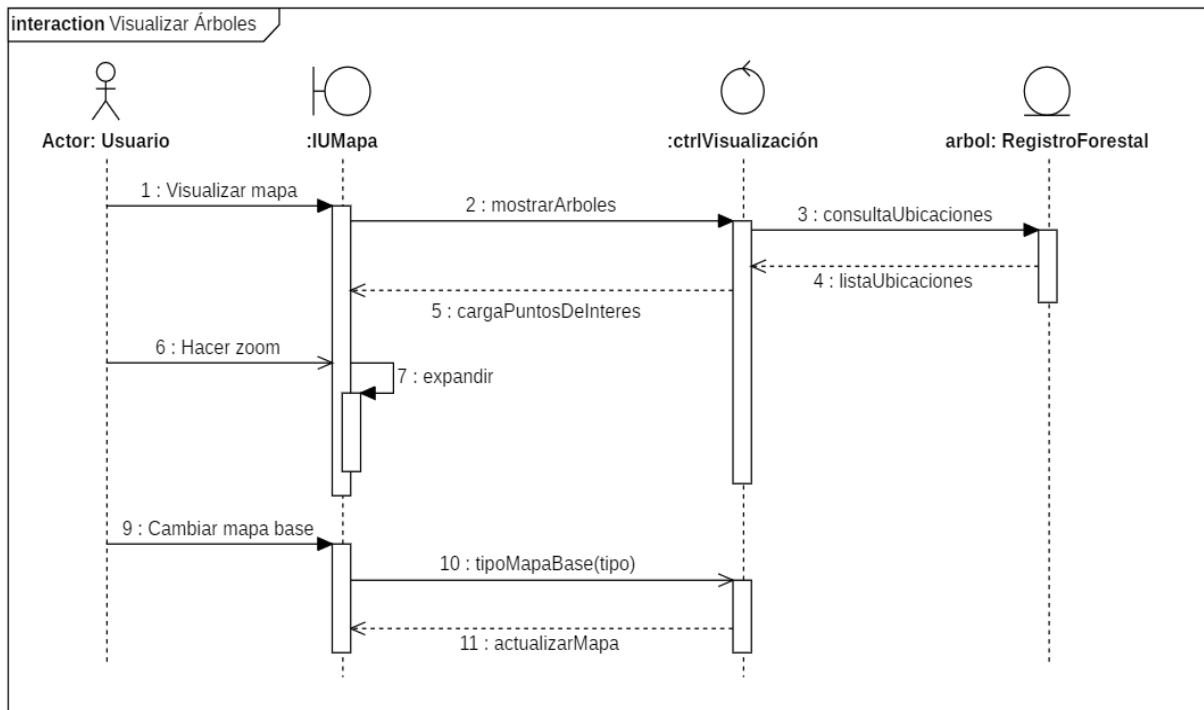


Figura 41. Diagrama de secuencia para el caso de uso Visualizar árboles.

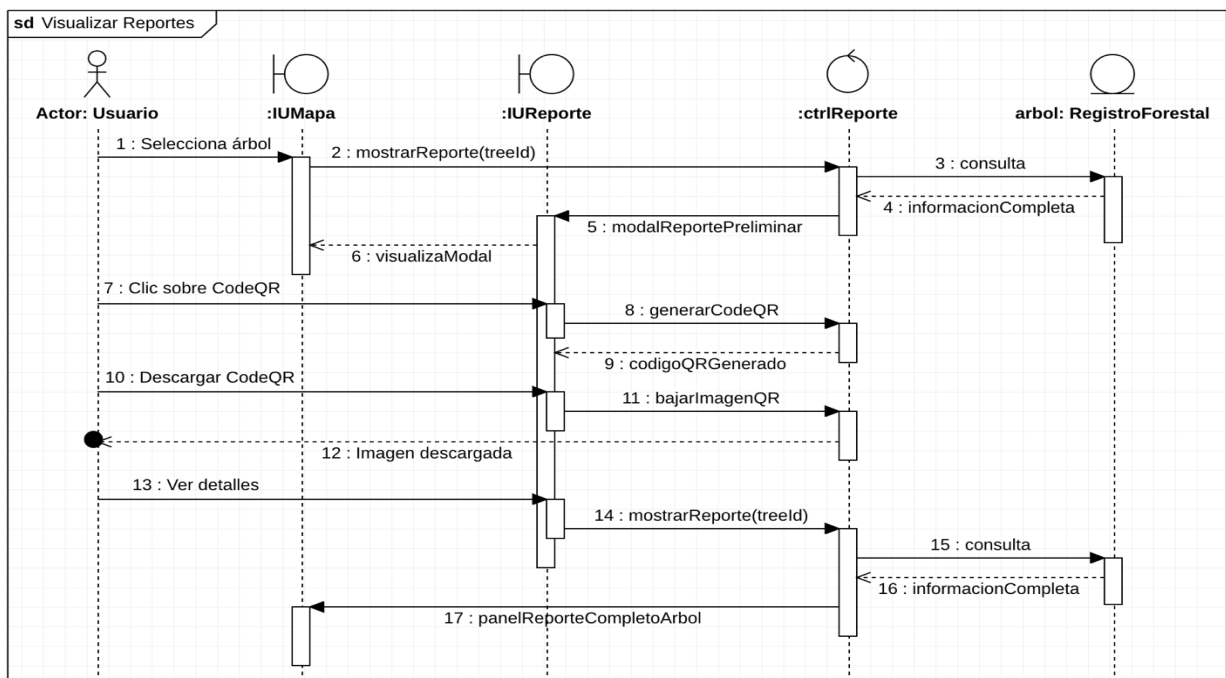


Figura 42. Diagrama de secuencia para el caso de uso Visualizar reportes.

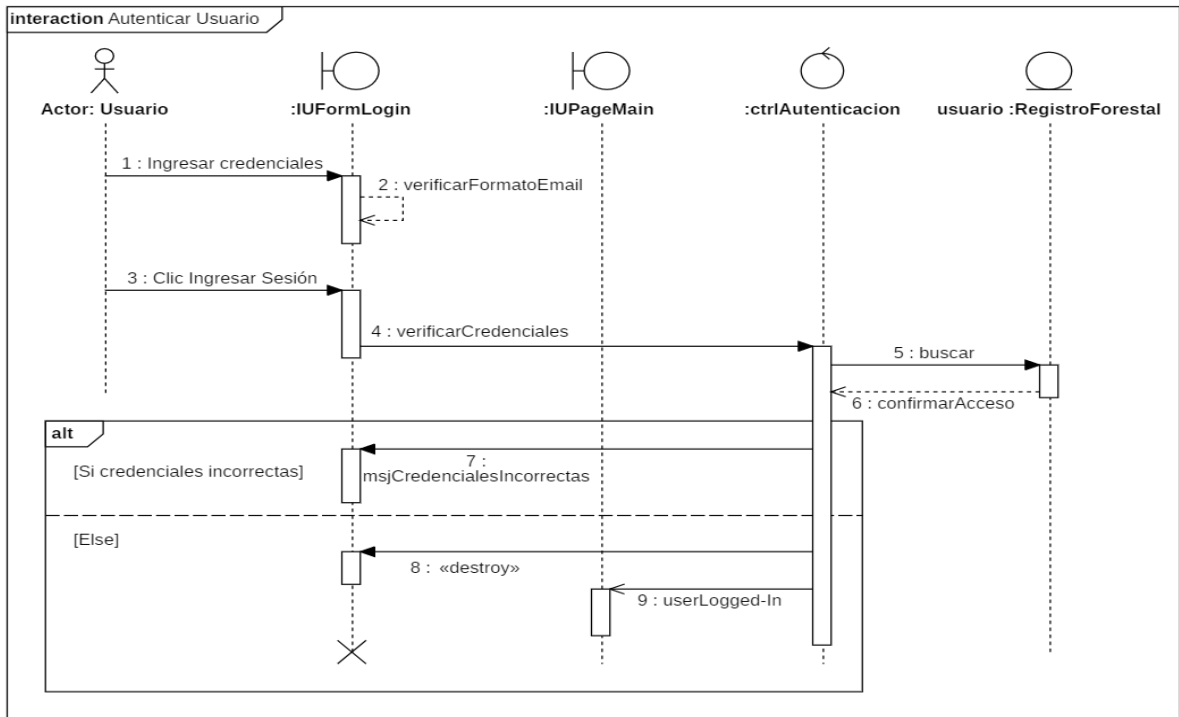


Figura 43. Diagrama de secuencia para el caso de uso Autenticar usuario.

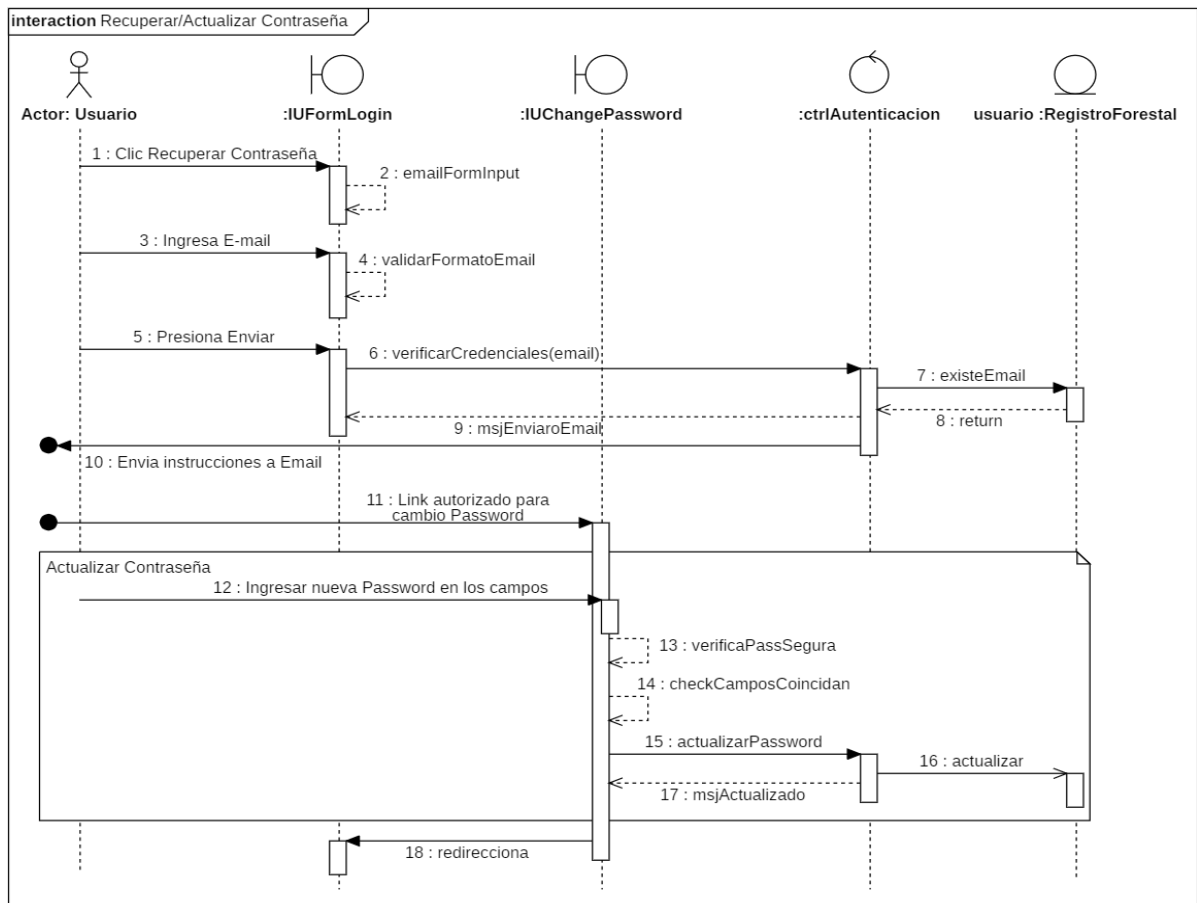


Figura 44. Diagrama de secuencia para el caso de uso Recuperar/Actualizar contraseña.

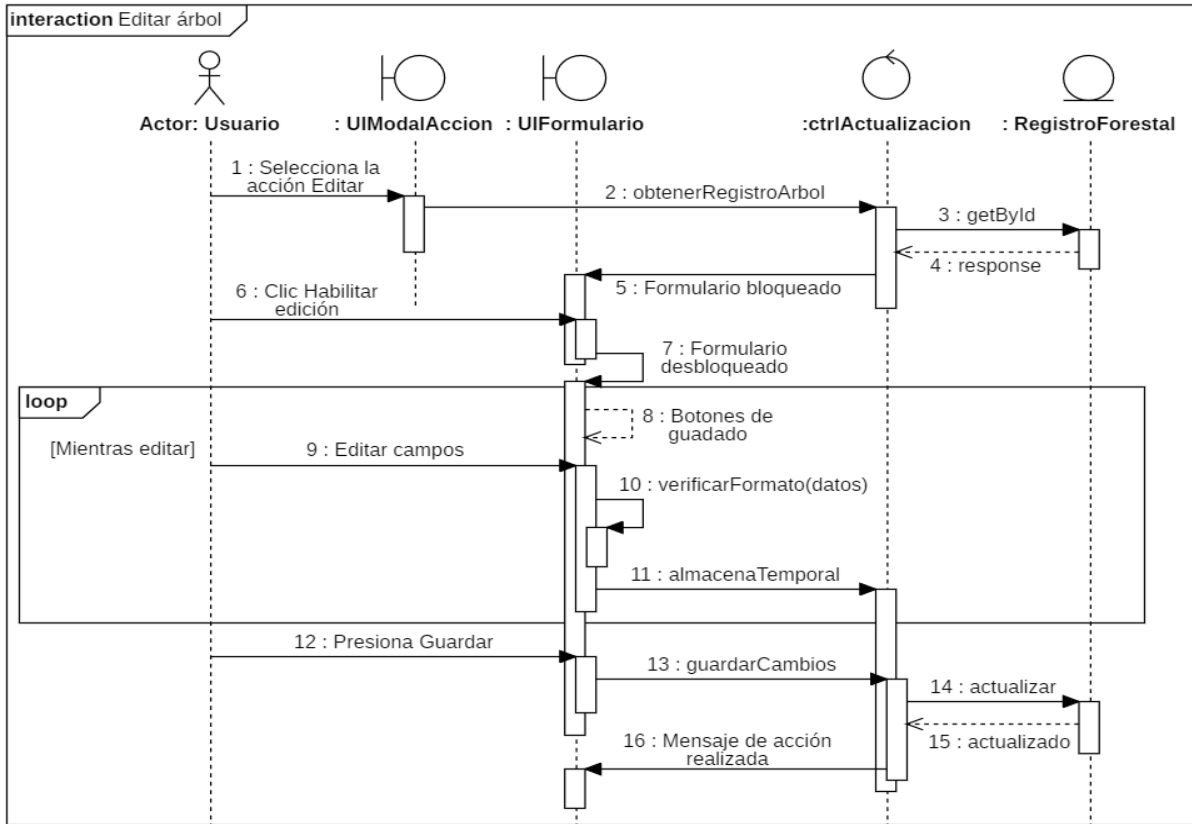


Figura 45. Diagrama de secuencia para el caso de uso Editar árbol.

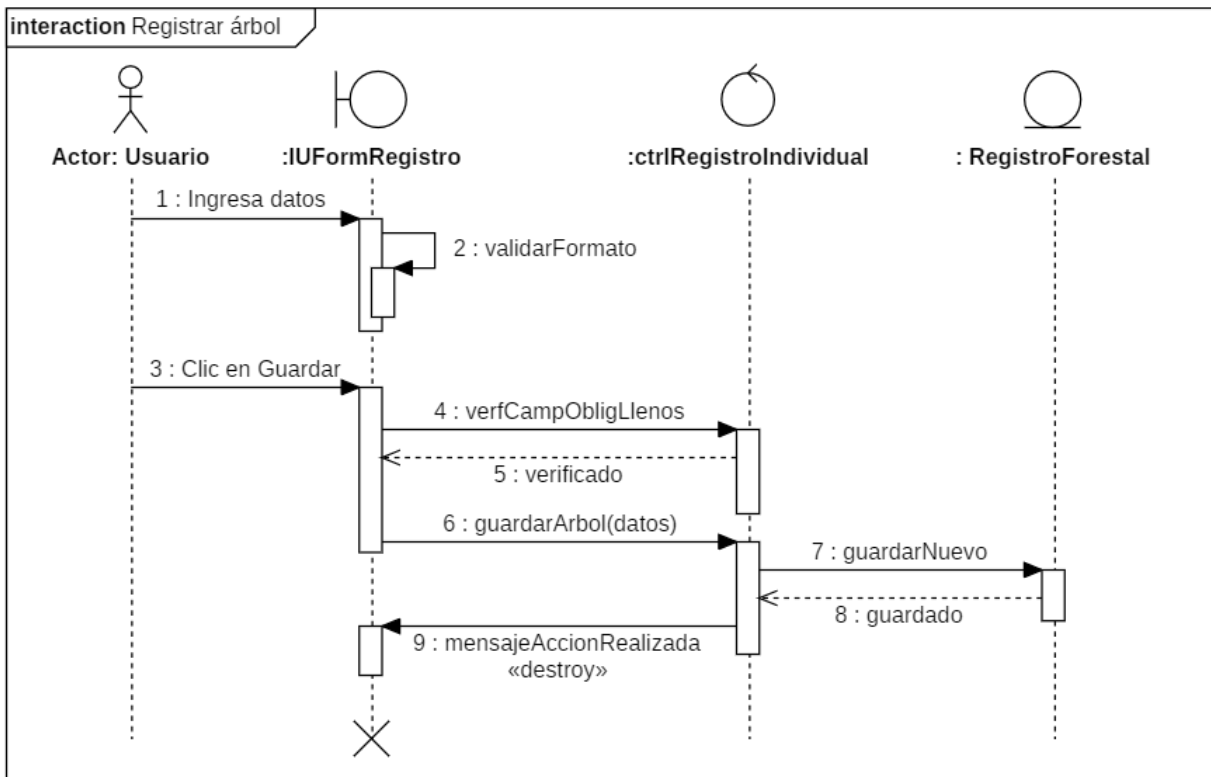


Figura 46. Diagrama de secuencia para el caso de uso Registrar árbol.

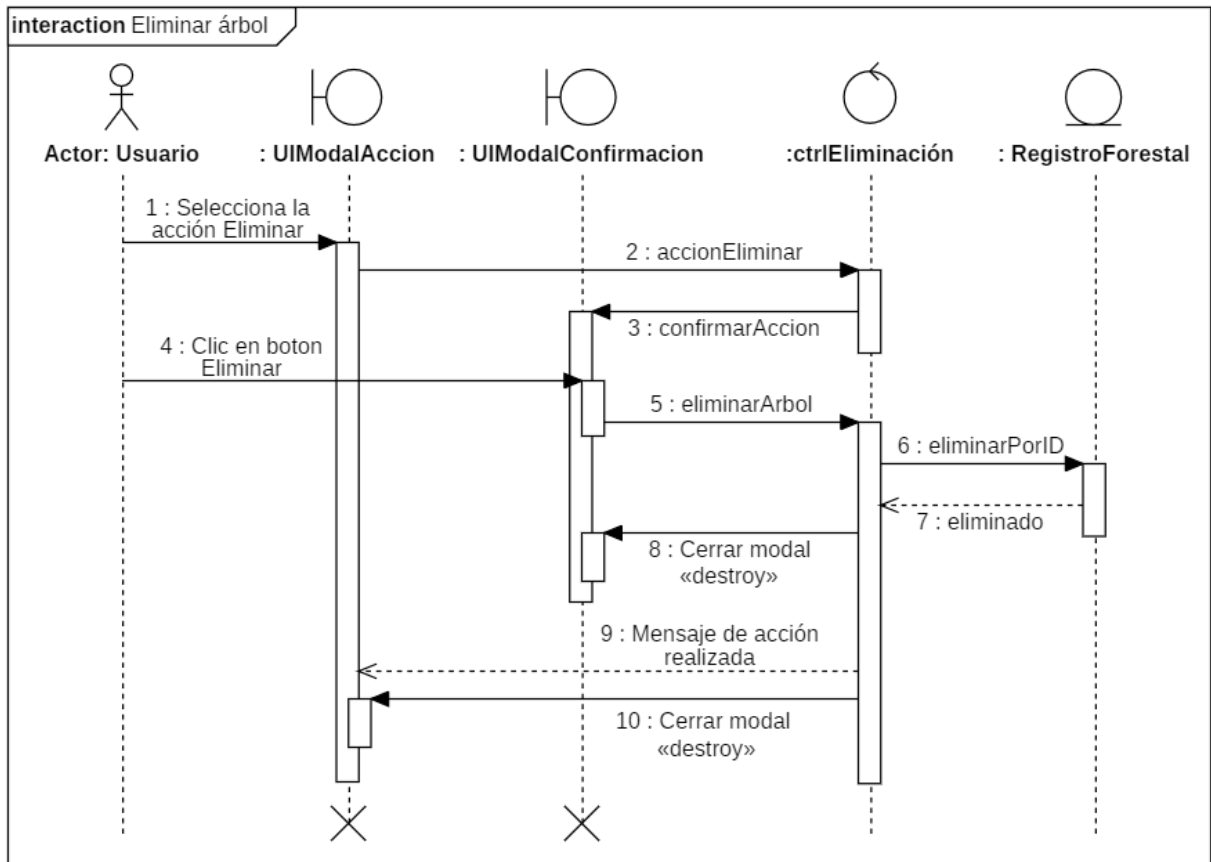


Figura 47. Diagrama de secuencia para el caso de uso Eliminar árbol.

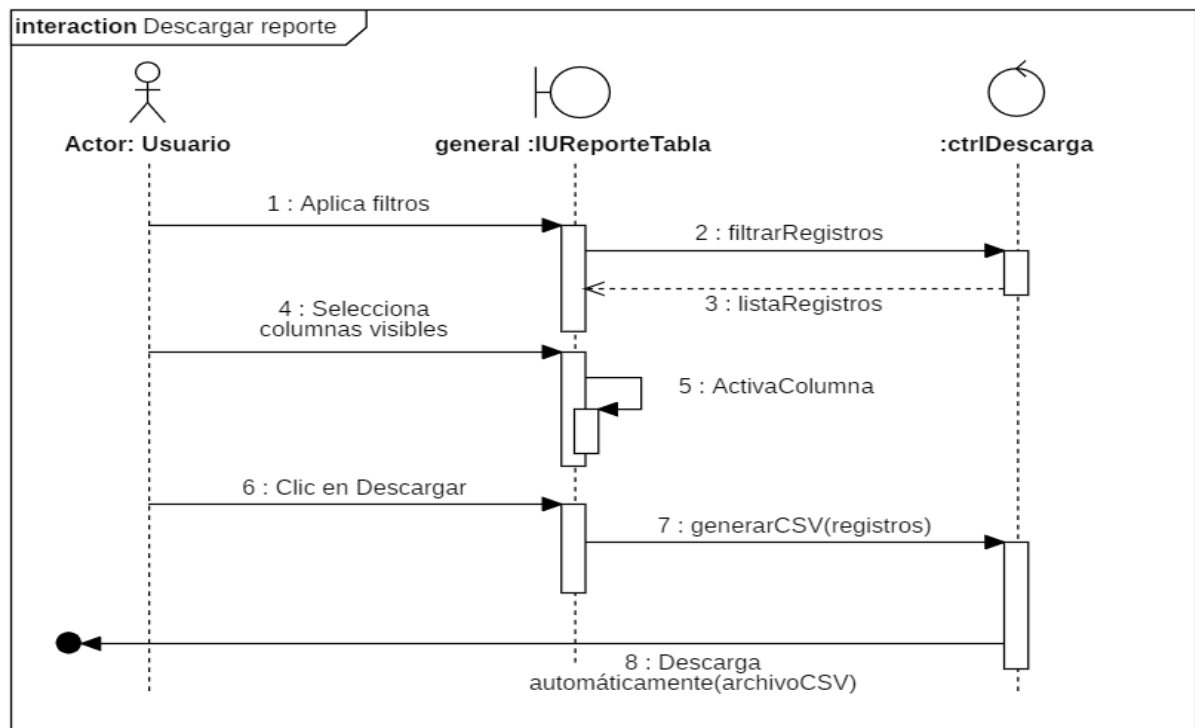


Figura 48. Diagrama de secuencia para el caso de uso Descargar reporte.

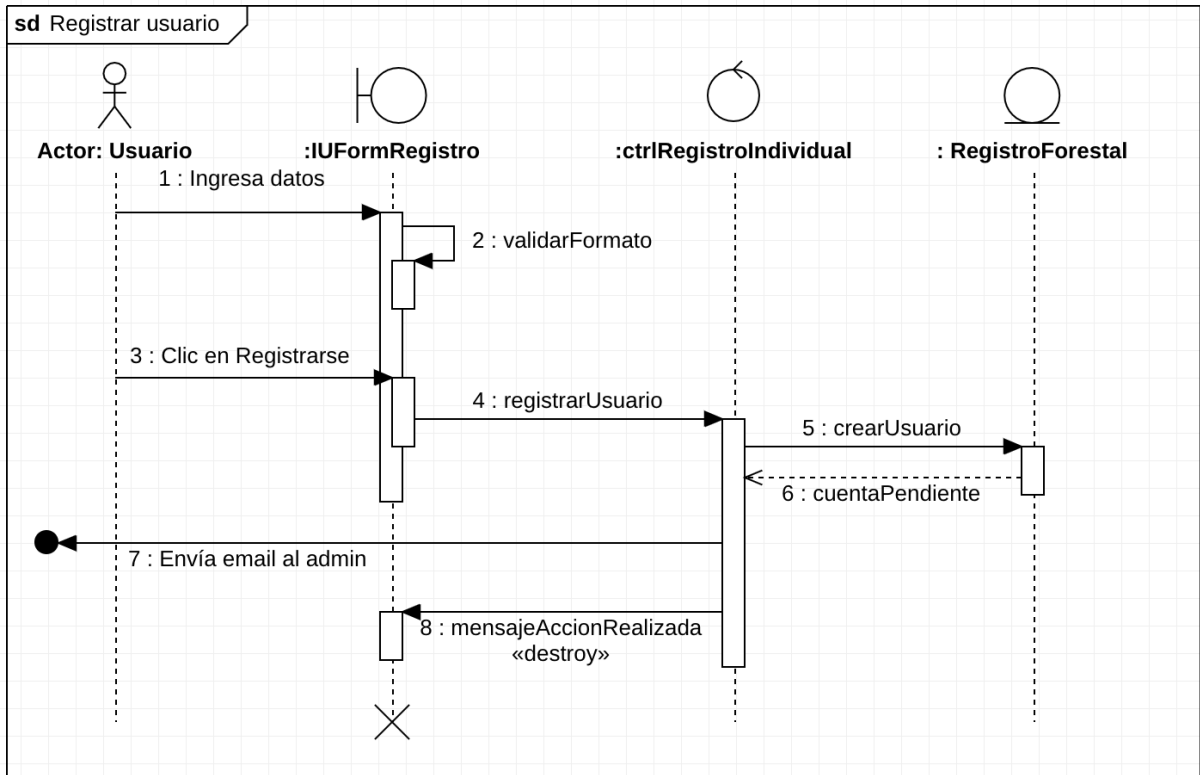


Figura 49. Diagrama de secuencia para el caso de uso Registrar usuario.

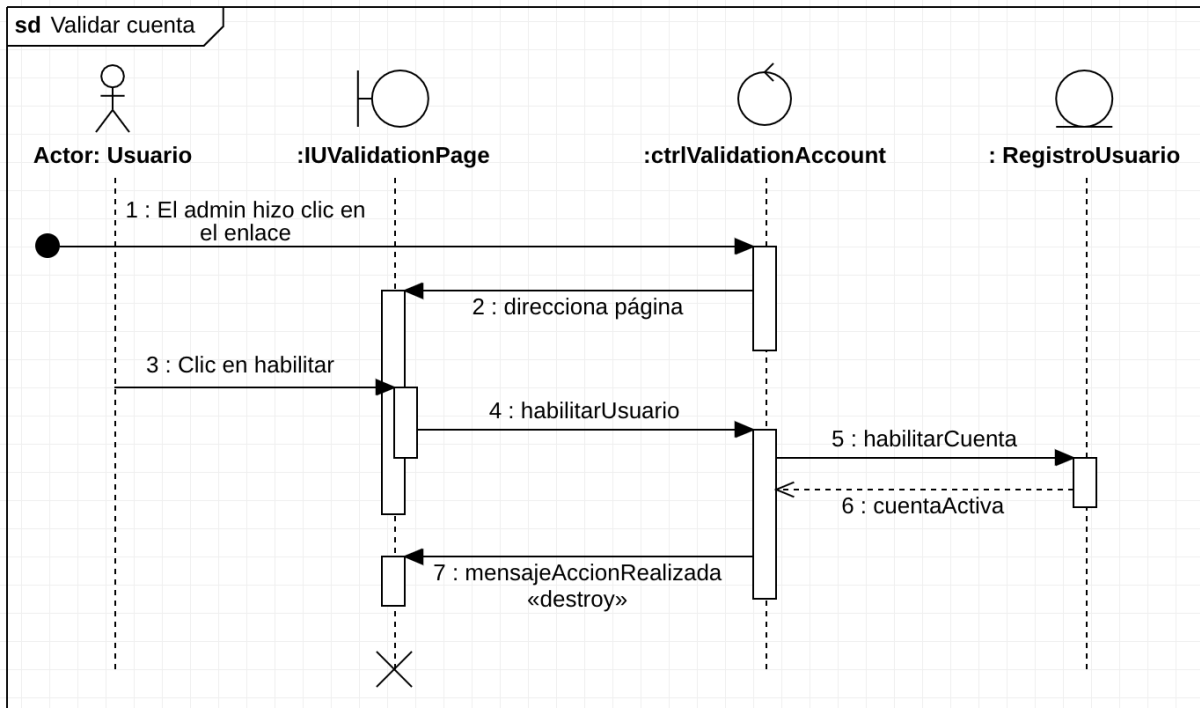


Figura 50. Diagrama de secuencia para el caso de uso Validar Cuenta usuario.

4.2. Diagramas de clases

En la Figura 51 se muestra el diagrama de clases según el modelo de base de datos del aplicativo móvil ArbolQR. Este modelo inicial se analizó la estructura que para posteriormente adaptar los nuevos requerimientos al aplicativo web.

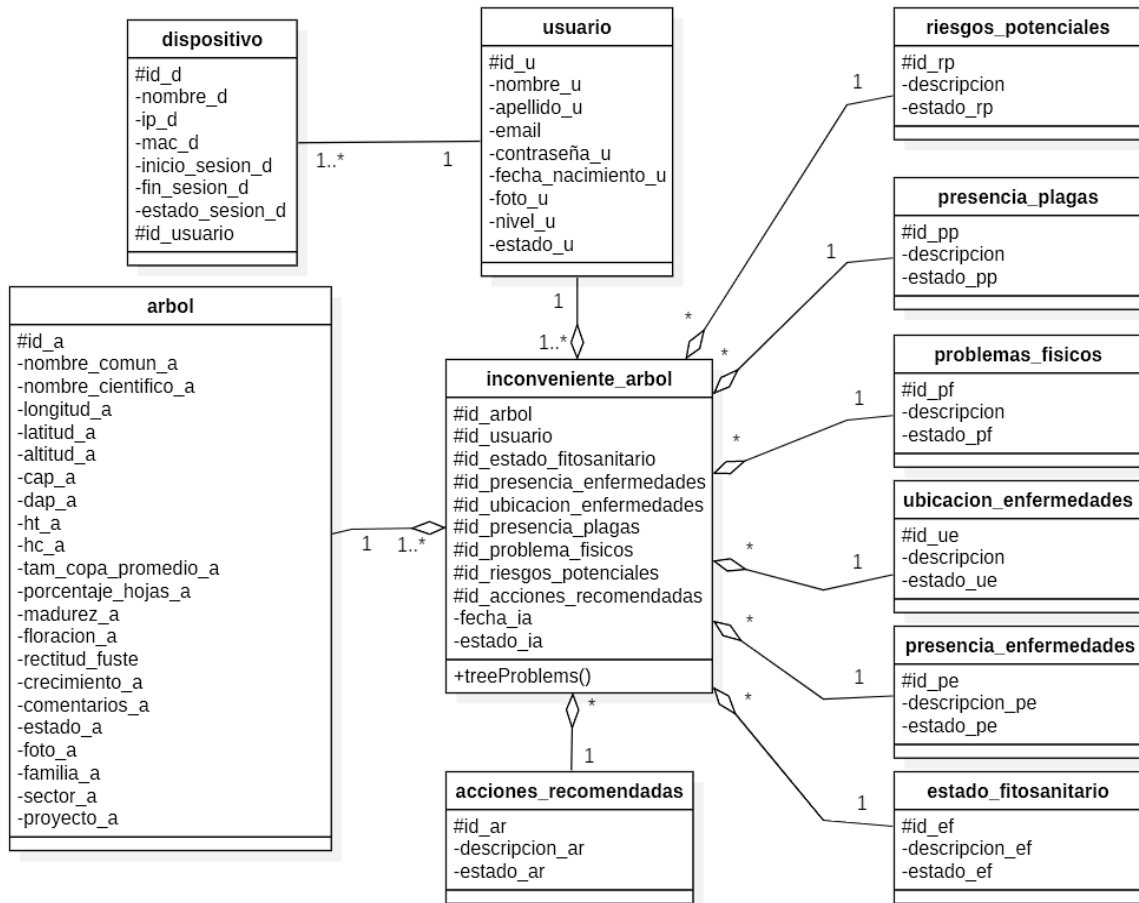


Figura 51. Diagrama de clase inicial basado en el aplicativo móvil.

De acuerdo a las nuevas características del proyecto y los requerimientos funcionales previamente levantados en **Tabla 3**, al modelo base se rediseñó, previa socialización con el director del Proyecto Forestal, para mejorar la parte administrativa. Dichos cambios se presentan en la **Figura 52** que muestra la organización realizada después del análisis.

4.3. Diagramas de datos

En la **Figura 53** se muestra el diagrama de datos diseñado para el soporte de la base de datos en MySQL.

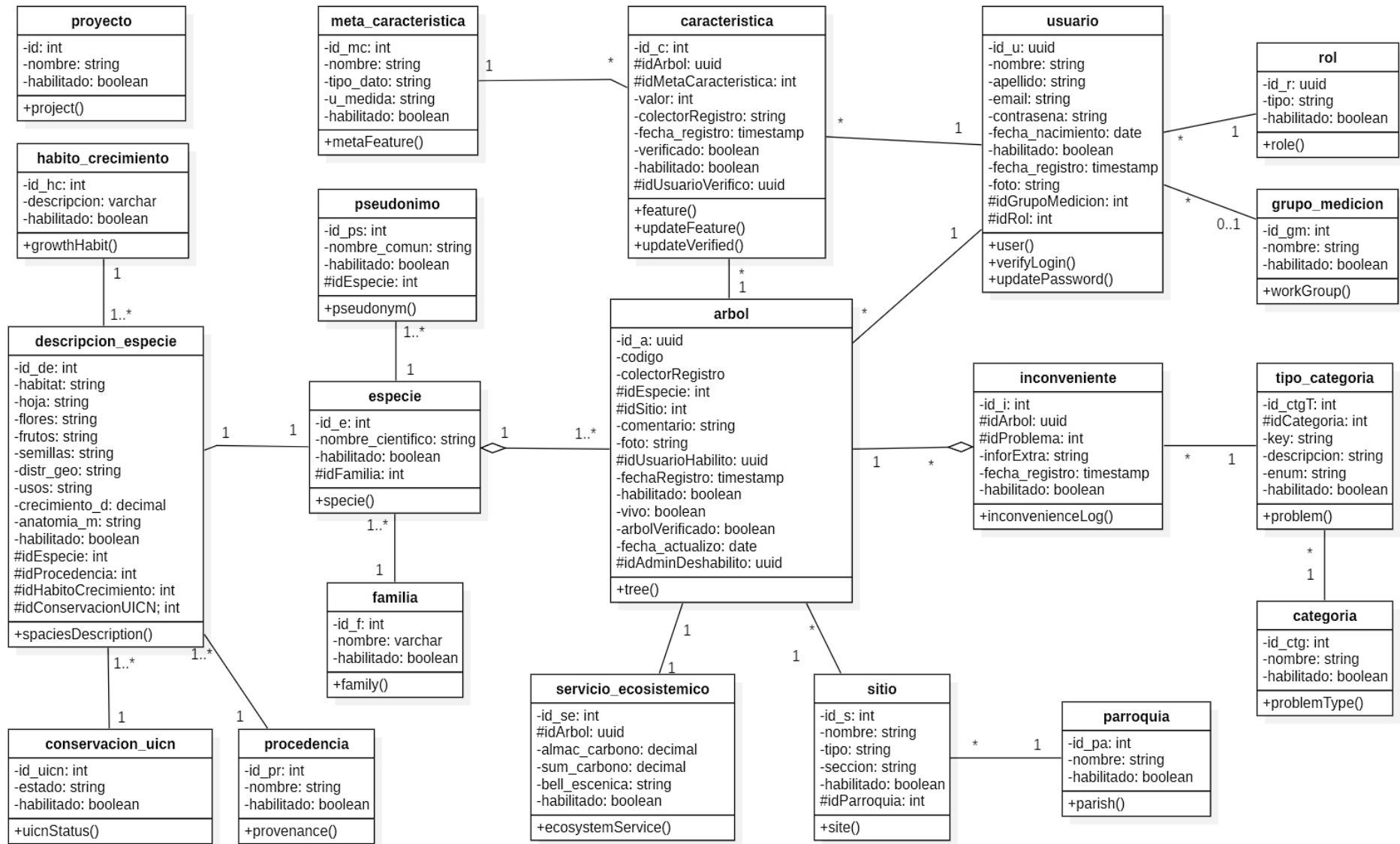


Figura 52. Diagrama de clase adapta a los requerimientos del aplicativo web.

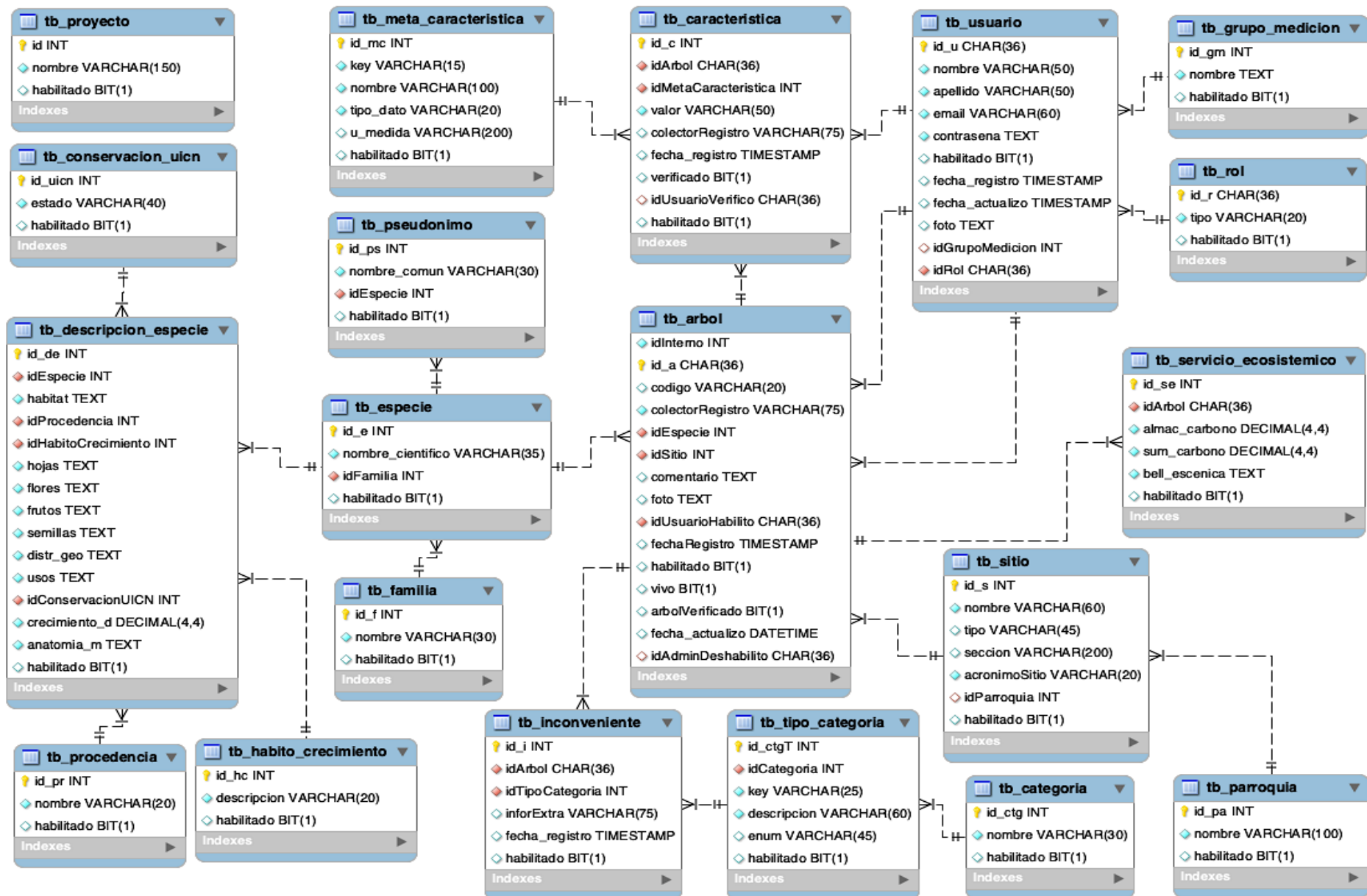


Figura 53. Diagrama de datos del aplicativo web obtenidos del MySQL Workbench.

5. FASE 4: Implementación

Esta fase comprende el desarrollo del aplicativo web en base a los modelos establecidos anteriormente.

5.1. Codificación de la aplicación web

Se realizó el levantamiento del entorno de desarrollo para la codificación, consultas a base de datos y pruebas al servidor utilizando las siguientes herramientas:

- Visual Studio Code v1.74.3
- MySQL v8.0.29
- Postman v10.8.7

La codificación se realizó bajo el entorno de ejecución multiplataforma NodeJS, donde se instaló el framework para backend ExpressJS, y para el frontend ReactJS. Se utilizaron las siguientes versiones para la instalación de librerías:

- NPM v8.11.0
- NodeJS v16.16.0

El entorno de desarrollo se construyó bajo el modelo de arquitectura cliente-servidor, con la división del backend (ver **Figura 64**) y del frontend (ver **Figura 65**) respectivamente. Se utilizó el patrón de diseño Modelo Vista Controlador (MVC) donde los modelos y controladores trabajan en el Servidor y la vista en lado del Cliente.

El código fuente se respaldó en GitHub (Repositorio Privado) (ver **Figura 61**, **Figura 62**), los enlaces a estos repositorios los puede encontrar a continuación:

- **webGISTree-backend**
- **webGISTree-frontend**
- **arbol_QR_db**

5.1.1. Codificación en el backend

La estructura del directorio en el backend se muestra en la **Figura 64** en ella se observa los siguientes archivos y carpetas:

- **src**, es la carpeta principal que contendrá la aplicación (app.js):
 - **apiServices**, contine los servicios que ofrece la API, agrupadas por sus nombres. En ellas se encuentran el modelo (*model.js*) para el mapeo de la estructura en la base de datos MySQL, las rutas (*routes.js*) que contienen el detalle de las peticiones específicas a hacia ese recurso, y los controladores que maneja dicha petición (ver **Figura 54**).

- **config**, configuraciones de variables de entorno, email, mysql.
 - **middleware**, funciones comunes para todos los servicio.
 - **routes**, configuración de las rutas principales hacia los servicios de la api.
 - **server**, carpeta principal para la ejecución de los servicios.
 - **templates**, plantilla para autorizar usuarios y recuperar contraseña.
 - **utils**, funciones de encriptación, generación de tokens y formateo de fatos.
- **.env**, son las configuraciones de variables de entorno que el proyecto necesita y que no se encuentran en el repositorio GitHub debido al archivo **.gitignore**
 - **.gitignore**, describen los archivos que serán ignorados por comando git al subir el repositorio.
 - **package.json**, es el archivo que contine los paquetes instalados (ver **Figura 64**).

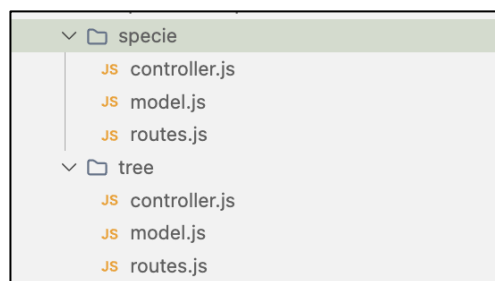


Figura 54. Estructura interna de los servicios en apiServices.

***routes**: Las rutas determinan los servicios API REST por donde se accederán a los diferentes recursos. Su estructura se define como se muestra en la **Figura 55**:

```

src > apiServices > user > JS routes.js > ...
 1 import { Router } from 'express'
 2 import { methods as controller } from './controller'
 3 import { authorization, isAdmin, authorization_alt } from '../middleware/authorization'
 4 import { validateInfo } from '../middleware/validateInfo'
 5 const routes = Router()
 6
 7 routes.get('/me', authorization, controller.me)
 8 routes.get('/', authorization, controller.getUserAll)
 9 routes.get('/:id', authorization, controller.getUserById)
10 routes.post('/register-user', validateInfo, controller.createUser)
11 routes.put('/update-account', authorization, controller.updateUserAccount)
12 routes.put('/update-password', authorization_alt, controller.updatePassword)
13 routes.put('/:userId', authorization, controller.updateUserById)
14 routes.put('/disable/:id', authorization, isAdmin, controller.disableUserById)
15 routes.put('/enable/:id', authorization, isAdmin, controller.enableUserById)
16 export default routes

```

Figura 55. Definición de las rutas para el servicio usuario

***models**: Para los modelos se utilizó el ORM Sequelize que permite el mapeo relacional de objetos, crear la base de datos MySQL y sus relaciones, su contenido es similar a la **Figura 56**:

```

JS model.js
src > apiServices > specie > JS model.js > ...
1  import { DataTypes, Model } from 'sequelize'
2  import { sequelize } from '../config/mysql'
3
4  class SpecieModel extends Model {
5    static associate(models) {
6      SpecieModel.belongsTo(models.FamiliaModel, { targetKey: 'idFamilia' })
7      SpecieModel.hasMany(models.PseudonymModel, { foreignKey: 'id_e' })
8      SpecieModel.hasMany(models.TreeModel, { foreignKey: 'id_e' })
9    }
10 }
11 SpecieModel.init(
12   {
13     id_e: {
14       type: DataTypes.INTEGER, autoIncrement: true, primaryKey: true,
15     },
16     nombre_cientifico: {
17       type: DataTypes.STRING(35), allowNull: false, unique: true,
18     },
19     idFamilia: {
20       type: DataTypes.INTEGER, allowNull: false, references: {
21         model: 'tb_familia',
22         key: 'id_f',
23       },
24     },
25     habilitado: { type: 'BIT', defaultValue: true },
26   },
27   { sequelize, timestamps: false, modelName: 'tb_especie' }
28 )
29
30 export default SpecieModel

```

Figura 56. Estructura de un modelo con Sequelize ORM.

***controller**: Los controladores manejan los diferentes aspectos de la petición, uno de ellos se muestra en la **Figura 57**:

```

175 const updateSpecie = (req, res) => {
176   const { especieId } = req.params
177   const { rol } = req.payload
178   const { nombre_cientifico, idFamilia, habilitado } = req.body
179
180   SpecieModel.update(
181     {
182       nombre_cientifico,
183       idFamilia,
184       habilitado,
185     },
186     {
187       where: { id_e: especieId },
188     }
189   )
190   .then(specie => {
191     if (!Boolean(specie[0])) return res.status(200).json({ update: false, res: specie })
192     res.status(200).json({ update: true, res: specie })
193   })
194   .catch(err => {
195     console.error(err.message)
196     res.status(500).json({ error: 'Server Error', message: err.message })
197   })
198 }

```

Figura 57. Estructura de un controlador para la actualización de datos.

También se destacan los siguientes fragmentos de códigos que fueron importantes al momento de realiza las peticiones hacia los diferentes servicios:

- El manejo de la autorización para permitir el acceso a las peticiones (ver Figura 58), e identificar si tiene el rol de administrador (ver **Figura 59**), estos declarados en el **middleware**.
- El método de generación de tokens, encriptación de contraseñas y formateo de valores ingresados definidos en **utils** (ver **Figura 60**).

```
JS authorization.js X
src > middleware > JS authorization.js > isAdmin
1  import 'dotenv/config'
2  import jwt from 'jsonwebtoken'
3  import UserModel from '../apiServices/user/model'
4  import RoleModel from '../apiServices/role/model'
5
6  // Authorization: Bearer <token>
7  export const authorization = (req, res, next) => {
8    try {
9      const accessToken = req.headers['authorization']
10
11     if (!accessToken)
12       return res.status(401).json({
13         auth: false,
14         message: 'Acceso no autorizado. Proporcione JWT.',
15       })
16
17     req.token = accessToken.split(' ')[1]
18     jwt.verify(req.token, process.env.JWT_SECRET, (err, payload) => {
19       if (err)
20         return res.status(403).json({
21           auth: false,
22           message: 'Acceso no autorizado. Token incorrecto o vencido.',
23         })
24       req.payload = payload.user // { id: user.id_u, rol: ['administrador'] }
25       next()
26     })
27   } catch (error) {
28     return res.status(500).json({ Error: error })
29   }
30 }
```

Figura 58. Middleware de autorización de token.

```

JS authorization.js ×
src > middleware > JS authorization.js > isAdmin
71
72 export const isAdmin = async (req, res, next) => {
73   try {
74     const user = await UserModel.findById(req.payload.id)
75     if (!user)
76       return res.status(403).json({
77         message: 'El idUsuario no existe.',
78       })
79
80     const role = await RoleModel.findById(user.idRol)
81
82     if (role.tipo !== 'administrador')
83       return res.status(403).json({
84         isAdmin: false,
85         message: 'Acceso no autorizado. Requiere rol Admin',
86       })
87     next()
88   } catch (error) {
89     return res.status(500).json({ Error: error })
90   }
91 }
92

```

Figura 59. Middleware de autorización para identificar usuario administrador.

```

src > utils > JS index.js > ...
1  import jwt from 'jsonwebtoken'
2  import bcrypt from 'bcryptjs'
3  import _ from 'lodash'
4
5  const saltRound = 10
6
7  export const generateAccessToken = ({
8    user,
9    secretKey = process.env.JWT_SECRET,
10   expiresIn = process.env.JWT_EXPIRES_IN,
11 }) => jwt.sign({ user }, secretKey, { expiresIn })
12
13 export const encryptPassword = async password => {
14   const salt = await bcrypt.genSalt(saltRound)
15   return bcrypt.hash(password, salt)
16 }
17
18 export const formatAndGetUniqueList = (data, field) => {
19   const uniqueList = _.uniqBy(data, obj => {
20     let valueTrim = _.trim(obj[`${field}`])
21     obj[`${field}`] = valueTrim
22     return valueTrim
23   })
24   let emptyFields = _.map(uniqueList, field).filter(value => !value)
25   if (emptyFields.length) return null
26   return uniqueList
27 }

```

Figura 60. Métodos más utilizados del archivo utils.

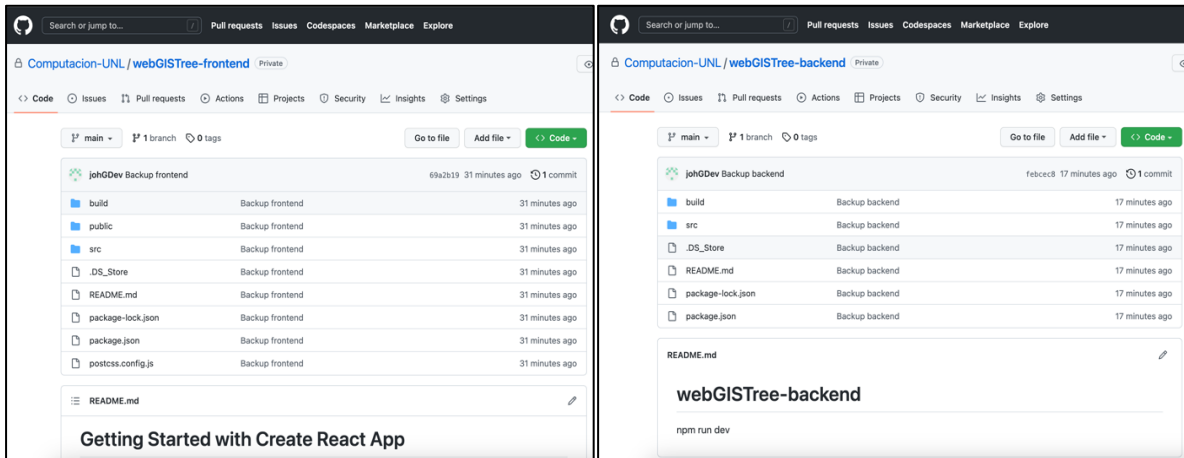


Figura 61. Repositorio GitHub de frontend y backend.

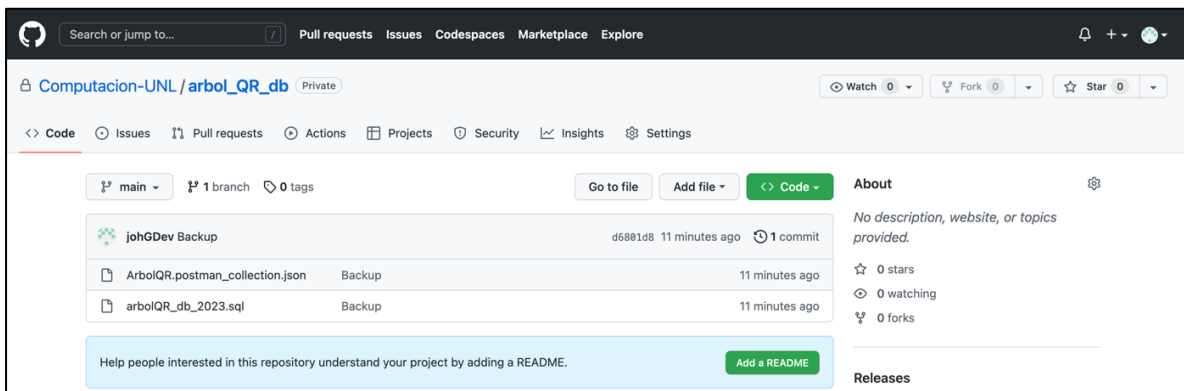


Figura 62. Repositorio GitHub de base de datos.

5.1.2. Codificación en el frontend

La estructura del directorio en el frontend se muestra en la **Figura 65**, contiene los siguientes elementos principales: elementos visuales envueltos en la carpeta *components*,

- **src**, es la carpeta principal que contiene la aplicación (App.js), dentro se encuentra:
 - **components**, representan pequeños grupos de elementos visuales.
 - **config**, declaraciones de tipos de rutas, configuración para generar CódigoQR.
 - **container**, representa las páginas, reutiliza los componentes visuales previamente construidos.
 - **PrivateRoutes**, es un componente configurado para el direccionamiento a rutas privadas.
 - **redux**, contiene la declaración de los estados globales de la aplicación.
 - **services**, maneja las funciones para la comunicación con el backend.
 - **theme**, es la configuración global de colores usados en la aplicación.

- **accions**, maneja declaraciones de variables de entorno globales.
- **utils**, contiene funciones que comparten algunos componentes.
- **.env**, son las configuraciones de variables de entorno que el proyecto necesita y que no se encuentran en el repositorio GitHub debido al archivo **.gitignore**
- **.gitignore**, describen los archivos que serán ignorados por comando git al subir el repositorio.
- **package.json**, es el archivo que contine los paquetes instalados (ver Figura 65)

Es importante destacar que los campos de ingreso de texto en los formularios contienen validaciones de entrada, esto se realiza mediante el uso de la librería Yup y Formik y el esquema de validación se muestra en la **Figura 63**.

```

src > utils > JS schemaValidation.js > ...
1  import * as Yup from 'yup'
2
3  const EMAIL = {
4    email: Yup.string().email('Email Inválido.').required('Obligatorio'),
5  }
6
7  const USERNAME = {
8    username: Yup.string()
9      .strict(true)
10     .trim('No inicie o termine con espacios en blanco.')
11     .min(3, 'Nombre demasiado corto, digite más de 3 caracteres.')
12     .max(50, 'Nombre demasiado largo, máx 50 caracteres.')
13     .required('Obligatorio'),
14  }
15
16  const LASTNAME = {
17    lastname: Yup.string()
18      .strict(true)
19      .trim('No inicie o termine con espacios en blanco.')
20      .min(3, 'Apellido demasiado corto, digite más de 3 caracteres.')
21      .max(50, 'Apellido demasiado largo, max. 50 caracteres.')
22      .required('Obligatorio'),
23  }
24
25  const PASSWORD_LOGIN = {
26    password_login: Yup.string()
27      .strict(true)
28      .trim('Evite espacios en blanco.')
29      .required('Obligatorio'),
30  }

```

Figura 63. Esquema de validación de entradas con Yup.

```

1  {
2    "name": "webgistree-backend",
3    "version": "1.0.0",
4    "description": "ArbolQR App WEB",
5    "main": "index.js",
6    "scripts": {
7      "babel-node": "babel-node --presets=@babel/preset-env",
8      "dev": "nodemon --exec npm run babel-node src/server/index.js",
9      "build": "babel src -d build",
10     "start": "npm run babel-node src/server/index.js"
11   },
12   "keywords": [],
13   "author": "Johnny González",
14   "license": "ISC",
15   "dependencies": {
16     "bcryptjs": "^2.4.3",
17     "consola": "^2.15.3",
18     "cors": "^2.8.5",
19     "dotenv": "^16.0.1",
20     "express": "^4.18.1",
21     "http-errors": "^2.0.0",
22     "jsonwebtoken": "^8.5.1",
23     "lodash": "^4.17.21",
24     "mysql2": "^2.3.3",
25     "nodemailer": "^6.8.0",
26     "sequelize": "^6.21.2",
27     "sequelize-cli": "^6.4.1"
28   },
29   "devDependencies": {
30     "@babel/cli": "^7.18.6",
31     "@babel/core": "^7.18.6",
32     "@babel/node": "^7.18.6",
33     "@babel/preset-env": "^7.18.6",
34     "morgan": "^1.10.0",
35     "nodemon": "^2.0.19"
36   }
37 }
38

```

Figura 64. Estructura de carpetas y paquetes del backend.

```

4  "private": true,
5  "dependencies": {
6    "@emotion/react": "^11.10.5",
7    "@emotion/styled": "^11.10.5",
8    "@mui/icons-material": "^5.11.0",
9    "@mui/material": "^5.11.1",
10   "@mui/styled-engine-sc": "^5.8.0",
11   "@mui/x-data-grid": "^5.17.17",
12   "@mui/x-date-pickers": "^5.0.6",
13   "@reduxjs/toolkit": "^1.8.3",
14   "@testing-library/jest-dom": "^5.16.4",
15   "@testing-library/react": "^13.3.0",
16   "@testing-library/user-event": "^14.3.0",
17   "axios": "^0.27.2",
18   "date-fns": "^2.29.3",
19   "formik": "^2.2.9",
20   "lodash": "^4.17.21",
21   "mapbox-gl": "^2.9.2",
22   "moment": "^2.29.4",
23   "mui-datatables": "^4.2.2",
24   "qr-code-styling": "^1.6.0-rc.1",
25   "react": "^18.2.0",
26   "react-detect-offline": "^2.4.5",
27   "react-dom": "^18.2.0",
28   "react-redux": "^8.0.2",
29   "react-router-dom": "^6.3.0",
30   "react-scripts": "^5.0.1",
31   "styled-components": "^5.3.5",
32   "web-vitals": "^2.1.4",
33   "xlsx": "^0.18.5",
34   "yup": "^0.32.11"
35 },
36 "scripts": {
37   "start": "react-scripts start",
38   "build": "react-scripts build",
39   "test": "react-scripts test",
40   "eject": "react-scripts eject"
41 },

```

Figura 65. Estructura de carpetas y paquetes del frontend.

5.1.3. Diseño final de Interfaces de Usuario de la Aplicación web

En esta sección se muestran las interfaces de usuario de acuerdo al nivel de acceso que tienes en la aplicación web.

5.1.3.1. Usuario común

**Pantalla principal.*

En esta interfaz se hace énfasis a la visualización de los puntos de ubicaciones de árboles en un mapa cuyo requerimiento funcional es **RF01**, y tienen acceso todos los usuarios.

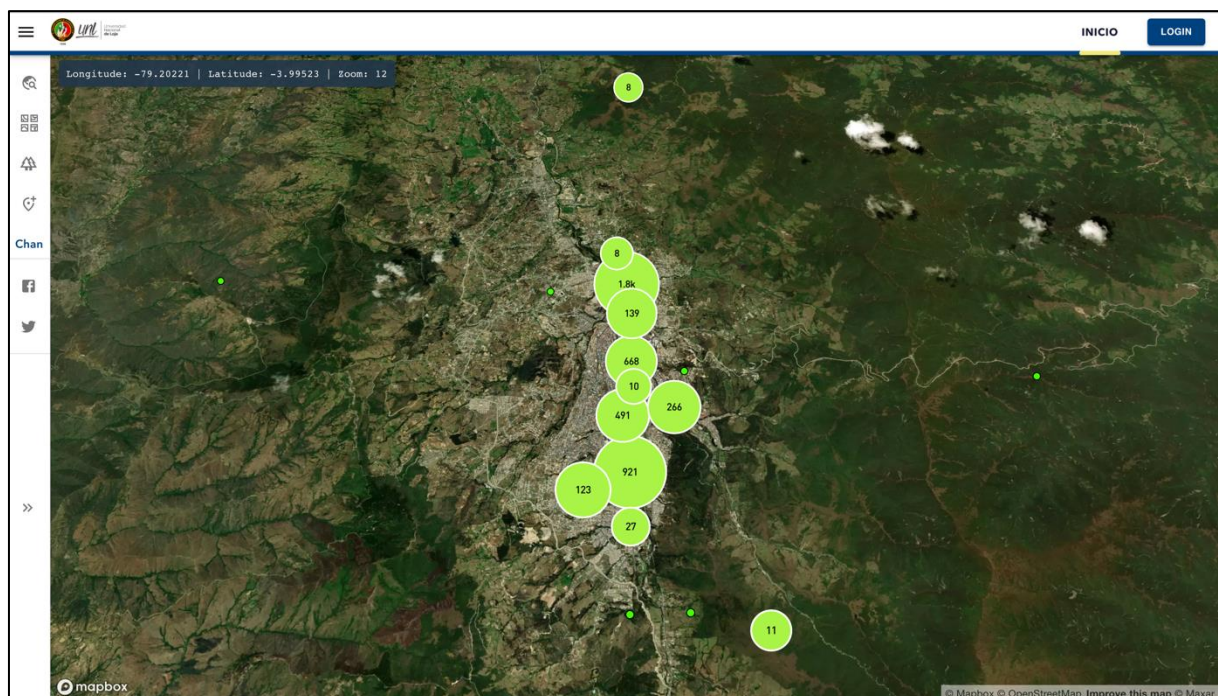


Figura 66. Pantalla principal de la aplicación web.

En el menú lateral izquierdo, puede realizarse filtros (**RF04**) de árboles por sitio (ver **Figura 67**), características (ver **Figura 68**) y por especies (ver **Figura 69**), donde al presionar aplicar se actualiza los puntos de árboles en el mapa y muestra en mensaje de la cantidad de árboles encontrados. Así mismos puede cambia a diferentes estilos de mapas base (**RF02**):

- *Imágenes*, por defecto se mostrará al iniciar la aplicación (ver **Figura 66**).
- *Claro, Oscuro, Calles, y Al aire libre*, son las alternativas disponibles (ver **Figura 70**).

Los puntos mostrados en los mapas aplican clústeres, estos agruparán de acuerdo al nivel de zoom aplicado sobre la capa base del mapa.

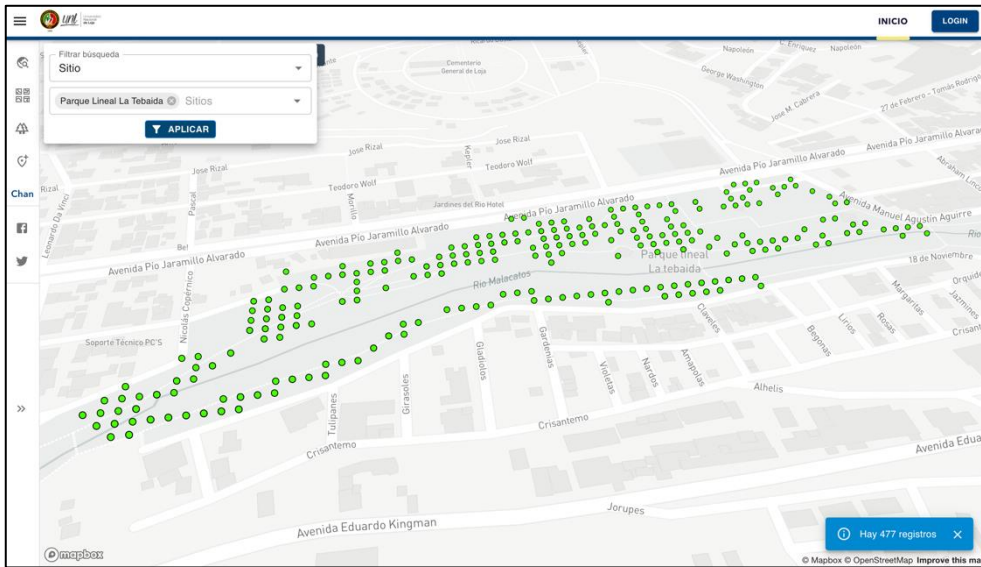


Figura 67. Aplicar filtros por lugar de ubicación.

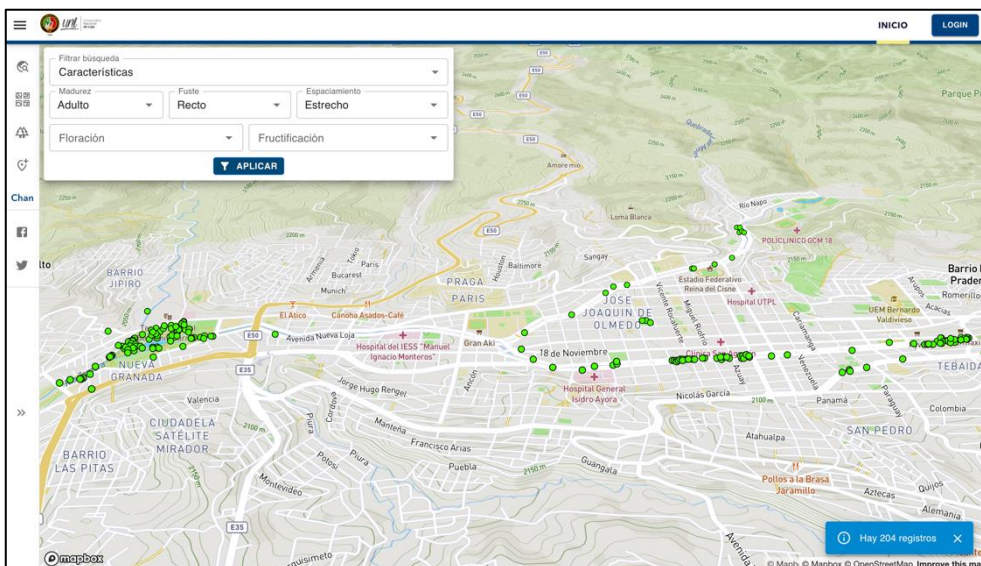


Figura 68. Aplicar filtros por características de árboles.

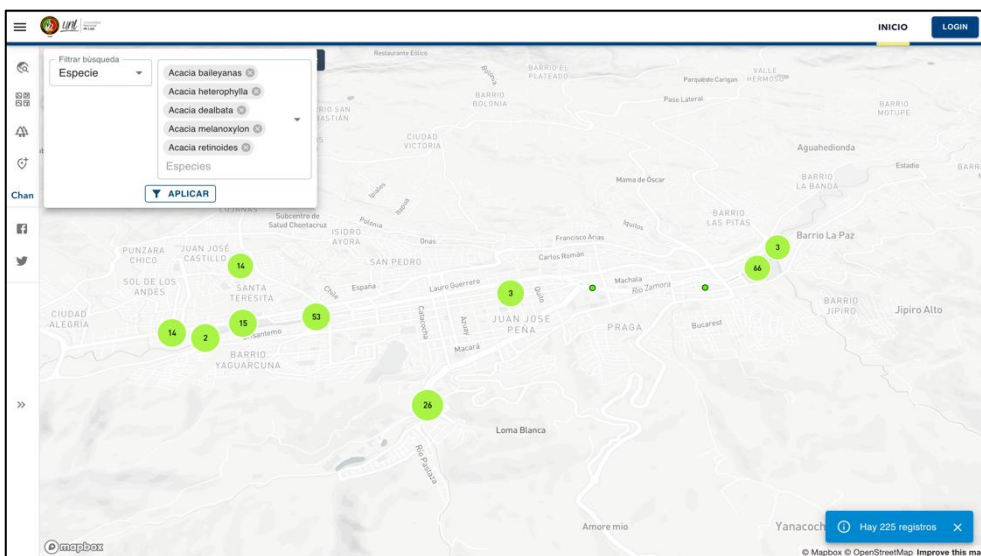


Figura 69. Aplicar filtros por nombre científico del árbol.

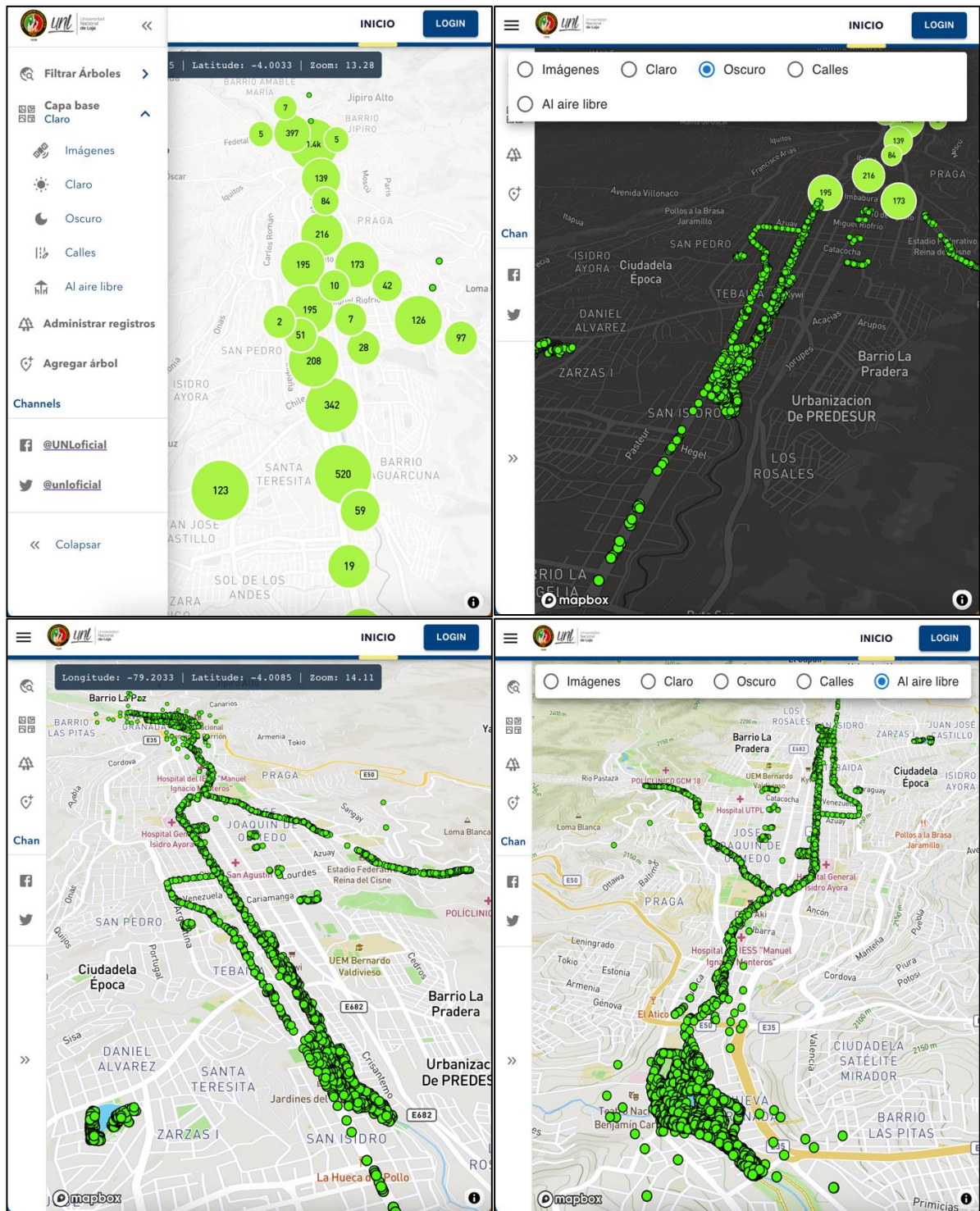


Figura 70. Alternativas de mapas base de la aplicación web.

Se puede mostrar información de los árboles al hacer clic sobre un punto verde en el mapa (**RF03**), y se desplegará un popup que contiene un reporte básico del árbol seleccionado, además de contener un código QR que podrá ver, leer o descargar en el formato png, jpeg o webp (ver **Figura 71**). Finalmente, podrá, a través del botón “Ver más”, direccionar hacia un reporte completo, que se mostrará en forma de panel, a la derecha del mapa (ver **Figura 72**).

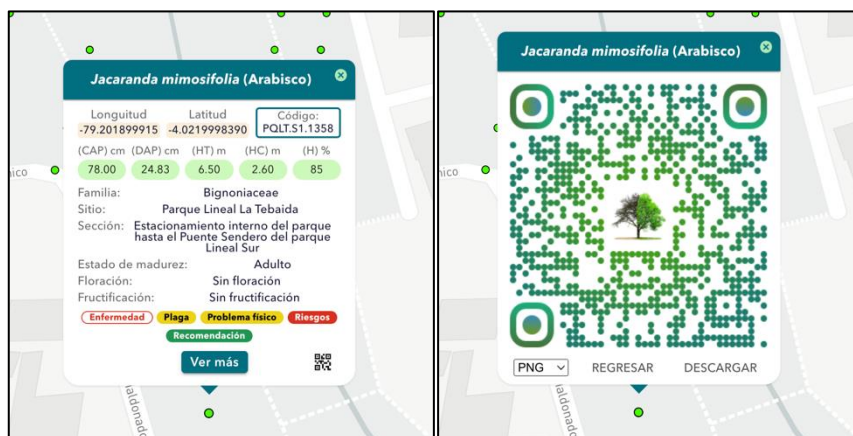


Figura 71. Reporte preliminar del árbol y observación de código QR

Jacaranda mimosifolia es llamado comúnmente Arabisco		UBICACIÓN DE LAS PLAGAS		Fructificación: Sin fructificación	
UBICACIÓN DEL ÁRBOL		Fuste		Rectitud de fuste: Torcido	
Sitio: Parque Lineal La Tebaida Sección: Estacionamiento interno del parque hasta el Puente Sendero del parque Lineal Sur Tipo de sitio: Parque Altitud: 2070 m.s.n.m. Longitud: -79.2019 Latitud: -4.0220		PROBLEMAS FÍSICOS		ESTADO FITOSANITARIO	
DESCRIPCIÓN GENERAL		RIESGOS POTENCIALES		Raíz: Buena Fuste: Regular Corteza: Regular Ramas: Regulares Hojas: Regulares Cima: Buena Copa: Poco simétrica	
Código: PQLT.S1.1358 Nombre común: Arabisco Nombre científico: <i>Jacaranda mimosifolia</i> Familia: Bignoniaceae Circunferencia de fuste: 78.00 cm Diámetro: 24.83 cm Altura total: 6.50 m Altura comercial: 2.60 m Tam. de copa media (Norte-Sur): 8.00 m Tam. de copa media (Este-Oeste): 8.00 m Follaje o porcentaje de hojas: 85 % Estado de madurez: Adulto Floración: Sin floración		Caída: Bajo Afectación a construcciones: Bajo Interferencia con la circulación: Bajo Daño a infraestructura por raíces: Bajo Interferencia con redes aéreas (eléctricas): Alto		PRESENCIA DE ENFERMEDADES	
PRESENCIA DE PLAGAS		ESTADO FITOSANITARIO		UBICACIÓN DE LAS ENFERMEDADES	
Plantas parásitas Epifitas o líquenes		Fructificación: Sin fructificación Rectitud de fuste: Torcido Espacio de crecimiento: Amplio Estado del actual: Vivo		Fuste Hojas	
		ACCIONES RECOMENDADAS			
		Raíz: Buena Fuste: Regular Corteza: Regular Ramas: Regulares Hojas: Regulares Cima: Buena Copa: Poco simétrica		Control de plantas epifitas o parásitas Tratamiento de daños mecánicos Eliminación de objetos extraños en tallo y ra... Derribo	

Figura 72. Reporte completo de un árbol seleccionado.

5.1.3.2. Usuario registrado (Investigador)

Este usuario puede acceder a todas las interfaces especificadas para el usuario común.

***Pantalla de Inicio de Sesión.**

El inicio de sesión (**RF05**) está destinado a los usuarios que forman parte del proyecto de investigación forestal, realizan el levantamiento, control del estado y condiciones del árbol.

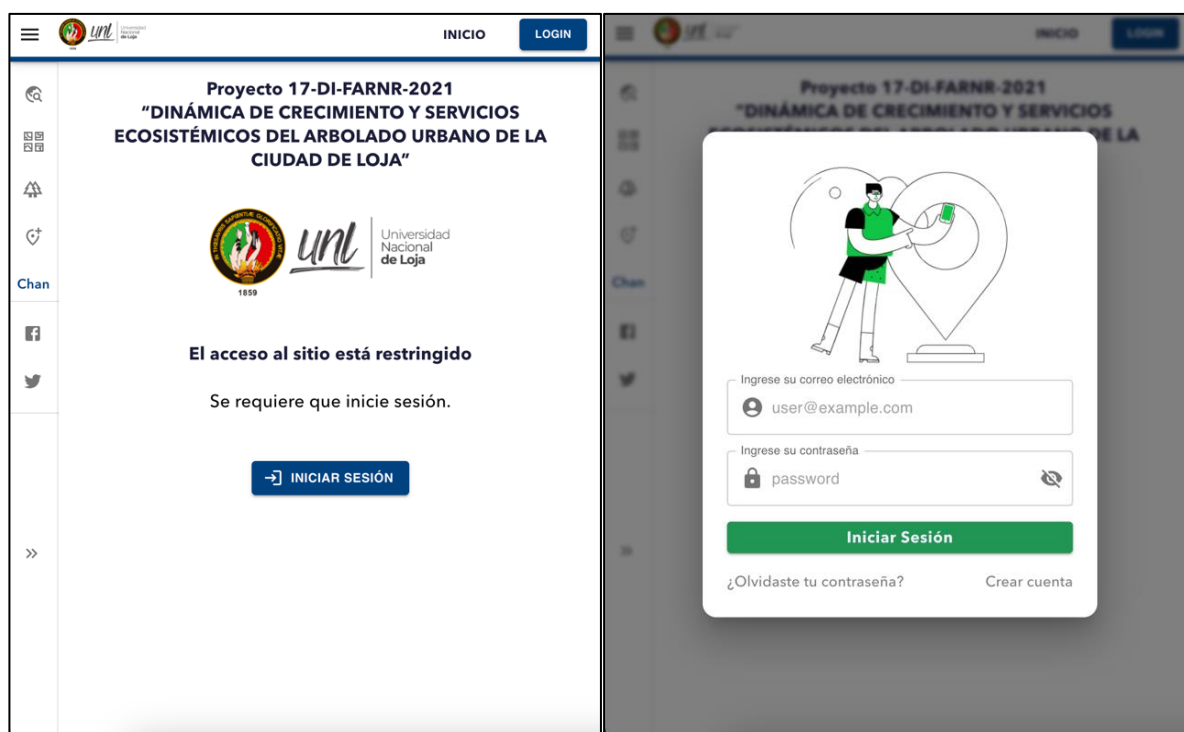


Figura 73. Pantallas para la autenticación de usuario.

En esta pantalla se puede acceder al sitio de recuperación de contraseñas (**RF06**) (ver **Figura 74**) y creación de cuenta (**RF12**) (ver **Figura 75**).

Una vez accedido a la cuenta con las credenciales correctas, la actualización del perfil de usuario se la realiza en la sección "**Mi Cuenta**" (**RF06**), donde se puede modificar los datos de usuario y contraseña (ver **Figura 76**).

***Pantalla Administración.**

La página de administración permitirá el acceso aquellos usuarios registrados y cuya cuenta se encuentre en estado habilitado (ver **Figura 77**). Dos acciones están disponibles para el usuario investigador: **Editar** árbol (ver **Figura 78**, **Figura 80**) y **Desplazar** ubicación longitud y latitud de árbol (ver **Figura 79**).

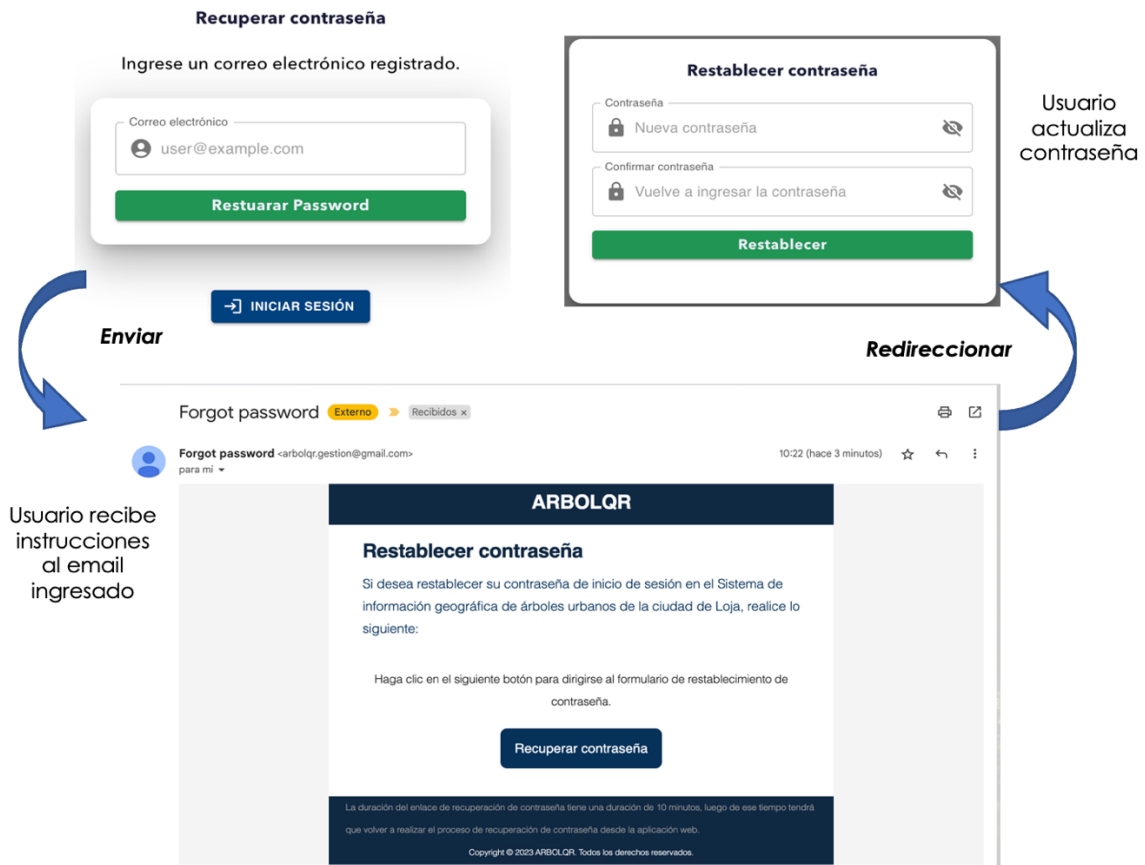


Figura 74. Proceso de recuperación de contraseña.

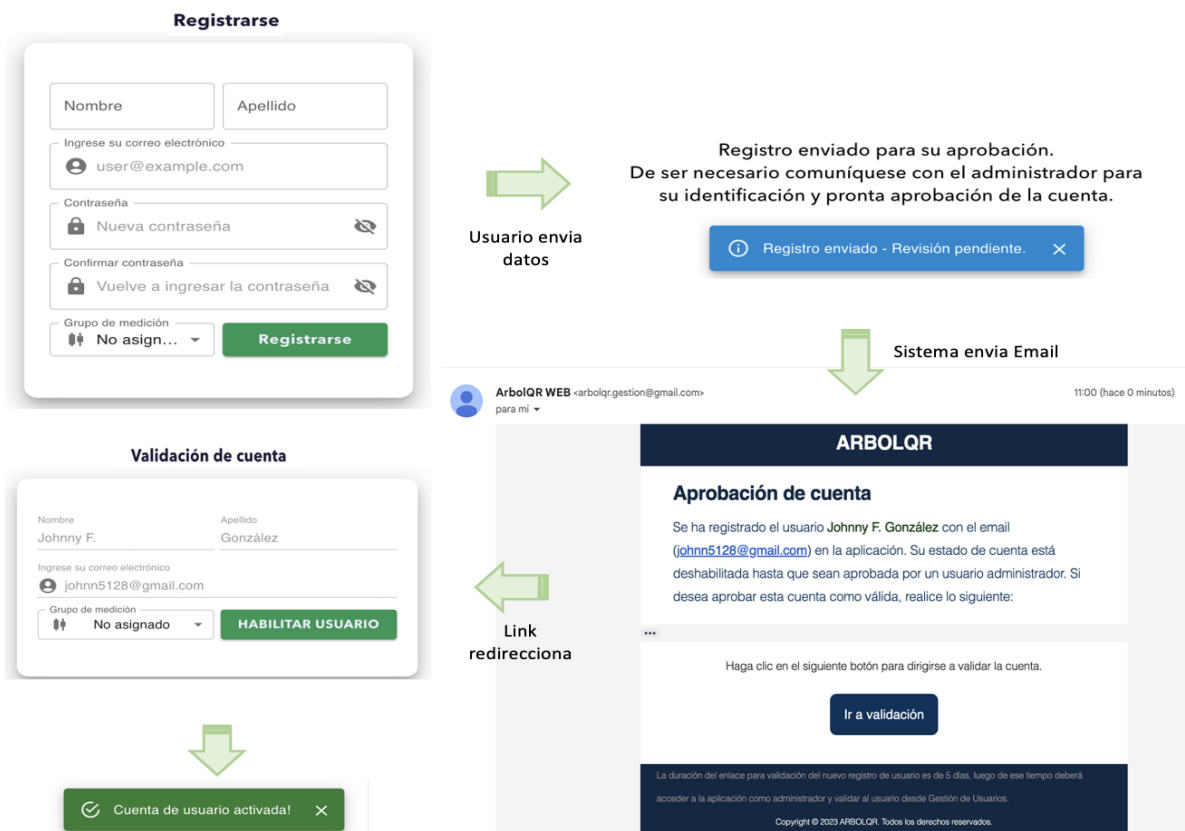


Figura 75. Proceso de registro de usuario, envió de email al usuario administrador.

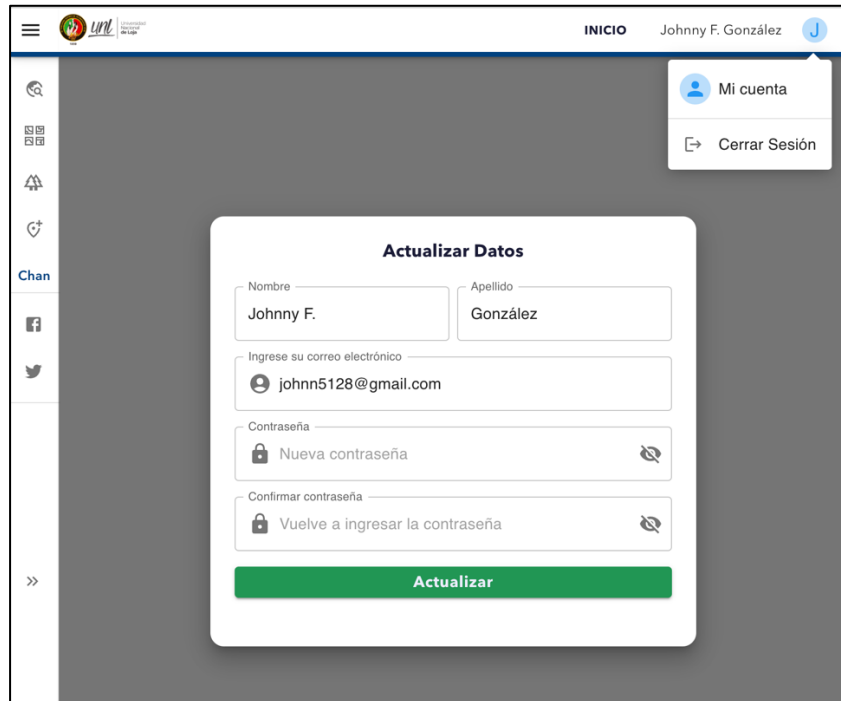


Figura 76. Vista Actualizar perfil de usuario.

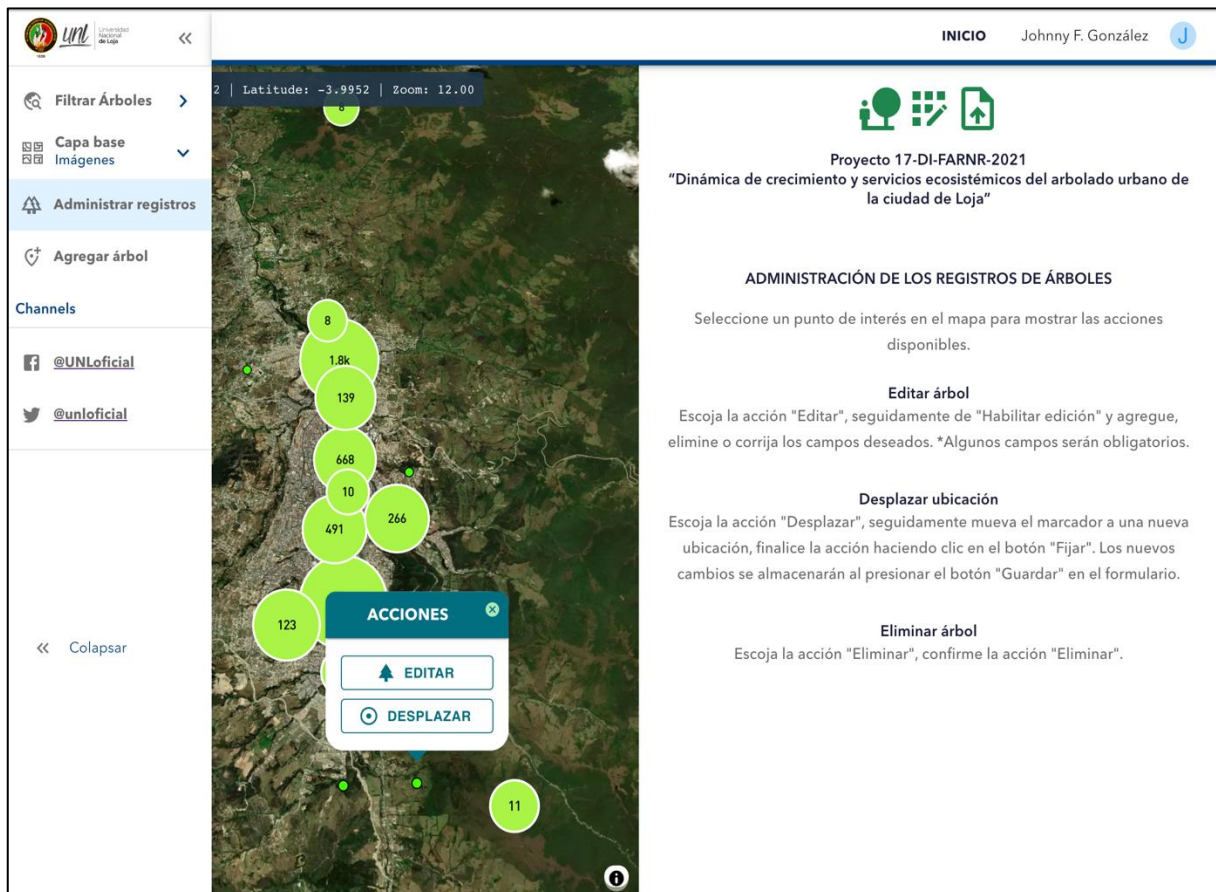


Figura 77. Vista Administración de árboles, acciones para usuarios con rol investigador.

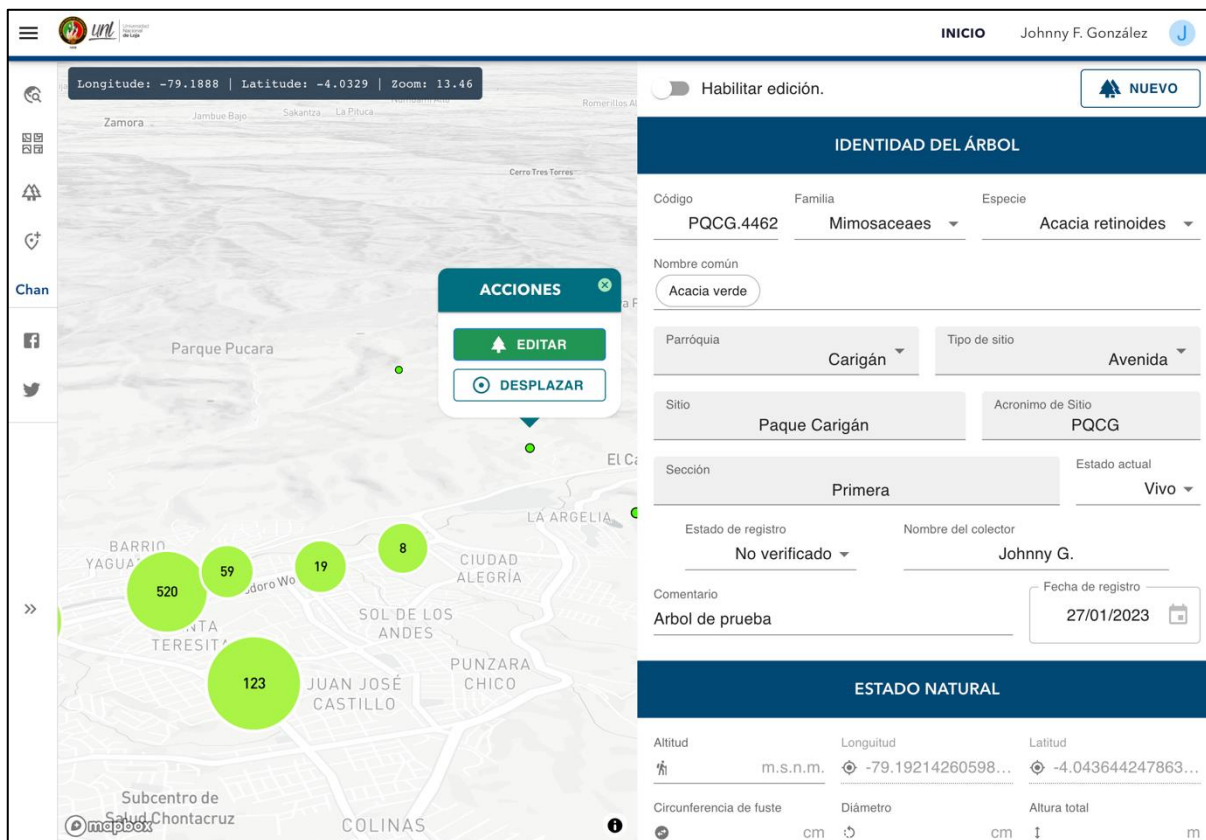


Figura 78. Acción Editar árboles en página Administración.



Figura 79. Acción Desplazar ubicación de árbol en mapa.

***Pantalla de Registración de árboles.**

En esta pantalla el registro de nuevos árboles se realiza completando los campos obligatorios como son los correspondientes a Identidad de árbol (ver **Figura 80**), más el campo de longitud y latitud (ver **Figura 81**).

Deshabilitar edición. NUEVO

IDENTIDAD DEL ÁRBOL

Código	Familia	Especie
PQCG.4462	Mimosaceas	Acacia retinoides

Nombre común: Nuevo Item [ENTER]

Parróquia: Tipo de sitio:

Sitio: Acronimo de Sitio:

Sección: Estado actual:

Estado de registro: Nombre del colector: Fecha de registro:

Comentario:

ESTADO NATURAL

Altitud Longitud Latitud

Circunferencia de fuste	cm	Diámetro	cm	Altura total	m
Altura comercial	m	Tam. de copa media (Norte-...)	m	Tam. de copa media (Este-O...)	m
Follaje o porcentaje de hojas	%	Estado de madurez		Floración	
Fructificación		Rectitud de fuste		Espacio de crecimiento	

CONDICIÓN GENERAL

ESTADO FITOSANITARIO

PRESENCIA DE ENFERMEDADES

UBICACIÓN DE LAS ENFERMEDADES

Raíz	Fuste
Corteza	Ramas
Hojas	Cima
Copa	

CANCELAR GUARDAR

Figura 80. Campos para la Edición o Creación de árbol.

INICIO Johnny F. González

IDENTIDAD DEL ÁRBOL

Familia	Especie
Nombre común	Parróquia
Tipo de sitio	Sitio
Acronimo de Sitio	Sección
Estado actual	Estado de registro
	Vivo
Nombre del colector	Comentario

ESTADO NATURAL

Altitud	Longitud	Latitud
m.s.n.m.	-79.2025440626...	-4.02047661748...
Circunferencia de fuste	Diámetro	Altura total
cm	cm	m
Altura comercial	Tam. de copa media (Nort-...)	Tam. de copa media (Este-...)
m	m	m

CANCELAR GUARDAR

Figura 81. Vista de pantalla para Registrar árbol.

5.1.3.3. Usuario registrado (administrador)

Este usuario puede acceder a todas las interfaces anteriormente especificadas, más la habilitación de la acción “*Eliminar*” en la vista de Administración de árboles (**RF10**), habilitación de la vista de Generar Reporte en el menú lateral izquierdo (**RF11**), y la sección de administración de familias, especies y de usuarios registrados (ver **Figura 83**).

***Pantalla Administración de árboles.**

La acción “*Eliminar*” es *habilitado* para este usuario. La eliminación del árbol implica un borrado lógico en la base de datos, por tanto, no se mostrará en el mapa cuando cargue los datos. Para esta acción se requiere confirmación (ver **Figura 82**).

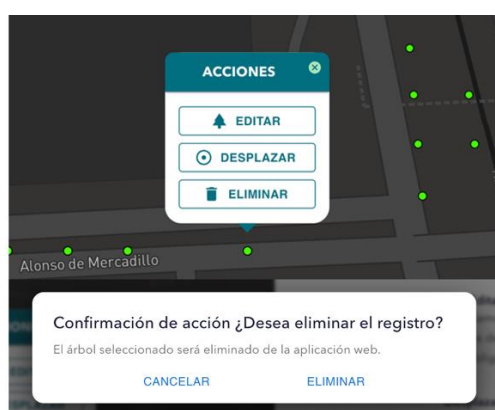


Figura 82. Acción Eliminar habilitado para usuario con rol Administrador.



Figura 83. Vista de Administración de árboles, acciones para usuario con rol administrador.

*Pantalla Generar reporte

Las acciones posible en esta sección son:

- Mostrar los registros de árboles actuales en la tabla (ver **Figura 84**).
- Filtrar búsquedas (ver **Figura 86**).
- Mostrar u ocultar las columnas (ver **Figura 85**).
- Descargar los registros en formato CSV (ver **Figura 87**).

Registros de árboles urbanos de la ciudad de Loja		🔍	📄	☰	☰
Acacia longifolia ✕					
Sitio	Parque Recreacional Jipiro				
Sección	Calle Velasco Ibarra				
Especie	Acacia longifolia				
Familia	Fabaceae				
Árbol verificado	No verificado				
Longitud	-79.20595667				
Latitud	-3.96963500				
Altura total	3.51				
Altura comercial	0.77				
Tam. copa (NS)	2.57				
Tam. copa (EW)	2.44				

Figura 84. Vista de la interfaz Reporte adaptado a un tamaño de pantalla menor a 900px de ancho.

Registros de árboles urbanos de la ciudad de Loja

SITIO	SECCIÓN	ESPECIE	FAMILIA	ÁRBOL VERIFICADO	LONGITUD	LATITUD	ALTURA TOTAL	ALTURA COMERCIAL
Parque Recreacional Jipiro	Calle Velasco Ibarra	Fraxinus chinensis	Oleaceae	Verificado	-79.2034	-3.9738	9.63	8.00
Parque Lineal La Tebaida	Estacionamiento interno del parque hasta el Puente Sendero del parque Lineal Sur	Platanus occidentalis	Platanaceae	No verificado	-79.2025	-4.0178	8.87	3.06
Parque Recreacional Jipiro	Calle Velasco Ibarra	Salix humboldtiana	Salicaceae	No verificado	-79.20378947	-3.97116134	22.22	5.73
Parque Lineal La Tebaida	Estacionamiento interno del parque hasta el Puente Sendero del parque Lineal Sur	Alnus acuminata	Betulaceae	No verificado	-79.2028	-4.0170	7.90	3.80

Show Columns

- Nombre de colector
- Sitio
- Tipo de sitio
- Parróquia
- Sección
- Acrónimo
- Código
- Especie
- Familia
- Nombre Común
- Nombre Común(2)
- Nombre Común(3)
- Nombre Común(4)
- Fecha de registro
- Árbol verificado
- Vivo
- Comentario
- Foto
- Altitud
- Longitud
- Latitud
- Circunferencia de fuste
- Diámetro

Figura 85. Mostrar u ocultar las columnas de la tabla.

Registros de árboles urbanos de la ciudad de Loja

Acacia longifolia

SITIO	SECCIÓN	ESPECIE	FAMILIA	ÁRBOL VERIFICADO	LONGITUD
Parque Recreacional Jipiro	Calle Velasco Ibarra	Acacia longifolia	Fabaceae	No verificado	-79.20595667
Parque Recreacional Jipiro	Calle Velasco Ibarra	Acacia longifolia	Fabaceae	No verificado	-79.20317933
Parque Recreacional Jipiro	Calle Velasco Ibarra	Acacia longifolia	Fabaceae	No verificado	-79.20324655
Parque Recreacional Jipiro	Calle Velasco Ibarra	Acacia longifolia	Fabaceae	No verificado	-79.20344755
Parque Recreacional Jipiro	Calle Velasco Ibarra	Acacia longifolia	Fabaceae	No verificado	-79.20314010

FILTERS RESET

- Nombre de colector: All
- Sitio: All
- Tipo de sitio: All
- Parróquia: All
- Sección: All
- Acrónimo: All
- Código: All
- Especie: Acacia longifolia
- Familia: Nombre Común (All selected)
- Nombre Común(2): All
- Nombre Común(3): All
- Nombre Común(4): All
- Fecha de registro: All
- Árbol verificado: All
- Vivo: All

Figura 86. Aplicar filtros por columnas.

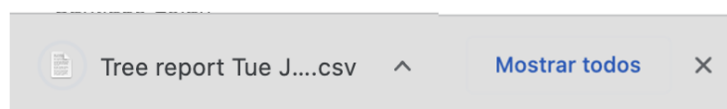


Figura 87. Archivo descargado en formato csv.

*Pantalla Administrar Tablas

El usuario con rol de administrador puede gestionar el registro y actualización de tres tablas: **familias** de árboles, **especies** por familias y **usuarios** que se muestran a continuación:

ID	Nombre	Habilitado	Acciones
1	Mimosaceae	✓	
2	Fabaceae	✓	
3	Leguminosae	✓	
4	Betulaceae	✓	

Figura 88. Administrar registros de familias.

ID	Nombre científico	FamilialID	Habilitado	Acciones
1	Acacia baileyana	2	✓	
2	Acacia heterophylla	2	✓	
3	Acacia retinoides	1	✓	
4	Acacia macracantha	2	✓	
5	Acacia dealbata	3	✓	

Figura 89. Administrar registros de especies.

ID	Nombre	Apellido	Email	IDGrupoMedición	Habilitado	Rol	Acciones
8...	JONAS	CASTRO	jonas@unl.edu...	1	✓	administrador	
9...	EMAN	TRINIDAD	eman@unl.edu...	4	✓	investigador	
10...	INTEGRAL	CASTRO	integral@gmail.com	-	✓	investigador	

Figura 90. Administrar registros de usuarios.

5.2. Diagramas de componentes

En la **Figura 91** se detalla la arquitectura de la aplicación WEBGISTREE, tanto del cliente web React y el servicio Web (API - REST).

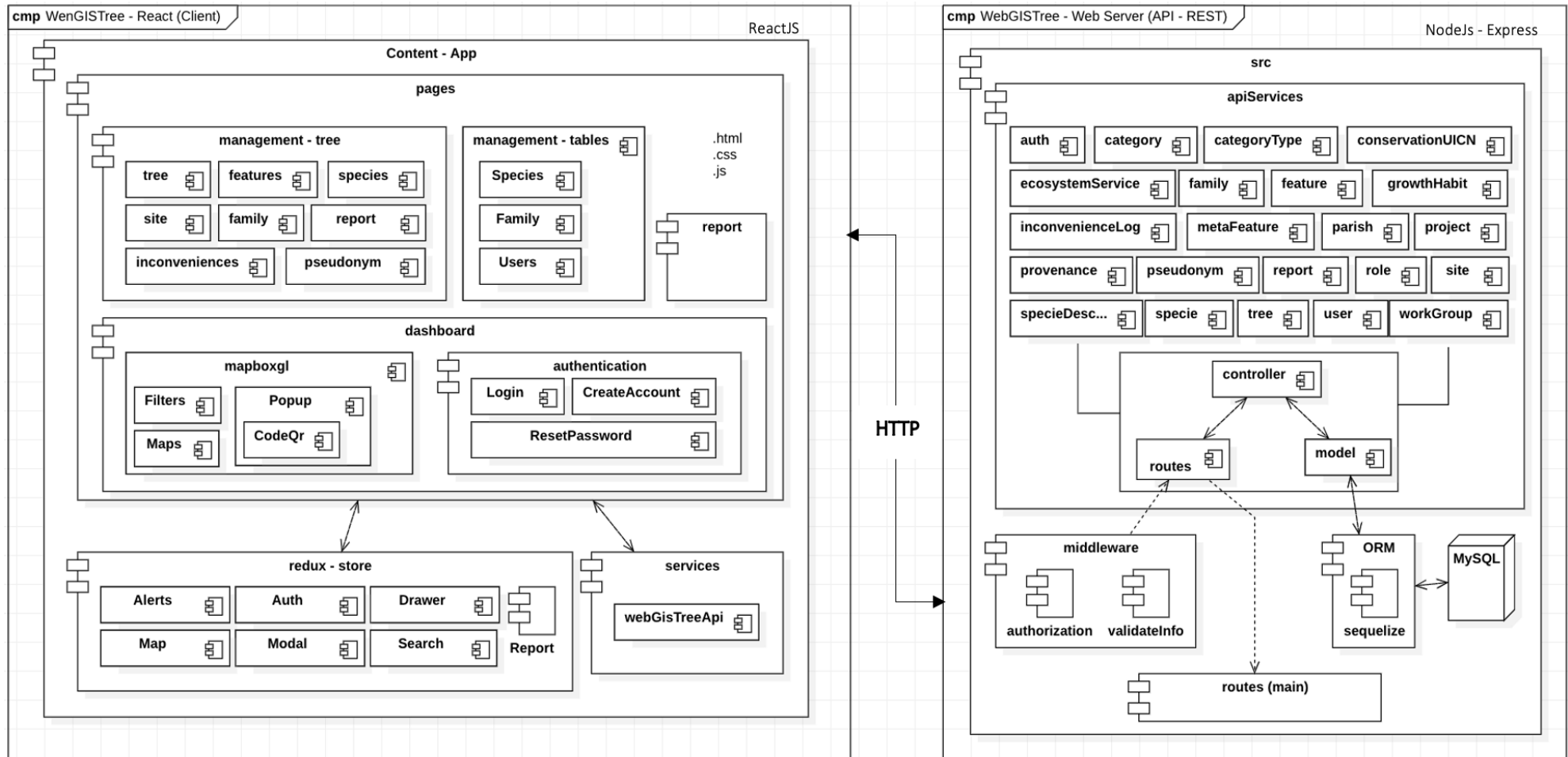


Figura 91. Diagrama de componentes de la aplicación web.

5.3. Diagramas de despliegue

El siguiente diagrama de despliegue se ilustra el diseño de la arquitectura física de la aplicación (ver **Figura 92**), este consta de un Cliente web React que se comunica con el Servidor Node, donde se encuentra el servicio web (API-REST) WebGISTree-backend, que interactúa a través de un Mapeador Relacional de Objetos (ORM) con el servidor de base de datos relacional MySQL.

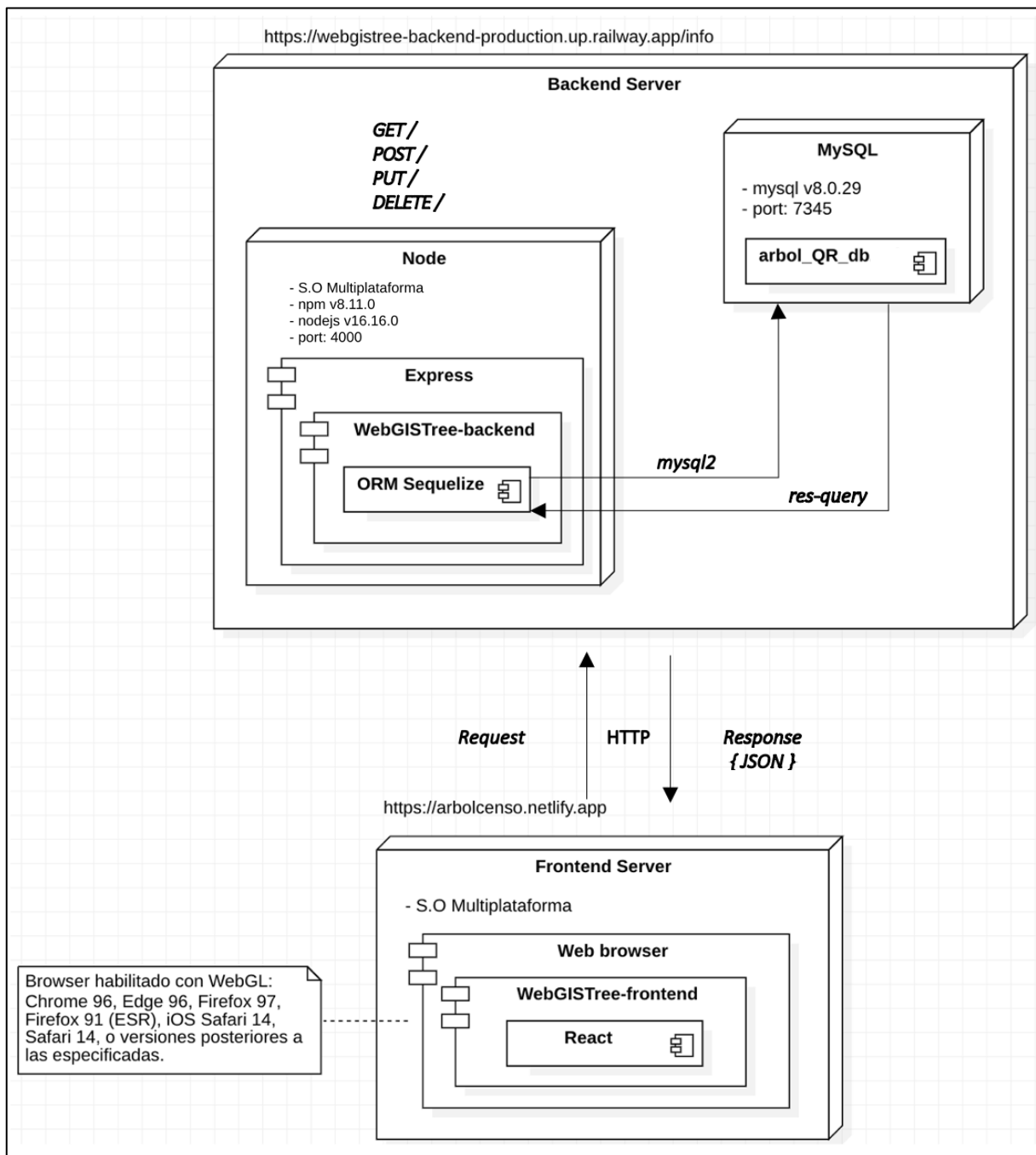


Figura 92. Diagrama de despliegue.

Anexo 5. Pruebas al servicio Web API-REST



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Pruebas al servicio web API-REST

Proyecto: “Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”.

Autor:

Johnny Fabián González Guamán

1. Introducción

En el presente documento se presentan las pruebas realizadas a los servicios del API-REST del sistema WebGISTree, con la finalidad de comprobar el correcto funcionamiento de los end-points, controlando que las respuestas devueltas sean correctas y en un tiempo aceptable.

2. Propósito

Validar el correcto funcionamiento de los servicios implementados para el sistema WebGISTree, de igual manera detectar posibles errores presentes en la implementación del servicio web. Para todo esto se hizo uso de la herramienta Postman.

3. Referencias

Referencia	Sitio web
Postman	https://www.postman.com
Railway	https://railway.app

4. Pruebas

Para las pruebas del API-REST se utilizó la herramienta Postman, misma que permitió el envío de peticiones HTTP hacia los servicios web, obteniendo una respuesta en formato json. Para realizar estas pruebas se subió la aplicación a un servidor gratuito (Railway), con el fin de obtener tiempos de respuesta más similares a un ambiente real.

Tabla 23. Petición para obtener todos los registros de árboles.

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/tree	No	GET	Obtener todos los registros de árboles	2.34s

```
1
2  *auth*: true,
3  *total*: 4457,
4  *tree*: [
5    {
6      *codigo*: "PQ3P61",
7      *colectorRegistro*: "Ronald Fabian Gualan Gordillo y Jean Jaramillo",
8      *idEspecie*: 46,
9      *idSitio*: 72,
10     *comentario*: "Realizar una para optimizar recursos y prevenir rompimientos de ramas.",
11     *foto*: null,
12     *idUsuarioHabilito*: "85b7687e-56ab-4edf-b871-b905b0d59545",
13     *fechaRegistro*: "2021-09-25T00:00:00.000Z",
14     *habilitado*: {
```

Tabla 24. Petición para obtener un registros de árbol por su ID

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/tree/:treeld	No	GET	Obtener un registro de árbol por su ID	331ms


```

1  {
2    "tree": {
3      "idInterno": 1219,
4      "id_a": "426fa069-c6b1-472b-b348-2bc60a5b2e58",
5      "codigo": "SPMX143",
6      "colectorRegistro": "Josué Armijos y Leonardo Granda",
7      "idEspecie": 3,
8      "idSitio": 20,
9      "comentario": "Frente al estacionamiento del Kiwi.",
10     "foto": null,
11     "idUsuarioHabilito": "85b7687e-56ab-4edf-b871-b905b0d59545",
12     "fechaRegistro": "2021-10-10T00:00:00.000Z",
13     "habilitado": {
14       "type": "Buffer",

```

Tabla 25. Petición para conocer si un árbol está activo.

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/tree/:treeld	No	GET	Petición para conocer si un árbol está activo	744ms


```

1  {
2    "valid": false,
3    "enable": false,
4    "verifiedTree": false
5  }

```


Tabla 28. Petición para actualizar datos en tabla árbol

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/tree/update-tree/:treeld	Sí	PUT	Actualizar datos en tabla árbol	752ms

PUT https://webgistree-backend-production.up.railway.app/tree/update-tree/426fa069-c6b1-472b-b348-2bc60a5b2e58

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  {
2    "specieId": 12,
3    "siteId": 20,
4    "alive": false,
5    "photo": null,
6    "treeVerified": false,
7    "commit": "Frente al estacionamiento del Kiwi.",
8    "recorded": "Josué Armijos y Leonardo Granda"
9  }

```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 752 ms Size: 298 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "update": true,
3    "res": [
4      1
5    ]
6  }

```

Tabla 29. Petición para habilitar estado de árbol

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/tree/update-status-register/:treeld	Sí	PUT	Habilitar estado de árbol	343ms

PUT https://webgistree-backend-production.up.railway.app/tree/update-status-register/426fa069-c6b1-472b-b348-2bc60a5b2e58

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1  {
2    "habilitado": true
3  }

```

Body Cookies Headers (9) Test Results Status: 200 OK Time: 343 ms Size: 316 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "update": true,
3    "res": [
4      0
5    ]
6  }

```

Tabla 30. Petición para obtener los roles de usuarios

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/role	Sí	GET	Obtener los roles de usuarios	892ms

GET https://webgistree-backend-production.up.railway.app/role

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Headers 7 hidden

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1Ijoi...			
Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 892 ms Size: 532 B Save Response

Pretty Raw Preview Visualize JSON

```

1
2  "auth": true,
3  "roles": [
4    {
5      "id_r": "912b105c-018c-49e4-8e57-7a51b7d0d0e4",
6      "tipo": "administrador",
7      "habilitado": {
8        "type": "Buffer",
9        "data": [
10         1
11        ]
12      }
13    },
14    {
15      "id_r": "ba1c005c-2711-4a7a-899a-2e4176b309e4",
16      "tipo": "administrador",
17      "habilitado": {
18        "type": "Buffer",
19        "data": [
20         1
21        ]
22      }
23    }
24  ]

```

Tabla 31. Petición para obtener lista de usuarios

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/app-user	Sí	GET	Obtener lista de usuarios	778ms

GET https://webgistree-backend-production.up.railway.app/app-user

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Headers 7 hidden

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1Ijoi...			
Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 778 ms Size: 1.62 KB Save Response

Pretty Raw Preview Visualize JSON

```

1
2  "auth": true,
3  "users": [
4    {
5      "id_u": "e...",
6      "nombre": "...n",
7      "apellido": "P...",
8      "email": "...edu.ec",
9      "habilitado": {
10       "type": "Buffer",
11       "data": [
12        1
13       ]
14     },
15     "fecha_registro": "2023-01-27T17:01:22.000Z",
16     "fecha_actualizo": "2023-01-27T17:09:00.000Z",
17     "foto": null,
18     "idGrupoMedicacion": null,
19     "idRol": "5"
20   }

```

Tabla 32. Petición para obtener información de usuarios logueado

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/app-user/me	Sí	GET	Obtener información de usuarios logueado	816ms

The screenshot shows a REST client interface for the endpoint `https://webgistree-backend-production.up.railway.app/app-user/me`. The request method is GET. The response status is 200 OK, with a time of 816 ms and a size of 414 B. The response body is a JSON object:

```

1  {
2    "auth": true,
3    "user": {
4      "nombre": "Johnny",
5      "apellido": "González",
6      "email": "jfgonzalezg@unl.edu.ec",
7      "foto": "foto/Johnyy.jpg",
8      "idGrupoMedicacion": 1
9    }
10 }

```

Tabla 33. Petición para traer datos de usuario por su Id

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/app-user/:userId	Sí	GET	Obtener un usuario por su Id	189ms

The screenshot shows a REST client interface for the endpoint `https://webgistree-backend-production.up.railway.app/app-user/85b7687e-56ab-4edf-b871-b905b0d59545`. The request method is GET. The response status is 200 OK, with a time of 189 ms and a size of 522 B. The response body is a JSON object:

```

1  {
2    "auth": true,
3    "user": {
4      "nombre": "Johnny",
5      "apellido": "González",
6      "email": "jfgonzalezg@unl.edu.ec",
7      "habilitado": {
8        "type": "Buffer",
9        "data": [
10       1
11      ]
12     },
13     "foto": "foto/Johnyy.jpg",
14     "idGrupoMedicacion": 1,
15     "idRol": "bacccaef-2c71-43fa-994d-298f76d309ee"
16   }
17 }

```


Tabla 34. Petición para registrar un usuario

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/app-user/register-user	No	POST	Registrar un usuario	892ms

The screenshot shows a REST client interface for a POST request. The URL is `https://webgistree-backend-production.up.railway.app/app-user/register-user`. The request body is a JSON object:

```

1  {
2    "nombre": "Johnny F",
3    "apellido": "Gonzalez",
4    "email": "jfgonzalez@gmail.com",
5    "contrasena": "J12SonTTa5",
6    "idGrupoMedicacion": null,
7    "foto": null
8  }

```

The response is a JSON object:

```

1  {
2    "typeaction": "created",
3    "message": "Usuario creado!"
4  }

```

Tabla 35. Petición para habilitar un usuario

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/app-user/enable/:userId	Sí	PUT	Habilitar usuario	1044ms

The screenshot shows a REST client interface for a PUT request. The URL is `https://webgistree-backend-production.up.railway.app/app-user/enable/06024e4c-96be-4790-bfc5-20214383d998`. The request headers include an Authorization token:

KEY	VALUE	DESCRIPTION	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...			

The response is a JSON object:

```

1  {
2    "message": "Usuario habilitado!"
3  }

```

Tabla 36. Petición para actualizar usuario

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/app-user/update-account	Sí	PUT	Actualizar usuario	124ms

The screenshot shows a REST client interface for a PUT request. The URL is https://webgistree-backend-production.up.railway.app/app-user/update-account. The request body is a JSON object with the following fields: nombre: "JOHNNY", apellido: "GONZALEZ", email: "johnn5128@gmail.com", contrasena: "SF45sonda", idGrupoMedicacion: 3, and foto: "foto/kljohnnyGONZALEZGUAMAN.jpg". The response is a JSON object with the following fields: update: true and message: "Usuario actualizado". The status is 200 OK, the time is 124 ms, and the size is 314 B.

Tabla 37. Petición para deshabilitar usuario

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/app-user/disable/:userId	Sí	PUT	Deshabilitar usuario	953ms

The screenshot shows a REST client interface for a PUT request. The URL is https://webgistree-backend-production.up.railway.app/app-user/disable/06024e4c-96be-4790-bfc5-20214383d998. The request has an Authorization header with a Bearer token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ... The response is a JSON object with the following field: message: "Usuario deshabilitado". The status is 200 OK, the time is 953 ms, and the size is 307 B.

Tabla 38. Petición para verificar token.

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/auth/is-verify	Sí	GET	Verificar token	234ms

GET ▼ https://webgistree-backend-production.up.railway.app/auth/is-verify Send ▼

Params Authorization **Headers (8)** Body Pre-request Script Tests Settings Cookies

Authorization Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

Key Value Description

Body Cookies Headers (9) Test Results Status: 200 OK Time: 234 ms Size: 380 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ ≡

```

1  {
2    "auth": true,
3    "user": {
4      "id": "85b7687e-56ab-4edf-b871-b905b0d59545",
5      "rol": [
6        "administrador"
7      ]
8    }
9  }

```

Tabla 39. Petición para verificar si existe el email

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/auth/is-verify	No	POST	Verificar si existe el email	7ms

POST ▼ https://webgistree-backend-production.up.railway.app/auth/verify-user Send ▼

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON ▼ Beautify

```

1  {
2    "email": "jfgonzalezg@unl.edu.ec"
3  }

```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 7 ms Size: 279 B Save Response ▼

Pretty Raw Preview Visualize JSON ▼ ≡

```

1  {
2    "auth": true
3  }

```

Tabla 40. Petición para enviar link de recuperar contraseña

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/auth/send-email-reset-password	No	POST	Enviar link de recuperar contraseña	198ms

The screenshot shows a REST client interface for a POST request. The URL is `https://webgistree-backend-production.up.railway.app/auth/send-email-reset-password`. The request body is a JSON object: `{ "email": "jfgonzalezg@unl.edu.ec" }`. The response status is `200 OK` with a response time of `198 ms` and a size of `303 B`. The response body is a JSON object: `{ "auth": true }`.

Tabla 41. Petición para obtener las meta-características.

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/meta-feature	No	GET	Obtener las meta-características	868ms

The screenshot shows a REST client interface for a GET request. The URL is `https://webgistree-backend-production.up.railway.app/meta-feature`. The response status is `200 OK` with a response time of `868 ms` and a size of `2.56 KB`. The response body is a JSON object: `{ "auth": true, "total": 15, "res": [{ "id_mc": 1, "key": "altitud", "nombre": "Altitud", "tipo_dato": "decimal", "u_medida": "m.s.n.m.", "habilitado": { "type": "Buffer", "data": [1] } }] }`.

Tabla 42. Petición para obtener especies

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/specie	No	GET	Obtener especies	199ms


```

1  {
2    "auth": true,
3    "total": 116,
4    "res": [
5      {
6        "id_e": 1,
7        "nombre_cientifico": "Acacia baileyana",
8        "idFamilia": 2,
9        "habilitado": {
10         "type": "Buffer",
11         "data": [
12           1
13         ]
14       }
15     },
16     {
17       "id_e": 2,
18       "nombre_cientifico": "Acacia heterophylla",
19       "idFamilia": 2,
20       "habilitado": {
21         "type": "Buffer",
22         "data": [
23           1
24         ]
25       }
26     }
27   ]
28 }
  
```

Tabla 43. Petición para filtrar especies por familia

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/specie/filter-family/:id	No	GET	Filtrar especies por familia	727ms


```

1  {
2    "res": [
3      {
4        "id_e": 1,
5        "nombre_cientifico": "Acacia baileyana",
6        "idFamilia": 2,
7        "habilitado": {
8         "type": "Buffer",
9         "data": [
10          1
11        ]
12       }
13     },
14     {
15       "id_e": 2,
16       "nombre_cientifico": "Acacia heterophylla",
17       "idFamilia": 2,
18       "habilitado": {
19         "type": "Buffer",
20         "data": [
21          1
22        ]
23       }
24     }
25   ]
26 }
  
```

Tabla 44. Petición para registrar especie

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/specie/register-specie	Sí	POST	Registrar especie	329ms

The screenshot shows a REST client interface for a POST request to `https://webgistree-backend-production.up.railway.app/specie/register-specie`. The request body is a JSON object:

```

1 {
2   "idFamilia": 1,
3   "nombreCientifico": "Acacia baileyana (Prueba)",
4   "habilitado": false
5 }

```

The response status is `200 OK` with a time of `329 ms` and a size of `354 B`. The response body is:

```

1 {
2   "action": "created",
3   "message": "Registro creado correctamente!"
4 }

```

Tabla 45. Petición para actualizar especie

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/specie/update-specie/:specieId	Sí	PUT	Actualizar especie	560ms

The screenshot shows a REST client interface for a PUT request to `https://webgistree-backend-production.up.railway.app/specie/update-specie/:specieId`. The request body is a JSON object:

```

1 {
2   "nombre_cientifico": "Tecoma stans",
3   "idFamilia": 17,
4   "habilitado": false
5 }

```

The response status is `200 OK` with a time of `560 ms` and a size of `297 B`. The response body is:

```

1 {
2   "update": true,
3   "res": [
4     1
5   ]
6 }

```

Tabla 46. Petición para obtener lista de nombre comunes

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/pseudonym	Sí	GET	Obtener nombres comunes	707ms

GET https://webgistree-backend-production.up.railway.app/pseudonym

Status: 200 OK Time: 707 ms Size: 11.56 KB

```

1  {
2    "auth": true,
3    "total": 118,
4    "pseudonym": [
5      {
6        "id_ps": 1,
7        "nombre_comun": "Acacia verde",
8        "idEspecie": 3,
9        "habilitado": {
10       "type": "Buffer",
11       "data": [
12         1
13       ]
14     }
15   ],

```

Tabla 47. Petición para registrar familia

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/family/update-family/:familyId	Sí	PUT	Actualizar familia	295ms

PUT https://webgistree-backend-production.up.railway.app/family/update-family/:familyId

Status: 200 OK Time: 295 ms Size: 316 B

```

1  {
2    "nombre": "Asparagaceaes",
3    "habilitado": false
4  }

```

```

1  {
2    "update": true,
3    "res": [
4      1
5    ]
6  }

```

Tabla 48. Petición para obtener sitios

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/site	No	GET	Obtener sitios	1058ms

GET https://webgistree-backend-production.up.railway.app/site Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results Status: 200 OK Time: 1058 ms Size: 17.08 KB Save Response

```

1  {
2    "total": 92,
3    "res": [
4      {
5        "id_s": 1,
6        "nombre": "Barrio Tebaida",
7        "tipo": null,
8        "acronimoSitio": "BLTE.S4-5",
9        "seccion": "Cuarta - Quinta",
10       "idParroquia": null,
11       "habilitado": {
12         "type": "Buffer",
13         "data": [
14           1
15         ]
16       }
17     },
18     ...

```

Tabla 49. Petición para registrar nuevo sitio

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/site/register-site	Sí	POST	Registrar nuevo sitio	550ms

POST https://webgistree-backend-production.up.railway.app/site/register-site Send

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```

1  {
2    "sitio": "La Argelia (Prueba2)",
3    "tipoSitio": "Avenida",
4    "seccion": "Parque Lineal",
5    "acronimoSitio": "HSAS.S9",
6    "idParroquia": 1
7  }

```

Body Cookies Headers (8) Test Results Status: 200 OK Time: 550 ms Size: 389 B Save Response

```

1  {
2    "message": "Acción realizada",
3    "action": {
4      "typeAction": "creation",
5      "message": "Registro creado exitosamente!",
6      "id": 94
7    }
8  }

```


Tabla 50. Petición para registrar nuevo sitio

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/site/register-site	Sí	POST	Registrar nuevo sitio	550ms

The screenshot shows a REST client interface for a POST request to `https://webgistree-backend-production.up.railway.app/site/register-site`. The request body is a JSON object with the following fields: `sitio` (La Argelia (Prueba2)), `tipoSitio` (Avenida), `seccion` (Parque Lineal), `acronimoSitio` (HSAS.S9), and `idParroquia` (1). The response is a JSON object with a success message and an `id` of 94.

```

1 {
2   "sitio": "La Argelia (Prueba2)",
3   "tipoSitio": "Avenida",
4   "seccion": "Parque Lineal",
5   "acronimoSitio": "HSAS.S9",
6   "idParroquia": 1
7 }

```

```

1 {
2   "message": "Acción realizada",
3   "action": {
4     "typeAction": "creation",
5     "message": "Registro creado exitosamente!",
6     "id": 94
7   }
8 }

```

Tabla 51. Petición para obtener localizaciones de árboles

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/feature/locations	No	GET	Obtener localizaciones de árboles	8.56s

The screenshot shows a REST client interface for a GET request to `https://webgistree-backend-production.up.railway.app/feature/locations`. The response is a JSON array of location objects. The first object shown has a `type` of "Feature", a `geometry` of "Point" with coordinates `[-79.2034, -3.9738]`, and `properties` including `id`, `code`, and `treeVerified`.

```

1 {
2   "locations": [
3     {
4       "type": "Feature",
5       "geometry": {
6         "type": "Point",
7         "coordinates": [
8           "-79.2034",
9           "-3.9738"
10        ]
11      },
12      "properties": {
13        "id": "000ddf26-5029-46ad-bb59-a3acba0bfa9a",
14        "code": "PQJV.S8.1",
15        "treeVerified": true
16      }
17    }
18  ]
19 }

```

Tabla 52. Petición para obtener propiedades para Id de árbol

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/feature/:treeld	No	GET	Obtener propiedades del árbol por Id	382ms

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** https://webgistree-backend-production.up.railway.app/feature/426fa069-c6b1-472b-b348-2bc60a5b2e58
- Status:** 200 OK
- Time:** 382 ms
- Size:** 6.29 KB

The response body is displayed in JSON format (Pretty view):

```

1  {
2    "properties": {
3      "id": "426fa069-c6b1-472b-b348-2bc60a5b2e58",
4      "code": "CCGO.1219",
5      "recorded": "Josué Armijos y Leonardo Granda",
6      "specie": {
7        "id": 12,
8        "name": "Alnus acuminata",
9        "family": {
10         "id": 4,
11         "name": "Betulaceae"
12       },
13       "pseudonym": [
14         {
15           "id": 12,
16           "name": "Aliso"
17         }
18       ]
19     },
20     "site": {
21       "id": 20,
22       "name": "Río Malacatos",
23       "type": null,
24       "siteAcronym": "CCGO",
25       "section": "Calle Chile hasta calle Gonzanamá",
26       "parish": null
27     },

```

Tabla 53. Petición para actualizar lista de características para un árbol

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/feature/update-feature/:treeld	Sí	POST	Actualizar lista de características para un árbol	707ms

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** https://webgistree-backend-production.up.railway.app/feature/update-feature/:treeld
- Body (Request):** A JSON object with 15 characteristics:


```

1 {
2   "altitud": {"id": 1, "value": "2000", "verified": null},
3   "longX": {"id": 2, "value": "-79.20305125", "verified": null},
4   "latY": {"id": 3, "value": "-4.0439803", "verified": null},
5   "cap": {"id": 4, "value": "208.50", "verified": null},
6   "dap": {"id": 5, "value": "66.37", "verified": null},
7   "ht": {"id": 6, "value": "16.10", "verified": null},
8   "hc": {"id": 7, "value": "3.00", "verified": null},
9   "copaNS": {"id": 8, "value": "6.00", "verified": null},
10  "copaEW": {"id": 9, "value": "6.00", "verified": null},
11  "follaje": {"id": 10, "value": "90", "verified": null},
12  "madurez": {"id": 11, "value": "Viejo", "verified": null},
13  "floracion": {"id": 12, "value": "Iniciando floración", "verified": null},
14  "fructificacion": {"id": 13, "value": "Sin fructificación", "verified": null},
15  "fusteRectitud": {"id": 14, "value": "Muy inclinado", "verified": null},
16  "espaciamiento": {"id": 15, "value": "Amplio", "verified": null}
17 }
```
- Status:** 200 OK, Time: 707 ms, Size: 1.45 KB
- Body (Response):** A JSON object containing a response with request details and action messages:


```

1 {
2   "response": {
3     "request": "creation",
4     "actions": [
5       {
6         "typeAction": "created",
7         "message": "La característica 'altitud' fue creada."
8       },
9       {
10        "typeAction": "created",
11        "message": "La característica 'longX' fue creada."

```

Tabla 54. Petición para filtrar árboles por características

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/feature/geojson-filters	No	POST	Filtrar árboles por características	4.98s


```

1 {
2   "queryList": {
3     "11": "Viejo",
4     "12": "Sin floración",
5     "13": "Sin fructificación",
6     "14": "Muy inclinado",
7     "15": "Moderado"
8   }
9 }

```

```

1 {
2   "response": [
3     {
4       "type": "Feature",
5       "geometry": {
6         "type": "Point",
7         "coordinates": [
8           -79.20707323,
9           -3.96884500
10        ]
11      },
12      "properties": {
13        "id": "024b4cb1-0c01-4fe5-a07e-0a009ca547db",
14        "code": "PQJV.S8.51",
15        "treeVerified": false
16      }
17    }
18  ],
19 }

```

Tabla 55. Petición para actualizar datos en tabla inconveniente

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/inconvenience/update-inconvenience/:treeld	Sí	POST	Actualizar datos en tabla inconveniente	915ms


```

1 {
2   "calidadRaiz": { "categorytype": 1, "beforeValue": 3 },
3   "calidadFuste": { "categorytype": 4, "beforeValue": 5 },
4   "calidadCorteza": { "categorytype": 7, "beforeValue": 9 },
5   "calidadRamas": { "categorytype": 10, "beforeValue": 11 },
6   "calidadHojas": { "categorytype": 13, "beforeValue": "" },
7   "calidadCima": { "categorytype": 16, "beforeValue": "" },
8   "calidadCopa": { "categorytype": 19, "beforeValue": "" },
9   "enFHongos": { "categorytype": 22, "value": "" },
10  "enFTumores": { "categorytype": 23, "value": "" },
11  "enFVieja": { "categorytype": 24, "value": "" }
12 }

```

```

1 {
2   "res": {
3     "request": "creation",
4     "actions": [
5       {
6         "field": 1,
7         "typeAction": "Created",
8         "message": "Característica registrada!"
9       },
10      {
11        "field": 4,
12        "typeAction": "Created",
13        "message": "Característica registrada!"
14      }
15    ]
16  }
17 }

```

Tabla 56. Petición para obtener parroquias

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/parish	No	GET	Obtener parroquias	185ms

GET https://webgistree-backend-production.up.railway.app/parish

Status: 200 OK Time: 185 ms Size: 773 B

```

1  {
2    "auth": true,
3    "total": 6,
4    "res": [
5      {
6        "id_pa": 1,
7        "nombre": "El Sagrario",
8        "habilitado": {
9          "type": "Buffer",
10         "data": [
11           1
12         ]
13       }
14     ]
  }
  
```

Tabla 57. Petición para obtener categorías

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/category	No	GET	Obtener categorías	786ms

GET https://webgistree-backend-production.up.railway.app/category

Status: 200 OK Time: 786 ms Size: 1017 B

```

1  {
2    "auth": true,
3    "total": 8,
4    "res": [
5      {
6        "id_ctg": 1,
7        "nombre": "Estado fitosanitario",
8        "habilitado": {
9          "type": "Buffer",
10         "data": [
11           1
12         ]
13       }
14     ]
  }
  
```

Tabla 58. Petición para obtener tipos de categorías

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/category-type	No	GET	Obtener tipo de categorías	786ms

GET https://webgistree-backend-production.up.railway.app/category-type

Status: 200 OK Time: 194 ms Size: 13.03 KB

```

1  {
2    "auth": true,
3    "total": 92,
4    "res": [
5      {
6        "id_ctgT": 1,
7        "idCategoria": 1,
8        "key": "calidadRaiz",
9        "descripcion": "Raiz",
10       "enum": "Buena",
11       "habilitado": {
12         "type": "Buffer",
13         "data": [
14           1
15         ]
16       }
17     ]
18   }

```

Tabla 59. Petición para obtener columnas para tabla reportes

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/report/headers	Sí	GET	Obtener columnas para tabla reportes	229ms

GET https://webgistree-backend-production.up.railway.app/report/headers

Status: 200 OK Time: 229 ms Size: 5.79 KB

```

1  [
2    {
3      "field": "id",
4      "headerName": "ID"
5    },
6    {
7      "field": "recorded",
8      "headerName": "Nombre de colector"
9    },
10   {
11     "field": "siteName",
12     "headerName": "Sitio"
13   },
14   {
15     "field": "siteType",
16     "headerName": "Tipo de sitio"
17   }
18 ]

```

Tabla 60. Petición para obtener filas para tabla reporte

URL	Token	Método	Descripción	Respuesta (tiempo)
https://webgistree-backend-production.up.railway.app/report/data	Sí	GET	Obtener datos para tabla reportes	40.42s

GET Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (8) Test Results Status: 200 OK Time: 40.42 s Size: 5.67 MB Save Response

Pretty Raw Preview Visualize

```
{
  "resultAll": [
    {
      "id": "000ddf26-5029-46ad-bb59-a3acba0bfa9a",
      "recorded": "Ronald Fabian Gualan Gordillo y Jean Jaramillo",
      "siteName": "Parque Recreacional Jipiro",
      "siteType": "Parque",
      "siteParish": null,
      "siteSection": "Calle Velasco Ibarra",
      "siteAcronym": "PQJV.S8",
      "code": "PQJV.S8.1",
      "specieName": "Fraxinus chinensis",
      "familyName": "Oleaceae",
      "pseudonym1": "Fresno",
      "pseudonym2": null,
      "pseudonym3": null,
      "pseudonym4": null,
      "dateLog": "2021-09-25T00:00:00.000Z",
      "treeVerified": true,
      "alive": true,
      "commit": "Realizar una para optimizar recursos y prevenir rompimientos de ramas.",
      "photo": null,
      "altitud": "2065",
      "longX": "-79.2034",
      "latY": "-3.9738",
      "cap": "254.50",
      "dap": "81.01",
      "ht": "9.63",
      "hc": "8.00",
      "copaNS": "4.80",
      "copaEW": "4.80",
      "follaje": "81",
      "madurez": "Viejo",
      "floracion": "Sin floración",
      "fructificacion": "Sin fructificación",
      "fusteRectitud": "Recto",
      "espaciamiento": "Moderado",
      "calidadRaiz": "Buena",
      "calidadFuste": "Bueno",
      "calidadCorteza": "Buena",
      "calidadRamas": "Buenas",
      "calidadHojas": "Regulares",
      "calidadCima": "Buena",
      "calidadCopa": "Poco simétrica",
      "plgUbicFuste": true,
      "probPodaInadecuada": true,
      "probHeridaTalloRamas": true,
      "probRamasQuebradas": true,
      "rsgoCaida": "Bajo",
      "rsgoAfeccionConstr": "Bajo",
      "rsgoInterfCirculacion": "Bajo",
      "rsgoInfraestructura": "Bajo",
      "rsgoRedesElectricas": "Bajo",
      "recomEliminEpifitas": true,
      "recomDanosMecanicos": true,
      "recomElimObjExtranos": true,
      "recomDerribo": true
    },
    {
      "id": "0014bd00-ce40-444c-8812-edfa1dc71ae3",
      "recorded": "Ronny Santiago Loján Díaz y Karla Briggette Chuquimarca Montalván",
      "siteName": "Parque Lineal La Tebaida",
      "siteType": "Parque",
      "siteParish": null,
      "siteSection": "Estacionamiento interno del parque hasta el Puesto Sendero del parque Lineal Sur",
      "siteAcronym": "PQLT.S1",
      "code": "PQLT.S1.2",
      "specieName": "Platanus occidentalis",
      "familyName": "Platanaceae",
      "pseudonym1": "Sicomoro",
      "pseudonym2": null,
      "pseudonym3": null,
      "pseudonym4": null,
      "dateLog": "2021-08-27T00:00:00.000Z",
      "treeVerified": false,
      "alive": true,
      "commit": "Al lado del sendero sector izquierdo del parque.",
      "photo": null,
      "altitud": "2070",
      "longX": "-79.2025",
      "latY": "-4.0178",
      "cap": "37.40",
      "dap": "11.91",
      "ht": "8.87",
      "hc": "3.06",
      "copaNS": "2.40",
      "copaEW": "2.40",
      "follaje": "15",
      "madurez": "Viejo",
      "floracion": "Sin floración",
      "fructificacion": "Sin fructificación",
      "espaciamiento": "Amplio",
      "fusteRectitud": "Recto",
      "calidadRaiz": "Buena",
      "calidadFuste": "Bueno",
      "calidadCorteza": "Buena",
      "calidadRamas": "Buenas",
      "calidadHojas": "Buenas",
      "calidadCima": "Buena",
      "calidadCopa": "Poco simétrica",
      "probPodaInadecuada": true,
      "rsgoCaida": "Bajo",
      "rsgoAfeccionConstr": "Bajo",
      "rsgoInterfCirculacion": "Bajo",
      "rsgoInfraestructura": "Bajo",
      "rsgoRedesElectricas": "Bajo",
      "recomPoda": true,
      "id": "001ee10e-d47b-4c2c-b8e8-51903fc5bc7c",
      "recorded": "Marcia Janina Loaiza Sanchez",
      "siteName": "Parque Recreacional Jipiro",
      "siteType": "Parque",
      "siteParish": null,
      "siteSection": "Calle Velasco Ibarra",
      "siteAcronym": "PQJV.S8",
      "code": "PQJV.S8.3",
      "specieName": "Salix humboldtiana",
      "familyName": "Salicaceae",
      "pseudonym1": "Sauce común",
      "pseudonym2": null,
      "pseudonym3": null,
      "pseudonym4": null,
      "dateLog": "2022-01-13T00:00:00.000Z",
      "treeVerified": false,
      "alive": true,
      "commit": "A 4 m diagonal al letrero como usar los equipos de gimnasia.",
      "photo": null,
      "altitud": "2060",
      "longX": "-79.20376947",
      "latY": "-3.97116134",
      "dap": "66.95",
      "ht": "22.22",
      "hc": "5.73",
      "cap": "209.01",
      "copaEW": "20.96",
      "follaje": "98",
      "copaNS": "13.92",
      "floracion": "Sin floración",
      "madurez": "Viejo",
      "fructificacion": "Sin fructificación",
      "fusteRectitud": "Inclinado",
      "espaciamiento": "Estrecho",
      "calidadRaiz": "Buena",
      "calidadFuste": "Regular",
      "calidadCorteza": "Buena",
      "calidadRamas": "Buenas",
      "calidadHojas": "Buenas",
      "calidadCima": "Buena"
    }
  ]
}
```

Anexo 6. Pruebas de navegador



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Pruebas a navegadores más comunes

Proyecto: “Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”.

Autor:

Johnny Fabián González Guamán

1. Navegador Chrome

Longitud: -79.1989 | Latitud: -4.0224 | Zoom: 13.23

ACCIONES

- EDITAR
- DESPLAZAR
- ELIMINAR

Habilitar edición. NUEVO

IDENTIDAD DEL ÁRBOL

Código	Familia	Especie
CCGO.1219	Mimosaceae	Acacia retinoides

Nombre común: Acacia verde, Acacia Roja

Parróquia: No definido | Tipo de sitio: No definido

Sitio: Rio Malacatos | Acronimo de Sitio: CCGO

Sección: Calle Chile hasta calle Gonzanamá | Estado actual: Vivo

Estado de registro: No verificado | Nombre del colector: Josué Armijos y Leonardo Granda

Comentario: Frente al estacionamiento del Kiwi. | Fecha de registro: 09/10/2021

ESTADO NATURAL

Altitud	Longitud	Latitud
2086	-79.20304656028748	-4.043984304903418
Circunferencia de fuste	Diámetro	Altura total
29.50	9.39	6.00

Figura 93. Prueba de navegador Chrome versión 109.0.5414.119

2. Navegador Firefox

Longitud: -79.2076 | Latitud: -4.0020 | Zoom: 14.00

ACCIONES

- EDITAR
- DESPLAZAR
- ELIMINAR

Habilitar edición. NUEVO

IDENTIDAD DEL ÁRBOL

Código	Familia	Especie
PQCE.S1.4319	Proteaceae	Fraxinus excelsior

Nombre común: Grevillea

Parróquia: No definido | Tipo de sitio: Parque

Sitio: Parque Central | Acronimo de Sitio: PQCE.S1

Sección: Calle Bolívar hasta Calle Bernardo Valdivieso

Estado actual: Vivo | Estado de registro: No verificado

Nombre del colector: Brayan Guarnizo

Comentario: Frente a la iglesia. | Fecha de registro: 26/08/2021

Figura 94. Prueba de navegador Firefox versión 109.0.1

3. Navegador Safari

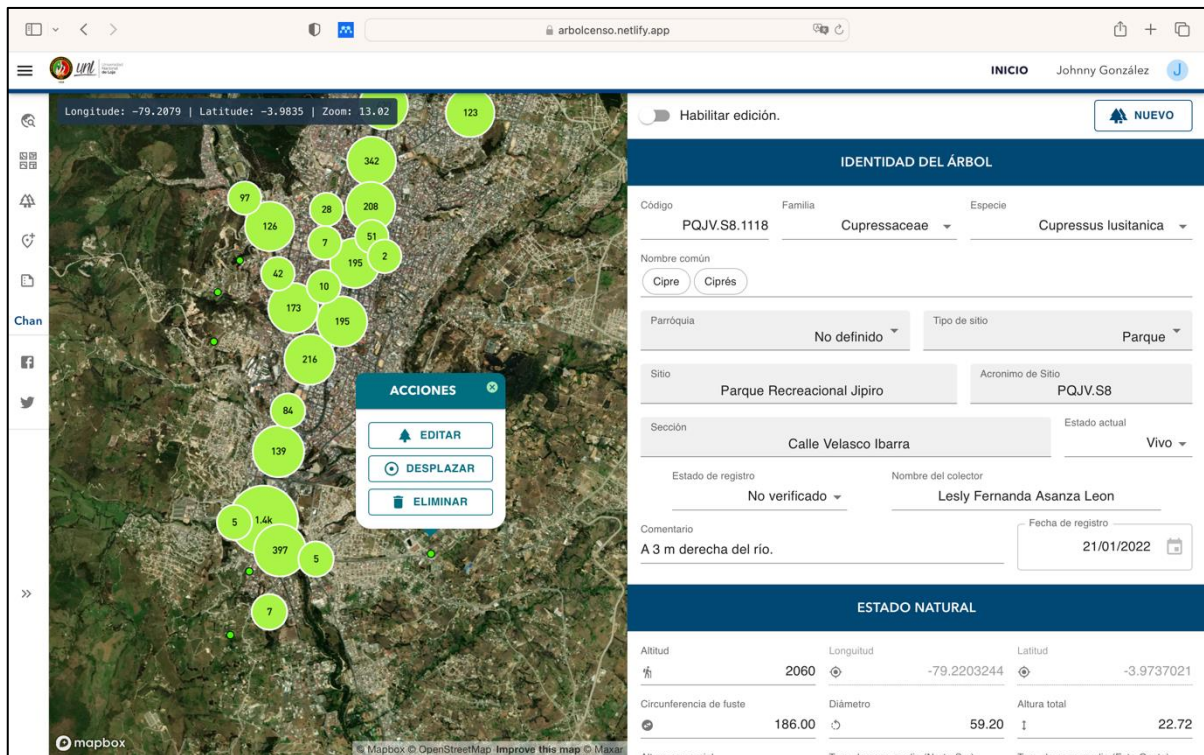


Figura 95. Prueba de navegador Safari versión 16.2

4. Navegador Opera

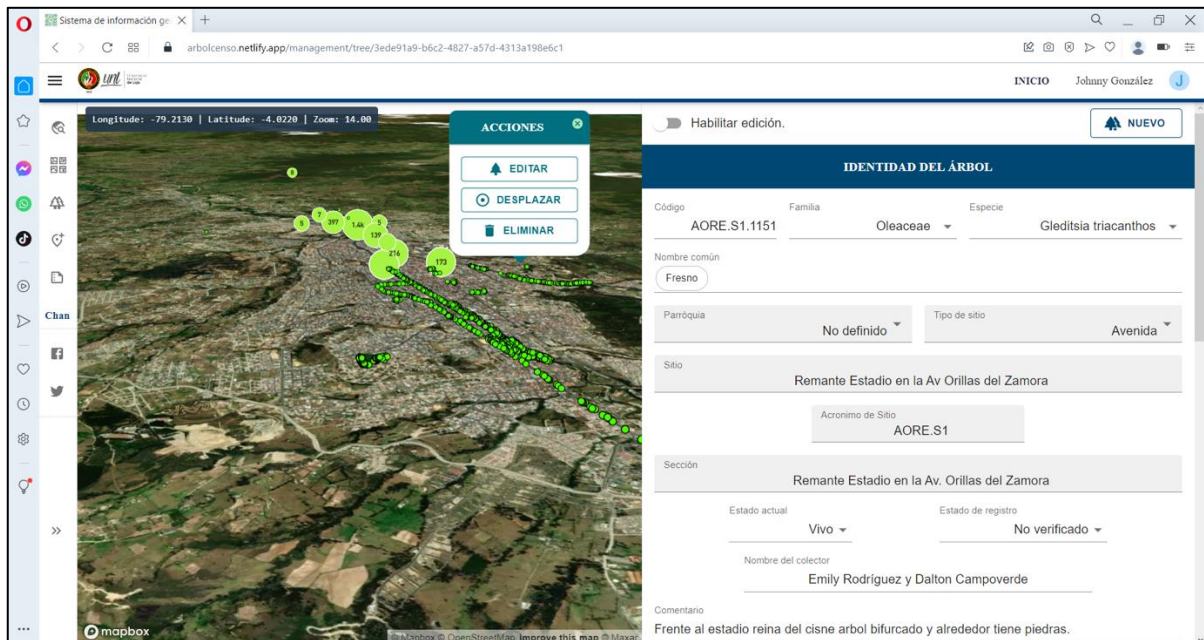


Figura 96. Prueba de navegador Opera versión 56.03051.104

5. Navegador Edge

The screenshot displays a web application interface for tree management. The left side features a satellite map with numerous green circular markers, each labeled with a number (e.g., 7, 5, 397, 148, 5, 139, 84, 216, 195, 10, 42, 126, 97, 2, 195, 7, 51, 28, 208, 342, 430, 50, 123). A context menu titled "ACCIONES" is overlaid on the map, containing three options: "EDITAR", "DESPLAZAR", and "ELIMINAR". The right side of the interface is a form titled "IDENTIDAD DEL ÁRBOL" with the following fields and values:

- Habilitar edición:** **NUEVO** (button)
- Código:** AEJE.724
- Familia:** Bignoniaceae
- Especie:** Leucaena leucocephala
- Nombre común:** Arabisco
- Parróquia:** No definido
- Tipo de sitio:** Avenida
- Sitio:** Río Zamora
- Acronimo de Sitio:** AEJE
- Sección:** Av. Emiliano Ortega y José Egas
- Estado actual:** Vivo
- Estado de registro:** No verificado
- Nombre del colector:** Gina Quizhpe
- Comentario:** Ubicado a 3 metros de la Av. Emiliano Ortega frente a la casa N° 203-81.
- Fecha de registro:** 10/06/2022

The browser's address bar shows the URL: <https://arbolcenso.netlify.app/management/tree/27441949-87f6-40f1-abdf-d047de5dba40>. The browser title is "Sistema de información geográ...". The browser's user interface includes a search bar with "INICIO" and "Johnny González", and a "mapbox" logo at the bottom left of the map area.

Figura 97. Prueba de navegador Edge versión 109.0.1518.70

Anexo 7. Pruebas de Caja Negra



1859



Universidad
Nacional
de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Pruebas de Caja Negra

	Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja
Investigador	Ing. Darwin Pucha Cofrep
Cargo	Director del Proyecto
Fecha	Viernes 20 de mayo de 2022
Objetivo	Validar las funcionalidades con las pruebas de caja negra
Encargado de revisión	Johnny Fabián González Guamán
Firma Encargado	
Firma Investigador	

Pruebas de Caja Negra

Las pruebas de caja negra se llevaron a cabo tomando en cuenta la descripción de los Casos de Uso especificados para la aplicación (**ver Anexo 4, FASE 2**) y fueron ejecutadas por el Ing. Darwin Pucha Cofrep, Director del Proyecto Forestal **UNL-17-DI-FARNR-2021** que evaluó las pruebas de caja negra evidenciando las funcionalidades que había solicitado y aprobando cada caso de prueba. Posteriormente, realizó las pruebas de funcionalidad (**ver Anexo 8**) a través de la ejecución de los flujos normales de eventos en la aplicación.

En esta sección se muestran las tablas corresponden a las pruebas de Caja Negra:

Tabla 61. Caso de prueba - Visualizar Árboles.

PCN01	Caso de uso: Visualizar árboles	Aprobado		Observaciones
		Sí	No	
	Descripción de la prueba			
	Mostrar un mapa con estilo de imágenes satelital.	X		
	Centrar el mapa en longitud, latitud y zoom para visualizar la ciudad de Loja.	X		
	Visualizar los puntos de ubicación de los árboles agrupados en el mapa.	X		
	Al hacer clic derecho sostenido en el mapa y desplazar hacia cualquier posición dentro del mapa, actualizar/cargar la vista del mapa faltante.	X		
	Al hacer clic izquierdo sobre el mapa y desplazarlo, cambiar la orientación de la vista del mapa y su inclinación.	X		
	Actualizar información en el cuadro superior izquierdo, sobre la longitud, latitud y zoom al navegar en el mapa.	X		
	Al hacer clic derecho en un grupo de árboles, debe desagrupar y aumentar el zoom del mapa.	X		
	Al aumentar o disminuir el zoom en el mapa, desagrupar o agrupar uniformemente de acuerdo rango de zoom aplicado (12-13-14)	X		
	Limitar el zoom mínimo aplicable a nivel 12 (altura).	X		

Tabla 62. Caso de prueba – Mostrar mapas

PCN02	Caso de uso: Mostrar mapas	Aprobado		Observaciones
		Sí	No	
	Descripción de la prueba			
	Mostrar opciones para cambiar estilos de mapas en la barra de menú lateral izquierda.	X		
	Con la barra de menú colapsada, en la opción Capa Base, mostrar cuadro de opciones modal de los estilos de mapa.	X		
	Con la barra de menú expandida, en la opción Capa Base, desplegar en el menú las opciones de estilos de mapa.	X		
	Cambiar entre los estilos de mapas disponibles, desde el menú expandido y colapsado.	X		
	Mantener el zoom en el mapa al cambiar el estilo del mapa	X		
	Mantener el estilo de mapa al cambia de rutas en la aplicación.	X		

Tabla 63. Caso de prueba – Filtrar árboles

PCN03	Caso de uso: Filtrar árboles	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
Al seleccionar “Filtrar Árboles” del menú, mostrar el componente de filtros para especies, sitio y características.		X		
Al seleccionar filtros de árboles por especie, mostrar un cuadro de lista desplegable con los nombres científicos.		X		
Al filtrar búsquedas por sitio, mostrar un cuadro de lista desplegable con los nombres de los sitios disponibles.		X		
Al filtrar búsquedas por características, mostrar cinco cuadros con opciones de listas desplegables para Madurez, Fuste, Espaciamiento, Floración, Fructificación.		X		
Agregar o quitar los filtros uno a uno de la lista de opciones disponible.		X		
Clic botón “Aplicar” para mostrar resultados en mapa.		X		
Mostrar un mensaje del número actual árboles.		X		
Quitar todos los filtros al hacer clic en el ícono de limpiar.		X		

Tabla 64. Caso de prueba – Visualizar reportes

PCN04	Casos de uso: Visualizar reportes	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
Al hacer clic sobre un punto de interés en el mapa, abrir una ventana emergente con información básica del árbol.		X		
Al desplazarse en el mapa, la ventana emergente abierta deberá seguir el punto de ubicación del árbol seleccionado.		X		
Al seleccionar otros puntos de interés en el mapa, con la ventana emergente abierto, actualizar posición de la ventana emergente e información.		X		
Mostrar el código QR al hacer clic sobre el ícono de CodeQR ubicado parte inferior derecho de la ventana emergente.		X		
Descargar código QR en los formatos png, jpeg, webp.		X		
Al dar clic en el botón “Ver más”, mostrar un reporte completo del estado del árbol en un panel lateral derecho.		X		
Cerrar el panel de información del árbol.		X		
Cerrar ventana emergente del árbol seleccionado.		X		
En caso de aplicar filtros y se encuentren abiertos la ventana emergente o el panel lateral derecho, estos deben cerrarse.		X		

Tabla 65. Caso de prueba – Autenticar usuario

PCN05	Casos de uso: Autenticar usuario	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
Mostrar el formulario de inicio de sesión a través del botón Login.		X		
Validar que las credenciales de acceso hayan sido ingresadas.		X		
Validar el formato del campo email este correcto.		X		
Validar las credenciales ingresadas, mostrar mensaje “Usuario o contraseña no tienen acceso” si las credenciales de acceso son incorrectas o no existen.		X		
Mostrar mensaje si la cuenta existe (email registrado) y no está activa (validación de cuenta pendiente).		X		
Mostrar al final del formulario las opciones para recuperación de contraseña y creación de cuenta.		X		
Permitir el acceso a las funcionalidades de la aplicación de acuerdo a rol del usuario autenticado.		X		

Tabla 66. Caso de prueba – Resetear contraseña

PCN06	Casos de uso: Resetear Contraseña	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
Mostrar el formulario de recuperación de contraseña a través del enlace “¿Olvidaste tu contraseña?”.		X		
Validar que el formulario no admita el campo email vacío.		X		
Validar el formato del campo email este correcto.		X		
Validar si el email ingresado existe. Mostrar un mensaje de “Usuario no se encuentra registrado” si email no existe.		X		
Si email es válido, mostrar mensaje que se ha enviado las instrucciones al email ingresado.		X		
Enviar instrucciones de recuperación de contraseña al correo ingresado.		X		
Validar los campos del formulario restablecer contraseña no estén vacíos.		X		
Validar campo “Contraseña” contenga al menos una mayúscula, una minúscula, un número y contenga al menos 8 caracteres y máximo 16 caracteres.		X		
Validar que el campo “Confirmar contraseña” coincida con el campo “Contraseña”.		X		
Mostrar mensaje de contraseña actualizada.		X		

Tabla 67. Caso de prueba – Cambiar contraseña

PCN07	Casos de uso: Cambiar contraseña	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
Mostrar el formulario “Actualizar Datos” al hacer clic en “Mi Cuenta” en el icono de la sección de configuración de cuenta de la barra superior.		X		
Validar que los campos “nombre” y “apellido” no se admitan campos vacíos.		X		
Validar el formato del campo email este correcto.		X		
Si no se ingresa el campo contraseña mantener la misma contraseña.		X		
Si se ingresa el campo contraseña, validar campo contraseña contenga al menos una mayúscula, una minúscula, un número y contenga al menos 8 caracteres y máximo 16 caracteres.		X		
Validar que los campos “Contraseñas” y “Confirmar contraseña”, coincidan.		X		
Al actualizar, mostrar mensaje de Guardado.		X		

Tabla 68. Caso de prueba – Registrar usuario

PCN08	Casos de uso: Registrar usuarios	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
Mostrar el formulario de registro al hacer clic en “Crear cuenta” en el formulario de inicio de sesión.		X		
Validar que no se admitan campos vacíos en el formulario.		X		
Validar el formato del campo email este correcto.		X		
Validar campo contraseña contenga al menos una mayúscula, una minúscula, un número y contenga al menos 8 caracteres y máximo 16 caracteres.		X		
Validar que los campos Contraseña y Confirmar contraseña, coincidan.		X		

Validar que correo electrónico no este registrado, mostrar mensaje de aviso, que el correo electrónico ya se encuentra registrado.	X		
Mostrar un mensaje al terminar de registrarse, que la cuenta se encuentra pendiente/en revisión.	X		

Tabla 69. Caso de prueba – Validar cuenta

PCN09	Casos de uso: Validar cuenta	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
Mostrar el formulario de autorizar usuario, al hacer clic en el enlace envió al correo del administrador.		X		
Habilitar cuenta del usuario registrado.		X		
Mostrar un mensaje de cuenta de usuario activada o habilitada.		X		

Tabla 70. Caso de prueba – Editar árbol

PCN10	Casos de uso: Editar árbol	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
En la página de "Administrar registros", al seleccionar un árbol del mapa, mostrar una ventana emergente con las acciones disponibles según el tipo del rol del usuario.		X		
Mostrar el formulario de Edición, al seleccionar el botón "Editar".		X		
Al seleccionar la acción "Desplazar", si el formulario de edición no ha sido mostrado con anterioridad, direccionar al formulario de edición de árbol, caso contrario, mostrar un marcador para el desplazamiento de ubicación en el mapa.		X		
El switch "Habilitar edición" o "Deshabilitar edición" permite o no, editar todos campos del formulario. A excepción de código de árbol, fecha de registro, latitud y longitud.		X		
Validar que los campos sean obligatorios en la sección de "Identidad del árbol" del formulario, a excepción del campo familia.		X		
Al seleccionar una familia, mostrar en el campo "Especie", la lista filtrada de los nombres científicos correspondiente a esa familia seleccionada.		X		
Al seleccionar una especie, mostrar en el campo "Nombre Común", la lista filtrada de los seudónimos correspondientes a la especie seleccionada.		X		
En el campo "Nombre común" seleccionar de la lista o agregar nuevos nombres al hacer 'Enter'.		X		
Si el campo "Acrónimo de ubicación" no se encuentra registrado o no corresponden a una ubicación registrada para los campos: ("Parroquia", "Sitio", "Tipo de sitio" y "Sección") mostrar un mensaje (Crear nuevo acrónimo) al momento de Guardar.		X		
Validar el formato de los campos en la sección de "Estado Natural" del formulario. Los campos no son obligatorios, a excepción de "Longitud" y "Latitud" que se encuentran deshabilitados para la edición textual.		X		
Si la acción "Desplazar" fue habilitada, desplazar el marcador del mapa hacia una nueva ubicación.		X		
Capturar ubicación (longitud, latitud) del marcador en el formulario, al presionar el botón "Capturar ubicación".		X		
En las pestaña 1 y 7 del apartado "Condición General", los campos son de tipo selección simple. Y las 2, 3, 4, 5, 6 y 8 son de selección múltiple.		X		

En las pestañas 2, 3, 4, 5, 6 y 8 (correspondientes a las Opciones múltiples) fijar los elementos antes de cambiar de pestaña o antes de presionar el botón guardar.	X		
Al presionar "Guardar", presentar un mensaje de guardado y actualizar el formulario, caso contrario presentar un mensaje del estado de validación.	X		

Tabla 71. Caso de prueba – Registrar Árbol

PCN11	Casos de uso: Registrar árbol	Aprobado		Observaciones
		Sí	No	
	Descripción de la prueba			
	En la página de "Registrar árbol", al seleccionar un árbol del mapa, mostrar el marcador en el mapa.	X		
	Habilitar el desplazamiento de la ubicación actual del marcador, al presionar el botón sobre el marcador (Botón "Ubicación actual/Reubicar árbol")	X		
	Al desplazar el marcador, activar el botón "Capturar ubicación"	X		
	Presionar "Capturar ubicación" en el mapa, capturar la longitud y latitud en los campos del formularios respectivos.	X		
	Validar que los campos sean obligatorios en la sección de "Identidad del árbol" del formulario, a excepción del campo familia.	X		
	Al seleccionar una familia, mostrar en el campo "Especie", la lista filtrada de los nombres científicos correspondiente a esa familia seleccionada.	X		
	Al seleccionar una especie, mostrar en el campo "Nombre Común", la lista filtrada de los seudónimos correspondientes a la especie seleccionada.	X		
	En el campo "Nombre común" seleccionar de la lista o agregar nuevos nombres al hacer 'Enter'.	X		
	Si el campo "Acrónimo de ubicación" no se encuentra registrado o no corresponden a una ubicación registrada para los campos: ("Parroquia", "Sitio", "Tipo de sitio" y "Sección") mostrar un mensaje (Crear nuevo acrónimo) al momento de Guardar.	X		
	Validar el formato de los campos en la sección de "Estado Natural" del formulario. Los campos no son obligatorios, a excepción de "Longitud" y "Latitud" que se encuentran deshabilitados para la edición textual.	X		
	En las pestaña 1 y 7 del apartado "Condición General", los campos son de tipo selección simple. Y las 2, 3, 4, 5, 6 y 8 son de selección múltiple.	X		
	En las pestañas 2, 3, 4, 5, 6 y 8 (correspondientes a las Opciones múltiples) fijar los elementos antes de cambiar de pestaña o antes de presionar el botón guardar.	X		
	Al presionar "Guardar", presentar un mensaje de guardado y actualizar el formulario, caso contrario presentar un mensaje del estado de validación.	X		

Tabla 72. Caso de prueba – Eliminar árbol

PCN12	Casos de uso: Eliminar árbol	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
En la página de "Administrar árbol", al seleccionar un árbol del mapa, mostrar las acciones disponibles para el usuario administrador.		X		
Al seleccionar "Eliminar", mostrar ventana de confirmación.		X		
Cancelar acción, cierra la ventana de confirmación.		X		
Confirmar Eliminación, eliminar punto de ubicación del árbol y redireccionar a la página principal de "Administrar árbol".		X		

Tabla 73. Caso de prueba – Descargar reporte de árboles

PCN13	Casos de uso: Descargar reporte de árboles	Aprobado		Observaciones
		Sí	No	
Descripción de la prueba				
Ir a la página de Generar reporte al hacer clic en la opción del menú Generar reporte (solo Admin)		X		
Mostrar una ventana de loading, mientras se procesa la solicitud para generar la tabla con los datos actualizados.		X		
Mostrar los datos cargados en la tabla.		X		
Mostrar opción para Filtrar los resultados de la tabla usando los filtros disponibles.		X		
Mostrar opción para mostrar columnas que se desean mantener visibles en la tabla.		X		
Mostrar opción para descargar el reporte en formato CSV.		X		

Anexo 8. Pruebas Funcionales



Universidad
Nacional
de Loja



Carrera de Ingeniería en
Sistemas / Computación

UNIVERSIDAD NACIONAL DE LOJA

Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables
Carrera de Ingeniería en Sistemas / Computación

Proyecto: “Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”.

Desarrollador: Sr. Johnny Fabián González Guamán

Producto a evaluar: Requerimientos funcionales de la aplicación web.

Fecha de la revisión: Martes, 07 de febrero de 2023

Requerimientos Funcionales

Id	Requisito	Estado
RF01	Visualizar árboles	Realizado
RF02	Mostrar tipos de mapas base	Realizado
RF03	Visualizar reportes	Realizado
RF04	Filtra consultas	Realizado
RF05	Autenticar usuario	Realizado
RF06	Actualizar contraseña	Realizado
RF07	Recuperar contraseña	Realizado
RF08	Editar árboles	Realizado
RF09	Registrar árbol	Realizado
RF10	Eliminar árbol	Realizado
RF11	Descargar reportes	Realizado
RF12	Registrar cuenta	Realizado
RF13	Validar cuenta	Realizado

F. _____

Dr. Darwin Pucha Cofrep
Director del Proyecto
UNL-17-DI-FARNR-2021

Figura 98. Prueba de requerimientos funcionales

Anexo 9. Presentación de Poster del Trabajo de Titulación en Casa Abierta, Ingeniería Forestal - UNL



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

Presentación en Casa Abierta de Ingeniería Forestal, UNL

Proyecto: “Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”.

Autor:

Johnny Fabián González Guamán

APLICACIÓN WEB DE LOS ÁRBOLES URBANOS DE LA CIUDAD DE LOJA

Johnny González¹, Darwin Pucha Cofrep², Wilman Chamba³

jfgonzalezg@unl.edu.ec - darwin.pucha@unl.edu.ec - wpchamba@unl.edu.ec

1. Proyecto 17-DI-FARNR-2021: Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja
2. Facultad agropecuaria y de recursos naturales renovables, Carrera de Ingeniería Forestal.
3. Facultad de la energía, las industrias y los recursos naturales no renovables. Carrera de Computación.

INTRODUCCIÓN

En la Universidad Nacional de Loja, las TIC son una importante herramienta para los proyectos de investigación internos. En particular, para el Proyecto de investigación 17-DI-FARNR-2021 de la Carrera de Ingeniería Forestal, que busca mostrar en un mapa los resultados de datos dasométricos, morfológicos, y ecológicos de cada árbol urbano de la ciudad. Para ello, el diseño de una aplicación Web es esencial para llevar a cabo este proyecto.

OBJETIVOS

- Establecer el diseño de la Aplicación Web en base a la metodología ICONIX.
- Implementar el diseño de la aplicación mediante el framework ReactJS y NodeJS.
- Ejecutar pruebas de funcionalidad para la validación de la Aplicación Web.

METODOLOGÍA

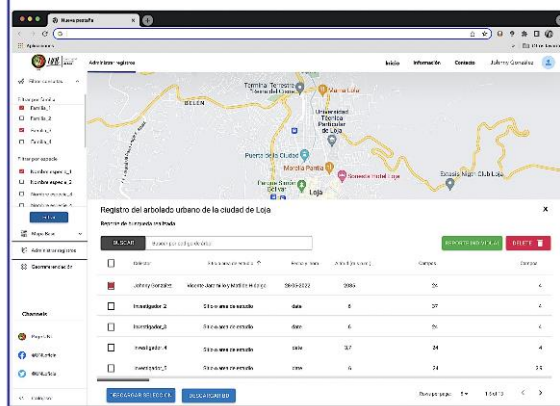
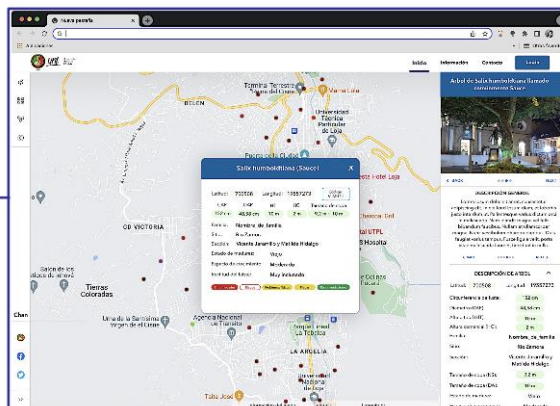
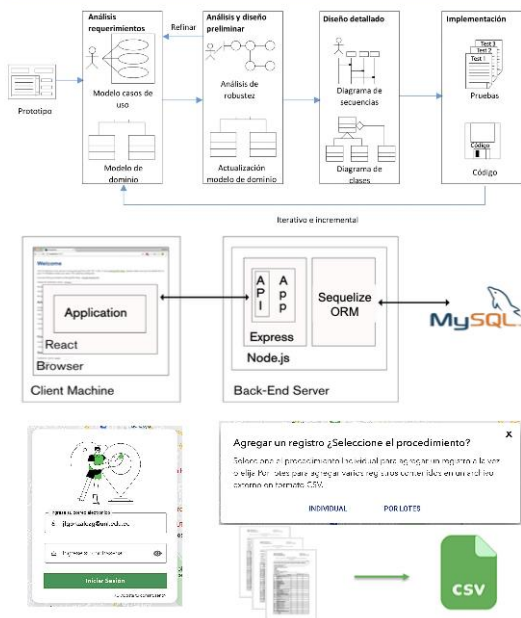


Figura 99. Diseño de Poster Científico para Casa Abierta, Ingeniería Forestal, UNL.

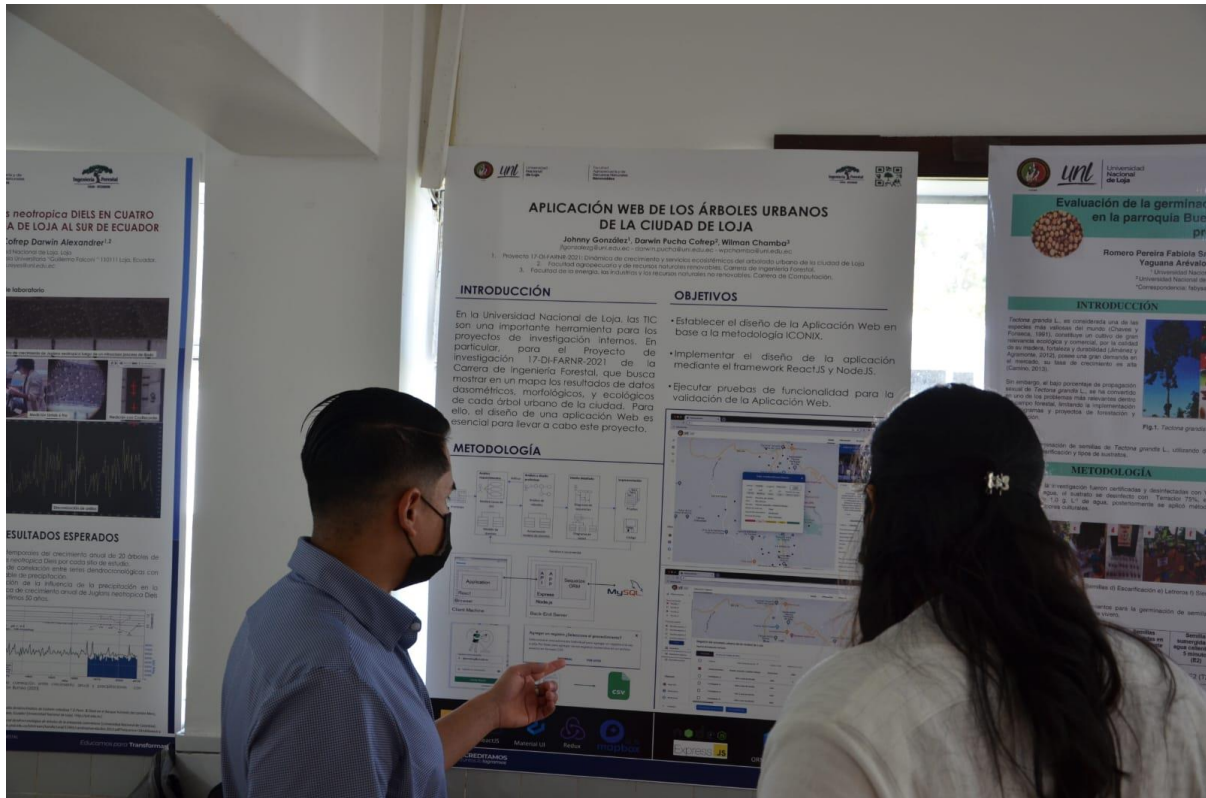


Figura 100. Exposición de poster en Casa Abierta, Ingeniería Forestal-UNL.

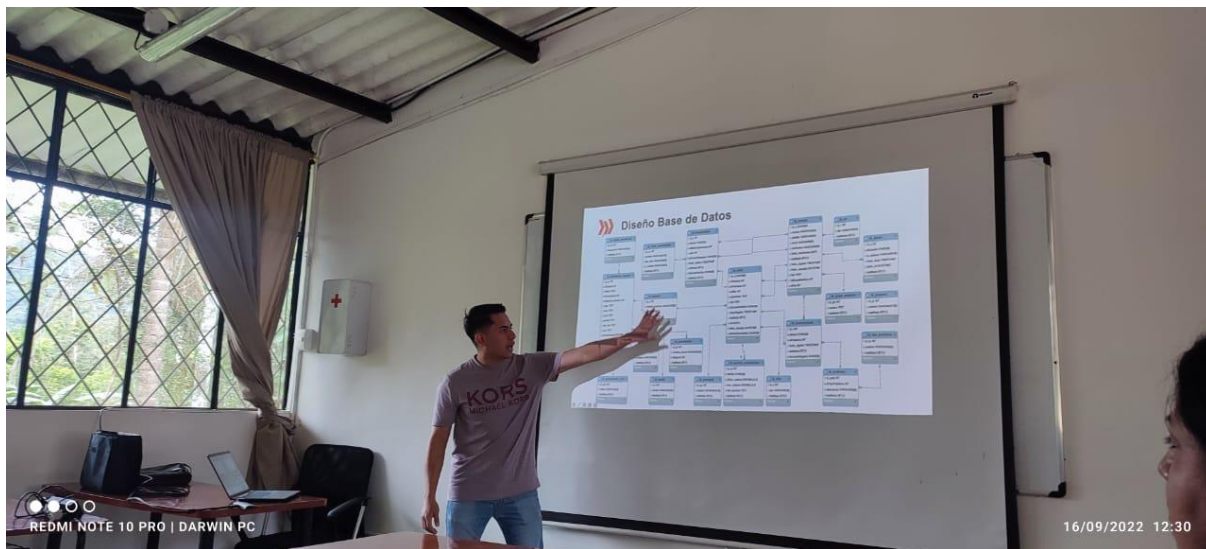


Figura 101. Socialización de aspectos técnicos de la aplicación web.

Anexo 10. Acuerdo de confidencialidad y no divulgación de la base de datos del proyecto 17-DI-FARNR-2021: Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja.

ACUERDO DE CONFIDENCIALIDAD Y NO DIVULGACIÓN DE LA BASE DE DATOS DEL PROYECTO 17-DI-FARNR-2021: DINÁMICA DE CRECIMIENTO Y SERVICIOS ECOSISTÉMICOS DEL ARBOLADO URBANO DE LA CIUDAD DE LOJA

Intervienen en la celebración del presente *"ACUERDO DE CONFIDENCIALIDAD Y USO DE BASE DE DATOS"*, por una parte, **DARWIN ALEXANDER PUCHA COFREP** con cédula de identidad Nro. **1104031529**, en mi calidad de DIRECTOR DEL PROYECTO 17-DI-FARNR-2021 en adelante y para efectos del presente instrumento en representación de la UNIVERSIDAD NACIONAL DE LOJA en calidad de **PROVEEDOR DE LA BASE DE DATOS**; y por otro lado **JOHNNY FABIÁN GONZÁLEZ GUAMÁN** con cédula de identidad Nro. **1150036125** en mi calidad de TESISTA perteneciente a la CARRERA INGENIERÍA EN SISTEMAS de la UNIVERSIDAD NACIONAL DE LOJA, en adelante y para efectos del presente instrumento en calidad de **RECEPTOR DE LA BASE DE DATOS** quienes libre y voluntariamente celebran el presente acuerdo.

Ambas partes reconocen recíprocamente su capacidad para obligarse, por lo que suscriben el presente Acuerdo de Confidencialidad y No Divulgación de la Base de Datos con base a las siguientes cláusulas.

CLÁUSULA PRIMERA. - ANTECEDENTES:

El artículo 226 de la Constitución de la República del Ecuador prevé que: *"Las instituciones del Estado sus organismos y dependencias, y las servidoras o servidores públicos, tendrán el deber de coordinar acciones para el cumplimiento de sus fines y hacer efectivo el goce y ejercicio de los derechos reconocidos en la Constitución"*;

En virtud de lo establecido en el numeral 19 del artículo 66 de la Norma Suprema se dispone: *"Se reconoce y garantizará a las personas: (...) El derecho a la protección de datos de carácter personal, que incluye el acceso y la decisión sobre información y datos de este carácter, así como su correspondiente protección. La recolección, archivo, procesamiento, distribución o difusión de estos datos o información requerirán la autorización del titular o el mandato de la Ley"*;

El artículo 178 del Código Orgánico Integral Penal establece: *"La persona que, sin contar con el consentimiento o la autorización legal, acceda, intercepte, examine, retenga, grabe, reproduzca, difunda o publique datos personales, mensajes de datos, voz, audio y vídeo, objetos postales, información contenida en soportes informáticos, comunicaciones privadas o reservadas de otra persona por cualquier medio, será sancionada con pena privativa de libertad de uno a tres años..."*;

El artículo 190 *ibídem* señala: *“La persona que utilice fraudulentamente un sistema informático o redes electrónicas y de telecomunicaciones para facilitar la apropiación de un bien ajeno o que procure la transferencia no consentida de bienes, valores o derechos en perjuicio de esta o de una tercera, en beneficio suyo o de otra persona alterando, manipulando o modificando el funcionamiento de redes electrónicas, programas, sistemas informáticos, telemáticos y equipos terminales de telecomunicaciones, será sancionada con pena privativa de libertad de uno a tres años(...)”*;

El artículo 230 del Código Orgánico Integral Penal determina: *“Será sancionada con pena privativa de libertad de tres a cinco años: (...) La persona que sin orden judicial previa, en provecho propio o de un tercero, intercepte, escuche, desvíe, grabe u observe, en cualquier forma un dato informático en su origen, destino o en el interior de un sistema informático, una señal o una transmisión de datos o señales con la finalidad de obtener información registrada o disponible. (...)”*;

La Ley de Comercio Electrónico, Firmas Electrónicas y Mensajes de Datos, en los artículos 2 y 44, respectivamente, reconoce ante el Estado la validez jurídica de los mensajes de datos electrónicos, así como el valor y efecto jurídicos de cualquier actividad, transacción mercantil, financiera o de servicios que se realice con los mismos por medio de redes electrónicas;

La Carta Iberoamericana de Gobierno Electrónico, en la sección 24, recomienda a los gobiernos tomar en consideración la importancia de la interoperabilidad de las comunicaciones y servicios, así como disponer las medidas necesarias, para que todas las entidades públicas, cualquiera que sea su nivel y con independencia del respeto a su autonomía, establezcan sistemas que sean interoperables;

La Ley del Sistema Nacional de Registro de Datos Públicos publicada en el Registro Oficial No. 162 de 31 de marzo de 2010, en su artículo 4, cita: *“Las instituciones del sector público y privado y las personas naturales que actualmente o en el futuro administren bases o registros datos públicos, son responsables de la integridad, protección y control de los registros y bases de datos a su cargo. Dichas instituciones responderán por la veracidad, autenticidad, custodia y debida conservación de los registros. La responsabilidad sobre la veracidad y autenticidad de los datos registrados, es exclusiva de la o el declarante cuando esta o este provee toda la información...”*;

El artículo 27 de la Ley *ibídem* establece: *“Las Registradoras o Registradores y máximas autoridades, a quienes se autoriza el manejo de las licencias para el acceso a los registros de datos autorizados por la ley, serán las o los responsables directos administrativa, civil y penalmente por el mal uso de las mismas”*;

El artículo 14 del Reglamento de Régimen Académico de la UNL, determina que la Falta de ética y deshonestidad académica se refiere a *“toda acción que, inobservado el principio de transparencia académica, viola los derechos de autor o incumple la normas éticas establecidas por la Universidad Nacional de Loja”* y el literal i) *“Otras que contravengan la Ley Orgánica de Educación Superior, Reglamento de Régimen Académico del Consejo de Educación Superior, el Código de ética de la Universidad Nacional de Loja y normativa aplicable vigente”*.

Bajo este marco regulatorio, la información que se dispone en el PROYECTO 17-DI-FARNR-2021: DINÁMICA DE CRECIMIENTO Y SERVICIOS ECOSISTÉMICOS DEL ARBOLADO URBANO DE LA CIUDAD DE LOJA se clasifica como reservada y confidencial, a tal efecto se acuerda suscribir el presente acuerdo entre el Proveedor y Receptor de la Base de Datos con la finalidad de proteger la información que se utilice para fines investigativos, con la finalidad de garantizar la confidencialidad, integridad, disponibilidad, reserva y protección de los datos e información que se será proporcionada por el proveedor responsable del proyecto.

CLÁUSULA SEGUNDA. - OBJETO:

Por medio del presente instrumento el **RECEPTOR DE LA BASE DE DATOS** se obliga expresamente a guardar sigilo, confidencialidad y reserva sobre el contenido de toda la información generada, verbal o escrita, que se comparta entre las partes.

Además, se compromete a hacer uso de la información, únicamente para las actividades relacionadas con la investigación, enseñanza, publicaciones, gestión y planificación sin fines de lucro conforme a las obligaciones y prohibiciones legales pertinentes.

CLÁUSULA TERCERA. - DERECHOS Y OBLIGACIONES DE LAS PARTES:

Son obligaciones del **Proveedor de la base de datos** del proyecto 17-DI-FARNR-2021: DINÁMICA DE CRECIMIENTO Y SERVICIOS ECOSISTÉMICOS DEL ARBOLADO URBANO DE LA CIUDAD DE LOJA las siguientes:

1. Suministrar al **RECEPTOR DE LA BASE DE DATOS** la información que estime necesaria para el desarrollo de las actividades relacionadas con la investigación, enseñanza, publicaciones, gestión y planificación sin fines de lucro, conforme a las obligaciones y prohibiciones legales pertinentes.
2. Velar por el cumplimiento del presente acuerdo.

Por su parte el **RECEPTOR DE LA BASE DATOS** se obligue a:

1. Mantener el compromiso de confidencialidad respecto a la información y material suministrado por el Proveedor de la Base de Datos de forma indefinida.
2. A no revelar, divulgar o facilitar bajo cualquier forma o circunstancia a persona alguna sea natural o jurídica, pública o privada, o de cualquier otra naturaleza, y a no utilizar para su propio beneficio o para beneficio de un tercero, la información suministrada sin la debida autorización del Proveedor de la Base de Datos.
3. A no hacer público de manera oral o escrita y/o tecnológico por cualquier medio de difusión sin la debida autorización la información que será suministrada por el Proveedor de la Base de Datos del Proyecto 17-DI-FARNR-2021: DINÁMICA DE CRECIMIENTO Y SERVICIOS ECOSISTÉMICOS DEL ARBOLADO URBANO DE LA CIUDAD DE LOJA.
4. Informarse, entender, apoyar y cumplir con las normas de seguridad que gobiernan la protección de la base de datos en la UNL.
5. A no reproducir o duplicar la información suministrada, sino cuenta con el visto bueno del Proveedor de la Base de Datos. Se excluye la información que sea de dominio público o que sea de conocimiento previo del proyecto.
6. En el informe final y difusión de resultados se reconozca de manera expresa a la Universidad Nacional de Loja.

CLÁUSULA CUARTA. - PATRÓN DE CONDUCTA, IMPLICACIONES DE LA RECEPCIÓN DE LA INFORMACIÓN Y RESPONSABILIDAD

Las partes actuarán con responsabilidad en el buen uso de la información, lo que supone entre otros deberes, el de limitar la divulgación autorizada al menor número de personas, y el de tomar las medidas idóneas y eficaces para evitar el tráfico y fuga indebida de la información, así como su uso por fuera de los límites de este acuerdo.

El incumplimiento del deber de reserva establecido en la Cláusula Cuarta del presente acuerdo constituye violación de secreto y justa causa de terminación unilateral de la relación, sin desmedro de las indemnizaciones (sólo para proveedores) legales correspondientes.

El **RECEPTOR DE LA BASE DE DATOS** reconoce que la información confidencial a la que se refiere el presente acuerdo posee una valoración en imagen institucional y su indebida divulgación o utilización causa un perjuicio al proyecto en ejecución.

CLÁUSULA QUINTA. - MATERIALES:

Todos los materiales como, documentos, bases de datos, actas de reunión, fichas de campo, imágenes, fotografías, mapas, software, muestras orgánicas entre otras que son entregadas al **RECEPTOR DE LA BASE DE DATOS** por parte del proyecto 17-DI-FARNR-2021: DINÁMICA DE CRECIMIENTO Y SERVICIOS ECOSISTÉMICOS DEL ARBOLADO URBANO DE LA CIUDAD DE LOJA se considera como información confidencial y se debe guardar absoluta reserva de la misma.

CLÁUSULA SEXTA. - SANCIONES:

Para la aplicación de sanciones se tomará en cuenta lo establecido en la Constitución de la República del Ecuador, la Ley Orgánica de Transparencia y Acceso a la Información Pública, Ley Orgánica del Sistema Nacional de Registro de Datos Públicos, Reglamento de Régimen Académico UNL Art.14 en cuanto a la Falta de ética y deshonestidad académica, y demás normativa aplicable; sin perjuicio de las acciones civiles y penales que procedan en cada caso.

CLÁUSULA SÉPTIMA. - VIGENCIA:

El presente instrumento tendrá una vigencia indefinida partir de la fecha de suscripción.

CLÁUSULA OCTAVA - ACUERDO TOTAL:

Este acuerdo incluye el total entendimiento entre las partes con relación a la materia de la cual se trata este documento. Cualquier añadidura o modificación del mismo deberá ser conocido por escrito y firmada por ambas partes.

CLÁUSULA NOVENA. - NOTIFICACIONES:

En el evento de que se produzca el incumplimiento de alguna de las cláusulas estipuladas en el presente acuerdo, la parte afectada, notificará del incumplimiento a la máxima autoridad de la Universidad Nacional de Loja, sin perjuicio de las acciones y sanciones previstas en la normativa vigente.

Una vez comprendido por los comparecientes el contenido y efectos del presente instrumento expresamente se ratifican en él, para fe y constancia se firma el presente documento por quienes en él intervinieron, en la ciudad de Loja, el día 29 del mes de junio del año **2022**, en dos ejemplares del mismo tenor y validez.



.....

Nombres: Darwin Alexander Pucha Cofrep

C.I.: 1104031529

**CARRERA DE INGENIERÍA FORESTAL,
UNIVERSIDAD NACIONAL DE LOJA**

CARGO: DIRECTOR DEL PROYECTO 17-DI-FARNR-2021: DINÁMICA DE CRECIMIENTO Y SERVICIOS ECOSISTÉMICOS DEL ARBOLADO URBANO DE LA CIUDAD DE LOJA



.....

Nombres: Johnny Fabián González Guamán

C.I.: _ _ _ 1150036125 _ _ _ _

**CARRERA INGENIERÍA EN SISTEMAS,
UNIVERSIDAD NACIONAL DE LOJA**

CARGO: TESISTA DEL PROYECTO 17-DI-FARNR-2021: DINÁMICA DE CRECIMIENTO Y SERVICIOS ECOSISTÉMICOS DEL ARBOLADO URBANO DE LA CIUDAD DE LOJA

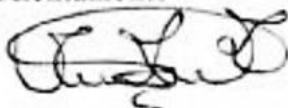
Anexo 11. Certificado de traducción del resumen

Loja, 17 de febrero de 2023.

Yo, Cristian Efrain Flores Pasaca con cédula 1105164907, licenciado en Ciencias de la Educación en el área de Idioma Inglés, certifico haber revisado el proyecto de Investigación de Johnny Fabián González Guamán, con cédula 1150036125, el cual tiene el nombre de “Aplicación web para la gestión de la representatividad de árboles urbanos del proyecto de investigación Dinámica de crecimiento y servicios ecosistémicos del arbolado urbano de la ciudad de Loja”.

Es todo lo que puedo mencionar en honor a la verdad.

Atentamente



Lic. Cristian Efrain Flores Pasaca.