

**DESARROLLO DE UN SISTEMA DE GESTIÓN DE INFORMACIÓN DEL
PROYECTO AGRICULTURA INTELIGENTE DEL CIBIOFI**

ALEXANDER CLAVIJO LEDESMA

**UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERÍA CIVIL Y GEOMÁTICA
2020**

**DESARROLLO DE UN SISTEMA DE GESTIÓN DE INFORMACIÓN DEL
PROYECTO AGRICULTURA INTELIGENTE DEL CIBIOFI**

ALEXANDER CLAVIJO LEDESMA

*Trabajo de grado
presentado como requisito
parcial para optar al título
de ingeniero topográfico*

**DIRECTOR:
MAURICIO EDILBERTO RINCÓN ROMERO, PhD.**

**UNIVERSIDAD DEL VALLE
ESCUELA DE INGENIERÍA CIVIL Y GEOMÁTICA
INGENIERÍA TOPOGRÁFICA
SANTIAGO DE CALI
2020**

PÁGINA DE ACEPTACIÓN

Nota de Aceptación

Presidente del Jurado

Jurado

Jurado

Santiago de Cali, 2020.

AGRADECIMIENTOS

Expreso mi gratitud a la Universidad del Valle y a los docentes que contribuyeron en mi formación académica a lo largo de estos años. A mis padres por su grandísimo e incondicional apoyo en todo momento. A Isabel por su constante motivación. A Carlos Esteban por su colaboración en el desarrollo de software.

Igualmente, agradezco al Centro de Investigación e Innovación en Bioinformática y Fotónica (CIBioFi), del cual fui becario durante el desarrollo de este proyecto. Gracias al Sistema General de Regalías y al Grupo de Investigación GISMODEL, especialmente al director Mauricio Rincón por permitirme hacer parte de este proceso.

TABLA DE CONTENIDO

RESUMEN.....	13
INTRODUCCIÓN.....	14
CAPÍTULO 1	16
1.1. PROBLEMA.....	16
1.1.1. Contextualización	16
1.1.2. Definición del problema	17
1.1.3. Pregunta de investigación	18
1.2. OBJETIVOS	19
1.2.1. Objetivo General	19
1.2.2. Objetivos Específicos	19
1.3. JUSTIFICACIÓN.....	20
CAPÍTULO 2	23
2.1. MARCO TEÓRICO	23
2.1.1. Datos, información y conocimiento	23
2.1.1.1. Bases de datos	24
2.1.1.2. Intercambio de datos	25
2.1.1.3. Curación de datos.....	26
2.1.1.4. Datos abiertos.....	26
2.1.1.5. Metadatos.....	27
2.1.1.6. Datos FAIR	29
2.1.1.6.1. Principios rectores FAIR	29
2.1.2. Sistemas de Información	30
2.1.2.1. Sistemas de Información Basados en Computadora	31
2.1.2.2. Sistemas de administración del conocimiento.....	31
2.1.3. Software de código abierto.....	32
2.1.4. Interfaz de Programación de Aplicaciones	32
2.1.5. Servicios web RESTful.....	33
2.1.5.1. Hypertext Transfer Protocol	33
2.2. MARCO REFERENCIAL	34
2.2.1. Importancia de los datos	34
2.2.2. Diluvio de datos.....	34

2.2.3.	Intercambio de datos	34
2.2.4.	Metadatos	35
2.2.5.	Plataformas web para el intercambio de datos	36
CAPÍTULO 3		39
3.1.	METODOLOGÍA.....	39
3.1.1.	Análisis de sistemas.....	40
3.1.1.1.	Objetivo general del sistema.....	40
3.1.1.2.	Objetivos específicos del sistema	40
3.1.1.3.	Alcance del sistema.....	41
3.1.1.4.	Limitaciones del sistema.....	42
3.1.2.	Análisis de requerimientos	43
3.1.2.1.	Definición conceptual de los requerimientos	43
3.1.2.2.	Esquema para los metadatos	45
3.1.2.3.	Identificación de los módulos del sistema	45
3.1.2.3.1.	Módulo de usuarios	45
3.1.2.3.2.	Módulo de consulta.....	46
3.1.2.3.3.	Seguridad	47
3.1.3.	Diseño del sistema.....	48
3.1.3.1.	Modelo conceptual.....	48
3.1.3.1.1.	Elementos del sistema.....	49
3.1.3.1.2.	Operatividad del sistema	50
3.1.3.2.	Modelo lógico.....	52
3.1.3.3.	Modelo de datos	56
3.1.3.3.1.	Tablas de la base de datos	58
3.1.3.3.2.	Normalización de la base de datos	63
3.1.3.3.3.	Auditoría de la base de datos	64
3.1.3.3.4.	Respaldo de la base de datos.....	66
3.1.3.3.5.	Almacenamiento de los datos de investigación.....	66
3.1.3.4.	Modelo cartográfico	66
3.1.3.5.	Modelo físico.....	68
3.1.3.5.1.	Categorización de la información	68
3.1.3.5.2.	Selección de software.....	69
3.1.3.5.3.	Arquitectura del sistema	70

3.1.4.	Implementación del sistema	72
3.1.4.1.	Organización de la información.....	72
3.1.4.2.	Creación de la base de datos	74
3.1.4.2.1.	Configuración del esquema de metadatos	77
3.1.4.3.	Back-End	81
3.1.4.4.	Front-End.....	82
3.1.4.4.1.	Leaflet.....	86
3.1.4.4.2.	Plantilla de ReactJS.....	86
3.1.4.5.	Integración del Back-End y el Front-End.....	87
3.1.4.6.	Hardware	87
3.1.5.	Pruebas.....	87
CAPÍTULO 4		90
4.1.	RESULTADOS	90
4.1.1.	Estructura de almacenamiento de los datos	90
4.1.1.1.	Archivos con información comprimida.....	90
4.1.1.2.	Base de datos de SILAIN.....	91
4.1.2.	Metadatos	92
4.1.2.1.	Datos FAIR	93
4.1.3.	Gestión de la información en el sistema	94
4.1.4.	Plataforma web para el acceso a datos e información.....	95
4.1.4.1.	Manual de usuario para la plataforma	105
4.2.	DISCUSIÓN.....	106
4.2.1.	Gestión de datos e información de investigación.....	106
4.2.2.	Metadatos	107
4.2.3.	Intercambio y reutilización de datos	109
4.2.3.1.	Características de las plataformas web.....	111
4.3.	CONCLUSIONES Y RECOMENDACIONES	114
CAPÍTULO 5		116
5.1.	Anexos	116
5.1.1.	Anexo 1. Códigos del Back-End de SILAIN.....	116
5.1.1.1.	Conexión a la base de datos.....	116
5.1.1.2.	Rutas	116
5.1.1.2.1.	article.js	116

5.1.1.2.2.	basicSearch.js	122
5.1.1.2.3.	filters.js	125
5.1.1.2.4.	licence.js	132
5.1.1.2.5.	makepublic.js	133
5.1.1.2.6.	map.js	134
5.1.1.2.7.	users.js	139
5.1.1.3.	Índice del Back-End	141
5.1.2.	Códigos del Front-End de SILAIN	142
5.1.2.1.	Componentes	142
5.1.2.1.1.	CreateFile.jsx	142
5.1.2.1.2.	CreateLicence.jsx	146
5.1.2.1.3.	CustomFilters	147
5.1.2.1.4.	Footer.jsx	152
5.1.2.1.5.	Pagination.jsx	153
5.1.2.1.6.	ResultsTable.jsx	154
5.1.2.1.7.	SNavBar.jsx	156
5.1.2.1.8.	CustomFiltersByMap.jsx	159
5.1.2.1.9.	FiltersOnMap.jsx	162
5.1.2.1.10.	PropiedadByMap.jsx	165
5.1.2.1.11.	ResultsTableByMap	169
5.1.2.1.12.	Project1.jsx	171
5.1.2.1.13.	Project2.jsx	182
5.1.2.1.14.	Project3.jsx	190
5.1.2.1.15.	Project4.jsx	202
5.1.2.2.	Vistas	209
5.1.2.2.1.	Acerca.jsx	209
5.1.2.2.2.	Article.jsx	211
5.1.2.2.3.	Clave.jsx	218
5.1.2.2.4.	Login.jsx	224
5.1.2.2.5.	Perfil.jsx	227
5.1.2.2.6.	Propiedad.jsx	230
5.1.2.2.7.	Proyectos.jsx	233
5.1.2.2.8.	Registro.jsx	236

5.1.2.2.9.	Start.jsx.....	241
5.1.2.2.10.	TheMap.jsx	243
5.1.2.2.11.	Search.jsx	252
5.1.2.3.	Ruta de conexión con el Back-End	253
5.1.2.4.	Compilación de vistas y componentes	254
5.1.2.5.	Índice del Front-End.....	255
5.1.3.	Anexo 2. Manual de usuario	256
5.1.3.1.	Descripción de las páginas de la plataforma.....	256
5.1.3.1.1.	Página de inicio	256
5.1.3.1.2.	Página de inicio de sesión	257
5.1.3.1.3.	Página de creación de cuenta.....	258
5.1.3.1.4.	Página del Buscador por Palabra Clave.....	259
5.1.3.1.5.	Resultados en el Buscador por Palabra Clave	260
5.1.3.1.6.	Página de la exploración de datos	261
5.1.3.1.7.	Página de resultados en el Explorador de Datos	261
5.1.3.1.8.	Página del Buscador Geográfico	262
5.1.3.1.9.	Página de resultados del Buscador Geográfico	263
5.1.3.1.10.	Ventana emergente al usar buscar finca más cercana.....	264
5.1.3.1.11.	Página de Proyectos de Investigación	265
5.1.3.1.12.	Página al ingresar a un proyecto de investigación	266
5.1.3.1.13.	Página al ingresar a un dato	267
5.1.3.1.14.	Ventana emergente al descargar un dato	267
5.1.3.1.15.	Ventana emergente al solicitar acceso a un dato.....	268
5.1.3.1.16.	Página del panel de usuario (cliente)	268
5.1.3.1.17.	Página de resultados para el administrador	269
5.1.3.1.18.	Página del panel de usuario (administrador)	270
5.1.3.2.	Operaciones del sistema	271
5.1.3.2.1.	Registro en la plataforma	271
5.1.3.2.2.	Inicio de sesión	271
5.1.3.2.3.	Uso del Buscador por Palabra Clave	272
5.1.3.2.4.	Filtrado de resultados en buscador por Palabra Clave.....	272
5.1.3.2.5.	Uso del explorador de datos	272
5.1.3.2.6.	Uso del Buscador Geográfico	272

5.1.3.2.7. Uso de Proyecto de Investigación	273
5.1.3.2.8. Descarga de un dato.....	273
5.1.3.2.9. Solicitud de acceso a un dato	274
5.1.3.2.10. Historial de descargas de datos	274
5.1.3.2.11. Carga de datos y metadatos	274
5.1.3.2.12. Autorización de descargas	274
5.1.3.2.13. Cambio del tipo de acceso a un dato	274
5.1.3.2.14. Cierre de sesión.....	275
CAPÍTULO 6	276
6.1. BIBLIOGRAFÍA.....	276

TABLA DE TABLAS

Tabla 1. Tipos de archivos generado en las investigaciones de Agricultura Inteligente	20
Tabla 2. Tabla metadato	58
Tabla 3. Tabla categoria	59
Tabla 4. Tabla subcategoria	59
Tabla 5. Tabla metadato_has_subcategoria	59
Tabla 6. Tabla departamento	60
Tabla 7. Tabla municipio	60
Tabla 8. Tabla finca	60
Tabla 9. Tabla metadatos_has_finca	61
Tabla 10. Tabla role	61
Tabla 11. Tabla usuario	61
Tabla 12. Tabla descargas	62
Tabla 13. Tabla licencias	62
Tabla 14. Triggers, funciones y tablas auditoras	65
Tabla 15. Simbología en el Buscador Geográfico	67
Tabla 16. Escalas aproximadas en el Buscador Geográfico	67
Tabla 17. Categorización de la información	69
Tabla 18. Organización de la información según zona de estudio y fase	73
Tabla 19. Esquema de metadatos en la base de datos	78
Tabla 20. Vistas de la interfaz de usuario	83
Tabla 21. Componentes de la interfaz de usuario	84
Tabla 22. Pruebas realizadas a las operaciones del sistema	88
Tabla 23. Estructura de metadatos de SILAIN	92

TABLA DE FIGURAS

Figura 1. Esquema metodológico en cascada para el desarrollo del sistema	40
Figura 2. Diagrama del flujo de operaciones en el sistema	51
Figura 3. Diagrama lógico del ingreso a SILAIN	53
Figura 4. Diagrama lógico del uso de la utilidad Búsqueda por Palabras Clave	53
Figura 5. Diagrama lógico del uso de la utilidad Exploración de Datos	54
Figura 6. Diagrama lógico del uso de la utilidad Buscador geográfico	54
Figura 7. Diagrama lógico del uso de la utilidad Proyectos de Investigación	54
Figura 8. Diagrama lógico del acceso a un dato	55
Figura 9. Diagrama lógico del uso del panel de usuario.....	55
Figura 10. Diagrama lógico del uso de las utilidades de búsqueda por parte del administrador	56
Figura 11. Diagrama lógico del uso del panel de administración	56
Figura 12. Modelo relacional de la base de datos	57
Figura 13. Cardinalidad de las relaciones	63
Figura 14. Auditoría de cambios en la base de datos	64
Figura 15. Sistema de gestión de bases de datos PostgreSQL y la extensión PostGIS	70
Figura 16. Arquitectura del sistema	71
Figura 17. Entorno de ejecución NodeJS.....	71
Figura 18. Biblioteca ReactJS.....	71
Figura 19. Librería Leaflet.....	72
Figura 20. Tablas de la base de datos	74
Figura 21. Triggers implementados en la base de datos	76
Figura 22. Tablas auditoras implementadas en la base de datos.....	76
Figura 23. Funciones implementadas en la base de datos	77
Figura 24. Relación de vistas y componentes en la interfaz de usuario	85
Figura 25. Integración de los componentes del software del sistema	87
Figura 26. Pantalla de inicio de SILAIN.....	95
Figura 27. Interfaz para inicio de sesión	96
Figura 28. Formulario de registro en SILAIN.....	96
Figura 29. Búsqueda por Palabras Clave	97
Figura 30. Resultados de búsqueda	97
Figura 31. Filtrado de los resultados de búsqueda	98
Figura 32. Exploración de Datos.....	98
Figura 33. Buscador geográfico	99
Figura 34. Resultados en Buscador Geográfico.....	99
Figura 35. Proyectos de Investigación	100
Figura 36. Ejemplo de Proyecto de Investigación	100
Figura 37. Acceso a datos en Proyectos de Investigación	101

Figura 38. Visualización de los metadatos	101
Figura 39. Formulario para uso del dato	102
Figura 40. Dato con acceso restringido.....	102
Figura 41. Mensaje en datos restringidos	103
Figura 42. Historial de descargas	103
Figura 43. Carga de nuevos datos y metadatos.....	104
Figura 44. Asignación de autorizaciones de descarga	104
Figura 45. Modificación del acceso a un dato	105
Figura 46. Descripción de la página de inicio.....	256
Figura 47. Descripción de la página de inicio de sesión.....	257
Figura 48. Descripción de la página de registro	258
Figura 49. Descripción del buscador de palabras clave	259
Figura 50. Descripción de resultados en buscador de palabras clave.....	260
Figura 51. Continuación descripción de resultados en buscador de palabras clave	260
Figura 52. Descripción del explorador de datos	261
Figura 53. Descripción de resultados en explorador de datos.....	262
Figura 54. Descripción del buscador geográfico	263
Figura 55. Descripción de resultados en buscador geográfico	264
Figura 56. Descripción de utilidad finca más cercanas	265
Figura 57. Descripción de Proyectos de Investigación	265
Figura 58. Descripción de ejemplo de proyecto de investigación	266
Figura 59. Descripción de página del dato	267
Figura 60. Descripción del formulario de uso del dato	267
Figura 61. Descripción de solicitud de acceso a dato	268
Figura 62. Descripción del panel de usuario	268
Figura 63. Descripción de resultados para administrador	269
Figura 64. Descripción de panel de administración.....	270
Figura 65. Continuación de descripción de panel de administración	270

RESUMEN

En el marco del Proyecto del Sistema General de Regalías 2014-2020 se consolidó el Centro de Investigación e Innovación en Bioinformática y Fotónica (CIBioFi). Dentro de CIBioFi se construyó un Subproyecto Agricultura Inteligente (SAI) para el cual se desarrollaron cinco proyectos en diversos temas de Agricultura Inteligente con el apoyo del Grupo de Investigación en Simulación y Modelación Dinámica Espacial (GISMODEL). Estos proyectos generaron un gran volumen de información agronómica y ambiental mediante la utilización de variadas técnicas como la fotografía multiespectral, espectrometría, muestreo de suelos y vegetación, levantamientos posicionales de campo y estaciones de monitoreo ambiental. Reconociendo el alto valor que tiene la información del SAI para el desarrollo de cualquier tipo de investigación, se identificó la necesidad contar con una herramienta para gestionar la información en términos de almacenamiento, facilitar su consulta y acceso. Para tal fin se construyó el Sistema de Información del Laboratorio de Agricultura Inteligente (SILAIN), siendo este una plataforma web desarrollada con herramientas de código abierto que cuenta con diferentes utilidades para consultar, localizar, explorar y acceder a información, datos, publicaciones, metodologías, autores, técnicas, resultados, geoportales y demás productos del SAI. SILAIN se apoya en los principios de datos FAIR (Findable: Localizable, Accessible: Accesible, Interoperable: Interoperable, Reusable: Reutilizable), los cuales son un conjunto de cualidades con las que deben contar los datos científicos para garantizar su validez, su reproductibilidad y para conducir a nuevos descubrimientos. De este modo, SILAIN busca convertirse en una memoria de investigación y a la vez en un medio de divulgación de los datos, haciendo posible la reutilización de la información, procesos y productos que se generan en el Centro, dando visibilidad a los autores para potenciar la gestión científica del CIBioFi en la comunidad Vallecaucana.

Palabras clave: Sistema de Información, Agricultura Inteligente, Gestión de Información, Datos de investigación, Metadatos, Reutilización de datos, FAIR Data.

INTRODUCCIÓN

La gestión de los datos y la información es una preocupación dentro de cualquier entorno de investigación (Androulakis et al., 2009). Los altos costos y esfuerzos que implican la captura, procesamiento y análisis de los datos, son razones que motivan a un cuidado espacial de los objetos de investigación.

En la actualidad, la sociedad depende en gran medida de la ciencia y la investigaciones, pero no siempre se hace lo suficiente por optimizar el trabajo científico (Frederick, 2016). Se debe comenzar por reconocer el valor que representan los datos para las investigaciones donde se originan e incluso para otras distintas a la inicial, ya que los datos adquieren aún más valor cuando son agregados junto con más datos (Van der Kwast, 2017).

Con el propósito de aprovechar todo el potencial que se concentra en los datos de investigación, estos deben ser dispuestos de una manera organizada, contar con documentación que describa detalladamente sus características y su acceso ágil y fácil debe estar asegurado.

Para el Centro de Investigación CIBioFi, la seguridad alimentaria es uno de los enfoques de investigación, esto, a raíz del rápido crecimiento poblacional. Según proyecciones, la población mundial será de 9700 millones de habitantes para el año 2050 (ONU, 2017), situación que traerá consigo muchos retos de carácter político, social y económico. En vista de esto, en el Centro se formuló un proyecto denominado Agricultura Inteligente, que tiene como objetivo la producción de conocimiento científico para apoyar la toma de decisiones en el sector agroindustrial.

A partir de las actividades de investigación del proyecto Agricultura Inteligente adelantadas por el grupo de investigación GISMODEL, se han generado grandes volúmenes de datos e información útiles para el sector agro-productor regional y nacional.

De momento no se cuenta con un mecanismo que gestione de forma eficiente los datos e información de Agricultura Inteligente, y que a su vez proporcione a la comunidad científica un acceso expedito a estos recursos de investigación.

Reconociendo el estado actual de la información del Proyecto Agricultura Inteligente, que impide una rápida localización y consulta de los datos, se propone en el presente proyecto el desarrollo de un sistema de gestión de información Web que brinde a los científicos del Centro y a la comunidad científica en general, un acceso fácil y rápido a conjuntos de datos descritos a través de metadatos. Al mismo tiempo, con el sistema apoyado en los principios rectores de datos FAIR (Craig, 2019; Wilkinson, 2016), se pretende favorecer la reutilización de los datos.

Este documento está estructurado de la siguiente manera:

El primer capítulo contiene la contextualización y definición del problema abordado, lo que también lleva a la formulación de la pregunta de investigación. Seguidamente se presentan los objetivos por lograr en el proyecto. Luego se expone la justificación, donde se presenta las razones y la conveniencia de resolver el problema.

El segundo capítulo corresponde a la descripción de los conceptos teóricos involucrados en el problema y la solución planteada. Posteriormente reúne los antecedentes, la evolución, la situación actual y las tendencias frente a las temáticas del proyecto.

En el tercer capítulo se expone la estrategia metodológica de la solución propuesta y el desarrollo de la metodología a través de cinco fases.

En el cuarto capítulo se presentan y describen los resultados obtenidos, los cuales son discutidos de acuerdo con los resultados alcanzados por otros autores. A esto le siguen las conclusiones del trabajo desarrollado y recomendaciones.

El quinto capítulo contiene los anexos del documento.

Finalmente, en el quinto capítulo se lista la bibliografía consultada.

CAPÍTULO 1

1.1. PROBLEMA

1.1.1. Contextualización

El Centro de Investigación e Innovación en Bioinformática y Fotónica (CIBioFi) es una institución científica de la Universidad del Valle, que nació en el año 2015, a partir de un macroproyecto de investigación financiado por el Sistema General de Regalías (ACUERDO No. 27 DE 05/08/2014, 2014). El Centro está conformado por grupos de investigación de las facultades de ciencias naturales y exactas e ingeniería. Así mismo, cuenta con la colaboración externa de un grupo de la Universidad ICESI y otro de la Universidad Nacional sede Palmira. Como parte de la convocatoria se vinculó al sector empresa, donde participan las empresas Hugo Restrepo & Cía. S.A. y La Arrocera la Esmeralda. Por medio de investigación de alta calidad, CIBioFi busca la generación de conocimiento científico como apoyo para la solución de problemáticas en el sector agrario a nivel regional y nacional, buscando una mejor calidad de los productos, bienestar social y un uso eficiente de los recursos naturales (Vicerrectoría de Investigaciones, 2016).

La seguridad alimentaria es uno de los temas abordados dentro de la investigación del Centro, motivado por el rápido crecimiento de la población mundial (Godfray et al., 2010). Dentro de este campo de acción, se priorizó la investigación en los cultivos de arroz y ají, reconociéndolos como una posible solución alimentaria.

Dentro de los temas de investigación de CIBioFi, se busca identificar cómo los procesos productivos agrícolas representan un impacto sobre los recursos naturales, para los cuales se propende sean enmarcados bajo el tema de sostenibilidad ambiental. De acuerdo con esto, se formuló el proyecto de Agricultura Inteligente, dirigido al diseño e implementación de metodologías que mejoren la producción agrícola inicialmente para el Departamento del Valle del Cauca. Los estudios se enfocaron en la caracterización básica de los cultivos, partiendo del conocimiento detallado de las propiedades del cultivo en todas sus etapas, conformando un conjunto de información que brinda apoyo para la toma de decisiones y que desencadena en un mejor rendimiento de los cultivos, un uso adecuado de los recursos y una reducción de costos productivos (Basso et al., 2013).

Las actividades de CIBioFi en el Proyecto Agricultura Inteligente se dividen en tres áreas de conocimiento y su integración. La primera área es el conocimiento del medio natural, entendida desde la relación sistémica entre la atmósfera, vegetación, suelo y subsuelo. En segundo lugar, están las acciones humanas en la apropiación de recursos para su desarrollo y subsistencia. Por último, está se encuentra el marco de la sustentabilidad como estudio de la interacción de las dos anteriores, en búsqueda de la preservación de la vida, los recursos y el medio en que se desarrollan. Estos temas pueden ser analizados desde las dimensiones espaciotemporales, estudiadas en los procesos agronómicos.

Las investigaciones de CIBioFi inician con la formulación de una premisa o hipótesis que direccionan su estudio en busca del entendimiento de los fenómenos naturales y las actividades humanas dentro del sistema productivo agrícola de la región, para lo que se propone la construcción de diseños experimentales que permiten orientar las actividades de investigación. Una de estas actividades es la captura de información agronómica y ambiental, que puede ser realizada a través de múltiples técnicas como fotografía multiespectral (Candiago et al., 2015), espectrometría, (Thomasson & Cox, 2013) muestreo de suelos y vegetación, levantamientos posicionales de campo, estaciones de monitoreo ambiental, entre otras, y tiene como resultado grandes volúmenes de datos e información en diferentes formatos y a distintos niveles de agregación.

Varios subproyectos de investigación se han propuesto dentro del tema de Agricultura Inteligente que generan bloques de información en sus diferentes estadios. Esta información constituye grandes volumen de datos de diversos tipos en cada proceso y momento de las investigaciones, siendo esta muy valiosa no solo por sus costos de producción, sino por su importancia en la investigación científica para revisar y recrear cada uno de los aspectos de investigación formulados, como también con un alto valor para continuar y ser base de investigaciones posteriores, igualmente para apoyar procesos académicos que precisan de información confiable y de calidad.

1.1.2. Definición del problema

A partir de las investigaciones del proyecto de Agricultura Inteligente se ha derivado un número elevado de datos de distintas clases; alrededor de 5 TB de información se encuentran almacenadas en discos duros externos y en computadores del laboratorio GISMODEL del área de Geomática de la Universidad del Valle y del Laboratorio de Agricultura Inteligente (LAIN) de CIBioFi; se cuenta con archivos en formatos ráster, vectoriales, documentos de texto, hojas de cálculo, archivos estadísticos, imágenes en diferentes formatos, PDF, etc.

Adicionalmente, desde investigaciones propuestas se continúa recolectando datos y produciendo información en forma aislada; dos ejemplos de estos estudios son la identificación de macronutrientes en el cultivo de arroz a partir de firmas espectrales, y la caracterización espectral del ají según su estado de madurez. La obtención de los datos en estas investigaciones implica disponer de equipos, el uso de sensores, personal científico, apoyo en las labores en campo, análisis de laboratorio y demás técnicas, gestión logística y administrativa, donde todo ello constituye un alto costo de la investigación, y la recreación de momentos únicos en la investigación.

En la actualidad, toda captura de datos en investigación tiene un alto costo y toma un papel relevante, no solo por su utilidad en esa misma investigación, sino en los posteriores usos que se le pueda dar a dicha información. Por tal razón, pensar en gestionar la información, generar diccionarios de datos y metadatos, se convierten en requisitos para un aprovechamiento más eficiente del conocimiento producido en la investigación.

De momento, no se cuenta con un mecanismo para localizar y acceder fácilmente a la información y los conjuntos de datos obtenidos en el proyecto de Agricultura Inteligente, de manera que la comunidad científica del Centro no puede extraer todo el potencial que radica en dichos datos para la toma de decisiones en el sector agro-productor. Igualmente, la falta de gestión de la información impide su aprovechamiento en futuras investigaciones, ya sean de CIBioFi o de otras instituciones.

Es así como surge la necesidad de disponer de forma organizada el almacenamiento y gestión de la información resultante de la investigación en cada uno de sus etapas, como la captura, depuración, procesamiento, análisis y síntesis de los procesos estudiados, aprovechando el conocimiento generado de las múltiples técnicas empleadas y asimismo disponerla en una estructura de datos que permita una fácil localización, cuyo acceso sea lo menos restringido posible, y a la vez cumpliendo con estándares que hagan reutilizables los datos para la generación de nueva información y por ende de nuevo conocimiento, configurando un mecanismo que sirva de soporte para la toma de decisiones en el sector agro-productor del Departamento.

1.1.3. Pregunta de investigación

¿Cómo gestionar la información producida en las investigaciones del Proyecto de Agricultura Inteligente de CIBioFi adelantadas por GISMODEL, para disponerla a la comunidad científica?

1.2. OBJETIVOS

1.2.1. Objetivo General

Disponer de un sistema de información que gestione de forma eficiente el conocimiento producido por parte de GISMODEL en el proyecto Agricultura Inteligente de CIBioFi.

1.2.2. Objetivos Específicos

- Diseñar una estructura de almacenamiento de datos para la información generada en el proyecto de Agricultura Inteligente desde GISMODEL.
- Acondicionar una base de datos que sea localizable, accesible, interoperable y reutilizable.
- Establecer los métodos, protocolos y procedimientos para la gestión de la información en el sistema.
- Desarrollar una plataforma web que facilite la localización y acceso a los datos e información generada por los subproyectos de Agricultura Inteligente.

1.3. JUSTIFICACIÓN

Actualmente, el Grupo de Investigación GISMODEL ha adelantado cuatro investigaciones en el marco del Proyecto de Agricultura Inteligente de CIBioFi. Estos proyectos están relacionados con las áreas de modelación climática, sustentabilidad ambiental del recurso hídrico en el manejo de cultivos de arroz, identificación de la concentración de macronutrientes y metodologías para la estimación del rendimiento del cultivo de arroz, a partir de imágenes multiespectrales capturadas desde vehículos aéreos no tripulados. Otros estudios en el tema que están en curso se desarrollan sobre cultivos de ají, dirigidos a la identificación de macronutrientes y caracterización espectral de frutos de ají en diferentes etapas de madurez.

Las investigaciones mencionadas involucran vastos procesos de captura de datos en diferentes etapas del cultivo, en los que se deben tener referencias espaciales y temporales. En consecuencia, cada dato debe estar geoposicionado dentro de un sistema de referencia, los muestreos de campo y los resultados de laboratorio de los datos de los cultivos, suelos, subsuelo y aguas deben contar con sus tres dimensiones espaciales y la temporal.

Adicionalmente, por el enfoque de investigación basado en imágenes multiespectrales y espectrometría de las muestras, todos deben estar ligados a su componente espacial y temporal con alta calidad en su posicionamiento. De tal forma que, por cada investigación se obtienen densos conjuntos de datos, que podrían ser de alto valor dentro de los procesos de toma de decisiones del sector productor agrícola. A través de una revisión preliminar de los medios dispuestos para el almacenamiento, se hizo un inventario de información sin depurar de los subproyectos de Agricultura Inteligente, la Tabla 1 relaciona los principales tipos de archivos que contienen la información y el tamaño aproximado de estos.

Tabla 1. Tipos de archivos generado en las investigaciones de Agricultura Inteligente

Tipos de archivos	Tamaño aproximado (GB)
Imágenes	990
Documentos	3.5
Texto plano	16
Vectorial	6
Hojas de cálculo	0.2

Tan solo en los cuatro proyectos iniciales liderados por GISMODEL, se contó con cuatro sitios experimentales conformados por 13 parcelas, de las que se tienen muestras de 2500 puntos. Cada uno de ellos debió ser medido al menos 5 veces y se correlacionó con imágenes multiespectrales capturadas de las mismas fechas. Por cada vuelo de levantamiento de las imágenes multiespectrales se capturaron alrededor de 400 imágenes por cada una de las 5 bandas espectrales (rojo, verde, azul, borde rojo e infrarrojo cercano). Así mismo, cada muestra de planta en cada fecha tiene datos fisiográficos (altura, número de macollos y verdor), como también datos de su análisis de biomasa en laboratorio con sus contenidos de macronutrientes (nitrógeno, fósforo y potasio).

De este modo, se resalta que, para poder realizar un análisis integral con los propósitos de cada investigación, se debe asimilar la información en las condiciones que garantice inicialmente la accesibilidad al dato y a la información, y eventualmente disponerla para la continuación de la investigación, lo que permite tener la trazabilidad del dato en todo momento.

Hoy día, en los procesos de investigación planteados en CIBioFi se tiene conciencia de que los datos e información capturada, transformada y analizada en cada proceso no es solo útil para la misma investigación, sino que en todo ámbito de desarrollo de conocimiento la información podrá apoyar futuras investigaciones. En consecuencia, en CIBioFi se está pensando en construir un gran banco de datos de información que permita ser localizado, consultado, accedido y manejado en muchos más ámbitos que la misma investigación inicial (Wilkinson, 2016)

Una gestión adecuada de la información del Proyecto de Agricultura Inteligente proporcionaría a los investigadores del centro acceso expedito a cualquier dato en cualquier etapa del estudio, que sea requerido dentro de la formulación de soluciones para problemáticas del agro. Igualmente, la información podría ser consultada por científicos de otros centros e instituciones de investigación (Androulakis et al., 2009), permitiendo en primer lugar la verificación de los resultados de los experimentos (Wagenaar et al., 2013), así como la comparación de resultados con la información producida por el centro.

En ausencia de procesos de gestión de la información, existe la posibilidad de que los datos de investigación queden inutilizados al no ser fácilmente localizables, produciendo pérdidas económicas, teniendo en cuenta los elevados costos de la captura de la información, y se invertiría mucho tiempo en la ubicación de un conjunto de datos requerido, corriendo el riesgo de no encontrarlos, o hallar una versión de la cual no se tiene la certeza que sea la última versión del dato o la de mejor calidad. De igual forma se impediría la transmisión del conocimiento científico generado en los proyectos.

La implementación de un sistema de gestión de información agilizaría considerablemente las búsquedas de datos, lográndose el acceso a los datos en cuestión de segundos, esto, gracias a la utilización de metadatos que identifiquen con precisión los datos mediante información proporcionada por los encargados de la investigación (Bordogna et al., 2016). Con lo anterior, se evitaría depender del encargado de cada investigación para el acceso y localización de un conjunto de datos.

Por otra parte, la seguridad del dato y la información es otra preocupación del Centro, dadas las características de concepción del centro, las investigaciones que se desarrollan en CIBioFi deben estar bien respaldadas y aseguradas. El Centro se puede exponer a tener copias de los datos pasando de mano en mano o equipos de cada investigador. La custodia de la información de las investigaciones no puede estar en manos de terceros; esto es responsabilidad total del Centro. Por consiguiente, se debe garantizar más allá del acceso y la calidad, la confiabilidad, seguridad y respaldo de esta. Así, se deben generar los mecanismos que garanticen todo el respaldo y seguridad de lo que es de mayor valor en una investigación, la información.

CAPÍTULO 2

2.1. MARCO TEÓRICO

Como se ha podido apreciar en las secciones anteriores, el eje central de este proyecto son los datos, la información y el conocimiento; sin embargo, es necesario conocer las diferencias entre estos conceptos. De la misma forma, términos como bases de datos y sistemas de información son importantes para comprender la solución que se plantea para hacer que los datos sean accesibles y reutilizables. También, paradigmas como Open Data y FAIR Data han sido involucrados en el sistema, por lo que es indispensable conocer más a fondo sobre estos temas.

2.1.1. Datos, información y conocimiento

La palabra dato se refiere a un valor específico que, por sí solo, no proporciona ningún conocimiento útil, mientras que el concepto de información corresponde a un par ordenado, el conjunto de un dato y una descripción, en otras palabras, la información es un dato dentro de un contexto (Cooper, 2017; Murdick, 1989) y que tiene un valor real en el apoyo de la toma de decisiones (Lapiedra et al., 2011). Otra definición aceptada para datos consiste en que estos son hechos, acontecimientos y transacciones que se almacenan en un código convenido, los cuales han sido obtenidos mediante la lectura, la observación, el cálculo, la medición, etc. (Lapiedra et al., 2011).

Por otro lado, el conocimiento se define como la comprensión y evaluación de un conjunto de información que puede representar utilidad para el apoyo en tarea específicas o para la toma de decisiones (R. Stair & Reynolds, 2010).

Los datos son el alma de la ciencia y la tecnología, cumplen un papel importante en la formulación, prueba y revisión de hipótesis, teorías y modelos (Frederick, 2016). Gracias a la diversidad de datos producidos por la ciencia, se tiene capacidad para responder a los brotes de enfermedades, administrar los recursos, tomar acciones frente al cambio climático y mejorar el transporte (Hanson et al., 2011), entre otras acciones.

Se podría añadir un eslabón más a la cadena de datos, información y conocimiento, esto vendría a ser la sabiduría, que se construye a partir de la acumulación de conocimiento en combinación con la experiencia, lo que permite un profundo entendimiento de un tema en particular (Bourgeois, 2014).

2.1.1.1. Bases de datos

Las bases de datos, también llamados bancos de datos, son conjuntos organizados de datos que pertenecen a un mismo contexto y que están almacenados de forma sistemática para su posterior uso (González-Longatt & Francisco, 2007; R. M. Stair & Reynolds, 2010).

En una base de datos, todos los datos están descritos y asociados con otros datos, igualmente, toda la información en una base de datos debería estar relacionada (Bourgeois, 2014).

Desde antes de la era digital y la creación de los computadores y sistemas informáticos existían las bases de datos, puesto que existía la necesidad de llevar registros detallados de grandes cantidades de información para que pudieran ser consultada posteriormente (Berg et al., 2013). Esto sucedía con frecuencia en bibliotecas, entes gubernamentales, instituciones de ciencia, hospitales, etc.

El término base de datos, en lo que respecta a sistemas computacionales, surgió en la década de 1960 cuando se crearon los primeros medios de almacenamiento de acceso directo. Las primeras concepciones fueron el modelo de red (CODASYL) y el modelo jerárquico IMS (Information Management System; Sistema de administración de información en español). Estas primeras bases de datos permitían explorar registro por registro, lo que forzaba al usuario a conocer la estructura física de la base de datos para poder realizar consultas, además de que eran complicadas y requerían de mucho tiempo (Berg et al., 2013).

Luego, en la década de 1970, Edgar Frank Codd desarrolló un nuevo tipo de base de datos, las BBDD relacionales. En estas, los datos se representaban en estructuras tabulares de filas y columnas (Mansour et al., 2017), los datos eran buscados por su contenido y no por sus enlaces. Se basan en la teoría de conjuntos y lógica de predicados. La organización lógica es independiente de la información física. Los dos sistemas pioneros del modelo relacional de base de datos fueron Ingres y el Sistema R.

Posteriormente, de mano de Peter Chen vino el modelo entidad-relación en 1976, lo que permitió un mejor enfoque a los datos y no a la estructura lógica de la tabla. Paralelo a esto, se construyeron los primeros sistemas de gestión de bases de datos relacionales.

En la década de 1980, a medida que los computadores se volvieron más asequibles y se dio un aumento en la compra de computadores, las bases de datos relacionales llegaron a su auge. Structured Query Language (SQL, por sus siglas en inglés: Lenguaje de Consulta Estructurada) se convierte en el estándar para la administración y recuperación de información de las bases de datos. En esta época, los modelos jerárquicos y red cayeron en desuso (Berg et al., 2013).

Otra clase de BBDD nacieron en el año de 1985, las bases de datos orientadas a objetos, en este modelo la información se almacenaba en forma de objetos, operando de manera semejante a como lo hacen diferentes lenguajes de programación orientados a objetos. Estas bases de datos resultaron eficientes para gran cantidad de variados tipos de datos, sin embargo, era complejo migrar de bases relacionales hacia estas nuevas orientadas a objetos. Como solución a este problema, en la década de 1990 surgieron las BBDD relacionales de objetos (Mansour et al., 2017).

Con la aparición del internet se desarrollan las bases de datos web, haciendo posible el acceso remoto a estas. En 1997 se crea el eXtensible Markup Language (XML por sus siglas en inglés: Lenguaje de Marcado Extensible) este sirvió para procesar mayores volúmenes de datos (Berg et al., 2013).

Ya en el siglo XXI, los avances en bases de datos estuvieron dedicados a la sofisticación de los sistemas, asimismo los lenguajes de programación adoptados fueron más complejos. Se añadieron características más avanzadas como disparadores, actualización en cascada y eliminación, dándole una mayor consistencia a las tablas de la base de datos. También, se crearon nuevas tecnologías pensadas para campos como la minería de datos (Mansour et al., 2017).

En este nuevo siglo nació un nuevo modelo para BBDD, llamadas bases de datos NoSQL, las cuales permiten almacenar datos en una estructura de clave-valor, estas admiten una inmensa variedad de datos, cuenta con almacenes de documentos y gráficos. Poseen una mayor velocidad y soportan grandes volúmenes de información. No obstante, las consultas en estas bases de datos se tornan complejas, de modo que requieren de programadores experimentados (Harrison, 2015).

Actualmente, las tendencias en cuanto a bases de datos conducen a nuevos modelos como las bases de datos distribuidas, la computación en la nube y el Big Data.

2.1.1.2. Intercambio de datos

El intercambio de datos es la publicación de datos de investigación para su uso por parte de otros (C. Borgman, 2012). Esta práctica puede traer consigo muchas ventajas (Tenopir et al., 2011) como:

- Verificación de resultados de investigación.
- Múltiples interpretaciones o enfoques en entornos interdisciplinarios.
- Preservación a largo plazo los datos.
- Evitar la repetición en la recopilación datos.

- Proporcionar garantías en contra la falsificación de datos.
- Orientar a nuevas generaciones de investigadores.
- Ofrecer una única versión válida y original para el uso de todos.

También, el intercambio de datos entre la comunidad científica puede propiciar la generación de nuevas preguntas, en especial cuando se pueden combinar datos de múltiples fuentes, momentos y lugares (Whitlock, 2011).

La reutilización de datos es el propósito principal del intercambio de datos, pero debe tenerse en cuenta la diferencia entre reutilizar y usar datos. Si un científico produce un dato para una investigación, lo usa, y posteriormente dentro del mismo contexto lo vuelve a utilizar, esto sería un nuevo uso. Por otro lado, reutilizar es cuando el dato es utilizado en un contexto distinto o es utilizado por alguien diferente a su creador (Pasquetto et al., 2019).

2.1.1.3. Curación de datos

La curación de datos, también llamada conservación de datos, contempla una serie de acciones dirigidas hacia la gestión activa y continua de los datos a través de su ciclo de vida de interés y utilidad para las actividades académicas y educativas (Hudson-Vitale et al., 2017).

Debido a que cada vez es más frecuente que se exija a los investigadores hacer públicos sus conjuntos de datos, los datos deben ser preparados para su reutilización, lo que incluye verificar su calidad e integridad, creación de metadatos, estandarización y especificación de licencias o derechos de uso adecuados.

La curación de datos hace parte de la curación digital, la cual se define como el proceso de establecer y hacer confiable la información digital dentro de repositorios a largo plazo para uso actual y futuro por investigadores, científicos, historiadores y académicos en general. (Wood, 2010). Las actividades que reúne la curación digital son la selección, preservación, mantenimiento, recopilación y almacenamiento de activos digitales (Ball, 2010).

2.1.1.4. Datos abiertos

Permitir que los datos sean accesibles de forma libre por cualquier individuo u organización y sin restricciones de uso, concediéndole la posibilidad de reutilizar recursos y agregar datos nuevos para la producción de nueva información, es la forma de darle un valor agregado a los datos recolectados en investigación, este concepto se conoce como datos abiertos.

Los datos abiertos son datos que pueden ser utilizados libremente, reutilizados y redistribuidos por cualquier persona (Open Knowledge Foundation, s.f.)

Las características de los datos abiertos son:

- Disponibilidad y acceso: los datos deben estar disponibles en su conjunto y a un costo razonable de reproducción, preferiblemente mediante descarga a través de Internet. Los datos también deben estar disponibles de forma conveniente y modificable.
- Reutilización y redistribución: los datos deben proporcionarse en términos que permitan la reutilización y la redistribución, incluida la mezcla con otros conjuntos de datos.
- Participación universal: todos deben poder usar, reutilizar y redistribuir; no debe haber discriminación contra los campos de trabajo o contra personas o grupos. Por ejemplo, no se permiten restricciones no comerciales que impidan el uso comercial, o restricciones de uso para ciertos fines (por ejemplo, solo en educación).

La definición de datos abiertos según la ley colombiana de transparencia y acceso a la información dice que son *todos aquellos datos primarios o sin procesar, que se encuentran en formatos estándar e interoperables que facilitan su acceso y reutilización, los cuales están bajo la custodia de las entidades públicas o privadas que cumplen con funciones públicas y que son puestos a disposición de cualquier ciudadano, de forma libre y sin restricciones, con el fin de que terceros puedan reutilizarlos y crear servicios derivados de los mismos* (Congreso de la República de Colombia, 2014).

2.1.1.5. Metadatos

Uno de los principales requisitos para que los datos sean fácilmente accesibles y localizables, es que cuenten con una descripción juiciosa y detallada conocida como metadatos.

Metadato o "dato sobre dato" es un término comúnmente empleado en diversas comunidades profesionales que diseñan, crean, describen, preservan y utilizan los recursos y sistemas de información. Hasta mediados de la década de 1990, la palabra metadatos era principalmente usada por comunidades involucradas con la gestión e interoperabilidad de los datos geoespaciales y con la gestión de datos y el diseño y mantenimiento de sistemas en general, para estas comunidades, el término se refería a un conjunto de estándares industriales o disciplinarios, así como documentación interna y externa adicional y otros datos necesarios para la identificación, representación, interoperabilidad, gestión técnica, rendimiento y uso de los datos contenidos en un sistema de información (Baca & Getty Research Institute, 2008).

El Instituto Geográfico Agustín Codazzi (IGAC), entidad encargada de la cartografía oficial de Colombia, define al metadato como el *conjunto de información sobre datos previamente levantados y buscan establecer un lenguaje común y concreto que describan un dato. Es decir, es toda información descriptiva sobre el contexto, calidad, condición o características de un recurso, dato u objeto que tiene la finalidad de facilitar su recuperación, autenticación, evaluación, preservación e interoperabilidad* (IGAC, s.f.).

Con el fin de identificar adecuadamente y mejorar el acceso a los datos y la información que circula en diferentes organizaciones y en la web, se han diseñado diferentes especificaciones y estándares para el uso de metadatos, algunos empleados a nivel internacional, otros adaptados a las necesidades de regiones, igualmente, estos modelos de descripción de la información varían según los tipos de datos existentes.

El gran volumen de los contenidos en la web llevó a que el World Wide Web Consortium (W3C) buscara facilitar el acceso a datos e información relevante a sus usuarios; para ello W3C desarrolló una serie de especificaciones que denominaron Marco de Descripción de recursos (RDF), que ofrece un esquema para la organización de los recursos en la web. El RDF propuso un modelo de datos para metadatos asociados a los recursos de la red, que los diferenciara y permitiera que los motores de acceso e integración de información fueran más eficientes y precisos (Candan et al., 2001).

Dublin Core es otro estándar de metadatos sencillo pero eficiente para la descripción de diversos recursos en la red. Este estándar se apoya en quince elementos que son de utilidad para la recuperación y localización de contenidos en internet. Dichos elementos están clasificados en tres grupos: contenido, propiedad intelectual e instanciación (Hillman, 2005). En Dublin Core no es de obligatoriedad especificar todos los elementos de metadatos, pues considera preferible que los recursos tengan descripciones, así sean simples, a que no cuenten con descripción alguna. Los modelos de Dublin Core son aceptados en más de 20 países en Norte América, Europa, Asia y Oceanía.

En el contexto colombiano, el Instituto Colombiano de Normas Técnicas y Certificación (ICONTEC) establece en la Norma Técnica Colombiana 4611 (NTC 4611) el estándar que proporciona la estructura para describir datos geográficos en formato análogo y digital o servicios geográficos. Esta norma técnica facilita la documentación y el acceso a la información, además permite la interoperabilidad entre catálogos de metadatos por ser esta un perfil de ISO 19115, es decir, es un subconjunto de ese estándar internacional, destinado únicamente para Colombia. (NORMA TÉCNICA NTC COLOMBIANA 4611, 2011). Desde el 2016, el IGAC hace uso de gestor de metadatos conocido como Catálogo Nacional de Metadatos, implementado sobre el software libre GeoNetwork y administrado por la Infraestructura Colombiana de Datos Espaciales (ICDE).

La Norma Internacional ISO 19115 “Información Geográfica – Metadatos”, suministra un modelo para describir información o recursos que pueden tener una extensión geográfica. De manera general, en esta norma se definen los elementos de metadatos, sus propiedades, las relaciones entre elementos y, terminología, definiciones y procedimientos para la extensión de los metadatos (Pascual et al., 2014). La ISO 19115 tiene la particularidad de que puede ser aplicada para la descripción de recursos de información que no tengan un carácter geográfico, es decir, que se definen metadatos de propósito general.

2.1.1.6. Datos FAIR

El mundo de los datos abiertos abre paso a la generación de nuevo conocimiento, no obstante, un nuevo concepto denominado datos FAIR amplía aún más las oportunidades de reutilización de los datos y la información.

FAIR es el acrónimo en inglés para Findable, Accessible, Interoperable and Reusable, que se traduce al español como Localizable, Accesible, Interoperable y Reutilizable, se trata de los cuatro principios fundamentales de un nuevo paradigma para la gestión y administración de datos de carácter científico que reconoce el valor de los datos dentro de investigaciones posteriores, ya sea por sí solos o mediante la agregación de nuevos datos (Wilkinson et al., 2016).

A pesar de que los humanos puedan identificar, interpretar y seleccionar datos en una amplia variedad de contextos y que se hallen estructurados de diferentes maneras, no logran realizar tales actividades con la misma rapidez ni a la escala que exigen los datos actuales y su complejidad. Los principios FAIR están dirigidos a mejorar el descubrimiento y reutilización de datos tanto para humanos como para máquinas.

2.1.1.6.1. Principios rectores FAIR

Para ser localizables:

- Los datos y metadatos poseen un único identificador persistente.
- Los datos se describen con metadatos enriquecidos.
- Los metadatos incluyen clara y explícitamente el identificador de los datos que describe.
- Los datos se registran o indexan en un recurso de búsqueda.

Para ser accesibles:

- Los datos y metadatos deben ser recuperables por su identificador utilizando un protocolo de comunicaciones estandarizado.
 - El protocolo es abierto, gratuito y universalmente implementable.
 - El protocolo permite un procedimiento de autenticación y autorización, cuando sea necesario.
- Los metadatos son accesibles, incluso cuando los datos ya no están disponibles.

Para ser interoperables:

- Los datos y metadatos usan un lenguaje formal, accesible, compartido y ampliamente aplicable para la representación del conocimiento.
- Los datos y metadatos usan vocabularios que siguen los principios FAIR.
- Los datos y metadatos incluyen referencias calificadas a otros (meta) datos.

Para ser reutilizables:

- Los datos y metadatos se describen detalladamente con una pluralidad de atributos precisos y relevantes.
 - Los datos y metadatos se liberan con una licencia de uso de datos clara y accesible.
 - Los datos y metadatos están asociados con una procedencia detallada.
 - Los datos y metadatos cumplen con los estándares de la comunidad relevantes para el dominio.

2.1.2. Sistemas de Información

Los sistemas de información son un conjunto de componentes interrelacionados que tienen como función recopilar, procesar, almacenar y difundir información para apoyar la toma de decisiones, coordinación, control, análisis y visualización en una organización (Laudon et al., 2012).

Los componentes de los sistemas de información son el hardware, software y las redes de telecomunicaciones por medio de los cuales se crean y distribuyen datos útiles en organizaciones (Valacich et al., 2016).

2.1.2.1. Sistemas de Información Basados en Computadora

Desde el punto de vista de las tecnologías de la información, existe el término Sistema de Información Basado en Computadora (CBIS, por sus siglas en inglés: Computer Based Information System), que está definido como el conjunto único de hardware, software, bases de datos, telecomunicaciones, personas y procedimientos configurado para recolectar, manipular, almacenar y procesar datos para convertirlos en información (R. Stair & Reynolds, 2010). Esta serie de acciones sobre los datos están inscritas en tres actividades principales que son: Entrada, Procesamiento y Salida.

La entrada es la actividad en la que se realiza la recopilación y captura de datos; en el procesamiento se hacen conversiones o transformaciones de datos que implican cálculos, comparaciones y almacenamiento de datos para usos futuros que se convierten en un resultado o salida útil; la salida corresponde a la producción de información útil, generalmente a manera de documentos o reportes.

El hardware en los sistemas de información comprende los equipos de cómputo requeridos para realizar actividades de entrada, procesamiento y salida.

El software lo conforman los programas que permiten la operación de los computadores y los procesos que estos ejecutan.

Las bases de datos son conjuntos organizados de hechos e información que por lo general constan de dos o más archivos de datos relacionados.

Las telecomunicaciones consisten en la transmisión electrónica de señales de comunicación, que permiten a las organizaciones llevar a cabo procesos y tareas a través del uso de eficientes redes de computadoras.

Las personas son el componente encargado de administrar, operar, programar y mantener el sistema.

Los procedimientos constituyen las estrategias, políticas, métodos y reglas para utilizar los sistemas de información, lo cual incluye la operación, mantenimiento y seguridad de la computadora.

2.1.2.2. Sistemas de administración del conocimiento

Los sistemas de administración del conocimiento (KMS, por sus siglas en inglés: knowledge management system) son conjuntos organizados de personas, procedimientos, software, bases de datos y dispositivos utilizados para crear, almacenar, compartir y usar el conocimiento y la experiencia de una organización (Chua, 2007).

Los KMS están dirigidos a proporcionar ayuda a una organización en el cumplimiento de sus objetivos y atender las necesidades de grupos de personas (Seneviratne, 2017). Por lo general, el personal involucrado en los sistemas de administración de conocimiento reúne profesionales en ciencias, ingeniería y negocios, así mismo puede llegar a incluir investigadores, educadores y diseñadores corporativos (McGregor & Hamm, 2008).

2.1.3. Software de código abierto

El software de código abierto, en inglés llamado Open Source Software (OSS), es un código diseñado con el fin de que sea accesible al público, de modo que todos pueden ver, modificar y distribuir el código de acuerdo con sus necesidades (Red Hat, s.f.).

El desarrollo de software de código abierto se da de forma descentralizada y colaborativa, por lo que depende de la revisión entre compañeros y la producción de la comunidad. Las principales diferencias frente al software propietario, en el que no se puede acceder al código fuente, están en que el OSS suele ser más económico e incluso gratuito, flexible y duradero, esto se debe a que las encargadas de su desarrollo son las comunidades y no un solo autor o empresa.

Este tipo de software tiene la posibilidad de ser reutilizado por cualquier usuario de cualquier lugar del mundo, de manera que un desarrollo tecnológico generado en un lugar en particular puede ser transferido y adaptado a un contexto local de otro lugar en el que exista un problema similar (Basañes, 2019).

2.1.4. Interfaz de Programación de Aplicaciones

Las Interfaces de Programación de Aplicaciones (API por la sigla inglés de Application Programming Interface) son un conjunto de constantes, funciones y protocolos que permiten programar aplicaciones. Una buena API facilita la tarea de desarrollar aplicaciones, ya que facilita todas las piezas y el programador sólo tiene que unir las para lograr el fin que desea (Luján, 2002).

Las API permiten que los productos y servicios se comuniquen con otros, sin necesidad de conocer a fondo cómo están implementados. Esto simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo y dinero. Las API otorgan flexibilidad, simplifican el diseño, la administración y el uso de las aplicaciones, y proporcionan oportunidades de innovación, lo cual es ideal al momento de diseñar herramientas y productos nuevos.

2.1.5. Servicios web RESTful

Los servicios web son un conjunto de componentes independientes de software que se comunican entre sí (Fensel et al., 2006). En el desarrollo de estos servicios web uno de los esquemas que actualmente viene tomando fuerza es el REpresentational State Transfer (REST, por sus siglas en inglés: Transferencia de estado Representacional). REST es un estilo arquitectónico basado en la web con formatos de mensajes flexibles como XML y JSON.

En los servicios que emplean la arquitectura REST se establece una comunicación cliente-servidor específica, con una operación sin estado, así que no se almacena la información del cliente entre las solicitudes; cada una es independiente y está desconectada del resto. La interfaz REST es uniforme, de manera que la información se transfiere de forma estandarizada. Igualmente, cuenta con almacenamiento en caché de recursos para optimizar las interacciones entre el cliente y el servidor.

Una RESTful API o servicio RESTful web es una implementación de la arquitectura REST basada en el protocolo HTTP (Richardson & Ruby, 2008). Una mayor escalabilidad, compatibilidad, rendimiento y simplicidad son las razones comunes para elegir el estándar REST.

2.1.5.1. Hypertext Transfer Protocol

El Hypertext Transfer Protocol (HTTP por sus siglas en inglés: Protocolo de Transferencia de Hipertexto), es el nombre de un protocolo que permite realizar peticiones de datos y recursos, por ejemplo, de documentos HTML. Es la base de cualquier intercambio de datos en la Web, maneja la estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos, normalmente un navegador Web (Corporación Mozilla, 2020).

Los métodos de HTTP las peticiones van dirigidas hacia elementos llamados entidades y colecciones, en el caso de las bases de datos, las tablas equivaldrían a estas colecciones y cada registro a una entidad (Amodeo, 2013). Los principales métodos HTTP son GET para lectura de datos, PUT para la actualización de datos, DELETE para el borrado de datos y POST para la creación de datos.

2.2. MARCO REFERENCIAL

2.2.1. Importancia de los datos

La base de todo avance en la ciencia y la tecnología son los datos (Frederick, 2016), estos se ven involucrados en el desarrollo de experimentos, modelación de fenómenos, creación de teorías, etc. Los datos son de gran importancia para diversas áreas como medicina, física, ciencias naturales y sociales, economía. Es una realidad que, sin datos no habría ciencia y del mismo modo, en ausencia de ciencia no existirían datos, esto indica una urgente necesidad de que los datos de investigación pasen por procesos de curación y que existan mecanismos para que los datos estén ampliamente disponibles para la comunidad científica (Hanson et al., 2011).

2.2.2. Diluvio de datos

Antiguamente, en la era previa a los sistemas de bases de datos y los computadores, en espacios como hospitales, bibliotecas e incluso en gobiernos, se implementaron métodos para recopilar, organizar, procesar y almacenar datos (Harrison, 2015; Mansour et al., 2017), permitiendo un acceso ágil a los recursos gestionados. Sin embargo, con el progreso de las ciencias e ingenierías junto con la evolución de la tecnología, se ha llegado a que se produzcan, recopilen y analicen datos a ritmos muy elevados y a volúmenes desbordantes, fenómeno que ha sido denominado como el “diluvio de datos” (Baraniuk, 2011; C. Borgman, 2012).

Gracias a los nuevos dispositivos, software, técnicas y demás desarrollos, se ha facilitado en gran medida la obtención de datos, que son el insumo principal dentro de las investigaciones (Frederick, 2016). A partir de los datos capturados al inicio de las investigaciones, se deriva otro gran número de datos, que dentro del contexto de la investigación se convertirán en información y conocimiento. Esta situación impone nuevos desafíos frente al manejo de los datos y la información, ya que, si se pretende cosechar las recompensas del diluvio de datos, los investigadores que producen esos datos deben compartirlos y hacerlo de tal manera que los datos sean interpretables y reutilizables por otros (C. Borgman, 2012).

2.2.3. Intercambio de datos

Compartir e intercambiar datos de investigación son prácticas que hasta hace pocos años no eran muy comunes (Tenopir et al., 2011), a pesar de los diferentes beneficios que podrían existir al permitir que otros investigadores dispusieran de los conjuntos de datos de sus proyectos, eran muy pocos los campos donde esto se realizaba (C. Borgman, 2012). Incluso hasta el día de hoy, resulta común que investigadores publiquen resúmenes de datos, gráficos o cuadros sin tener disponibles los conjuntos de datos reales (Frederick, 2016).

Antes de que los datos sean compartidos, deben pasar por procesos de gestión que hagan posible su conservación a largo plazo. Una de las entidades pioneras en esta área fue la Fundación Nacional para la Ciencia de Estados Unidos (NSF, por sus siglas en inglés: National Science Foundation) (C. Borgman, 2012), si bien sus planes y estrategias de gestión de datos no iban dirigidas hacia el intercambio de datos, estas fueron un hito en el manejo de datos de investigación.

A medida que pasa el tiempo, son más los financiadores y revistas científicas que exigen que los datos subyacentes a las investigaciones publicadas sean depositados en repositorios de datos adecuados, principalmente para que los datos puedan ser citados (Alam et al., 2015). Esto hace pensar que en el futuro cercano, la publicación de datos de acceso abierto se vuelva una práctica cada vez más común, en tanto que más organismos de financiación lo exijan a sus solicitantes, las universidades desarrollen políticas sobre la publicación de datos de investigación, espacios más conocidos y más fáciles de usar para hacer que los datos de investigación estén disponibles, y que los investigadores individuales comiencen a reconocer el valor de compartir sus datos (Frederick, 2016).

Una vez que la curación y el intercambio de datos se conviertan en actividades rutinarias de los investigadores, la reutilización de los datos va a ser más frecuente haciendo posible actividades como volver a inspeccionar los propios datos para comparaciones posteriores, adquirir conjuntos de datos de fuentes públicas o privadas para compararlos con datos recién recopilados, incluir conjuntos de datos disponibles como investigación de antecedentes para un nuevo proyecto o realizar reanálisis de uno o más conjuntos de datos para abordar nuevas preguntas de investigación (Pasquetto et al., 2019).

2.2.4. Metadatos

Los metadatos son un conjunto básico de atributos descriptivos que acompañan los datos que se pretenden publicar. Para que los datos científicos sean útiles más allá de su propósito original, deben estar bien descritos, ser detectables, verificables y accesibles. Esta inclusión de los metadatos permite que los datos provenientes de un estudio o experimento sean comprendidos, replicados y utilizados en futuros estudios. (Blaiszik et al., 2016).

Sin metadatos descriptivos, los conjuntos de datos no son fácilmente localizables. Cuando los datos son encontrados por otro investigador, resulta complejo comprender su significado y condiciones bajo las cuales se capturó o procesó. Así, los datos destinados a ser publicados tienen que contar con descripciones que agilicen su descubrimiento y faciliten su comprensión en el futuro (Blaiszik et al., 2016).

La reutilización de datos reúne una serie de requisitos mínimos para ser llevada a cabo, los científicos que deseen intercambiar sus datos deberán incluir información contextual, como documentación sobre equipos, protocolos, procedimientos para recopilar y procesar datos, y condiciones experimentales o de laboratorio de manejo de datos (Culina et al., 2018), puesto que los reutilizadores posteriores de los conjuntos de datos van a buscar cualquier conocimiento necesario a través de metadatos, documentación e incluso entrando en contacto con los creadores de los datos (Pasquetto et al., 2019).

Dado que la cultura de compartir datos no se encuentra aún muy extendida, los investigadores no suelen producir metadatos, ya sea porque no cuentan con las herramientas necesarias para hacerlo, ignoran la importancia de los metadatos en la comunidad científica, no están familiarizados con los estándares o simplemente desconocen qué son los metadatos (Tenopir et al., 2011).

2.2.5. Plataformas web para el intercambio de datos

A medida que los investigadores han venido reconociendo el valor que poseen los datos y cómo la reutilización de estos puede conllevar a la producción de nuevo conocimiento, se han desarrollado diferentes soluciones en cuanto a la curación, la publicación y el intercambio de datos, además de esto se han tenido que enfrentar al denominado diluvio de datos, lo que implica la gestión de volúmenes de información colosales y tamaños de datos cada vez mayores.

El campo de la medicina es una de las áreas que se ha visto obligada a implementar estrategias para la recopilación, almacenamiento, gestión y publicación de los datos, un ejemplo de estos es el IEEG-Portal (Wagenaar et al., 2013), siendo esta una plataforma web que permite el intercambio y análisis de datos de electroencefalografías y datos neuronales, los cuales tienen tamaños del orden de Terabytes.

Este tipo de plataformas cuentan con interfaces de usuario intuitivas que facilitan a los investigadores publicar y descubrir conjuntos de datos, asimismo llevar registro de quiénes reutilizan los datos.

Para la investigación es de gran importancia gestionar adecuadamente los datos científicos desde su captura hasta su publicación, pues, agiliza diferentes tareas, facilita que existan entornos colaborativos y brinda la posibilidad de verificar los resultados de los experimentos, otorgando transparencia a la investigación. Tal es el caso del proyecto australiano ARCHER (Androulakis et al., 2009), que implementó herramientas informáticas que daban un manejo ordenado a los datos desde su recolección, donde los sensores e instrumentos de captura los enviaban directamente a un repositorio capaz de almacenar grandes volúmenes de información, en el que los científicos podían cargar y descargar conjuntos de datos.

La ingeniería de materiales es otra de las disciplinas en las que se ha hecho un esfuerzo por implementar herramientas para la publicación, descubrimiento y acceso a datos específicos de esta área. La Instalación de Datos de Materiales (MDF, por sus siglas en inglés: Material Data Facility) (Blaiszik et al., 2016), es un aplicativo cuyo enfoque es permitir que datos brutos y derivados estén disponibles a la comunidad de investigadores, del mismo modo permite compartir, descubrir y acceder a grandes y pequeños conjuntos de datos. En el desarrollo del MDF se tuvo en cuenta el papel que desempeñan los metadatos, así que, se presta especial atención en la descripción profunda de los conjuntos de datos, indicando a los posibles reutilizadores las condiciones bajo las cuales fueron recopilados los datos y la forma en que fueron procesados, además de otra información que permita el descubrimiento y comprensión de los datos.

El Centro de Datos de Materiales (MDC, por sus siglas en inglés: Materials Data Centre) (Scott et al., 2014) es una plataforma similar a la anterior que opera en función del intercambio de datos de ingeniería de materiales de diferentes tamaños y complejidades, la base de datos permite la gestión de conjuntos de datos heterogéneos acompañados de metadatos.

Los investigadores en el campo de las ciencias sociales también han creado conciencia en cuanto a la publicación de los datos y otros materiales de soporte que se ven implicados en los artículos científicos publicados. El Instituto de Ciencias Sociales GESIS-Leibniz desarrolló la plataforma gratuita datorium (Alam et al., 2015), pensada para mejorar la visibilidad de los datos y permitir su reutilización. En datorium los usuarios pueden describir y cargar sus datos en un entorno seguro, tras ser almacenados son revisado y posteriormente son publicados junto a un identificador para que los investigadores puedan buscarlos y descargarlos.

La dificultad de encontrar y recuperar datos espectrales condujo a que se creara la Base de Datos Espectral Abierta (OSDB, por sus siglas en inglés: Open Spectral Database) (Chalk, 2016). A través del uso de herramientas de código abierto se desarrolló este portal web con el fin de brindar accesos a los datos espectrales y metadatos de forma estandarizada. Gracias a su integración con otras plataformas, se cuenta con capacidades de búsqueda ampliadas, lo que maximiza la accesibilidad a los datos.

También se han construido herramientas web específicas para metadatos que no se enfocan a un área o disciplina en particular, Collaborative Open Plant Omics (COPO) es una de ellas (Etuk et al., 2019). Esta plataforma sirve como intermediación entra la construcción de metadatos y la publicación de datos siguiendo la filosofía FAIR. COPO permite a los investigadores describir conjuntos de datos de acuerdo con los estándares de la comunidad y negociar el envío de dichos datos a repositorios para que estén disponibles para otros investigadores.

Ante la ausencia de plataformas para acceso libre y que propiciaran la reutilización de datos de química sintética y analítica, se desarrolló Chemotion (Tremouilhac et al., 2020), la cual se apoya en los principios de datos FAIR. Los científicos pueden publicar y compartir sus conjuntos de datos en formatos legibles por máquina. Dentro de sus funciones, cuenta con opciones para preparación, procesamiento y reutilización de datos. Chemotion genera de forma automática identificadores de objetos digitales (DOI), para hacer que los datos sean recursos citables.

En el sector de la agricultura, el Grupo Consultor sobre Investigación Agrícola Internacional (CGIAR, por sus siglas en inglés: Consultative Group for International Agricultural Research) construyó La Red Mundial de Innovación y Aceleración de Datos de Investigación Agrícola (GARDIAN, por su sigla en inglés: Global Agricultural Research Data Innovation & Acceleration Network) (Craig, 2019), que da acceso a información y datos agrícolas producidos por 15 centros de investigación, esta también opera bajo la filosofía de datos FAIR, procurando que los datos sean localizables, accesibles, interoperables y reutilizables.

Una alternativa que ha sido usada para gestionar datos y proporcionar su acceso son las Infraestructuras de Datos Espaciales (IDE's), estructuras informáticas dedicadas a la administración de datos e información geográfica. Se han construido IDE's especializadas en el campo de la agricultura (Barrera & Libaque, 2017; Bordogna et al., 2016), que brindan apoyo a los entes administradores en la toma de decisiones.

Estas infraestructuras, cumplen con estándares internacionales, normas y demás requerimientos, con el fin de que los datos sean interoperables, en otras palabras, que estos sean manipulables en diferentes entornos de software. Adicionalmente, en algunas infraestructuras de datos espaciales, se suministran herramientas para realizar análisis y consultas de carácter espacial, permitiendo incluso la creación de cartografía personalizada agricultura (Barrera & Libaque, 2017; Bordogna et al., 2016), convirtiéndolos en sistemas de información geográfica (SIG).

En el contexto colombiano, el Ministerio de Agricultura y Desarrollo Rural (MADR) creó la Red de Información y Comunicación del Sector Agropecuario de Colombia (Agronet) (Ruiz, 2014), siendo esta una plataforma digital que integra diversos sistemas de información agropecuaria proporcionando un único punto de acceso a información sintética y oportuna para la toma de decisiones del sector productor y todos los agentes de las cadenas productivas. Agronet gestiona información estadística y geográfica, proveniente de instituciones gubernamentales, gremios, universidades y centros de investigación.

CAPÍTULO 3

3.1. METODOLOGÍA

Los proyectos de investigación en Agricultura de Precisión llevados a cabo por el trabajo conjunto del Grupo de Investigación GISMODEL y el Centro de Investigación CIBioFi han dejado como resultado un gran volumen de datos e información.

Dicha información requiere de un proceso de gestión con el fin de sea visible y por tanto pueda ser accedida y reutilizada por parte de la comunidad científica para propósitos como el apoyo para la toma de decisiones. De igual forma se puede hacer uso de los datos como punto de partida para nuevos proyectos de investigación y así elaborar nuevos productos y generar nueva información. Los datos también pueden ser aprovechados como recursos valiosos dentro de procesos académicos, entre otra cantidad de posibles usos.

Un sistema de información en la web se convierte en una herramienta conveniente para llevar a cabo ese proceso de gestión de la información. Mediante una plataforma que sea accesible desde internet se puede proporcionar el acceso a este sistema, brindando la posibilidad a la sociedad de reutilizar los datos para sus diferentes necesidades.

En esta sección, se describe el proceso metodológico adelantado para el desarrollo del sistema propuesto, que se convertirá en el medio de visibilización de los datos e información de las cuatro investigaciones de GISMODEL en el Proyecto Agricultura Inteligente.

Siguiendo el esquema de construcción tradicional de sistemas de información, se contemplan las cinco fases descritas brevemente, a continuación:

El desarrollo inicia con la fase de análisis de sistemas, donde se hace una revisión de los objetivos del sistema y de las acciones que este debe llevar a cabo para dar solución con la problemática planteada, lo que permite a su vez determinar el alcance y limitaciones que tendrá el sistema.

Luego, en una segunda fase se realiza el análisis de requerimientos, donde se establecen y describen los requerimientos del sistema, así como la definición de los módulos.

La tercera fase corresponde al diseño del sistema, donde se planea el funcionamiento del sistema a diferentes niveles. En esta fase se construyen los modelos conceptuales, lógico, de datos, cartográfico y físico del sistema.

Después del diseño viene la fase de implementación en la que se construye el sistema con base en lo plasmado en los diferentes modelos.

Como última fase, se efectúan pruebas a los componentes del sistema para verificar su correcto funcionamiento.

La Figura 1 muestra las fases del desarrollo del sistema según el modelo tradicional.

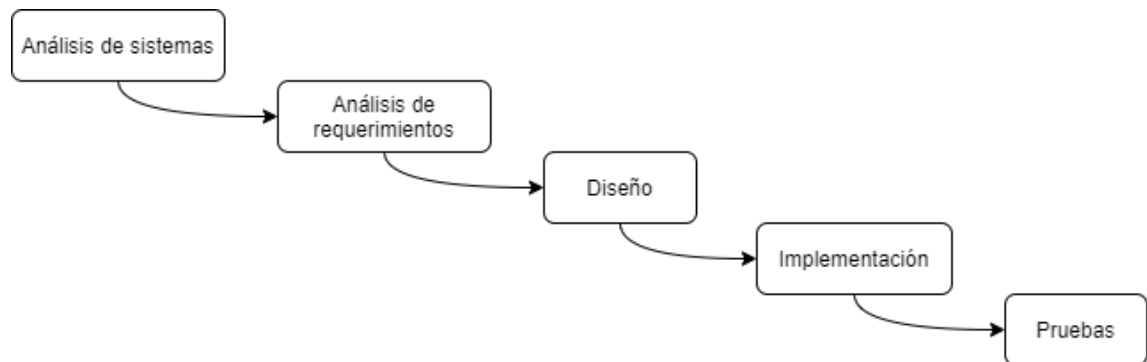


Figura 1. Esquema metodológico en cascada para el desarrollo del sistema

3.1.1. Análisis de sistemas

Esta primera fase contempla desde la comprensión del problema y la propuesta de un sistema para su solución, parte de esto ya fue tratado en las primeras tres secciones de este documento. Adicional a esto, se definen los objetivos del sistema, el alcance y las limitaciones de este.

3.1.1.1. Objetivo general del sistema

Proporcionar diferentes formas de búsqueda, localización y acceso a los datos, metadatos e información de Agricultura de Precisión producidas en los proyectos de investigación realizados por GISMODEL y CIBioFi.

3.1.1.2. Objetivos específicos del sistema

- Gestionar una base de datos con la información de los proyectos de investigación en Agricultura de Precisión.
- Brindar métodos de consulta alfanuméricos para la información disponible en la base de datos.
- Permitir realizar búsquedas de datos e información a través de criterios geográficos
- Facilitar la exploración de los datos a partir de la selección de atributos como su ubicación geográfica, la categoría de la información y características del dato.

- Filtrar resultados de búsqueda según criterios como su ubicación espacial, categoría de la información o tipo del dato.
- Presentar los metadatos de cada recurso, describiendo el contexto, condiciones y características del dato en función de facilitar su reutilización.
- Presentar una síntesis de los proyectos de investigación desarrollados por GISMODEL en el área de Agricultura de Precisión.
- Permitir la creación de cuentas de usuario para autorizar la descarga de los datos e información.
- Solicitar el motivo de descarga de cada dato albergado en la base de datos.
- Llevar un historial de los datos a los cuales accede cada usuario registrado.
- Administrar el acceso a los datos, de modo que, algunos sean de acceso inmediato mientras que otros deban requerir autorización para su descarga.
- Gestionar las solicitudes de acceso a los datos que así lo requieran.
- Proporcionar al administrador un método para la carga de nuevos datos.

3.1.1.3. Alcance del sistema

El sistema, de ahora en adelante llamado Sistema de Información del Laboratorio de Agricultura Inteligente (SILAIN) está concebido para hacer visibles los datos e información que formaron parte en el desarrollo de los proyectos de investigación de Agricultura Inteligente. Con este propósito, este aplicativo accesible desde la web, dejará a disposición de sus usuarios una serie de módulos que harán posible descubrir y acceder a los distintos conjuntos de datos de una forma fácil y rápida.

Sus cuatro módulos de localización y acceso a los datos (Búsqueda por Palabras Clave, la Exploración de Datos, el Buscador Geográfico y Proyectos de Investigación) pretenderán maximizar la accesibilidad a la información, generando diversas vías para llegar a un dato o conjunto de datos en particular.

La búsqueda por palabras clave retornará resultados relacionados con el término que haya sido ingresado. La exploración de datos no necesitará de la especificación de un término de búsqueda, sino que permitirá descubrir datos a partir de las categorías, subcategorías, ubicación geográfica y tipos de dato. El buscador geográfico permitirá de una forma muy gráfica e interactiva descubrir conjuntos de datos mediante la selección de entidades geográficas como departamento, municipio o finca, las cuales serán visibles en el mapa. La sección de Proyectos de Investigación además de ofrecer acceso a datos involucrados en estas investigaciones brindará un extracto de cada proyecto que incluirá un resumen, una breve descripción de la metodología, resultados obtenidos y conclusiones.

Al momento de acceder a cualquiera de los datos gestionados desde SILAIN, se expondrá su respectivo metadato que entregará información referente al contexto geográfico del dato, incluyendo su localización en un mapa, así como una descripción y resumen del dato que informará sobre el origen, contexto y contenido del recurso. De igual forma, se entregarán características del dato como la fecha de creación, fecha desde la cual está disponible y a qué bloque de información pertenece. También se mostrará información relativa al archivo que contiene la información, como son su tamaño y formato. Por último, se listarán las palabras clave y mediante un clic se podrá realizar una nueva búsqueda.

Como medida de seguridad, los usuarios del sistema deberán realizar un registro en la plataforma y un posterior ingreso para ser autorizados en la descarga de los datos. De acuerdo con las políticas de acceso a los datos de CIBioFi, desde el módulo de administración se podrá controlar qué recursos podrán ser descargados de forma inmediata y cuáles irán a requerir de una solicitud y posterior confirmación por parte del Centro.

A través del sistema, desde su módulo de administración se podrá cargar nuevos datos a la plataforma, diligenciando la información requerida para su correcto almacenamiento y despliegue de metadatos.

3.1.1.4. Limitaciones del sistema

SILAIN está diseñado para funcionar adecuadamente en equipos de escritorio, no estará optimizado para su despliegue en dispositivos móviles, por lo que algunos elementos de su interfaz gráfica podrían ser mostrados incorrectamente.

La búsqueda por palabras clave será altamente sensible a la cadena de texto ingresada, es decir, la correspondencia debe ser muy alta para obtener resultados acertados, si bien, esta función de búsqueda contará con un algoritmo de coincidencia aproximada donde la coincidencia del término de búsqueda y el término real podrá ser menor del 100%, no asegurará la obtención adecuada de los resultados de búsqueda.

En cuanto a la gestión de los datos, desde el sistema no será posible eliminar o editar datos o metadatos, lo cual sólo se podrá realizar manipulando directamente la base de datos.

Similar a la limitante anterior, los usuarios no podrán dar de baja su usuario en la plataforma que, de igual forma, solo se podrá efectuar desde la base de datos. No se contará con una función para la carga múltiple o masiva de datos desde el aplicativo.

El sistema no permitirá obtener una vista previa de los datos que se descargan, por lo que su contenido no será visible hasta que sea descargado.

Al momento de realizar descargas de datos con tamaños superiores a 1 GB, se podrá producir un retraso en la respuesta del sistema, que podría tardar varios segundos de acuerdo con el tamaño del recurso y la capacidad del Internet contratado por el usuario.

Aunque los metadatos serán visibles desde el sistema, no se encontrarán disponibles para su descarga.

El sistema no poseerá herramientas para la generación de reportes o informes sobre las descargas de datos realizadas por los usuarios.

3.1.2. Análisis de requerimientos

A continuación, se llevará a cabo la definición de los requerimientos y módulos necesarios para el cumplimiento de los objetivos del sistema.

Se debe reconocer la amplia gama de formatos de datos e información en sus diferentes niveles de agregación con las que cuenta el proyecto Agricultura Inteligente, los cuales pueden ser representados desde los puntos de vista espacial y temporal.

Dado que el propósito principal del sistema es proporcionar un mecanismo de localización y acceso a los datos, se piensa en la necesidad de generar herramientas que permitirán acceder de diversas maneras al dato, facilitando al usuario hacer uso de variados criterios de búsqueda que entreguen resultados efectivos sobre sus consultas. Igualmente, se proyecta el uso de un aplicativo de búsqueda geográfica para localizar datos de acuerdo con su posición.

Con el fin de mantener la trazabilidad del dato, el usuario debe ser un elemento para tener en cuenta dentro del sistema, por lo que se deberán destinar acciones para la gestión de registro de los usuarios de la plataforma y el control de los permisos que estos tienen al momento de acceder a los datos.

3.1.2.1. Definición conceptual de los requerimientos

Los usuarios del sistema deberán contar con una opción que les permita la creación de perfiles de usuario, donde se les solicitará unos datos personales y tendrán que asignar una contraseña con la que posteriormente podrán hacer inicio de sesión en el sistema.

Para cumplir con el objetivo general del sistema, este deberá proporcionar al usuario distintas herramientas para consultar la información de la base de datos, primeramente, el sistema deberá estar facultado para que retorne los datos disponibles tras haber sido ingresada una cadena de texto que contenga palabras clave. También, el usuario podrá filtrar los resultados obtenidos según diferentes criterios como el tipo de información, locación geográfica o tipo de archivo.

SILAIN deberá poder consultar la información de la base de datos sin que sea necesario especificar una cadena de texto para su búsqueda, esto se hará a través de la selección de elementos en una serie de listas desplegables posibilitando al usuario explorar la información de acuerdo con el tipo de información, la ubicación geográfica o los tipos de archivo.

Otra alternativa para la localización de datos e información en el sistema deberá involucrar un mapa, donde el usuario interactuará para hacer búsquedas a través de diferentes criterios y entidades espaciales.

Los subproyectos del proyecto de Agricultura Inteligente son el origen de los datos e información de SILAIN, así, el sistema habrá de poseer una sección dedicada a la presentación de las investigaciones adelantadas por GISMODEL-CIBioFi, incluyendo el acceso a datos ligados a cada uno de los subproyectos.

Con el fin de facilitar la reutilización de los datos, al acceder a cualquier dato en el sistema, se deberán suministrar los metadatos, indicando al usuario las condiciones y características del dato.

Previo a la descarga de cualquier dato del sistema, se deberá preguntar al usuario el motivo por el cual accederá a este recurso. Lo que permitirá registrar las razones por las cuales serán reutilizados los datos.

Cuando un usuario no haya iniciado sesión y este acceda a uno de los datos, deberá poder visualizar los metadatos, pero no podrá descargar los metadatos, se le informará que debe realizar el inicio de sesión. Así, con el usuario ingresado al sistema, se deberá llevar a cabo un registro de los datos que sean descargados por el usuario.

Es necesario que el sistema permita la carga de nuevos datos junto con sus respectivos metadatos a la base de datos.

Se requiere que el sistema cuente con la característica de habilitar o deshabilitar la descarga inmediata de los datos, es decir que, cuando esté habilitada el usuario podrá acceder al dato justo después de diligenciar el motivo de uso, mientras que, en el estado deshabilitado el usuario tendrá que solicitar el acceso al dato y la administración del Centro decidirá autorizar o denegar la descarga.

3.1.2.2. Esquema para los metadatos

El descubrimiento de los datos depende en gran medida de cómo los datos estén descritos y caracterizados. Esto se resuelve a través de los metadatos que conforman otro requerimiento para el sistema.

Dada la variedad de la información de Agricultura Inteligente con la que se cuenta, se debe establecer un modelo de metadatos general que haga posible la descripción de los datos a pesar de provenir del uso de diversos procesos, técnicas e instrumentos.

El esquema de metadatos de la Dublin Core es flexible y adaptable a diferentes propósitos, pues permite que se haga uso únicamente de los elementos que respondan a las necesidades de cada caso, de modo que no se obliga a usar la totalidad de etiquetas y elementos del estándar. De esta forma, para el sistema se considerará la adopción de un esquema de metadatos basado en las características ofrecidas por Dublin Core.

Los elementos de los metadatos deberán proporcionar información al usuario que le permita, entre otras acciones, la identificación del dato, la descripción del contenido y contexto y la ubicación geográfica del dato. Para la búsqueda y descubrimiento de los datos, se deberá contar con un elemento que contenga palabras y términos clave relacionados con el dato, como instrumentos, técnicas, zonas de estudio u otras propiedades.

3.1.2.3. Identificación de los módulos del sistema

SILAIN con su propósito de hacer que la información sea fácilmente accesible, ha de contar con un conjunto de funciones de consulta de la información y los datos, lo que conformaría el módulo de consulta. Adicionalmente, un módulo de usuarios va a permitir diferenciar los alcances y privilegios que tendrán los usuarios al momento de hacer uso del sistema. Un módulo de seguridad brindará las garantías para el uso del sistema por parte de los usuarios, así como permitir una correcta gestión de los datos y la información.

3.1.2.3.1. Módulo de usuarios

Es importante establecer los permisos con los que cuenta un usuario a la hora de utilizar un sistema, controlando así el alcance de las acciones que pudiera realizar quien interactúa con este.

Cualquier usuario que haga uso del sistema previamente el proceso de registro podrá usar funcionalidades como la consulta de información, lectura de los metadatos y exploración de los proyectos de información, pero, si previamente no ha iniciado sesión en el sistema no podrá hacer descarga de los datos obtenidos como resultados de sus consultas.

En SILAIN se tendrá una interfaz para que los usuarios lleven a cabo la creación de un perfil de usuario, lo que le concederá el privilegio de descarga de los datos presentes en la plataforma, una vez hayan iniciado sesión en el sistema. Además, los usuarios registrados van a encontrar en el panel de su perfil, un historial de los datos a los que han descargado.

Existe un tercer tipo de usuario distinto a los dos ya descritos, usuarios no registrados y usuarios registrados, este es el administrador que además de contar con los permisos del usuario registrado, ejercerá la función de decidir cuáles datos del sistema podrán ser descargados libremente y qué otros van a estar sujetos a una solicitud por parte del usuario que desee obtenerlo. Por otro lado, el administrador atenderá las solicitudes de descarga de datos, autorizando o denegando dichas peticiones. Asimismo, desde la administración se podrá hacer la carga de nuevos datos en la base de datos del sistema.

3.1.2.3.2. Módulo de consulta

La consulta de información es clave dentro de SILAIN para dar una máxima accesibilidad a los datos a través de diferentes medios, en consecuencia, el módulo de consulta debe contar con distintas herramientas de fácil uso que ayuden al usuario a descubrir o encontrar los diferentes conjuntos de datos.

Las búsquedas de datos podrán ser realizadas a través de la escritura de palabras clave, explorando entre diferentes criterios de búsqueda, apoyándose en las características geográficas de los datos o de acuerdo con los proyectos de investigación.

- **Buscador por palabras clave**

Las palabras clave tomarán parte dentro de los metadatos de cada dato del sistema, son términos que permiten relacionar cada dato con diferentes temáticas, lugares, técnicas o características.

Así, como método principal para la consulta y descubrimiento de información estará el buscador por palabras clave, en el que a través del ingreso de uno o más términos podrán obtenerse resultados relacionados con dichos términos.

- **Explorador de datos**

Debido a que no todos los usuarios tendrán conocimiento de los contenidos específicos de la información, se debe contemplar la consulta de datos mediante la selección de unos campos predefinidos que permitan identificar y filtrar los diferentes bloques de información que son gestionados con el sistema, esto, sin necesidad de ingresar cadenas de texto para la búsqueda.

- **Buscador Geográfico**

Todos y cada uno los datos de investigación poseen atributos espaciales o geográficos, es decir, están relacionados con una porción del globo terráqueo, esta condición puede aprovecharse para llevar a cabo consultas de información con la ayuda de un entorno geográfico o mapa.

El buscador geográfico se convertirá en una herramienta de búsqueda intuitiva, que permitirá localizar la información según su aspecto espacial.

- **Proyectos de Investigación**

En el Proyecto de Agricultura Inteligente, se desarrollaron cuatro subproyectos relacionados con temáticas como sustentabilidad del recurso hídrico en cultivos de arroz, estimación de la concentración de nitrógeno en cultivos de arroz, efectos del grosor de la lámina de agua en cultivos de arroz y modelación climática con técnicas de inteligencia artificial.

En el sistema habrá una sección dedicada a cada uno de estos subproyectos, presentando brevemente lo realizado en cada uno de ellos y permitiendo el acceso y descarga de datos involucrados en estas investigaciones.

3.1.2.3.3. Seguridad

En el sistema habrá mecanismos para llevar un control adecuado sobre los usuarios y permitir un óptimo manejo y gestión de los datos e información.

En primer lugar, ningún archivo de datos estará disponible para su descarga hasta tanto no se haya iniciado sesión en el sistema, esto con el fin de que se lleve seguimiento de qué usuarios hacen uso de SILAIN y qué datos descargan.

Por parte del usuario, será necesario asignar una contraseña y confirmarla por segunda vez al momento de crear la cuenta para poder ingresar al sistema.

También, existirá la posibilidad de otorgar restricción a algunos datos, esto según criterio de la administración del CIBioFi, de modo que, los usuarios deberán solicitar permiso para acceder a estos datos y por tal motivo no podrán descargarlos de forma inmediata.

3.1.3. Diseño del sistema

Esta fase del desarrollo del sistema contempla la transformación de los requerimientos del sistema en representaciones de la operatividad del sistema a diferentes niveles que van desde lo conceptual, pasando por los aspectos lógico y estructural, el modelo de la base de datos, los atributos geográficos y por último el modelo físico del sistema.

3.1.3.1. Modelo conceptual

El Sistema de Información del Laboratorio de Agricultura Inteligente (SILAIN) está pensado para ser una herramienta que ponga a disposición de la comunidad científica en general, los datos e información producidos en las investigaciones de los subproyectos de Agricultura de Precisión.

Previo al desarrollo del sistema, las condiciones de almacenamiento y gestión de los datos impedían el fácil acceso de estos, por consiguiente, su reutilización no era posible. Del mismo modo, esta situación de los datos no permitía que fueran visibles comunidades académicas o científicas ajenas a los procesos de investigación del Centro.

Con un sistema como este se proveerá un variado conjunto de opciones para poder buscar, descubrir y acceder a los datos de investigación. Al tratarse de un sistema en la web, usuarios de todo el mundo con distintas necesidades podrán reutilizar los datos disponibles en la plataforma.

A través de una interfaz de usuario amable con el usuario, este podrá realizar fácilmente las acciones para encontrar un dato que necesite o explorar la información de los proyectos.

Si el usuario cuenta con un conocimiento previo de las temáticas involucradas en las investigaciones, podrá hacer uso de un buscador de palabras clave para encontrar datos y posteriormente podrá hacer uso de filtros para descartar información y facilitar el acceso al dato precisado. En el caso de los usuarios que no estén familiarizados con los proyectos, podrán realizar exploración de los datos sin especificar términos textuales. Sacando provecho del carácter geográfico de los datos, el usuario podrá adelantar consultas por medio de un mapa, siendo esta otra forma sencilla y cómoda para localizar la información. También de forma organizada y sintética, se presentarán los cuatro subproyectos desarrollados.

Para que el usuario esté facultado para realizar descargas de datos, se le notificará que requiere de estar registrado en la plataforma, esto lo podrá hacer de forma rápida diligenciando unos datos básicos podrá crear una cuenta con la que podrá iniciar sesión y así estará autorizado para descargar los archivos de datos.

El sistema habrá interacciones que únicamente podrán ser llevadas a cabo por el administrador, estas están directamente relacionadas con la gestión de los datos.

El administrador tendrá el privilegio de decidir qué información será accesible únicamente posterior a una solicitud del usuario informando el uso que dará a tal información. Otra característica solo disponible para el administrador es la carga de nuevos datos a la plataforma.

3.1.3.1.1. Elementos del sistema

El funcionamiento y operatividad de SILAIN depende de la presencia de un amplio conjunto de elementos dentro de los cuales algunos se pueden considerar activos y otros pasivos.

El primer grupo reúne a aquellos elementos que directamente contribuyen en la resolución del problema planteado, en este caso, la gestión de los datos e información de las investigaciones en Agricultura de Precisión.

Por otra parte, los elementos pasivos que, si bien no participan directamente en la solución del problema, aportan características que mejoran la visualización de las interfaces, proporcionan información de carácter instructivo, facilitan la interpretación del sistema, entre otros aspectos importantes.

Los elementos activos o que intervienen en la solución de solución del problema corresponden a los llamados componentes básicos de los sistemas de información. Primero, se tiene el hardware o equipo físico que es utilizado para el procesamiento y almacenamientos de los datos.

Seguidamente, está el software que provee el soporte lógico del sistema, envía las instrucciones al hardware, el cual las ejecuta.

Los datos son otros de los elementos activos, cuyo acceso se pretende facilitar con este sistema, también se tiene en cuenta la estructura donde están almacenadas, es decir, la base de datos e igualmente, el gestor de la base de datos.

La infraestructura de red se constituye como otro componente, en el caso de SILAN, se busca que sea un sistema web, por consiguiente, operará desde la internet siendo accesible desde cualquier parte del mundo.

Por último, están las personas, donde toman parte los usuarios, los administradores y encargados de mantener el sistema.

Dentro de los elementos pasivos del sistema se pueden encontrar secciones como Acerca, donde se mostrarán detalles sobre el propósito del sistema, así como de su historia y autores.

Al pertenecer a una iniciativa del Centro, el sistema poseerá la opción de redireccionar a la página web de CIBioFi.

Como elementos pasivos también se pueden considerar aquellos mensajes que orientarán al usuario en la navegación sobre la plataforma, algunos de ellos serán visibles situando el cursor sobre objetos como botones o listas, otros mensajes serán proyectados al hacer uso de las utilidades del sistema o para advertir al usuario sobre datos faltantes u otras situaciones que se pudieran presentar durante la interacción.

Imágenes, logotipos y otra serie de elementos estáticos, si bien no contribuyen directamente en la solución del problema, cumplen la función de generar un entorno o interfaz armónica y amena para el usuario.

3.1.3.1.2. Operatividad del sistema

Tanto los usuarios de SILAIN como el Administrador podrán hacer uso de las cuatro utilidades de búsqueda del sistema, filtrado de resultados y acceso a los metadatos. El inicio de sesión será requisito si un usuario desea descargar uno de los datos del sistema. Algunos datos podrán estar restringidos para su descarga, exigiendo que el usuario solicite acceso a la administración.

El administrador tendrá exclusividad para efectuar ciertas operaciones en el sistema como lo son la carga de nuevos datos, la autorización de usuario para el acceso a datos restringidos y el cambio de los permisos de acceso a un dato.

En el siguiente diagrama (Figura 2) se muestra de forma general las diferentes operaciones disponibles para el administrador y para los usuarios del sistema.

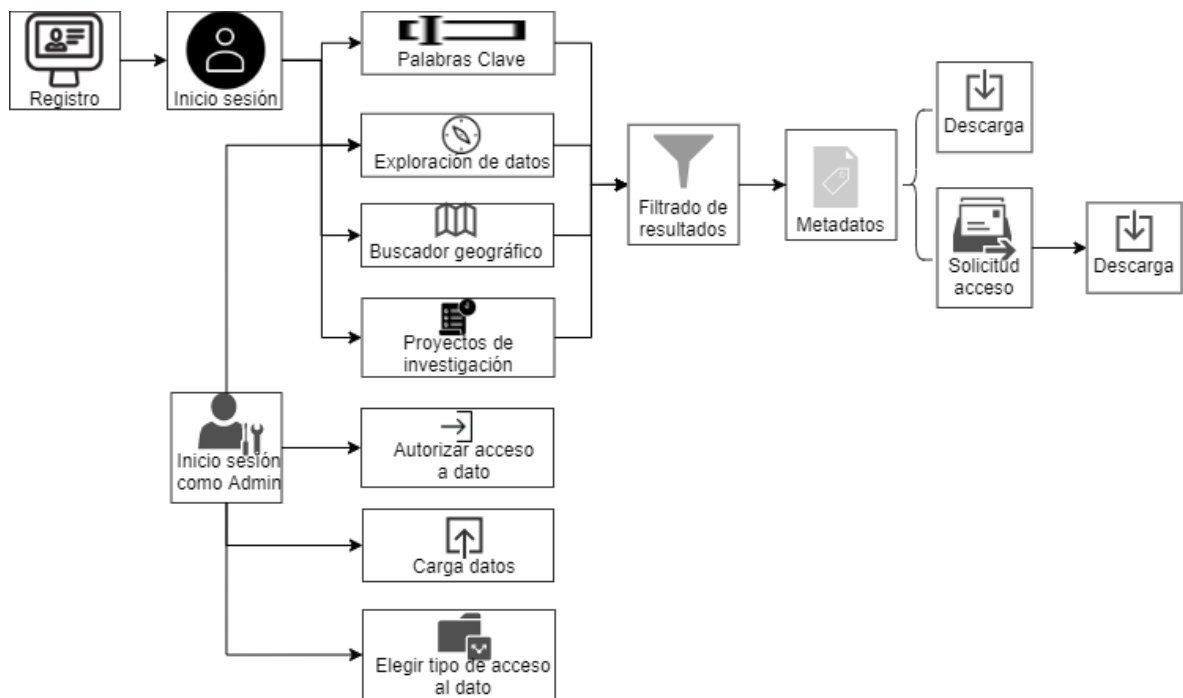


Figura 2. Diagrama del flujo de operaciones en el sistema

- Si el usuario ingresa por primera vez a la plataforma debe crear una cuenta para estar habilitado para descargar.
- Inicia sesión con la cuenta creada
- El usuario puede hacer uso de uno de los cuatro modos de consulta de datos
 - En búsqueda por palabras clave, el usuario ingresa un término de búsqueda y obtiene un listado de resultados, los que podrá filtrar y posteriormente ingresar a uno de ellos y ver sus metadatos
 - A través de exploración de datos, una vez seleccionados los campos deseados para la búsqueda, se muestran los resultados y al hacer clic en uno se presentan los metadatos.
 - Mediante el buscador geográfico, el usuario selecciona la entidad geográfica sobre la cual desea obtener resultados o hace uso de la herramienta de distancia a las fincas y escoge una de las fincas, se desplegarán los resultados y tendrá la opción de filtrarlos, posteriormente, ingresará a uno de ellos con lo que se mostrarán los metadatos.
 - En la sección de proyectos de información, se podrá leer información referente al proyecto y si se hace clic en uno de los datos relacionados se les redirigirá a los metadatos.

- Estando en los metadatos de cualquier dato, el usuario podría realizar una búsqueda nueva haciendo clic en una de las palabras clave o yendo de nuevo a una de las cuatro secciones de búsqueda, si desea acceder al dato pueden suceder dos situaciones dependiendo de si el dato tiene acceso directo o requiere autorización para descarga, en el primer caso se pedirá explicar el motivo de la descarga del dato y posteriormente podrá efectuar la descarga; si se da la segunda situación, el usuario deberá realizar la solicitud de acceso al dato vía correo electrónico, después de que la administración del centro atienda a esta solicitud, el usuario podrá continuar con la descarga.
- Ahora el usuario podrá repetir cualquiera de las anteriores acciones o ingresar al panel de usuario donde se encontrará el historial de datos descargados y si desea podría acceder por segunda vez a un dato o cerrar sesión en la plataforma.
- El administrador podrá llevar a cabo todas las acciones mencionadas previamente al igual que los demás usuarios, además de esto, al obtener el listado del resultado de la búsqueda tendrá la facultad de cambiar el tipo de acceso al dato, es decir, decidir si este estará habilitado para acceso directo o si va a requerir autorización por parte de la administración. Igualmente, en el panel de usuario se contará con una sección destinada a la carga de nuevos datos y metadatos y otra para dar autorización a usuarios que previamente hubiesen solicitado el acceso a un dato.

3.1.3.2. Modelo lógico

Con la caracterización y descripción de los módulos, elementos, y procesos necesarios para la solución del problema, según los requerimientos del sistema, se procede ahora a la representación de organización estructural por medio de la descripción del flujo de procesos y transacciones lógicas, así como de las relaciones entre elementos.

En primer lugar, se tiene el proceso de inicio de sesión (Figura 3), el cual es requerido para la descarga de datos, además permite la gestión de autorizaciones de acceso a datos y llevar un registro de las descargas realizadas.

Para acceder a las funciones previamente mencionadas, el usuario deberá iniciar sesión en la plataforma, en caso de no contar con una cuenta para ello podrá hacer el registro de su cuenta.

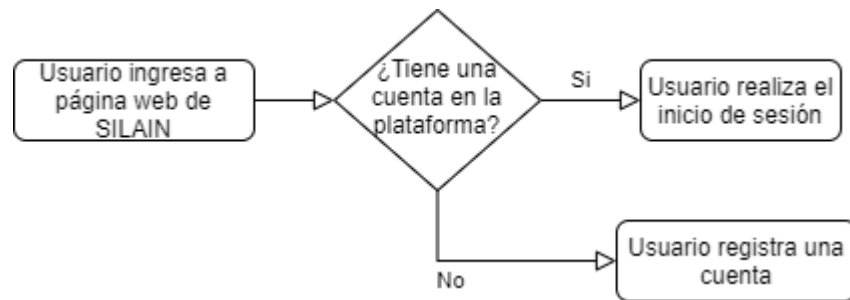


Figura 3. Diagrama lógico del ingreso a SILAIN

Los siguientes flujos de procesos corresponden a las funciones de localización y acceso a los datos con las que cuenta SILAIN que, incluyen la búsqueda por palabras clave, exploración de datos, buscador geográfico y proyectos de investigación.

La primera de estas herramientas es la búsqueda de palabras clave (Figura 4), donde el usuario que ha ingresado a la plataforma efectúa la búsqueda tras ingresar las palabras clave en la caja de texto y se presentarán una serie de resultados. Si el usuario desea, podrá obtener un resultado más específico aplicando filtros al resultado. Mientras el resultado de la búsqueda no sea satisfactorio, es posible realizar la cantidad de búsquedas adicionales que se requieran. En caso de localizar el dato esperado, podrá ingresar a este, de lo contrario podrá usar otra herramienta de búsqueda.

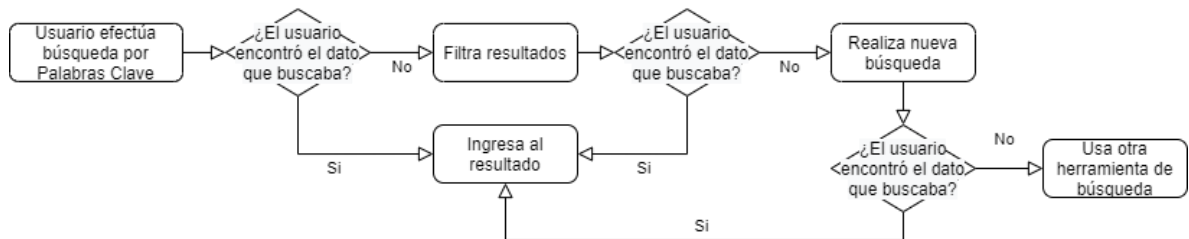


Figura 4. Diagrama lógico del uso de la utilidad Búsqueda por Palabras Clave

La exploración de datos opera de forma similar (Figura 5). El usuario elige los valores en las diferentes listas desplegables para obtener el listado de resultados que, podría filtrar según sus necesidades. Si el recurso esperado no es encontrado, se pueden modificar los valores en las listas hasta localizar el dato o podría hacerse uso de otra de las herramientas de búsqueda.

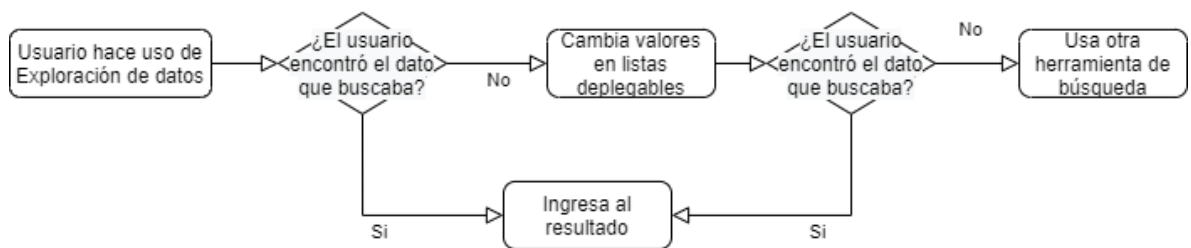


Figura 5. Diagrama lógico del uso de la utilidad Exploración de Datos

El buscador geográfico es la tercera herramienta para descubrir datos (Figura 6). El usuario podrá seleccionar entidades geográficas que serán mostradas en el mapa. Mediante un clic sobre dichas entidades podrá llevar a cabo búsquedas. En el caso de no llegar al dato preciso, puede cambiar la entidad y efectuar una nueva búsqueda. Alternativamente, el buscador geográfico cuenta con la utilidad de hallar la finca más cercana. Una vez sea localizado el dato puede acceder a él o en caso contrario puede usar otra función de búsqueda.

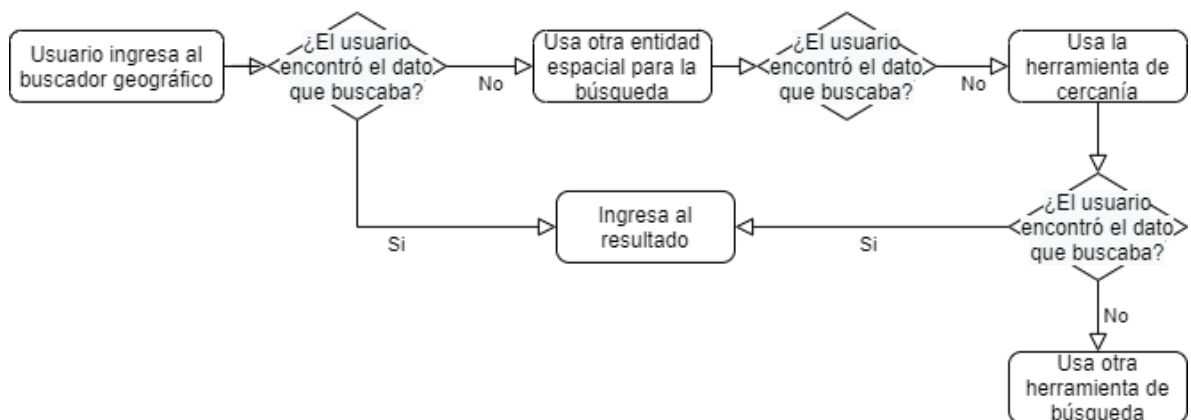


Figura 6. Diagrama lógico del uso de la utilidad Buscador geográfico

Por último, se tiene la sección de proyectos de investigación (Figura 7). El usuario podrá consultar una de las cuatro investigaciones. Si su objetivo es encontrar algún dato relacionado con el proyecto, en la sección de metodología se presentarán los datos accesibles. Si esta búsqueda no satisface sus requerimientos puede usar otra de las herramientas de búsqueda.

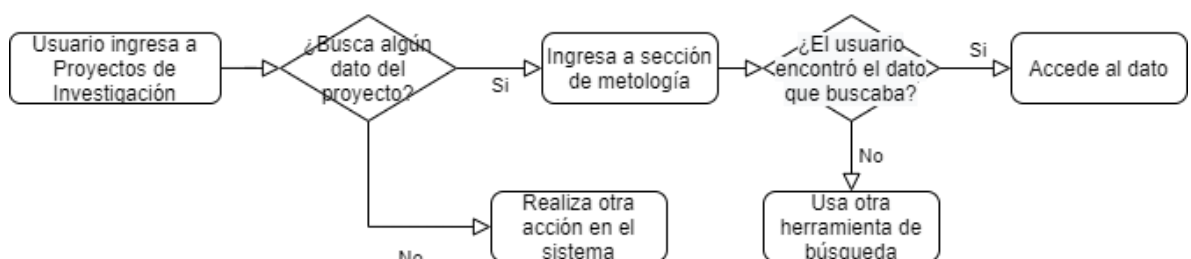


Figura 7. Diagrama lógico del uso de la utilidad Proyectos de Investigación

Ahora, cuando el usuario ingresa a un dato (Figura 8) a través de cualquiera de las funciones de búsqueda, se presentarán los metadatos y habrá un botón para la descarga del dato. Si el usuario no ha realizado el inicio de sesión, se le solicitará hacerlo, pues de otro modo, el botón de descarga no operará.

Con la sesión iniciada, si el dato cuenta con acceso inmediato, al presionar el botón de descarga se solicitará el motivo de descarga del dato. Si el acceso al dato está sujeto a autorización por parte de la administración, el botón indicará el proceso para realizar la solicitud de acceso. Cuando el acceso le sea concedido al usuario, el botón de descarga se habilitará.



Figura 8. Diagrama lógico del acceso a un dato

Otra acción que el usuario puede efectuar en el sistema es dirigirse al panel de usuario (Figura 9). En este panel encontrará el historial de descargas de datos que, le permitirá acceder nuevamente a un dato que haya descargado anteriormente. Igualmente, encontrará el botón para cerrar sesión en la plataforma.

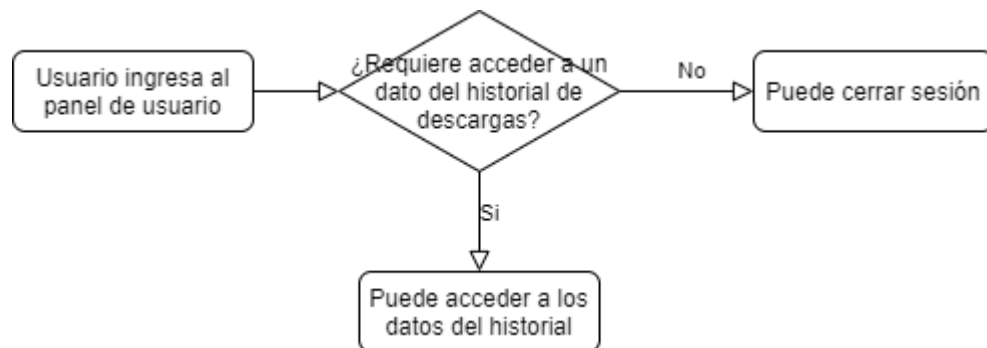


Figura 9. Diagrama lógico del uso del panel de usuario

Por otro lado, se tienen las acciones que realiza el administrador (Figura 10). Dentro de los privilegios del administrador están la modificación del estatus de acceso a un dato, es decir, cambiar de un acceso directo o inmediato a un acceso que requiera autorización y viceversa. El administrador podrá realizar esto usando cualquiera de los tres primeros métodos de búsqueda, o sea, buscador de palabras clave, exploración de datos y buscador geográfico. Desde la lista de resultados de búsqueda tendrá la libertad de cambiar mediante un botón el tipo de acceso del dato.

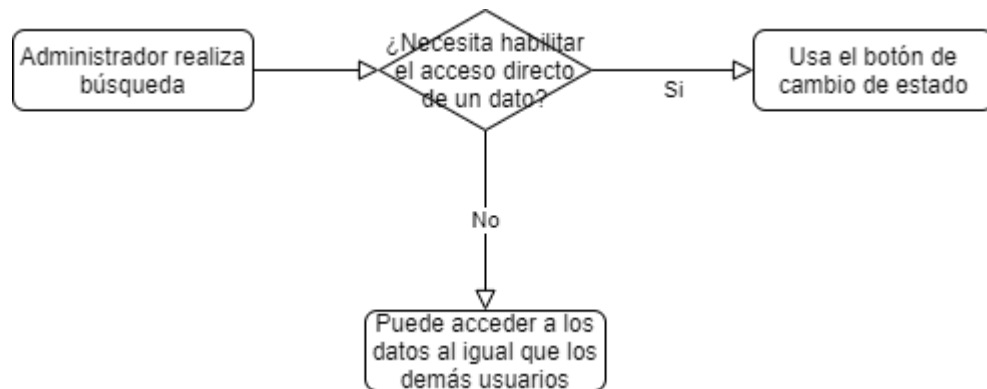


Figura 10. Diagrama lógico del uso de las utilidades de búsqueda por parte del administrador

Desde el panel de usuario, el administrador tendrá otro conjunto de utilidades exclusivas (Figura 11). La primera consiste en la carga de nuevos datos al sistema. En el panel también será posible la asignación de autorizaciones para acceso a los datos, esto para los elementos que hayan sido designados con ese estatus. Al igual que con los demás usuarios, habrá un historial de las descargas realizadas y un botón para el cierre de sesión.



Figura 11. Diagrama lógico del uso del panel de administración

3.1.3.3. Modelo de datos

La base de datos del sistema tendrá que responder a las necesidades de facilitar el acceso tanto a los datos como a los metadatos de las investigaciones, permitiendo a los usuarios conocer las características del dato, sus propiedades espaciales y demás atributos que hagan posible la reutilización de los datos.

Teniendo en cuenta que los datos están inmersos dentro de un contexto geográfico, la base de datos además de poseer elementos de carácter alfanumérico deberá tener soporte para almacenar el aspecto espacial de la información, de modo que cada dato pueda ser relacionado ya sea con una finca, un municipio o departamento.

La variedad de la información a gestionar demanda que la estructura de almacenamiento de los datos posea una organización que permita al usuario distinguir los diferentes bloques de información, diferenciando datos obtenidos por el uso de diferentes técnicas o procedimientos.

Asimismo, el sistema tendrá la capacidad de albergar cuentas de usuario, por lo que la base de datos ha de contar con elementos para la gestión de los usuarios y su acceso a los datos, esto último implica también un control sobre las autorizaciones que la administración otorgue a los usuarios que soliciten el acceso a datos.

Considerando que se tiene una estructura claramente definida con unas relaciones establecidas entre usuarios, datos, metadatos y un entorno geográfico, se opta por hacer uso en SILAIN de una base de datos relacional, para esta se construyó el siguiente modelo de datos (Figura 12), donde se presentan las entidades, atributos y relaciones que compondrán la base de datos.

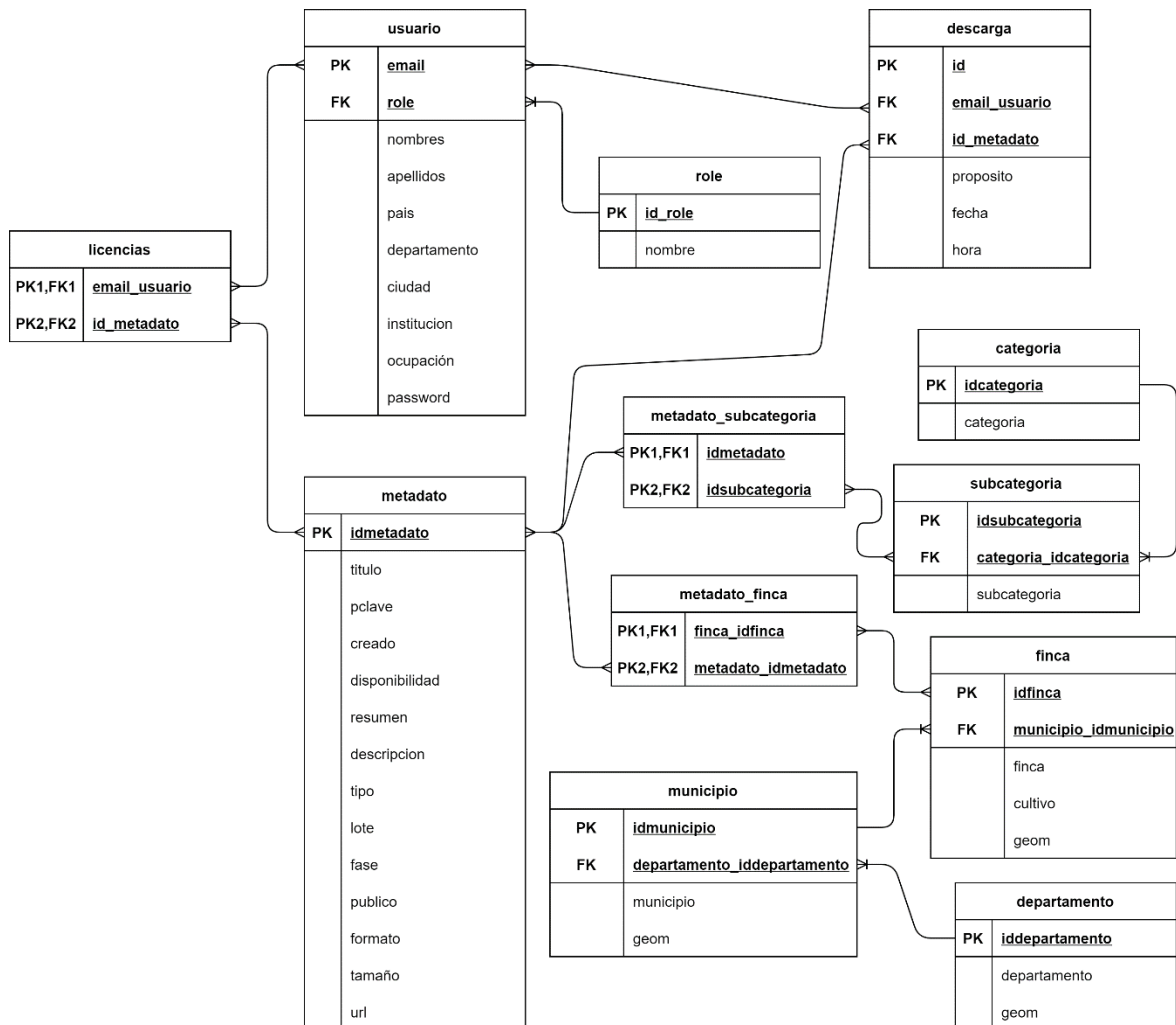


Figura 12. Modelo relacional de la base de datos

3.1.3.3.1. Tablas de la base de datos

A continuación, se presentará y describirá cada una de las tablas que componen la base de datos.

- **metadato**

Esta tabla contiene los metadatos (Tabla 2) de cada dato albergado en la base de datos, de modo que identifiquen, clasifiquen, describan y sitúen temporal y espacialmente a los datos.

Tabla 2. Tabla metadato

Columna	Tipo de dato	Dominio	Descripción
idmetadato	integer	-	Número de identificación para el dato y el metadato
titulo	character varying	-	Título del dato
pclave	character varying	-	Palabras clave del dato
creado	date	-	Fecha de creación de dato
disponibilidad	date	-	Fecha de carga del dato al sistema
resumen	character varying	-	Resumen del contexto del dato
descripcion	character varying	-	Descripción del contenido del dato
tipo	character varying	Archivo crudo Archivo procesado Imagen cruda Imagen procesada Compilación	Tipo del dato
lote	character varying	-	Lote experimental del dato
fase	character varying	-	Fase experimental del dato
publico	boolean	-	Estatus del acceso al dato
formato	character varying	-	Formato del archivo que contiene el dato
tamano	character varying	-	Tamaño del archivo que contiene el dato
url	character varying	-	Ruta de localización del dato

- **categoría**

Almacena las diferentes categorías de los datos e información (Tabla 3).

Tabla 3. Tabla categoria

Columna	Tipo de dato	Dominio	Descripción
idcategoria	integer	-	Número de identificación de la categoría
categoria	character varying	-	Nombre de la categoría

- **subcategoria**

Las categorías (Tabla 4) están divididas en subcategorías, así, esta tabla contiene las subcategorías derivadas de cada categoría. Existe una relación de uno a muchos con la tabla de categoría, por lo que cada categoría puede contar con una o más subcategorías y cada subcategoría pertenece a una única categoría.

Tabla 4. Tabla subcategoria

Columna	Tipo de dato	Dominio	Descripción
idsubcategoria	integer	-	Número de identificación de la categoría
categoría_idcategoria	integer	-	Número de identificación de la categoría de la subcategoría
subcategoria	character varying	-	Nombre de la subcategoría

- **metadato_has_subcategoria**

Esta tabla relaciona (Tabla 5) cada metadato con una subcategoría con una cardinalidad de muchos a muchos, es decir, que un metadato podrá enmarcarse en una o más subcategorías y una subcategoría puede relacionarse con uno o más metadatos.

Tabla 5. Tabla metadato_has_subcategoria

Columna	Tipo de dato	Dominio	Descripción
metadato_id_metadato	integer	-	Número de identificación del dato y metadato
subcategoria_idsubcategoria	integer	-	Número de identificación de la categoría

- **departamento**

Alberga la información alfanumérica y espacial de los departamentos relacionados con los datos (Tabla 6).

Tabla 6. Tabla departamento

Columna	Tipo de dato	Dominio	Descripción
iddepartamento	integer	-	Número de identificación del departamento
departamento	character varying	-	Nombre del departamento
geom	geometry	-	Geometría del departamento

- **municipio**

Comprende la información alfanumérica y espacial de los municipios relacionados con los datos (Tabla 7). Esta tabla se encuentra relacionada con *departamento* en una relación de uno a muchos, donde un departamento puede estar relacionado con uno o más municipios y un municipio está ligado a un solo departamento.

Tabla 7. Tabla municipio

Columna	Tipo de dato	Dominio	Descripción
idmunicipio	integer	-	Número de identificación del municipio
departamento_iddepartamento	integer	-	Número de identificación del departamento del municipio
municipio	character varying	-	Nombre del municipio
geom	geometry	-	Geometría del municipio

- **finca**

Contiene la información alfanumérica y espacial de los municipios relacionados con los datos (Tabla 8). Esta tabla se relaciona con una cardinalidad de uno a muchos con la tabla de municipio, pues, cada municipio puede albergar una o más fincas, mientras que cada finca pertenece a un único municipio.

Tabla 8. Tabla finca

Columna	Tipo de dato	Dominio	Descripción
idfinca	integer	-	Número de identificación de la finca
idmunicipio	integer	-	Número de identificación del municipio de la finca
finca	character varying	-	Nombre de la finca
geom	geometry	-	Geometría de la finca

- **metadatos_has_finca**

Esta tabla relaciona los metadatos con las fincas donde se desarrollaron las investigaciones que dieron origen los datos (Tabla 9). Cada metadato puede relacionarse con uno o más fincas. Cada finca puede estar relacionada con uno o más metadatos.

Tabla 9. Tabla metadatos_has_finca

Columna	Tipo de dato	Dominio	Descripción
metadato_id_metadato	integer	-	Número de identificación del dato y metadato
finca_idfinca	integer	-	Número de identificación de la finca

- **role**

La tabla *role* contiene los roles de usuario del sistema (Tabla 10), lo que definirá los privilegios de los usuarios y el alcance de las acciones que podrá realizar.

Tabla 10. Tabla role

Columna	Tipo de dato	Dominio	Descripción
id_role	integer	-	Número de identificación del rol de usuario
nombre	character varying	admin cliente	Nombre del rol de usuario

- **usuario**

Esta tabla almacena la información de los usuarios que se registren en la plataforma del sistema (Tabla 11). La tabla está relacionada con role, de manera que cada usuario estará ligado a un rol específico.

Tabla 11. Tabla usuario

Columna	Tipo de dato	Dominio	Descripción
nombres	character varying	-	Nombre(s) del usuario
apellidos	character varying	-	Apellido(s) del usuario
email	character varying	-	Correo electrónico del usuario
pais	character varying	-	País de residencia del usuario
departamento	character varying	-	Departamento de residencia del usuario
ciudad	character varying	-	Ciudad de residencia del usuario
institucion	character varying	-	Institución a la que pertenece el usuario
ocupacion	character varying	-	Ocupación del usuario
password	character varying	-	Contraseña de la cuenta del usuario
role	integer	1 2	Número de rol del usuario

- **descargas**

La tabla *descargas* registra información de las descargas realizadas en la plataforma (Tabla 12). Descargas se encuentra relacionada con dos tablas, que son metadato y usuario. Con metadato se mantiene una relación de 1 a muchos, donde un metadato puede estar referenciado en una o más descargas, mientras que una descarga implica la relación con un solo metadato. Por otro lado, está la relación con un usuario, nuevamente la cardinalidad es uno a muchos, pues en una descarga está implicado un usuario y un usuario puede efectuar múltiples descargas.

Tabla 12. Tabla descargas

Columna	Tipo de dato	Dominio	Descripción
id	integer	-	Número de identificación de la descarga
email_usuario	character varying	-	Correo electrónico del usuario
id_metadato	character varying	-	Número de identificación del dato y metadato
proposito	character varying	-	Propósito de la descarga y uso del dato
fecha	date	-	Fecha de la descarga
hora	time without timezone	-	Hora de la descarga

- **licencias**

En *licencias* se almacena la información concerniente a las autorizaciones de descarga de datos que se encuentren con restricciones (Tabla 13). Aquí se relacionan los usuarios con los metadatos, permitiendo que un usuario pueda ser autorizado a descargar uno o más datos restringido y que un dato restringido pueda ser proporcionado a uno o más usuarios.

Tabla 13. Tabla licencias

Columna	Tipo de dato	Dominio	Descripción
email_usuario	character varying	-	Correo electrónico del usuario
id_metadato	integer	-	Número de identificación del dato y metadato

La cardinalidad de todas las relaciones de la base de datos puede verse con mayor claridad en el siguiente diagrama (Figura 13).

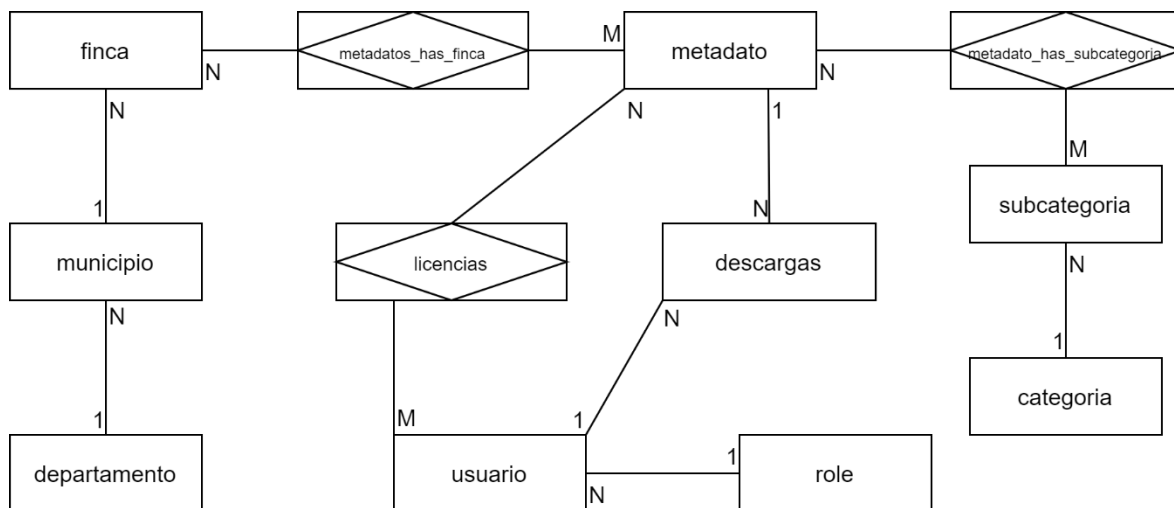


Figura 13. Cardinalidad de las relaciones

Estas relaciones entre tablas permiten que haya una sincronización durante las operaciones de actualización y eliminación, ya que, si se intenta realizar cambios o borrados de registros, la presencia de llaves foráneas genera una dependencia en las tablas que impide que esas acciones se efectúen. Del mismo modo, al momento de hacer inserciones en tablas dependientes de una principal, si la llave foránea no existe en la tabla principal no se admitirá ese nuevo registro. Estos mecanismos aseguran la integridad referencial en la base de datos.

3.1.3.3.2. Normalización de la base de datos

La base de datos está diseñada para que la información sea almacenada sin que existan redundancias y así permitir que las operaciones de mantenimiento y la recuperación de la información sea lo más sencilla posible. Para lo anterior, se aseguró que la estructura de la base de datos se encontrara en una de las formas normales (Silberschatz et al., 2002).

En cada una de las tablas descritas previamente no hay presencia de campos que repitan información similar, sino que la variabilidad de los diferentes atributos está determinada por tablas independientes que se encuentran relacionadas por sus llaves primarias. De acuerdo con esto, la base de datos cumpliría con la primera forma normal (Microsoft, 2020).

La segunda forma normal (Microsoft, 2020) implica la creación de tablas independientes para valores que estén asociados a registros de diferentes tablas, de manera que estos valores no se inserten como entradas independientes en cada una de las tablas. Esta condición fue contemplada en el modelo de la base de datos.

La no dependencia entre atributos que no sean llaves constituye la tercera forma normal (Microsoft, 2020). Este requisito se cumple en el esquema propuesto, por lo que la base de datos estaría en esta forma normal.

3.1.3.3. Auditoría de la base de datos

Adicional a las tablas presentadas en la sección anterior, la base de datos cuenta con otro conjunto de tablas que permiten realizar auditoría de cambios. Para las acciones de inserción, actualización y eliminación de datos se crearon triggers o disparadores que ejecutan funciones encargadas de registrar las diferentes transacciones realizadas sobre los datos.

En la Figura 14 se ilustra un ejemplo del proceso de auditoría que se efectúa sobre las tablas de la base de datos.

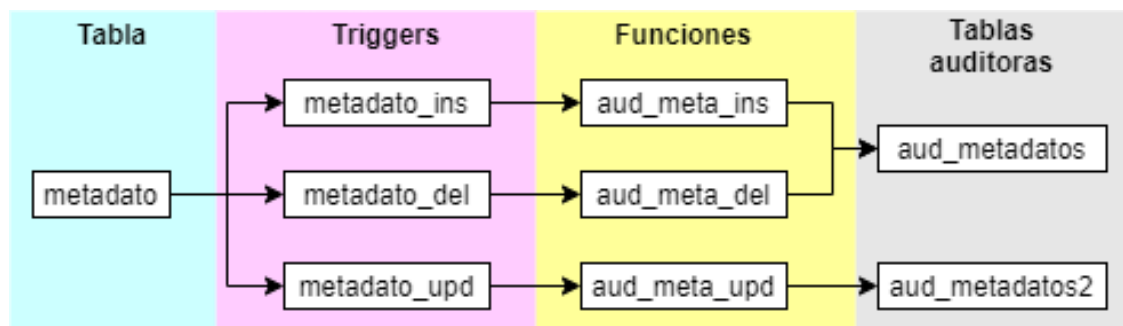


Figura 14. Auditoría de cambios en la base de datos

En el ejemplo, la tabla de metadatos cuenta con tres triggers cuyos nombres se componen del nombre de la tabla auditada y de sufijos que indican el tipo de manipulación que activa al disparador; *ins* hace referencia a inserciones en la tabla, *del* corresponde a eliminaciones y *upd* a actualizaciones.

Cada trigger activa una función específica que automatiza el registro en una tabla auditora del cambio realizado sobre la base de datos. Los nombres de las funciones contienen el prefijo *aud*, luego se hace una referencia a la tabla auditada y al final usan los mismos sufijos que los triggers.

Las inserciones y eliminaciones de datos son registradas en una tabla con el prefijo *aud* seguido del nombre de la tabla auditada, en el ejemplo *aud_metadatos*. Los cambios causados por actualizaciones son almacenados en la tabla auditora denominada *aud_metadatos2*.

Así, cada vez que se inserten datos se hará registro de esta acción. Lo mismo sucede con cualquier eliminación que sea efectuada sobre las tablas. Por otro lado, las actualizaciones son también auditadas. Esto permite llevar un control detallado de los procesos de mantenimiento de la base de datos.

De forma análoga al ejemplo, cada una de las 11 tablas restantes cuenta con tres triggers, tres funciones y dos tablas auditoras. En la Tabla 14 se exhiben los triggers, funciones y tablas auditoras para cada tabla de la base de datos.

Tabla 14. Triggers, funciones y tablas auditoras

Tabla	Triggers	Funciones	Tablas auditoras
metadato	metadato_ins	aud_metadato_ins	aud_metadato
	metadato_del	aud_metadato_del	
	metadato_upg	aud_metadato_upg	aud_metadato2
subcategoria	subcategoria_ins	aud_subcat_ins	aud_subcat
	subcategoria_del	aud_subcat_del	
	subcategoria_upg	aud_subcat_upg	aud_subcat2
categoria	categoria_ins	aud_cat_ins	aud_categoria
	categoria_del	aud_cat_del	
	categoria_upg	aud_cat_upg	aud_categoria2
metadato_has_subcategoria	metasub_ins	aud_metasub_ins	aud_metasub
	metasub_del	aud_metasub_del	
	metasub_upg	aud_metasub_upg	aud_metasub
departamento	departamento_ins	aud_depto_ins	aud_departamento
	departamento_del	aud_depto_del	
	departamento_upg	aud_depto_upg	aud_departamento2
municipio	municipio_ins	aud_mpio_ins	aud_municipio
	municipio_del	aud_mpio_del	
	municipio_upg	aud_mpio_upg	aud_municipio2
finca	finca_ins	aud_finca_ins	aud_finca
	finca_del	aud_finca_del	
	finca_upg	aud_finca_upg	aud_finca2
metadatos_has_finca	metafinca_ins	aud_metafinca_ins	aud_metafinca
	metafinca_del	aud_metafinca_del	
	metafinca_upg	aud_metafinca_upg	aud_metafinca2
role	rol_ins	aud_rol_ins	aud_rol
	rol_del	aud_rol_del	
	rol_upg	aud_rol_upg	aud_rol2
usuario	usuario_ins	aud_usuario_ins	aud_usuario
	usuario_del	aud_usuario_del	
	usuario_upg	aud_usuario_upg	aud_usuario2
descargas	descargas_ins	aud_descarga_ins	aud_descarga
	descargas_del	aud_descarga_del	
	descargas_upg	aud_descarga_upg	aud_descarga2
licencias	licencias_ins	aud_licencias_ins	aud_licencias
	licencias_del	aud_licencias_del	
	licencias_upg	aud_licencias_upg	aud_licencias2

3.1.3.3.4. Respaldo de la base de datos

Debido a múltiples causas pueden ocurrir pérdidas de información de la base de datos, entre otras, están los daños físicos en el servidor, infecciones de software maligno y operaciones malintencionadas por parte de usuarios de la web.

A raíz de lo anterior, como mecanismo de seguridad se deben hacer copias de respaldo de la base de datos. Periódicamente, a través de un script, se generará dicha copia de seguridad y esta será almacenada en un medio diferente al servidor donde estará implementado el sistema.

Con esto, se pretende subsanar cualquier problema que pudiera surgir por la eliminación parcial o total de la información en la base de datos del sistema.

3.1.3.3.5. Almacenamiento de los datos de investigación

En las secciones previas se ha mostrado el modelo de la base de datos, pero no se ha mencionado la manera en que se almacenan los datos de investigación como tal. Algunos de los datos tienen tamaños superiores a 1 GB, de modo que no fueron incluidos directamente en la base de datos, ya que podrían afectar el rendimiento de la base de datos.

En lugar de albergar los archivos de datos de investigación en la base de datos, en la tabla metadato se incluyó una columna denominada *url* para almacenar la ruta del archivo contenedor del dato dentro del servidor. Dentro de la unidad de almacenamiento se creó una estructura de directorios para la organización de la información, la cual será descrita con más detalle en la fase de implementación.

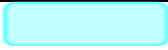
3.1.3.4. Modelo cartográfico

La información en SILAIN está referenciada espacialmente, lo que provee la capacidad para que se realicen búsquedas mediante este aspecto geográfico de los datos. La existencia de un buscador geográfico dentro del sistema exige que se definan las características de índole geográfico, que incluyen la simbología, sistema de información, escala, entre otros.

Lo primero para tener en cuenta, es que los elementos espaciales de la base de datos tienen como sistema de referencia WGS1984, que consiste en un sistema de referencia geodésico de coordenadas geográficas.

La herramienta de búsqueda geográfica permitirá la visualización de entidades administrativas como departamentos y municipios, también de fincas o zonas de estudio de los proyectos de investigación. Con el fin de distinguir cada uno de estos elementos se incluirá una leyenda dentro de la utilidad que indicará la correspondencia de las simbologías. En la Tabla 15 se muestra cada entidad y la forma en que es representada sobre el mapa.

Tabla 15. Simbología en el Buscador Geográfico

Entidad	Tipo de Vector	Color
Departamento	Polígono	 (Púrpura)
Municipio	Polígono	 (Cian)
Finca	Polígono	 (Azul)

En cuanto a la escala de las entidades, al tratarse de un aplicativo web, no puede definirse con precisión una escala de presentación puesto que habrá una dependencia del dispositivo de visualización del usuario, más precisamente de su resolución de pantalla, por otra parte, el mapa cuenta con una escala dinámica, puesto que el usuario podrá controlar el nivel de acercamiento. Sin embargo se puede establecer una relación aproximada entre metros y píxeles (Tabla 16), esto para ciertos estados estáticos del mapa, es decir, la escala justo al momento de ingresar al buscador geográfico, cuando se selecciona un departamento, cuando se elige un municipio o se escoge una finca.

Tabla 16. Escalas aproximadas en el Buscador Geográfico

Estado	Escala (m/pix)
Al iniciar el buscador	1223
Al seleccionar departamento	611
Al seleccionar municipio	153
Al seleccionar finca	2

Otra característica del buscador geográfico es la posibilidad de escoger entre tres tipos diferentes de mapas base proporcionados por OpenStreetMap (OSM), el primero es el mapa estándar OSM, el segundo es wmflabs OSM B&W (Blanco y Negro) y el tercero es el ÖPNVKarte, mapa de transporte público.

Adicional al buscador geográfico, en los metadatos habrá otro elemento cartográfico para la representación de la zona de estudio o finca a la que pertenece el dato. A diferencia del buscador, aquí solo se verá una entidad sobre el mapa, por lo que no hay leyenda. El sistema de referencia continúa siendo WGS1984, el tipo de vector sigue siendo un polígono, el color de representación es verde y la escala de visualización es de aproximadamente 2 m/pix.

3.1.3.5. Modelo físico

En el modelo físico implica la descripción del proceso mediante el cual se realiza la implementación del sistema, de modo que se planifica la construcción de la arquitectura del sistema como de la estructura de almacenamiento de los datos. De igual forma se define el hardware y software requeridos para proceder con la implementación del sistema.

3.1.3.5.1. Categorización de la información

A pesar de que en los subproyectos de agricultura de precisión se involucró el uso de variadas técnicas y metodologías para la captura, procesamiento y análisis de los datos, se llevó a cabo una identificación de unos bloques principales de información para así diseñar una estructura de agrupación y clasificación.

Uno de los bloques de información debe su origen al uso de vehículos aéreos no tripulados que portaban cámaras multiespectrales, a partir de las imágenes multiespectrales capturadas se derivaron otra serie de productos o información como ortomosaicos, índices de vegetación y modelos digitales de superficie. De forma similar, se capturaron y procesaron imágenes de cámaras RGB.

Según este mismo orden de ideas, se contaba con información obtenida mediante técnicas de topografía y geoposicionamiento que, a través de otros procesos resultó en más información.

La espectrometría fue otro de los métodos empleados en algunos de los proyectos para la captura de información.

Otro de los tipos de información presente en las investigaciones correspondía a información ambiental que se subdivide en temáticas de clima, aguas y suelos.

En el curso de las investigaciones se desarrollaron otros productos como aplicativos web, geovisores y otras herramientas que deben ser tenidas en cuenta dentro de la estructura de datos.

Como último grupo de información se encuentran los documentos escritos de los proyectos de investigación, donde dos forman parte de tesis de pregrado y los dos restantes son tesis de postgrado.

Con esta diferenciación de los diferentes bloques de información, se diseñó una estructura de categorías y subcategorías (Tabla 17) que pudieran reunir la variedad de datos mencionado y a la vez que fueran lo suficientemente claras para que los usuarios del sistema pudiesen distinguir los diferentes grupos de información.

Tabla 17. Categorización de la información

Categoría	Subcategoría
Multiespectral	Imágenes crudas
	Ortomosaicos
	Modelos Digitales de Superficie
	Índices
RGB	Imágenes crudas
	Ortomosaicos
	Modelos Digitales de Superficie
Ambiental	Clima
	Agua
	Suelos
Espectrometría	Espectrometría
Productos	Geovisores
Tesis	Pregrado
	Postgrado

Considerando el alto volumen de información y el gran tamaño de algunos de los archivos que la contienen, no convendría la inclusión de estos datos de forma directa en la base de datos, puesto que podría verse afectado su rendimiento al momento de acceder a la información a través de las consultas.

En lugar de introducir los archivos en la base de datos se resolvió almacenar los archivos que contienen la información en una estructura de carpetas para ser ubicada físicamente en un sector específico de una unidad de almacenamiento, esto de acuerdo con la clasificación diseñada previamente. Luego, la ruta o URL de cada archivo es registrada dentro de los metadatos, con el fin de poder acceder posteriormente al dato.

3.1.3.5.2. Selección de software

En cuanto al software para la gestión de la base de datos se tomó la decisión de utilizar PostgreSQL (Figura 15), esto por la utilidad que presta la extensión PostGIS (Figura 15) para la administración de información espacial, la cual es de gran relevancia dentro de SILAIN al contar los datos con referencias geográficas.



Figura 15. Sistema de gestión de bases de datos PostgreSQL y la extensión PostGIS

3.1.3.5.3. Arquitectura del sistema

Tras haber definido la conformación de la base de datos y la estrategia para el almacenamiento de los archivos de información, se pensó en la arquitectura de software para la construcción del sistema.

Las interfaces de programación de aplicaciones (API) basadas en los principios de desarrollo REST se ajustan a las necesidades de aplicaciones web ligeras, además de que son consideradas flexibles y fáciles de configurar (Red Hat, s.f.), es por este motivo que se destinó esta arquitectura para la creación de SILAIN.

La tecnología REST utiliza los diferentes métodos que ofrece el protocolo de transferencia de hipertexto (HTTP) para la solicitud de datos por parte de un cliente hacia un servidor, luego la API REST puede retornar una respuesta en diferentes formatos como HTML, XML, texto sin formato y JSON, siendo este último uno de los más comúnmente usados porque es de fácil comprensión y también es legible por cualquier lenguaje de programación (Richardson & Ruby, 2008).

Ahora bien, esta API REST puede ser considerada por si sola como una aplicación, cuya función es la de acceder a los datos del servidor, a este tipo de aplicación se le conoce como Back-End. Se requiere de una segunda aplicación para interpretar las respuestas que son retornadas por el servidor, esta se va a denominar Front-End y consiste en la interfaz del cliente o usuario.

Se podría hablar de dos capas, la primera corresponde al Front-End que será presentada al usuario en su dispositivo de navegación como una interfaz con la cual podrá interactuar a través de los diferentes módulos. La segunda capa es la REST API, que en este caso equivale al Back-End (Giménez, 2018).

Estas dos capas están en constante comunicación, así, cuando el usuario realiza alguna acción en el Front-End, este hace uso del Back-End para transmitir por medio del protocolo HTTP las peticiones al servidor que a su vez se comunica con la base de datos, luego el servidor retorna a la API una respuesta en formato JSON, por ejemplo, y aquí el Front-End vuelve a interactuar con el Back-End para tomar la respuesta y representarla según como haya sido programada. En la Figura 16 se esquematiza esta arquitectura.

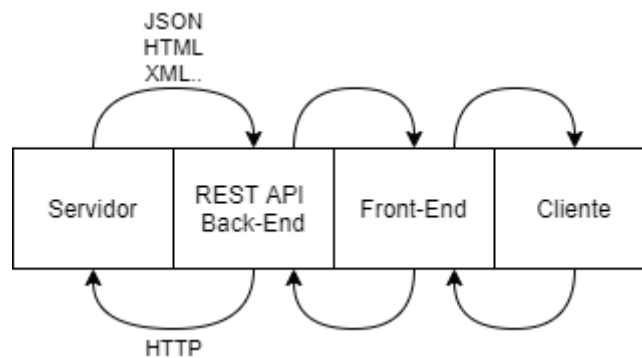


Figura 16. Arquitectura del sistema

Este modelo de diferenciación de capas permite que la API REST opere de forma independiente de manera que, si se quisiera, podrían desarrollarse otros servicios que consuman al Back-End para acceder de manera simultánea los datos del servidor.

Para el desarrollo del Back-end se usó un entorno de ejecución de código abierto llamado NodeJS (Figura 17), basado en el lenguaje de programación JavaScript, este se destaca por permitir una construcción rápida y escalable de aplicaciones de red, capaz de soportar múltiples conexiones simultáneas con un gran rendimiento.



Figura 17. Entorno de ejecución NodeJS

Se definió el uso de ReactJS (Figura 18) para la construcción de la capa del Front-End, se trata de una biblioteca de JavaScript, igualmente de código abierto, destinada a la creación de interfaces de usuario. Facilita la creación de vistas simples para la visualización de los datos, se basa en componentes encapsulados que manejan estados propios.

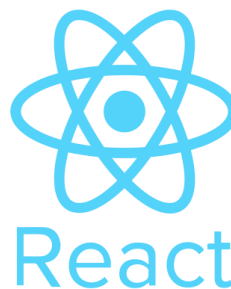


Figura 18. Biblioteca ReactJS

JavaScript cuenta con un gestor de paquetes llamado Node Package Manager (NPM por sus siglas en inglés: Administrador de paquetes de Node). NPM viene incluido con la instalación de NodeJS. Con este gestor se facilita agregar dependencias, distribuir paquetes y administrar los proyectos (Npm Inc, s.f.).

El componente cartográfico se desarrollará con otra herramienta de código abierto llamada Leaflet (Figura 19), esta es una librería de JavaScript capaz de permitir la creación de mapas interactivos. Leaflet es simple, fácil de usar y ofrece buen rendimiento.



Figura 19. Librería Leaflet

Ya definido el componente de software resta establecer cuál es el software requerido para la implementación. En primer lugar, es preciso contar con un servidor físico con conexión a internet, que disponga de un sistema operativo apto para ejecutar el sistema de gestión de bases de datos. Las características requeridas de este servidor están centradas en tres aspectos, la memoria de almacenamiento, memoria de acceso aleatorio (RAM) y procesador.

Para un óptimo funcionamiento del API REST, se espera contar con una memoria RAM de 2GB, mientras que para el almacenamiento de la información de los subproyectos de agricultura de precisión se necesita como mínimo 200 GB de espacio disponible.

3.1.4. Implementación del sistema

En la fase anterior se elaboró el diseño del sistema a nivel conceptual, lógico, cartográfico, físico y de la base de datos. Ahora, en la implementación se realiza lo establecido en las estrategias propuestas en el diseño, desde la organización de la información y la creación de la base de datos, hasta la creación de las capas del Back-End y el Front-End y su despliegue en el servidor.

3.1.4.1. Organización de la información

La información de los subproyectos de Agricultura Inteligente se encontraba almacenada de forma dispersa en varios dispositivos de almacenamiento, así que la primera actividad en cuanto a la organización de la información fue la recopilación en un único medio de almacenamiento.

Después de contar con la información en un único dispositivo, se procedió a la búsqueda de información duplicada, tras esto se continuó con la segmentación de la información de acuerdo con las categorías y subcategorías propuestas. Se contaba con algunos de los itinerarios de vuelos de los UAV, lo que facilitó la organización de este bloque de información.

Además de las categorías y subcategorías, la información estaba referida geográficamente a una finca como zona de estudio, al mismo tiempo estas fincas se sitúan en un municipio y estos últimos se ubican en un departamento. Igualmente, en algunas de las fincas se tenía más de un lote de experimentación para las investigaciones. Esto permitió incluir otros niveles de desglose de la información.

Finalmente, los ciclos fenológicos de los cultivos se convirtieron en otra manera de separar de forma ordenada los datos, por lo tanto, se cuenta con bloques de información discriminados según distintas fases.

En la Tabla 18 se muestran las diferentes zonas de estudio de las investigaciones que dieron origen a la información.

Tabla 18. Organización de la información según zona de estudio y fase

Departamento	Municipio	Finca	Lote	Fase
Valle Cauca	del Jamundí	Calderón	-	-
		Edén	-	-
			Samanes	1
				2
				3
				4
				5
				6
		Eucadia	-	-
	Rozo	-	-	-
Cauca	Santander de Quilichao	San José	La Isla	1
				2
				3
				4
			Lote 8	1
				2
				3
				4
			Lote 4	-
			Islita	-

Las celdas de la tabla con guión (-) indican que no se tenía información ya sea del nombre de la finca o del lote, y en el caso de las fases que la información no se recolectó con tal periodicidad.

3.1.4.2. Creación de la base de datos

Como se mencionó en la fase del diseño, se empleó el software PostgreSQL para la construcción de la base de datos relacional, la cual contaba con información de carácter alfanumérico y espacial.

Según el modelo de datos, se procedió a la creación de la base de datos llamada *silain* y luego de esto a la escritura de las sentencias SQL para la creación de las tablas, relaciones, triggers y funciones requeridas para la gestión de la información (Figura 20).

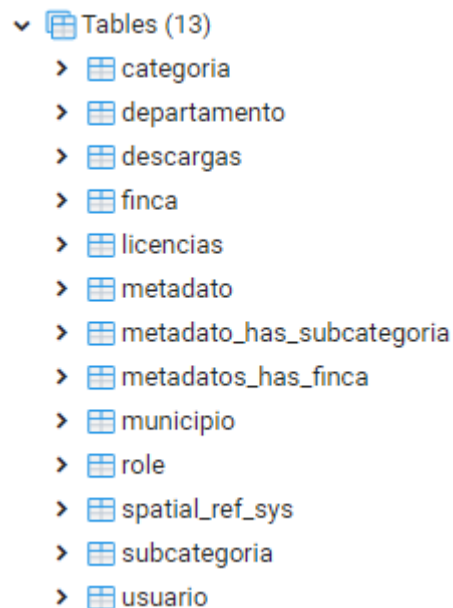


Figura 20. Tablas de la base de datos

En esta base de datos se dispone de 13 tablas. La tabla metadato se construyó para contener, como su nombre indica, los metadatos o descripción de los atributos y características de los datos y la ruta de estos dentro de la unidad de almacenamiento del servidor.

La ubicación geográfica de los datos se ve representada por las tablas de departamento, municipio y finca. Una tabla permite la relación entre metadatos y fincas con una cardinalidad de muchos a muchos, con el fin de que exista la posibilidad de relacionar un dato con distintas zonas de estudio.

La clasificación de los datos está regida por las tablas de categoría y subcategoría. Al igual que en el caso de las fincas, se tiene una tabla para relacionar con una cardinalidad de muchos a muchos, los metadatos con una o más subcategorías, y al mismo tiempo las subcategorías se ven contenidas dentro de categorías.

Una tabla de usuarios permite llevar el registro de los usuarios del sistema, lo que también asiste con la asignación de permisos para el acceso a datos que cuenten con alguna medida de restricción.

La tabla de rol hace posible la diferenciación entre los usuarios cliente y el administrador, ya que este último cuenta con privilegios exclusivos dentro del sistema. En esta tabla, el rol denominado *admin* corresponde al usuario administrador, mientras que *cliente* será el rol que se asociará a cualquier otro usuario que cree una cuenta en la plataforma.

Con el fin de llevar una trazabilidad del dato, se destinó la tabla descargas para el registro de los accesos que se hagan a los datos que reposan en SILAIN.

Como se ha mencionado en varias ocasiones, el administrador tiene la potestad de restringir el acceso a los datos, obligando a los usuarios a realizar una solicitud para el acceso. La tabla licencias gestiona las autorizaciones que haya emitido el administrador para conceder a un usuario el acceso a un determinado dato.

La tabla *spatial_ref_systems* es creada luego de habilitar la extensión POSTGIS que permite la inclusión de información espacial dentro de la base de datos. Las tablas con atributos espaciales en esta base de datos son departamento, municipio y finca. Esta condición se evidencia a través de una columna de tipo geometría presente en cada una de esas tablas.

Luego de haber sido creadas las tablas se realizó la inserción de los datos previamente consignados en archivos de texto plano, de modo que ya quedaría lista la base de datos para ser consultada.

Para auditar cualquier cambio efectuado en los registros de la base de datos, se asociaron los triggers, funciones y tablas auditoras propuestas en el modelo de datos.

En total se implementaron 36 triggers (Figura 21). Cada tabla está asociada a tres disparadores, para las acciones de inserción, eliminación y actualización.

→ categoria_del	→ licencias_del	→ municipio_del
→ categoria_ins	→ licencias_ins	→ municipio_ins
→ categoria_upd	→ licencias_upd	→ municipio_upd
→ departamento_del	→ metadato_del	→ rol_del
→ departamento_ins	→ metadato_ins	→ rol_ins
→ departamento_upd	→ metadato_upd	→ rol_upd
→ descarga_del	→ metafinca_del	→ subcategoria_del
→ descarga_ins	→ metafinca_ins	→ subcategoria_ins
→ descarga_upd	→ metafinca_upd	→ subcategoria_upd
→ finca_del	→ metasub_del	→ usuario_del
→ finca_ins	→ metasub_ins	→ usuario_ins
→ finca_upd	→ metasub_upd	→ usuario_upd

Figura 21. Triggers implementados en la base de datos

Para el registro de cambios se crearon dos tablas auditoras por cada una de las tablas de la base de datos (Figura 22), una el seguimiento de las inserciones y eliminaciones y la otra para las actualizaciones.

aud_categoria	aud_metafinca
aud_categoria2	aud_metafinca2
aud_departamento	aud_metasub
aud_departamento2	aud_metasub2
aud_descargas	aud_municipio
aud_descargas2	aud_municipio2
aud_finca	aud_rol
aud_finca2	aud_rol2
aud_licencias	aud_subcategoria
aud_licencias2	aud_subcategoria2
aud_metadato	aud_usuario
aud_metadato2	aud_usuario2

Figura 22. Tablas auditoras implementadas en la base de datos

También, se construyeron 36 funciones para automatizar el llenado de las tablas auditoras (Figura 23). Cada vez que un trigger es accionado, una de las funciones es ejecutada para realizar una inserción en la tabla auditora.

<code>{aud_cat_del() }</code>	<code>{aud_licencias_del() }</code>	<code>{aud_mpio_del() }</code>
<code>{aud_cat_ins() }</code>	<code>{aud_licencias_ins() }</code>	<code>{aud_mpio_ins() }</code>
<code>{aud_cat_upd() }</code>	<code>{aud_licencias_upd() }</code>	<code>{aud_mpio_upd() }</code>
<code>{aud_depto_del() }</code>	<code>{aud_metadato_del() }</code>	<code>{aud_rol_del() }</code>
<code>{aud_depto_ins() }</code>	<code>{aud_metadato_ins() }</code>	<code>{aud_rol_ins() }</code>
<code>{aud_depto_upd() }</code>	<code>{aud_metadato_upd() }</code>	<code>{aud_rol_upd() }</code>
<code>{aud_descarga_del() }</code>	<code>{aud_metafinca_del() }</code>	<code>{aud_subcat_del() }</code>
<code>{aud_descarga_ins() }</code>	<code>{aud_metafinca_ins() }</code>	<code>{aud_subcat_ins() }</code>
<code>{aud_descarga_upd() }</code>	<code>{aud_metafinca_upd() }</code>	<code>{aud_subcat_upd() }</code>
<code>{aud_finca_del() }</code>	<code>{aud_metasub_del() }</code>	<code>{aud_usuario_del() }</code>
<code>{aud_finca_ins() }</code>	<code>{aud_metasub_ins() }</code>	<code>{aud_usuario_ins() }</code>
<code>{aud_finca_upd() }</code>	<code>{aud_metasub_upd() }</code>	<code>{aud_usuario_upd() }</code>

Figura 23. Funciones implementadas en la base de datos

Otra característica de la base de datos es la adición de la extensión *pg_trgm*. Esta utilidad de PostgreSQL permite que, al momento de realizar consultas con cadenas de texto, se obtengan resultados a pesar de que el parámetro entregado no coincida exactamente con el valor en el la base de datos.

3.1.4.2.1. Configuración del esquema de metadatos

Si bien existe una tabla en la base de datos llamada *metadato*, por sí sola no contiene todos los elementos descriptores de los datos. La categorización del dato y la ubicación geográfica del dato están contenidas en tablas diferentes que se encuentran relacionadas con *metadato*.

Haciendo la unión de las tablas que conforman los metadatos se obtiene el siguiente esquema (Tabla 19).

Tabla 19. Esquema de metadatos en la base de datos

Tabla	Columna	Elemento del metadato
metadato	idmetadato	Identificador
metadato	titulo	Título
metadato	pclave	Palabras Clave
metadato	resumen	Resumen
metadato	descripcion	Descripción
categoria	categoria	Categoría
subcategoria	subcategoria	Subcategoría
metadato	tipo	Tipo
metadato	creacion	Fecha de creación
metadato	disponibilidad	Fecha de subida
finca	finca	Finca
municipio	municipio	Municipio
departamento	departamento	Departamento
metadato	formato	Formato
metadato	tamano	Tamaño
metadato	publico	Tipo de acceso
metadato	url	URL

De esta forma, el metadato está compuesto por 17 elementos diferentes que se integran a partir de seis tablas.

- **Identificador**

El identificador es el primer elemento del metadato, consiste en un número entero que es exclusivo de cada dato, de manera que permite distinguir sin lugar a equivocaciones los datos en el sistema.

- **Título**

Es el nombre dado a cada uno de los datos, es presentado en los resultados de búsqueda. Permite crear una noción sobre el contenido del dato. Pueden existir datos con nombres similares que solo varían por aspectos como parámetros experimentales, la localización o fase, entre otros; por ello no reemplaza al identificador para la distinción de los datos.

- **Palabras Clave**

Las palabras clave conforman un elemento de gran importancia para el descubrimiento de los datos. Son palabras o términos relacionadas con el dato, pueden incluir lugares, fases experimentales, herramientas, procesos, técnicas, temáticas y demás. Son usadas como entradas por la primera de las herramientas de búsqueda del sistema para generar las consultas de información.

- **Resumen**

Consiste en un escrito corto que menciona los aspectos básicos del dato, información contextual del dato. Puede incluir información sobre la captura o procesamiento del dato.

- **Descripción**

Complementa al anterior elemento, suministrando información más específica del contenido del dato. Aborda atributos o características propias de cada dato, de acuerdo con el tipo de archivo que lo contenga.

- **Categoría**

Permite clasificar al dato en una de las categorías propuestas en la Tabla 5. Es útil también para la aplicación de filtros al momento de localizar datos dentro del sistema.

- **Subcategoría**

Es otro nivel de clasificación más específico para los datos. Las subcategorías pueden observarse en la Tabla 5.

- **Tipo**

El elemento tipo actúa como otro mecanismo para clasificar los datos y facilitar el filtrado de resultados. Los tipos de datos que están contemplados en el sistema son: Archivos crudos, Archivos procesados, Imágenes crudas, Imágenes procesadas y Compilaciones.

- **Fecha de creación**

Es la fecha en que se produjo el dato, ya sea porque el dato haya sido capturado en campo o experimentalmente o porque el dato sea resultado de algún proceso o transformación de otro dato.

- **Fecha de subida**

La fecha de subida corresponde a la fecha en que el dato fue cargado en la plataforma.

- **Finca**

Los datos del sistema están asociados a fincas donde fueron desarrollados los proyectos de investigación. Así, este elemento informa la finca relacionada con el dato. Este elemento interviene también en la exploración y el filtrado de los datos, además es una de las entidades administrativas a través de las cuales se puede encontrar datos en el buscador geográfico.

- **Municipio**

Dado que las fincas están distribuidas en diferentes municipios, este elemento relaciona cada finca con el municipio donde está ubicada. Al igual que la finca, hace parte de los filtros y de la herramienta del buscador geográfico.

- **Departamento**

Como su nombre indica, es el departamento al que pertenece el municipio donde se sitúa una finca. Es otra de las entidades administrativas para efectuar búsquedas geográficamente.

- **Formato**

Hace referencia al formato del archivo del dato. Este elemento es útil también para el filtrado de resultados y el descubrimiento de datos.

- **Tamaño**

El elemento tamaño equivale al espacio en almacenamiento que ocupa el archivo contenedor del dato.

- **Tipo de acceso**

Este elemento permite conocer si el dato puede ser accedido y descargado sin ninguna restricción o si, por el contrario, su descarga requiere ser autorizada por parte de la administración.

- **URL**

La url es la ruta que señala la ubicación del archivo que contiene el dato, dentro de la unidad de almacenamiento del servidor. Es utilizada implícitamente para realizar la petición al servidor al momento de hacer una descarga desde la interfaz de usuario.

Como se puede apreciar, los metadatos además de identificar, clasificar y describir las propiedades y atributos de los datos desempeñan funciones importantes en la búsqueda, localización y exploración de los datos. Son utilizados en las diferentes herramientas del sistema para filtrar información y facilitar la búsqueda y exploración de los datos.

3.1.4.3. Back-End

Implementada la base de datos, se debe crear la capa encargada de comunicarse con el servidor para realizar las peticiones de acceso a los datos. En el diseño se definió el uso de una arquitectura REST API para el desarrollo de este componente del sistema. De igual forma, en el modelo físico se determinó el uso de NodeJS para construir la API.

Inicialmente se debe establecer una conexión a la base de datos especificando parámetros como el usuario en el sistema de gestión de la base datos, la contraseña, el host, el nombre de la base de datos y el puerto.

Luego se comienza por codificar todas las peticiones que se harán al servidor a través del protocolo HTTP, que permiten principalmente recuperar, enviar o eliminar datos. NodeJS permite hacer esto de forma muy sintética, mediante la creación de rutas.

Cada ruta implica un método de petición enviado a través de un EndPoint o URL para realizar una determinada acción. Con la finalidad de mantener una mejor organización del código, las rutas se agruparon de acuerdo con diferentes secciones del sistema.

El primer grupo de rutas denominado *article.js* reúne una serie de peticiones relacionadas con los metadatos. Entre ellas están la de recuperar todo el metadato a través de su ID o identificador, esto incluye la información espacial asociada al metadato, es decir, la finca. También controla las operaciones concernientes a la creación de nuevos datos.

basicSearch.js es la segunda agrupación de rutas, comprende un conjunto de peticiones destinadas a obtener los resultados de una determinada búsqueda de datos mediante las diferentes utilidades de búsquedas proporcionadas.

Relacionada con el grupo anterior se encuentra el grupo *filter.js* cuyo propósito es el de brindar la función de filtrado de resultados, tanto para la búsqueda por palabras clave, como para la exploración de datos y el buscador geográfico.

La asignación de autorizaciones para el acceso a datos se ve controlado por el grupo *licence.js*, por tanto, recurre al método de creación de datos para conceder las licencias o autorizaciones.

El administrador podrá hacer que un dato que tenga acceso sin restricción sea modificado para que requiera autorización para su descarga, e inversamente podrá quitar las restricciones a un dato. Estas acciones están contenidas en *makepublic.js*.

Las operaciones que implican peticiones de información espacial se ven agrupadas en *map.js*. Entre ellas se tiene la recuperación de las geometrías de los departamentos, municipios y fincas para ser representadas como polígonos sobre el mapa, igualmente la búsqueda a través del clic sobre una de esas entidades desplegadas en el buscador geográfico y la utilidad del buscador de finca más cercana.

El último grupo de rutas, *users.js*, se centra en las funcionalidades del módulo de usuarios. Se incluyen las peticiones para llevar a cabo el registro de usuarios, realizar el inicio de sesión, mostrar la información del usuario incluyendo el historial de descargas.

Con lo anterior, el sistema ya posee el motor para entablar comunicación con el servidor y acceder a la base de datos para recuperar, modificar o crear información en esta.

Los códigos creados para la construcción del Back-End se pueden encontrar en el Anexo 1.

3.1.4.4. Front-End

En vista de que la capa del Back-End ya se encuentra construida, hace falta de desarrollar la interfaz que permitirá la interacción del usuario con el sistema, es decir, el Front-End. Esta parte del software se implementó con la librería ReactJS.

ReactJS se caracteriza por permitir la creación de interfaces de usuario a través de componentes reactivos, los cuales operan de forma independiente para mostrar los elementos funcionales de los que hace uso el usuario como lo son botones, listas, formularios y demás, asimismo cada componente está programado para representar los datos recibidos por Back-End según los estilos de visualización asignados.

El desarrollo por componentes reactivos trae consigo una estructura de codificación más ordenadas, ya que se permite segmentar el código y puede hacerse reutilización de los componentes. Cuando se realizan las peticiones y se retorna la información, solo se actualiza el componente asociado, o sea que no se produce una actualización de la página en su totalidad.

Los componentes en SILAIN están inmersos en otras estructuras llamadas vistas, una vista sería equivalente a una página o pantalla en el navegador. Cada vista puede contener uno o más componentes y estos componentes pueden estar presentes en varias vistas.

En total SILAIN cuenta con 10 vistas, mostradas en la Tabla 20.

Tabla 20. Vistas de la interfaz de usuario

Vista	Descripción
Acerca.jsx	Muestra información general (Autor, Origen, Propósito, etc.) acerca del Sistema de Información del Laboratorio de Agricultura Inteligente.
Registro.jsx	Contiene el formulario empleado para la creación de cuentas de usuario en la plataforma.
Login.jsx	Presenta el formulario con el cual se hace inicio de sesión en el sistema.
Start.jx	Es la pantalla principal de SILAIN, donde se muestra el logo y permite hacer uso de la búsqueda por palabras clave.
Clave.jsx	Corresponde a otra vista para acceder a la utilidad de búsqueda por palabras clave, sin mostrar el logo de SILAIN.
Propiedad.jsx	Esta vista es desplegada cuando se utiliza la exploración de datos, es decir, sin el uso de palabras clave.
Proyectos.jsx	Brinda el acceso al resumen de cada uno de los cuatro subproyectos de Agricultura de Precisión.
TheMap.jsx	Exhibe el método de búsqueda llamado buscador geográfico.
Article.jsx	Ofrece la visualización de los metadatos al ingresar a uno de los resultados de búsqueda.
-Perfil.jsx	Permite visualizar la sección del perfil del usuario, la cual contiene los datos del usuario y el historial de descargas.

Por otra parte, están los componentes, cada uno con funciones específicas y que pueden estar contenidos en más de una vista, a continuación (Tabla 21), se listan los componentes del sistema.

Tabla 21. Componentes de la interfaz de usuario

Componente	Descripción
CreateFile.jsx	Este componente proporciona el formulario que permite al administrador la carga de nuevos datos y la creación de sus metadatos.
CreateLicence.jsx	El administrador lo puede usar para generar la autorización de acceso a un dato restringido a un usuario.
CustomFilters.jsx	Contiene las listas desplegables para realizar el filtrado de los resultados de búsqueda.
Footer.jsx	Corresponde al Footer o Pie de página, desde la cual se puede acceder a la sección Acerca y a la página web de CIBioFi.
Pagination.jsx	Muestra la paginación de los resultados, es decir, la numeración de las diferentes páginas con los resultados de búsqueda.
ResultsTable.jsx	Presenta la tabla con los resultados obtenidos tras haber realizado un proceso de búsqueda.
SNavBar.jsx	Comprende la barra de navegación mostrada en la parte superior de la plataforma, para acceder a las utilidades de búsqueda, ir al inicio de la página o ingresar al panel de usuario.
CustomFiltersByMap.jsx	Contiene las listas desplegables para realizar el filtrado de los resultados al usar el buscador geográfico.
FiltersOnMap.jsx	Expone las listas empleadas para desplegar en el buscador geográfico las entidades de Departamento, Municipio y Finca.
PropiedadByMap.jsx	Este componente agrupa los dos componentes anteriores y añade la cantidad de resultados de acuerdo con el tipo de dato.
ResultsTableByMap	Presenta la tabla con los resultados obtenidos tras haber realizado hecho uso del buscador geográfico.
Project1.jsx	Da acceso al resumen de la Investigación sobre Sustentabilidad Ambiental del Recurso Hídrico en Cultivos de Arroz.
Project2.jsx	Da acceso al resumen de la Investigación sobre el Efecto de la Lámina de Agua en Cultivos de Arroz.
Project3.jsx	Da acceso al resumen de la Investigación sobre Modelación Climática con Técnicas de IA.
Project4.jsx	Da acceso al resumen de la Investigación sobre Estimación de la Concentración de Nitrógeno en Cultivos de Arroz.

Como se mencionó, las vistas integran varios componentes, aunque en algunos casos las vistas están contenidas en otras vistas y similarmente, los componentes pueden hacer agrupación de varios componentes, con el fin de que se comprenda la estructura de toda la interfaz de usuario se presenta el siguiente diagrama (Figura 24) donde los rectángulos azules representan vistas mientras que los rojos corresponden a componentes.

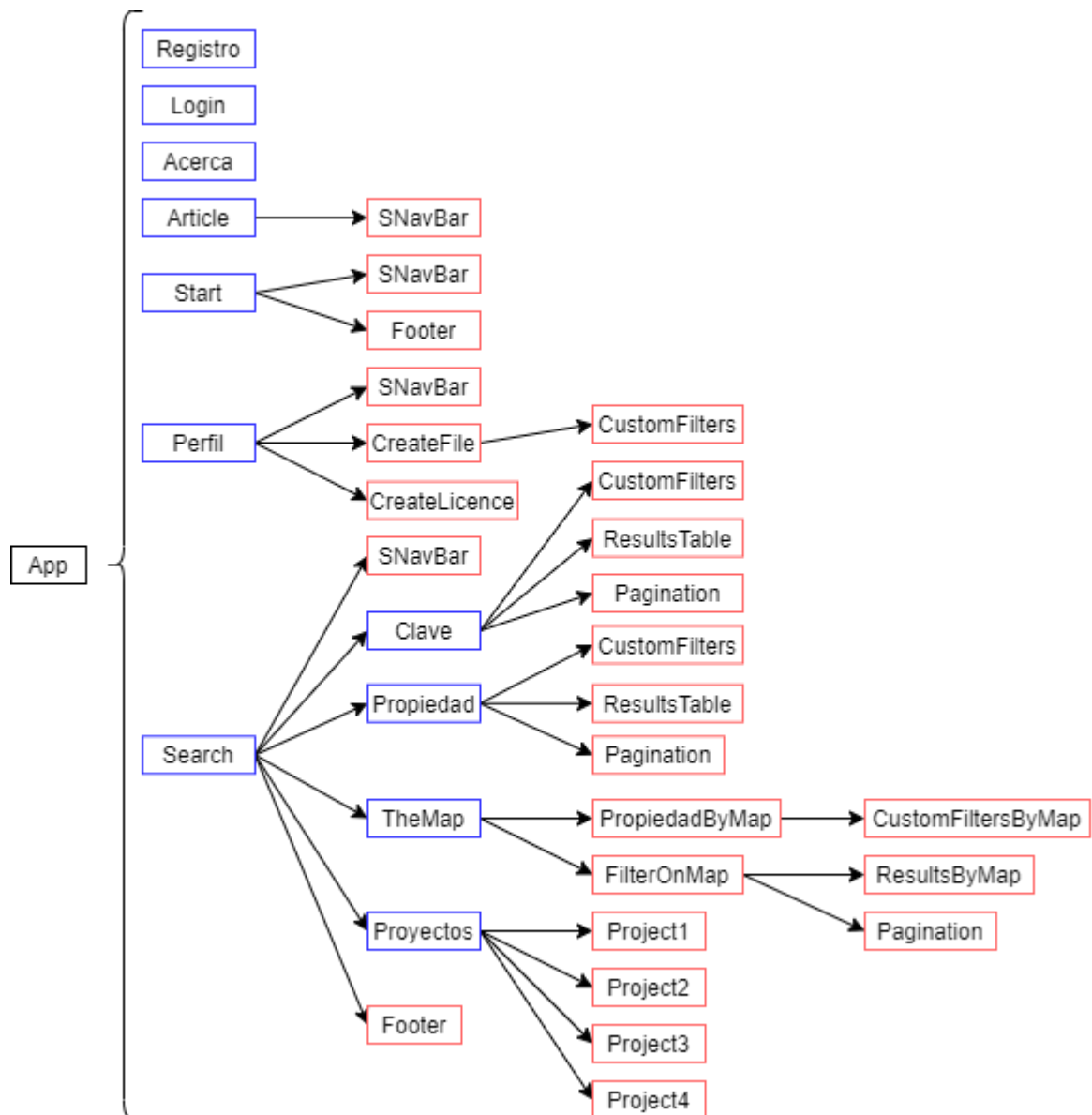


Figura 24. Relación de vistas y componentes en la interfaz de usuario

El usuario del sistema, en un navegador Web, mantiene una interacción por medio de una serie de elementos como botones, listas, cajas de texto y formularios. Dichos elementos conforman el Front-End y están estructurados mediante componentes y vistas que dan lugar a las diferentes utilidades y herramientas de búsqueda y descubrimiento de datos e información.

Luego, el usuario haciendo uso de la interfaz envía parámetros desde el Front-End, a través de las rutas anteriormente mencionadas, en las que se hacen peticiones mediante el protocolo HTTP al servidor.

En el servidor se recibe la petición y se accede a la base de datos para añadir, modificar o consultar datos, según el tipo de operación ejecutada. En el caso de las descargas de datos, la URL en los metadatos permite ubicar el archivo requerido dentro del disco de almacenamiento.

Después, el servidor retorna una respuesta y el Back-End se encarga de traducirla en formato JSON hasta el Front-End. De acuerdo al tipo de petición que se hubiera realizado, el JSON es interpretado y transformado para ser presentado posteriormente en la interfaz y visualizado por el usuario.

En el Anexo 1 se incluyen los códigos escritos en la construcción del Front-End.

3.1.4.4.1. Leaflet

La librería Leaflet estuvo presente en la construcción del Front-End, fue implementado para la elaboración del buscador geográfico, ya que permitió incluir interacciones con el mapa, las cuales se destinaron para la búsqueda de información.

Leaflet provee las funcionalidades para desplegar un mapa web y a través de sus diferentes métodos agregar presentar información recuperada desde una base de datos. De esta manera, se exhiben las geometrías de los departamentos, municipios y fincas. Con esta librería se pueden definir la simbología, escala y añadir leyendas al mapa. También permite la captura de coordenadas sobre el mapa, que en el caso de SILAIN fue aprovechado para implementar la utilidad de búsqueda de la finca más cercana. Leaflet es usado en la vista denominada *TheMap.jsx*.

Una segunda inclusión de elementos cartográficos se efectuó en la visualización de los metadatos, donde se presenta un mapa con el fin de mostrar la finca con la cual se relaciona el dato, el componente *Article* contiene la definición de este mapa.

3.1.4.4.2. Plantilla de ReactJS

Con intención de agilizar la creación de la interfaz de usuario se hizo uso de una plantilla prediseñada de React. Está codificada en lenguaje JavaScript y hace uso de lenguaje HTML para la creación de componentes Web. Los estilos han sido elaborados a partir de la librería de código abierto Bootstrap.

Paper Dashboard React es el nombre de esta plantilla cuyo autor es Creative Tim, cuenta con una licencia de uso MIT, esta es bastante permisiva y no impone restricciones de uso, con la condición de que se conserve el aviso de derechos de autor, en consecuencia, esta marca será visible en el pie de página o Footer del aplicativo web (Creative Tim, 2018).

La plantilla descargada trae consigo una serie de componentes y vistas predeterminadas con estilos predefinidos. Por lo tanto, se conservaron los elementos que se ajustaban a las necesidades del sistema. Los elementos incompatibles con los requerimientos fueron suprimidos. En cuanto a los estilos, algunos se tuvieron que modificar y adaptar de acuerdo con las características planeadas para los diferentes módulos y secciones.

3.1.4.5. Integración del Back-End y el Front-End

El sistema está conformado por dos capas, el Back-End y el Front-End. Como ya se ha mencionado, el Back-End es la parte encargada de establecer una conexión con el servidor y realizar las peticiones mediante unas rutas específicas para enviar y obtener datos. Por otro lado, el Front-End consta de las vistas y componentes que son presentadas al usuario como elementos de interacción y visualización, entre los que se incluyen botones, listas desplegables, cajas de texto, párrafos, ventanas, etc.

Estas dos capas están comunicadas entre sí, permitiendo al usuario interactuar con una interfaz en la que realiza operaciones traducidas en peticiones que llegan al servidor y son retornadas nuevamente a la interfaz para ser visualizadas por el usuario (Figura 25).

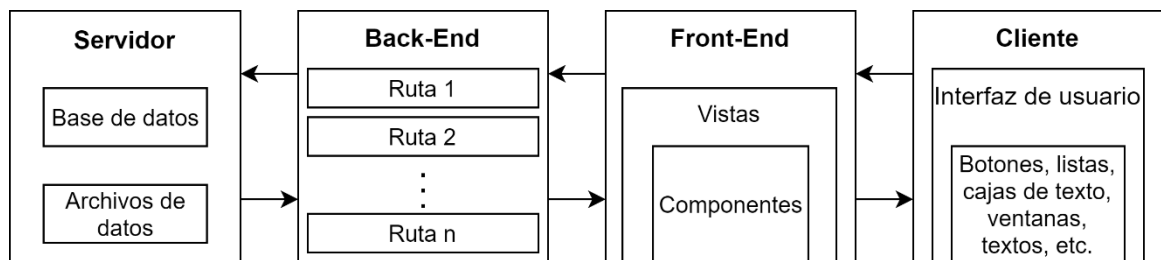


Figura 25. Integración de los componentes del software del sistema

3.1.4.6. Hardware

El sistema se implementó en un servidor del Laboratorio de Agricultura Inteligente. Este equipo cuenta con las siguientes características: Procesador Core i7, Memoria RAM de 16 GB y Sistema Operativo Windows 10.

3.1.5. Pruebas

Antes de la puesta en funcionamiento del sistema es necesario efectuar pruebas sobre cada uno de los componentes del sistema, con el propósito de que la versión final o de producción opere de forma correcta y la experiencia del usuario sea satisfactoria.

En SILAIN, a medida que se iba implementando cada funcionalidad en la REST API, se llevaba a cabo la prueba del envío de la petición a través del protocolo HTTP, una vez se comprobaba el retorno de la respuesta esperada, se consideraba correcto el funcionamiento del elemento específico.

Igualmente, luego de la construcción de la interfaz de usuario o Front-End, se llevaron a cabo las pruebas de estas mismas funcionalidades, esta vez a través del uso de los diferentes elementos dispuestos en la interfaz, simulando el uso del sistema por parte de un usuario.

Las pruebas del envío de peticiones fueron aplicadas en un equipo con un procesador Intel Core i7 -7500, memoria RAM de 8GB y sistema operativo Windows 10.

En la Tabla 22 se muestran las pruebas de las diferentes operaciones del sistema y el tiempo que tardó en responder el servidor.

Tabla 22. Pruebas realizadas a las operaciones del sistema

Ruta	Método HTTP	Descripción	Resultado	Tiempo de respuesta (ms)
/article/get/:id	GET	Obtiene la información de un metadato según su ID, incluyendo la geometría de la finca con la que guarda relación	Correcto	104.507
/article/get/:id/:userid	GET	Obtiene la información de un metadato según su ID, incluyendo la geometría de la finca con la que guarda relación, esto para datos restringidos que hayan sido autorizados para que un usuario específico acceda.	Correcto	20.898
/article/download/:id	GET	Obtiene la URL de un archivo de datos para su descarga.	Correcto	402.015
/article/download/filename/:id	GET	Obtiene el nombre de un archivo de datos para su descarga.	Correcto	2.041
/article/proposito	POST	Envía el propósito o razón por la cual se descarga un dato.	Correcto	0.481
/article/crear	POST	Envía el dato y los metadatos que se pretenden añadir a la base de datos	Correcto	106.278
/basic/search_by_filter	POST	Envía una consulta a la base de datos para realizar una búsqueda.	Correcto	127.275

/getfilters/	GET	Obtiene los datos de las listas desplegables para filtrar los resultados de búsqueda.	Correcto	79.663
/getfilters/ubication	GET	Obtiene los datos de las listas desplegables para filtrar los resultados en el buscador geográfico.	Correcto	3.095
/getfilters/onmap	GET	Obtiene los datos de las listas desplegables para buscar a través de Departamento, Municipio o Finca en el buscador geográfico.	Correcto	69.845
/licence/add	POST	Envía el email de un usuario y el id del dato que le ha sido autorizado para descargar	Correcto	64.112
/makepublic/	POST	Envía el cambio del tipo de acceso a un dato (requiere autorización o descarga inmediata)	Correcto	4.363
/map/ubication_by_filter	POST	Envía una consulta realizada según un departamento, municipio o finca en el buscador geográfico.	Correcto	33.759
/map/getpoly	POST	Envía una consulta sobre la geometría de municipio o finca en el buscador geográfico.	Correcto	175.955
/map/finca_closer	POST	Envía coordenadas del mapa para consultar distancia a fincas.	Correcto	81.678
/users/register	POST	Envía los datos de un usuario que se registre en la plataforma.	Correcto	53.793
/users/login	POST	Envía los datos de un usuario haciendo inicio de sesión.	Correcto	0.132
/users/user/:id	GET	Obtiene los datos de un usuario	Correcto	2.553
/users//user/des_hist/:id	GET	Obtiene el historial de descargas de un usuario	Correcto	3.454

CAPÍTULO 4

4.1. RESULTADOS

En seguida se presentarán los resultados obtenidos tras el desarrollo del Sistema de Información del Laboratorio de Agricultura Inteligente.

4.1.1. Estructura de almacenamiento de los datos

Con la finalidad de disponer de forma organizada la información generada en los subproyectos de Agricultura Inteligente se llevó a cabo un proceso recopilación, depuración y clasificación de la información. De este modo, se consolidó un grupo de información de un tamaño de 135 GB, conformado por cerca de 426 archivos.

La información fue agrupada de acuerdo con las categorías y subcategorías propuestas en la Tabla 5 y donde las temáticas lo permitían, se agregaron otros niveles de clasificación como zona de estudio y fase, según lo expresado en la Tabla 6.

Esta estrategia de organización de la información facilitó más adelante el desarrollo de procesos y funcionalidades para la consulta y acceso a la información esto a fin de disponerla de forma clara a los usuarios del sistema.

4.1.1.1. Archivos con información comprimida

En el caso de las imágenes multiespectrales y RGB sin procesar, se cuenta con un solo archivo comprimido por cada vuelo fotogramétrico, así, cada fichero comprimido podría contener hasta 400 imágenes en el caso de las imágenes RGB; mientras que cada archivo comprimido de imágenes multiespectrales podría albergar entre 1000 y 1500 elementos en formato TIF.

De forma semejante, la información capturada mediante espectrometría se conglomeró en ficheros comprimidos, según la zona de estudio cada archivo reúne entre 30 y 90 lecturas almacenadas en formato TXT.

La compresión de archivos también se dio en algunos casos de los muestreos fisiológicos de las plantas de arroz, por esto podrían encontrarse hasta 15 archivos TXT por cada comprimido.

La Información espacial contenida en archivos vectoriales también fue comprimida, más específicamente shapefiles, esto en virtud de que estos elementos son de formato multiarchivo, así que cada vector, ya sea de tipo polígono, línea o punto, está ligado a por lo menos tres archivos que contienen la geometría de los objetos, el índice de estas geometrías y la base de datos que almacena sus atributos.

Además de esto también podría encontrarse archivos que contienen el sistema de coordenadas, índices espaciales o metadatos.

4.1.1.2. Base de datos de SILAIN

Como se mencionó en la implementación, la información no se albergó directamente en la base de datos. Es decir, los archivos antes mencionados fueron almacenados en forma de archivos y carpetas dentro de una unidad física o disco duro.

Posteriormente, se construyó la base de datos que permite la gestión de la información. Se crearon tablas destinadas a el registro de los usuarios, la clasificación de la información según sus atributos geográficos y el sistema de categorías propuesto, la descripción de los datos a través de metadatos, el control del acceso a los datos y el seguimiento de la descarga de datos.

Por parte de los usuarios del sistema se lleva registro de datos básicos de identificación y ubicación como el nombre, apellidos, país, departamento, ciudad, institución y ocupación. Para el ingreso a la plataforma, se registra un correo electrónico y una contraseña creada por el usuario.

Otra tabla contiene los dos roles que se contemplan en el sistema, siendo el primero el administrador y el otro de cliente.

Ocho categorías y 18 subcategorías fueron tenidas en cuenta dentro de la base de datos. Además de esto, la información se ordenó de acuerdo con la zona de estudio, una tabla con atributos espaciales contiene las cinco fincas donde se centraron las investigaciones. Estas fincas están ubicadas en tres municipios para los cuales se dispuso de una tabla, también con propiedades espaciales. Los municipios pertenecían a dos departamentos diferentes, los cuales se incluyeron dentro de otra tabla junto con su información espacial.

Otra tabla contiene los metadatos que cumplen la función de describir los datos. En esta tabla se tiene un registro por cada uno de los archivos que conforman la información. En una de las columnas de esta tabla se almacena la URL del archivo dentro de la unidad de almacenamiento, usada para el acceso y descarga de la información.

Una tabla permite la relación entre los metadatos y las fincas, mientras que otra tabla mantiene la correspondencia entre un metadato y una subcategoría.

La información puede tener un libre acceso o también podría contar con una restricción que condicione su acceso a una autorización del administrador. Estas autorizaciones son administradas desde la tabla llamada licencias, que relaciona el correo electrónico de un usuario con la identificación de uno de los datos.

Las descargas son registradas en una tabla, donde además de incluir el correo electrónico del usuario que realiza la descarga, se almacena el identificador del dato y la hora en que fue efectuada la descarga.

4.1.2. Metadatos

Con el sistema desarrollado se busca, además de facilitar el acceso a los datos, favorecer las condiciones para su reutilización. Esto se logra principalmente con la documentación o descripción detallada de los datos.

En la base de datos del sistema se introdujo una tabla para el registro de metadatos. Esta permite la documentación de diferentes aspectos de los datos. La ubicación geográfica, contexto del dato, temporalidad, información del archivo son algunos de los componentes de los metadatos.

Este modelo de metadatos estuvo inspirado en el estándar Dublin Core, esto debido a su flexibilidad, pues permite seleccionar los grupos y elementos de metadatos que mejor se adapten a las necesidades del investigador.

En la Tabla 23 se muestra la estructura de los metadatos agrupados según el ámbito del metadato, cada elemento del metadato se acompaña de una descripción.

Tabla 23. Estructura de metadatos de SILAIN

Grupo	Elemento	Descripción
Identificación	Identificador	Corresponde a un número entero que es único para cada dato y por tanto lo puede identificar.
	Título	Es el nombre a través del cual se conoce el dato.
	Palabras Clave	Son palabras que describen temáticas, características, técnicas, lugares y otros aspectos relacionados con el dato.
Contenido	Resumen	Se trata de un breve escrito que resalta el contexto del dato.
	Descripción	Este ítem explica cómo es el dato, su contenido y sus características propias.
Clasificación	Categoría	Hace referencia a una temática general en la que se enmarca el dato.
	Subcategoría	Indica con más especificidad el área o tema a la que pertenece el dato.
	Tipo	Expresa la diferencia de los datos en cuanto a su nivel de transformación.
Temporalidad	Fecha de creación	Es la fecha en la que el dato fue creado, ya sea a través de su captura, o como resultado de un proceso.

	Fecha de subida	Es la fecha en la que el dato fue subido a la base de datos del sistema.
Espacialidad	Finca	Corresponde a la finca zona de estudio donde se efectuó la investigación donde surgió el dato.
	Municipio	Es el municipio donde está ubicada la finca relacionada con el dato.
	Departamento	Indica el departamento al que pertenece el municipio.
De archivo	Formato	Es la extensión del formato del archivo que contiene al dato.
	Tamaño	Suministra el tamaño del archivo del dato.
Acceso	Tipo de acceso	Informa si el dato cuenta restricción para su descarga.
	URL	Es la ruta o dirección del dato en la unidad de almacenamiento.

El llenado de los metadatos se realizó con base en la información proporcionada por los encargados de cada una de las investigaciones, así como por información entregada por el director del Grupo de Investigación GISMODEL.

Los metadatos desempeñan un papel importante en la caracterización y descripción de los datos. Gracias a estos, quienes accedan a la información a través de SILAIN, podrán contar con elementos necesarios para su comprensión y por consiguiente contar con criterios para hacer la reutilización de los datos.

4.1.2.1. Datos FAIR

La filosofía de datos FAIR busca mejorar las costumbres frente a los datos en los entornos investigativos. En SILAIN se hizo una aproximación del seguimiento de los principios rectores FAIR.

En cuanto a la cualidad de localizable, los datos cuentan con un identificador que permite distinguirlos de otros. De igual forma, se construyeron metadatos que describen a los datos.

En el tema de accesibilidad, los datos son recuperables a través de una URL que hace referencia a una dirección en la unidad de almacenamiento de datos. También, se hace uso de un protocolo estandarizado (HTTP) y existen procedimientos para la autenticación de las cuentas de usuario y para autorizaciones para el acceso a los datos.

La interoperabilidad se buscó mediante el uso de un modelo de metadatos basado en el estándar Dublin Core, el cual es ampliamente usado en la comunidad científica.

Con respecto al principio reutilizable, los datos se encuentran descritos con atributos relevantes para su comprensión, además se hace claridad sobre las restricciones que pudieran tener algunos datos.

4.1.3. Gestión de la información en el sistema

En SILAIN se implementaron métodos y procedimientos para la gestión de los datos y la información enfocados en brindar a los usuarios diversas utilidades de acceso a la información, esto sin descuidar la trazabilidad del dato y permitiendo mantener la reserva de información según las políticas del Centro.

Los archivos de la información de las investigaciones de Agricultura Inteligente se conservan en la unidad de almacenamiento de un servidor en el LAIN. Luego, se construyó una base de datos para el registro de metadatos y otros elementos que permiten la caracterización, descripción y comprensión de los datos y la información dispuesta a los usuarios.

De esta forma, se proporciona además del dato, información contextual que explica el origen el dato, el espacio geográfico al que pertenece, el marco temporal, entre otras características.

La gestión de la información implica también el control sobre los usuarios que acceden a esta, por tal razón, la base de datos registra cada cliente que hace uso del sistema.

Desde el módulo de administración del sistema se cuenta con la posibilidad de añadir nueva información complementada por metadatos a la base de datos. El administrador puede decidir el tipo de acceso que tendrá el dato, es decir, si su descarga podrá realizarse sin necesidad de autorización o si, por el contrario, su descarga estará condiciona a una autorización del administrador.

Otras funcionalidades para la gestión de la información son la facultad del administrador para cambiar el tipo de acceso al dato y la asignación de autorizaciones de acceso a los datos, esto último para el caso en que un usuario requiera acceder a un dato restringido.

Tanto administrador como clientes pueden utilizar las funcionalidades de búsqueda, descubrimiento y acceso a los datos. Esto incluye el Buscador por Palabras Clave, el Explorador de Datos, el Buscador Geográfico y los Proyecto de Investigación. Al acceder a un dato, el usuario se encontrará con sus metadatos.

Cuando el usuario proceda con la descarga, se le solicitará enviar un formulario en el que exponga el uso que le dará a la información. Esto permite hacer una trazabilidad del dato, que incluye el registro del usuario que accede al dato, el identificador del dato, el motivo de uso del dato y la fecha de descarga. Si el dato está restringido, el cliente deberá solicitar el acceso a la administración mediante correo electrónico, tras ser autorizado deberá igualmente indicar el propósito de la descarga.

4.1.4. Plataforma web para el acceso a datos e información

SILAIN es un sistema accesible desde la internet. Al ingresar a la dirección del sitio web (<http://45.5.164.43/cibiofi/lain/silain/>), el usuario se encontrará con una plataforma amigable, de fácil uso y con elementos claramente definidos para realizar las diferentes operaciones disponibles en el sistema.

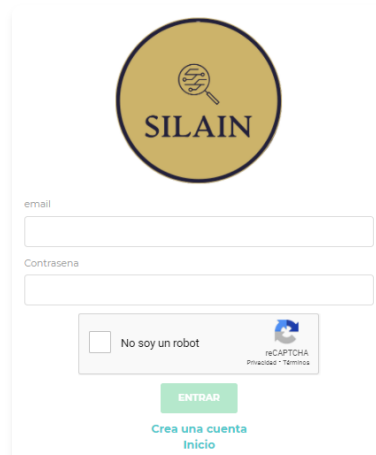
La página principal de la plataforma (Figura 26) muestra el logo de SILAIN y una barra para realizar la búsqueda por palabras clave. En el encabezado está el botón de inicio, que redirige a la pantalla principal, asimismo se localizan las cuatro utilidades de búsqueda con las que cuenta el sistema. En el extremo derecho del encabezado está el panel de usuario, donde se podrá hacer el registro de usuarios y el inicio de sesión.



Figura 26. Pantalla de inicio de SILAIN

En el pie de página se encuentra el enlace para visualizar la sección Acerca de SILAIN, donde se muestra información sobre el origen del sistema, su objetivo y autores. También se muestran los derechos de autor de la plantilla utilizada para el diseño de la plataforma. Igualmente, se puede observar el logo de CIBioFi donde al realizar clic, se redirige a la página web del Centro.

Al ingresar al panel de usuario se podrá hacer inicio de sesión (Figura 27) o en caso de no contar con una cuenta en la plataforma, el usuario podrá registrarse para obtener su cuenta.



The login interface for the SILAIN platform features a central logo with a magnifying glass icon and the text 'SILAIN'. Below the logo are two input fields labeled 'email' and 'Contraseña'. A CAPTCHA verification box is present, containing a checkbox labeled 'No soy un robot' and a 'RECAPTCHA' icon with links for 'Privacidad' and 'Términos'. A green 'ENTRAR' button is positioned below the input fields. At the bottom, there are two links: 'Crea una cuenta' and 'Inicio'.

Figura 27. Interfaz para inicio de sesión

Para el inicio de sesión el usuario deberá ingresar el correo electrónico registrado y la contraseña que haya asignado a su cuenta, después hará una verificación antibots y podrá entrar a la plataforma.

Si el usuario no posee una cuenta en la plataforma deberá diligenciar un formulario (figura 28) con unos datos básicos.



The registration form, titled 'Crear Cuenta', includes several input fields for user information: 'Nombres', 'Apellidos', 'email', 'Pais', 'Departamento', 'Ciudad', 'Institucion', 'Ocupacion', 'Contraseña', and 'Confirmar Contraseña'. A prominent pink error bar is visible below the 'Confirmar Contraseña' field. At the bottom, there is a green 'REGISTRAR' button and two links: 'Iniciar Sesión' and 'Inicio'.

Figura 28. Formulario de registro en SILAIN

Haciendo clic en Búsqueda por Palabras Clave (Figura 29), se mostrará nuevamente la barra para llevar a cabo una búsqueda mediante la escritura de una palabra.

Figura 29. Búsqueda por Palabras Clave

Tras efectuar la búsqueda se indicará el número de resultados y un listado con los datos encontrados (Figura 30). Los resultados incluyen el nombre del dato y otra información extraída de los metadatos. Una serie de listas desplegables actúan como filtros para obtener un resultado más específico (Figura 31).

#	TITULO Y RESUMEN	TIPO	FORMATO	TAMAÑO	CREADO	DISPONIBILIDAD	ACCESO
1	Lectura de espectrometría en cultivo de arroz, finca Calderón tratamiento D1 Datos hiperespectrales capturados con espectrorradiómetro de plantas de arroz en la finca Calderón, localizada en Jamundi, Valle del Cauca.	Archivo crudo	txt	303.3 KB	12/07/2017	01/06/2020	Privado
2	Lectura de espectrometría en cultivo de arroz, finca Calderón tratamiento N2 Datos hiperespectrales capturados con espectrorradiómetro de plantas de arroz en la finca Calderón, localizada en Jamundi, Valle del Cauca.	Archivo crudo	txt	295.1 KB	12/07/2017	01/06/2020	Público

Figura 30. Resultados de búsqueda

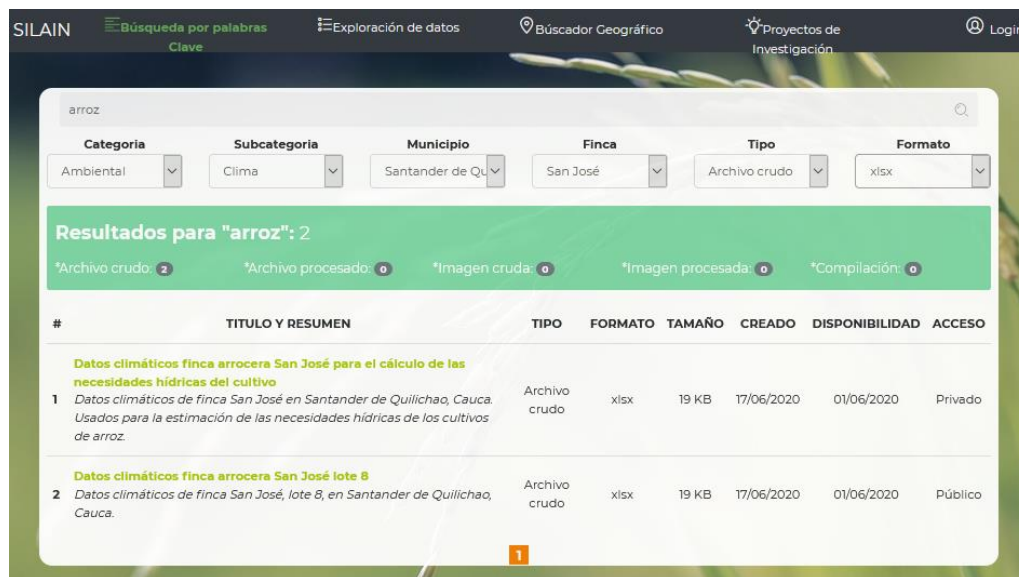


Figura 31. Filtrado de los resultados de búsqueda

La herramienta de Exploración de Datos contiene listas desplegables (Figura 32), a través de las cuales se puede obtener resultados según los valores que sean asignados en las listas. De modo que no es necesario ingresar texto para la búsqueda. Los resultados son mostrados de la misma forma que en Búsqueda por Palabras Clave.



Figura 32. Exploración de Datos

La tercera utilidad para descubrimiento de datos es el Buscador Geográfico (Figura 33), donde a través de un mapa, el usuario podrá visualizar entidades como departamentos, municipios y fincas para realizar una búsqueda relacionada con estas ubicaciones.

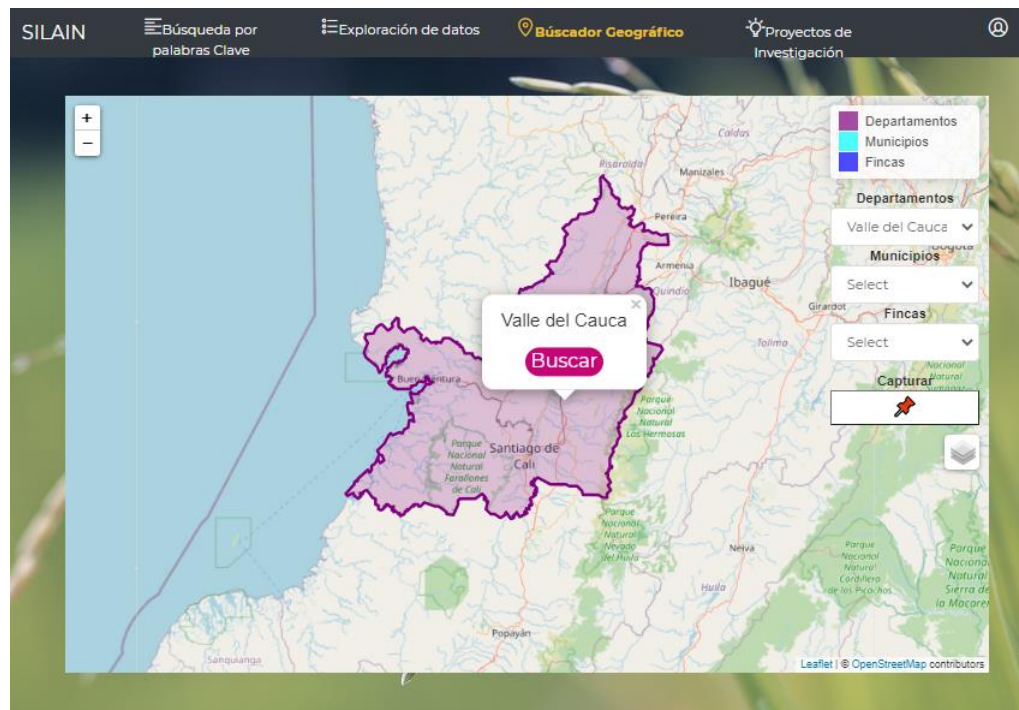


Figura 33. Buscador geográfico

Los resultados que se obtienen por medio de este buscador pueden ser filtrados como en los casos anteriores (Figura 34).

Valle del Cauca

Cultivo: arroz | Categoría: Select | Subcategoría: Select | Tipo: Select | Formato: Select

Resultados: 249

- *Archivo crudo: 32
- *Imagen cruda: 12
- *Archivo procesado: 6
- *Imagen procesada: 190
- *Compilación: 9

#	TITULO Y RESUMEN	TIPO	FORMATO	TAMAÑO	ACCESO
1	Lectura de espectrometría en cultivo de arroz, finca Calderón tratamiento D1 Datos hiperespectrales capturados con espectrorradiómetro de plantas de arroz en la finca Calderón, localizada en Jamundí, Valle del Cauca.	Archivo crudo	txt	303.3 KB	Privado
2	Lectura de espectrometría en cultivo de arroz, finca Calderón tratamiento N2 Datos hiperespectrales capturados con espectrorradiómetro de plantas de arroz en la finca Calderón, localizada en Jamundí, Valle del Cauca.	Archivo crudo	txt	295.1 KB	Público

Figura 34. Resultados en Buscador Geográfico

La sección de Proyectos de Investigación (Figura 35) es una utilidad para la consulta de las cuatro investigaciones en Agricultura Inteligente adelantadas por miembros del Grupo de Investigación GISMODEL.

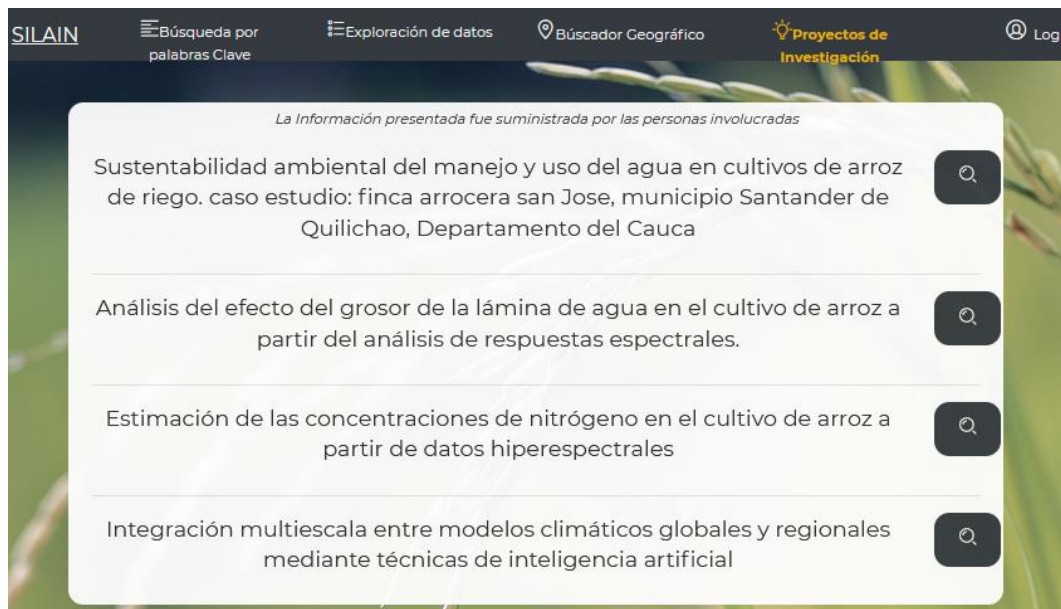


Figura 35. Proyectos de Investigación

En esta utilidad, el usuario podrá revisar un resumen de cada proyecto, una breve muestra de la metodología, los resultados y las conclusiones (Figura 36).

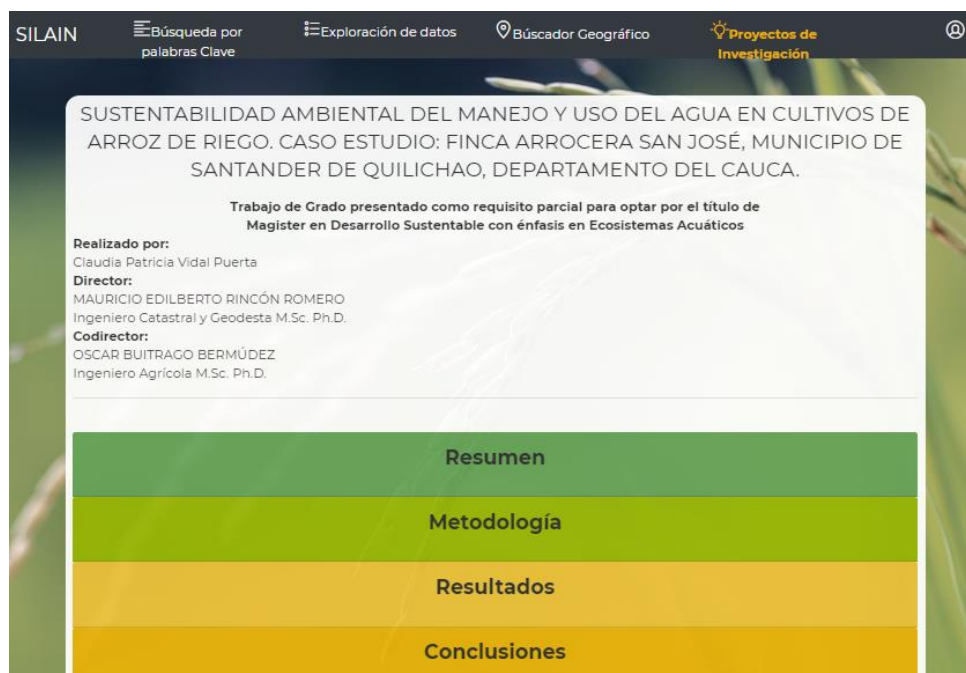


Figura 36. Ejemplo de Proyecto de Investigación

En la metodología de los proyectos se podrá encontrar datos relacionados con la investigación (Figura 37).

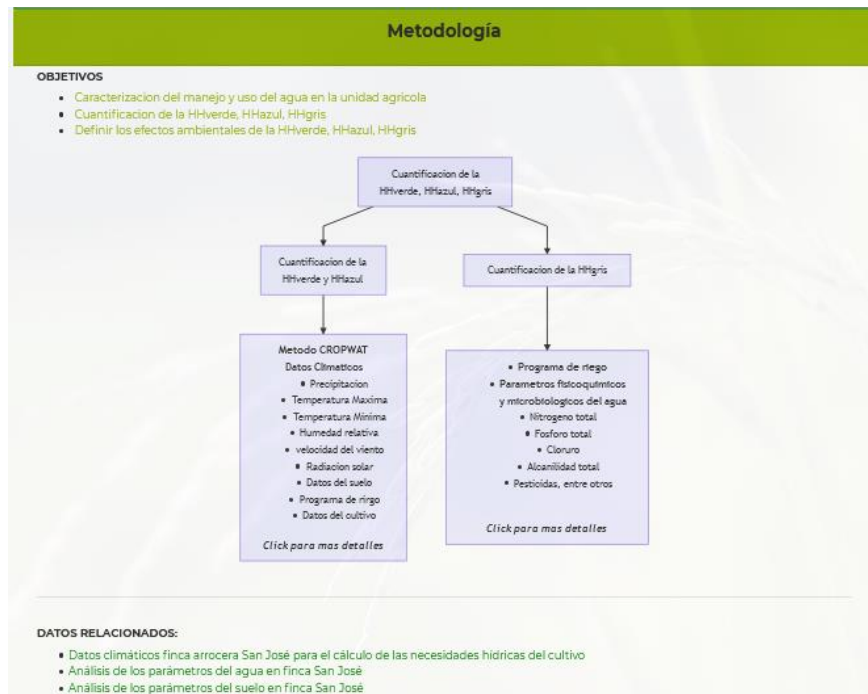


Figura 37. Acceso a datos en Proyectos de Investigación

Al momento de ingresar a un dato, el usuario visualizará los metadatos (Figura 38) y un mapa donde se representa la finca relacionada con el dato. Haciendo clic en las palabras clave, podrá efectuar una nueva búsqueda según el término. Mediante el botón descargar, el usuario podrá obtener el dato.

SILAIN

Búsqueda por palabras clave

Exploración de datos

Búscador Geográfico

Proyectos de Investigación

nom

Departamento: Cauca

Municipio: Santander de Quilichao

Finca: San José

Características

- Categoría:** Ambiental
- Subcategoría:** Suelo
- Fecha de creación:** 17/06/2020
- Fecha de disponibilidad:** 01/06/2020
- Acceso:** Inmediato

Datos de descarga

- Tipo:** Archivo procesado
- Formato:** docx
- Tamaño:** 105 KB

DESCARGAR

Palabras clave

arroz análisis muestra huella hídrica suelo

san José santander de quilichao cauca

Identificador: 409

Resumen:

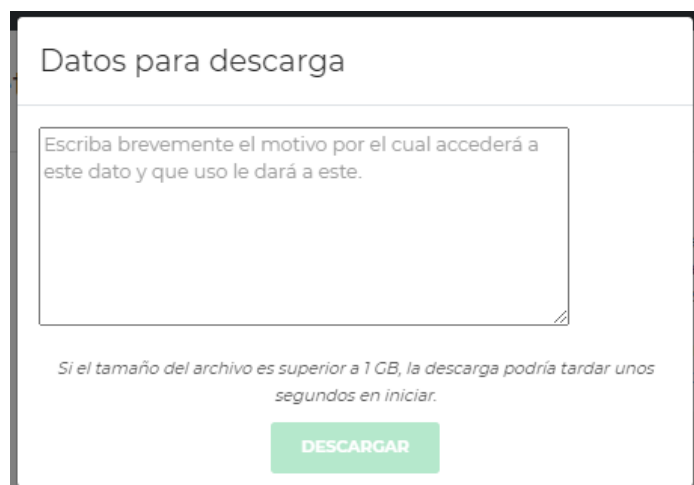
Resultado de análisis de los parámetros de suelo en parcela experimental de la Finca arrocería San José en Santander de Quilichao, Cauca. Muestra tomada el 27 de julio de 2018, análisis realizado por AGQ Labs.

Descripción:

Este análisis de suelo contiene información acerca de la fertilidad física, fertilidad, relaciones catiónicas y otros parámetros.

Figura 38. Visualización de los metadatos

Antes de que comience la descarga, el usuario deberá indicar el motivo o razón por la que hará uso del dato, lo cual será registrado en la base de datos (Figura 39).



Datos para descarga

Escriba brevemente el motivo por el cual accederá a este dato y que uso le dará a este.

Si el tamaño del archivo es superior a 1 GB, la descarga podría tardar unos segundos en iniciar.

DESCARGAR

Figura 39. Formulario para uso del dato

En caso de que el dato tenga un acceso restringido, en lugar del botón descargar aparecerá solicitar descarga (Figura 40), al hacer clic sobre este, se presentará el siguiente mensaje (Figura 41).

Datos climáticos finca arrocera San José para el cálculo de las necesidades hídricas del cultivo

Departamento: Cauca

Municipio: Santander de Quilichao

Finca: San José

Identificador: 406

Resumen:
Datos climáticos de finca San José en Santander de Quilichao, Cauca. Usados para la estimación de las necesidades hídricas de los cultivos de arroz.

Descripción:
Los datos climáticos abarcan un periodo desde el 26 de agosto de 2018 hasta el 18 de febrero de 2019. Se presentan valores de temperatura mínima, temperatura máxima, humedad, viento, insolación y precipitación.

Características

- **Categoría:** Ambiental
- **Subcategoría:** Clima
- **Fecha de creación:** 17/06/2020
- **Fecha de disponibilidad:** 01/06/2020
- **Acceso:** Sujeto a autorización

Datos de descarga

- **Tipo:** Archivo crudo
- **Formato:** xlsx
- **Tamaño:** 19 KB

SOLICITAR DESCARGA

Palabras clave

arroz clima cropwat huella hídrica agua temperatura humedad viento precipitación insolación san José santander de quilichao cauca



Mapa de la finca San José en Santander de Quilichao, Cauca. El mapa muestra la finca delimitada por un polígono verde y su ubicación geográfica en relación con la carretera y el río. El mapa incluye la leyenda de OpenStreetMap.

Figura 40. Dato con acceso restringido



Figura 41. Mensaje en datos restringidos

Al dirigirse al panel de usuario con la sesión iniciada, se podrá ver el historial de descargas (Figura 42) que muestra el dato y fecha de descarga. Después, se encuentra el botón para salir de la plataforma.

 Bienvenido Alexander Clavijo Ledesma

alexander.clavijo@correounivalle.edu.co | Colombia

Historial de descargas

#	ARTICULO	FECHA	HORA
1	Espectrometría en cultivo de arroz, finca Calderón tratamiento D1	17/10/2020	20:06:17
2	Espectrometría en cultivo de arroz, finca Calderón tratamiento D1	17/10/2020	20:04:17
3	Espectrometría en cultivo de arroz, finca Calderón tratamiento D1	17/10/2020	20:02:17

[SALIR](#)

Figura 42. Historial de descargas

Cuando el administrador inicia sesión, en el panel de usuario se encontrará en primer lugar con la funcionalidad para la carga de nuevos datos en el sistema (Figura 43). Para esto, se tendrá que diligenciar un formulario para la creación del metadato. Aquí definirá el tipo de acceso al dato, sin tendrá o no restricción para su descarga.

Carga de datos y metadatos

Categoría	Subcategoría	Municipio	Finca	Tipo	Formato
Select ▼	Select ▼	Select ▼	Select ▼	Select ▼	Select ▼

Título

Resumen

Descripción

Lote

Fase

Palabras Clave

Acceso: PUBLICO

No se ha seleccionado ningún archivo.

Figura 43. Carga de nuevos datos y metadatos.

Luego, en el panel de usuario se verá la sección para asignar las licencias o autorizaciones (Figura 44) de acceso a datos que se encuentren restringidos.

Autorización de descargas

Email del usuario	Id del artículo

Figura 44. Asignación de autorizaciones de descarga

El administrador puede hacer uso de todas las utilidades de búsqueda, exploración y descubrimiento de datos al igual que los usuarios cliente.

En las listas de resultados de búsqueda de datos el administrador podrá modificar el tipo de acceso a los datos, esto mediante el botón público/privado (Figura 45).

SILAIN

Búsqueda por palabras Clave

Exploración de datos

Búscador Geográfico

Proyectos de Investigación

admin

Categoría

Subcategoría

Municipio

Finca

Tipo

Formato

Espectrometría

Select

Select

Select

Select

Select

Resultados según los filtros: 87

*Archivo crudo: 87

*Archivo procesado: 0

*Imagen cruda: 0

*Imagen procesada: 0

*Compilación: 0

#	TÍTULO Y RESUMEN	TIPO	FORMATO	TAMAÑO	CREADO	DISPONIBILIDAD	ACCESO
1	Lectura de espectrometría en cultivo de arroz, finca Calderón tratamiento D1 Datos hiperespectrales capturados con espectrorradiómetro de plantas de arroz en la finca Calderón, localizada en Jamundí, Valle del Cauca.	Archivo crudo	txt	303.3 KB	12/07/2017	01/06/2020	PRIVADO
2	Lectura de espectrometría en cultivo de arroz, finca Calderón tratamiento N2 Datos hiperespectrales capturados con espectrorradiómetro de plantas de arroz en la finca Calderón, localizada en	Archivo crudo	txt	295.1 KB	12/07/2017	01/06/2020	PÚBLICO

Figura 45. Modificación del acceso a un dato

4.1.4.1. Manual de usuario para la plataforma

Con el fin de brindar al usuario una guía sobre el manejo de la plataforma, se elaboró un manual de usuario que contiene una descripción de los elementos que allí encontrará. De igual manera, se detalla paso a paso, cada una de las operaciones que pueden realizar los usuarios, tanto el administrador como el cliente.

Este manual de usuario puede encontrarse en el Anexo 2.

4.2. DISCUSIÓN

4.2.1. Gestión de datos e información de investigación

Los datos de investigación son los insumos esenciales para la producción del conocimiento científico. Gracias a este conocimiento se logra dar solución a problemas en todas las áreas y disciplinas, esto a través de desarrollos tecnológicos, productos y servicios.

Es evidente el gran valor que poseen los datos, por lo tanto, su gestión es una preocupación para las diferentes organizaciones, instituciones y demás entornos donde se desarrollan investigaciones. El incremento en el ritmo de producción de datos que se viene dando en los últimos años demanda que se construyan herramientas para optimizar la recopilación, almacenamiento, procesamiento y difusión de los datos.

El Sistema de Información del Laboratorio de Agricultura Inteligente responde a las necesidades surgidas tras los procesos de investigación de GISMODEL en el CIBioFi. La cantidad y variedad de información generada debía ser gestionada para así facilitar que otros investigadores pudieran contar con un acceso fácil y rápido, y eventualmente poder reutilizar con fines científicos o académicos.

Así como SILAIN, se han implementado diferentes plataformas, servicios y portales Web para publicar datos y metadatos de investigación y permitir su acceso mediante interfaces de usuario.

El proyecto australiano de investigación ARCHER (Androulakis et al., 2009) surgió para dar un manejo ordenado a los datos capturados y procesados. De esta forma se crearon herramientas de software con métodos estandarizados y seguros para almacenar, acceder y analizar resultados de investigación, facilitando el trabajo colaborativo. Con motivaciones afines se concibió la creación de SILAIN que, si bien no está enfocado en la colaboración entre investigadores, si opera como mecanismo para la gestión de datos e información y como medio para su visibilización y difusión.

La confidencialidad y seguridad de los datos de investigación es otro tema que se pretende abordar con las plataformas de intercambio de datos, pues se constituyen en respaldos de la información y al mismo tiempo limitan el acceso a los datos de investigación según las necesidades de los científicos (Androulakis et al., 2009).

Los múltiples formatos de archivos que existen hoy en día para contener la información es otro factor que incentiva el desarrollo de soluciones tecnológicas para el manejo de datos. Los datos de las investigaciones en Agricultura Inteligente no son ajenos a esta diversidad. Las diferentes técnicas e instrumentos con los que se capturaron y transformaron los datos dan cuenta de los distintos bloques de información. Similarmente en la aplicación de Open Spectral Database (Chalk, 2016) se buscó hacer frente a la heterogeneidad de formatos en información espectral. Con características comunes a SILAIN, a través de la OSDB los usuarios pueden buscar, descargar y compartir datos espectrales y sus metadatos.

Otras plataformas para la gestión de información cuentan con un alcance mayor al de SILAIN. GARDIAN además de proporcionar herramientas de búsqueda y descubrimiento de datos e información agrícola, emplea algoritmos que vinculan recursos relacionados, presentando a los usuarios información adicional relevante para su búsqueda (Craig, 2019). Igualmente, este portal de la CGIAR actúa para la gestión y análisis de Big Data pues provee la capacidad de agregar datos de varias fuentes y disciplinas.

Como se dijo, los datos son de gran relevancia en la ciencia, pero si se piensa en compartir y publicar datos para que otros investigadores hagan uso de ellos, es indispensable que estos sean descritos. La base de datos de SILAIN, entre elementos, cuenta con tablas destinadas para creación de metadatos. Ahora, también se han desarrollado sistemas computacionales especializados en la descripción de objetos de investigación. COPO (Etuk et al., 2019) brinda las utilidades necesarias para la creación de metadatos estandarizados de datos crudos, procesados, publicaciones, muestras, etc. Luego de la creación de metadatos, COPO ayuda en el envío de los datos a repositorios institucionales para que la comunidad científica pueda reutilizarlos.

4.2.2. Metadatos

Una de las principales necesidades en el desarrollo de SILAIN fue la inclusión de metadatos con el fin de describir detalladamente los datos y disponerlos a los usuarios de la plataforma, facilitando la posible reutilización de dichos datos.

Es ampliamente reconocido por la comunidad científica que la creación de metadatos es indispensable para la comprensión de los datos. Asimismo, los procesos de curación y otras actividades relacionadas con la preservación y archivado de los datos son necesarios para su reutilización en el futuro (Hey & Trefethen, 2003).

Los metadatos en el sistema implementado proveen diferentes características de los datos que permiten entender su origen, el contexto, situación espacial y temporal. Estos elementos descriptores también informan a los usuarios sobre las técnicas e instrumentos involucrados en la recopilación de los datos y los procesos que condujeron a la generación de otros datos (Gray et al., 2002).

La inexistencia de los metadatos hace que los conjuntos de datos sean difíciles de descubrir, e incluso cuando se descubren, es un desafío para otros comprender su significado y las condiciones bajo las cuales se recopiló y procesó los datos, por lo tanto, se espera que los datos publicados sean descritos de manera que otros puedan descubrirlos y comprenderlos en el futuro (Blaiszik et al., 2016). Para reutilizar los datos recopilados por otros, los investigadores necesitan información contextual, como documentación sobre equipos, protocolos, procedimientos de recopilación y procesamiento de datos, y condiciones experimentales o de laboratorio de manejo de datos (Culina et al., 2018).

El intercambio de datos es una de las actividades que se pretende fomentar con SILAIN. Anteriormente la ausencia de estándares para publicación de datos y metadatos era uno de los motivos por los cuales era muy reducido el intercambio de datos científicos, la documentación de los datos era considerada laboriosa (Gray, Szalay, Thakar, & Stoughton, 2002).

Con el paso del tiempo se desarrollaron múltiples modelos para la construcción de metadatos, uno de los más conocidos a nivel mundial es el estándar Dublin Core, este sirvió de inspiración para el diseño de la estructura de metadatos de SILAIN. Otras plataformas como Datorium (Alam et al., 2015) se basaron en el estándar Dublin Core para la construcción de metadatos y se complementaron con elementos previamente usados en el instituto GESIS donde se originó.

Así, la publicación de los datos contempla más que hacer que los datos sean accesibles en la Web, también abarca la asociación de un conjunto básico de atributos descriptivos con los datos publicados. Para que los datos científicos sean ampliamente útiles más allá de su propósito original, deben estar bien descritos, ser detectables, verificables y accesibles. (Blaiszik et al., 2016).

4.2.3. Intercambio y reutilización de datos

El sistema de gestión de información creado dispone los datos e información de Agricultura Inteligente de una manera organizada y que favorece su descubrimiento y acceso. Las bondades de los sistemas y la computación hacen de este tipo de herramientas, medios ideales para el intercambio de datos de investigación. Sin embargo, la información no solamente se puede hacer accesible a través de sistemas de información Web, aplicativos o plataformas informáticas. Durante años, las bibliotecas académicas y de investigación han hecho accesible datos geoespaciales, climáticos, estadísticos y de otro tipo que se utilizan con frecuencia para fines de enseñanza, aprendizaje e investigación (Frederick, 2016).

La sociedad actual depende en gran medida de la ciencia y las investigaciones, pero no siempre se hace lo suficiente por optimizar el trabajo científico (Frederick, 2016). La construcción de una plataforma como SILAIN busca que se aprovechen al máximo los datos de investigación producidos en los subproyectos de Agricultura Inteligente, puesto que se realizó una gran inversión a nivel económico, logístico e intelectual para llevar adelante todos los procesos que dieron origen a los diferentes bloques de conocimiento.

Comúnmente se tiene acceso a informes de investigación científica como artículos en revistas, publicaciones en libros, etc., sin embargo esto solo corresponde a hallazgos o interpretaciones de los datos que son el resultado de la investigación y no los datos sí mismo (Frederick, 2016). El Sistema de Información del Laboratorio de Agricultura Inteligente sirve como medio de visibilización de los datos generados en diferentes fases de las investigaciones y que no son accesibles desde las publicaciones que se realizan convencionalmente al finalizar los proyectos, por ejemplo, informes, documentos de trabajo de grado o artículos científicos.

Aunque lo ideal es que los datos de investigación sean accesibles para otros investigadores, son muchas las excusas y razones dadas para no seguir esta costumbre.

Una por las razones cuales se suele evitar la publicación de datos de investigación tiene un trasfondo económico. Esto se debe a que algunas investigaciones, tanto de origen gubernamental como privado, van dirigidas lograr ventajas competitivas o producir ganancias. Por esto, las organizaciones mantienen en secreto información de las investigaciones mientras se desarrollan los productos que permitan aventajar a la competencia y obtener el provecho monetario (Frederick, 2016).

Algunos científicos manifiestan la intención de compartir datos y de permitir su reutilización siempre y cuando obtengan los créditos a través de citas formales, ser informados sobre los usos que se darán a sus datos, entre otras condiciones y/o restricciones sobre los datos (Tenopir et al., 2011). Unas de las plataformas Web creadas con el propósito de intercambiar datos, por ejemplo Chemotion (Tremouilhac et al., 2020), permiten la generación de identificadores de objetos digitales (DOI), de manera que los datos se convierten en recursos citables. SILAIN no cuenta con esta funcionalidad, de modo que, actualmente la información y datos de Agricultura Inteligente no es citable.

Los investigadores reconocen que hacen falta herramientas para la creación y preservación de datos y metadatos, asimismo afirman que las instituciones a las que pertenecen no cuentan con políticas ni estándares para la preservación a largo plazo de los datos (Tenopir et al., 2011). Esta situación es compatible con los presentado en CIBioFi, de momento en el Centro no se cuentan con unas políticas establecidas en cuanto al manejo de la información, así, en proyectos como el de Agricultura Inteligente, no se dispuso de un plan o estrategia para la gestión de los datos y la información. Tampoco se han definido estándares para la creación de metadatos y la documentación de los diferentes objetos de investigación.

A menudo parece que no hay tiempo, ni fondos ni incentivos para la publicación de datos y, dado que sigue siendo una actividad poco común, tampoco parece haber aprecio por hacerlo (Alam, Müller, & Schumann, 2015). Contrario a esto, en el caso del Proyecto de Agricultura Inteligente si se pensó desde un principio en la implementación de una herramienta para la publicación y descubrimiento de datos.

Otras barreras que obstaculizan el intercambio de datos incluyen la falta de incentivos para dedicar el tiempo y el esfuerzo adicionales necesarios para publicar datos y la creencia de que otros investigadores se pondrán en contacto con ellos personalmente para solicitar los datos si es necesario (Frederick, 2016). Y, es que hacer que los datos sean reutilizables implica múltiples esfuerzos e inversiones. Para administrar datos se requiere de hardware, software, instrumentos, experiencia humana para la conservación y el mantenimiento (Pasquetto et al., 2019).

Lo que no se puede negar es que las bases de datos y repositorios son cruciales en la ciencia, ya que permiten el acceso a datos obtenidos por otros investigadores. Igualmente, los sistemas de información desempeñan un papel fundamental para el trabajo científico sostenible y constituyen una base importante para la difusión y aplicación del conocimiento (C. L. Borgman, 2015).

La reutilización de datos en las ciencias es un proceso complejo que depende de la confianza, la experiencia, las políticas y las infraestructuras de conocimiento. Las inversiones públicas en infraestructura científica, especialmente en la conservación de datos, son necesarias para que los datos estén disponibles y sean reutilizables (Pasquetto et al., 2019).

Si bien es cierto que las revistas claramente tienen un papel importante que desempeñar para hacer que los datos estén disponibles pública y permanentemente, los pasos más importantes para mejorar la forma en que se practica y se transmite la ciencia deben provenir de la comunidad científica en general (Hanson et al., 2011).

4.2.3.1. Características de las plataformas web

Para facilitar el descubrimiento y acceso a los datos es necesario que se brinden interfaces de usuario tanto intuitivas como seguras, la arquitectura de desarrollo web REST proporciona los elementos necesarios para la creación de interfaces para el acceso a los datos que sean amables con el usuario. Al igual que la Material Data Facility (Blaiszik et al., 2016), SILAIN se basa en un servicio web RESTful, que hace posible establecer de forma sencilla una comunicación entre el servidor y el cliente, abriendo un abanico amplio de oportunidades para la creación de utilidades para la gestión de los datos e información.

En el desarrollo de software casi siempre se busca simplificar la construcción y el diseño de los productos. Se suele hacer uso de sistemas o partes de sistemas ya existentes para dar solución a problemáticas particulares. Para alivianar los costos de mantenimiento y prescindir de instalaciones de software, en la MDF (Blaiszik et al., 2016), se optó por subcontratar servicios en la nube para el funcionamiento de las utilidades de publicación y descubrimiento de datos. Parecido a esto, en el Material Data Centre (Scott et al., 2014) se decidió usar un software ya consolidado, Microsoft SharePoint, para las interfaces de usuario. En el Sistema para datos de Agricultura Inteligente también se buscó el ahorro en términos del diseño de la interfaz de usuario, ya que se usó una plantilla predefinida con una licencia de uso gratuito.

Los conjuntos de datos en los que se pueden realizar búsquedas solo son útiles para otros investigadores si pueden encontrarlos. En el MDF ha sido diseñado con poderosos métodos de búsqueda para filtrar rápidamente a través de grandes colecciones de datos, de forma análoga, en SILAIN se construyeron diferentes utilidades para la búsqueda de datos, uno de ellos aprovecha el carácter geográfico de la información y asimismo se cuenta con la opción de filtrar los resultados.

Las interfaces de navegación simples mejoran el descubrimiento de datos conocidos y el descubrimiento de coincidencias similares, pero no exactas y facilitan el descubrimiento fortuito. Esto ocurre de modo semejante en SILAIN, con la búsqueda por palabras clave y el algoritmo usado para obtener resultados con coincidencias menores al 100% de los caracteres ingresados.

En el tema de usuarios y privilegios, el Sistema de Información del Laboratorio de Agricultura Inteligente cuenta con dos roles de usuario, uno corresponde al administrador y el otro al cliente. El cliente puede usar las herramientas para descubrimiento y acceso a datos, mientras que el administrador cuenta con esas utilidades más otras para la gestión de los datos y la carga de datos y metadatos.

A diferencia de SILAIN, en otros aplicativos como la MDF (Blaiszik et al., 2016), la publicación de datos y metadatos no es exclusiva del administrador del sistema, los usuarios cliente de esta herramienta también pueden participar en la publicación de conjuntos de datos. En Chemotion (Tremouilhac et al., 2020) añaden un nuevo rol que es el de revisor, quienes permiten aceptar y rechazar envíos de datos, o solicitar a los autores una revisión.

Otras soluciones que han sido implementadas para el almacenamiento de los datos de investigación han sido el de contratar servicios en la nube (Blaiszik et al., 2016), de modo que la información, además de estar almacenada localmente, contaba con un respaldo en línea.

El software y las librerías de código abierto son un común denominador en las plataformas de publicación de datos de investigación. En la OSDB (Chalk, 2016) han hecho uso de soluciones JavaScript como Bootstrap para el diseño en HTML5 y de marcos de trabajo como CakePHP para desarrollar el código.

El sistema COPO (Etuk et al., 2019) en parte está implementado mediante el lenguaje Python y como marco de referencia web se empleó Django. En COPO hicieron uso de PostgreSQL para la gestión de usuarios y sesiones, mientras que usaron una base de datos no relacional llamada MongoDB para los datos y metadatos.

Otro marco de referencia usado en el desarrollo de aplicaciones del lado de servidor es Ruby on Rails, siendo este el caso de Chemotion (Tremouilhac et al., 2020). La base de datos en este repositorio de datos químicos se implementó en PostgreSQL y su interfaz de usuario se desarrolló con la librería ReactJS. Las bases de datos relacionales suelen ser una solución frecuente, en el MDC (Scott et al., 2014) también optaron por este modelo.

El código abierto estuvo presente en varios momentos del desarrollo de SILAIN. Semejante a varias de las plataformas mencionadas, la base de datos del sistema fue construida en el software PostgreSQL. El lenguaje de programación predominante en el sistema fue JavaScript, ya que para el desarrollo de la REST API se empleó el entorno de código abierto NodeJS, mientras que para la interfaz de usuario se utilizó la librería ReactJS desarrollada por la comunidad de Facebook. Otras librerías de código abierto incluyeron a Leaflet para los elementos cartográficos y Bootstrap en los diseños de la interfaz.

4.3. CONCLUSIONES Y RECOMENDACIONES

Se llevó a cabo el desarrollo del Sistema de Información del Laboratorio de Agricultura Inteligente (SILAIN), una plataforma Web para la búsqueda, exploración y acceso a los datos e información generada en los subproyectos de Agricultura Inteligente adelantados por el Grupo de Investigación GISMODEL en el Centro de Investigación GISMODEL.

Con la puesta en marcha de SILAIN se espera que se convierta en una herramienta que visibilice un gran volumen de datos e información que se encontraba previamente desorganizada, desarticulada y sin facilidades para su consulta. Sirviendo también como una memoria de las investigaciones y resaltando el valor que tienen los datos no solo en la investigación donde fueron generados, sino que también reconociendo la utilidad que estos podrían tener en futuros proyectos.

Mediante el uso de herramientas de código abierto se construyó un producto que gestiona datos e información. Se proveen diferentes herramientas y utilidades de fácil manejo para el descubrimiento, localización y disposición organizada de los datos para su consulta por parte de la comunidad científica y académica, tanto del Centro de Investigación como de otras instituciones.

Ahora los datos se encuentran respaldados en un entorno seguro, evitando que la información sea eliminada o extraviada. Además, el sistema permite controlar el acceso a los datos, de manera que se apliquen las restricciones según las políticas del CIBioFi. Asimismo, con SILAIN se hace trazabilidad a los datos, registrando cada uno de los datos que sean descargados de la plataforma.

Inspirado en los principios rectores de datos FAIR, el sistema procura la reutilización de los datos, gracias al esquema de metadatos diseñado, que tomó como base el modelo Dublin Core y que permite una descripción amplia y detallada de los diferentes atributos y características de los datos.

Las múltiples utilidades de búsqueda proporcionadas por la plataforma hacen posible que los datos sean encontrados a través de diferentes criterios como la temática, los atributos geográficos y el tipo archivo que contiene la información. A futuro, se podrían optimizar e incluir nuevos algoritmos de consulta de datos e información que conduzcan a resultados de búsqueda más acertados y satisfactorios.

La base de datos de tipo relacional empleada permite mantener la integralidad de los datos, es fácil de consultar y cuenta con un gran soporte y variedad de herramientas; sin embargo, la rigidez del esquema dificulta la realización de cambios grandes en la estructura de la base de datos. El uso de una base de datos no relacional (NoSQL) podría ofrecer una mayor flexibilidad en cuanto a los tipos de datos admitidos y se contaría con un esquema dinámico y apto para volúmenes de información más grandes, pero se sacrificaría la integralidad de los datos y la capacidad de establecer relaciones entre ellos. De manera que, en trabajos futuros se podría optar por implementar soluciones híbridas como en el caso de COPO (Etuk et al., 2019), en las que se usan ambos tipos de bases de datos brindando una mayor versatilidad al sistema.

Este sistema de gestión de información es un punto de partida para un cambio en la forma en que los datos son gestionados en el CIBioFi. Se busca que se tome conciencia de la importancia de los datos y se generen políticas en torno a su almacenamiento, custodia, preservación y publicación. Igualmente, con esta iniciativa se espera que posteriormente se anexas nuevas aplicaciones que promuevan el trabajo colaborativo entre investigadores (Androulakis et al., 2009) y que estos desarrollos tecnológicos mejoren las condiciones de las labores científicas. Del mismo modo, posteriormente con la inclusión de políticas y mecanismos para la conservación y publicación de datos, los datos deberían ser asociados a Identificadores de Objetos Digitales (DOI) (Tremouilhac et al., 2020), de manera que puedan ser recursos citables y se den los créditos a los autores de los datos.

Entre los aspectos por mejorar en la plataforma se encuentran ampliar su alcance, incluyendo utilidades, herramientas y algoritmos que adapten al sistema a nuevos paradigmas como el Big Data (Craig, 2019) y la minería de datos. Por otro lado, en el estado actual de la integridad de los datos y la información, no todos los procesos y experimentos podrían ser replicados, de manera que la base de datos está abierta a ser robustecida con el fin de consolidar grupos de información más completos y que propicien la reproducción más experimentos.

Se espera también que el sistema motive a que en el Centro se adopten o diseñen estándares y esquemas en cuanto a la construcción de metadatos, para que así los investigadores realicen una descripción minuciosa de los datos, publicaciones, productos y otros objetos de investigación, con el fin de prepararlos para ser comprendidos y reutilizados en futuras investigaciones (Culina et al., 2018).

CAPÍTULO 5

5.1. Anexos

5.1.1. Anexo 1. Códigos del Back-End de SILAIN

A continuación, se presentan los códigos que componen el Back-End de la plataforma Web SILAIN.

5.1.1.1. Conexión a la base de datos

```
const { Pool } = require('pg')

var pool;

module.exports = {
  getPool: () => {
    if (pool) {
      return pool;
    } else {
      pool = new Pool({
        user: 'postgres',
        host: 'localhost',
        database: 'silain',
        password: 'p',
        port: 5432
      });
      console.log('Connected to DB');
      return pool;
    }
  }
}
```

5.1.1.2. Rutas

5.1.1.2.1. article.js

```
const express = require('express')
const router = express.Router()
const pg = require('../db/database.js').getPool();

function swapCoors(data) {
  for (let i in data) {
    var aux = JSON.parse(data[i].poly).coordinates[0][0];
```

```

    for (let j = 0; j < aux.length; j++) {
      var row = aux[j];
      var new_row = [row[1], row[0]];
      aux[j] = new_row;
    }
    data[i].poly = aux;
  }
  return data;
}

function formatCentroidPoint(data) {
  var { centroid } = data[0];
  var s = centroid.split(" ");
  var s2 = s[0].split("(");
  var c = [parseFloat(s[1]), parseFloat(s2[1])];
  data[0].centroid = c;

  return data;
}

router.get('/get/:id', async (req, res) => {

  var id = req.params.id;

  const query = {
    text: "select * from metafull where idmetadato=$1;",
    values: [id]
  }

  try {
    const result = await pg.query(query);
    var queryresults = result.rows[0];

    const { publico } = queryresults;

    var queryFin = {
      text: "select st_asgeojson(geom) as poly, st_astext(st_centroid(geom)) as centroid from finca where finca = $1;",
      values: [queryresults.finca]
    }

    var dataFin = await pg.query(queryFin);
    dataFin = await swapCoors(dataFin.rows);
    dataFin = await formatCentroidPoint(dataFin);
  }

```

```

        // console.log(result.rows);
        res.status(200).send({ info: queryresults, finca: dataFin, publico: publico });
    } catch (e) {
        console.log(e);
        res.sendStatus(400);
    }
})

router.get('/get/:id/:userid', async (req, res) => {

    var id = req.params.id;
    var userid = req.params.userid;

    const query = {
        text: "select * from metafull where idmetadato = $1;",
        values: [id]
    }

    try {
        const result = await pg.query(query);
        var queryresults = result.rows[0];

        var statePublico = true;
        const { publico } = queryresults;
        if (!publico) {
            const queryConfirmDist = {
                text: "select email_usuario, id_metadato from licencias where email_usuario = $1 and id_metadato = $2;",
                values: [userid, id]
            };
            const confirmDist = await pg.query(queryConfirmDist);
            if (confirmDist.rows.length == 0) {
                statePublico = false;
            }
        }
    }

    var queryFin = {
        text: "select st_asgeojson(geom) as poly, st_astext(st_centroid(geom)) as centroid from finca where finca = $1;",
        values: [queryresults.finca]
    }

    var dataFin = await pg.query(queryFin);
    dataFin = await swapCoors(dataFin.rows);
    dataFin = await formatCentroidPoint(dataFin);

```

```

        res.status(200).send({ info: queryresults, finca: dataFin, publico: statePublico
    });
    } catch (e) {
        console.log(e);
        res.sendStatus(400);
    }
})

```

```

router.get('/download/:id', async (req, res) => {

    var id = req.params.id;
    var path = "D:/Datos SILAIN";

    const query = {
        text: "select url from metadato where idmetadato = $1",
        values: [id]
    }

    try {
        const result = await pg.query(query);
        var url = result.rows[0].url;
    } catch (e) {
        console.log(e);
        res.sendStatus(400);
    }

    res.download(path + url, err => {
        if (err) {
            console.log(err);
            res.sendStatus(400);
        }
    });
})

```

```

router.get('/download/filename/:id', async (req, res) => {

    var id = req.params.id;

    const query = {
        text: "select url from metadato where idmetadato = $1",
        values: [id]
    }

```

```

    try {
      const result = await pg.query(query);
      var url = result.rows[0].url;
      var splitted = url.split("/");
      var filename = splitted[splitted.length - 1];
      res.status(200).send({ filename: filename });
    } catch (e) {
      console.log(e);
      res.sendStatus(400);
    }
  })

  router.post('/proposito', async (req, res) => {

    const { id_usuario, id_metadato, proposito } = req.body;
    const query = {
      text: "insert into descargas (email_usuario, id_metadato, proposito, fecha,
hora) values ($1, $2, $3, current_timestamp, current_timestamp)",
      values: [id_usuario, id_metadato, proposito]
    }

    try {
      const result = await pg.query(query);
      res.status(200);
    } catch (e) {
      console.log(e);
      res.sendStatus(400);
    }
  })

  router.post('/crear', async (req, res) => {
    if (req.files) {
      var file = req.files.file;
      var filename = file.name;

      let { titulo, resumen, descripcion, lote, fase, pclave, publico, filters } = req.body;
      filters = JSON.parse(filters);
      let { categoria, subcategoria, tipo, formato, finca } = filters;

      let tamaño = file.size + " B";
      let url = "/root/" + categoria + "/" + subcategoria + "/" + filename;

      const queryID = {
        text: "select count(*) from metadato"
      }
    }
  })

```

```

const querySubcategorialD = {
  text: "select idsubcategoria from categoria inner join subcategoria on
idcategoria = categoria_idcategoria where categoria = $1 and subcategoria = $2",
  values: [categoria, subcategoria]
}

const queryFincalD = {
  text: "select idfinca from finca where finca = $1",
  values: [finca]
}

let ID, subID, finID;

try {
  const resultID = await pg.query(queryID);
  ID = parseInt(resultID.rows[0].count) + 1;

  const resultSubcategorialD = await pg.query(querySubcategorialD);
  subID = parseInt(resultSubcategorialD.rows[0].idsubcategoria);

  const resultFincalD = await pg.query(queryFincalD);
  finID = parseInt(resultFincalD.rows[0].idfinca);

} catch (e) {
  console.log(e);
  res.sendStatus(400);
}

const queryInsertMetadato = {
  text: "INSERT INTO metadato(idmetadato, titulo, pclave, creado,
disponibilidad, resumen, descripcion, " +
      "tipo, lote, fase, publico, formato, tamano, url) " +
      "VALUES($1, $2, $3, current_timestamp, current_timestamp, $4, $5, $6,
$7, $8, $9, $10, $11, $12)",
  values: [ID, titulo, pclave, resumen, descripcion, tipo, lote, fase, publico,
formato, tamano, url]
}

const queryInsertSub = {
  text: "INSERT INTO
metadato_has_subcategoria(subcategoria_idsubcategoria, metadato_idmetadato)
VALUES ($1, $2)",
  values: [subID, ID]
}

```

```

const queryInsertFin = {
  text: "INSERT INTO metadatos_has_finca(finca_idfinca,
metadato_idmetadato) VALUES ($1, $2)",
  values: [finID, ID]
}

try {
  await pg.query(queryInsertMetadato);
  await pg.query(queryInsertSub);
  await pg.query(queryInsertFin);
} catch (e) {
  console.log(e);
  res.sendStatus(400);
}
console.log("D:/Datos SILAIN/" + url);
file.mv("D:/Datos SILAIN/" + url, err => {
  if (err) {
    res.send(err);
  } else {
    res.send({});
  }
})
}
})

```

```
module.exports = router;
```

5.1.1.2.2. basicSearch.js

```

const express = require('express')
const router = express.Router()
const pg = require('../db/database.js').getPool();

function getHowMany(filters) {
  j = 0;
  for (let i in filters) {
    if (filters[i] !== "Select") {
      j++;
    }
  }
  return { counter: j };
}

```

```

function getTextWithFilters(text, filters) {
  var s = "";
  var c = false;
  j = 1;
  for (let i in filters) {
    if (filters[i] !== "Select") {
      j++;
      if (c) {
        s += " and";
        c = false;
      }
      s += " " + i + " = $" + j + "";
      c = true
    }
  }
  if (s !== "") {
    text = text + " and" + s;
  }
  return text;
}

```

```

function getValuesFromFilters(filters, word) {
  var vals = [`${word}%`]
  for (let i in filters) {
    if (filters[i] !== "Select") {
      vals.push(filters[i])
    }
  }
  return vals;
}

```

```

router.post('/search_by_filter', async (req, res) => {

  const { filters, word, currentPage } = req.body;

  var { counter } = getHowMany(filters);

  var limitResults = 20;
  var initPage = currentPage * limitResults;

  if (counter == 0 && word == "") {

    res.status(200).send({ result: [], counts: { AC: 0, AP: 0, IC: 0, IP: 0, C: 0 } });
  } else {

```

```

if (counter > 0 && word == "") {

    var text = "select distinct idmetadato, titulo, formato, tamano, resumen, tipo,
creado, disponibilidad, publico from metafull where pclave iLike $1";
    var countText = "select distinct idmetadato, tipo from metafull where pclave iLike
$1";

    } else {
        var text = "select distinct idmetadato, titulo, formato, tamano, resumen, tipo,
creado, disponibilidad, publico from metafull where $1 %
ANY(STRING_TO_ARRAY(pclave, ' '))";
        var countText = "select distinct idmetadato, tipo from metafull where $1 %
ANY(STRING_TO_ARRAY(pclave, ' '))";

    }

    var query_text = getTextWithFilters(text, filters);
    query_text = query_text + " OFFSET " + initPage + " LIMIT " + limitResults;
    var values = getValuesFromFilters(filters, word);

    countText = getTextWithFilters(countText, filters);
    countText = "select tipo, count(*) from (" + countText + ") as foo GROUP BY
foo.tipo";
    var countValues = getValuesFromFilters(filters, word);

    var query = { text: query_text, values: values };
    var countQuery = {text: countText, values: countValues};

    try {
        const result = await pg.query(query);
        const resultCounts = await pg.query(countQuery);

        let AC = 0, AP = 0, IC = 0, IP = 0, C = 0;

        for (let i in resultCounts.rows) {
            var tipo = resultCounts.rows[i].tipo;
            var count = parseInt(resultCounts.rows[i].count);

            switch (tipo) {
                case "Archivo crudo":
                    AC = count;
                    break;
                case "Archivo procesado":
                    AP = count;
                    break;
            }
        }
    }

```

```

    case "Imagen cruda":
      IC = count;
      break;
    case "Imagen procesada":
      IP = count;
      break;
    case "Compilación":
      C = count;
      break;
    default:
      break;
  }
}

let totalResults = AC + AP + IC + IP + C;
let npages = totalResults / limitResults ;

let pages = [];
for (let i = 0; i < npages; i++) {
  pages.push(i);
}

res.status(200).send({
  totalResults: totalResults,
  result: result.rows,
  pages: pages,
  counts: { AC: AC, AP: AP, IC: IC, IP: IP, C: C }
});

} catch (e) {
  console.log(e);
  res.sendStatus(400);
}
}
});

```

```
module.exports = router;
```

5.1.1.2.3. filters.js

```

const express = require('express')
const router = express.Router()
const pg = require('../db/database.js').getPool();

```

```

function processCats(catsubs) {

    var current = catsubs[0].idcategoria;
    var cats = [catsubs[0].categoria];

    for (let i = 0; i < catsubs.length; i++) {
        if (catsubs[i].idcategoria != current) {
            current = catsubs[i].idcategoria;
            cats.push(catsubs[i].categoria)
        }
    }
    return cats;
}

function processSubs(catsubs) {

    var current = catsubs[0].idcategoria;
    var subs = [];
    var aux = [];

    for (let i = 0; i < catsubs.length; i++) {
        if (catsubs[i].idcategoria != current) {
            current = catsubs[i].idcategoria;
            subs.push(aux);
            aux = [];
            aux.push(catsubs[i].subcategoria);
        } else {
            aux.push(catsubs[i].subcategoria);
        }
    }
    subs.push(aux);
    return subs;
}

function processMunicipios(municipios) {

    var muns = [];
    for (let i in municipios) {
        muns.push(municipios[i].municipio);
    }
    return muns;
}

function processFincas(fincas) {

```

```

var current = fincas[0].municipio_idmunicipio;
var fins = [];
var aux = [];

for (let i = 0; i < fincas.length; i++) {
  if (fincas[i].municipio_idmunicipio !== current) {
    current = fincas[i].municipio_idmunicipio;
    fins.push(aux);
    aux = [];
    aux.push(fincas[i].finca);
  } else {
    aux.push(fincas[i].finca);
  }
}
fins.push(aux);
return fins;
}

```

```

function extractFiltersOnMap(array) {

  var currentDept = array[0].departamento;
  var currentMuni = array[0].municipio;
  var currentFin = array[0].finca;

  var departamentos = [currentDept];
  var municipios = [];
  var fincas = [];

  var munis_for_dept = [currentMuni];
  var fins_for_munis = [currentFin];

  for (let i = 0; i < array.length; i++) {

    if (currentDept !== array[i].departamento) {
      currentDept = array[i].departamento;
      departamentos.push(array[i].departamento);

      municipios.push(munis_for_dept);
      munis_for_dept = [];
    }

    if (currentMuni !== array[i].municipio) {
      currentMuni = array[i].municipio;
      munis_for_dept.push(array[i].municipio);
    }
  }
}

```

```

        fincas.push(fins_for_munis);
        fins_for_munis = [];
    }

    if (currentFin !== array[i].finca) {
        currentFin = array[i].finca;
        fins_for_munis.push(array[i].finca);
    }
}

municipios.push(munis_for_dept);
fincas.push(fins_for_munis);

return { departamentos, municipios, fincas };
}

function getHashCultivo_Departamento(array) {

    hash = {};

    var c_cul = array[0].cultivo;
    var c_dep = array[0].departamento;
    var deptos = [c_dep];

    for (let i = 0; i < array.length; i++) {

        if (c_cul !== array[i].cultivo) {
            c_cul = array[i].cultivo;
            deptos = [];
        }

        if (c_dep !== array[i].departamento) {
            c_dep = array[i].departamento;
            deptos.push(c_dep);
        }
        hash[c_cul] = deptos;
    }

    return hash;
}

function getHashDepartamento_Municipio(array) {

```

```

hash = {};

var c_dep = array[0].departamento;
var c_mun = array[0].municipio;
var munis = [c_mun];

for (let i = 0; i < array.length; i++) {

    if (c_dep !== array[i].departamento) {
        c_dep = array[i].departamento;
        munis = [];
    }

    if (c_mun !== array[i].municipio) {
        c_mun = array[i].municipio;
        munis.push(c_mun);
    }

    hash[c_dep] = munis;

}

return hash;
}

function getHashMunicipio_Finca(array) {

    hash = {};

    var c_mun = array[0].municipio;
    var c_fin = array[0].finca;
    var fins = [c_fin];

    for (let i = 0; i < array.length; i++) {

        if (c_mun !== array[i].municipio) {
            c_mun = array[i].municipio;
            fins = [];
        }

        if (c_fin !== array[i].finca) {
            c_fin = array[i].finca;
            fins.push(c_fin);
        }

    }

}

```

```

    hash[c_mun] = fins;

}

return hash;
}

function getDepartamentoList(list) {

    var array = [];
    for (let i = 0; i < list.length; i++) {
        array.push(list[i].departamento);
    }
    return array;
}

router.get('/', async (req, res) => {

    const catsubs = {
        text: "select * from categoria inner join subcategoria on categoria_idcategoria =
idcategoria order by idcategoria;",
    }

    const municipios = {
        text: "select idmunicipio, municipio from municipio;"
    }

    const tipos = {
        text: "select DISTINCT ON(tipo) tipo from metadato;"
    }

    const formatos = {
        text: "select DISTINCT ON(formato) formato from metadato;"
    }

    const fincas = {
        text: "select municipio_idmunicipio, finca from finca order by 1;"
    }

    try {
        const result_catsubs = await pg.query(catsubs);
        const result_municipios = await pg.query(municipios);
        const result_tipos = await pg.query(tipos);
        const result_formatos = await pg.query(formatos);
        const result_fincas = await pg.query(fincas);

```

```

    var cats = processCats(result_catsubs.rows);
    var subs = processSubs(result_catsubs.rows);
    var munis = processMunicipios(result_municipios.rows);
    var fins = processFincas(result_fincas.rows);

    res.status(200).send({ cats: cats, subs: subs, municipios: munis, tipos:
result_tipos.rows, formatos: result_formatos.rows, fincas: fins });
  } catch (e) {
    console.log(e);
    res.sendStatus(400);
  }
})

router.get('/ubicacion', async (req, res) => {

  const cults = {
    text: "select distinct on(cultivo) cultivo from finca;"
  }

  const catsubs = {
    text: "select * from categoria inner join subcategoria on categoria_idcategoria =
idcategoria order by idcategoria;",
  }

  const tipos = {
    text: "select DISTINCT ON(tipo) tipo from metadato;"
  }

  const formatos = {
    text: "select DISTINCT ON(formato) formato from metadato;"
  }

  try {
    const result_cults = await pg.query(cults);
    const result_catsubs = await pg.query(catsubs);
    const result_tipos = await pg.query(tipos);
    const result_formatos = await pg.query(formatos);

    var cats = processCats(result_catsubs.rows);
    var subs = processSubs(result_catsubs.rows);

```

```

    res.status(200).send({ cultivos: result_cults.rows, cats: cats, subs: subs, tipos:
result_tipos.rows, formatos: result_formatos.rows });
  } catch (e) {
    console.log(e);
    res.sendStatus(400);
  }
})

```

```

router.get('/onmap', async (req, res) => {

```

```

  const query = "select departamento from departamento";

```

```

  const query2 = "select departamento, municipio from (select * from departamento
inner join municipio on iddepartamento = departamento_iddepartamento) AS foo
inner join finca on foo.idmunicipio = municipio_idmunicipio order by departamento;";

```

```

  const query3 = "select municipio, finca from (select * from departamento inner join
municipio on iddepartamento = departamento_iddepartamento) AS foo inner join
finca on foo.idmunicipio = municipio_idmunicipio order by municipio;";

```

```

  const result = await pg.query(query);
  const result2 = await pg.query(query2);
  const result3 = await pg.query(query3);

```

```

  var hash_Dept = getDepartamentoList(result.rows);
  var hash_Muni = getHashDepartamento_Municipio(result2.rows);
  var hash_Fin = getHashMunicipio_Finca(result3.rows);

```

```

  res.json({
    departamentos: hash_Dept,
    municipios: hash_Muni,
    fincas: hash_Fin,
  });
})

```

```

module.exports = router;

```

5.1.1.2.4. licence.js

```

const express = require('express')
const router = express.Router()
const pg = require('../db/database.js').getPool();

```

```

router.post('/add', async (req, res) => {

```

```

var { iduser, idarticulo } = req.body;

const query = {
  text: "INSERT into licencias (email_usuario, id_metadato) VALUES ($1, $2);",
  values: [iduser, idarticulo]
}

try {
  await pg.query(query);
  res.status(200).send({});
} catch (e) {
  console.log(e);
  res.sendStatus(400);
}
})

```

```

module.exports = router;

```

5.1.1.2.5. makepublic.js

```

const express = require('express')
const router = express.Router()
const pg = require('../db/database.js').getPool();

router.post('/', async (req, res) => {

  var { id, val } = req.body;

  const query = {
    text: "UPDATE metadato SET publico = $1 where idmetadato = $2",
    values: [val, id]
  }

  try {
    await pg.query(query);
    res.status(200).send({});
  } catch (e) {
    console.log(e);
    res.sendStatus(400);
  }
})

module.exports = router;

```

5.1.1.2.6. map.js

```
const express = require('express')
const router = express.Router()
const pg = require('../db/database.js').getPool();

function getHowMany(filters) {
  j = 0;
  for (let i in filters) {
    if (filters[i] !== "Select") {
      j++;
    }
  }
  return { counter: j };
}

function getTextWithFilters(text, filters) {
  var s = "";
  var c = false;
  j = 1;
  for (let i in filters) {
    if (filters[i] !== "Select") {
      j++;
      if (c) {
        s += " and";
        c = false;
      }
      s += " " + i + " = $" + j + "";
      c = true
    }
  }

  if (s !== "") {
    text = text + " and" + s;
  }
  return text;
}

function getValuesFromFilters(filters, ubication) {
  var vals = [ubication]
  for (let i in filters) {
    if (filters[i] !== "Select") {
      vals.push(filters[i])
    }
  }
}
```

```

    return vals;
}

function swapCoors(data) {
  for (let i in data) {
    var aux = JSON.parse(data[i].poly).coordinates[0][0];
    for (let j = 0; j < aux.length; j++) {
      var row = aux[j];
      var new_row = [row[1], row[0]];
      aux[j] = new_row;
    }
    data[i].poly = aux;
  }
  return data;
}

function formatCentroidPoint(data) {
  var { centroid } = data[0];
  var s = centroid.split(" ");
  var s2 = s[0].split("(");
  var c = [parseFloat(s[1]), parseFloat(s2[1])];
  data[0].centroid = c;

  return data;
}

router.post('/ubication_by_filter', async (req, res) => {

  const { filters, ubication, ubi_type, currentPage } = req.body;

  var { counter } = getHowMany(filters);

  var limitResults = 20;
  var initPage = currentPage * limitResults;

  if (counter == 0 && ubication == "") {
    res.status(200).send({ result: [], counts: { AC: 0, AP: 0, IC: 0, IP: 0, C: 0 } });
  } else {

    var text = "select idmetadato, titulo, formato, tamano, resumen, tipo, publico from metafull where " + ubi_type + " = $1";
    var countText = "select tipo from metafull where " + ubi_type + " = $1";

    var query_text = getTextWithFilters(text, filters);

```

```

query_text = query_text + " OFFSET " + initPage + " LIMIT " + limitResults;
var values = getValuesFromFilters(filters, ubicacion);

countText = getTextWithFilters(countText, filters);
countText = "select tipo, count(*) from (" + countText + ") as foo GROUP BY
foo.tipo";
var countValues = getValuesFromFilters(filters, ubicacion);

var query = { text: query_text, values: values };
var countQuery = { text: countText, values: countValues };

try {
  const result = await pg.query(query);
  const resultCounts = await pg.query(countQuery);

  let AC = 0, AP = 0, IC = 0, IP = 0, C = 0;

  for (let i in resultCounts.rows) {
    var tipo = resultCounts.rows[i].tipo;
    var count = parseInt(resultCounts.rows[i].count);

    switch (tipo) {
      case "Archivo crudo":
        AC = count;
        break;
      case "Archivo procesado":
        AP = count;
        break;
      case "Imagen cruda":
        IC = count;
        break;
      case "Imagen procesada":
        IP = count;
        break;
      case "Compilación":
        C = count;
        break;
      default:
        break;
    }
  }

  let totalResults = AC + AP + IC + IP + C;
  let npages = totalResults / limitResults;

```

```

    let pages = [];
    for (let i = 0; i < npages; i++) {
        pages.push(i);
    }

    res.status(200).send({
        totalResults: totalResults,
        result: result.rows,
        pages: pages,
        counts: { AC: AC, AP: AP, IC: IC, IP: IP, C: C }
    });

    } catch (e) {
        console.log(e);
        res.sendStatus(400);
    }
    }
    });

router.post('/getpoly', async (req, res) => {

    const { filters } = req.body;

    var { counter } = getHowMany(filters);

    if (counter === 0) {
        res.status(200).send({ departamento: [], municipio: [], finca: [] });
    } else {

        const { departamento, municipio, finca } = filters;

        try {

            var dataDept = [];
            var dataMuni = [];
            var dataFin = [];

            if (departamento !== 'Select') {
                var query = {
                    text: "select      departamento,      st_asgeojson(geom)      as      poly,
st_astext(st_centroid(geom)) as centroid from departamento where departamento =
$1;",
                    values: [departamento]
                }
                dataDept = await pg.query(query);
            }
        }
    }
}

```

```

    dataDept = await swapCoors(dataDept.rows);
    dataDept = await formatCentroidPoint(dataDept);
  }

  if (municipio !== 'Select') {
    var query = {
      text: "select municipio, st_asgeojson(geom) as poly,
st_astext(st_centroid(geom)) as centroid from municipio where municipio = $1;",
      values: [municipio]
    }
    dataMuni = await pg.query(query);
    dataMuni = await swapCoors(dataMuni.rows);
    dataMuni = await formatCentroidPoint(dataMuni);
  }

  if (finca !== 'Select') {
    var queryFin = {
      text: "select finca, st_asgeojson(geom) as poly, st_astext(st_centroid(geom))
as centroid from finca where finca = $1;",
      values: [finca]
    }
    dataFin = await pg.query(queryFin);
    dataFin = await swapCoors(dataFin.rows);
    dataFin = await formatCentroidPoint(dataFin);
  }

  res.status(200).send({
    departamento: dataDept,
    municipio: dataMuni,
    finca: dataFin
  });

} catch (e) {
  console.log(e);
  res.sendStatus(400);
}
})

router.post('/finca_closer', async (req, res) => {

  const marker = req.body.marker;

  var query = {

```

```

    text: "select finca, ROUND(CAST(st_distance(st_transform(geom,3115),
st_transform(st_setsrid(st_makepoint($1,$2),4326),3115)) / 1000 as numeric), 3) as
distancia_km from finca ORDER BY distancia_km",
    values: [marker[1], marker[0]]
  }

  try {
    var data = await pg.query(query);
  } catch (e) {
    console.log(e);
    res.sendStatus(400);
  }

  res.json(data.rows);
})

module.exports = router;

```

5.1.1.2.7. users.js

```

const express = require('express')
const router = express.Router()
const pg = require('../db/database.js').getPool();

router.post("/register", async (req, res) => {

  const { nombres, apellidos, email, pais, departamento, ciudad, institucion,
ocupacion, password } = req.body;

  const query = {
    text: "INSERT INTO usuario (nombres, apellidos, email, pais, departamento,
ciudad, institucion, ocupacion, password, role) VALUES ($1, $2, $3, $4, $5, $6, $7,
$8, $9, 2)",
    values: [nombres, apellidos, email, pais, departamento, ciudad, institucion,
ocupacion, password]
  }
  try {
    const result = await pg.query(query);
    res.status(200).send({ msg: "success" });
  } catch (e) {
    console.log(e);
    res.sendStatus(400);
  }
})

```

```

router.post("/login", async (req, res) => {

  const { user, password } = req.body;

  const query = {
    text: "SELECT email, role from usuario where email=$1 and password=$2",
    values: [user, password]
  }

  try {
    const result = await pg.query(query);
    var email = result.rows[0].email;
    var role = result.rows[0].role;
    if (email) {
      res.status(200).send({ user_id: email, role: role});
    } else {
      res.sendStatus(400);
    }
  } catch (e) {
    console.log(e);
    res.sendStatus(400);
  }
})

router.get("/user/:id", async (req, res) => {

  var id = req.params.id;

  const query = {
    text: "SELECT nombres, apellidos, email, pais from usuario where email=$1",
    values: [id]
  }
  try {
    const result = await pg.query(query);
    var id = result.rows[0];
    if (id) {
      res.status(200).send({ user: result.rows[0] });
    } else {
      res.sendStatus(400);
    }
  }
  } catch (e) {
    console.log(e);
  }
}

```

```

        res.sendStatus(400);
    }
})

router.get("/user/des_hist/:id", async (req, res) => {

    var id = req.params.id;

    const query = {
        text: "select titulo, idmetadato, fecha as orderfecha, hora as orderhora,
to_char(fecha, 'DD/MM/YYYY') as fecha, to_char(hora, 'HH24:MI:SS') as hora from
descargas inner join metadato on id_metadato=idmetadato where email_usuario=$1
order by orderfecha DESC, orderhora DESC;",
        values: [id]
    }
    try {
        var hist = []
        const result = await pg.query(query);
        hist = result.rows;
        res.status(200).send({ hist: hist });
    } catch (e) {
        console.log(e);
        res.sendStatus(400);
    }
})

module.exports = router;

```

5.1.1.3. Índice del Back-End

```

const express = require('express');
const morgan = require('morgan');
const bodyParser = require('body-parser');
const upload = require('express-fileupload');
const cors = require('cors');
const routes = require('./routes/index.js');
const app = express();

app.set('port',process.env.PORT || 8000);
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended: false}));
app.use(upload())
app.use(morgan('dev'));

```

```

app.use(cors());
app.use('/', routes);

app.listen(app.get('port'), () => {
  console.log('Backend Running on port:', app.get('port'));
})

```

5.1.2. Códigos del Front-End de SILAIN

Así como con el Back-End, en esta sección se exhiben los códigos con los cuales está construido el Fron-End.

5.1.2.1. Componentes

5.1.2.1.1. CreateFile.jsx

```

import CustomFilters from 'components/CustomFilters';
import axios from 'axios';
import api from "../api_route.js";
import {
  Button, Form, FormGroup, Row, Col, Input
} from 'reactstrap';

class CreateFile extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      titulo: "",
      resumen: "",
      descripcion: "",
      lote: "",
      fase: "",
      pclave: "",
      publico: true,
      file: "",
      filters: {
        categoria: "Select",
        subcategoria: "Select",
        municipio: "Select",
        tipo: "Select",
        formato: "Select",
        finca: "Select",
      },
    }
  }
}

```

```

validateFilters() {
  for (let i in this.state.filters) {
    if (this.state.filters[i] !== "Select") {
      return false;
    }
  }
  return true;
}

validateParameters() {
  return this.state.titulo !== "" &&
    this.state.resumen !== "" &&
    this.state.descripcion !== "" &&
    this.state.pclave !== "" &&
    this.state.file !== ""
}

sendForm() {
  var formData = new FormData();

  formData.append("file", this.state.file, this.state.file.name);
  formData.append("titulo", this.state.titulo);
  formData.append("resumen", this.state.resumen);
  formData.append("descripcion", this.state.descripcion);
  formData.append("lote", this.state.lote);
  formData.append("fase", this.state.fase);
  formData.append("pclave", this.state.pclave);
  formData.append("publico", this.state.publico);
  formData.append("filters", JSON.stringify(this.state.filters));

  axios.post(api.route + "/article/crear", formData, {
    headers: {
      'Content-Type': 'multipart/form-data'
    }
  })
  .then(res => {
    document.getElementById("createForm").reset();
    this.setState({
      titulo: "",
      resumen: "",
      descripcion: "",
      lote: "",
      fase: "",
      pclave: "",

```

```

        publico: true,
        file: "",
    })
    alert("Cargado exitosamente");
  })
  .catch(err => {
    console.log(err);
  })
}

handleInput(e) {
  this.setState({ [e.target.name]: e.target.value });
}

handleSubmit(e) {
  e.preventDefault();
  this.sendForm();
}

handleFile(e) {
  let file = e.target.files[0];
  this.setState({ "file": file });
}

getFilters(obj) {
  this.setState({ filters: obj });
}

render() {
  return (
    <div>
      <center>
        <h4>Carga de datos y metadatos</h4>
      </center>
      <Form onSubmit={this.handleSubmit.bind(this)} id="createForm">
        <FormGroup>
          <CustomFilters getFilters={this.getFilters.bind(this)} />
          <br />
          <Row>
            <Col>
              <b>Titulo</b>
              <Input type="text" placeholder="Titulo" name="titulo"
onChange={this.handleInput.bind(this)} required />
            </Col>
          </Row>
        </FormGroup>
      </Form>
    </div>
  );
}

```

```

<Row>
  <Col>
    <b>Resumen</b>
    <Input type="text" placeholder="Resumen" name="resumen"
onChange={this.handleInput.bind(this)} required />
  </Col>
  <Col>
    <b>Descripcion</b>
    <Input type="text" placeholder="Descripcion"
name="descripcion" onChange={this.handleInput.bind(this)} required />
  </Col>
</Row>
<Row>
  <Col>
    <b>Lote</b>
    <Input type="text" placeholder="Lote" name="lote"
onChange={this.handleInput.bind(this)} required />
  </Col>
  <Col>
    <b>Fase</b>
    <Input type="number" placeholder="Fase" name="fase"
onChange={this.handleInput.bind(this)} required />
  </Col>
</Row>
<Row>
  <Col>
    <b>Palabras Clave</b>
    <Input type="text" placeholder="Palabras clave separadas por
coma (,)" name="pclave" onChange={this.handleInput.bind(this)} required />
  </Col>
  <Col>
    <b>Acceso:</b> <Button
color={this.state.publico ? "success" : "danger"} onClick={()
=> this.setState({ publico: !this.state.publico })} required>
      {this.state.publico ? "publico" : "privado"}
    </Button>
  </Col>
</Row>
</FormGroup>
<Input type="file" name="file" onChange={this.handleFile.bind(this)}
required />
<center>
  <Button color="warning" block>Create</Button>
</center>
</Form>

```

```

        <hr />
      </div>
    )
  }
}

export default CreateFile;

```

5.1.2.1.2. CreateLicence.jsx

```

import React from "react";
import axios from 'axios';
import api from "../api_route.js";

import {
  Button, Form, FormGroup, Row, Col, Input
} from 'reactstrap';

class CreateLicence extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      iduser: "",
      idarticulo: 0,
    }
  }

  handleInput(e) {
    this.setState({ [e.target.name]: e.target.value });
  }

  handleSubmit(e) {
    e.preventDefault();
    axios.post(api.route + "/licence/add", this.state)
      .then(res => {
        document.getElementById("licenceForm").reset();
        alert("Licencia asignada con exito");
      })
      .catch(err => {
        alert("Error");
      })
  }

  render() {

```

```

    return (
      <div>
        <center>
          <h4>Autorización de descargas</h4>
        </center>
        <Form onSubmit={this.handleSubmit.bind(this)} id="licenceForm">
          <FormGroup>
            <Row>
              <Col>
                <b>Email del usuario</b>
                <Input type="email" name="iduser"
onChange={this.handleInput.bind(this)} required />
              </Col>
              <Col>
                <b>Id del articulo</b>
                <Input type="number" name="idarticulo"
onChange={this.handleInput.bind(this)} required />
              </Col>
            </Row>
          </FormGroup>
          <center>
            <Button color="second" block>Create</Button>
          </center>
        </Form>
        <hr />
      </div>
    )
  }
}

```

export default CreateLicence;

5.1.2.1.3. CustomFilters

```

import React from "react";
import axios from "axios";
import {
  Row, Col, Input
} from "reactstrap";

import api from "api_route.js";

class CustomFilters extends React.Component {
  constructor(props) {
    super(props);

```

```

this.state = {
  cats: [],
  subs: [],
  municipios: [],
  tipos: [],
  formatos: [],
  fincas: [],
  current_sub: [],
  current_fin: [],
  selections: {
    categoria: "Select",
    subcategoria: "Select",
    municipio: "Select",
    tipo: "Select",
    formato: "Select",
    finca: "Select"
  }
}

handleSelects(e) {
  var sele = { ...this.state.selections };
  sele[e.target.name] = e.target.value;
  this.props.getFilters(sele);
  this.setState({ selections: sele })
}

handleSelectCAT(e) {
  var sele = { ...this.state.selections };
  var val = e.target.value;
  var c_sub = [];

  if (val === "-1") {
    sele["categoria"] = "Select";
  } else {
    sele["categoria"] = this.state.cats[val];
    c_sub = this.state.subs[val]
  }
  sele["subcategoria"] = "Select";
  this.props.getFilters(sele);
  this.setState({ selections: sele, current_sub: c_sub });
}

handleSelectMUNICIPIO(e) {

```

```

var sele = { ...this.state.selections };
var val = e.target.value;
var c_fin = [];

if (val === "-1") {
  sele["municipio"] = "Select";
} else {
  sele["municipio"] = this.state.municipios[val];
  c_fin = this.state.fincas[val]
}
sele["finca"] = "Select";
this.props.getFilters(sele);
this.setState({ selections: sele, current_fin: c_fin });
}

componentDidMount() {

  const url = api.route + "/getfilters/";

  axios.get(url)
    .then(res => {
      var { cats, subs, municipios, tipos, formatos, fincas } = res.data;
      this.setState({ cats: cats, subs: subs, municipios: municipios, tipos: tipos,
formatos: formatos, fincas: fincas });
    })
    .catch(err => {
      console.log(err);
    })
  }

  render() {
    return (
      <div>

        <Row>
          <Col>
            <center>
              <b>Categoria</b>
            </center>
            <Input onChange={this.handleSelectCAT.bind(this)} type="select"
name="cat" id="exampleSelect">
              <option value={-1}>Select</option>
              {
                this.state.cats.map((e, i) => (

```

```

        <option value={i} key={i}>{e}</option>
      ))
    }
  </Input>
</Col>
<Col>
  <center>
    <b>Subcategoria</b>
  </center>
  <Input
    onChange={this.handleSelects.bind(this)}
value={this.state.selections.subcategoria} type="select" name="subcategoria"
id="exampleSelect">
    <option>Select</option>
    {
      this.state.current_sub.map((e, i) => (
        <option key={i}>{e}</option>
      ))
    }
  </Input>
</Col>
<Col>
  <center>
    <b>Municipio</b>
  </center>
  <Input
    onChange={this.handleSelectMUNICIPIO.bind(this)}
type="select" name="municipio" id="exampleSelect">
    <option value=-1>Select</option>
    {
      this.state.municipios.map((e, i) => (
        <option value={i} key={i}>{e}</option>
      ))
    }
  </Input>
</Col>
<Col>
  <center>
    <b>Finca</b>
  </center>
  <Input
    onChange={this.handleSelects.bind(this)}
value={this.state.selections.finca} type="select" name="finca" id="exampleSelect">
    <option>Select</option>
    {
      this.state.current_fin.map((e, i) => (
        <option key={i}>{e}</option>
      ))
    }
  </Input>
</Col>

```

```

    }
    </Input>
  </Col>
  <Col>
    <center>
      <b>Tipo</b>
    </center>
    <Input  onChange={this.handleSelects.bind(this)}  type="select"
name="tipo" id="exampleSelect">
      <option>Select</option>
      {
        this.state.tipos.map((e, i) => (
          <option key={i}>{e.tipo}</option>
        ))
      }
    </Input>
  </Col>
  <Col>
    <center>
      <b>Formato</b>
    </center>
    <Input  onChange={this.handleSelects.bind(this)}  type="select"
name="formato" id="exampleSelect">
      <option>Select</option>
      {
        this.state.formatos.map((e, i) => (
          <option key={i}>{e.formato}</option>
        ))
      }
    </Input>
  </Col>
</Row>
</div>
)
}
}

```

export default CustomFilters;

5.1.2.1.4. Footer.jsx

```
/*!  
  
=====br/>* Paper Dashboard React - v1.1.0  
=====br/>  
* Product Page: https://www.creative-tim.com/product/paper-dashboard-react  
* Copyright 2019 Creative Tim (https://www.creative-tim.com)  
  
* Licensed under MIT (https://github.com/creativetimofficial/paper-dashboard-react/blob/master/LICENSE.md)  
  
* Coded by Creative Tim  
  
=====br/>  
* The above copyright notice and this permission notice shall be included in all copies  
or substantial portions of the Software.  
  
*/  
  
import React from "react";  
import { Container, Row } from "reactstrap";  
  
import PropTypes from "prop-types";  
  
import { Link } from "react-router-dom";  
import silainLogo from "./silainLogo.jpg";  
  
class Footer extends React.Component {  
  render() {  
    return (  
      <footer  
        style={{  
          "backgroundColor": "white", "opacity": "0.9",  
          "bottom": "0px", "paddingTop": "10px"  
        }}  
      >  
        <Container fluid={this.props.fluid ? true : false}>  
          <Row>  
            <nav className="footer-nav">  
              <Link to="/acerca">Acerca de SILAIN</Link>  
            </nav>  
          </Row>  
        </Container>  
      </footer>  
    );  
  }  
}
```

```

    <div className="credits ml-auto">
      <div className="copyright">
        &copy; {1900 + new Date().getFullYear()}, made with{" "}
        <i className="fa fa-heart heart" /> by Creative Tim
      </div>
    </div>
  </nav>
  <div className="credits ml-auto">
    <a href="http://cibiofi.univalle.edu.co/index.php/es/inicio/">
      <img width="230px" src={silainLogo} alt="" />
    </a>
  </div>
</Row>
</Container>
</footer>
);
}
}

```

```

Footer.propTypes = {
  default: PropTypes.bool,
  fluid: PropTypes.bool
};

```

```
export default Footer;
```

5.1.2.1.5. Pagination.jsx

```
import React from "react";
```

```

class Pagination extends React.Component {
  render() {
    return (
      <center>
        {
          this.props.pages.map(ele => (
            <button className='ButtonLikeLinkSelected'
              key={ele}
              style={ele === this.props.currentPage ? { backgroundColor:
'#F47C00', color: 'white' } : {}}
              onClick={this.props.changePage.bind(this, ele)}>{ele}
            </button>
          ))
        }
      </center>
    )
  }
}

```

```

    )
  }
}

```

```
export default Pagination;
```

5.1.2.1.6. ResultsTable.jsx

```
import React from "react";
```

```
import {
  Table,
  Button
} from "reactstrap";
```

```
import axios from "axios";
import api from "../api_route.js";
```

```
import { NavLink } from "react-router-dom";
import auth from "components/auth/auth.js";
```

```
class ResultsTable extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      results: this.props.results,
    }
  }
}
```

```
  changeLabel(pos, id) {
```

```
    var r = [...this.state.results];
    var new_val = !r[pos].publico
    r[pos].publico = new_val;
```

```
    axios.post(api.route + "/makepublic/", { id: id, val: new_val })
      .then(res => {
        this.setState({ results: r });
      })
      .catch(err => {
        console.log(err);
      })
  }
```

```
  render() {
```

```

return (
  <div>
    <Table hover={true}>
      <thead>
        <tr>
          <th>#</th>
          <th>Titulo y resumen</th>
          <th>Tipo</th>
          <th>Formato</th>
          <th>Tamaño</th>
          <th>creado</th>
          <th>disponibilidad</th>
          <th>Acceso</th>
        </tr>
      </thead>
      <tbody>
        {
          this.state.results.map((e, i) => (
            <tr key={i}>
              <th scope="row">{(20 * this.props.currentPage) + i + 1}</th>
              <td>
                <b>
                  <NavLink to={"/article/" + e.idmetadato}
                    className="silain_green_links"
                  >
                    {e.titulo}
                  </NavLink>
                </b>
                <br />
                <i>{e.resumen}</i>
              </td>
              <td className="centerTd">
                {e.tipo}
              </td>
              <td className="centerTd">
                {e.formato} <br />
              </td>
              <td className="centerTd">
                {e.tamano}
              </td>
              <td className="centerTd">
                {e.creado} <br />
              </td>
              <td className="centerTd">
                {e.disponibilidad}
              </td>
            </tr>
          ))
        }
      </tbody>
    </Table>
  </div>
)

```

```

        </td>
        {
            auth.isAdmin() ?
                <td className="centerTd">
                    <Button    onClick={this.changeLabel.bind(this,    i,
e.idmetadato)} color={e.publico ? "success" : "danger"}>
                        {e.publico ? "Público" : "Privado"}
                    </Button>
                </td>
                :
                <td className="centerTd">
                    {e.publico ? "Público" : "Privado"} <br />
                </td>
            }
        </tr>
    ))
    }
</tbody>
</Table>
</div>
)
}
}

```

export default ResultsTable;

5.1.2.1.7. SNavBar.jsx

import React from "react";

```

import {
    Navbar,
    NavbarBrand,
    Row, Col
} from "reactstrap";

```

import { NavLink } from "react-router-dom";

import auth from "components/auth/auth.js";

```

class SNavBar extends React.Component {
    constructor(props) {
        super(props);
        this.state = {}
    }
}

```

```

render() {
  return (
    <Navbar color="dark" expand="md" className="navbar_silain_links">
      <NavbarBrand>
        <NavLink to="/">
          SILAIN

        </NavLink>
      </NavbarBrand>
      <Col md="12">
        <Row >
          <Col className="silain_link silain_link_yellow">
            <center>
              <NavLink to="/search/clave"
                activeStyle={{
                  fontWeight: "bold",
                  color: "#65A357",
                  fontSize: "14px",

                }}
                title="Realiza búsquedas escribiendo palabras clave">
                <i className="nc-icon nc-align-left-2" style={{ "fontSize": "18px" }}></i>
                Búsqueda por palabras Clave
              </NavLink>
            </center>
          </Col>
          <Col className="silain_link silain_link_orange">
            <center>
              <NavLink to="/search/propiedad"
                activeStyle={{
                  fontWeight: "bold",
                  color: "#98B503",
                  fontSize: "14px",

                }}
                title="Realiza búsquedas seleccionando una propiedad">
                <i className="nc-icon nc-bullet-list-67" style={{ "fontSize": "18px"
              </i></i>
              Exploración de datos
            </NavLink>
          </center>
        </Col>
        <Col className="silain_link silain_link_greenorange">
          <center>

```

[illegible]

```

        </Col>
      </Row>
    </Col>
  </Navbar>
)
}
}

```

```
export default SNavBar;
```

5.1.2.1.8. CustomFiltersByMap.jsx

```

import React from "react";
import axios from "axios";
import {
  Row, Col, Input
} from "reactstrap";

import api from "api_route.js";

class CustomFiltersByMap extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      cultivos: [],
      cats: [],
      subs: [],
      tipos: [],
      formatos: [],
      current_sub: [],
      current_fin: [],
      selections: {
        cultivo: "Select",
        categoria: "Select",
        subcategoria: "Select",
        tipo: "Select",
        formato: "Select",
      }
    }
  }

  handleSelects(e) {
    var sele = { ...this.state.selections };
    sele[e.target.name] = e.target.value;
    this.props.getFilters(sele);
  }
}

```

```

    this.setState({ selections: sele })
  }

  handleSelectCAT(e) {
    var sele = { ...this.state.selections };
    var val = e.target.value;
    var c_sub = [];

    if (val === "-1") {
      sele["categoria"] = "Select";
    } else {
      sele["categoria"] = this.state.cats[val];
      c_sub = this.state.subs[val]
    }
    sele["subcategoria"] = "Select";
    this.props.getFilters(sele);
    this.setState({ selections: sele, current_sub: c_sub });
  }

  componentDidMount() {

    const url = api.route + "/getfilters/ubication";

    axios.get(url)
      .then(res => {
        var { cultivos, cats, subs, tipos, formatos } = res.data;
        this.setState({
          cultivos: cultivos,
          cats: cats,
          subs: subs,
          tipos: tipos,
          formatos: formatos
        });
      })
      .catch(err => {
        console.log(err);
      })
  }

  render() {
    return (
      <div>

```

```

<Row>
  <Col>
    <b>Cultivo</b>
    <Input  onChange={this.handleSelects.bind(this)}    type="select"
name="cultivo" id="exampleSelect">
      <option>Select</option>
      {
        this.state.cultivos.map((e, i) => (
          <option key={i}>{e.cultivo}</option>
        ))
      }
    </Input>
  </Col>
  <Col>
    <b>Categoria</b>
    <Input  onChange={this.handleSelectCAT.bind(this)}  type="select"
name="cat" id="exampleSelect">
      <option value=-1>Select</option>
      {
        this.state.cats.map((e, i) => (
          <option value={i} key={i}>{e}</option>
        ))
      }
    </Input>
  </Col>
  <Col>
    <b>Subcategoria</b>
    <Input                                onChange={this.handleSelects.bind(this)}
value={this.state.selections.subcategoria}  type="select"  name="subcategoria"
id="exampleSelect">
      <option>Select</option>
      {
        this.state.current_sub.map((e, i) => (
          <option key={i}>{e}</option>
        ))
      }
    </Input>
  </Col>
  <Col>
    <b>Tipo</b>
    <Input  onChange={this.handleSelects.bind(this)}    type="select"
name="tipo" id="exampleSelect">
      <option>Select</option>
      {

```

```

        this.state.tipos.map((e, i) => (
          <option key={i}>{e.tipo}</option>
        ))
      }
    </Input>
  </Col>
  <Col>
    <b>Formato</b>
    <Input onChange={this.handleSelects.bind(this)} type="select"
name="formato" id="exampleSelect">
      <option>Select</option>
      {
        this.state.formatos.map((e, i) => (
          <option key={i}>{e.formato}</option>
        ))
      }
    </Input>
  </Col>
</Row>
</div>
)
}
}
export default CustomFiltersByMap;

```

5.1.2.1.9. FiltersOnMap.jsx

```

import React from "react";
import {
  Input
} from "reactstrap";

import api from "api_route.js";

class FiltersOnMap extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      depts: [],
      munis: {},
      fins: {},
      current_mun: [],
      current_fin: [],
      val_muni: -1,
      val_fin: -1,

```

```

        selections: {
            departamento: "Select",
            municipio: "Select",
            finca: "Select",
        }
    }
}

handleSelects(e) {
    var sele = { ...this.state.selections };
    var val = e.target.value;
    console.log(val);
    if (val === "-1") {
        val = -1;
        sele["finca"] = "Select";
    } else {
        sele["finca"] = this.state.current_fin[val];
    }
    this.props.getFilters(sele);
    this.setState({ selections: sele, val_fin: val })
}

handleSelectDEPT(e) {
    var sele = { ...this.state.selections };
    var val = e.target.value;
    var c_mun = [];

    if (val === "-1") {
        val = -1;
        sele["departamento"] = "Select";
    } else {
        sele["departamento"] = this.state.depts[val];
        c_mun = this.state.munis[sele["departamento"]];
    }
    sele["municipio"] = "Select";
    sele["finca"] = "Select";
    this.props.getFilters(sele);
    this.setState({
        selections: sele,
        current_mun: c_mun,
        current_fin: [],
        val_muni: -1,
        val_fin: -1
    });
}

```

```

handleSelectMUNI(e) {

    var sele = { ...this.state.selections };
    var val = e.target.value;
    var c_fin = [];

    if (val === "-1") {
        val = -1
        sele["municipio"] = "Select";
    } else {
        sele["municipio"] = this.state.current_mun[val];
        c_fin = this.state.fins[sele["municipio"]];
    }
    sele["finca"] = "Select";
    this.props.getFilters(sele);
    this.setState({
        selections: sele,
        current_fin: c_fin,
        val_muni: val,
        val_fin: -1
    });
}

async componentDidMount() {

    const res = await fetch(api.route + "/getfilters/onmap");
    var { departamentos, municipios, fincas } = await res.json();
    this.setState({
        depts: departamentos,
        munis: municipios,
        fins: fincas,
    });
}

render() {
    return (
        <div style={{ "width": "155px", "fontSize": "14px" }}>

            <b>Departamentos</b>
            <Input    onChange={this.handleSelectDEPT.bind(this)}    type="select"
id="exampleSelect">
                <option value={-1}>Select</option>
                {
                    this.state.depts.map((e, i) => (

```

```

        <option value={i} key={i}>{e}</option>
      ))
    }
  </Input>
  <b>Municipios</b>
  <Input
    onChange={this.handleSelectMUNI.bind(this)}
    value={this.state.val_muni} type="select" id="exampleSelect">
    <option value=-1>Select</option>
    {
      this.state.current_mun.map((e, i) => (
        <option value={i} key={i}>{e}</option>
      ))
    }
  </Input>
  <b>Fincas</b>
  <Input
    onChange={this.handleSelects.bind(this)}
    value={this.state.val_fin} type="select" name="finca" id="exampleSelect">
    <option value=-1>Select</option>
    {
      this.state.current_fin.map((e, i) => (
        <option value={i} key={i}>{e}</option>
      ))
    }
  </Input>
</div>
)
}
}

```

export default FiltersOnMap;

5.1.2.1.10. PropiedadByMap.jsx

import React from "react";

import {
 Row, Col, Alert, Badge
} from 'reactstrap';

import axios from "axios";
import ReactLoading from "react-loading";

import CustomFiltersByMap from "components/Map/CustomFiltersByMap.jsx";
import ResultsTableByMap from "components/Map/ResultsTableByMap";
import Pagination from "components/Pagination";

```

import api from "api_route.js";

class PropiedadByMap extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      results: [],
      counts_tipos: {},
      ubicacion: "",
      ubi_type: "",
      pages: [],
      currentPage: 0,
      totalResults: 0,
      filters: {
        cultivo: "Select",
        categoria: "Select",
        subcategoria: "Select",
        tipo: "Select",
        formato: "Select",
      },
      loading: false,
    }
  }

  allSelect() {
    for (let i in this.state.filters) {
      if (this.state.filters[i] !== "Select") {
        return false;
      }
    }
    return true;
  }

  getFilters(obj) {
    const basicURL = api.route + "/map/ubicacion_by_filter";
    this.setState({ loading: true, currentPage: 0 })
    axios.post(basicURL, { filters: obj, ubicacion: this.state.ubicacion, ubi_type:
this.state.ubi_type, currentPage: 0 })
      .then(res => {

        this.setState({
          results: res.data.result,
          counts_tipos: res.data.counts,
          loading: false,

```

```

        filters: obj,
        totalResults: res.data.totalResults,
        pages: res.data.pages
    });
    })
    .catch(err => {
        console.log(err);
    })
}

changePage(page) {
    this.props.changePage(page);
    this.setState({ currentPage: page });
}

componentDidMount() {
    this.setState(
        {
            counts_tipos: this.props.counts_tipos,
            results: this.props.results,
            ublication: this.props.ublication,
            ubi_type: this.props.ubi_type,
            pages: this.props.pages,
            currentPage: this.props.currentPage,
            totalResults: this.props.totalResults
        }
    )
}

render() {
    return (
        <div>
            <CustomFiltersByMap getFilters={this.getFilters.bind(this)} />
            {
                this.state.loading ? (
                    <center>
                        <ReactLoading type={"bars"} color={"#A5C80A"} />
                    </center>
                ) :
                <div>
                    <br />
                    <div>
                        <Alert color={this.state.results.length > 0 || this.allSelect() ?
"success" : "danger"}>
                            {
                                this.state.results.length > 0

```

```

        ? (
            <div>
                <h5><b>Resultados:
</b>{this.state.totalResults}</h5>
                <Row className="datsBigger">
                    <Col>
                        *Archivo          crudo:          <Badge
pill><b>{this.state.counts_tipos.AC}</b></Badge>
                    </Col>
                    <Col>
                        *Archivo          procesado:        <Badge
pill><b>{this.state.counts_tipos.AP}</b></Badge>
                    </Col>
                    <Col>
                        </Col>
                    </Row>
                    <Row className="datsBigger">
                        <Col>
                            *Imagen          cruda:          <Badge          pill>
<b>{this.state.counts_tipos.IC}</b></Badge>
                        </Col>
                        <Col>
                            *Imagen          procesada:        <Badge
pill><b>{this.state.counts_tipos.IP}</b></Badge>
                        </Col>
                        <Col>
                            *Compilación:          <Badge
pill><b>{this.state.counts_tipos.C}</b></Badge>
                        </Col>
                    </Row>
                </div>
            )
        : (
            this.allSelect() ?
                <h5>          <b>Selecciona          un          atributo          de
busqueda</b></h5>
                : <h5><b>No hay datos relacionados</b></h5>
        )
    }
</Alert>
</div>
{
    this.state.results.length > 0 ?
        (

```

```

        <div>
            <ResultsTableByMap      results={this.state.results}
currentPage={this.state.currentPage} />
            <Pagination            pages={this.state.pages}
currentPage={this.state.currentPage} changePage={this.changePage.bind(this)} />
        </div>
    ) : true
    }
</div>
}
</div>
)
}
};

```

export default PropiedadByMap;

5.1.2.1.11. ResultsTableByMap

```

import React from "react";

import {
    Table,
    Button
} from "reactstrap";

import axios from "axios";
import api from "../api_route.js";

import { NavLink } from "react-router-dom";
import auth from "components/auth/auth.js";

class ResultsTableByMap extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            results: this.props.results,
        }
    }

    changeLabel(pos, id) {

        var r = [...this.state.results];
        var new_val = !r[pos].publico
    }
}

```

```

r[pos].publico = new_val;

axios.post(api.route + "/makepublic/", { id: id, val: new_val })
  .then(res => {
    this.setState({ results: r });
  })
  .catch(err => {
    console.log(err);
  })
}

render() {
  return (
    <div>
      <Table hover={true}>
        <thead>
          <tr>
            <th>#</th>
            <th>Titulo y resumen</th>
            <th>Tipo</th>
            <th>Formato</th>
            <th>Tamaño</th>
            <th>Acceso</th>
          </tr>
        </thead>
        <tbody>
          {
            this.state.results.map((e, i) => (
              <tr key={i}>
                <th scope="row">{(20 * this.props.currentPage) + i + 1}</th>
                <td>
                  <b>
                    <NavLink to={"/article/" + e.idmetadato}
                      className="silain_green_links"
                    >
                      {e.titulo}
                    </NavLink>
                  </b>
                  <br />
                  <i>{e.resumen}</i>
                </td>
                <td className="centerTd">
                  {e.tipo}
                </td>
                <td className="centerTd">

```

```

        {e.formato} <br />
      </td>
      <td className="centerTd">
        {e.tamano}
      </td>
    {
      auth.isAdmin() ?
        <td className="centerTd">
          <Button onClick={this.changeLabel.bind(this, i,
e.idmetadato)} color={e.publico ? "success" : "danger"}>
            {e.publico ? "Público" : "Privado"}
          </Button>
        </td>
        : <td className="centerTd">
          {e.publico ? "Público" : "Privado"} <br />
        </td>
    }
  </tr>
))
}
</tbody>
</Table>
</div>
)
}
}

```

export default ResultsTableByMap;

5.1.2.1.12. Project1.jsx

```

import React from "react";
import { Accordion, Card } from "react-bootstrap";
import { Link } from "react-router-dom";
import diagrams from "./diagrams.js";
import mermaid from 'mermaid';
import Eimage from "./Eimage.png";
import Fimage from "./Fimage.png";
import Gimage from "./Gimage.png";
import O2 from "./O2.png";
import O3 from "./O3.png";
import R11 from "./R1.1.png";
import R12 from "./R1.2.png";
import R122 from "./R1.2.2.png";
import R13 from "./R1.3.png";

```

```

import R21 from "./R2.1.png";
import R22 from "./R2.2.png";
import R23 from "./R2.3.png";
import R24 from "./R2.4.png";
import R31 from "./R3.1.png";
import R312 from "./R3.1.2.png";
import R32 from "./R3.2.png";
import R33 from "./R3.3.png";
import R332 from "./R3.3.2.png";
import C1 from "./C1.png";
import C2 from "./C2.png";

```

```

class Project1 extends React.Component {

  constructor(props) {
    super(props);
    this.state = {
      obj: "",
      showObj: "",
    }
  }

  diagramEvent(e) {
    this.setState({ showObj: e });
  }

  showDiagram(d) {

    this.setState({ "obj": d });

    var output;

    if (d[0] === "O") {
      output = document.getElementById('output');
    } else {
      output = document.getElementById('outputResults');
    }

    var config = {
      startOnLoad: true,
      flowchart: {
        useMaxWidth: true,
        htmlLabels: true,
        curve: 'cardinal',

```

```

    },
    securityLevel: 'loose',
  };

  mermaid.initialize(config);

  var graphDefinition;

  switch (d) {
    case "O1":
      graphDefinition = diagrams.o1;
      break;
    case "O2":
      graphDefinition = diagrams.o2;
      break;
    case "O3":
      graphDefinition = diagrams.o3;
      break;
    case "R1":
      graphDefinition = diagrams.r1;
      break;
    case "R2":
      graphDefinition = diagrams.r2;
      break;
    case "R3":
      graphDefinition = diagrams.r3;
      break;
    default:
      break;
  }

  mermaid.render('theGraph', graphDefinition, (svgCode, bindFunctions) => {
    output.innerHTML = svgCode;
    bindFunctions()
  });
}

componentDidMount() {
  window.foo = e => { this.diagramEvent(e); }
}

render() {
  return (
    <div className="animated bounceIn fast">
      <center>

```

SUSTENTABILIDAD AMBIENTAL DEL MANEJO Y USO DEL AGUA EN CULTIVOS DE ARROZ DE RIEGO. CASO ESTUDIO: FINCA ARROCERA SAN JOSÉ, MUNICIPIO DE SANTANDER DE QUILICHAO, DEPARTAMENTO DEL CAUCA.

Trabajo de Grado presentado como requisito parcial para optar por el título de

Magister en Desarrollo Sustentable con énfasis en Ecosistemas Acuáticos

Realizado por: Claudia Patricia Vidal Puerta
Director: MAURICIO EDILBERTO RINCÓN ROMERO
Ingeniero Catastral y Geodesta M.Sc. Ph.D.
Codirector: OSCAR BUITRAGO BERMÚDEZ
Ingeniero Agrícola M.Sc. Ph.D.
[CONSULTAR TESIS](/article/421)

 Resumen

El cultivo de arroz inundado es uno de los agroecosistemas que ejerce mayor presión sobre el bien hídrico. Por ello, la evaluación de la sustentabilidad ambiental a partir del concepto de huella hídrica es importante ya que permite determinar cómo el manejo y uso del agua en el cultivo puede afectar la disponibilidad tanto en cantidad como en calidad. De ahí que, esta investigación parte de caracterizar el manejo y uso del agua en la producción primaria del arroz, el cual se define por las dimensiones culturales de los agricultores; posteriormente, se cuantifican los tres (3) componentes del indicador biofísico de sustentabilidad

Huella Hídrica y, finalmente, se determinan los efectos ambientales mediante índices de sustentabilidad e indicadores de calidad y contaminación del agua. El estudio se centró en la finca San José perteneciente a la arrocería La Esmeralda S.A.S., ubicada en la zona plana del municipio de Santander de Quilichao – Departamento del Cauca durante el periodo comprendido entre septiembre de 2018 - febrero de 2019. Los resultados evidencian que el manejo y uso del agua está determinado por la percepción de abundancia del bien hídrico que tiene el agricultor, hecho que conlleva a su excesivo uso para mantener altos rendimientos y disminuir la aplicación de agroquímicos. Por otro lado, la cuantificación de la huella hídrica en el periodo indicado fue de referida a de , cero (0) de y de , mostrando que las aguas verdes, en la temporada de alto régimen pluviométrico, satisfacen las necesidades hídricas de la planta de arroz, por lo cual no es necesario el riego. No obstante, a partir del diseño e implementación de la estación automática medidora de caudal se encontró que se incorporaron de agua de riego extraídos de los ríos y drenajes próximos para establecer la inundación permanente y garantizar el control residual de malezas; de este volumen retornaron a las fuentes hídricas naturales. Finalmente, la evaluación de la sustentabilidad ambiental del manejo y uso del agua verde, azul y gris muestra una condición crítica de contaminación por agentes microbiológicos, compuestos químicos y un estado hipertrófico por el alto contenido de nutrientes, con lo cual se contribuye al deterioro progresivo de la calidad del agua cuando se reintegran a los ríos.

```

</div>
</Accordion.Collapse>
</div>
<div>
  <Accordion.Toggle as={Card.Header} eventKey="1" style={{
"backgroundColor": "#98B503" }}>
    <h5 style={{ "cursor": "pointer", "textAlign": "center"
}}><b>Metodología</b></h5>
  </Accordion.Toggle>
  <Accordion.Collapse eventKey="1">
    <div className="projectBody">
      <h6>Objetivos</h6>
      <ul>
        <li>
          <button className="ButtonLikeLink"
onClick={this.showDiagram.bind(this, "O1")}>
            Caracterización del manejo y uso del agua en la unidad
            agrícola
          </button>
        </li>
        <li>
          <button className="ButtonLikeLink"
onClick={this.showDiagram.bind(this, "O2")}>
            Cuantificación de la HHverde, HHazul, HHgris

```

```

        </button>
    </li>
    <li>
        <button
            className="ButtonLikeLink"
            onClick={this.showDiagram.bind(this, "O3")}>
            Definir los efectos ambientales de la HHverde, HHazul,
            HHgris
        </button>
    </li>
</ul>
<center>
    <div id="output" />
    {
        this.state.obj === "O1" ?
        <div>
            <br />
            <hr />
            {
                this.state.showObj === "SS1O1" ? <img
width="800px" height="480px" src={Eimage} alt="" /> : true
            }
            {
                this.state.showObj === "SS2O1" ? <img
width="800px" height="480px" src={Fimage} alt="" /> : true
            }
            {
                this.state.showObj === "SS3O1" ? <img
width="800px" height="480px" src={Gimage} alt="" /> : true
            }
            <div style={{ "textAlign": "left" }}>
                <h6>Articulos Relacionados:</h6>
                <ul>
                    <li>
                        <Link to="/article/407" style={{ "color": "green"
}}>
                        Caudales promedio diarios en cultivos de
                        arroz
                    </Link>
                </li>
            </ul>
        </div>
    }
}

```

```

{
  this.state.obj === "O2" ?
    <div>
      <br />
      <hr />
      {
        this.state.showObj === "SS1O2" ||
this.state.showObj === "SS2O2" ? <img width="700px" height="480px" src={O2}
alt="" /> : true
      }
    <br />
    <div style={{ "textAlign": "left" }}>
      <h6>Articulos Relacionados:</h6>
      <ul>
        <li>
          <Link to="/article/406" style={{ "color": "green"
}}>
            Datos climáticos finca arrocera San José
            para el cálculo de las necesidades hídricas del cultivo
          </Link>
        </li>
        <li>
          <Link to="/article/408" style={{ "color": "green"
}}>
            Análisis de los parámetros del agua en
            finca San José
          </Link>
        </li>
        <li>
          <Link to="/article/409" style={{ "color": "green"
}}>
            Análisis de los parámetros del suelo en
            finca San José
          </Link>
        </li>
      </ul>
    </div>
  </div>
  : true
}
{
  this.state.obj === "O3" ?
    <div>
      <br />
      <hr />

```

```

{this.state.showObj === "SS1O3" ||
this.state.showObj === "SS2O3" || this.state.showObj === "SS3O3" ? <img
width="700px" height="480px" src={O3} alt="" /> : true}
<div style={{ "textAlign": "left" }}>
  <h6>Datos Relacionados:</h6>
  <ul>
    <li>
      <Link to="/article/417" style={{ "color": "green"
}}>
        Polígonos de Coberturas Vegetales, finca
        San José
      </Link>
    </li>
    <li>
      <Link to="/article/410" style={{ "color": "green"
}}>
        Imagen MODIS Evapotranspiración finca
        San José, Enero 2018
      </Link>
    </li>
    <li>
      <Link to="/article/411" style={{ "color": "green"
}}>
        Imagen MODIS Evapotranspiración finca
        San José, Febrero 2018
      </Link>
    </li>
    <li>
      <Link to="/article/412" style={{ "color": "green"
}}>
        Imagen MODIS Evapotranspiración finca
        San José, Septiembre 2017
      </Link>
    </li>
    <li>
      <Link to="/article/413" style={{ "color": "green"
}}>
        Imagen MODIS Evapotranspiración finca
        San José, Octubre 2017
      </Link>
    </li>
    <li>
      <Link to="/article/414" style={{ "color": "green"
}}>

```

San José, Noviembre 2017

Imagen MODIS Evapotranspiración finca

San José, Diciembre 2017

Imagen MODIS Evapotranspiración finca

San José, Periodo Septiembre 2017 - Febrero 2018

Imagen MODIS Evapotranspiración finca

Caracterización del manejo y uso del agua en la unidad agrícola

```

        </button>
    </li>
    <li>
        <button                                className="ButtonLikeLink"
onClick={this.showDiagram.bind(this, "R2")}>
            Cuantificacion de la HHverde, HHazul, HHgris
        </button>
    </li>
    <li>
        <button                                className="ButtonLikeLink"
onClick={this.showDiagram.bind(this, "R3")}>
            Definir los efectos ambientales de la HHverde, HHazul,
            HHgris
        </button>
    </li>
</ul>
<center>
    < div id="outputResults" />
    {
        this.state.obj === "R1" ?
        <div>
            <br />
            <hr />
            {
                this.state.showObj === "ST1R1" ? <img
width="800px" height="480px" src={R11} alt="" /> : true
            }
            {
                this.state.showObj === "ST2R1" ?
                <div>
                    <img          width="800px"          height="480px"
src={R12} alt="" />
                    <br />
                    <img          width="800px"          height="480px"
src={R122} alt="" />
                </div>
                : true
            }
        }
        {
            this.state.showObj === "ST3R1" ? <img
width="800px" height="480px" src={R13} alt="" /> : true
        }
    </div>
    : true
}

```

```

{
  this.state.obj === "R2" ?
    <div>
      <br />
      <hr />
      {
        this.state.showObj === "ST1R2" ? <img
width="800px" height="480px" src={R21} alt="" /> : true
      }
      {
        this.state.showObj === "ST2R2" ? <img
width="800px" height="480px" src={R22} alt="" /> : true
      }
      {
        this.state.showObj === "ST3R2" ? <img
width="800px" height="480px" src={R23} alt="" /> : true
      }
      {
        this.state.showObj === "ST4R2" ? <img
width="800px" height="480px" src={R24} alt="" /> : true
      }
    </div>
    : true
}
{
  this.state.obj === "R3" ?
    <div>
      <br />
      <hr />
      {
        this.state.showObj === "ST1R3" ?
          <div>
            <img width="800px" height="480px"
src={R31} alt="" />
            <br />
            <img width="800px" height="480px"
src={R312} alt="" />
          </div>
          : true
        }
      }
      {
        this.state.showObj === "ST2R3" ? <img
width="800px" height="480px" src={R32} alt="" /> : true
      }
    }
  }

```

```

        this.state.showObj === "ST3R3" ?
        <div>
            <img width="800px" height="480px"
src={R33} alt="" />
            <br />
            <img width="800px" height="480px"
src={R332} alt="" />
        </div>
        : true
    }
</div>
: true
}
</center>
</div>
</Accordion.Collapse>
</div>
<div>
    <Accordion.Toggle as={Card.Header} eventKey="3" style={{
"backgroundColor": "#E7B107" }}>
        <h5 style={{ "cursor": "pointer", "textAlign": "center"
}}><b>Conclusiones</b></h5>
    </Accordion.Toggle>
    <Accordion.Collapse eventKey="3">
        <center>
            <div className="projectBody">
                <img width="800px" height="380px" src={C1} alt="" />
                <br />
                <img width="800px" height="380px" src={C2} alt="" />
            </div>
        </center>
    </Accordion.Collapse>
</div>
</Accordion>
</div>
)
}
}

```

export default Project1;

5.1.2.1.13. Project2.jsx

```

import React from "react";
import { Accordion, Card } from "react-bootstrap";

```

```
import { Link } from "react-router-dom";
import metodo from "./metodo.png";
import R1 from "./R1.png";
import R12 from "./R1.2.png";
import R2 from "./R2.png";
import R22 from "./R2.2.png";
import R23 from "./R2.3.png";
import R3 from "./R3.png";
```

```
class Project2 extends React.Component {
```

```
  constructor(props) {
    super(props);
    this.state = {
      obj: "",
    }
  }
}
```

```
  diagramEvent(e) {
    this.setState({ obj: e });
  }
```

```
  render() {
    return (
      <div className="animated bounceIn fast">
        <center>
          <h5>
            ANÁLISIS DEL EFECTO DEL GROSOR DE LA LÁMINA DE AGUA
            EN EL CULTIVO DE ARROZ A PARTIR DEL ANÁLISIS DE RESPUESTAS
            ESPECTRALES.
          </h5>
          <b>
            Trabajo de Grado presentado como requisito para optar por el título
            de <br></br>
            Ingeniero Topográfico
          </b>
        </center>
        <b>Realizado por:</b> <br />
        Nelson Jair Centeno Moreno <br />
        <b>Director:</b> <br />
        MAURICIO EDILBERTO RINCÓN ROMERO <br />
        Ingeniero Catastral y Geodesta M.Sc. Ph.D. <br />
        <b>Tesis:</b> <br />
        <Link to="/article/422" style={{ "color": "green" }}>
```

```

        <b>CONSULTAR      TESIS</b>      <span      role="img"      aria-
label="."></span>
      </Link>
    <hr />
    <br />
    <Accordion>
      <div>
        <Accordion.Toggle      as={Card.Header}      eventKey="0"      style={{
"backgroundColor": "#65A357" }}>
          <h5      style={{      "cursor":      "pointer",      "textAlign":      "center"
}}><b>Resumen</b></h5>
        </Accordion.Toggle>
        <Accordion.Collapse eventKey="0">
          <div className="projectBody" style={{ "textAlign": "justify" }}>

```

La planta de arroz, al presentar una característica semiacuática le permite crecer en cultivos de inundación, pero manejar el cultivo en estas condiciones con alta concentración afecta negativamente el aspecto físico y económico. El presente trabajo muestra la aplicación de diversos análisis dentro de un marco metodológico para estudiar la relación existente entre las variables físicas y biológicas de las plantas de arroz con la lámina de agua de inundación que presentan las plantas durante el ciclo del cultivo. Se estudiaron cuatro variables de control: Altura, número de macollos, verdor y respuesta espectral, estas variables que permiten conocer la sanidad y estado fenológico de la planta y fueron medidas durante un ciclo de un cultivo de arroz en dos parcelas experimentales denominadas "Testigo" (con una altura media de lámina de agua de 10 cm) y "Réplica" (con una altura media de lámina de agua de 15 cm), pero de forma detallada, cada planta presentó una lámina de agua independiente, debido a la micro nivelación del terreno, mostrando que las plantas en sus variables físicas como la altura y el número de macollos se encuentran entre los valores estándar independientemente de la lámina de agua, y la relación entre estas variables y la lámina de agua, presentaron R^2 menores a 0,1 y 0,4 respectivamente. Utilizando un análisis de componentes principales y matriz de correlación entre las firmas espectrales, se encontró una ligera relación negativa entre la banda infrarroja y los índices de vegetación (Índice de vegetación de diferencia normalizado, NDVI; Índice de vegetación de diferencia normalizada verde, GNDVI y Diferencia normalizada de rojo, NDRE) indicando existe una leve afectación sobre estos indicadores. En contraste con la lámina de agua, esta sí afecta de manera negativa los parámetros biológicos de las plantas a medida que ella aumenta, pero la correlación entre las firmas espectrales de las plantas muestreadas y las firmas espectrales de referencia son cercanas a 0.98, mostrando que todas estas se encuentran sanas. Se logró identificar que existe un uso excesivo del agua de riego del 78% y 83% respectivamente, y que sus índices CWP (Productividad del agua del cultivo) son de 0,25 y 0,21 para Testigo y Réplica. Estos resultados permiten mostrar que existe

inefectividad en el uso del recurso hídrico y que el exceso de este no beneficia al cultivo.

```

    </div>
  </Accordion.Collapse>
</div>
<div>
  <Accordion.Toggle as={Card.Header} eventKey="1" style={{
"backgroundColor": "#98B503" }}>
    <h5 style={{ "cursor": "pointer", "textAlign": "center"
}}><b>Metodología</b></h5>
  </Accordion.Toggle>
  <Accordion.Collapse eventKey="1">
    <div className="projectBody">
      <img src={metodo} alt="" />
      <div style={{ "textAlign": "left" }}>
        <h6>Datos Relacionados:</h6>
        <ul>
          <li>
            <Link to="/article/407" style={{ "color": "green" }}>
              Caudales promedio diarios en cultivos de arroz, finca
San José
            </Link>
          </li>
          <li>
            <Link to="/article/418" style={{ "color": "green" }}>
              Datos espectrales por planta en Lote 8, finca San
José
            </Link>
          </li>
          <li>
            <Link to="/article/419" style={{ "color": "green" }}>
              Datos climáticos finca arrocería San José lote 8
            </Link>
          </li>
          <li>
            <Link to="/article/420" style={{ "color": "green" }}>
              Información fisiológica de plantas de arroz en finca
San José, lote 8
            </Link>
          </li>
        </ul>
      </div>
    </div>
  </div>
</Accordion.Collapse>
</div>

```

```

<div>
  <Accordion.Toggle as={Card.Header} eventKey="2" style={{
"backgroundColor": "#EBC038" }}>
    <h5 style={{ "cursor": "pointer", "textAlign": "center"
}}><b>Resultados</b></h5>
  </Accordion.Toggle>
  <Accordion.Collapse eventKey="2">
    <div className="projectBody">
      <h6>Resultados Por objetivo</h6>
      <ul>
        <li>
          <button className="ButtonLikeLink"
onClick={this.diagramEvent.bind(this, "R1")}>
            Sanidad Física
          </button>
        </li>
        <li>
          <button className="ButtonLikeLink"
onClick={this.diagramEvent.bind(this, "R2")}>
            Sanidad Espectral
          </button>
        </li>
        <li>
          <button className="ButtonLikeLink"
onClick={this.diagramEvent.bind(this, "R3")}>
            Análisis CWP
          </button>
        </li>
      </ul>
      {
        (this.state.obj === "R1") ? (
          <div>
            <p>
              La altura promedio final para las plantas de arroz
              presentes en la parcela Testigo fue de 101,9 cm con una desviación del 6,4% y una
              altura final para las plantas en la parcela Réplica de 103 cm con una desviación del
              6,6%. En el caso de este estudio las plantas de las parcelas Testigo y Réplica en
              su característica de altura presentan valores acordes a una planta de arroz sana, a
              pesar de estar en láminas de agua diferentes.
            </p>
            <center><img src={R1} alt="" width="400px"
height="160px" /></center>
            <br />
            <p>

```

Se realizaron tres modelos de regresión con los datos de altura de la planta como variable dependiente y los días del cultivo como variable independiente y se encontró que el patrón de crecimiento presente, tanto para la parcela Testigo como para la parcela Réplica, estos se ajustan de mejor manera con el modelo de regresión polinómica, donde el coeficiente de determinación (R^2) es del 0,99 en ambos casos, mostrando un crecimiento acorde con las plantas de semillas que actualmente son utilizadas en el cultivo, su crecimiento presenta un comportamiento sigmoideal.

```

    </p>
    <center><img src={R12} alt="" width="600px"
height="400px" /></center>
    <br />
</div>

```

```

    ) : true
  }
  {
    (this.state.obj === "R2") ? (
      <div>
        <p>

```

Para determinar el efecto del grosor de la lámina de agua en las respuestas espectrales de las plantas se aplicó un ajuste de modelos de regresión entre las bandas espectrales y el nivel de lámina de agua, encontrando que el ajuste con mayor coeficiente de R^2 fue 0.32 en la banda infrarroja.

```

    </p>
    <br />
    <center><img src={R2} alt="" width="400px"
height="200px" /></center>
    <br />
    <p>

```

Con el valor de las bandas espectrales en las fases estudio en ambas parcelas, se procedió a calcular el promedio de cada uno de los índices en cada fase para las dos parcelas con sus respectivas desviaciones. Para el caso del NDVI fue el índice que tuvo los valores de desviación más bajos, y para este índice los valores de referencia de un arroz sano están alrededor de 0,63 y 0,99 (García, 2010 y Saavedra et al., 2018). En el caso de los índices GNDVI y NDRE presentaron desviaciones ligeramente más grandes que el NDVI, pero igualmente son representativas para las plantas en ambas parcelas, donde para el índice GNDVI las plantas sanas suelen rondar valores de 0,6 (Candiago et al., 2015) y para el índice NDRE valores de 0,5, evidenciando un aspecto positivo en los valores nutricionales de la planta (Zhang et al., 2019).

```

    </p>
    <br />
    <center><img src={R22} alt="" width="450px"
height="380px" /></center>

```


<p>

Usando firmas espectrales de plantas de arroz sanas presentadas por autores de investigaciones relacionadas, se calculó la correlación de las firmas espectrales de referencia y las firmas espectrales observadas, se obtuvo que el comportamiento de las firmas para las parcelas experimentales las fases de muestreo es compatible en mayor medida con la firma espectral correspondiente a una planta de arroz sana con correlaciones cercanas a 0,98.

</p>

<center></center>

<p>

Teniendo en cuenta los resultados de correlación anterior, se observa que tanto las correlaciones de las firmas de referencia para las plantas de arroz sanas como las firmas de referencia para las plantas de arroz enferma y ligeramente sana, obtuvieron valores altos.

</p>

</div>

) : true

}

{

(this.state.obj === "R3") ? (

<div>

<p>

Se calculó el índice CWP para medir la eficiencia del uso del recurso hídrico en la productividad del cultivo. Para esto se usaron los valores de rendimiento para la parcela Testigo que fue de 59,09 kg con un volumen de agua azul de 187,85 m³ y un volumen de agua verde 43,67 m³ para un uso total de 231,54 m³, para la parcela Réplica el rendimiento fue de 39,53 kg con un volumen de agua azul de 160,25 m³ y un volumen de agua verde 28,03 m³ para un uso total de 188,28 m³.

El índice CWP para la parcela Testigo fue de 0,255, siendo este bajo para los valores estándar de los cultivos de arroz (González y Alonso, 2016), pero el rendimiento presenta un indicador de 5,5 Ton/Ha que es un valor aceptable para los cultivos de arroz según los valores estudiados por González y Alonso (2016), mostrando así que el problema que presenta el índice CWP tiene mayor peso en el uso ineficiente del recurso hídrico más que en el rendimiento que la parcela presentó.

Para la parcela Réplica el índice CWP fue similar a la otra parcela, pues presentó un valor de 0,21, que como fue explicado anteriormente es bajo, pero de igual forma su rendimiento tiene un indicador de 5,7 Ton/Ha que es admisible según lo estudiado por González y Alonso (2016), y contrastando con el resultado anterior, se evidencia el problema sobre el uso del recurso hídrico

excesivo y que este no está influyendo de manera óptima para el mejoramiento de la productividad del cultivo.

```

        </p>
        <br />
        <center><img src={R3} alt="" width="400px"
height="160px" /></center>
    </div>
    ) : true
    }
</div>
</Accordion.Collapse>
</div>
<div>
    <Accordion.Toggle as={Card.Header} eventKey="3" style={{
"backgroundColor": "#E7B107" }}>
        <h5 style={{ "cursor": "pointer", "textAlign": "center"
}}><b>Conclusiones</b></h5>
    </Accordion.Toggle>
    <Accordion.Collapse eventKey="3">
        <div className="projectBody">

```

A partir del estudio realizado, examinando la información obtenida por medio del muestreo de las plantas de arroz de las parcelas experimentales con diferentes láminas de agua, fue posible evidenciar el bajo efecto del grosor de la lámina de agua en las características físicas, biológicas y productivas de las plantas a través del ciclo del cultivo. Esto se puede evidenciar a través de los tratamientos realizados a los datos, pues los modelos de regresión lineal estimados no superan un valor de 0,30 de R² en ninguno de los análisis para las diferentes combinaciones realizadas.

La metodología empleada consideró una muestra de nivel de lámina de agua entre 5 y 20 cm lo cual lo ubica en un experimento de entorno controlado con resultados esperados, sin embargo, resulta necesario considerar escenarios de respuesta crítica para la variable estudiada, como lo son valores de lámina de agua entre saturada (0 cm con suelo humedecido) y mayor a 20 cm, con el fin de detectar casos de estrés hídrico en la planta y su incidencia directa en las características físicas y biológicas.

Con respecto al uso eficiente del recurso hídrico en las parcelas experimentales se encontró que existe un desaprovechamiento de más del 80% del agua de riego que se proporciona al cultivo, pues no se toma en cuenta que la precipitación suple gran parte de la necesidad hídrica del cultivo (Santana, 2008) y al realizar el cálculo del índice CWP se observa que los valores obtenidos para ambas parcelas es de 0,2 siendo un valor bajo, pero el rendimiento del cultivo sí se encuentra entre los valores aceptables, dejando así solo el problema sobre el uso excesivo del recurso hídrico.

```

    </div>
</Accordion.Collapse>

```

```

        </div>
      </Accordion>
    </div>
  )
}
}

```

```
export default Project2;
```

5.1.2.1.14. Project3.jsx

```
import React from "react";
```

```
import { Accordion, Card } from "react-bootstrap";
import { Link } from "react-router-dom";
```

```
import m_1 from "./p3_m_1.png";
import m_2 from "./p3_m_2.png";
```

```
class Project3 extends React.Component {
```

```
  constructor(props) {
    super(props);
    this.state = {
      obj: "",
    }
  }

```

```
  diagramEvent(e) {
    this.setState({ obj: e });
  }

```

```
  render() {
    return (
      <div className="animated bounceIn fast">
        <center>
          <h5>
            ESTIMACIÓN DE LAS CONCENTRACIONES DE NITRÓGENO EN
            EL CULTIVO DE ARROZ A PARTIR DE DATOS HIPERESPECTRALES
          </h5>
          <b>
            Trabajo de Grado presentado como requisito para optar por el título
            de <br></br>
            Ingeniero Topográfico
          </b>

```

```

</center>
<b>Realizado por:</b> <br />
Duvan Felipe Quintero Perea <br />
<b>Director:</b> <br />
    MAURICIO EDILBERTO RINCÓN ROMERO <br />
    Ingeniero Catastral y Geodesta M.Sc. Ph.D. <br />
    <Link to="/article/423" style={{ "color": "green" }}>
        <b>CONSULTAR TESIS</b> <span role="img" aria-
label=". ">📄 </span>
    </Link>
    <hr />
    <br />
    <Accordion>
        <div>
            <Accordion.Toggle as={Card.Header} eventKey="0" style={{
"backgroundColor": "#65A357" }}>
                <h5 style={{ "cursor": "pointer", "textAlign": "center"
}}><b>Resumen</b></h5>
            </Accordion.Toggle>
            <Accordion.Collapse eventKey="0">
                <div className="projectBody" style={{ "textAlign": "justify" }}>
                    El rendimiento de un cultivo de arroz depende de la
disponibilidad y asimilación de
nutrientes, el tiempo de exposición de la planta a la radiación
solar, disponibilidad del
recurso hídrico y el manejo adecuado que se le dé a la
interacción del sistema
agrícola. La geomática en la gestión agrícola y en particular en
la agricultura de
precisión juega un papel muy importante, dado que permite
relacionar espacialmente
los individuos, procesos y variables que intervienen en un
evento, y de esta manera
atender puntualmente las necesidades de cada planta, y a partir
de ellos entender
mejor el proceso de producción, sus relaciones, efectos e
impactos para el apoyo a la
toma de decisiones. En la presente investigación, se realizó el
seguimiento de un
conjunto de plantas de arroz en todo el ciclo de crecimiento. Se
adoptaron cuatro
experimentos usando diferentes planes de manejo en la
aplicación del nitrógeno; una

```

con los niveles estándares usados en la producción normal (200 kg/ha) y las otras tres con -50%, +100% y +200%, respectivamente. Las variables de respuesta se capturaron a través de lecturas de fenotipo, espectroradiometría y análisis químico del contenido de N en muestras foliares en cuatro momentos del desarrollo del cultivo. Se estimaron las longitudes de onda más sensibles a la variabilidad del nitrógeno en el arroz, para así desarrollar un índice de vegetación (IDN) que permite estimar las condiciones de contenido de nitrógeno en el follaje del arroz para la variedad Fedearroz 60. Finalmente, se establecieron los cuatro modelos de estimación basados en el índice IDN con una muestra de 20 datos que presentaron valores de precisión de estimación prometedores (34 % R^2 87 %) y, además, se encontró concordancia frente a la respuesta de las variables evaluadas con la literatura, por lo cual, se recomienda la aplicación de esta metodología en el manejo del estado nutricional del cultivo de arroz.

</div>

</Accordion.Collapse>

</div>

<div>

<Accordion.Toggle as={Card.Header} eventKey="1" style={{ "backgroundColor": "#98B503" }}>

<h5 style={{ "cursor": "pointer", "text-align": "center" }}>Metodología</h5>

</Accordion.Toggle>

<Accordion.Collapse eventKey="1">

<div className="projectBody">

Como estrategia global de la investigación se propone el desarrollo de un experimento

que evalúa el efecto de cuatro dosis diferentes de nitrógeno en el cultivo de arroz con

el objetivo de estimar los niveles de tal concentración a partir de datos

hiperespectrales. Se formula la estructura metodología basado en la experiencia de

los estudios de Peña, Cruzan y Martineza (2010) y Stroppiana et al. (2006), además,

se incorporan las recomendaciones de uso de cada fabricante de los equipos utilizados PhilRice, (2007); Ocean Optics, (2009) juntos a Primera et al, (2005) en su guía de adquisición de datos espectrales. De este modo, como diseño experimental se propuso el desarrollo de cuatro experimentos con diferentes planes de manejo en la aplicación de N; las dosis aplicadas fueron basadas en un nivel estándar recomendado para las condiciones del área de estudio (N= 200 kg/ha) y las otras tres con la variación del nitrógeno en -50%, +100% y +200%, respectivamente. Se capturaron tres tipos de variables de respuestas que se agrupan en datos de fenotipo, espectrometría y análisis químico del contenido de N foliar durante cuatro momentos del desarrollo del cultivo. De ahí que, el efecto de cada tratamiento fue medido por medio de la cantidad de N foliar absorbido, en contraste con lo que reflejo las características fisiológicas y la respuesta espectral para identificar tendencias físicas en el desarrollo y el aprovechamiento de los insumos en las plantas inducido por la disponibilidad de nutrientes, para desarrollar un índice de vegetación que permite estimar las condiciones de contenido de N en el follaje del arroz.

<center>

</center>


```
<div style={{ "textAlign": "left" }}>
  <h6>Datos Relacionados:</h6>
  <ul>
    <li><Link to='/article/90' style={{ 'color': 'green'
}}>Fisiología de plantas de arroz en finca San José Fase 1</Link></li>
    <li><Link to='/article/91' style={{ 'color': 'green'
}}>Fisiología de plantas de arroz en finca San José Fase 2</Link></li>
    <li><Link to='/article/92' style={{ 'color': 'green'
}}>Fisiología de plantas de arroz en finca San José Fase 3</Link></li>
    <li><Link to='/article/93' style={{ 'color': 'green'
}}>Fisiología de plantas de arroz en finca San José Fase 4</Link></li>

    <li><Link to='/article/37' style={{ 'color': 'green' }}>Lectura
de espectrometría en cultivo de arroz, finca San José tratamiento N1, fase
1</Link></li>
    <li><Link to='/article/38' style={{ 'color': 'green' }}>Lectura
de espectrometría en cultivo de arroz, finca San José tratamiento N2, fase
1</Link></li>
    <li><Link to='/article/39' style={{ 'color': 'green' }}>Lectura
de espectrometría en cultivo de arroz, finca San José tratamiento N3, fase
1</Link></li>
    <li><Link to='/article/40' style={{ 'color': 'green' }}>Lectura
de espectrometría en cultivo de arroz, finca San José tratamiento T, fase
1</Link></li>
    <li><Link to='/article/41' style={{ 'color': 'green' }}>Lectura
de espectrometría en cultivo de arroz, finca San José tratamiento K1, fase
2</Link></li>
    <li><Link to='/article/42' style={{ 'color': 'green' }}>Lectura
de espectrometría en cultivo de arroz, finca San José tratamiento K2, fase
2</Link></li>
    <li><Link to='/article/43' style={{ 'color': 'green' }}>Lectura
de espectrometría en cultivo de arroz, finca San José tratamiento K3, fase
2</Link></li>
    <li><Link to='/article/44' style={{ 'color': 'green' }}>Lectura
de espectrometría en cultivo de arroz, finca San José tratamiento N1-R, fase
2</Link></li>
    <li><Link to='/article/45' style={{ 'color': 'green' }}>Lectura
de espectrometría en cultivo de arroz, finca San José tratamiento N1, fase
2</Link></li>
```

<Link to='/article/46' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N2-R, fase 2</Link>

<Link to='/article/47' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N2, fase 2</Link>

<Link to='/article/48' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N3-R, fase 2</Link>

<Link to='/article/49' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N3, fase 2</Link>

<Link to='/article/50' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento P1, fase 2</Link>

<Link to='/article/51' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento P2, fase 2</Link>

<Link to='/article/52' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento P3, fase 2</Link>

<Link to='/article/53' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento T-R, fase 2</Link>

<Link to='/article/54' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento T, fase 2</Link>

<Link to='/article/55' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento K1, fase 3</Link>

<Link to='/article/56' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento K2, fase 3</Link>

<Link to='/article/57' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento K3, fase 3</Link>

<Link to='/article/58' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N1-R, fase 3</Link>

<Link to='/article/59' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N1, fase 3</Link>

<Link to='/article/60' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N2-R, fase 3</Link>

<Link to='/article/61' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N2, fase 3</Link>

<Link to='/article/62' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N3-R, fase 3</Link>

<Link to='/article/63' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N3, fase 3</Link>

<Link to='/article/64' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento P1, fase 3</Link>

<Link to='/article/65' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento T-R, fase 3</Link>

<Link to='/article/66' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento T, fase 3</Link>

<Link to='/article/67' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento K1, fase 4</Link>

<Link to='/article/68' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento K2, fase 4</Link>

<Link to='/article/69' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento K3, fase 4</Link>

<Link to='/article/70' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N1, fase 4</Link>

<Link to='/article/71' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N2, fase 4</Link>

<Link to='/article/72' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N3, fase 4</Link>

<Link to='/article/73' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento P1, fase 4</Link>

<Link to='/article/74' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento P2, fase 4</Link>

<Link to='/article/75' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento T, fase 4</Link>

- <Link to='/article/76' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento K1, fase 4</Link>

- <Link to='/article/77' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento K2, fase 4</Link>

- <Link to='/article/78' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento K3, fase 4</Link>

- <Link to='/article/79' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N1, fase 4</Link>

- <Link to='/article/80' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N2, fase 4</Link>

- <Link to='/article/81' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N3, fase 4</Link>

- <Link to='/article/82' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento N3_R, fase 4</Link>

- <Link to='/article/83' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento P1, fase 4</Link>

- <Link to='/article/84' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento P2, fase 4</Link>

- <Link to='/article/85' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento P3, fase 4</Link>

- <Link to='/article/86' style={{ 'color': 'green' }}>Lectura de espectrometría en cultivo de arroz, finca San José tratamiento T, fase 4</Link>

</div>

</div>

</Accordion.Collapse>

</div>

<div>

<Accordion.Toggle as={Card.Header} eventKey="3" style={{ "backgroundColor": "#E7B107" }}>

<h5 style={{ "cursor": "pointer", "textAlign": "center" }}>Conclusiones</h5>

</Accordion.Toggle>
<Accordion.Collapse eventKey="3">
<div className="projectBody">

Para la variedad de arroz Fedearroz 60 se analizaron cuatro fases fenológicas en un experimento en campo realizado en la finca San José de la Arrocería La Esmeralda S.A.S, ubicada en el Departamento del Cauca, municipio de Santander de Quilichao (zona del suroccidente colombiano)- En este estudio se encontró que cada etapa fenológica presenta una respuesta diferente para las variables de estudio, es decir que no hay una tendencia que trascienda durante todo el ciclo del cultivo aunque en las fases que se compararon contra los demás autores las respuestas encontradas fueron coherentes. Por otro lado, en las variables macollos y altura se presentaron casos en los que iguales niveles de fertilización se comportaron como diferentes poblaciones, mientras que diferentes niveles de fertilización demostraron ser de la misma población debido a las alteraciones de factores que no se pudieron controlar como las condiciones de aplicación homogénea, absorción homogénea por tener un riego inundado, volatilización de la urea o la deficiencia de otros nutrientes.

Se encontró también que, si se conserva la aplicación de N del tratamiento Testigo (200 kg/ha) la variedad Fedearroz 60 puede alcanzar una cantidad máxima de macollos que oscila entre 15 unidades con algunas alteraciones de plantas que alcanzan las 22 unidades, la altura promedio para la fase final estará entre 94 cm a 106 cm, sin embargo, acerca del verdor no se alcanzaron intensidades altas como se esperaban pero se evidenció la respuesta al déficit de N con valores críticos, no obstante, los tratamientos de mayor concentración de N no desarrollaron intensidades

mayores al umbral de 3,5.

Se determinaron las firmas espectrales para cada una de las fases estudiadas y se comprobó que cada parcela experimental se comporta como un tratamiento distinto dentro de su fase, esto mediante el análisis de varianza prueba H de Kruskal – Wallis, y, además, se pudo reconocer las zonas del espectro donde se presentan los cambios asociado a las concentraciones de N: la absorbancia en las longitudes de ondas del violeta(400 – 445 nm), del azul B(400 – 510 nm) con registros menores del 12 % y en la banda del rojo R(640 – 670 nm); en la región del verde G(530 – 590 nm), marcada por el primer pico de reflectancia, se crea una separación entre las firmas de los diferentes tratamiento que finaliza con tres comportamientos donde los registros de mayor y menor magnitud corresponden a N3R y N1 respectivamente, sin superar el 30 % de reflectancia; por último, el borde rojo NIR(670 – 750 nm) que se encuentra entre los valores entre 62 y 70 %. En cuanto a la variabilidad de los datos, la primera fase presentó dispersiones altas para todos los tratamientos debido al ruido generado por el tamaño de las hojas y las condiciones que propiciaba la lámina de agua junto con la condición climática, sin embargo, en el resto de las fases se obtuvo un balance de variabilidad bajo, a lo cual no se encontró relación con la magnitud de la reflectancia ni con la hora de la toma de datos, por lo cual, se afirma que la metodología planteada para los datos espectrales fue la adecuada.

El índice IDN general subestima los valores elevados de N, mientras que sobre estima los valores pequeños de N, por lo cual se planteó realizar el cálculo de un índice IDN sectorizado por fase fenológica, a los cuales se les desarrolló los modelos de

regresión que permitieron identificar que se mantuvo la correlación entre las concentraciones de N y la longitud de onda localizada en la región del infrarrojo cercano NIR(670 – 750 nm), mientras que aquella ubicada en el pico del verde por el IDN general se desplazó hacia las regiones en los que la clorofila genera la mayor absorbanza como lo fueron la banda del violeta (400 – 430 nm) y del rojo R(640 – 670 nm). Los datos usados en el modelo fueron tomados sobre parcelas a las que se les brindó las mejores condiciones de producción; es decir que a pesar de que se tuvo un experimento de dosis baja, no se tuvieron datos de plantas o parcelas con malas condiciones nutricionales. Esto implica que sería conveniente poder complementar el experimento con observaciones tomadas en parcelas deficitarias, y de esta forma se podría mejorar el espectro de representatividad de la información para afinar el modelo.

Para uno de los modelos sectorizados (Fase 1), se encontró una tendencia negativa para la correlación de modo que a medida que aumenta el valor del índice IDN F1, el valor de N foliar disminuye, mientras que en las demás fases se estableció una tendencia positiva. Esto se explica porque las regiones a las que pertenecen las longitudes de onda que se asociaron con el N foliar para IDN F1 son $p_1 = 723,88$ nm y $p_2 = 407,97$ nm, es decir que el comportamiento de una planta sana siempre estará regido por $p_1 > p_2$, lo que indica que para esta fase a medida que la diferencia de reflectancia de estas dos bandas aumentaba el valor de N foliar de la planta disminuía.

En cuanto a los modelos de estimación se encontró que aquellos generados con 4 muestras (Fase 1 y Fase 4) arrojaron pendientes elevadas y valores de ajustes altos con $R^2 = 87 \%$ y $R^2 = 83 \%$ respectivamente, mientras que aquellos con un número de muestras mayor a 12 (Fase 3 y el general) reducen el valor de la pendiente debido a que el modelo se entrena con una mayor varianza y, por lo tanto, el valor R^2 disminuye a valores de $R^2 = 34 \%$ y $R^2 = 35 \%$ respectivamente. De esto, se puede concluir que las condiciones del cultivo Fedearroz 60 requieren pendientes altas en el modelo para una capacidad de estimación alta y que es indispensable un grupo de muestras más grande para que el modelo sea confiable, además, es una excelente consideración el generar un modelo de estimación para cada fase ya que la pendiente de cada uno se ajustaría al momento de desarrollo del cultivo. Los resultados que se obtuvieron de este estudio son coherentes por lo que describen otros autores con investigaciones similares, a lo cual se considera como una buena herramienta para estimar las concentraciones de N foliar a partir de una metodología basada en datos hiperespectrales que resulta ser no destructiva y con resultados en un tiempo oportuno para la toma de decisiones frente a las necesidades de fertilizantes. Sin embargo, pese a que el modelo de IDN general reporta un valor similar al de la literatura es conveniente desarrollar un modelo específico para cada fase fenológica para que la pendiente y el intercepto se ajusten al desarrollo de ese momento del cultivo y, además, se debe de actualizar estos modelos con nuevos conjuntos de datos que entre 12 a 20 muestras para entrenar el modelo por etapa y así que los modelos tengan validez en las estimaciones del estado nutricional del cultivo de arroz.

```

        </div>
      </Accordion.Collapse>
    </div>
  </Accordion>
</div>
)
}
}

```

export default Project3;

5.1.2.1.15. Project4.jsx

import React from "react";

import { Accordion, Card } from "react-bootstrap";

import { Link } from "react-router-dom";

import arch from "./p4_arch.png";

class Project4 extends React.Component {

```

  constructor(props) {
    super(props);
    this.state = {
      obj: "",
    }
  }

```

```

  diagramEvent(e) {
    this.setState({ obj: e });
  }

```

```

  render() {
    return (
      <div className="animated bounceIn fast">
        <center>
          <h5>
            INTEGRACIÓN MULTIESCALA ENTRE MODELOS CLIMÁTICOS
            GLOBALES Y REGIONALES MEDIANTE TÉCNICAS DE INTELIGENCIA
            ARTIFICIAL
          </h5>
          <b>

```

Trabajo de Grado presentado como requisito para optar por el título
de Magister en Ingeniería de Sistemas

 </center>
 Realizado por:

 Wilmer Muñoz Herrera

 Director:

 MAURICIO EDILBERTO RINCÓN ROMERO

 Ingeniero Catastral y Geodesta M.Sc. Ph.D.

 Codirector:

 Oscar Fernando Bedoya, PhD.

 <Link to="/article/424" style={{ "color": "green" }}>
 CONSULTAR TESIS
 </Link>
 <hr />

 <Accordion>
 <div>
 <Accordion.Toggle as={Card.Header} eventKey="0" style={{
 "backgroundColor": "#65A357" }}>
 <h5 style={{ "cursor": "pointer", "textAlign": "center"
 }}>Resumen</h5>
 </Accordion.Toggle>
 <Accordion.Collapse eventKey="0">
 <div className="projectBody" style={{ "textAlign": "justify" }}>
 El cambio climático es un problema que afecta gran parte de las
 actividades humanas, en
 especial la agricultura. En esta investigación se realizó la
 exploración del comportamiento
 climático en la región del Valle del Cauca para futuros
 escenarios, con el uso de Modelos
 Climáticos Globales (GCM) que pronostican diferentes
 variables climáticas alrededor de
 los futuros 100 años, con resoluciones espaciales de 100 Km x
 100 Km
 aproximadamente. En particular, se integraron las
 metodologías de modelación climática
 regional para el mejoramiento de resolución espacial
 (Downscaling) de los GCM a
 resoluciones espaciales de 5.6 Km x 5.6 Km aproximadamente,
 con la ayuda de técnicas
 de inteligencia artificial para la modelación, información de
 datos satelitales y estaciones
 locales que permite generar información con resolución
 espacial útil para la toma de

decisiones frente a las futuras condiciones climáticas que se pueden presentar, en especial las variables climáticas de precipitación y temperatura. Como técnicas de inteligencia artificial se exploraron, redes neuronales, máquinas de soporte vectorial, Random Forest y Redes Bayesianas, identificando que, para la región del Valle del Cauca con sus condiciones ambientales, las técnicas de Random Forest, máquinas de soporte vectorial con kernel radial y redes neuronales multicapa con el algoritmo backpropagation y función de activación sigmoid de dos y una capa oculta presentan mejores resultados de predicción.

```

</div>
</Accordion.Collapse>
</div>
<div>
  <Accordion.Toggle as={Card.Header} eventKey="1" style={{
"backgroundColor": "#98B503" }}>
    <h5 style={{ "cursor": "pointer", "textAlign": "center"
}}><b>Metodología</b></h5>
  </Accordion.Toggle>
  <Accordion.Collapse eventKey="1">
    <div className="projectBody">
      <b> ZONA DE ESTUDIO </b>
      <br />
      <br />

```

La zona de estudio se encuentra ubicada en la región pacífica y andina de Colombia, el cual corresponde al departamento del Valle del Cauca, limitando con los departamentos de Chocó, Risaralda, Quindío, Tolima, Cauca y el océano pacífico. La precipitación

promedio anual en la parte media del Valle del Río Cauca, desde Puerto Tejada hasta Cartago no supera los 1500 mm/año, con excepción de Cali, Yumbo, Cerrito y Candelaria donde oscila alrededor de 1000 mm/año, en la parte alta de la vertientes la precipitación tiene valores cercanos a los 2000 mm/año, en la parte baja oscila los 4000 mm/año en las

estribaciones de las cordilleras, 1500 mm/año en la desembocadura del río Cauca y en la parte pacífica puede alcanzar los 6000 mm/año.

ARQUITECTURA

<center> </center>

<div style={{ "textAlign": "left" }}>

<h6>Datos Relacionados:</h6>

</div>

</div>

</Accordion.Collapse>

</div>

<div>

<Accordion.Toggle as={Card.Header} eventKey="2" style={{ "backgroundColor": "#EBC038" }}>

<h5 style={{ "cursor": "pointer", "textAlign": "center" }}>Resultados</h5>

</Accordion.Toggle>

<Accordion.Collapse eventKey="2">

<div className="projectBody">

PROTOTIPO WEB

Los resultados obtenidos del proceso de downscaling de las variables de precipitación y

temperatura con resolución espacial de 5.6 Km², mediante el uso de la técnica de

Random Forest están almacenados en el motor de base de datos PostgreSQL. El acceso

y consulta de esta información se realiza a través del lenguaje de programación PHP. El

prototipo web puede ser consultado a través del enlace http://45.5.164.43/downscaling_vc/.

API

Este sistema web tiene la posibilidad de prestar los servicios de la modelación de las

variables de precipitación y temperatura en los cuatro posibles escenarios para la región

del Valle del Cauca, mediante el API. El cual se encuentra documentada en la página web

http://45.5.164.43/downscaling_vc/#api del sistema, todos los procedimientos,

estructura de consulta y respuesta de los datos de modelación a través del método de

aprendizaje de Random Forest.

</div>

</Accordion.Collapse>

</div>

<div>

<Accordion.Toggle as={Card.Header} eventKey="3" style={{ "backgroundColor": "#E7B107" }}>

<h5 style={{ "cursor": "pointer", "text-align": "center" }}>Conclusiones</h5>

</Accordion.Toggle>

<Accordion.Collapse eventKey="3">

<div className="projectBody">

Los resultados obtenidos en esta investigación alcanzaron los objetivos propuestos y los

resultados esperados, de esta manera se presenta en los siguientes ítems las

conclusiones por cada uno de los objetivos propuestos.

La información satelital de temperatura de MODIS y datos globales de precipitación

CHIRPS permiten reconstruir información espacial y series de tiempo no existentes por

parte de las estaciones meteorológicas locales. En esta investigación se utilizaron redes neuronales como método para la elaboración de esta información, obteniendo niveles de correlación aceptables de alrededor de 0.75 para la precipitación y 0.87 para la temperatura. Siendo esta información indispensable para realizar proyecciones climáticas que abarquen toda la distribución espacial de la región de estudio.

Los valores bajos de exactitud obtenidos para las variables de precipitación y temperatura en la región del pacífico del Valle del Cauca se pueden presentar debido a la falta o insuficiente de información meteorológica de las estaciones locales. Por ejemplo, en la precipitación la diferencia entre los valores reales y estimados están alrededor de los 100 mm/mes y 150 mm/mes en la zona pacífica y alrededor de 20 mm/mes y 40 mm/mes en las otras regiones, de igual manera la temperatura alrededor de los 2 °C y 3 °C de diferencia en las zonas del pacífico y montañosa de la región, mientras que en las otras regiones son menores a 1 °C.

La presencia de múltiples modelos climáticos globales por diferentes instituciones, versiones y posibles escenarios hace que la investigación en estos temas en particular sea muy amplia. De esta manera se considera el uso del modelo climático global para esta investigación sea el modelo HadGEM2-ES del proyecto CMIP5, el cual ha obtenido mejores rendimientos en las regiones tropicales y suramericanas en comparación con otros modelos, por esta razón la región del Valle del Cauca no es ajena a los estudios que evaluaron el modelo HadGEM2-ES.

Entre las técnicas de inteligencia artificial utilizadas, Random Forest permite estimar de

forma más acertada la precipitación y temperatura. En particular, esta técnica alcanza un nivel de correlación de 0.81 para la precipitación y 0.93 para la temperatura, en comparación con las otras técnicas. No obstante, las técnicas de máquinas de soporte vectorial con kernel radial y redes neuronales con dos capas ocultas de cinco y dos neuronas pueden presentar resultados aceptables, con alrededor de 0.75 en precipitación y 0.90 en temperatura. Mientras que, los kernel lineales, polinomial y sigmoide de máquinas de soporte vectorial y las redes bayesianas presentan un bajo nivel de correlación.

El prototipo web está disponible al público para el uso de cualquier usuario que desea conocer cómo será el comportamiento de las variables climáticas de precipitación y temperatura en el futuro. Para aspectos de mitigación, evaluación de impactos ambientales, agricultura, entre otros ámbitos. Además, el uso de estos resultados para otras aplicaciones o análisis específicos mediante la disponibilidad de API. Como también,

el uso de esta herramienta puede realizarse a través de plataformas móviles, sin embargo, para obtener una buena experiencia en el uso es recomendable realizarlo a través de computador de escritorio o Tablet.

```

    </div>
  </Accordion.Collapse>
</div>
</Accordion>
</div>
)
}
}

```

```
export default Project4;
```

5.1.2.2. Vistas

5.1.2.2.1. Acerca.jsx

```
import React from "react";
import {
  Container
} from 'reactstrap';

import logotipos from "assets/logotipos.png";
import { NavLink } from "react-router-dom";

class Acerca extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
    }
  }

  render() {
    return (
      <div style={{ "text-align": "justify" }}>
        <Container>
          <h3>Sistema de Información del Laboratorio de Agricultura Inteligente
– SILAIN</h3>
          En el marco del Proyecto del Sistema General de Regalías 2014-
2020 se consolidó el Centro de
          Investigación e Innovación en Bioinformática y Fotónica (CiBioFi).
          Dentro de CiBioFi se construyó
          un Subproyecto Agricultura Inteligente (SAI) para el cual y con el
          apoyo del Grupo de Investigación
          en Simulación y Modelación Dinámica Espacial (GISMODEL) se
          desarrollaron cinco proyectos en
          diversos temas de Agricultura Inteligente.
          <br /><br />
          Estos proyectos generaron un gran volumen de información
          agronómica y ambiental mediante la
          utilización de variadas técnicas como la fotografía multiespectral,
          espectrometría, muestreo de
          suelos y vegetación, levantamientos posicionales de campo y
          estaciones de monitoreo ambiental.
          <br /><br />
          Ante la necesidad de una herramienta para facilitar la consulta y
          acceso a los datos e información
```

del SAI se construyó el Sistema de Información del Laboratorio de Agricultura Inteligente (SILAIN), siendo este una plataforma web desarrollada con software libre que cuenta con diferentes utilidades para consultar, localizar, explorar y acceder a información, datos, publicaciones, metodologías, autores, técnicas, resultados, geoportales y demás productos del SAI. SILAIN se apoya en los principios de datos FAIR (Findable: Localizable, Accessible: Accesible, Interoperable: Interoperable, Reusable: Reutilizable), los cuales son un conjunto de cualidades con las que deben contar los datos científicos para garantizar su validez, su reproductibilidad y para conducir a nuevos descubrimientos.

De este modo, SILAIN busca convertirse en una memoria de investigación y a la vez en un medio de divulgación de los datos, haciendo posible la reutilización de la información, procesos y productos que se generan en el Centro, dando visibilidad a los autores para potenciar la gestión científica del Centro en la comunidad Vallecaucana.

SILAIN hizo parte del trabajo de grado del estudiante de ingeniería topográfica Alexander Clavijo Ledesma del Grupo de Investigación GISMODEL. El desarrollo de SILAIN inició en marzo de 2020 y finalizó en diciembre de este mismo año.

```

<br /><br />
<h5>
  <NavLink to="/">
    Regresar al inicio
  </NavLink>
</h5>
<hr />
<center>
  <img src={logotipos} alt="logotipos" width="60%" height="20%" />
</center>
</Container>
</div>
)
}
};

```

```
export default Acerca;
```

5.1.2.2.2. Article.jsx

```
import React from "react";
import {
  Col, Container, Row, Button, Modal, ModalBody, ModalHeader
} from 'reactstrap';

import { Map, TileLayer, FeatureGroup, Polygon, Tooltip } from 'react-leaflet';
import axios from "axios";
import ReactLoading from "react-loading";
import download from "downloadjs";
import { Redirect } from "react-router-dom";
import auth from "components/auth/auth.js";

import SNavBar from "components/SNavBar.jsx";
import api from "api_route.js";

class Article extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      info: {},
      loading: true,
      finca: [],
      centroid: [],
      redirect: false,
      redirect_word: "",
      toggleModal: false,
      toggleModalRequest: false,
      textarea: "",
      textarea_request: "",
      loading_mail: false,
      publico: false,
      errorArticle: false
    }
  }

  sendPurpose() {
    var data = {
      id_usuario: auth.getSession().id,
      id_metadato: this.props.match.params.id,
      proposito: this.state.textarea
    }
  }
}
```

```

    };
    axios.post(api.route + "/article/proposito", data)
      .then(res => {
        this.setState({ textarea: "" });
      })
      .catch(err => {
        console.log(err);
        this.setState({ textarea: "" });
      })
  }

  async getFile() {
    this.toggleModal();
    const res = await fetch(api.route + "/article/download/filename/" +
this.props.match.params.id);
    const { filename } = await res.json();
    axios.get(api.route + "/article/download/" + this.props.match.params.id, {
      responseType: "blob"
    }).then(res => {
      console.log(res);
      let blob = new Blob([res.data])
      download(blob, filename);
      this.sendPurpose();
    }).catch(err => {
      console.log(err);
      alert("No se encontro el recurso");
    });
  }

  componentDidMount() {
    if (auth.isAuthenticated()) {
      axios.get(api.route + "/article/get/" + this.props.match.params.id + "/" +
auth.getSession().id)
        .then(res => {
          const { info, finca, publico } = res.data;
          this.setState({ info: info, loading: false, finca: finca, centroid:
finca[0].centroid, publico: publico });
        })
        .catch(err => {
          this.setState({ errorArticle: true });
        })
    } else {
      axios.get(api.route + "/article/get/" + this.props.match.params.id)
        .then(res => {
          const { info, finca, publico } = res.data;

```

```

        this.setState({ info: info, loading: false, finca: finca, centroid:
finca[0].centroid, publico: publico });
    })
    .catch(err => {
        this.setState({ errorArticle: true });
    })
}
}

sendRequestDownload() {
    this.setState({ loading_mail: true });
    axios.post(api.route + "/sendmail", {
        id: auth.getSession().id,
        textarea_request: this.state.textarea_request
    })
    .then(res => {
        this.setState({ textarea_request: "", loading_mail: false })
        this.toggleModalRequest();
        alert("Mensaje enviado con exito");
    })
    .catch(err => {
        console.log(err);
        alert("Error");
    })
}

redirectPclave(w) {
    this.setState({ redirect: true, redirect_word: w });
}

renderRedirect() {
    if (this.state.redirect) {
        return <Redirect to={{
            pathname: "/search/clave",
            words: this.state.redirect_word,
        }} />;
    }
}

toggleModal() {
    this.setState({ toggleModal: !this.state.toggleModal });
}

toggleModalRequest() {

```

```

    this.setState({ toggleModalRequest: !this.state.toggleModalRequest });
  }

  handleInput(e) {
    this.setState({ [e.target.name]: e.target.value });
  }

  render() {
    return (
      <div>
        <SNavBar />
        {
          this.state.errorArticle ? <center><h5>No se encuentra el
recurso</h5></center> : true
        }
        {
          this.state.loading ? (
            <center>
              <ReactLoading type={"bars"} color={"#A5C80A"} />
            </center>
          ) :
            <Container>

              {/* Modal */}
              <Modal
                toggle={this.toggleModal.bind(this)}
                isOpen={this.state.toggleModal} >
                <ModalHeader>Datos para descarga</ModalHeader>
                <ModalBody>
                  <textarea
                    onChange={this.handleInput.bind(this)}
                    name="textarea" cols="47" rows="7" placeholder="Escriba brevemente el motivo
por el cual accederá a este dato y que uso le dará a este.">
                  </textarea>
                </ModalBody>
                <center>
                  <i style={{ "fontSize": "12px" }}>Si el tamaño del archivo es
superior a 1 GB, la descarga podría tardar unos segundos en iniciar.</i>
                  <br/>
                  <Button
                    onClick={this.getFile.bind(this)}
                    color="success"
                    disabled={this.state.textarea === ""}>
                    Descargar
                  </Button>
                </center>
              </Modal>
            </Container>
          </div>

```

```

        <Modal request={this.toggleModalRequest.bind(this)}
isOpen={this.state.toggleModalRequest} >
        <ModalHeader
toggle={this.toggleModalRequest.bind(this)}>Datos para descarga</ModalHeader>
        <ModalBody>
        <p>
            Este dato se encuentra con acceso restringido, si desea
            contar con él póngase en
            contacto con la administración del CIBioFi a través del
            correo electrónico
            <a href="mailto: abc@example.com">
cibiofi@correounivalle.edu.co</a>, enviando una solicitud de acceso en la que se
            especifique el identificador o ID del dato, el email que
            registró en la plataforma
            SILAIN y el motivo por el cual desea contar con el dato.
            Una vez su solicitud sea
            atendida, en caso de ser autorizada podrá continuar con la
            descarga del dato.
        </p>
        </ModalBody>

    </Modal>

    <h3>{this.state.info.titulo}</h3>
    <hr />
    {
        !auth.isAuthenticated() ? (
            <center>
                <p style={{ "backgroundColor": "yellow", "color": "red" }}>
                    Para descargar debes iniciar sesión
                </p>
            </center>
        ) : true
    }
    <Row>
        <Col md="6" lg="6">
            <Row>
                <Col lg="1">
                    <b>
                        <i className="nc-icon nc-pin-3" style={{ "fontSize":
"25px" }}></i>
                    </b>
                </Col>
                <Col lg="6">
                    <b>Departamento: </b>

```

```

        {this.state.info.departamento}
        <br />
        <b>Municipio: </b>
        {this.state.info.municipio}

        <br />
        <b>Finca: </b>
        {this.state.info.finca}
    </Col>
</Row>
<br />
<ul style={{ "textAlign": "left" }}>
    <h5>Características</h5>
    <li><b>Categoria: </b>{this.state.info.categoria}</li>
    <li><b>Subcategoria:
</b>{this.state.info.subcategoria}</li>
    <li><b>Fecha de creacion: </b>{this.state.info.creado}</li>
    <li><b>Fecha de disponibilidad:
</b>{this.state.info.disponibilidad}</li>
    <li><b>Acceso: </b>{this.state.publico ? "Inmediato" :
"Sujeto a autorización"}</li>

</ul>
<ul style={{ "textAlign": "left" }}>
    <h5>Datos de descarga</h5>
    <li><b>Tipo: </b>{this.state.info.tipo}</li>
    <li><b>Formato: </b>{this.state.info.formato}</li>
    <li><b>Tamano: </b>{this.state.info.tamano}</li>

    {
        auth.isAdmin() ? (
            <Button onClick={this.toggleModal.bind(this)}
                color="success">
                Descargar
            </Button>
        ) : (
            <Button onClick={this.state.publico ?
this.toggleModal.bind(this) : this.toggleModalRequest.bind(this)}
                color={this.state.publico ? "success" : "primary"}
                disabled={!auth.isAuthenticated()}>
                {
                    this.state.publico ? "Descargar" : "Solicitar
Descarga"
                }
            </Button>
        )
    }

```

```

        )
    }

</ul>
<ul style={{ "textAlign": "left" }}>
    <h5>Palabras clave</h5>
    {
        this.state.info.pclave.split(",").map((e, i) => (
            <button                                key={i}
onClick={this.redirectPclave.bind(this,         e)}
className="ButtonLikeLink">{e}</button>
        ))
    }
</ul>
{this.state.redirect ? this.renderRedirect() : true}

</Col>

<Col md="6" lg="6">
    <b style={{ "color": "#6bd098" }}>Identificador: </b>
    {this.state.info.idmetadato}
    <br />
    <b>Resumen: </b>
    <br />
    {this.state.info.resumen}
    <br />
    <b>Descripcion: </b>
    <br />
    {this.state.info.descripcion}

    <Map      className="amapa-minimap    leaflet-container-
minimap"    center={this.state.centroid}    zoom={16}    zoomControl={false}
minZoom={16} maxZoom={16}>
        <TileLayer
            attribution='&copy;
            <a
href="http://osm.org/copyright">OpenStreetMap</a> contributors'
            url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
        />
        {
            this.state.finca.length === 1 ? (
                <FeatureGroup color="green">
                    <Tooltip direction="top">
                        <center>

```

```

        <h5>Finca: {this.state.info.finca}</h5>
        <b>Lat: {this.state.centroid[0]}</b>
        <br />
        <b>Lng: {this.state.centroid[1]}</b>
        </center>
      </Tooltip>
      <Polygon positions={this.state.finca[0].poly} />
    </FeatureGroup>
  ) : true
  }
</Map>
</Col>
</Row>
</Container >
}
</div >
)
}
};

```

export default Article;

5.1.2.2.3. Clave.jsx

```

import React from "react";

import {
  Row, Input, Col,
  Alert, InputGroup, InputGroupAddon, InputGroupText, Badge, Container
} from 'reactstrap';

import axios from "axios";
import ReactLoading from "react-loading";

import CustomFilters from "components/CustomFilters.jsx";
import ResultsTable from "components/ResultsTable";
import Pagination from "components/Pagination";

import api from "api_route.js";

class Clave extends React.Component {
  constructor(props) {
    super(props);
    this.state = {

```

```

words: this.props.location.words || "",
word_searched: "",
results: [],
totalResults: 0,
counts_tipos: {},
pages: [],
currentPage: 0,
filters: {
  categoria: "Select",
  subcategoria: "Select",
  municipio: "Select",
  tipo: "Select",
  formato: "Select",
  finca: "Select",
},
loading: false,
}
}

handleInput(e) {
  this.setState({ [e.target.name]: e.target.value });
}

handleSubmit(e) {
  e.preventDefault();
  this.newBasicSearch();
}

newBasicSearch(e) {
  if (this.state.words !== "") {
    const basicURL = api.route + "/basic/search_by_filter";
    this.setState({ loading: true, currentPage: 0 });

    axios.post(basicURL, {
      filters: this.state.filters,
      word: this.state.words,
      currentPage: 0
    })
    .then(res => {
      this.setState({
        results: res.data.result,
        totalResults: res.data.totalResults,
        pages: res.data.pages,
        counts_tipos: res.data.counts,

```

```

        word_searched: this.state.words,
        loading: false,
    });
    })
    .catch(err => {
        console.log(err);
    })
    }
}

getFilters(obj) {
    if (this.state.word_searched !== "") {
        const basicURL = api.route + "/basic/search_by_filter";
        this.setState({ loading: true, currentPage: 0 })

        axios.post(basicURL, {
            filters: obj,
            word: this.state.word_searched,
            currentPage: 0
        })
        .then(res => {

            this.setState({
                results: res.data.result,
                totalResults: res.data.totalResults,
                pages: res.data.pages,
                counts_tipos: res.data.counts,
                word_searched: this.state.words,
                loading: false,
                filters: obj
            });
        })
        .catch(err => {
            console.log(err);
        })
    } else {
        this.setState({
            filters: obj
        });
    }
}

changePage(page) {

```

```

const basicURL = api.route + "/basic/search_by_filter";
this.setState({ loading: true })
axios.post(basicURL, {
  filters: this.state.filters,
  word: this.state.word_searched,
  currentPage: page
})
.then(res => {
  this.setState({
    results: res.data.result,
    totalResults: res.data.totalResults,
    pages: res.data.pages,
    currentPage: page,
    counts_tipos: res.data.counts,
    word_searched: this.state.words,
    loading: false,
  });
})
.catch(err => {
  console.log(err);
})
}

componentDidMount() {
  this.newBasicSearch();
}

render() {
  const keyWord = <div>
    <Col md="12" lg="12">
      <Row>
        <Col md="12" lg="12">
          <form onSubmit={this.handleSubmit.bind(this)}>
            <InputGroup className="no-border">
              <Input
                value={this.state.words} name="words" placeholder="Escriba una palabra clave..."
                onChange={this.handleInput.bind(this)}
              />
              <InputGroupAddon addonType="append">
                <InputGroupText
                  className="icon-click"
                  onClick={this.handleSubmit.bind(this)}
                  <i className="nc-icon nc-zoom-split" />
                </InputGroupText>
              </InputGroupAddon>
            </InputGroup>
          </form>
        </Col>
      </Row>
    </Col>
  </div>

```

```

        </form>
      </Col>
    </Row>
  </Col>
</div>

return (
  <Container>
    <div className="searchBody" style={{ "marginBottom": "323px" }}>
      {keyWord}
      <CustomFilters getFilters={this.getFilters.bind(this)} />
      {
        this.state.loading ? (
          <center>
            <ReactLoading type="bars" color={"#A5C80A"} />
          </center>
        ) : (
          <div>
            <br />
            <div>
              <Alert color={this.state.results.length > 0 ||
this.state.word_searched === "" ? "success" : "danger"}
                style={{ paddingLeft: "10px" }}
              >
                {
                  this.state.results.length > 0
                    ? (
                      <div>
                        <h5><b>Resultados para
                        "{this.state.word_searched}": </b>{this.state.totalResults}</h5>
                        <Row className="datsBigger">
                          <Col>
                            *Archivo crudo: <Badge
pill><b>{this.state.counts_tipos.AC}</b></Badge>
                          </Col>
                          <Col>
                            *Archivo procesado: <Badge
pill><b>{this.state.counts_tipos.AP}</b></Badge>
                          </Col>
                          <Col>
                            *Imagen cruda: <Badge
pill><b>{this.state.counts_tipos.IC}</b></Badge>
                          </Col>
                        </Col>

```

```

                                *Imagen      procesada:      <Badge
pill><b>{this.state.counts_tipos.IP}</b></Badge>
                                </Col>
                                <Col>
                                *Compilación:      <Badge
pill><b>{this.state.counts_tipos.C}</b></Badge>
                                </Col>
                                </Row>
                                </div>
                                )
                                : (
                                this.state.word_searched !== ""
                                ?
                                <h5><b>No hay datos relacionados con:
"{this.state.word_searched}"</b></h5>
                                :
                                <h5><b>Escriba una palabra clave</b></h5>
                                )
                                }
                                </Alert>
                                </div>
                                {
                                this.state.results.length > 0 ?
                                (
                                <div>
                                <ResultsTable      results={this.state.results}
currentPage={this.state.currentPage} />
                                <Pagination      pages={this.state.pages}
currentPage={this.state.currentPage} changePage={this.changePage.bind(this)} />
                                </div>
                                ) : true
                                }
                                </div>
                                }
                                </div>
                                </Container>
                                )
                                }
};

export default Clave;

```

5.1.2.2.4. Login.jsx

```
import React from "react";

import logo from "s.png";

import auth from "components/auth/auth.js";
import ReCAPTCHA from "react-google-recaptcha";
import axios from 'axios';
import api from "api_route.js";

import { Link } from "react-router-dom";

import {
  Card,
  CardHeader,
  CardBody,
  CardFooter,
  Col, Form, FormGroup, Label, Input, Button, Alert
} from "reactstrap";

class Login extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      disabledLoginBtn: true,
      username: "",
      password: "",
      doAnime: false,
      errorLogin: false,
      messageError: "",
    };
  }

  onChange = params => {
    this.setState({
      disabledLoginBtn: false
    });
  };

  handleInput = e => {
    this.setState({
      [e.target.name]: e.target.value
    });
  };
}
```

```

    };

    signin = e => {
      e.preventDefault();

      const delay = 800;

      let data = { user: this.state.username, password: this.state.password };
      let obj;

      axios.post(api.route + "/users/login", data)
        .then(res => {
          obj = {
            "id": res.data.user_id,
            "role": res.data.role
          };

          auth.login(obj, () => {
            this.setState({ doAnime: true });
            window.setTimeout(() => {
              this.props.history.push("/")
            }, delay);
          });
        })
        .catch(err => {
          console.log(err);
          document.getElementById("loginForm").reset();
          this.setState({ errorLogin: true, messageError: "Revisar datos de entrada"
});
          window.setTimeout(() => {
            this.setState({ errorLogin: false });
          }, 2000);
        })
    }

    render() {

      const alert = (this.state.errorLogin) ? <Alert className="animated
rubberBand" color="danger">
        <center>
          <h6>{this.state.messageError}</h6>
        </center>
      </Alert> : true;

      return (

```

```

<div>
  <Col md="4" id="login">
    {alert}
    <Card id="cardLogin" className={this.state.doAnime ? "animated
rotateOut fast" : ""}>
      <CardHeader>
        <center>
          <img src={logo} width="210px" height="210px"
alt="description"></img>
        </center>
      </CardHeader>
      <CardBody>
        <Form onSubmit={this.signin} id="loginForm">
          <FormGroup>
            <Label for="exampleEmail">email</Label>
            <Input onChange={this.handleInput} type="email"
name="username" id="exampleEmail" required />
          </FormGroup>
          <FormGroup>
            <Label for="examplePassword">Contraseña</Label>
            <Input onChange={this.handleInput} type="password"
name="password" id="examplePassword" required />
          </FormGroup>

          <center>
            <ReCAPTCHA

sitekey="6LfP99wUAAAAAPHZbEWQ8x6fXjgBkasCd4ztoqL8"
            onChange={this.onChange}
            />
          </center>

          <center>
            <Button color="success" type="submit"
disabled={this.state.disabledLoginBtn}>ENTRAR</Button>
            <br />
            <b>
              <Link to="/registro">Crea una cuenta</Link>
            </b>
            <br />
            <b>
              <Link to="/">Inicio</Link>
            </b>
          </center>
        </Form>
      </CardBody>
    </Card>
  </Col>
</div>

```

```

        </CardBody>
        <CardFooter>
        </CardFooter>
    </Card>
</Col>
</div>
    )
  }
}

```

export default Login;

5.1.2.2.5. Perfil.jsx

```
import React from "react";
```

```
import {
  Container, Button, Row, Table
} from 'reactstrap';
import { Link } from "react-router-dom";
```

```
import auth from "components/auth/auth.js";
import CreateFile from "components/CreateFile.jsx";
import SNavBar from "components/SNavBar.jsx";
import api from "api_route.js";
import CreateLicence from "components/CreateLicence";
```

```
class Perfil extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      user: {},
      hist: [],
      loading: true
    }
  }
}
```

```
  async componentDidMount() {
    const res = await fetch(api.route + "/users/user/" + auth.getSession().id);
    const res_hist = await fetch(api.route + "/users/user/des_hist/" +
auth.getSession().id);
    const { user } = await res.json();
    const { hist } = await res_hist.json();
    this.setState({
      user: user,
```

```

        hist: hist,
        loading: false
    });
}

logout() {
    auth.logout(() => this.props.history.push("/"));
}

render() {
    return (
        <div>
            <SNavBar />
            <Container>
                <Row>
                    {
                        auth.isAdmin() ?
                        (
                            
                            ) : (
                                
                            )
                        }
                        <h2>Bienvenido {this.state.user.nombres}
{this.state.user.apellidos}</h2>
                    </Row>
                    <b>{this.state.user.email}</b> | <b>{this.state.user.pais}</b>
                    {
                        auth.isAdmin() ? (
                            <div>
                                <CreateFile />
                                <CreateLicence />
                            </div>
                        ) : true
                    }
                </div>
                <center>
                    <h4>Historial de descargas</h4>
                </center>
                <ul>
                    {

```

```

this.state.hist.length !== 0 ?
<Table striped bordered hover>
  <thead>
    <tr style={{ "textAlign": "left" }}>
      <th>#</th>
      <th>Articulo</th>
      <th>Fecha</th>
      <th>Hora</th>
    </tr>
  </thead>
  <tbody>
    {
      this.state.hist.map((ele, i) => (
        <tr key={i}>
          <th>{i + 1}</th>
          <th>
            <Link
              to={"/article/"
ele.idmetadato}>{ele.titulo}</Link>
          </th>
          <th>
            {ele.fecha}
          </th>
          <th>
            {ele.hora}
          </th>
        </tr>
      ))
    }
  </tbody>
</Table>
: <b>No ha hecho ninguna descarga todavía</b>
  }
</ul>
<hr />
<center>
  <Button
    color="danger">Salir</Button>
  </center>
</Container>
</div >
)
}
};

```

```
export default Perfil;
```

5.1.2.2.6. Propiedad.jsx

```
import React from "react";
```

```
import {  
  Row, Col, Alert, Badge, Container  
} from 'reactstrap';
```

```
import axios from "axios";  
import ReactLoading from "react-loading";
```

```
import CustomFilters from "components/CustomFilters.jsx";  
import ResultsTable from "components/ResultsTable";  
import Pagination from "components/Pagination";
```

```
import api from "api_route.js";
```

```
class Propiedad extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      results: [],  
      totalResults: 0,  
      counts_tipos: {},  
      pages: [],  
      currentPage: 0,  
      filters: {  
        categoria: "Select",  
        subcategoria: "Select",  
        municipio: "Select",  
        tipo: "Select",  
        formato: "Select",  
        finca: "Select",  
      },  
      loading: false,  
    }  
  }  
  
  allSelect() {  
    for (let i in this.state.filters) {  
      if (this.state.filters[i] !== "Select") {  
        return false;  
      }  
    }  
  }  
}
```

```

    }
    return true;
  }

  getFilters(obj) {
    const basicURL = api.route + "/basic/search_by_filter";
    this.setState({ loading: true, currentPage: 0 })
    axios.post(basicURL, { filters: obj, word: "", currentPage: 0 })
      .then(res => {

        this.setState({
          results: res.data.result,
          totalResults: res.data.totalResults,
          pages: res.data.pages,
          counts_tipos: res.data.counts,
          loading: false,
          filters: obj
        });
      })
      .catch(err => {
        console.log(err);
      })
  }

  changePage(page) {

    const basicURL = api.route + "/basic/search_by_filter";
    this.setState({ loading: true })
    axios.post(basicURL, {
      filters: this.state.filters,
      word: "",
      currentPage: page
    })
      .then(res => {
        this.setState({
          results: res.data.result,
          totalResults: res.data.totalResults,
          pages: res.data.pages,
          currentPage: page,
          counts_tipos: res.data.counts,
          loading: false,
        });
      })
      .catch(err => {
        console.log(err);
      })
  }

```

```

    })
  }

  render() {
    return (
      <Container>
        <div className="searchBody" style={{ "marginBottom": "372px" }}>
          <CustomFilters getFilters={this.getFilters.bind(this)} />
          {
            this.state.loading ? (
              <center>
                <ReactLoading type={"bars"} color={"#A5C80A"} />
              </center>
            ) : (
              <div>
                <br />
                <div>
                  <Alert
                    color={this.state.results.length > 0 || this.allSelect() ?
"success" : "danger"}
                    style={{ paddingLeft: "10px" }}
                    >
                      {
                        this.state.results.length > 0
                          ? (
                              <div>
                                <h5><b>Resultados según los filtros:
</b>{this.state.totalResults}</h5>
                                <Row className="datsBigger">
                                  <Col>
                                    *Archivo crudo: <Badge
pill><b>{this.state.counts_tipos.AC}</b></Badge>
                                  </Col>
                                  <Col>
                                    *Archivo procesado: <Badge
pill><b>{this.state.counts_tipos.AP}</b></Badge>
                                  </Col>
                                  <Col>
                                    *Imagen cruda: <Badge
pill><b>{this.state.counts_tipos.IC}</b></Badge>
                                  </Col>
                                  <Col>
                                    *Imagen procesada: <Badge
pill><b>{this.state.counts_tipos.IP}</b></Badge>
                                  </Col>
                                </Row>
                              </div>
                            )
                          : null
                        }
                      </div>
                    </Alert>
                  </div>
                <br />
              </div>
            )
          }
        </div>
      </Container>
    )
  }
}

```

```

        <Col>
          *Compilación:
        </Col>
      </Row>
    </div>
  )
  : (
    this.allSelect() ?
    <h5> <b>Seleccione un atributo de
búsqueda</b></h5>
    : <h5><b>No hay datos relacionados</b></h5>
  )
}
</Alert>
</div>
{
  this.state.results.length > 0 ?
  (
    <div>
      <ResultsTable results={this.state.results}
currentPage={this.state.currentPage} />
      <Pagination pages={this.state.pages}
currentPage={this.state.currentPage} changePage={this.changePage.bind(this)} />
    </div>
  ) : true
}
</div>
}
</div>
</Container>
)
}
};

```

export default Propiedad;

5.1.2.2.7. Proyectos.jsx

import React from "react";

import {

```

    Row, Col, Container
  } from 'reactstrap';

import { Router, Route, Switch, NavLink } from "react-router-dom";

import Project1 from "components/P1/Project1.jsx";
import Project2 from "components/P2/Project2.jsx";
import Project3 from "components/P3/Project3.jsx";
import Project4 from "components/P4/Project4.jsx";

class Proyectos extends React.Component {
  constructor(props) {
    super(props);
    this.state = {}
  }

  render() {
    return (
      <Container>
        <div className="searchBody" style={{ "backgroundColor": "white",
"marginBottom": "255px", "paddingBottom": "15px" }}>
          <Router history={this.props.history}>
            <Switch>
              <Route exact path="/search/proyecto" render={props =>
                <div>
                  <center>
                    <i>La Información presentada fue suministrada por las
personas involucradas</i>
                  </center>
                  <br />
                  <Col size="sm">
                    <Row>
                      <Col md="11">
                        <center>
                          <h5>
                            Sustentabilidad ambiental del manejo y uso del
agua en cultivos de arroz de riego. caso estudio: finca arrocera san Jose, municipio
Santander de Quilichao, Departamento del Cauca
                          </h5>
                        </center>
                      </Col>
                      <Col md="1">

```

```

<NavLink                                to="/search/proyecto/p1"
className="proLink">
    <i className="nc-icon nc-zoom-split" />
        </NavLink>
    </Col>
</Row>
<hr />
<Row>
    <Col md="11">
        <center>
            <h5>
                Análisis del efecto del grosor de la lámina de
                agua en el cultivo de arroz a partir del análisis de respuestas espectrales.
            </h5>
        </center>
    </Col>
    <Col md="1">
        <NavLink                                to="/search/proyecto/p2"
className="proLink">
            <i className="nc-icon nc-zoom-split" />
                </NavLink>
            </Col>
        </Row>
        <hr />
        <Row>
            <Col md="11">
                <center>
                    <h5>
                        Estimación de las concentraciones de nitrógeno
                        en el cultivo de arroz a partir de datos hiperespectrales
                    </h5>
                </center>
            </Col>
            <Col md="1">
                <NavLink                                to="/search/proyecto/p3"
className="proLink">
                    <i className="nc-icon nc-zoom-split" />
                        </NavLink>
                    </Col>
                </Row>
                <hr />
                <Row>
                    <Col md="11">
                        <center>

```

```

                    <h5>
                        Integración multiescala entre modelos climáticos
globales y regionales mediante técnicas de inteligencia artificial
                    </h5>
                </center>
            </Col>
            <Col md="1">
                <NavLink
                    to="/search/proyecto/p4"
className="proLink">
                    <i className="nc-icon nc-zoom-split" />
                </NavLink>
            </Col>
        </Row>
    </Col>
</div>
} />
<Route exact path="/search/proyecto/p1" component={Project1}
/>
<Route exact path="/search/proyecto/p2" component={Project2}
/>
<Route exact path="/search/proyecto/p3" component={Project3}
/>
<Route exact path="/search/proyecto/p4" component={Project4}
/>

        </Switch>
    </Router>
</div>
</Container>
)
}
};

```

export default Proyectos;

5.1.2.2.8. Registro.jsx

```

import React from "react";

import axios from 'axios';
import api from "api_route.js";

import { Link } from "react-router-dom";

import {

```

```

    Card,
    CardHeader,
    CardBody,
    CardFooter,
    Col, Row, Form, FormGroup, Label, Input, Button, Alert
  } from "reactstrap";

```

```

class Login extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      nombres: "",
      apellidos: "",
      email: "",
      pais: "",
      departamento: "",
      ciudad: "",
      institucion: "",
      ocupacion: "",
      password: "",
      confirm: "",
      doAnime: false,
      errorRegistro: false,
      registroSuccess: false,
    };
  }

  handleInput = e => {
    this.setState({
      [e.target.name]: e.target.value
    });
  };

  signup = e => {
    e.preventDefault();
    if (!this.validation()) {
      this.setState({ errorRegistro: true });
      window.setTimeout(() => {
        this.setState({ errorRegistro: false });
      }, 2000);
      return ;
    }
    const data = {
      nombres: this.state.nombres,
      apellidos: this.state.apellidos,

```

```

        email: this.state.email,
        pais: this.state.pais,
        departamento: this.state.departamento,
        ciudad: this.state.ciudad,
        institucion: this.state.institucion,
        ocupacion: this.state.ocupacion,
        password: this.state.password
    }
    axios.post(api.route + "/users/register", data)
        .then(res => {
            this.setState({ registroSuccess: true, password: "" });
            document.getElementById("registerForm").reset();
            window.setTimeout(() => {
                this.setState({ registroSuccess: false });
            }, 2000);
        })
        .catch(err => {
            this.setState({ errorRegistro: true });
            window.setTimeout(() => {
                this.setState({ errorRegistro: false });
            }, 2000);
        })
    }

    validation() {
        return this.state.nombres !== "" &&
            this.state.apellidos !== "" &&
            this.state.email !== "" &&
            this.state.pais !== "" &&
            this.state.departamento !== "" &&
            this.state.ciudad !== "" &&
            this.state.institucion !== "" &&
            this.state.ocupacion !== "" &&
            this.state.password !== "" &&
            this.state.confirm !== "" &&
            this.state.password === this.state.confirm
    }

    render() {

        const alert = (this.state.errorRegistro) ? <Alert className="animated
rubberBand" color="danger">
            <center>
                <h6>Verifica la entrada</h6>
            </center>

```

```

</Alert> : true;

const alertSuccess = (this.state.registroSuccess) ? <Alert
className="animated rubberBand" color="success">
  <center>
    <h6>Registrado Exitosamente!</h6>
  </center>
</Alert> : true;

return (
  <div>
    <Col md="6" id="login">
      {alert}
      {alertSuccess}
      <Card id="cardLogin" className={this.state.doAnime ? "animated
zoomOutUp" : ""}>
        <CardHeader>
          <center>
            <h2>Crear Cuenta</h2>
          </center>
        </CardHeader>
        <CardBody>
          <Form onSubmit={this.signin} id="registerForm">
            <FormGroup>
              <Row>
                <Col>
                  <Label>Nombres</Label>
                  <Input onChange={this.handleInput} name="nombres"
type="text" required />
                </Col>
                <Col>
                  <Label>Apellidos</Label>
                  <Input onChange={this.handleInput} name="apellidos"
type="text" required />
                </Col>
              </Row>
              <Label>email</Label>
              <Input onChange={this.handleInput} name="email"
type="email" required />
              <Row>
                <Col>
                  <Label>Pais</Label>
                  <Input onChange={this.handleInput} name="pais"
type="text" required />
                </Col>

```

```

        <Col>
            <Label>Departamento</Label>
            <Input
name="departamento" type="text" required />
            </Col>
        </Row>
        <Row>
            <Col>
                <Label>Ciudad</Label>
                <Input onChange={this.handleInput} name="ciudad"
type="text" required />
            </Col>
            <Col>
                <Label>Institucion</Label>
                <Input onChange={this.handleInput} name="institucion"
type="text" required />
            </Col>
        </Row>
        <Label>Ocupacion</Label>
        <Input onChange={this.handleInput} name="ocupacion"
type="text" required />
        <Label for="examplePassword">Contrasena</Label>
        <Input onChange={this.handleInput} type="password"
name="password" required />
        <Label
Contrasena</Label>
            for="examplePassword">Confirmar
Contrasena</Label>
        <Input onChange={this.handleInput}
            style={this.state.confirm === this.state.password &&
this.state.confirm !== " " && this.state.password !== " " ? { "backgroundColor":
"#a7eb77" } : { "backgroundColor": "pink" }}
            type="password" name="confirm" required />
    </FormGroup>
    <center>
        <Button color="success" type="submit"
onClick={this.signup.bind(this)} >REGISTRAR</Button>
        <br />
        <b>
            <Link to="/login">Iniciar Sesión</Link>
        </b>
        <br />
        <b>
            <Link to="/">Inicio</Link>
        </b>
    </center>

```

```

        </Form>
      </CardBody>
      <CardFooter>
      </CardFooter>
    </Card>
  </Col>
</div>
)
}
}

```

export default Login;

5.1.2.2.9. Start.jsx

```

import React from "react";
import Footer from "components/Footer.jsx";

import { Redirect } from "react-router-dom";

import {
  InputGroup,
  Input,
  InputGroupAddon,
  InputGroupText,
  Container,
  Button,
  Col
} from "reactstrap";

import SNavBar from "components/SNavBar.jsx";

import logo from "s.png";

class Start extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      redirect: false,
      redirect_by_searchType: false,
      words: "",
      search_type: "/search/clave",
    }
  }
}

```

```

handleInput(e) {
  this.setState({
    [e.target.name]: e.target.value
  });
}

setRedirect() {
  if (this.state.words !== "") {
    this.setState({ redirect: true });
  }
}

renderRedirect() {
  if (this.state.redirect) {
    const w = this.state.words;
    return <Redirect to={{
      pathname: this.state.search_type,
      words: w,
    }} />;
  }
}

setRedirect_searchType(type) {
  this.setState({ redirect: true, search_type: "/search/" + type })
}

render() {
  return (
    <div className="bgForest_top">
      <SNavBar />
      <div style={{ "paddingBottom": "110px" }} >
        <center>
          <img src={logo} width="280px" height="280px" alt="description"
className="animated fadeInDown fast"></img>
          <br></br>
          <br></br>
          <b id="logoMeans">Sistema de Información del Laboratorio de
Agricultura Inteligente</b>
          <br></br>
          <br></br>
          <br></br>
          <Container className="animated fadeIn">
            <Col md="7" sm="12" xs="12">

```

```

        <form      onSubmit={this.setRedirect.bind(this)}      style={{
"backgroundColor": "white" }}>
        <InputGroup className="no-border">
            <Input      value={this.state.words}
onChange={this.handleInput.bind(this)} name="words" placeholder="Escriba una
palabra clave..." />
            <InputGroupAddon addonType="append">
                <InputGroupText>
                    <i className="nc-icon nc-zoom-split" />
                </InputGroupText>
            </InputGroupAddon>
        </InputGroup>
    </form>
</Col>
</Container>
<Button      className="silain_green"
onClick={this.setRedirect.bind(this)}>Buscar</Button>
    {this.state.redirect ? this.renderRedirect() : true}
</center>
</div>
<Footer />
</div>
)
}
}
}

```

export default Start;

5.1.2.2.10. TheMap.jsx

```
import React from "react";
```

```
import {
    Modal, ModalHeader, ModalBody, Table, Button
} from "reactstrap";
```

```
import { Map, Popup, TileLayer, LayersControl, FeatureGroup, Polygon } from 'react-
leaflet';
import PropiedadByMap from "components/Map/PropiedadByMap";
import FiltersOnMap from "components/Map/FiltersOnMap";
```

```
import axios from "axios";
import ReactLoading from "react-loading";
import Control from 'react-leaflet-control';
```

```

import api from "api_route.js";

class TheMap extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      position: [3.2175377205303732, -76.53764390954167],
      zoom: 7,
      clickMarkerState: false,

      finca: [],
      departamento: [],
      municipio: [],
      modal: false,
      modalDist: false,

      loading: false,
      ublication: "",
      ubi_type: "",
      results: [],
      resultsDist: [],
      counts_tipos: {},

      totalResults: 0,
      pages: [],
      currentPage: 0,

      selections: {
        departamento: "Select",
        municipio: "Select",
        finca: "Select",
      }
    }
  }

  openToggle = () => {
    this.setState({
      modal: true
    });
  }

  closeToggle = () => {
    this.setState({
      modal: false,
      results: [],
    })
  }
}

```

```

        counts_tipos: {}
    });
}

openToggleDist = () => {
    this.setState({
        modalDist: true
    });
}

closeToggleDist = () => {
    this.setState({
        modalDist: false,
    });
}

buscarUbication(ubi_type, ubicacion) {
    console.log(ubi_type);
    this.setState({ loading: true, ubicacion: ubicacion, ubi_type: ubi_type,
currentPage: 0 });
    this.openToggle();
    axios.post(api.route + "/map/ubicacion_by_filter", {
        filters: {
            cultivo: "Select",
            categoria: "Select",
            subcategoria: "Select",
            tipo: "Select",
            formato: "Select",
        },
        ubicacion: ubicacion,
        ubi_type: ubi_type,
        currentPage: 0
    })
    .then(res => {
        this.setState({
            results: res.data.result,
            totalResults: res.data.totalResults,
            pages: res.data.pages,
            counts_tipos: res.data.counts,
            loading: false,
            ubi_type: this.state.ubi_type,
        });
    })
    .catch(err => {
        console.log(err);
    });
}

```

```

    })
  }

  buscarFinsaCercana(marker) {
    this.setState({ loading: true, currentPage: 0, clickMarkerState: false });
    document.getElementById("mapdiv").style.cursor = "auto";
    this.openToggleDist();
    axios.post(api.route + "/map/finsa_closer", { marker: marker })
      .then(res => {
        this.setState({
          resultsDist: res.data,
          loading: false,
        });
      })
      .catch(err => {
        console.log(err);
      })
  }

  resultadosFinsaCercana(nameUbication) {
    this.closeToggleDist();
    this.buscarUbication("finsa", nameUbication);
  }

  changePage(page) {

    this.setState({ loading: true })
    axios.post(api.route + "/map/ubication_by_filter", {
      filters: {
        cultivo: "Select",
        categoria: "Select",
        subcategoria: "Select",
        tipo: "Select",
        formato: "Select",
      },
      ubication: this.state.ubication,
      ubi_type: this.state.ubi_type,
      currentPage: page
    })
      .then(res => {
        this.setState({
          results: res.data.result,
          totalResults: res.data.totalResults,
          pages: res.data.pages,
          counts_tipos: res.data.counts,

```

```

        loading: false,
        ubi_type: this.state.ubi_type,
        currentPage: page
    });
    })
    .catch(err => {
        console.log(err);
    })
}

getFilters(obj) {
    const basicURL = api.route + "/map/getpoly";
    axios.post(basicURL, { filters: obj })
        .then(res => {
            var { departamento, municipio, finca } = res.data;
            this.setState({
                departamento: departamento,
                municipio: municipio,
                finca: finca
            });

            if (departamento.length === 0) {
                this.setState({ position: [3.2175377205303732, -76.53764390954167],
zoom: 7 });
            } else if (municipio.length === 0) {
                this.setState({ position: departamento[0].centroid, zoom: 8 });
            } else if (finca.length === 0) {
                this.setState({ position: municipio[0].centroid, zoom: 10 });
            } else {
                this.setState({ position: finca[0].centroid, zoom: 16 });
            }
        })
        .catch(err => {
            console.log(err);
        })
    }

    clickMarkerOn() {
        this.setState({ clickMarkerState: true });
        document.getElementById("mapdiv").style.cursor = "crosshair";
    }

    searchByClick(e) {
        if (this.state.clickMarkerState) {
            this.buscarFincaCercana([e.latlng.lat, e.latlng.lng]);
        }
    }
}

```

```

    }
  }
}

render() {

  const modal = <div>
    <Modal lg="12" size="modal-dialog modal-lg" className="modal-lg"
    isOpen={this.state.modal} toggle={this.closeToggle.bind(this)}
    fade={false}>
      <ModalHeader toggle={this.closeToggle}>
        {this.state.ubication}
      </ModalHeader>
      <ModalBody>
        {
          this.state.loading ? (
            <center>
              <ReactLoading type="bars" color="#A5C80A" />
            </center>
          ) : (
            this.state.results.length > 0 ?
              <PropiedadByMap ubicacion={this.state.ubication}
                results={this.state.results}
                counts_tipos={this.state.counts_tipos}
                ubi_type={this.state.ubi_type}
                totalResults={this.state.totalResults}
                pages={this.state.pages}
                currentPage={this.state.currentPage}
                changePage={this.changePage.bind(this)}
              />
              : true
            )
          )
        }
      </ModalBody>
    </Modal>
  </div>

  const modalDist = <div>
    <Modal lg="12" size="modal-dialog modal-lg" className="modal-lg"
    isOpen={this.state.modalDist} toggle={this.closeToggleDist.bind(this)}
    fade={false}>
      <ModalHeader toggle={this.closeToggleDist}>
        Fincas
      </ModalHeader>
      <ModalBody>
        {

```

```

this.state.loading ? (
  <center>
    <ReactLoading type={"bars"} color={"#A5C80A"} />
  </center>
) : (
  this.state.resultsDist.length > 0 ?
    <Table striped bordered style={{ "textAlign": "center" }}>
      <thead>
        <tr>
          <th>#</th>
          <th>Nombre</th>
          <th>Distancia en KM</th>
          <th>Resultados por finca</th>
        </tr>
      </thead>
      <tbody>
        {
          this.state.resultsDist.map((ele, i) => (
            <tr>
              <th key={i} scope="row">{i + 1}</th>
              <td>{ele.finca}</td>
              <td>{ele.distancia_km}</td>
              <td style={{ "width": "20%" }}>
                <Button color="primary"
onClick={this.resultadosFincaCercana.bind(this, ele.finca)}>
                  <span role="img" aria-label=".">🔍</span>
                </Button>
              </td>
            </tr>
          ))
        }
      </tbody>
    </Table>
    : true
  )
}

</ModalBody>
</Modal>
</div >

return (
  <div>
    <center>

```

```

        {modal}
        {modalDist}
        <Map className="amapa" id="mapdiv" center={this.state.position}
zoom={this.state.zoom} onClick={this.searchByClick.bind(this)}>
        <LayersControl position="topright">
            <LayersControl.BaseLayer name="Normal" checked="true">
                <TileLayer
                    attribution='&copy;
href="http://osm.org/copyright">OpenStreetMap</a> contributors'
                    url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"
                />
            </LayersControl.BaseLayer>
            <LayersControl.BaseLayer name="Blanco y Negro">
                <TileLayer
                    attribution='&copy;
href="http://osm.org/copyright">OpenStreetMap</a> contributors'
                    url="https://tiles.wmflabs.org/bw-mapnik/{z}/{x}/{y}.png"
                />
            </LayersControl.BaseLayer>

            <LayersControl.BaseLayer name="Transporte">
                <TileLayer
                    attribution='&copy;
href="http://osm.org/copyright">OpenStreetMap</a> contributors'
                    url="https://tile.memomaps.de/tilegen/{z}/{x}/{y}.png"
                />
            </LayersControl.BaseLayer>
        {/* End Layers */}
        <Control position="topright">
            <div className="info legend">
                <i style={{ "backgroundColor": "purple" }} ></i>
                Departamentos
                <br />
                <i style={{ "backgroundColor": "cyan" }} ></i>
                Municipios
                <br />
                <i style={{ "backgroundColor": "blue" }} ></i>
                Fincas
            </div>
        </Control>
        <Control position="topright">
            <FiltersOnMap getFilters={this.getFilters.bind(this)} />
        </Control>
        <Control>
            <div style={{ "width": "155px", "fontSize": "14px" }}>

```

```

        <b>
            Capturar
        </b>
        <br />
        <button
            style={{
                "backgroundColor":
this.state.clickMarkerState ? "cyan" : "white", "width": "100%" }}
            onClick={this.clickMarkerOn.bind(this)}
            title="Con este boton se activara la busqueda por
distancias en el mapa al dar click sobre este">
            <span role="img" aria-label="." style={{ "fontSize": "20px"
}}>
                
            </span>
        </button>
    </div>
</Control>
</LayersControl>
{
    this.state.departamento.length === 1 ? (
        <FeatureGroup color="purple">
            <Popup>
                <center>
                    <h5>{this.state.departamento[0].departamento}</h5>
                    <button
                        onClick={this.buscarUbication.bind(this,
"departamento",
                        this.state.departamento[0].departamento)}
                        className="btn_search_map">
                        Buscar
                    </button>
                </center>
            </Popup>
            <Polygon positions={this.state.departamento[0].poly} />
        </FeatureGroup>
    ) : true
}
{
    this.state.municipio.length === 1 ? (
        <FeatureGroup color="cyan">
            <Popup>
                <center>
                    <h5>{this.state.municipio[0].municipio}</h5>
                    <button
                        onClick={this.buscarUbication.bind(this,
"municipio", this.state.municipio[0].municipio)}
                        className="btn_search_map2">
                        Buscar
                    </button>

```

```

        </center>
      </Popup>
      <Polygon positions={this.state.municipio[0].poly} />
    </FeatureGroup>
  ) : true
}
{
  this.state.finca.length === 1 ? (
    <FeatureGroup color="blue">
      <Popup>
        <center>
          <h5>{this.state.finca[0].finca}</h5>
          <button onClick={this.buscarUbication.bind(this, "finca",
this.state.finca[0].finca)} className="btn_search_map3">
            Buscar
          </button>
        </center>
      </Popup>
      <Polygon positions={this.state.finca[0].poly} />
    </FeatureGroup>
  ) : true
}
</Map>
</center>
</div >
)
}
}

```

export default TheMap;

5.1.2.2.11. Search.jsx

```

import React from "react";

import { Provider } from "react-redux";
import store from "../store";

import { Router, Route, Switch } from "react-router-dom";
import Footer from "components/Footer.jsx";

import SNavBar from "components/SNavBar.jsx";

import Clave from "views/Clave.jsx";
import Propiedad from "views/Propiedad.jsx";

```

```

import TheMap from "views/TheMap.jsx";
import Proyectos from "views/Proyectos.jsx";

class Search extends React.Component {
  constructor(props) {
    super(props);
    this.state = {}
  }

  render() {
    return (
      <Provider store={store}>
        <div className="bgForest_center">
          <SNavBar />
          <div>
            <center>

              </center>
            </div>
            <br />
            <Router history={this.props.history}>
              <Switch>
                <Route path="/search/clave" component={Clave} />
                <Route path="/search/propiedad" component={Propiedad} />
                <Route path="/search/mapa" component={TheMap} />
                <Route path="/search/proyecto" component={Proyectos} />
              </Switch>
            </Router>
            <Footer />
          </div>
        </Provider>
      )
    );
  }
};

export default Search;

```

5.1.2.3. Ruta de conexión con el Back-End

```

var api = { route: "http://localhost:8000" };

export default api;

```

5.1.2.4. Compilación de vistas y componentes

```
import React from "react";
import { createBrowserHistory } from "history";
import { Router, Route, Switch, Redirect } from "react-router-dom";
import Start from "views/Start.jsx";
import Searching from "views/Searching.jsx";
import Acerca from "views/Acerca.jsx";
import Article from "views/Article.jsx";
import Login from "views/Login.jsx";
import Registro from "views/Registro.jsx";
import Perfil from "views/Perfil.jsx";
import Search from "layouts/Search.jsx";

import { ProtectedRoute, ProtectedLoginRoute } from
"components/auth/ProtectedRoute.js";

const hist = createBrowserHistory();

function App() {
  return (
    <Router history={hist}>
      <link rel="stylesheet" href="https://unpkg.com/leaflet@1.6.0/dist/leaflet.css"
integrity="sha512-
xwE/Az9zrjBlphAcBb3F6JVqxf46+CDLwflMHloNu6KEQCAWi6HcDUbeOfBlptF7tc
CzusKFjFw2yuvEpDL9wQ==" crossOrigin="" />
      <script src="https://unpkg.com/leaflet@1.6.0/dist/leaflet.js"
integrity="sha512-
gZwlG9x3wUXg2hdXF6+rVkLF/0Vi9U8D2Ntg4Ga5I5BZpVkvVxIJWbSQtXPSiUTtC
0TjtGomxa1AJPuV0Cpthew==" crossOrigin=""></script>

      <Switch>
        <Route exact path="/" component={Start} />
        <Route exact path="/searching" component={Searching} />
        <Route exact path="/acerca" component={Acerca} />
        <Route exact path="/article/:id" component={Article} />

        <ProtectedLoginRoute exact path="/login" component={Login} />
        <ProtectedLoginRoute exact path="/registro" component={Registro} />
        <ProtectedRoute exact path="/perfil" component={Perfil} />

        <Route path="/search" render={props => <Search {...props} />}
history={hist} />

        <Redirect exact to="/" />
      </Switch>
    </Router>
  );
}
```

```

        </Switch>
      </Router>
    )
  }

export default App;

```

5.1.2.5. Índice del Front-End

```

/*!

=====
* Paper Dashboard React - v1.1.0
=====

* Product Page: https://www.creative-tim.com/product/paper-dashboard-react
* Copyright 2019 Creative Tim (https://www.creative-tim.com)

* Licensed under MIT (https://github.com/creativetimofficial/paper-dashboard-react/blob/master/LICENSE.md)

* Coded by Creative Tim

=====

* The above copyright notice and this permission notice shall be included in all copies
or substantial portions of the Software.

*/
import React from "react";
import ReactDOM from "react-dom";

import "bootstrap/dist/css/bootstrap.css";
import "assets/css/paper-dashboard.css";
import "assets/css/paper-dashboard.css.map";
import "assets/css/paper-dashboard.min.css";

import "assets/css/index.css";
import "perfect-scrollbar/css/perfect-scrollbar.css";
import "assets/css/animate.css";
import "assets/css/login.css";

import App from 'App.js';

ReactDOM.render(<App />, document.getElementById('root'));

```

5.1.3. Anexo 2. Manual de usuario

5.1.3.1. Descripción de las páginas de la plataforma

Enseguida se dará una descripción de los elementos presentes en cada una de las páginas y ventanas de la plataforma.

5.1.3.1.1. Página de inicio



Figura 46. Descripción de la página de inicio

- 1- Enlace para ir a la página de inicio
- 2- Enlace para ir a la utilidad de Búsqueda por Palabras Clave
- 3- Enlace para ir a la utilidad de Exploración de Datos
- 4- Enlace para ir a la utilidad del Buscador Geográfico
- 5- Enlace para ir a la utilidad de Proyectos de Investigación
- 6- Enlace para ir a iniciar sesión ver panel de usuario (si ya ha iniciado sesión)
- 7- Logo de SILAIN
- 8- Nombre del sistema
- 9- Barra de búsquedas por palabra clave
- 10- Botón para realizar la búsqueda por palabra clave
- 11- Enlace para ir a la sección Acerca de SILAIN
- 12- Derechos de autor de la plantilla utilizada
- 13- Logo y enlace para ir a la página del CIBioFi

5.1.3.1.2. Página de inicio de sesión

The image shows a login page for 'SILAIN'. At the top is a circular logo with a magnifying glass icon and the word 'SILAIN' (1). Below the logo is an email input field (2). Underneath the email field is a password input field labeled 'Contraseña' (3). Below the password field is a reCAPTCHA verification area with a checkbox and the text 'No soy un robot' (4). Below the reCAPTCHA area is a green button labeled 'ENTRAR' (5). Below the 'ENTRAR' button are two links: 'Crea una cuenta' (6) and 'Inicio' (7).

Figura 47. Descripción de la página de inicio de sesión

- 1- Logo de SILAIN
- 2- Caja de texto para ingresar el email
- 3- Caja de texto para ingresar la contraseña
- 4- Verificador antibots
- 5- Botón para iniciar sesión
- 6- Enlace para crear una cuenta en la plataforma
- 7- Enlace para ir a la página de inicio

5.1.3.1.3. Página de creación de cuenta

Crear Cuenta

Nombres 1 	Apellidos 2
email 3 	
Pais 4 	Departamento 5
Ciudad 6 	Institucion 7
Ocupacion 8 	
Contraseña 9 	
Confirmar Contraseña 10 	

REGISTRAR 11

[Iniciar Sesión](#) 12

[Inicio](#) 13

Figura 48. Descripción de la página de registro

- 1- Caja de texto para ingreso de nombre(s)
- 2- Caja de texto para ingreso de apellido(s)
- 3- Caja de texto para ingreso de email
- 4- Caja de texto para ingreso de país
- 5- Caja de texto para ingreso de departamento
- 6- Caja de texto para ingreso de ciudad
- 7- Caja de texto para ingreso de institución
- 8- Caja de texto para ingreso de ocupación
- 9- Caja de texto para ingreso de contraseña
- 10- Caja de texto para ingreso de confirmación de contraseña
- 11- Botón para registrar la cuenta
- 12- Enlace para ir a iniciar sesión
- 13- Enlace para ir a la página de inicio

5.1.3.1.4. Página del Buscador por Palabra Clave

The screenshot shows the SILAIN web application interface. At the top, a navigation bar contains links: SILAIN (1), Búsqueda por palabras Clave (2), Exploración de datos (3), Buscador Geográfico (4), Proyectos de Investigación (5), and a user profile icon labeled 'normal' (6). Below the navigation bar is a large search area with a background image of rice. It features a search input field (7) with the placeholder text 'Escriba una palabra clave...'. Below the input field are six filter dropdown menus: Categoría (8), Subcategoría (9), Municipio (10), Finca (11), Tipo (12), and Formato (13), each with a 'Select' button. Below these filters is a green button labeled 'Escriba una palabra clave'. At the bottom of the page, there is a footer section with a link 'Acerca de SILAIN' (14), a copyright notice '© 2020, made with ❤️ by Creative Tim' (15), and the CIBioFi logo with a link (16).

Figura 49. Descripción del buscador de palabras clave

- 1- Enlace para ir a la página de inicio
- 2- Enlace para ir a la utilidad de Búsqueda por Palabras Clave
- 3- Enlace para ir a la utilidad de Exploración de Datos
- 4- Enlace para ir a la utilidad del Buscador Geográfico
- 5- Enlace para ir a la utilidad de Proyectos de Investigación
- 6- Enlace para ir a iniciar sesión ver panel de usuario (si ya ha iniciado sesión)
- 7- Barra de búsquedas por palabra clave
- 8- Lista desplegable para filtrar resultados por categoría
- 9- Lista desplegable para filtrar resultados por subcategoría
- 10- Lista desplegable para filtrar resultados por municipio
- 11- Lista desplegable para filtrar resultados por finca
- 12- Lista desplegable para filtrar resultados por tipo
- 13- Lista desplegable para filtrar resultados por formato
- 14- Enlace para ir a la sección Acerca de SILAIN
- 15- Derechos de autor de la plantilla utilizada
- 16- Logo y enlace para ir a la página del CIBioFi

5.1.3.1.5. Resultados en el Buscador por Palabra Clave

arroz

Resultados para "arroz": 295

*Archivo crudo: 0 *Archivo procesado: 0 *Imagen cruda: 20 *Imagen procesada: 275 *Compilación: 0

#	TÍTULO Y RESUMEN	TIPO	FORMATO	TAMAÑO	CREADO	DISPONIBILIDAD	ACCESO
1	Imágenes multiespectrales crudas en finca Calderón Imágenes multiespectrales (B, C, R, BORDE ROJO, NIR) capturadas el 01 de mayo de 2017 en vuelo con dron, sobre finca arrocera Calderón en Jamundí, Valle del Cauca.	Imagen cruda	tif	1.7 GB	01/05/2017	01/06/2020	Público
2	Modelo Digital de Superficie de finca Calderón Modelo Digital de Superficie generado con las imágenes del 01 de mayo de 2017 de la finca arrocera Calderón en Jamundí, Valle del Cauca.	Imagen procesada	tif	738 KB	09/02/2017	01/06/2020	Público
3	Índice CL_re (Chlorophyll Index- Índice de clorofila) en finca Calderón Índice de Clorofila (borde rojo) generado de las imágenes del 01 de mayo de 2017 para la finca arrocera Calderón, ubicada en Jamundí, Valle del Cauca	Imagen procesada	tif	3 MB	29/05/2017	01/06/2020	Público

Figura 50. Descripción de resultados en buscador de palabras clave

20	Ortomosaico banda NIR de finca Calderón Ortomosaico generado con las imágenes de la banda espectral NIR (Infrarrojo cercano) del 01 de mayo de 2017 de la finca Calderón, en Jamundí, Valle del Cauca.	Imagen procesada	tif	1.2 MB	12/02/2017	01/06/2020	Público
----	--	------------------	-----	--------	------------	------------	---------

Figura 51. Continuación descripción de resultados en buscador de palabras clave

- 1- Número de datos encontrados tras la búsqueda
- 2- Número de datos encontrados de tipo archivo crudo
- 3- Número de datos encontrados de tipo archivo procesado
- 4- Número de datos encontrados de tipo imagen cruda
- 5- Número de datos encontrados de tipo imagen procesada
- 6- Número de datos encontrados de tipo compilación
- 7- Título del dato
- 8- Resumen del dato
- 9- Tipo del dato
- 10- Formato del dato
- 11- Tamaño del dato
- 12- Fecha de creación del dato

- 13- Fecha de carga del dato a la plataforma
- 14-Tipo de acceso al dato
- 15- Enlaces a otras páginas de resultados

5.1.3.1.6. Página de la exploración de datos

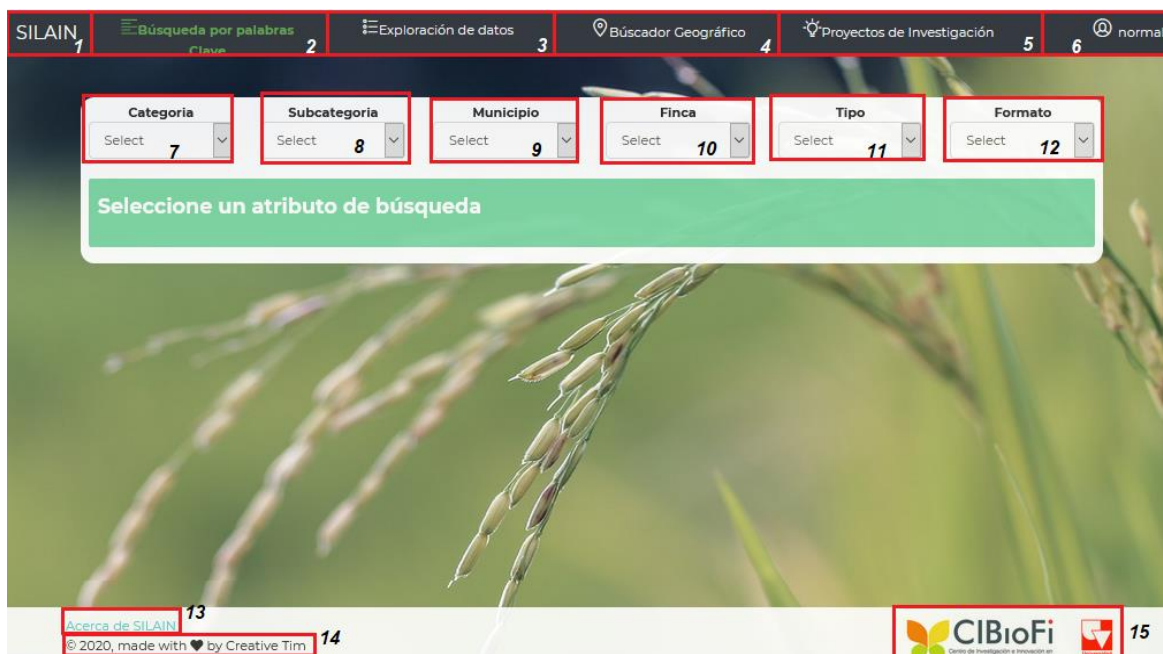


Figura 52. Descripción del explorador de datos

- 1- Enlace para ir a la página de inicio
- 2- Enlace para ir a la utilidad de Búsqueda por Palabras Clave
- 3- Enlace para ir a la utilidad de Exploración de Datos
- 4- Enlace para ir a la utilidad del Buscador Geográfico
- 5- Enlace para ir a la utilidad de Proyectos de Investigación
- 6- Enlace para ir a iniciar sesión ver panel de usuario (si ya ha iniciado sesión)
- 7- Lista desplegable para explorar por categoría
- 8- Lista desplegable para explorar por subcategoría
- 9- Lista desplegable para explorar por municipio
- 10- Lista desplegable para explorar por finca
- 11- Lista desplegable para explorar por tipo
- 12- Lista desplegable para explorar por formato
- 13- Enlace para ir a la sección Acerca de SILAIN
- 14- Derechos de autor de la plantilla utilizada
- 15- Logo y enlace para ir a la página del CIBioFi

5.1.3.1.7. Página de resultados en el Explorador de Datos



Figura 53. Descripción de resultados en explorador de datos

- 1- Número de datos encontrados tras la búsqueda
- 2- Número de datos encontrados de tipo archivo crudo
- 3- Número de datos encontrados de tipo archivo procesado
- 4- Número de datos encontrados de tipo imagen cruda
- 5- Número de datos encontrados de tipo imagen procesada
- 6- Número de datos encontrados de tipo compilación
- 7- Título del dato
- 8- Resumen del dato
- 9- Tipo del dato
- 10- Formato del dato
- 11- Tamaño del dato
- 12- Fecha de creación del dato
- 13- Fecha de carga del dato a la plataforma
- 14- Tipo de acceso al dato
- 15- Enlaces a otras páginas de resultados

5.1.3.1.8. Página del Buscador Geográfico

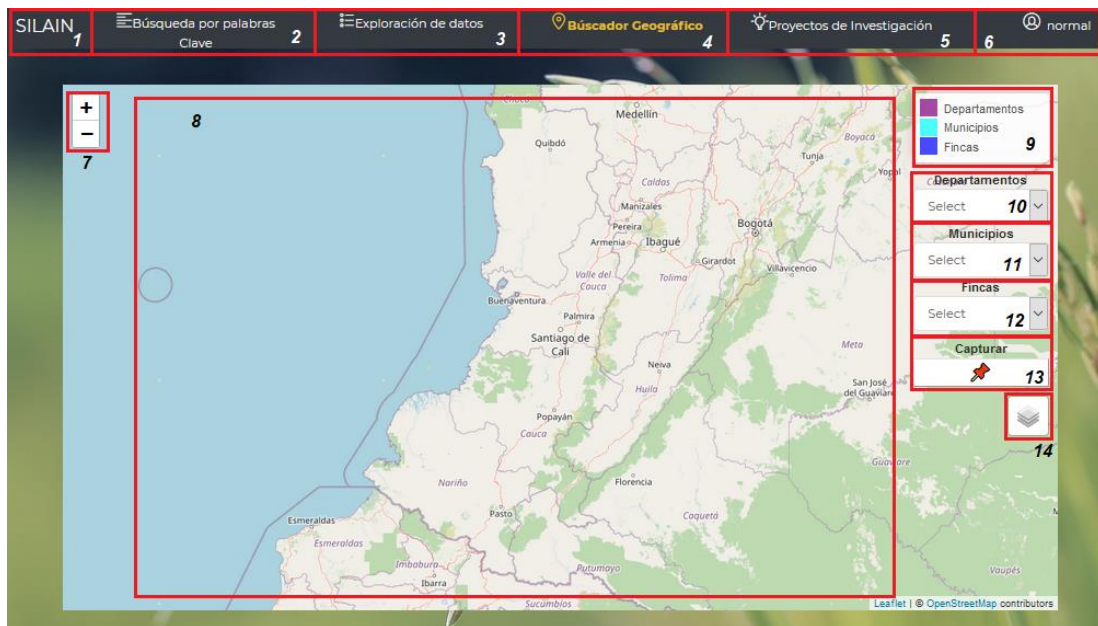


Figura 54. Descripción del buscador geográfico

- 1- Enlace para ir a la página de inicio
- 2- Enlace para ir a la utilidad de Búsqueda por Palabras Clave
- 3- Enlace para ir a la utilidad de Exploración de Datos
- 4- Enlace para ir a la utilidad del Buscador Geográfico
- 5- Enlace para ir a la utilidad de Proyectos de Investigación
- 6- Enlace para ir a iniciar sesión ver panel de usuario (si ya ha iniciado sesión)
- 7- Botones para acerca y alejar el mapa
- 8- Mapa
- 9- Leyenda del mapa
- 10- Lista desplegable para mostrar departamentos
- 11- Lista desplegable para mostrar municipios
- 12- Lista desplegable para mostrar fincas
- 13- Botón para buscar fincas más cercanas
- 14- Botón para cambiar de mapa base

5.1.3.1.9. Página de resultados del Buscador Geográfico

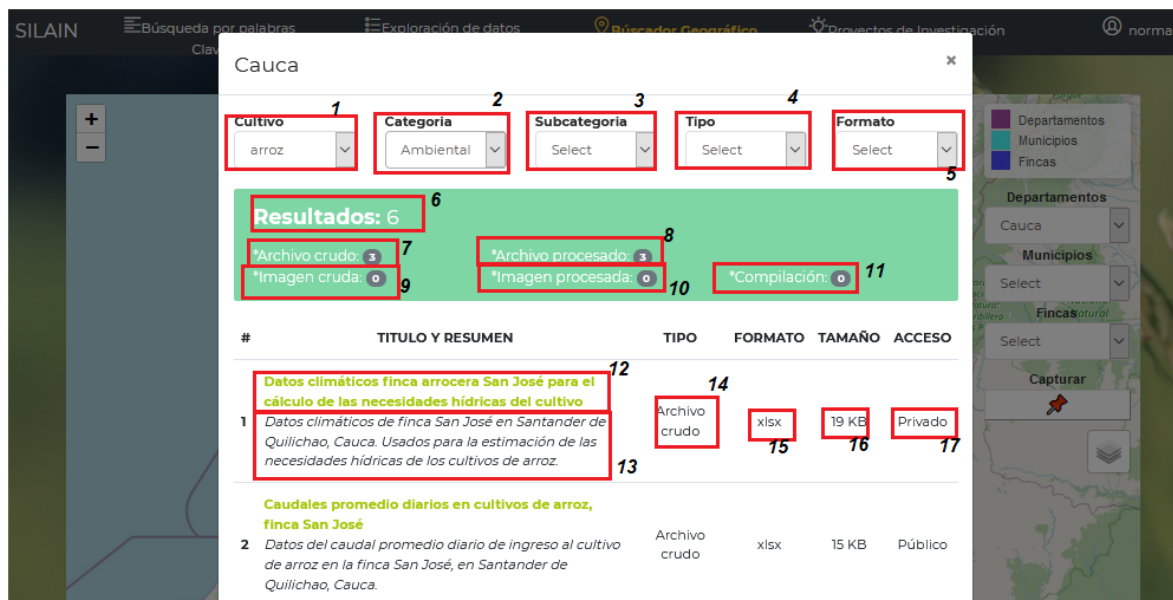


Figura 55. Descripción de resultados en buscador geográfico

- 1- Lista desplegable para filtrar resultados por cultivo
- 2- Lista desplegable para filtrar resultados por categoría
- 3- Lista desplegable para filtrar resultados por subcategoría
- 4- Lista desplegable para filtrar resultados por tipo de dato
- 5- Lista desplegable para filtrar resultados por formato de dato
- 6- Número de datos encontrados tras la búsqueda
- 7- Número de datos encontrados de tipo archivo crudo
- 8- Número de datos encontrados de tipo archivo procesado
- 9- Número de datos encontrados de tipo imagen cruda
- 10- Número de datos encontrados de tipo imagen procesada
- 11- Número de datos encontrados de tipo compilación
- 12- Título del dato
- 13- Resumen del dato
- 14- Tipo del dato
- 15- Formato del dato
- 16- Tamaño del dato
- 17- Tipo de acceso al dato

5.1.3.1.10. Ventana emergente al usar buscar finca más cercana

Fincas			
#	NOMBRE	DISTANCIA EN KM	RESULTADOS POR FINCA
1	Calderón	4.609	
2	Edén	4.782	
3	Eucadia	5.684	
4	San José	23.232	
5	Rozo	41.251	

Figura 56. Descripción de utilidad finca más cercanas

- 1- Nombre de la finca
- 2- Distancia a la finca
- 3- Buscar datos según la finca

5.1.3.1.11. Página de Proyectos de Investigación

SILAIN

Búsqueda por palabras Clave

Exploración de datos

Búscador Geográfico

Proyectos de Investigación

normal

La Información presentada fue suministrada por las personas involucradas

Sustentabilidad ambiental del manejo y uso del agua en cultivos de arroz de riego. caso estudio: finca arrocera san Jose, municipio Santander de Quilichao, Departamento del Cauca

Análisis del efecto del grosor de la lámina de agua en el cultivo de arroz a partir del análisis de respuestas espectrales.

Estimación de las concentraciones de nitrógeno en el cultivo de arroz a partir de datos hiperspectrales

Integración multiescala entre modelos climáticos globales y regionales mediante técnicas de inteligencia artificial

Figura 57. Descripción de Proyectos de Investigación

- 1- Enlace para ir a la página de inicio

- 2- Enlace para ir a la utilidad de Búsqueda por Palabras Clave
- 3- Enlace para ir a la utilidad de Exploración de Datos
- 4- Enlace para ir a la utilidad del Buscador Geográfico
- 5- Enlace para ir a la utilidad de Proyectos de Investigación
- 6- Enlace para ir a iniciar sesión ver panel de usuario (si ya ha iniciado sesión)
- 7- Título del proyecto 1
- 8- Botón para ingresar al proyecto 1
- 9- Título del proyecto 2
- 10-Botón para ingresar al proyecto 2
- 11-Título del proyecto 3
- 12-Botón para ingresar al proyecto 3
- 13-Título del proyecto 4
- 14-Botón para ingresar al proyecto 4

5.1.3.1.12. Página al ingresar a un proyecto de investigación

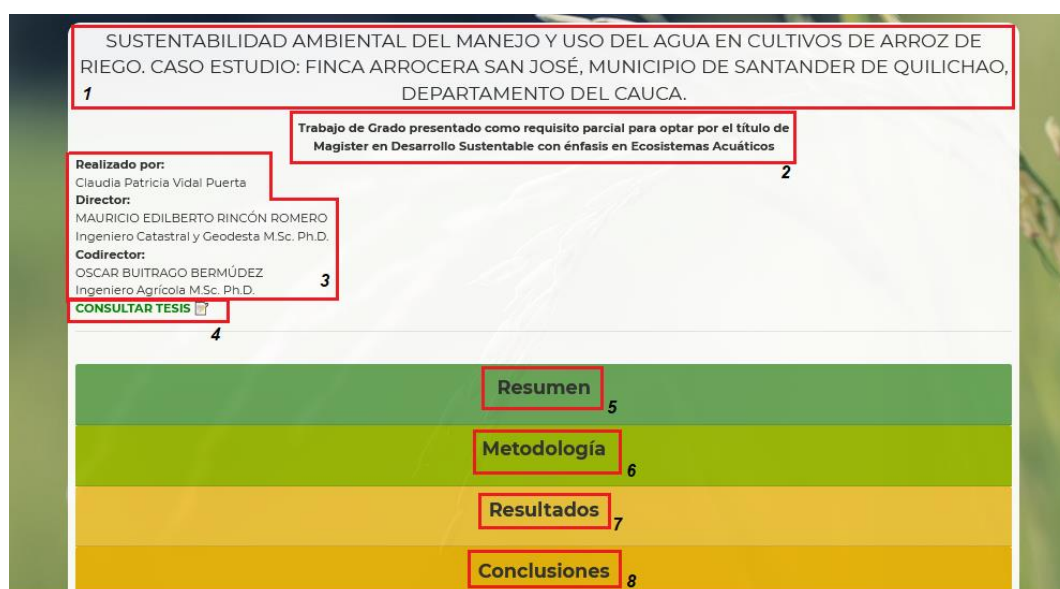


Figura 58. Descripción de ejemplo de proyecto de investigación

- 1- Título del proyecto
- 2- Tipo de proyecto
- 3- Información del autor y directores
- 4- Enlace para consultar el documento completo del proyecto
- 5- Enlace para desplegar resumen
- 6- Enlace para desplegar metodología
- 7- Enlace para desplegar resultados
- 8- Enlace para desplegar conclusiones

5.1.3.1.13. Página al ingresar a un dato

Análisis de los parámetros del agua en finca San José ¹

Departamento: Cauca
Municipio: Santander de Quilichao
Finca: San José

Características

- **Categoría:** Ambiental
- **Subcategoría:** Agua
- **Fecha de creación:** 17/06/2020
- **Fecha de disponibilidad:** 01/06/2020
- **Acceso:** Inmediato

Datos de descarga

- **Tipo:** Archivo procesado
- **Formato:** docx
- **Tamaño:** 323 KB

DESCARGAR ⁴

Identificador: 408 ²

Resumen:
Resultado de análisis de los parámetros de agua superficial en parcela experimental de la Finca arrocera San José en Santander de Quilichao, Cauca. Análisis realiz

Descripción:
Contiene el resultado del análisis de 5 muestras diferentes de agua superficial, se incluyen parámetros fisicoquímicos y microbiológicos.

Mapa ³

Palabras clave ⁵

arroz análisis muestra cropwat huella hídrica agua
fisioquímico microbiológico san José santander de quilichao
cauca

Figura 59. Descripción de página del dato

- 1- Título del dato
- 2- Metadatos
- 3- Mapa con la ubicación de la finca del dato
- 4- Botón para descargar/solicitar descarga del dato
- 5- Palabras clave con enlaces para realizar nuevas búsquedas

5.1.3.1.14. Ventana emergente al descargar un dato

Datos para descarga

Escriba brevemente el motivo por el cual accederá a este dato y que uso le dará a este.

DESCARGAR ²

Si el tamaño del archivo es superior a 1 GB, la descarga podría tardar unos segundos en iniciar.

Figura 60. Descripción del formulario de uso del dato

- 1- Caja de texto para ingresar el motivo de acceso y uso del dato
- 2- Botón para descargar el dato

5.1.3.1.15. Ventana emergente al solicitar acceso a un dato

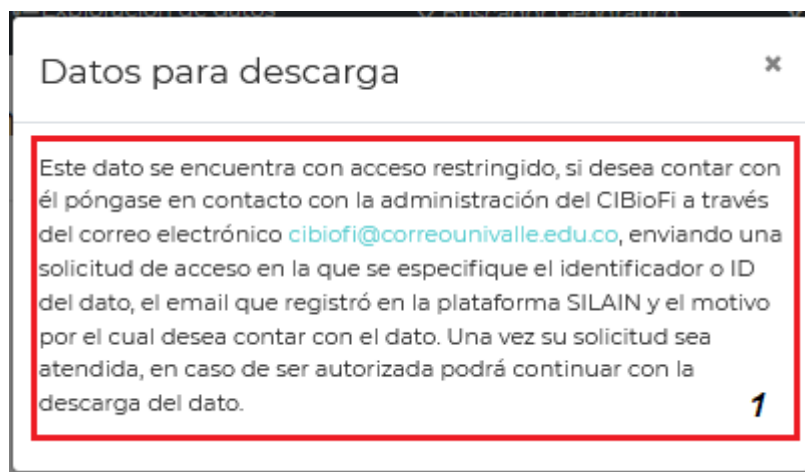


Figura 61. Descripción de solicitud de acceso a dato

- 1- Procedimiento para solicitar acceso a un dato

5.1.3.1.16. Página del panel de usuario (cliente)



Figura 62. Descripción del panel de usuario

- 1- Correo electrónico del usuario
- 2- País del usuario
- 3- Título del dato descargado

- 4- Fecha de la descarga
- 5- Hora de la descarga
- 6- Botón para cerrar sesión

5.1.3.1.17. Página de resultados para el administrador

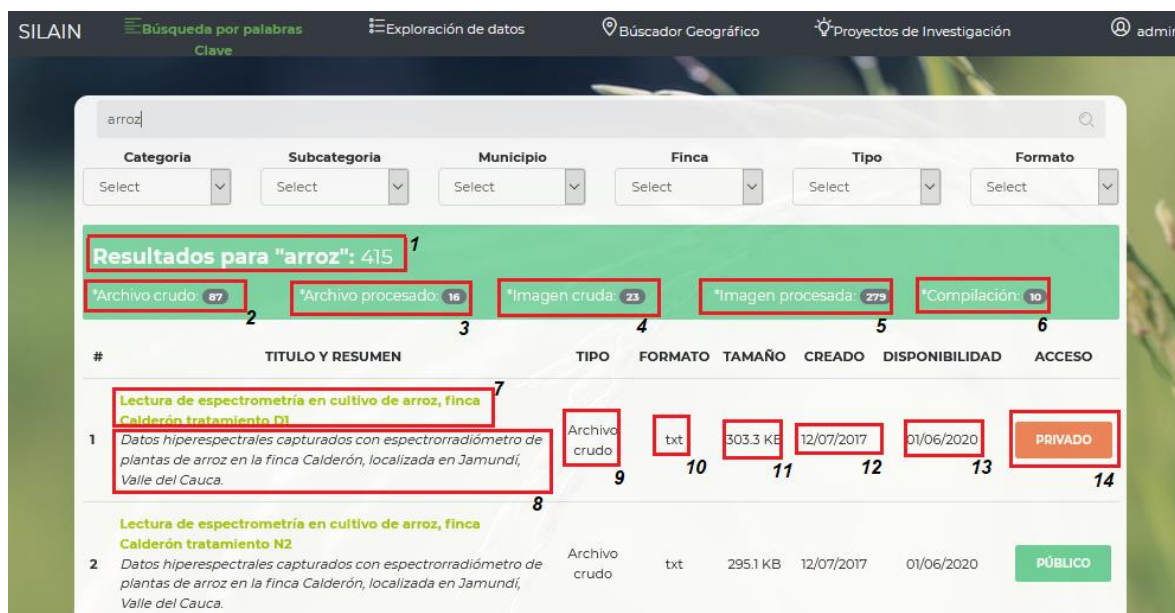


Figura 63. Descripción de resultados para administrador

- 1- Número de datos encontrados tras la búsqueda
- 2- Número de datos encontrados de tipo archivo crudo
- 3- Número de datos encontrados de tipo archivo procesado
- 4- Número de datos encontrados de tipo imagen cruda
- 5- Número de datos encontrados de tipo imagen procesada
- 6- Número de datos encontrados de tipo compilación
- 7- Título del dato
- 8- Resumen del dato
- 9- Tipo del dato
- 10- Formato del dato
- 11- Tamaño del dato
- 12- Fecha de creación del dato
- 13- Fecha de carga del dato a la plataforma
- 14- Botón para cambiar el tipo de acceso al dato

5.1.3.1.18. Página del panel de usuario (administrador)

Bienvenido ADMIN ADMIN

admin@admin.com | Colombia

Carga de datos y metadatos

Categoría Select 1 **Subcategoría** Select 2 **Municipio** Select 3 **Finca** Select 4 **Tipo** Select 5 **Formato** Select 6

Título 7

Resumen 8 **Descripción** 9

Lote 10 **Fase** 11

Palabras Clave 12 **Acceso:** PUBLICO 13

Examinar... No se ha seleccionado ningún archivo. 14

CREATE 15

Figura 64. Descripción de panel de administración

Autorización de descargas

Email del usuario 16 **Id del artículo** 17

CREATE 18

Historial de descargas

No ha hecho ninguna descarga todavía

SALIR 19

Figura 65. Continuación de descripción de panel de administración

- 1- Lista desplegable para asignar categoría al dato
- 2- Lista desplegable para asignar subcategoría al dato
- 3- Lista desplegable para asignar municipio al dato
- 4- Lista desplegable para asignar finca al dato
- 5- Lista desplegable para asignar tipo al dato
- 6- Lista desplegable para asignar formato al dato
- 7- Caja de texto para escribir título del dato
- 8- Caja de texto para escribir resumen del dato
- 9- Caja de texto para escribir descripción del dato
- 10- Caja de texto para escribir lote experimental del dato

- 11- Caja de texto para escribir fase experimental del dato
- 12- Caja de texto para escribir las palabras clave del dato
- 13- Botón para asignar el tipo de acceso al dato
- 14- Botón para ubicar el dato por cargar
- 15- Botón para cargar el dato
- 16- Caja de texto para ingresar el email del usuario por autorizar
- 17- Caja de texto para ingresar el id del metadato por autorizar
- 18- Botón para generar la autorización
- 19- Botón para cerrar sesión

5.1.3.2. Operaciones del sistema

En esta sección se describe detalladamente cada uno de los procedimientos disponibles en la plataforma SILAIN.

5.1.3.2.1. Registro en la plataforma

Para el registro o creación de una cuenta en la plataforma, el usuario debe dirigirse al botón en la parte superior izquierda de la pantalla de inicio (Figura 46). Al hacer clic allí ingresará a la página de inicio de sesión (Figura 47).

En la parte inferior de la pantalla encontrará el enlace *crear una cuenta*, mediante el cual será redirigido a la página de creación de cuenta en la plataforma (Figura 48).

En esta página el usuario deberá diligenciar todos los datos del formulario que se muestra, después al dar clic en *registrar* se indicará si el procedimiento fue exitoso. En caso contrario se deberán rectificar los valores ingresados en el formulario.

5.1.3.2.2. Inicio de sesión

Una vez el usuario ha creado una cuenta en la plataforma, podrá proceder con el inicio de sesión. Si acaba de crearla, en la página de creación de cuenta podrá hacer clic en el enlace llamado *iniciar sesión*. Si el usuario se encuentra en la página de inicio debe ingresar en el botón del extremo superior derecho.

Estando en la página de inicio de sesión (Figura 47), el usuario debe ingresar el email y contraseñas registrados, luego deberá hacer clic en el verificador antibots y finalmente con un clic en *entrar* hará inicio de sesión.

5.1.3.2.3. Uso del Buscador por Palabra Clave

El Buscador por Palabra Clave puede ser utilizado desde la pantalla de inicio (Figura 46) o también desde la página de la utilidad (Figura 49).

En cualquiera de las dos opciones, el usuario ingresará en la barra de búsqueda una palabra clave y hará clic en el botón *buscar* o presionará la tecla *enter*. Con esto se obtendrá un listado de los datos encontrados (Figura 50).

Dependiendo de la cantidad de resultados encontrados, podrán existir varias páginas de resultados a las que se puede ingresar por medio de los números localizados en la parte inferior de la lista (Figura 51).

5.1.3.2.4. Filtrado de resultados en buscador por Palabra Clave

Los resultados obtenidos en el Buscador por Palabra Clave tienen la posibilidad de ser filtrados (Figura 49), haciendo clic en las listas desplegables se mostrarán valores y al seleccionarlos, se mostrará un listado más específico.

Los campos a través de los cuales se puede filtrar los datos son: categoría, subcategoría, municipio, finca, tipo y formato.

5.1.3.2.5. Uso del explorador de datos

En la Exploración de Datos (Figura 52) el usuario podrá obtener resultados sin necesidad de conocer de antemano el tipo de información administrado en la plataforma. Aquí, hará uso únicamente de las listas desplegables y a medida que seleccione los valores de en estas, se generará el listado de datos encontrados.

Los campos de exploración corresponden a los mismos empleados en el filtrado de resultados en el Buscador por Palabra Clave.

5.1.3.2.6. Uso del Buscador Geográfico

Esta utilidad hace uso de un mapa para presentar información espacial relacionada con los datos de investigación (Figura 54).

El usuario encontrará el mapa y en la esquina superior izquierda tendrá un botón para ajustar el nivel de acercamiento. En la parte derecha del mapa están ubicados la leyenda y las listas desplegables que permiten seleccionar un departamento, un municipio o una finca.

Tras la selección de uno de los valores de la lista desplegable, en el mapa se visualizará la entidad correspondiente. Al realizar un clic sobre la entidad se mostrará un botón para buscar datos relacionados con esa ubicación.

Los datos encontrados se mostrarán en una lista (Figura 55) similar a las previamente descritas. Los resultados podrán ser igualmente filtrados de acuerdo con las listas desplegables que se presentan arriba de la lista.

En el buscador geográfico también se cuenta con una utilidad para buscar datos mediante un clic. Esta herramienta se sitúa bajo las listas desplegables. Con un clic sobre el botón, el usuario ubica una posición en el mapa y sobre esta hace clic. Se le mostrará la distancia a las fincas donde se realizaron las investigaciones (Figura 56) y con un nuevo clic en el botón de buscar, se le entregará una lista de resultados.

5.1.3.2.7. Uso de Proyecto de Investigación

La sección de proyectos de investigación permite acceder a una breve reseña de cada una de las cuatro investigaciones realizadas en el proyecto de Agricultura Inteligente (Figura 57).

Al lado derecho de los títulos de los proyectos, hay un botón para acceder a la información. Allí, se encontrará el título del proyecto y el tipo de trabajo al que pertenece. Se mostrará también la información del autor y los directores. Debajo de esto, habrá un enlace para consultar el documento completo del trabajo.

Se podrá encontrar el resumen, una breve muestra de la metodología, los resultados y las conclusiones del trabajo (Figura 58). Mediante un clic en una de estas secciones mencionadas, se desplegará la información que corresponde.

En la sección de metodología de cada proyecto se presentarán los enlaces a algunos de los datos relacionados con el proyecto en cuestión.

5.1.3.2.8. Descarga de un dato

Cuando el usuario ingresa a uno de los resultados de la lista, será redirigido a los metadatos del dato (Figura 59). Además de los metadatos, verá un mapa para visualizar el polígono de la finca a la que pertenece el dato.

Un botón de descarga estará disponible para proceder con la obtención del dato. Al hacer clic sobre este se abrirá una ventana emergente en la que el usuario deberá informar cuál es el motivo de la descarga del dato y qué uso le dará a este (Figura 60). Después de llenar este requisito podrá hacer clic en *descargar* y podrá contar con el dato.

5.1.3.2.9. Solicitud de acceso a un dato

Algunos datos en la plataforma podrán contar con restricciones a su acceso, por lo que el botón *descargar* en los metadatos será remplazado por uno que dice *solicitar descarga*. Al hacer clic sobre este una ventana emergente aparecerá con la información sobre cómo realizar la solicitud del dato deseado (Figura 61).

5.1.3.2.10. Historial de descargas de datos

Las descargas realizadas por un usuario son registradas en la base de datos del sistema. El usuario puede acceder al historial de descargas en el panel de usuario (Figura 62) que se encuentra ubicado en la parte superior derecha de la pantalla, corresponde al mismo enlace usado para iniciar sesión.

En este historial se muestran los títulos de los datos descargados, así como la fecha y la hora en que se realizó la descarga. El título actúa como enlace para poder acceder nuevamente al dato.

5.1.3.2.11. Carga de datos y metadatos

La carga de datos nuevos a la plataforma es una operación exclusiva del administrador. Cuando se inicia sesión como administrador, en el panel de usuario se encontrará el formulario titulado *Carga de datos y metadatos*. Este formulario se compone de listas desplegables y cajas de texto, las cuales deben ser diligenciadas para luego establecer el tipo de acceso que tendrá el dato, después ubicar el archivo que contiene el dato y finalizar con un clic sobre el botón *create*.

5.1.3.2.12. Autorización de descargas

Otra acción únicamente disponible para el administrador es la autorización de descargas, esta operación se localiza bajo el formulario para cargar datos (Figura 65).

Consta de dos cajas de texto, una para ingresar el correo electrónico del usuario que se pretende autorizar y el identificador (id) del metadato. Luego de ingresar estos dos valores, con un clic en *create* se genera la autorización y el usuario que solicitó acceso a la descarga, podrá realizarla sin restricción alguna.

5.1.3.2.13. Cambio del tipo de acceso a un dato

El administrador es el único usuario con el privilegio cambiar el tipo de acceso a un dato. Puede modificar un acceso restringido a uno que no requiera autorización. Igualmente, podrá restringir el acceso a un dato que anteriormente se podía descargar de forma inmediata.

Este cambio es realizable a través de los listados de resultados que se generan, ya sea en el Buscador por Palabra Clave, el Explorador de Datos o el Buscador geográfico.

En la Figura 63 se muestra una columna adicional en el listado de resultados, esta contiene botones para cada uno de los datos encontrados y mediante un clic sobre estos se producirá al instante el cambio en el tipo de acceso.

5.1.3.2.14. Cierre de sesión

El cierre de sesión, tanto para el administrador como para el usuario cliente, se encuentra en el panel de usuario. En la parte inferior de este panel se hallará un botón denominado *salir*. Con un clic sobre este se cerrará la sesión en la plataforma SILAIN (Figuras 62 y 65).

CAPÍTULO 6

6.1. BIBLIOGRAFÍA

Alam, A. W., Müller, S., & Schumann, N. (2015). datorium: Sharing Platform for Social Science Data. Inventing Information Science in the Networked Society. Proceedings of the 14th International Symposium on Information Science (ISI 2015), Zadar, Croatia, 19-21 de mayo 2015, 244–249.

Amodeo, E. (2013). Principios de diseño de APIs REST.

Androulakis, S., Buckle, A. M., Cook, J., Groenewegen, D., Nicholas, N., Affiliates, L., Treloar, A., & Centre, M. (2009). The International Journal of Digital Curation ARCHER – e-Research Tools for Research Data Management. 4(1), 22–33.

Baca, M., & Getty Research Institute. (2008). Introduction to metadata.

Ball, A. (2010). Preservation and Curation in Institutional Repositories. B5.2, 1–65. Recuperado de: <http://dcc.ac.uk/resources/briefing-papers/technology-watch-papers/curation-repositories>

Baraniuk, R. G. (2011). More is less: Signal processing and the data deluge. Science, 331(6018), 717–719.

Barrera, M. G., & Libaque, I. (2017). Nodo integrador de la información geográfica del Agro: Geoportal IDE MINAGRI de Chile. 185–191.

Basañes, F. (2019). Código abierto: conceptos y aplicaciones. El Banco Interamericano de Desarrollo. Recuperado de: <https://blogs.iadb.org/conocimiento-abierto/es/codigo-abierto/>

Basso, L., Pascale Medina, C., S. de Obschatko, E., & Preciado Patiño, J. (2013). Agricultura Inteligente, La iniciativa de la Argentina para la sustentabilidad en la producción de alimentos y energía (1a ed.).

Berg, K. L., Seymour, T., & Goel, R. (2013). History of databases. International Journal of Management & Information Systems, 17(1), 29–36.

Blaiszik, B., Chard, K., Pruyne, J., Ananthakrishnan, R., Tuecke, S., & Foster, I. (2016). The Materials Data Facility: Data Services to Advance Materials Science Research. Jom, 68(8), 2045–2052.

Bordogna, G., Kliment, T., Frigerio, L., Brivio, P., Crema, A., Stroppiana, D., Boschetti, M., & Sterlacchini, S. (2016). A Spatial Data Infrastructure Integrating Multisource Heterogeneous Geospatial Data and Time Series: A Study Case in Agriculture. *ISPRS International Journal of Geo-Information*, 5(5), 73.

Borgman, C. (2012). The Conundrum of Sharing Research Data. *Journal of the American Society for Information Science & Technology*.

Borgman, C. L. (2015). *Big Data, Little Data, No Data: Scholarship in the Networked World*. MIT Press.

Bourgeois, D. (2014). *Information Systems for Business and Beyond*. *Information Systems for Business and Beyond*, 1–167.

Candan, K. S., Liu, H., & Suvarna, R. (2001). Resource Description Framework: Metadata and Its Applications. *SIGKDD Explorations*, 3(1), 6–19.

Candiago, S., Remondino, F., Giglio, M. De, Dubbini, M., & Gattelli, M. (2015). Evaluating Multispectral Images and Vegetation Indices for Precision Farming Applications from UAV Images. *Vi*, 4026–4047.

Chalk, S. J. (2016). The Open Spectral Database: An open platform for sharing and searching spectral data. *Journal of Cheminformatics*, 8(1), 1–14.

Chua, A. (2007, April). The curse of success. *The Wall Street Journal*.

Congreso de la República de Colombia. (2014). Ley 1712 de 2014. Recuperado de: http://wsp.presidencia.gov.co/Normativa/Leyes/Documents/LEY_1712_DEL_06_DE_MARZO_DE_2014.pdf

Cooper, P. (2017). Data, information, knowledge and wisdom. *Anaesthesia and Intensive Care Medicine*, 18(1), 55–56.

Corporación Mozilla. (2020). Generalidades del protocolo HTTP. Recuperado de: <https://developer.mozilla.org/es/docs/Web/HTTP/Overview>

Craig, H. (2019). GARDIAN: The data search tool unlocking agricultural development innovation. <https://bigdata.cgiar.org/blog-post/gardian-data-search-unlocking-agricultural-development-innovation/>

Creative Tim. (2018). Paper Dashboard React. Recuperado de: <https://www.creative-tim.com/product/paper-dashboard-react>

Culina, A., Crowther, T. W., Ramakers, J. J. C., Gienapp, P., & Visser, M. E. (2018). How to do meta-analysis of open datasets. *Nature Ecology and Evolution*, 2(7), 1053–1056.

Etuk, A., Shaw, F., Gonzalez-Beltran, A., Johnson, D., Laporte, M.-A., Rocca-Serra, P., Arnaud, E., Devare, M., Kersey, P. J., Sansone, S., & Davey, R. P. (2019). COPO: a metadata platform for brokering FAIR data in the life sciences. *BioRxiv*.

Fensel, D., Lausen, H., Polleres, A., De Bruijn, J., Stollberg, M., Roman, D., & Domingue, J. (2006). *Enabling semantic web services the web service modeling ontology*. Springer Science & Business Media.

Frederick, D. E. (2016). Data, Open Science and libraries – The Data Deluge Column. *Library Hi Tech News*, 33(8), 11–16.

Giménez, A. (2018). Aplicaciones web con front-end y back-end desacoplado. Recuperado de: <https://www.aunitz.net/aplicaciones-web-front-end-back-end-desacoplado/>

Godfray, H. C. J., Beddington, J. R., Crute, I. R., Haddad, L., Lawrence, D., Muir, J. F., Pretty, J., Robinson, S., Thomas, S. M., & Toulmin, C. (2010). Food security: The challenge of feeding 9 billion people. In *Science* (Vol. 327, Issue 5967, pp. 812–818). American Association for the Advancement of Science.

González-Longatt, & Francisco. (2007). *Introducción a los Sistemas de Información: Fundamentos Sistemas de Información*. Recuperado de: <https://www.uv.mx/personal/artulopez/files/2012/08/FundamentosSistemasInformacion.pdf>

Hanson, B., Sugden, A., & Alberts, B. (2011). Making data maximally available. *Science*, 331(6018), 649.

Harrison, Guy. (2015). *The Lessons of Database History*. Julio.

Hey, T., & Trefethen, A. (2003). The Data Deluge: An e-Science Perspective. *Grid Computing – Making the Global Infrastructure a Reality*, enero 2003, 1–17.

Hillman, D. (2005). *DCMI: Using Dublin Core*. Recuperado de: <https://www.dublincore.org/specifications/dublin-core/usageguide/>

Hudson-Vitale, C., Imker, H., Johnston, L. R., Carlson, J., Kozlowski, W., Olendorf, R., Stewrt, C., & Stewart, C. (2017). *SPEC Kit 354 Data Curation*. Mayo 2017.

ICONTEC. (2011). *NORMA TÉCNICA NTC COLOMBIANA 4611*. Recuperado de: <https://tienda.icontec.org/wp-content/uploads/pdfs/NTC4611.pdf>

IGAC. (s.f.). Glosario Instituto Geográfico Agustín Codazzi. Recuperado de https://www.igac.gov.co/es/contenido/glosario#_g

Lapiedra, R., Devece, C., & Guiral, J. (2011). Introducción a la gestión de sistemas de información en la empresa. Universitat Jaume I. Servei de Comunicació i Publicacions.

Laudon, K., Laudon, J., & Brabston, M. (2012). Management information systems: Managing the digital firm. Prentice Hall.

Luján, S. (2002). Programación de aplicaciones web: historia, principios básicos y clientes web. E. C. Universitario.

Mansour, M., Blvd, J. C. W., & Blvd, J. C. W. (2017). History of Database Applications. 1(47), 1–3.

McGregor, J., & Hamm, S. (2008). Managing the global workforce. BusinessWeek.

Microsoft. (2020). Descripción de los conceptos básicos de la normalización de bases de datos. Documentación de Microsoft. Recuperado de: <https://docs.microsoft.com/es-es/office/troubleshoot/access/database-normalization-description>

Murdick, R. (1989). Sistemas de Información. Prentice Hall Latinoamérica.

Npm Inc. (s.f.). About npm. Recuperado de: <https://docs.npmjs.com/about-npm>

ONU. (2017). Volume I: World Population Prospects. World Population Prospects The 2017 Revision, I, 1–377. Recuperado de: https://esa.un.org/unpd/wpp/Publications/Files/WPP2017_Volume-I_Comprehensive-Tables.pdf

Open Knowledge Foundation. (s.f.). OPEN DATA HANDBOOK. Recuperado de: <http://opendatahandbook.org/guide/en/what-is-open-data/>

Pascual, A. F. R., Power, P. A., Romero, E. L., & Maganto, A. S. (2014). La Norma ISO 19115 de Metadatos : Características Y Aplicabilidad. Julio.

Pasquetto, I. V., Borgman, C. L., & Wofford, M. F. (2019). Uses and Reuses of Scientific Data: The Data Creators' Advantage. Harvard Data Science Review, 1(2), 1–35.

Red Hat. (s.f.). ¿Qué es el open source?. Recuperado de: <https://www.redhat.com/es/topics/open-source/what-is-open-source>

Red Hat. (s.f.). ¿Qué es una API de REST?. Recuperado de: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>

Richardson, L., & Ruby, S. (2008). RESTful web services. O'Reilly Media, Inc.

Ruiz, A. (2014). Red de Información y Comunicación del Sector Agropecuario Colombiano - Agronet.

Scott, M., Boardman, R. P., Reed, P. A., Austin, T., Johnston, S. J., Takeda, K., & Cox, S. J. (2014). A framework for user driven data management. *Information Systems*, 42, 36–58.

Seneviratne, V. (2017). IAEA team finds Canada plants using comprehensive KM practices. *Inside NRC*, 16.

Silberschatz, A., Korth, H. F. & Sudarshan, S. (2002). Fundamentos de bases de datos.

Sistema General de Regalías (2014). ACUERDO No. 27 DE 05/08/2014. Por medio del cual se adoptan decisiones relacionadas con proyectos de inversión financiados o cofinanciados con recursos del Sistema General de Regalías - SGR.

Stair, R., & Reynolds, G. (2010). Principios de sistemas de información (9na ed.).

Tenopir, C., Allard, S., Douglass, K., Aydinoglu, A. U., Wu, L., Read, E., Manoff, M., & Frame, M. (2011). Data sharing by scientists: Practices and perceptions. *PLoS ONE*, 6(6), 1–21.

Thomasson, J. A., & Cox, M. S. (2013). Soil reflectance sensing for determining soil properties in precision agriculture. January 2001.

Tremouilhac, P., Lin, C.-L., Huang, P.-C., Huang, Y.-C., Nguyen, A., Jung, N., Bach, F., Ulrich, R., Neumair, B., Streit, A., & Bräse, S. (2020). The Repository Chemotion: Infrastructure for Sustainable Research in Chemistry.

Valacich, J., Schneider, C., & Jessup, L. (2016). Information systems today: Managing in the digital world.

Van der Kwast, J. (2017). Open Source Software and Open Data. Recuperado de: <https://www.youtube.com/watch?v=s3uYGoazjFw>

Vicerrectoría de Investigaciones. (2016). Portafolio de Servicios de CIBioFi.

Wagenaar, J. B., Brinkmann, B. H., Ives, Z., Worrell, A., Litt, B., & Member, S. (2013). A Multimodal Platform for Cloud - based Collaborative Research. 2013 6th International IEEE/EMBS Conference on Neural Engineering (NER), 1386–1389.

Whitlock, M. C. (2011). Data archiving in ecology and evolution: Best practices. *Trends in Ecology and Evolution*, 26(2), 61–65.

Wilkinson, M. D., Dumontier, M., Aalbersberg, Ij. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., ... Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*.

Wood, D. (2010). Linking enterprise data. *Linking Enterprise Data*, 1–291.