

	UNIVERSIDAD FRANCISCO DE PAULA SANTANDER OCAÑA			
	Documento	Código	Fecha	Revisión
	FORMATO HOJA DE RESUMEN PARA TRABAJO DE GRADO	F-AC-DBL-007	02-03-2023	B
	Dependencia	Aprobado	Pág.	
DIVISIÓN DE BIBLIOTECA	SUBDIRECTOR ACADEMICO	1(92)		

RESUMEN – TRABAJO DE GRADO

AUTORES	Elver Quintero Ayala		
FACULTAD	Ingenierías		
PLAN DE ESTUDIOS	Ingeniería De Sistemas		
DIRECTOR	Juan Camilo Jaimes Fernández		
TÍTULO DE LA TESIS	Desarrollo De Un Geoportal En El Centro De Innovación Y Tecnológica (Cedit) De La Universidad Francisco De Paula Santander Ocaña		
TITULO EN INGLES	Development of a Geoportal at the Center for Innovation and Technology (Cedit) of the Francisco de Paula Santander Ocaña University		
RESUMEN			
<p>El presente documento contiene el desarrollo de un Geoportal en la ciudad de Ocaña Norte Santander y el Catatumbo, en el Centro de Innovación Tecnológica (CEDIT) de la Universidad Francisco de Paula Santander Ocaña el cual permite que los estudiantes y personas que quieran realizar diferentes búsquedas de ubicaciones y puntos de georreferenciación en el Geoportal puedan hacerlo y obtener información de forma rápida y segura, permitiendo un mejor aprendizaje.</p>			
RESUMEN EN INGLES			
<p>This document contains the development of a Geoportal in the city of Ocaña Norte Santander and Catatumbo, at the Center for Technological Innovation (CEDIT) of the Francisco de Paula Santander Ocaña University, which allows students and people who want to carry out different searches of locations and georeferencing points in the Geoportal can do so and obtain information quickly and safely, following a better learning.</p>			
PALABRAS CLAVES	Geocatatumbo, Geovisor, Georreferenciación, Portal.		
PALABRAS CLAVES EN INGLES	Geocatatumbo, Geovisor, Georreferencing, Portal.		
CARACTERÍSTICAS			
PÁGINAS: 92	PLANOS:	ILUSTRACIONES: 73	CD-ROM:



**Desarrollo De Un Geoportal En El Centro De Innovación Y Tecnológica (Cedit) De La
Universidad Francisco De Paula Santander Ocaña**

Elver Quintero Ayala

Facultad de Ingenierías, Universidad Francisco de Paula Santander Ocaña

Ingeniería de Sistemas

Ing. Juan Camilo Jaimes Fernández

10 febrero de 2023

Índice

Resumen.....	9
Introducción	10
1. Desarrollo de un Geoportal en el Centro de Innovación y Tecnológica (CEDIT) de la Universidad Francisco de Paula Santander Ocaña	11
1.1 Descripción breve de la Universidad Francisco de Paula Santander Ocaña.	11
1.1.1 Misión.....	11
1.1.2 Visión	11
1.1.3 Objetivos de la Empresa.....	12
1.1.4 Descripción de la estructura organizacional	13
1.1.5 Descripción de la dependencia y/o proyecto asignado.	15
1.2 Diagnóstico inicial de la dependencia asignada.	15
1.2.1 Planteamiento del problema.	17
1.3 Objetivos de la pasantía.....	19
1.3.1 General.	19
1.3.2 Específicos	19
1.4 Descripción de las actividades a desarrollar en el CEDIT	20
2. Enfoques referenciales	22
2.1 Enfoque conceptual.	22
2.2 Enfoque legal.....	25

3. Informe de cumplimiento de trabajo	29
3.1 Presentación de resultados.....	29
3.1.1 Objetivo 1. Analizar el uso de nuevas tecnologías para el desarrollo del Geo portal Ocaña y descifrar qué instrumentos deben implementarse para su funcionalidad.	29
3.1.2 Objetivo 2. Desarrollar funcionalidades en el Geoportal utilizando diferentes lenguajes de programación y patrones de diseño de Software.....	49
3.1.3 Objetivo 3. Realizar pruebas de los cambios implementados en el Geoportal que permitan verificar el correcto funcionamiento del mismo.	70
4. Diagnóstico final.	83
5. Conclusiones.	84
6. Recomendaciones.....	85
Referencias	86
Apéndice.....	90
Apéndice A Reunión inicio proyecto	90
Apéndice B Reunión para realizar los mock ups.....	90
Apéndice C Constancia continuidad del proyecto Geocatambo.	91

Lista de Figuras

Figura 1. Estructura organizacional de la Universidad Francisco de Paula Santander Ocaña.	13
Figura 2. Organigrama de la Universidad Francisco de Paula Santander Ocaña.....	14
Figura 3. Árbol de problemas.....	19
Figura 4. Carpeta Controllers.....	31
Figura 5. Carpeta Middleware.....	32
Figura 6. Carpeta Imports.	33
Figura 7. Carpeta Models.....	34
Figura 8. Carpeta Config.....	35
Figura 9. Carpeta Database.	36
Figura 10. Carpeta Módulos.....	37
Figura 11. Carpeta Public.....	38
Figura 12. Carpeta Views.....	39
Figura 13. Carpeta Layouts.....	40
Figura 14. Carpeta Routes.....	41
Figura 15. Archivo.env.	42
Figura 16. Mockup pantalla inicio Geoportal.	43
Figura 17. Mockup Menú Accesibilidad Geoportal.....	43
Figura 18. Mockup Pantalla de Carga.....	44
Figura 19. Mockup Geovisor del Geoportal.....	44
Figura 20. Mockup Tipos de Capas Geoportal.	45
Figura 21. Modelo relacional.	46
Figura 22. Modelado de tablas en postgres.....	48

Figura 23. Creación del login y registro en Laravel.....	50
Figura 24. Formulario Loguin.....	50
Figura 25. Formulario de Registro.....	51
Figura 26. Patrón de diseño llamado Front Controller.....	52
Figura 27. Patrón de diseño Symple Factory.	53
Figura 28. Patrón de diseño Strategy.	54
Figura 29. Llamado mediante el id a la clase agrícola.....	54
Figura 30. Patrón de diseño Observer.....	55
Figura 31. Funciones crear, editar y eliminar.	55
Figura 32. Rutas en el archivo web.php.....	56
Figura 33. Componentes.	57
Figura 34. Navbar o banner.....	58
Figura 35. Barra Menú.....	58
Figura 36. Tarjetas con Municipios.	59
Figura 37. Preload (Imagen de carga).....	60
Figura 38. Geovisor con Mapa Ocaña, descripción y puntos de georreferenciación.....	60
Figura 39. Geovisor con Mapa Abrego y descripción de los Barrios.	61
Figura 40. Menú Lateral Derecho.....	61
Figura 41. Botón Accesibilidad.....	62
Figura 42. Logos pie página.....	62
Figura 43. Footer (Pie de página).....	63
Figura 44. Sobre Geocatatumbo.....	63
Figura 45. Formulario de Loguin.....	64

Figura 46. Formulario de Registro.....	64
Figura 47. Formulario de Recuperación de contraseña.....	65
Figura 48. Creación Panel administración.....	65
Figura 49. Panel administración.....	66
Figura 50. Tabla Arbolado Urbano.....	67
Figura 51. Formulario para Crear Datos.....	67
Figura 52. Formulario para Editar Datos.....	68
Figura 53. Formulario para Eliminar Datos.....	69
Figura 54. Formulario Registro Masivo.....	69
Figura 55. Selección Archivo Registro Masivo.....	70
Figura 56. Directorio Feature y Unit.....	72
Figura 57. Test o prueba.....	73
Figura 58. Ruta /usuarios.....	74
Figura 59. Cadena de texto en pantalla.....	74
Figura 60. Error al ejecutar el Test.....	75
Figura 61. Métodos crear usuario.....	76
Figura 62. Usuarios creados en la base de datos.....	76
Figura 63. Usuarios en pantalla.....	77
Figura 64. Test usuarios creados.....	77
Figura 65. Usuarios eliminados.....	78
Figura 66. Búsqueda de usuarios.....	78
Figura 67. Líneas de código con usuarios comentadas.....	79
Figura 68. Base de datos dedicada.....	80

Figura 69. Geoportal sin Menú lateral derecho.....	81
Figura 70. Creación de ruta para crear backup en postgres.	82
Figura 71. Reunión inicio proyecto.....	90
Figura 72. Reunión para realizar los mockups.	90
Figura 73. Constancia de continuidad del proyecto geo Catatumbo.....	91

Lista de tablas

Tabla 1. Matriz DOFA	14
Tabla 2. Descripción de las actividades a desarrollar en el CEDIT	15

Resumen

Para la realización de este proyecto se buscó información sobre el Geoportal de Medellín el cual brinda diferentes tipos de georreferenciación, como hospitales, colegios, sitios de accidentes entre otros. Además de esto permite realizar capas de un mapa específico y subirlas al servidor, permitiendo visualizar en tiempo real los cambios realizados.

Estos estudios y la necesidad de un Geoportal en la ciudad de Ocaña Norte Santander y el Catatumbo llevaron a que se desarrollara el proyecto de geocatatumbo en el Centro de Innovación y Tecnológica (CEDIT) de la Universidad Francisco de Paula Santander Ocaña el cual permite que los estudiantes y personas que quieran realizar diferentes búsquedas de ubicaciones y puntos de georreferenciación en el Geoportal puedan hacerlo y obtener información de forma rápida y segura.

Para el desarrollo de este proyecto se utilizó la herramienta Arcgis y el Framework Laravel, Arcgis es una plataforma que ofrece la posibilidad de realizar mapas y subirlos a Arcgis online los cuales se modifican y se visualizan en tiempo real. En Laravel se realizó la interfaz del Geoportal en la que el usuario interactúa con el Geoportal y este a su vez hace un consumo de Arcgis online para brindar la información al usuario.

Introducción

En la actualidad, la tecnología se ha vuelto una herramienta necesaria para facilitar las tareas que se realizan a diario. En caso de realizar búsquedas sobre sitios específicos se acude a Google maps, pero si se quiere realizar una búsqueda en específico como es el lugar donde se encuentra ubicado un sitio u objeto con sus características y especificaciones no se encuentra información exacta. Otra opción es realizar búsquedas en internet en la que se encuentran diferentes tipos de respuestas, pero estas pueden llegar a contener información falsa que no es de fuentes confiables.

La búsqueda de lugares y puntos específicos se ha vuelto muy importante para los diferentes tipos de empresas y usuarios los cuales necesitan esta información. Es por esto que se decide crear el Geoportal para incluir estos tipos de servicios los cuales brindan a la población búsquedas de forma rápida y exacta, para aportar a la educación y a la población de Norte Santander la opción de tener un sitio web donde buscar este tipo de información el cual es obtenido de fuentes confiables y con información actualizada.

1. Desarrollo de un Geoportal en el Centro de Innovación y Tecnológica (CEDIT) de la Universidad Francisco de Paula Santander Ocaña

1.1 Descripción breve de la Universidad Francisco de Paula Santander Ocaña.

La Universidad Francisco de Paula Santander Ocaña (UFPSO) es una institución de educación superior de carácter oficial, la cual fue formalizada por el Congreso de la República de Colombia mediante la Ley 67 del 26 de diciembre de 1968, estableciendo la juridicidad y carácter de la universidad. A día de hoy, la UFPSO (sede central y seccional Ocaña) es reconocida como una Institución que enseña, investiga y transfiere sus conocimientos a la sociedad, las empresas y las entidades estatales, proyectando su quehacer hacia el ámbito global en coherencia con las tendencias actuales y las transformaciones necesarias para asegurar su sostenibilidad.

1.1.1 Misión.

La Universidad Francisco de Paula Santander Ocaña, institución pública de educación superior, es una comunidad de aprendizaje y autoevaluación en mejoramiento continuo, comprometida con la formación de profesionales idóneos en las áreas del conocimiento, a través de estrategias pedagógicas innovadoras y el uso de las tecnologías; contribuyendo al desarrollo nacional e internacional con pertinencia y responsabilidad social.

1.1.2 Visión

En el año 2025, seremos una Universidad acreditada de alta calidad, reconocida por la excelencia y eficiencia en el ejercicio de las funciones misionales con enfoque global, situando en valor las potencialidades de la comunidad universitaria y participando en los cambios del entorno mediante la transferencia del conocimiento y la innovación; aportando al desarrollo sostenible de

la sociedad.

1.1.3 Objetivos de la Empresa

Fortalecimiento de la cultura de la autoevaluación y aseguramiento de la calidad académica.

Comprende lo relacionado con el desarrollo docente para la excelencia académica; las actividades del Sistema Interno de Aseguramiento de la calidad base para la acreditación institucional y de programas académicos; la consolidación de las actividades de visibilidad, internacionalización y bilingüismo; y la virtualización e innovación de los programas académicos de cara al establecimiento de un campus virtual.

Gestión estudiantil pertinente y con calidad.

Comprende todo lo relacionado con la promoción de los servicios académicos; gestión de programas para mejorar las habilidades de los estudiantes y permitir resultados de aprendizaje; y mejorar la entrega del aprendizaje a través de estrategias locales con objetivos nacionales e internacionales.

Desarrollo sostenible institucional.

Modernizar la Universidad en cuanto a estructura, arquitectura de procesos y sistemas de información; acciones estratégicas para el desarrollo sostenible del Campus Universitario y la gestión del ciclo del talento como pilar de futuro de la institución.

Investigación y extensión con proyección global.

Consolidar la producción científica, fortalecer procesos de divulgación con pertinencia e impacto social, y desarrollar procesos de innovación, emprendimiento y transferencia de tecnología que beneficien a las instituciones y sus grupos de valor.

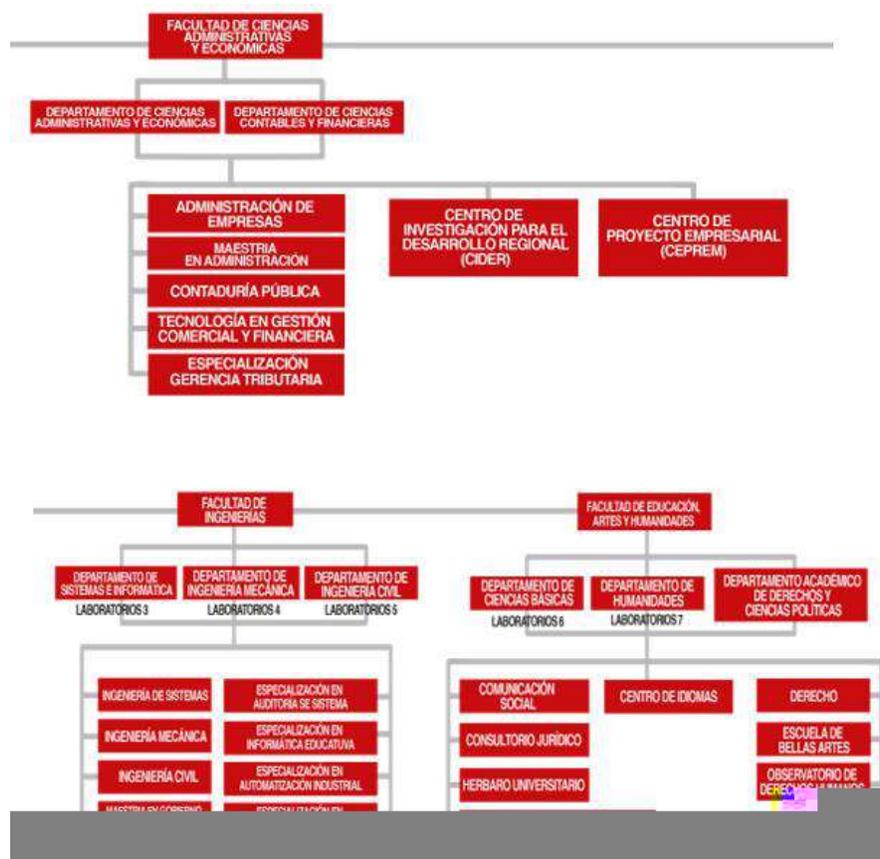
Bienestar universitario y responsabilidad social.

Potenciar los servicios y consolidar procesos saludables que favorezcan el clima y ambiente organizacional. Así mismo, articula los esfuerzos de la Universidad para cumplir con sus responsabilidades sociales, con especial énfasis en la educación inclusiva.

1.1.4 Descripción de la estructura organizacional

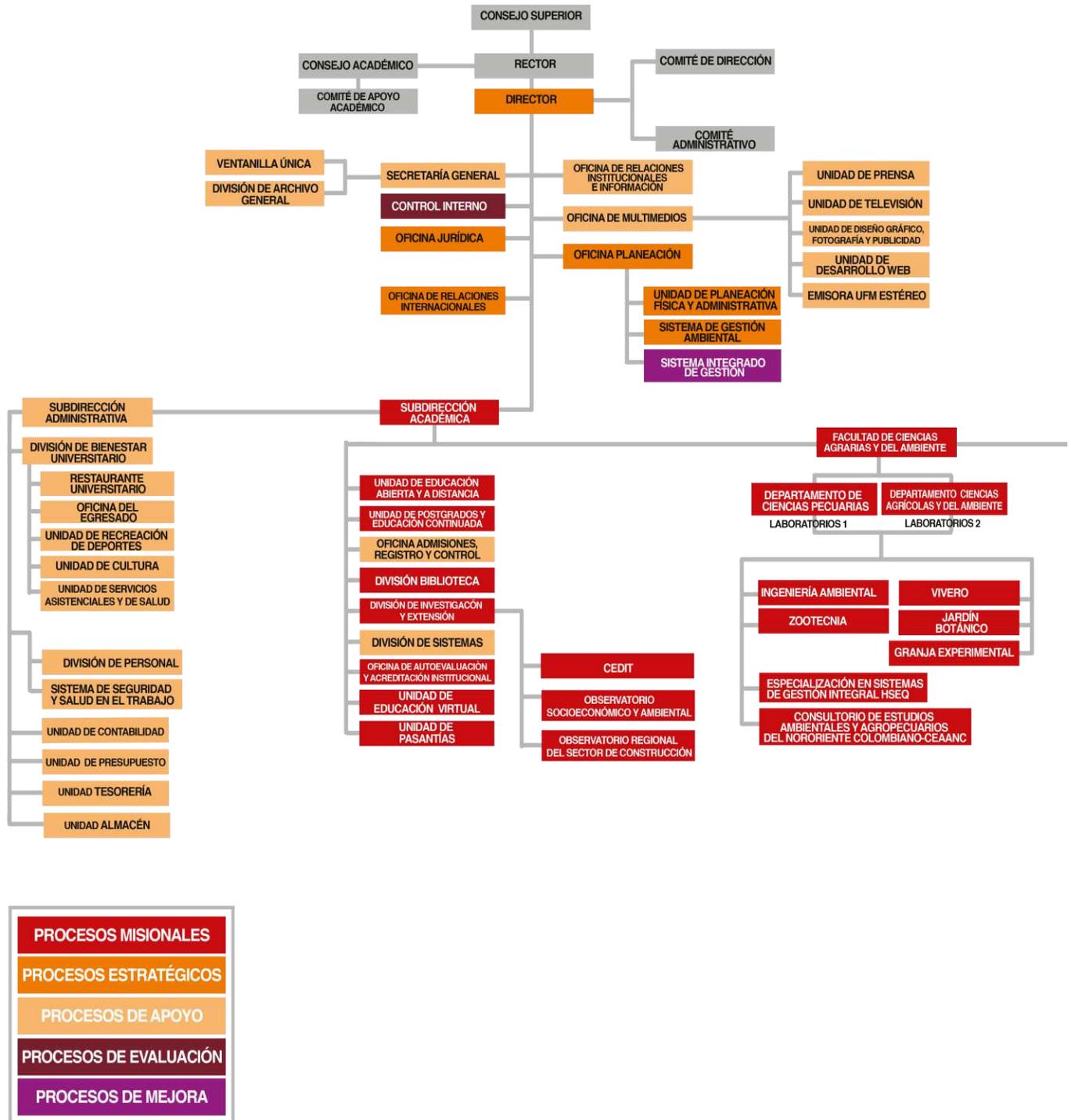
La Universidad Francisco de Paula Santander Ocaña cuenta con la Facultad de Ciencias Administrativas y Económicas, la Facultad de Ingenierías y con la Facultad de Educación Artes y Humanidades. Cómo se observa en la Ilustración 2.

Figura 1. Estructura organizacional de la Universidad Francisco de Paula Santander Ocaña.



Fuente: <https://ufpso.edu.co/Estructura>

Figura 2. Organigrama de la Universidad Francisco de Paula Santander Ocaña.



Fuente: <https://ufpso.edu.co/Estructura>

1.1.5 Descripción de la dependencia y/o proyecto asignado.

La dependencia en la cual se realizará el trabajo será el Centro de Desarrollo e Innovación Tecnológica (CEDIT) de la Universidad Francisco de Paula Santander Ocaña, en la cual se dará inicio al proyecto GeoPortal Ocaña.

1.1.5.1 Dependencia.

Centro de Desarrollo e Innovación Tecnológica (CEDIT) adscrito a la División de Investigación y Extensión DIE de la Universidad Francisco de Paula Seccional Ocaña es un centro que fomenta la investigación, el desarrollo tecnológico, la innovación, el emprendimiento digital en todos los contextos académicos y de la región en la solución de problemas del entorno a través del uso de las Tecnológicas de la Información y las Comunicaciones TIC y en la generación del conocimiento.

1.2 Diagnóstico inicial de la dependencia asignada.

Conocer la situación inicial en la que se encuentra el Centro de Desarrollo e Innovación Tecnológica (CEDIT) de la Universidad Francisco de Paula Santander Ocaña es fundamental para la ejecución del proyecto GeoPortal Ocaña. En la tabla 1 se puede observar la matriz DOFA, que es una herramienta que ayuda a identificar las Debilidades, las Oportunidades, las Fortalezas y las Amenazas presentes en el CEDIT.

Tabla 1. Matriz DOFA

	Fortalezas	Debilidades
Diagnóstico inicial	<p>1. La universidad cuenta con profesionales altamente capacitados en el área tecnológica para llevar a cabo el desarrollo del Geoportal.</p> <p>2. Se cuenta con un excelente ambiente laboral y espacio adecuado.</p> <p>3. La dependencia CEDIT cuenta con la infraestructura tecnológica necesaria para llevar a cabo las actividades a desarrollar.</p>	<p>1. La universidad no cuenta aún con un Geoportal.</p> <p>2. Estudiantes y docentes recurren a plataformas de terceros para realizar la búsqueda de recursos digitales.</p>
Oportunidades	Estrategia FO	Estrategia DO
<p>1. Brindar herramientas tecnológicas a los estudiantes de la UFPSO para desarrollar proyectos y mejorar sus habilidades.</p>	<p>1. Usar los conocimientos del personal de apoyo de la dependencia para garantizar un buen desarrollo y mejorar la organización de los objetos virtuales de</p>	<p>1. Asegurar la disponibilidad del Geo portal, así como la centralización y estandarización de su información.</p>

aprendizaje.

Amenazas	Estrategia FA	Estrategia DA
1. Desenfoco de los objetivos principales en la ejecución del proyecto por falta de comunicación.	1. La comunicación de forma asertiva con los profesionales internos permitirá mejores resultados gracias a la capacidad y experiencia de estos, generando claridad en el contexto y optimizando el tiempo, el esfuerzo y la calidad del trabajo.	1. Mantener un buen ambiente laboral y una buena comunicación permiten entender en cualquier contexto de tiempo y lugar la idea principal del mismo.

1.2.1 Planteamiento del problema.

Desde que la tecnología y los medios de comunicación empezaron a avanzar fue necesario adaptarse a este entorno ya que esto permitió entrar a una era digital en la que se puede estar en contacto con todo el resto del planeta en cualquier lugar y en cualquier momento, esto ha llevado a que se puedan realizar tareas de forma rápida gracias al avance que ha tenido la tecnología. (Wohner, 2019).

Hace aproximadamente tres años la mayor parte del planeta sufrió una pandemia de COVID-19 que impedía estar fuera de casa, esto evitaba poder realizar las labores diarias, lo que llevó a que muchas personas tuvieran que utilizar medios de comunicación digital para poder comunicarse con las demás personas, lo que motivó a que muchas personas utilizaran diferentes tipos de software para comunicarse y buscar información de todo tipo. Hoy en día la gran mayoría de las personas requieren de algún tipo de software para buscar información o localizar una persona u objeto. Es por esto que cada vez se hace más indispensable la tecnología y la utilización de páginas web, aplicaciones, portales que permitan realizar búsquedas en tiempo real para obtener información y mantener actualizados a las personas sobre todo lo que sucede alrededor. (Tait, 2005).

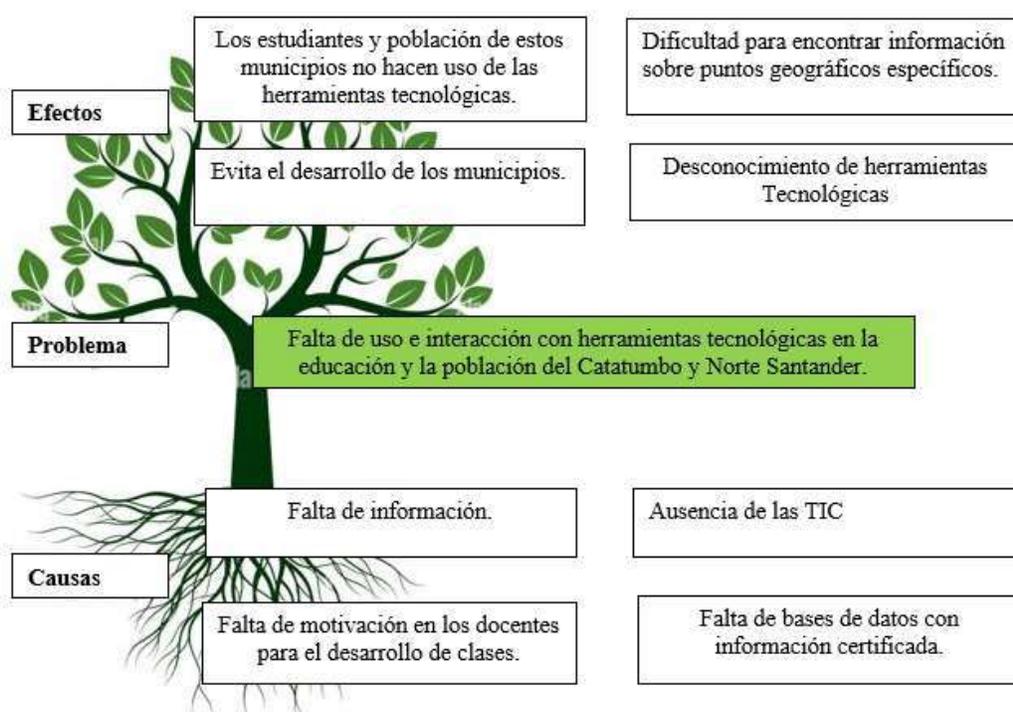
Debido a la falta de información sobre sitios específicos, con datos exactos en norte Santander y el Catatumbo, se hace necesario un geoportal donde consultar este tipo de información la cual se puede acceder en cualquier momento ya que esta debe estar disponible para que la población de estos municipios haga uso de ella.

Actualmente no se cuenta con ningún geoportal que brinde esta información a la población de norte Santander y el catatumbo lo que evita el desarrollo de estos y que sea difícil encontrar información sobre sitios geográficos, además de ello no se estaría haciendo uso de herramientas tecnológicas que permita motivar a los estudiantes y docentes a investigar.

El geoportal es una herramienta tecnológica la cual está siendo solicitada por la asociación de municipios del Catatumbo, donde se quiere brindar a la población un espacio para visualizar y obtener información mediante herramientas TIC como es el geoportal. Muchas veces se desconocen los sitios como colegios, farmacias, hospitales los cuales se pueden ubicar en el geoportal para que las personas de otros lugares puedan obtener este tipo de información.

Una estadística de gran importancia a mostrar en el geoportal, son los lugares en los que se ocasionan un mayor número de accidentes para las personas que en algunos casos son de otros municipios o desconocen este tipo de información. Según el artículo “determinación y análisis de las causas de accidentalidad en la ciudad de Ocaña norte de Santander”, desde el año 2014 hasta el 2017 se registraron 510 accidentes de tránsito, en los que muchas personas pierden la vida y otros quedan gravemente lesionados. (Prada et al., s. f.).

Figura 3. Árbol de problemas.



1.3 Objetivos de la pasantía

1.3.1 General.

Desarrollar un Geoportal en el Centro de Innovación y Tecnológica (CEDIT) de la universidad Francisco de Paula Santander Ocaña.

1.3.2 Específicos

1.3.2.1 Analizar el uso de nuevas tecnologías para el desarrollo del Geo portal Ocaña y

descifrar qué instrumentos deben implementarse para su funcionalidad.

1.3.2.2 Desarrollar funcionalidades en el Geoportal utilizando diferentes lenguajes de programación y patrones de diseño de Software.

1.3.2.3 Realizar pruebas de los cambios implementados en el Geoportal que permitan verificar el correcto funcionamiento del mismo.

1.4 Descripción de las actividades a desarrollar en el CEDIT

Tabla 2. Descripción de las actividades a desarrollar en el CEDIT.

Objetivo General	Objetivos Específicos	Actividades a desarrollar en la empresa para hacer posible el cumplimiento de los Obj. Específicos
	<p>Analizar el uso de nuevas tecnologías para el desarrollo del Geo portal Ocaña y descifrar qué instrumentos deben implementarse para su funcionalidad.</p>	<ol style="list-style-type: none"> 1. Investigar antecedentes bibliográficos y documentación de las tecnologías que se quieran utilizar. 2. Realizar la maquetación de todos los componentes. 3. Modelar la base de datos en postgres.
	<p>Desarrollar funcionalidades en el Geoportal utilizando</p>	<ol style="list-style-type: none"> 1. Construir e implementar la validación para el inicio de sesión para los usuarios.

Desarrollo de un Geoportal en el centro de innovación y tecnología (CEDIT) de la universidad Francisco de Paula Santander Ocaña.

diferentes lenguajes de

programación y patrones de diseño de Software.

2. Determinar y crear las rutas y componentes que requiere el Geo portal.

3. Desarrollar las funcionalidades requeridas para la interacción del usuario con el Geo portal.

Realizar pruebas de los cambios implementados en el Geoportal que permitan verificar el correcto funcionamiento del mismo.

1. Probar las distintas funcionalidades del Geoportal y solucionar posibles errores.

2. Verificar el correcto funcionamiento.

2. Enfoques referenciales

2.1 Enfoque conceptual.

Definición de conceptos relevantes utilizados en el estudio del proyecto:

- **Geoportal**

Un Geoportal está conformado por diferentes recursos como el visor geográfico que es donde se despliega la información espacial, los geo servicios que son el conjunto de funciones que se lograrán cómo, calcular número de aulas, definir su límites de uso de acuerdo con su área o disponibilidad de equipos de cómputo, o rutas de evacuación más cercana, y los metadatos que son la descripción de la información espacial que facilitan su inventario, consulta de calidad y condiciones y términos de uso, el contacto del custodio de la información y su forma de adquisición.(Cardona et al., s. f.)

- **Arcgis Online**

Incluye todo lo necesario para crear y compartir mapas, escenas, aplicaciones, notebooks y capas. Puede agregar contenido existente, por ejemplo, hojas de cálculo y servicios de ArcGIS Server, y cree nuevo contenido, tales como mapas, aplicaciones y capas web alojadas. (de, 2022)

- **Token**

Laravel genera automáticamente un "token" CSRF para cada sesión de usuario activa administrada por la aplicación. Este token se utiliza para verificar que el usuario autenticado es la persona que realmente realiza las solicitudes a la aplicación. Dado que este token se almacena en la sesión del usuario y cambia cada vez que se regenera la sesión, una aplicación malintencionada no puede acceder a él. (*Laravel-Clase2.pdf*, s. f.)

- **Modelo, vista, controlador**

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. (*Laravel-Clase1.pdf*, s. f.)

- **Dominio**

La búsqueda por facetas permite a los usuarios navegar por un espacio de información multidimensional al combinar la búsqueda por palabras clave con opciones de desglose en cada faceta. (Kong, 2016)

- **Hosting**

Es un alojamiento Web, donde los datos cibernéticos de una empresa residen en un servidor o un conjunto de servidores mantenidos por un proveedor de servicios de internet (ISP). Pagas el privilegio de almacenar tus datos. (Reichard, 1998)

- **Composer**

Composer es una herramienta para la gestión de dependencias en PHP. Le permite declarar las bibliotecas de las que depende su proyecto y las administra (instala/actualiza) por usted.

Composer. (s.f.)

- **Framework**

Es un conjunto de clases o librerías mediante las cuales se pueda desarrollar aplicaciones Web con navegación sensible a concerns de manera rápida, ordenada y segura. (Mendoza Vázquez, 2011).

- **Capa arcgis**

En ArcGIS Online, se puede trabajar con datos geográficos mediante capas. Las capas, también denominadas capas web, son conjuntos lógicos de datos geográficos que se utilizan para

crear mapas y escenas. También son la base del análisis geográfico. Por ejemplo, una capa de edificios podría representar un conjunto de edificios de un campus e incluye los atributos que describen las propiedades de cada edificio como, por ejemplo, el nombre del edificio, el tipo de edificio del que se trata, su tamaño y otros posibles atributos. (*Capas—Ayuda de ArcGIS Online _ Documentación.pdf*, s. f.)

- **Preloader**

Este atributo determina si el navegador debería comenzar a cargar el vídeo antes de ser reproducido. Acepta tres valores: none, metadato o auto. El primer valor indica que el vídeo no se debería cargar y generalmente se utiliza para minimizar tráfico web. El segundo valor, metadato, recomienda al navegador descargar información acerca del recurso, como las dimensiones, la duración, el primer cuadro, etc. El tercer valor, auto, solicita al navegador que descargue el archivo tan pronto como sea posible (este es el valor por defecto). (Gauchat, s. f.)

- **Responsive**

El Responsive Web Design (RWD de ahora en adelante) comprende una serie de técnicas y pautas de diseño que permiten adaptar sitios web al entorno de navegación del usuario, entendiendo como entorno de navegación la multiplicidad de dispositivos, móviles o no, por medio de los cuales los usuarios pueden acceder y navegar en internet (Lez, s. f.).

- **Modal**

Una ventana modal es una capa que se muestra por encima del contenido visual de una página web generalmente activado por una acción como un click o temporizador. (*modal.pdf*, s. f.)

- **Extensión Vscod**

Las extensiones de VS Code le permiten agregar idiomas, depuradores y herramientas a su instalación para respaldar su flujo de trabajo de desarrollo. (*Administrar extensiones en Visual*

Studio Code.pdf, s. f.)

- **Crud**

Permite estandarizar y obtener los procedimientos almacenados y/o funciones que realicen operaciones CRUD sobre bases de datos PostgreSQL, y así proveer a las aplicaciones que usen dichas Bases de Datos de operaciones CRUD centralizadas en el mismo clúster y lograr mejores tiempos de respuestas entre las aplicaciones y el servidor de base de datos PostgreSQL. (*crud.pdf*, s. f.)

2.2 Enfoque legal.

Dentro del marco legal para el desarrollo del Geoportal están:

Derechos de autor: Ley N. 1915. 12 de julio del 2018 por la cual se modifica la **Ley 23 de 1982** y se establecen otras disposiciones en materia de derecho de autor y derechos conexos.

Artículo 1°

Adiciónese al artículo 10 de la ley 23 de 1982 el siguiente párrafo.

Parágrafo. En todo proceso relativo al derecho de autor, y ante cualquier jurisdicción nacional se presumirá, salvo prueba en contrario, que la persona bajo cuyo nombre, seudónimo o su equivalente se haya divulgado la obra, será el titular de los derechos de autor. También se presumirá, salvo prueba en contrario, que la obra se encuentra protegida. (Ley 1915 del 12 de julio del 2018.pdf, s. f.)

Ley 64 del 10 de octubre de 2012

Artículo 22

Los programas de ordenador están protegidos por la presente Ley en los mismos términos que las obras literarias.

La protección se extiende a cualquiera de las versiones sucesivas del programa y a los programas derivados, tanto los operativos como los aplicativos, cualquiera que sea su modo de expresión, ya sea código fuente, código objeto o de cualquier otra forma. (*Gaceta Oficial Digital*, 2012).

Artículo 23

Sin perjuicio de la presunción que surja del registro a que se refiere el Capítulo II del Título XII de la presente Ley, se presume, salvo prueba en contrario, que es productor del programa de ordenador la persona que aparezca indicada como tal en la obra, de tal manera acostumbrada. (*Gaceta Oficial Digital*, 2012).

Ley 1273 DE 2009

"por medio de la cual se modifica el Código Penal, se crea un nuevo bien jurídico tutelado - denominado "de la protección de la información y de los datos y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones".

Artículo 269^a

ACCESO ABUSIVO A UN SISTEMA INFORMÁTICO. El que, sin autorización o por fuera de lo acordado, acceda en todo o en parte a un sistema informático protegido o no con una medida de seguridad, o se mantenga dentro del mismo en contra de la voluntad de quien tenga el legítimo derecho a excluirlo, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes.

(*Ley_1273_2009.pdf*, s. f.)

Artículo 269G.**SUPLANTACIÓN DE SITIOS WEB PARA CAPTURAR**

DATOS PERSONALES. El que con objeto ilícito y sin estar facultado para ello, diseñe, desarrolle, trafique, venda, ejecute, programe o envíe páginas electrónicas, enlaces o ventanas emergentes, incurrirá en pena de prisión de cuarenta y ocho (48) a noventa y seis (96) meses y en multa de 100 a 1000 salarios mínimos legales mensuales vigentes, siempre que la conducta no constituya delito sancionado con pena más grave. En la misma sanción incurrirá el que modifique el sistema de resolución de nombres de dominio, de tal manera que haga entrar al usuario a una IP diferente en la creencia de que acceda a su banco o a otro sitio personal o de confianza, siempre que la conducta no constituya delito sancionado con pena más grave. la pena señalada en los dos incisos anteriores se agravará de una tercera parte a la mitad, si para consumarlo el agente ha reclutado víctimas en la cadena del delito. (*Ley_1273_2009.pdf*, s. f.)

Ley 1065 DE 2006

Por la cual se define la administración de registros de nombres de dominio.co y se dictan otras disposiciones.

Artículo 1º

Definición. La administración del registro de nombres de dominio .co es aquella actividad a cargo del Estado, que tiene por objeto la organización, administración y gestión del dominio.co, incluido el mantenimiento de las bases de datos correspondientes, los servicios de información asociados al público, el registro de los nombres de dominio, su funcionamiento, la operación de sus servidores y la difusión de archivos de zona del dominio, y demás aspectos relacionados, de conformidad con las prácticas y definiciones de los organismos internacionales

competentes.(Rivera, s. f.)

Ley 565 DE 2000

Por medio de la cual se aprueba el "Tratado de la OMPI -Organización Mundial de la Propiedad Intelectual- sobre Derechos de Autor (WCT)", adoptado en

Ginebra, el veinte (20) de diciembre de mil novecientos noventa y seis (1996).

(ley565_2000.pdf, s. f.)

3. Informe de cumplimiento de trabajo

3.1 Presentación de resultados.

3.1.1 Objetivo 1. Analizar el uso de nuevas tecnologías para el desarrollo del Geo portal Ocaña y descifrar qué instrumentos deben implementarse para su funcionalidad.

- **Actividad 1.** Investigar antecedentes bibliográficos y documentación de las tecnologías que se quieran utilizar.

Inicialmente se investigó sobre los diferentes tipos de bases de datos relacionales y no relacionales, frameworks como Angular, Laravel, Bootstrap y Angular Material, lenguajes de programación como JS, PHP, HTML y estilos CSS, por otro lado, se estudió el Geoportal de Medellín para tener un amplio conocimiento en el tema, su conexión con la base de datos y Arcgis Online.

Arcgis permite crear diferentes tipos de mapas desde Arcgis desktop y subirlos a la plataforma Arcgis online, estos se pueden modificar en tiempo real y realizar la conexión desde el framework con la base de datos. Arcgis Online permite crear una cuenta de prueba por 30 días, en la que se pueden realizar búsquedas de capas creadas y poder usarlas para tener una visión de cómo podría quedar la implementación de esta. Si se desea subir las capas creadas a Arcgis Online, se debe adquirir la licencia la cual ofrece una gran variedad de ventajas.

En la investigación realizada se encontró que al usar la versión de prueba de Arcgis, esta no permite subir las capas creadas desde Arcgis desktop a Arcgis Online, por lo que es necesario comprar el plan de arcgis pro, online e Enterprise para poder hacer uso de todas sus funciones.

En cuanto a las bases de datos existe una gran variedad de estas, las cuales ofrecen una serie de ventajas y desventajas ya sean bases de datos relacionales o no relacionales.

Teniendo en cuenta que el Geoportal es un software escalable y que necesita almacenar grandes cantidades de información se debe utilizar una base de datos que permita realizar consultas de forma rápida y segura.

Actualmente la mayor parte de las personas hacen uso de dispositivos móviles como celulares, tablets y portátiles. Hoy en día el celular se ha vuelto una herramienta indispensable para el ser humano y de fácil acceso, debido a ello fue necesario realizar un estudio sobre el diseño responsive e implementarlo en el Geoportal para que se adapte a estos dispositivos y la información se pueda visualizar de forma correcta.

Una de las herramientas útiles para la creación de cualquier proyecto de software es el portal Github el cual permite alojar el código fuente y también ofrece la opción de subir los cambios que se hacen a diario, mantenerlos en línea y compartirlos con otros usuarios, para evitar la pérdida de información y permitir que otros usuarios puedan modificar el código.

Si observamos el framework Laravel tiene una arquitectura la cual es necesario comprender antes de empezar a crear un proyecto, esta se muestra a continuación:

En la carpeta Http, Controllers Auth se crean los controladores estos hacen consultas a un modelo el cual representa una tabla en la base de datos, el modelo le retorna datos al controlador y este retorna una vista.

Figura 4. Carpeta Controllers.

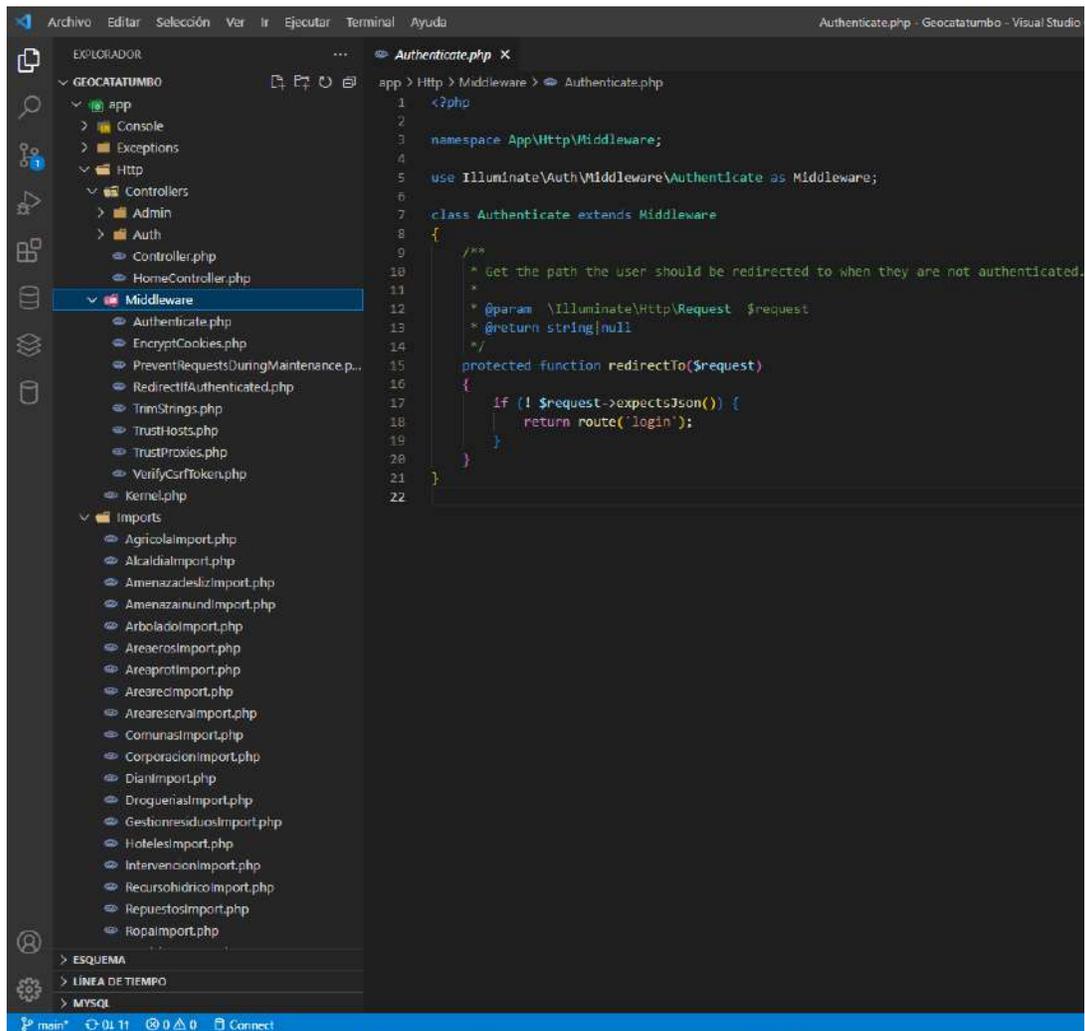
```

1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Controllers\Controller;
6  use Illuminate\Http\Request;
7  use App\Models\Agricola;
8
9  use App\Imports\AgricolaImport;
10 use Maatwebsite\Excel\Facades\Excel;
11
12 class AgricolaController extends Controller
13 {
14
15     public function __construct()
16     {
17         $this->middleware('auth');
18     }
19
20     public function index()
21     {
22         // aquery encargado de traer todo lo que esta en la tabla gestion
23         $agricola=Agricola::all();
24         return view('admin.agricola.index', ['agricola'=>$agricola]);
25     }
26
27     // creacion de contenido
28     public function store(Request $request)
29     {
30         $newAgricola= new Agricola();
31         $newAgricola ->cod_ciiu =$request->cod_ciiu;
32         $newAgricola ->nombre_empresa =$request->nombre_empresa;
33         $newAgricola ->ubic_geo =$request->ubic_geo;
34         $newAgricola ->comuna =$request->comuna;
35         $newAgricola ->barrio =$request->barrio;
36         $newAgricola ->save();
37         //redirigir hacia atras
38         return redirect()->back();
39     }
40
41     public function update(Request $request, $agricolaId)
42     {
43         $agricola= Agricola::find( $agricolaId);
44         $agricola ->cod_ciiu =$request->cod_ciiu;
45         $agricola ->nombre_empresa =$request->nombre_empresa;
46         $agricola ->ubic_geo =$request->ubic_geo;
47         $agricola ->comuna =$request->comuna;
48         $agricola ->barrio =$request->barrio;
49         $agricola->save();

```

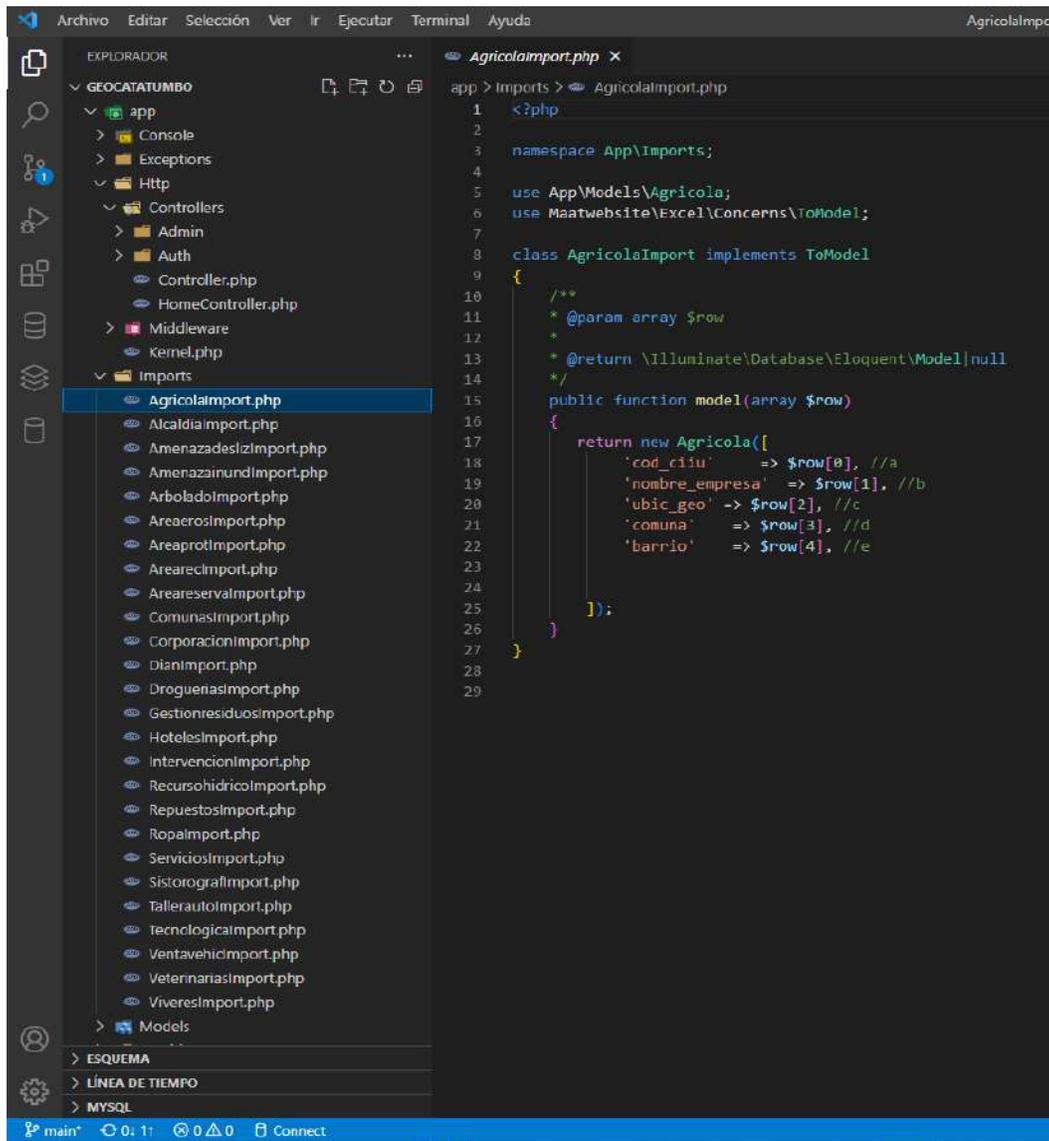
La carpeta Middleware contiene los archivos de autenticación y encriptación de contraseñas.

Figura 5. Carpeta Middleware.



La carpeta imports permite leer cada columna del archivo Excel para realizar la carga masiva de una tabla.

Figura 6. Carpeta Imports.

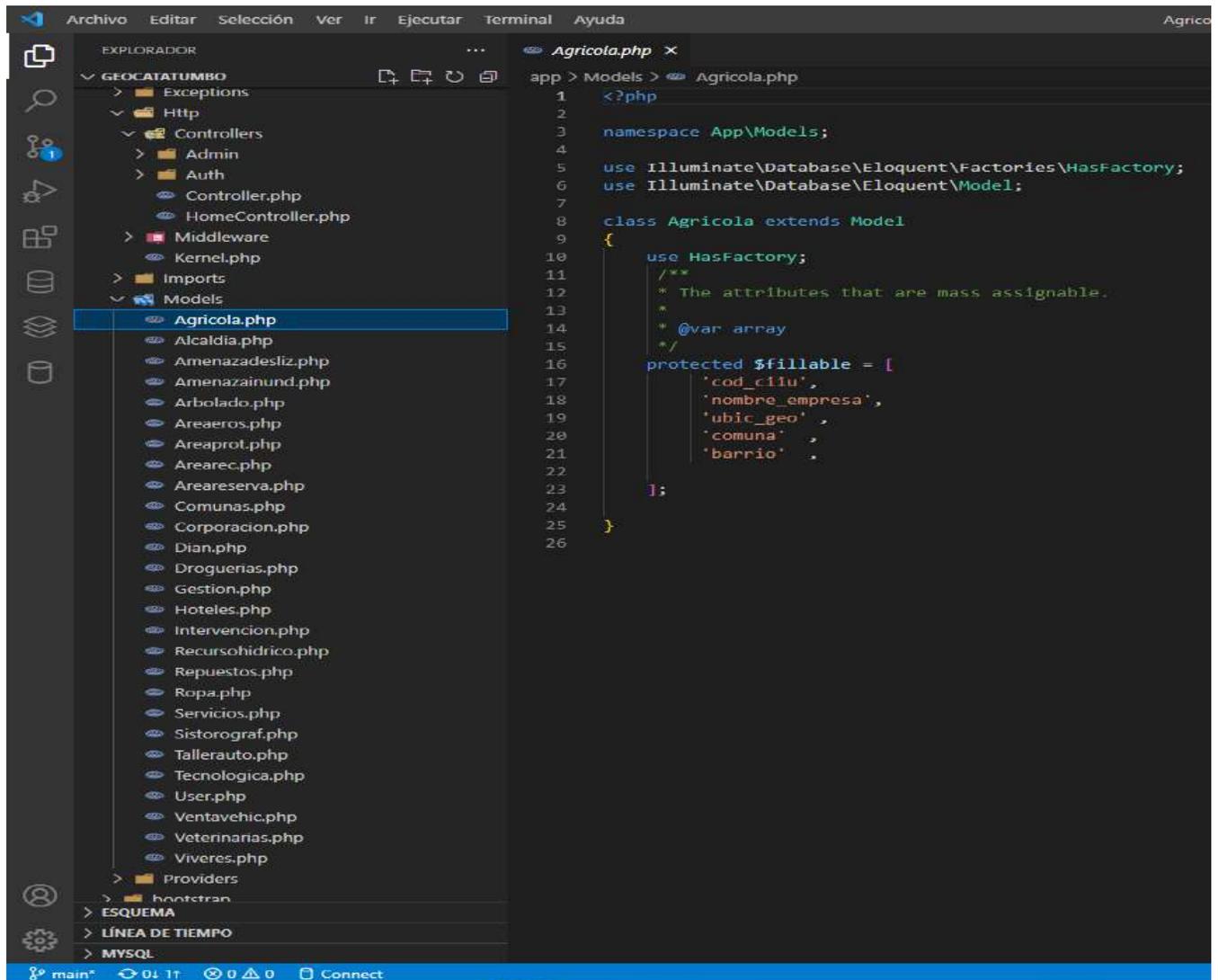


The image shows a screenshot of an IDE (Visual Studio Code) with a dark theme. The left sidebar displays the 'EXPLORADOR' (Explorer) view, showing a project structure for 'GEOCATATUMBO'. The 'app' folder is expanded, and the 'Imports' subfolder is selected, listing numerous PHP files such as 'AgricolaImport.php', 'AlcaldiaImport.php', 'AmenazadesIslImport.php', etc. The main editor window shows the code for 'AgricolaImport.php'. The code is as follows:

```
1 <?php
2
3 namespace App\Imports;
4
5 use App\Models\Agricola;
6 use Maatwebsite\Excel\Concerns\ToModel;
7
8 class AgricolaImport implements ToModel
9 {
10     /**
11      * @param array $row
12      *
13      * @return \Illuminate\Database\Eloquent\Model|null
14      */
15     public function model(array $row)
16     {
17         return new Agricola([
18             'cod_ciou' => $row[0], //a
19             'nombre_empresa' => $row[1], //b
20             'ubic_geo' => $row[2], //c
21             'comuna' => $row[3], //d
22             'barrio' => $row[4], //e
23         ]);
24     }
25 }
26
27
28
29
```

La carpeta Models contiene las clases que representan las tablas en la base de datos, aquí se realizan las consultas para acceder a los datos.

Figura 7. Carpeta Models.

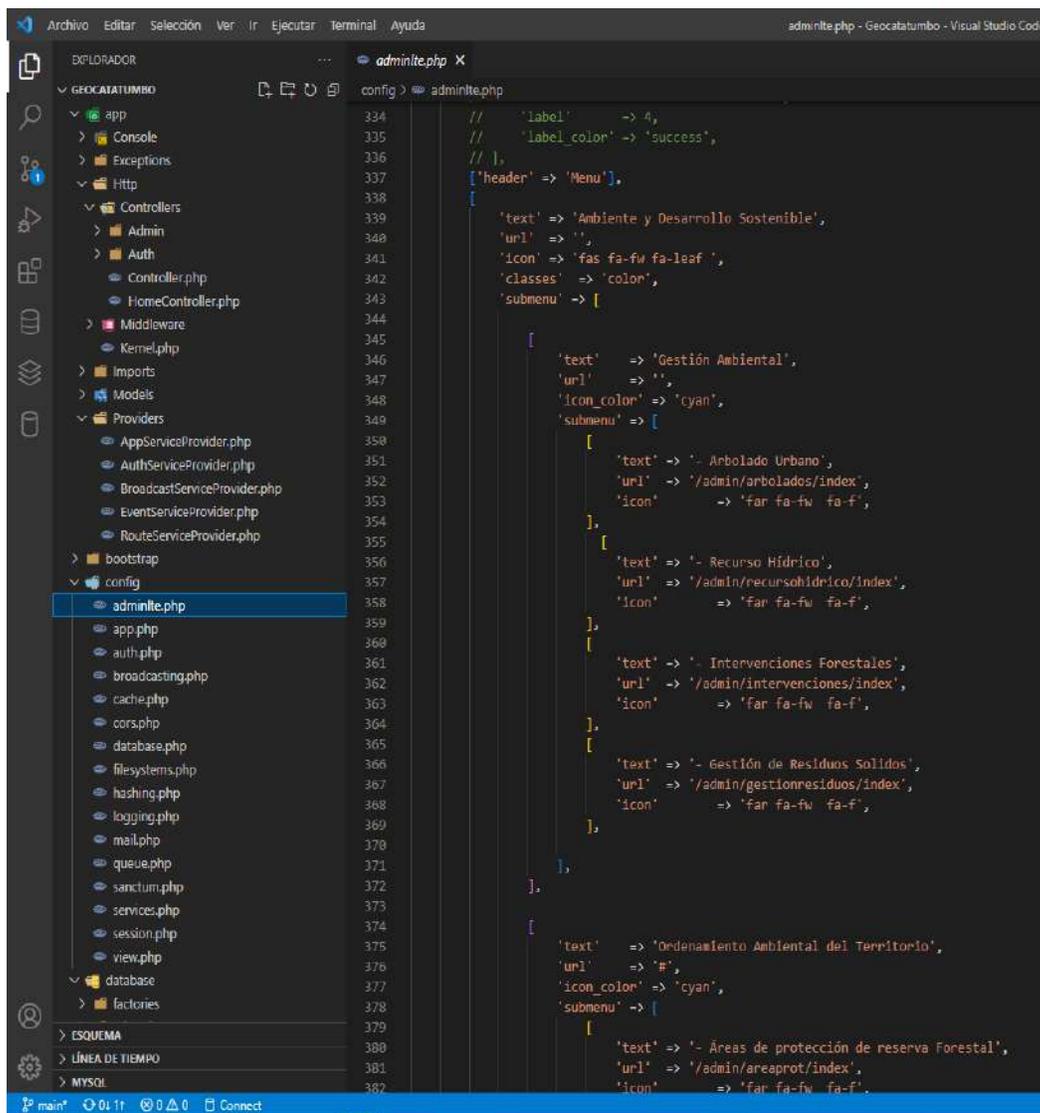


The image shows a screenshot of an IDE (likely Visual Studio Code) with the file explorer on the left and the code editor on the right. The file explorer shows the directory structure of a project, with the 'Models' folder expanded and 'Agricola.php' selected. The code editor displays the following PHP code for 'Agricola.php':

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Agricola extends Model
9 {
10     use HasFactory;
11     /**
12      * The attributes that are mass assignable.
13      *
14      * @var array
15      */
16     protected $fillable = [
17         'cod_cilu',
18         'nombre_empresa',
19         'ubic_geo',
20         'comuna',
21         'barrio'
22     ];
23 }
24
25
26
```

Al crear el panel de control se crea la carpeta Config la cual contiene el archivo adminLTE donde se puede modificar el menú lateral derecho.

Figura 8. Carpeta Config.



```
334 // 'label'    => '4',
335 // 'label_color' => 'success',
336 // ],
337 ['header' => 'Menu'],
338 [
339     'text' => 'Ambiente y Desarrollo Sostenible',
340     'url' => '#',
341     'icon' => 'fas fa-fw fa-leaf',
342     'classes' => 'color',
343     'submenu' => [
344         [
345             'text' => 'Gestión Ambiental',
346             'url' => '#',
347             'icon_color' => 'cyan',
348             'submenu' => [
349                 [
350                     'text' => '- Arbolado Urbano',
351                     'url' => '/admin/arboloados/index',
352                     'icon' => 'far fa-fw fa-f',
353                 ],
354                 [
355                     'text' => '- Recurso Hídrico',
356                     'url' => '/admin/recursohidrico/index',
357                     'icon' => 'far fa-fw fa-f',
358                 ],
359                 [
360                     'text' => '- Intervenciones Forestales',
361                     'url' => '/admin/intervenciones/index',
362                     'icon' => 'far fa-fw fa-f',
363                 ],
364                 [
365                     'text' => '- Gestión de Residuos Sólidas',
366                     'url' => '/admin/gestionresiduos/index',
367                     'icon' => 'far fa-fw fa-f',
368                 ],
369             ],
370         ],
371     ],
372 ],
373 [
374     'text' => 'Ordenamiento Ambiental del Territorio',
375     'url' => '#',
376     'icon_color' => 'cyan',
377     'submenu' => [
378         [
379             'text' => '- Áreas de protección de reserva Forestal',
380             'url' => '/admin/areaprot/index',
381             'icon' => 'far fa-fw fa-f',
382         ],
383     ],
384 ],
385 ],
386 ],
387 ],
388 ],
389 ],
390 ],
391 ],
392 ],
393 ],
394 ],
395 ],
396 ],
397 ],
398 ],
399 ],
400 ],
401 ],
402 ],
403 ],
404 ],
405 ],
406 ],
407 ],
408 ],
409 ],
410 ],
411 ],
412 ],
413 ],
414 ],
415 ],
416 ],
417 ],
418 ],
419 ],
420 ],
421 ],
422 ],
423 ],
424 ],
425 ],
426 ],
427 ],
428 ],
429 ],
430 ],
431 ],
432 ],
433 ],
434 ],
435 ],
436 ],
437 ],
438 ],
439 ],
440 ],
441 ],
442 ],
443 ],
444 ],
445 ],
446 ],
447 ],
448 ],
449 ],
450 ],
451 ],
452 ],
453 ],
454 ],
455 ],
456 ],
457 ],
458 ],
459 ],
460 ],
461 ],
462 ],
463 ],
464 ],
465 ],
466 ],
467 ],
468 ],
469 ],
470 ],
471 ],
472 ],
473 ],
474 ],
475 ],
476 ],
477 ],
478 ],
479 ],
480 ],
481 ],
482 ],
483 ],
484 ],
485 ],
486 ],
487 ],
488 ],
489 ],
490 ],
491 ],
492 ],
493 ],
494 ],
495 ],
496 ],
497 ],
498 ],
499 ],
500 ],
501 ],
502 ],
503 ],
504 ],
505 ],
506 ],
507 ],
508 ],
509 ],
510 ],
511 ],
512 ],
513 ],
514 ],
515 ],
516 ],
517 ],
518 ],
519 ],
520 ],
521 ],
522 ],
523 ],
524 ],
525 ],
526 ],
527 ],
528 ],
529 ],
530 ],
531 ],
532 ],
533 ],
534 ],
535 ],
536 ],
537 ],
538 ],
539 ],
540 ],
541 ],
542 ],
543 ],
544 ],
545 ],
546 ],
547 ],
548 ],
549 ],
550 ],
551 ],
552 ],
553 ],
554 ],
555 ],
556 ],
557 ],
558 ],
559 ],
560 ],
561 ],
562 ],
563 ],
564 ],
565 ],
566 ],
567 ],
568 ],
569 ],
570 ],
571 ],
572 ],
573 ],
574 ],
575 ],
576 ],
577 ],
578 ],
579 ],
580 ],
581 ],
582 ],
583 ],
584 ],
585 ],
586 ],
587 ],
588 ],
589 ],
590 ],
591 ],
592 ],
593 ],
594 ],
595 ],
596 ],
597 ],
598 ],
599 ],
600 ],
601 ],
602 ],
603 ],
604 ],
605 ],
606 ],
607 ],
608 ],
609 ],
610 ],
611 ],
612 ],
613 ],
614 ],
615 ],
616 ],
617 ],
618 ],
619 ],
620 ],
621 ],
622 ],
623 ],
624 ],
625 ],
626 ],
627 ],
628 ],
629 ],
630 ],
631 ],
632 ],
633 ],
634 ],
635 ],
636 ],
637 ],
638 ],
639 ],
640 ],
641 ],
642 ],
643 ],
644 ],
645 ],
646 ],
647 ],
648 ],
649 ],
650 ],
651 ],
652 ],
653 ],
654 ],
655 ],
656 ],
657 ],
658 ],
659 ],
660 ],
661 ],
662 ],
663 ],
664 ],
665 ],
666 ],
667 ],
668 ],
669 ],
670 ],
671 ],
672 ],
673 ],
674 ],
675 ],
676 ],
677 ],
678 ],
679 ],
680 ],
681 ],
682 ],
683 ],
684 ],
685 ],
686 ],
687 ],
688 ],
689 ],
690 ],
691 ],
692 ],
693 ],
694 ],
695 ],
696 ],
697 ],
698 ],
699 ],
700 ],
701 ],
702 ],
703 ],
704 ],
705 ],
706 ],
707 ],
708 ],
709 ],
710 ],
711 ],
712 ],
713 ],
714 ],
715 ],
716 ],
717 ],
718 ],
719 ],
720 ],
721 ],
722 ],
723 ],
724 ],
725 ],
726 ],
727 ],
728 ],
729 ],
730 ],
731 ],
732 ],
733 ],
734 ],
735 ],
736 ],
737 ],
738 ],
739 ],
740 ],
741 ],
742 ],
743 ],
744 ],
745 ],
746 ],
747 ],
748 ],
749 ],
750 ],
751 ],
752 ],
753 ],
754 ],
755 ],
756 ],
757 ],
758 ],
759 ],
760 ],
761 ],
762 ],
763 ],
764 ],
765 ],
766 ],
767 ],
768 ],
769 ],
770 ],
771 ],
772 ],
773 ],
774 ],
775 ],
776 ],
777 ],
778 ],
779 ],
780 ],
781 ],
782 ],
783 ],
784 ],
785 ],
786 ],
787 ],
788 ],
789 ],
790 ],
791 ],
792 ],
793 ],
794 ],
795 ],
796 ],
797 ],
798 ],
799 ],
800 ],
801 ],
802 ],
803 ],
804 ],
805 ],
806 ],
807 ],
808 ],
809 ],
810 ],
811 ],
812 ],
813 ],
814 ],
815 ],
816 ],
817 ],
818 ],
819 ],
820 ],
821 ],
822 ],
823 ],
824 ],
825 ],
826 ],
827 ],
828 ],
829 ],
830 ],
831 ],
832 ],
833 ],
834 ],
835 ],
836 ],
837 ],
838 ],
839 ],
840 ],
841 ],
842 ],
843 ],
844 ],
845 ],
846 ],
847 ],
848 ],
849 ],
850 ],
851 ],
852 ],
853 ],
854 ],
855 ],
856 ],
857 ],
858 ],
859 ],
860 ],
861 ],
862 ],
863 ],
864 ],
865 ],
866 ],
867 ],
868 ],
869 ],
870 ],
871 ],
872 ],
873 ],
874 ],
875 ],
876 ],
877 ],
878 ],
879 ],
880 ],
881 ],
882 ],
883 ],
884 ],
885 ],
886 ],
887 ],
888 ],
889 ],
890 ],
891 ],
892 ],
893 ],
894 ],
895 ],
896 ],
897 ],
898 ],
899 ],
900 ],
901 ],
902 ],
903 ],
904 ],
905 ],
906 ],
907 ],
908 ],
909 ],
910 ],
911 ],
912 ],
913 ],
914 ],
915 ],
916 ],
917 ],
918 ],
919 ],
920 ],
921 ],
922 ],
923 ],
924 ],
925 ],
926 ],
927 ],
928 ],
929 ],
930 ],
931 ],
932 ],
933 ],
934 ],
935 ],
936 ],
937 ],
938 ],
939 ],
940 ],
941 ],
942 ],
943 ],
944 ],
945 ],
946 ],
947 ],
948 ],
949 ],
950 ],
951 ],
952 ],
953 ],
954 ],
955 ],
956 ],
957 ],
958 ],
959 ],
960 ],
961 ],
962 ],
963 ],
964 ],
965 ],
966 ],
967 ],
968 ],
969 ],
970 ],
971 ],
972 ],
973 ],
974 ],
975 ],
976 ],
977 ],
978 ],
979 ],
980 ],
981 ],
982 ],
983 ],
984 ],
985 ],
986 ],
987 ],
988 ],
989 ],
990 ],
991 ],
992 ],
993 ],
994 ],
995 ],
996 ],
997 ],
998 ],
999 ],
1000 ],
```

En la carpeta Database, migrations se guardan todas las migraciones que se hacen hacia la base de datos postgres, estas contienen la información de las columnas de cada tabla almacenada en postgres.

Figura 9. Carpeta Database.

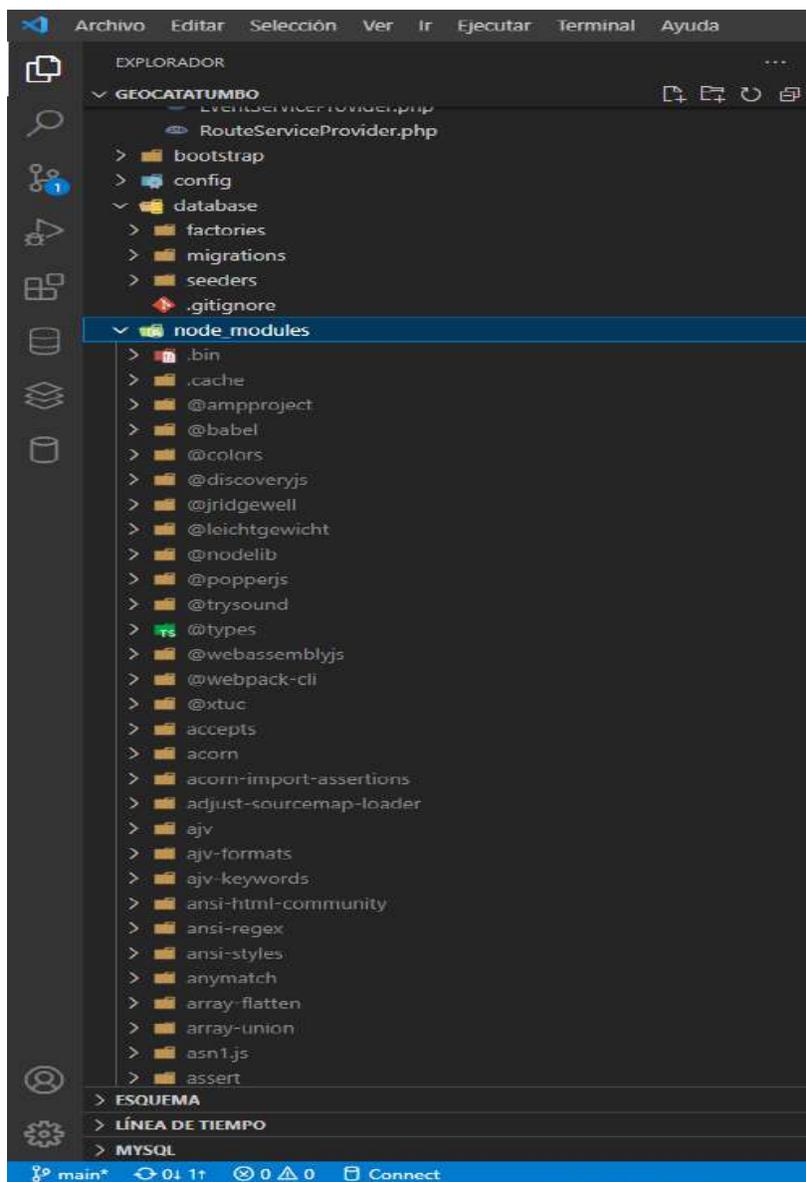
```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateUsersTable extends Migration
8  {
9
10     /**
11      * Run the migrations.
12      *
13      * @return void
14      */
15     public function up()
16     {
17         Schema::create('users', function (Blueprint $table) {
18             $table->id();
19             $table->string('name');
20             $table->string('email')->unique();
21             $table->timestamp('email_verified_at')->nullable();
22             $table->string('password');
23             $table->rememberToken();
24             $table->timestamps();
25         });
26     }
27
28     /**
29      * Reverse the migrations.
30      *
31      * @return void
32      */
33     public function down()
34     {
35         Schema::dropIfExists('users');
36     }
37 }

```

Carpeta donde se guardan los módulos de JavaScript vía npm.

Figura 10. Carpeta Módulos.



En la carpeta public se guardan todas las imágenes, estilos css y js del geoportal.

Figura 11. Carpeta Public.

The image shows a code editor window with two main panes. The left pane is the 'EXPLORADOR' (File Explorer) showing a project structure for 'GEOCATATUMBO'. The 'public' directory is expanded, showing sub-directories like 'css', 'images', and 'js'. The 'css' directory is further expanded to show 'cards', 'contenido', 'footer', 'logos', 'mapas', 'menu', 'menu-lateral', 'municipios', 'navbar', 'navbar-geovisor', 'panelcontrol', 'preloader', 'style', 'app.css', 'app.css.map', 'styles.css', and 'styles.css.map'. The right pane shows the content of the 'cards.css' file, which contains CSS rules for responsive design, card dimensions, button sizes, container layout, and card spacing.

```

1  /*responsive por debajo de 400px */
2  @media screen and (max-width: 400px) {
3      /*tamaño tarjeta */
4      .tarjeta{
5          width: 200px;
6          height: 250px;
7      }
8
9      /*tamaño boton */
10     .tarjetas .tarjeta .btn{
11         width: 70px !important;
12         height: 35px;
13         margin-left: 60px; /*boton centrado */
14         margin-top: -120px;
15         font-size: 9px;
16     }
17
18
19
20
21
22
23     /* tamaño contenedor */
24     .tarjetas {
25         margin-top: 50px;
26         width: 100%;
27         max-width: 1800px; /*tamaño pantalla*/
28         display: flex;
29         flex-wrap: wrap;
30         justify-content: center;
31         margin-left: 2px;
32     }
33
34
35     /* tamaño tarjeta */
36     .tarjetas .tarjeta .tarjeta_img {
37         width: 330px;
38         height: 320px;
39     }
40
41
42
43     /* separacion tarjetas */
44     .tarjeta {
45         margin-right: 25px;
46         margin-bottom: 5%;
47         transition: all .4s cubic-bezier(0.175, 0.885, 0, 1);
48         background-color: #rgb(136, 133, 133);
49     }

```


La carpeta layouts contiene las vistas principales como el index y los componentes los cuales son llamados dentro de una vista principal.

Figura 13. Carpeta Layouts.

```

20 </h1>
21 @stop
22
23 @section('content')
24
25 {{-- <div class="container-fluid">
26 <div class="row">
27 <div class="col-12">
28 <div class="card">
29 <div class="card-header">
30 <h1>Registrar</h1>
31 <form action="{{ route('admin.arbolado.index') }}" method="post" enctype
32 @csrf
33
34 @if(Session::has('message'))
35 <p>{{ Session::get('message') }}</p>
36 @endif
37
38 <input type="file" name="file" required>
39
40 <button class="btn btn-outline-secondary">Registro Masivo</button>
41 </form>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div> --}}
47
48 <div class="container-fluid">
49 <div class="row">
50 <div class="col-12">
51 <div class="card">
52 <div class="card-header">
53 <h3 class="card-title"><b>Gestión Ambiental</b>
54 <p>Arbolado Urbano
55 </h3>
56 </div>
57 <!-- /.card-header -->
58 <div class="card-body">
59 <table id="posts" class="table table-bordered table-striped">
60 <thead>
61 <tr>
62 <th>ID</th>
63 <th>Nombre Científico</th>
64 <th>Nombre Común</th>
65 <th>Familia</th>
66 <th>Genero</th>
67 <th>Especie</th>
68 <th>Origen</th>

```

La carpeta routes contiene un archivo web.php donde se almacena todas las rutas las cuales se comunican con los controladores para redirigir hacia las diferentes vistas.

Figura 14. Carpeta Routes.

```

148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196

```

```

route::get('/admin/ventavehic/index', 'App\Http\Controllers\Admin\VentavehicController@index')->name('admin.ventavehic.index');
route::post('/admin/ventavehic/store', [App\Http\Controllers\Admin\VentavehicController::class, 'store'])->name('admin.ventavehic.store');
route::post('/admin/ventavehic/{postId}/update', [App\Http\Controllers\Admin\VentavehicController::class, 'update'])->name('admin.ventavehic.update');
route::delete('/admin/ventavehic/{postId}/delete', [App\Http\Controllers\Admin\VentavehicController::class, 'delete'])->name('admin.ventavehic.delete');

route::get('/admin/droguerias/index', 'App\Http\Controllers\Admin\DrogueriasController@index')->name('admin.droguerias.index');
route::post('/admin/droguerias/store', [App\Http\Controllers\Admin\DrogueriasController::class, 'store'])->name('admin.droguerias.store');
route::post('/admin/droguerias/{postId}/update', [App\Http\Controllers\Admin\DrogueriasController::class, 'update'])->name('admin.droguerias.update');
route::delete('/admin/droguerias/{postId}/delete', [App\Http\Controllers\Admin\DrogueriasController::class, 'delete'])->name('admin.droguerias.delete');

route::get('/admin/alcaldia/index', 'App\Http\Controllers\Admin\AlcaldiaController@index')->name('admin.alcaldia.index');
route::post('/admin/alcaldia/store', [App\Http\Controllers\Admin\AlcaldiaController::class, 'store'])->name('admin.alcaldia.store');
route::post('/admin/alcaldia/{postId}/update', [App\Http\Controllers\Admin\AlcaldiaController::class, 'update'])->name('admin.alcaldia.update');
route::delete('/admin/alcaldia/{postId}/delete', [App\Http\Controllers\Admin\AlcaldiaController::class, 'delete'])->name('admin.alcaldia.delete');

route::get('/admin/corporacion/index', 'App\Http\Controllers\Admin\CorporacionController@index')->name('admin.corporacion.index');
route::post('/admin/corporacion/store', [App\Http\Controllers\Admin\CorporacionController::class, 'store'])->name('admin.corporacion.store');
route::post('/admin/corporacion/{postId}/update', [App\Http\Controllers\Admin\CorporacionController::class, 'update'])->name('admin.corporacion.update');
route::delete('/admin/corporacion/{postId}/delete', [App\Http\Controllers\Admin\CorporacionController::class, 'delete'])->name('admin.corporacion.delete');

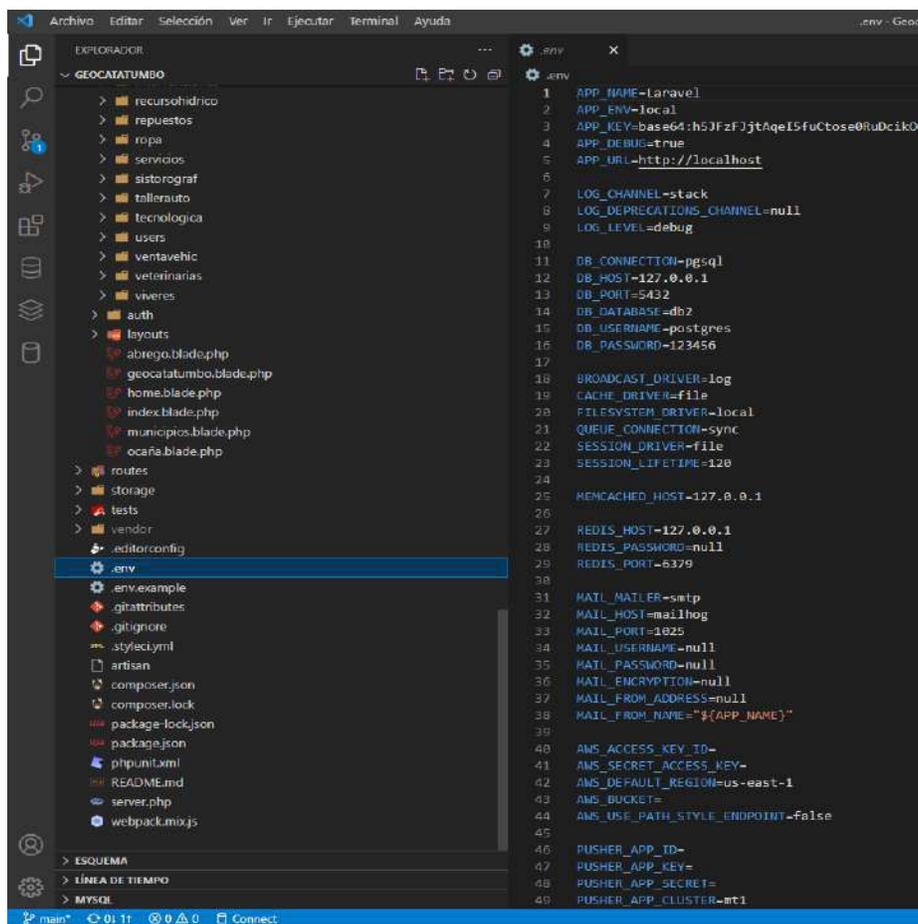
route::get('/admin/dian/index', 'App\Http\Controllers\Admin\DianController@index')->name('admin.dian.index');
route::post('/admin/dian/store', [App\Http\Controllers\Admin\DianController::class, 'store'])->name('admin.dian.store');
route::post('/admin/dian/{postId}/update', [App\Http\Controllers\Admin\DianController::class, 'update'])->name('admin.dian.update');
route::delete('/admin/dian/{postId}/delete', [App\Http\Controllers\Admin\DianController::class, 'delete'])->name('admin.dian.delete');

//Carga masiva
route::post('import-list-excel1', [App\Http\Controllers\Admin\ArboladosController::class, 'importExcel1'])->name('admin.arbolados.importExcel1');
route::post('import-list-excel2', [App\Http\Controllers\Admin\RecursosHidricoController::class, 'importExcel2'])->name('admin.recursosHidrico.importExcel2');
route::post('import-list-excel3', [App\Http\Controllers\Admin\IntervencionesController::class, 'importExcel3'])->name('admin.intervenciones.importExcel3');
route::post('import-list-excel4', [App\Http\Controllers\Admin\EstimacionesIdiomasController::class, 'importExcel4'])->name('admin.estimacionesIdiomas.importExcel4');
route::post('import-list-excel5', [App\Http\Controllers\Admin\AreasProteccionController::class, 'importExcel5'])->name('admin.areasProteccion.importExcel5');
route::post('import-list-excel6', [App\Http\Controllers\Admin\AreasReservaController::class, 'importExcel6'])->name('admin.areasReserva.importExcel6');
route::post('import-list-excel7', [App\Http\Controllers\Admin\AreasControlController::class, 'importExcel7'])->name('admin.areasControl.importExcel7');
route::post('import-list-excel8', [App\Http\Controllers\Admin\AreasAerosController::class, 'importExcel8'])->name('admin.areasAeros.importExcel8');
route::post('import-list-excel9', [App\Http\Controllers\Admin\SistorografController::class, 'importExcel9'])->name('admin.sistorograf.importExcel9');
route::post('import-list-excel10', [App\Http\Controllers\Admin\VenenazadeslizController::class, 'importExcel10'])->name('admin.venenazadesliz.importExcel10');
route::post('import-list-excel11', [App\Http\Controllers\Admin\VenenazalruidoController::class, 'importExcel11'])->name('admin.venenazalruido.importExcel11');
route::post('import-list-excel12', [App\Http\Controllers\Admin\ComunasController::class, 'importExcel12'])->name('admin.comunas.importExcel12');
route::post('import-list-excel13', [App\Http\Controllers\Admin\TecnologicaController::class, 'importExcel13'])->name('admin.tecnologica.importExcel13');
route::post('import-list-excel14', [App\Http\Controllers\Admin\AgriculturaController::class, 'importExcel14'])->name('admin.agricultura.importExcel14');
route::post('import-list-excel15', [App\Http\Controllers\Admin\ServiciosController::class, 'importExcel15'])->name('admin.servicios.importExcel15');
route::post('import-list-excel16', [App\Http\Controllers\Admin\ViveresController::class, 'importExcel16'])->name('admin.viveres.importExcel16');
route::post('import-list-excel17', [App\Http\Controllers\Admin\VotelesController::class, 'importExcel17'])->name('admin.voteles.importExcel17');
route::post('import-list-excel18', [App\Http\Controllers\Admin\VeterinariasController::class, 'importExcel18'])->name('admin.veterinarias.importExcel18');
route::post('import-list-excel19', [App\Http\Controllers\Admin\RepuestosController::class, 'importExcel19'])->name('admin.repuestos.importExcel19');
route::post('import-list-excel20', [App\Http\Controllers\Admin\RopaController::class, 'importExcel20'])->name('admin.ropa.importExcel20');
route::post('import-list-excel21', [App\Http\Controllers\Admin\TallerAutoController::class, 'importExcel21'])->name('admin.tallerAuto.importExcel21');
route::post('import-list-excel22', [App\Http\Controllers\Admin\VentavehicController::class, 'importExcel22'])->name('admin.ventavehic.importExcel22');

```

El archivo .env permite realizar la conexión con la base de datos ya sea postgres o MYSQL.

Figura 15. Archivo.env.



```

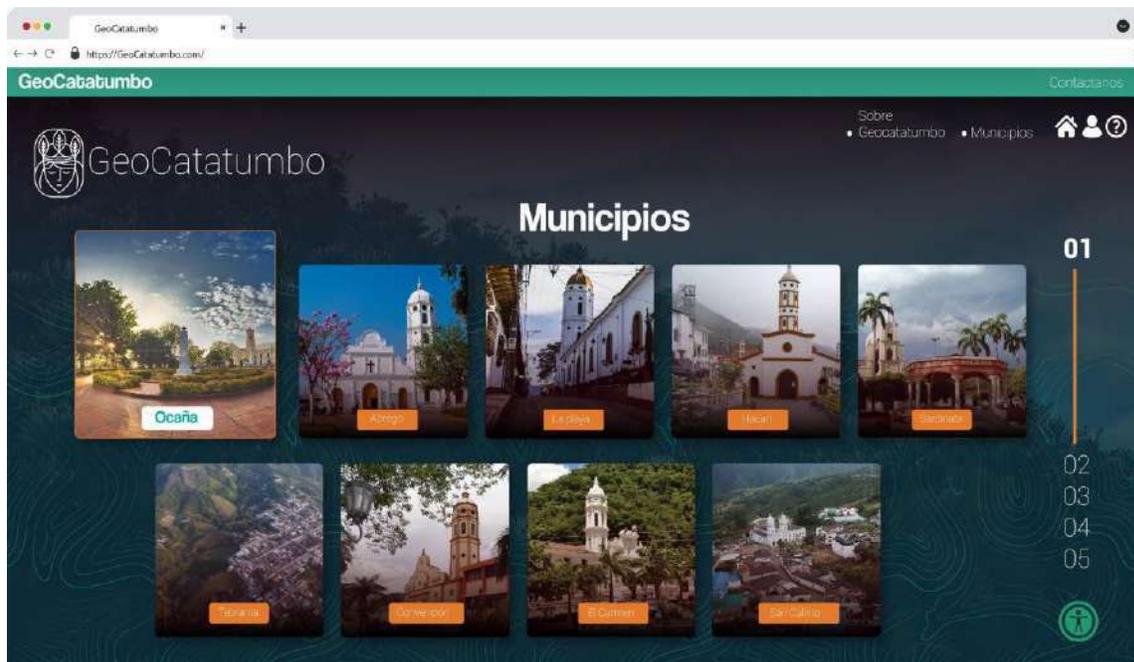
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:hsJfzFjtAqeI5fuCtose0RuDcik0
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=pgsql
12 DB_HOST=127.0.0.1
13 DB_PORT=5432
14 DB_DATABASE=db2
15 DB_USERNAME=postgres
16 DB_PASSWORD=123456
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DRIVER=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smtp
32 MAIL_HOST=mailhog
33 MAIL_PORT=1025
34 MAIL_USERNAME=null
35 MAIL_PASSWORD=null
36 MAIL_ENCRYPTION=null
37 MAIL_FROM_ADDRESS=null
38 MAIL_FROM_NAME="${APP_NAME}"
39
40 AWS_ACCESS_KEY_ID=
41 AWS_SECRET_ACCESS_KEY=
42 AWS_DEFAULT_REGION=us-east-1
43 AWS_BUCKET=
44 AWS_USE_PATH_STYLE_ENDPOINT=false
45
46 PUSHER_APP_ID=
47 PUSHER_APP_KEY=
48 PUSHER_APP_SECRET=
49 PUSHER_APP_CLUSTER=mt1

```

- **Actividad 2.** Realizar la maquetación de todos los componentes.

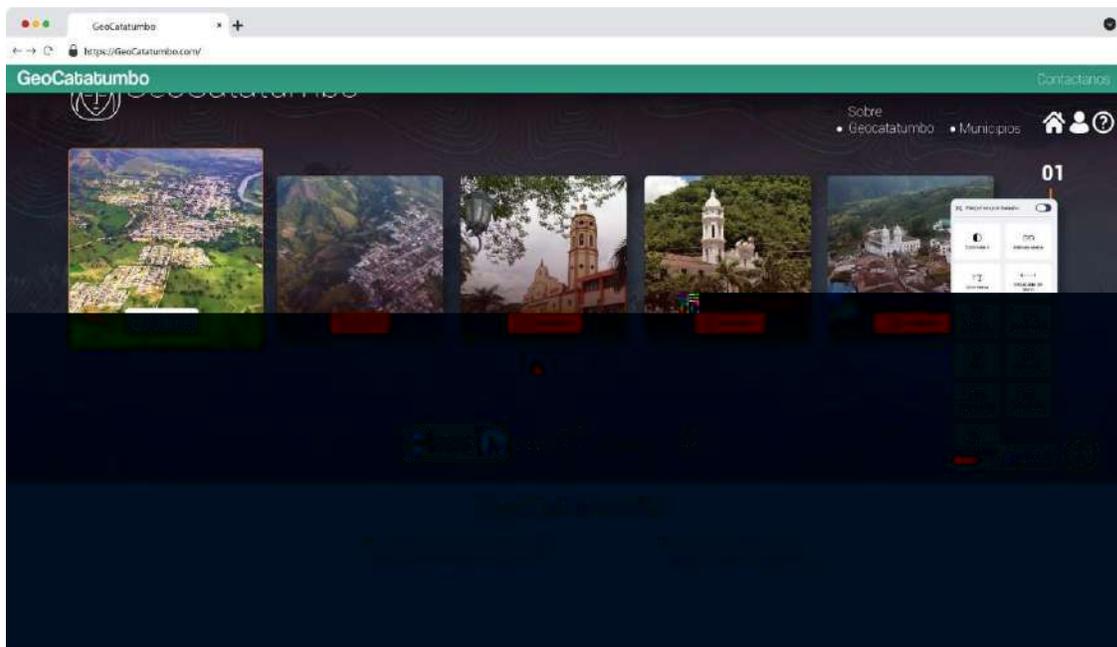
Luego de realizar la investigación y tener claro cuál es la base de datos con la que se desea trabajar y el framework, se realiza la maquetación del geoportal el cual es realizado por el compañero de trabajo Angelo Echavez Guerrero quien desempeña el cargo de Diseñador gráfico en el CEDIT, para poder tener un modelo y referencia de lo que se quiere lograr y de esta forma evitar inconvenientes a la hora de desarrollar cada componente.

Figura 16. Mockup pantalla inicio Geoportal.



(Echavez,2022)

Figura 17. Mockup Menú Accesibilidad Geoportal.



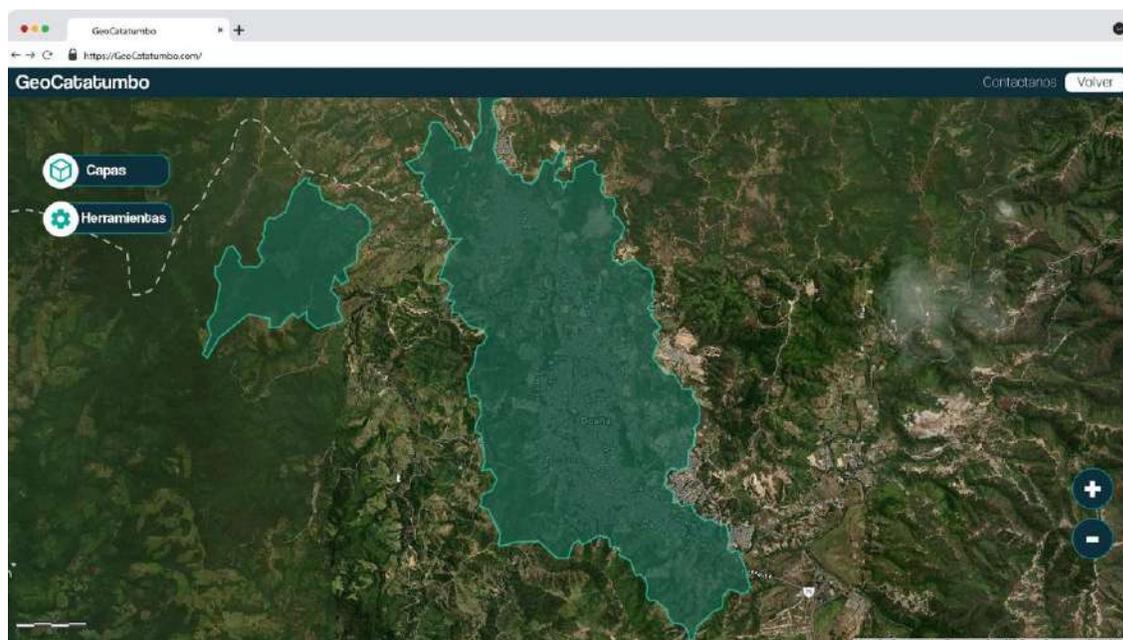
(Echavez,2022).

Figura 18. Mockup Pantalla de Carga.



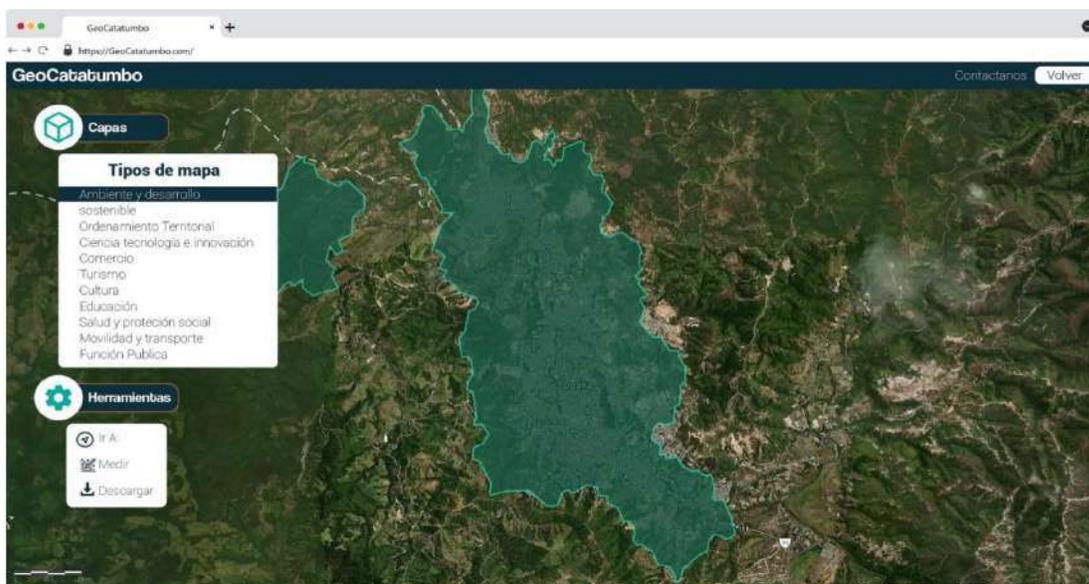
(Echavez,2022).

Figura 19. Mockup Geovisor del Geoportal.



(Echavez,2022).

Figura 20. Mockup Tipos de Capas Geoportal.

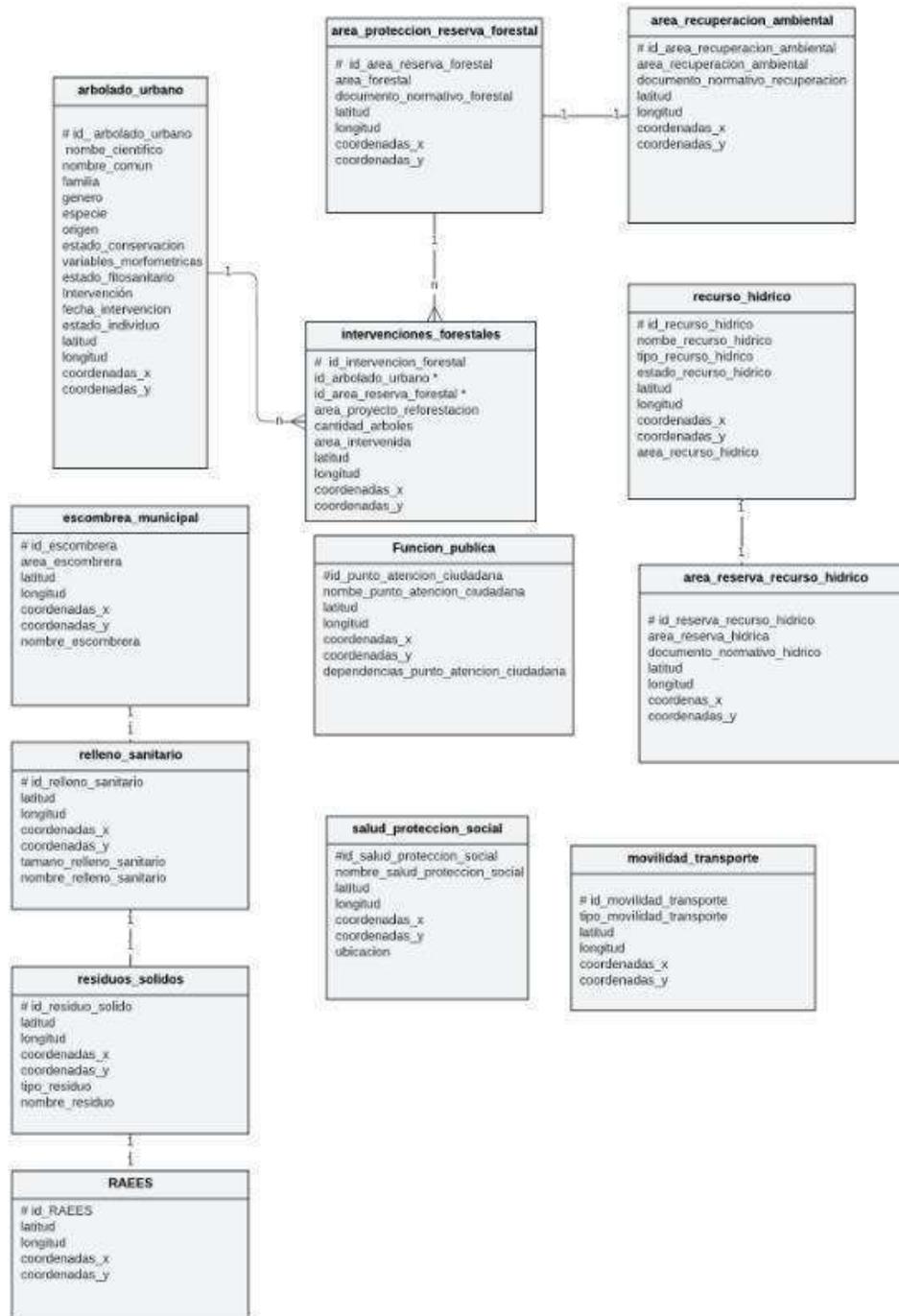


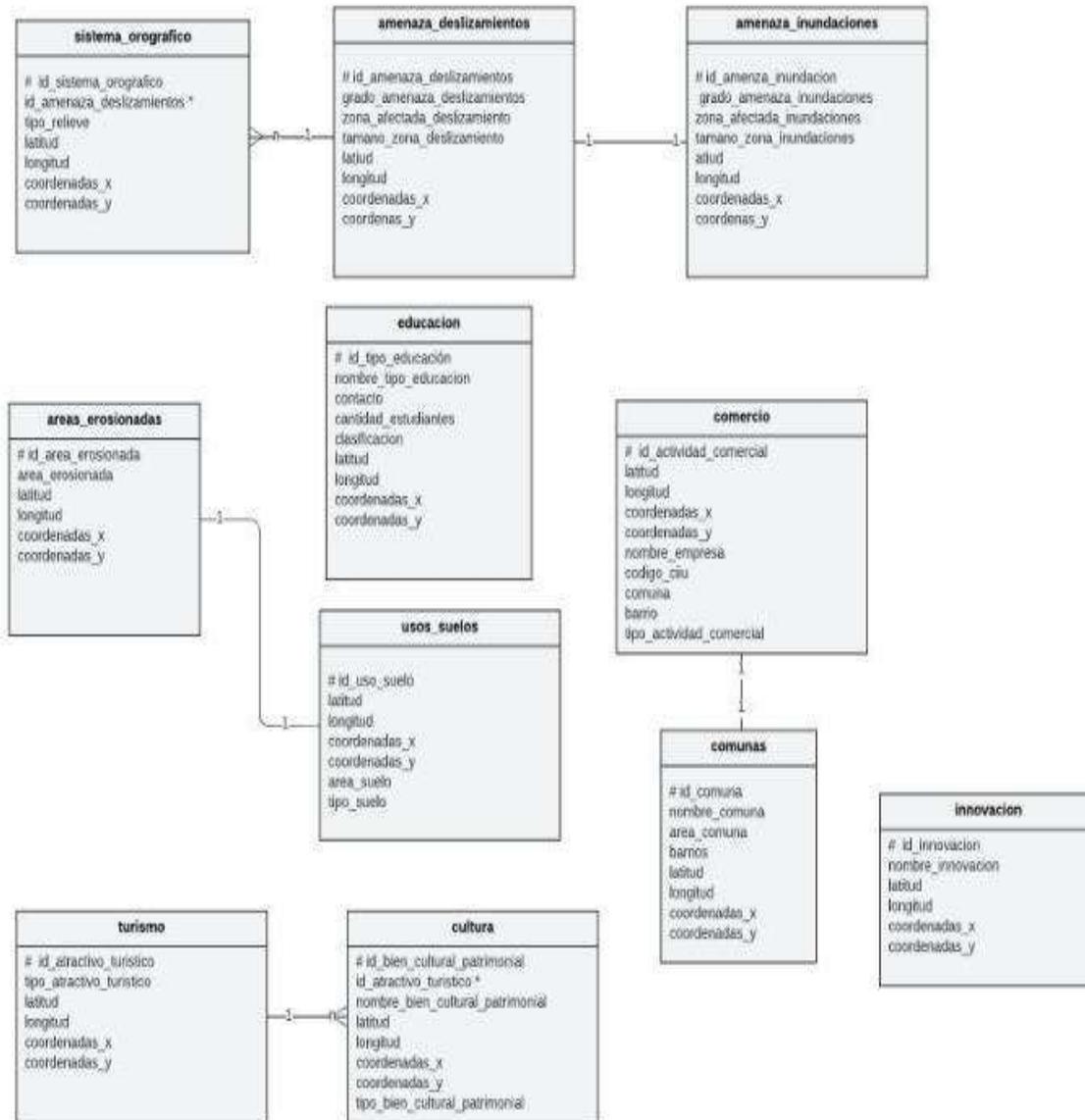
(Echavez,2022).

- **Actividad 3.** Modelar la base de datos en postgres.

Para poder almacenar la información del registro de los usuarios y la información creada por estos es necesario realizar las tablas que van a contener este tipo de información en la base de datos postgres, por lo que es necesario realizar el modelo Relacional para tener una guía de la información que va a contener la base de datos del geoportal.

Figura 21. Modelo relacional





(Hernández, 2022).

Figura 22. Modelado de tablas en postgres.

The screenshot shows the pgAdmin 4 interface. On the left, a tree view displays the database structure, including tables like 'agricolas', 'alcaldías', 'amenazadeslizs', etc. The main window shows a query editor with the following SQL query:

```
1 select * from agricultas
```

The query results are displayed in a table with the following columns and data:

id	cod_citi	nombre_empresa	ubic_geo	comuna	barrio
1	prueba	prueba	prueba	prueba	prueba
2	prueba2	prueba2	prueba2	prueba2	prueba2

At the bottom of the interface, it indicates 'Total rows: 2 of 2' and 'Query complete 00:00:00.309'.

A medida que avanza la tecnología avanzan los lenguajes de programación, para facilitar el desarrollo de software lo que trae una serie de ventajas ya que se puede analizar cual se ajusta a las necesidades del software como es el caso de Laravel y Angular a pesar de que son similares, Laravel permite realizar la conexión con la base de datos de forma rápida y segura, Laravel hace uso del lenguaje php lo que facilita trabajar en este a diferencia de Angular con el cual se batallo para poder realizar la conexión con postgres, debido a que no se logró se optó por utilizar Laravel.

3.1.2 Objetivo 2. Desarrollar funcionalidades en el Geoportail utilizando diferentes lenguajes de programación y patrones de diseño de Software.

- **Actividad 4.** Construir e implementar la validación para el inicio de sesión para los usuarios.

Laravel es un framework PHP gratis y de código abierto el cual brinda la posibilidad de trabajar en conjunto con otros frameworks como Bootstrap, hojas de estilo CSS y lenguajes de programación como JVS.

Para implementar la validación de inicio de sesión, es necesario crear un formulario de registro y Loguin el cual valida si el usuario que desea ingresar al sistema, se encuentra registrado en la base de datos o de lo contrario no podrá hacerlo, para esta validación, Laravel utiliza un token el cual protege el software de cualquier intruso, este consiste en generar automáticamente un token CSRF en cada sesión de usuario que ingresa al sistema brindando seguridad a este.

En laravel se puede crear un formulario de registro y login de forma rápida ya que con solo ejecutar una línea de comando como la siguiente se puede tener un inicio de sesión de forma rápida y segura.

Figura 23. Creación del login y registro en Laravel.

```

Microsoft Windows [Versión 10.0.17763.1]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.

C:\xampp\htdocs\ppp>php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Sat Nov 26 18:48:24 2022] 127.0.0.1:5722 [200]: /Favicon.ico
[Sat Nov 26 19:02:29 2022] 127.0.0.1:58024 [200]: /Favicon.ico
^C

C:\xampp\htdocs\ppp>composer require laravel/ui
Info from https://repo.packagist.org: #standaloneKvaine
Warning: laravel/ui's latest version v2.2.0 as it requires php ^8.0 which is not satisfied by your platform.
Using version ^2.2 for laravel/ui
./composer.json has been updated
Running composer update laravel/ui
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking laravel/ui (v2.5.8)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Downloading laravel/ui (v2.5.8)
  - Installing laravel/ui (v2.5.8): Extracting archive
Package symfony/console is abandoned, you should avoid using it. Use symfony/mailer instead.
Package symfony/finder is abandoned, you should avoid using it. No replacement was suggested.
Package symfony/http-token-stream is abandoned, you should avoid using it. No replacement was suggested.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facades/ignition
Discovered Package: fzaninotto/faker
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/tinker
Discovered Package: laravel/ui
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
All packages you are using are locked for funding.
Use the "composer fund" command to find out more!
No security advisories found!
  
```

A continuación, se observa el formulario de login creado anteriormente con solo ejecutar la línea de código composer require laravel/ui.

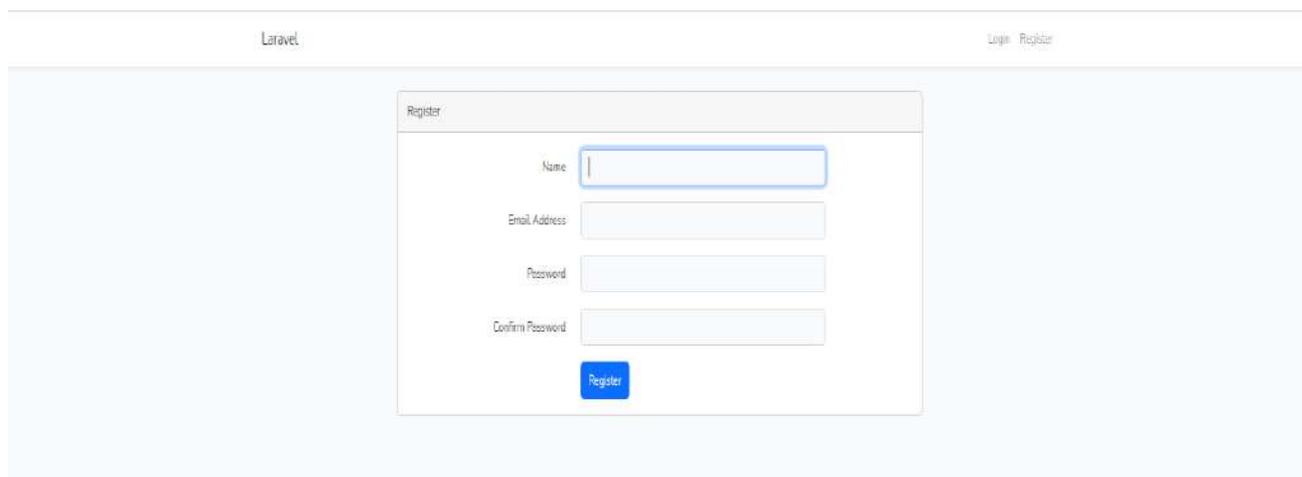
Figura 24. Formulario Login.

The screenshot shows a web browser window with the address bar containing '127.0.0.1:8000/login'. The page title is 'Laravel'. In the top right corner, there are links for 'Login' and 'Register'. The main content area features a 'Login' form with the following elements:

- An 'Email Address' input field containing the text 'pepito@hotmail.com'.
- A 'Password' input field with its content masked by dots.
- An unchecked checkbox labeled 'Remember Me'.
- A blue 'Login' button.
- A blue link labeled 'Forgot Your Password?'.

Al igual que el formulario de login Laravel crea un formulario de registro ejecutando la línea de código “composer require laravel/ui”.

Figura 25. Formulario de Registro.

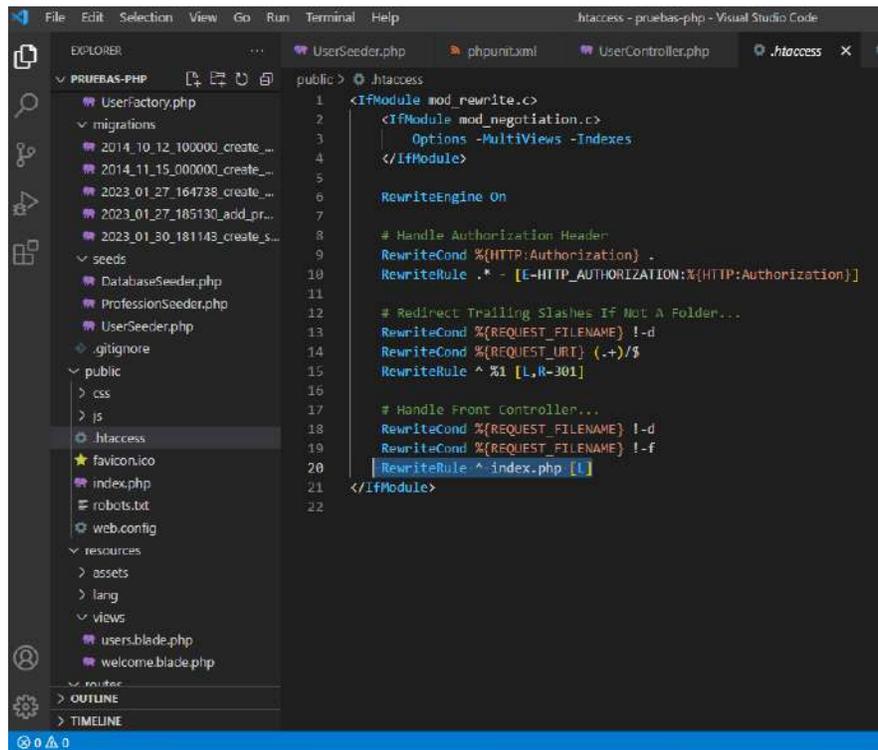
The image shows a screenshot of a web application's registration page. At the top left, the word "Laravel" is visible, and at the top right, there are links for "Login" and "Register". The main content is a registration form titled "Register". The form contains four input fields: "Name", "Email Address", "Password", and "Confirm Password". The "Name" field is currently active, with a blue border and a vertical cursor. Below the input fields is a blue button labeled "Register".

En Laravel se pueden implementar patrones de diseño, los cuales permiten organizar y estructurar el código para mantener el orden de este y mejorar el rendimiento. Laravel permite instalar librerías como illuminate/Suport y Laravel/UI, estas librerías ofrecen patrones de diseño como el patrón de fabricación, patrón observador y patrón singleton. La librería Laravel/UI es un paquete que incluye implementaciones de patrones como el patrón de inversión de dependencias y el patrón de inyección de dependencias.

Al momento de crear sitios web y aplicaciones con PHP se acostumbra a crear un archivo por cada página para mantener organizado el código, esto trae ciertos problemas como puede ser el código duplicado, código spaguetti, etc. Es por esto que frameworks como laravel, Sinfoni entre otros utilizan un patrón de diseño llamado Front Controller, este consiste en tener un solo punto de entrada a la aplicación, en Laravel se llama index.php, todas las peticiones que se realicen al framework Laravel llegan al index. Esto se logra cuando se utiliza Apache con mod_rewrite y el

archivo htaccess, donde se puede observar que todas las solicitudes que no sean directorios (public) o archivos sean redirigidas a index.php.

Figura 26. Patrón de diseño llamado Front Controller.



```

1 <IfModule mod_rewrite.c>
2 <IfModule mod_negotiation.c>
3     Options -MultiViews -Indexes
4 </IfModule>
5
6 RewriteEngine On
7
8 # Handle Authorization Header
9 RewriteCond %{HTTP:Authorization} .
10 RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
11
12 # Redirect Trailing Slashes If Not A Folder...
13 RewriteCond %{REQUEST_FILENAME} !-d
14 RewriteCond %{REQUEST_URI} (.+)/$
15 RewriteRule ^%1 [L,R=301]
16
17 # Handle Front Controller...
18 RewriteCond %{REQUEST_FILENAME} !-d
19 RewriteCond %{REQUEST_FILENAME} !-f
20 RewriteRule ^ index.php [L]
21 </IfModule>
22
  
```

El patrón symple factory permite hacer uso de la herencia, reutilizando el código para evitar volver a crear las mismas líneas de código en cada clase. Para ello se crea un componente el cual se llama en las clases que se desee mediante el comando extends.

Figura 27. Patrón de diseño Symple Factory.

```
<link rel="stylesheet" href="css/navbar/navbar.css">
<link rel="stylesheet" href="css/menu/menu.css">
<link rel="stylesheet" href="css/contenido/contenido.css">
<link rel="stylesheet" href="css/logos/logos.css">
<link rel="stylesheet" href="css/footer/footer.css">

</head>
<body>

  <!-- heredamos es navbar y footer -->
  <!-- se debe tener en cuenta el orden de los extends -->
  @extends('layouts.footer')
  @extends('layouts.logos')
  @extends('layouts.menu')
  @extends('layouts.navbar')

  @section('footer')
  @endsection('footer')

  @section('logos')
  @endsection('logos')
```

El patrón strategy permite ejecutar una acción la cual se debe cumplir para que esta se realice. En caso que se desee eliminar un dato de una tabla se debe emitir una alerta la cual indica si se desea eliminar la información o se desea cancelar la petición. Al seleccionar la opción el patrón strategy debe escoger la opción indicada, ejecutando de esta manera la opción correcta.

En caso de que la opción sea correcta se hace un llamado mediante el id haciendo referencia a la clase que se encuentra en el controlador, permitiendo ejecutar la función delete mediante una petición al servidor con un request, y a su vez retornando una respuesta del dato eliminado.

Figura 28. Patrón de diseño Strategy.

```

public function delete(Request $request, $agricolaId)
{
    $agricola= Agricola::find( $agricolaId);
    $agricola->delete();
    return redirect()->back();
}

```

Figura 29. Llamado mediante el id a la clase agrícola.

```

<div class="modal fade" id="modal-delete-agricola-{{ $agricola->id }}">
  <div class="modal-dialog">
    <div class="modal-content bg-default">
      <div class="modal-header">
        <h4 class="modal-title">Desea Eliminar esta Información</h4>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span></button>
      </div>
      <form action="{{ route('admin.agricola.delete', $agricola->id) }}" method="POST">
        {{ csrf_field() }}
        @method('DELETE')
        <div class="modal-footer justify-content-between">
          <button type="button" class="btn btn-outline-primary" data-dismiss="modal">No</button>
          <button type="submit" class="btn btn-outline-primary">Si</button>
        </div>
      </form>
    </div>
  </div>
  <!-- /.modal-content -->
</div>
<!-- /.modal-dialog -->
</div>

```

El patrón Observer permite ejecutar una función desde el controlador ya sea crear, editar o eliminar el cual se ejecuta según la acción requerida para lo cual se hace uso de la librería illuminate. El patrón Observer se encarga de que la acción solicitada por el usuario se cumpla.

Figura 30. Patrón de diseño Observer.

```
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\Agricola;
```

Figura 31. Funciones crear, editar y eliminar.

```
// creacion de contenido
public function store(Request $request)
{
    $newAgricola= new Agricola();
    $newAgricola ->cod_ciiu = $request->cod_ciiu;
    $newAgricola ->nombre_empresa = $request->nombre_empresa;
    $newAgricola ->ubic_geo = $request->ubic_geo;
    $newAgricola ->comuna = $request->comuna;
    $newAgricola ->barrio = $request->barrio;
    $newAgricola ->save();
    //redirigir hacia atras
    return redirect()->back();
}

public function update(Request $request, $agricolaId)
{
    $agricola= Agricola::find( $agricolaId);
    $agricola ->cod_ciiu = $request->cod_ciiu;
    $agricola ->nombre_empresa = $request->nombre_empresa;
    $agricola ->ubic_geo = $request->ubic_geo;
    $agricola ->comuna = $request->comuna;
    $agricola ->barrio = $request->barrio;
    $agricola->save();
    return redirect()->back();
}

public function delete(Request $request, $agricolaId)
{
    $agricola= Agricola::find( $agricolaId);
    $agricola->delete();
    return redirect()->back();
}
```

- **Actividad 5.** Determinar y crear las rutas y componentes que requiere el Geo portal.

En Laravel las rutas se crean en el archivo web.php las cuales hacen uso del controlador para redireccionar las diferentes vistas mediante funciones creadas en este, desde el controlador se puede traer la información de la base de datos y mostrarla mediante la ruta especificada, la cual permite la visualización de esta en pantalla.

Figura 32. Rutas en el archivo web.php.

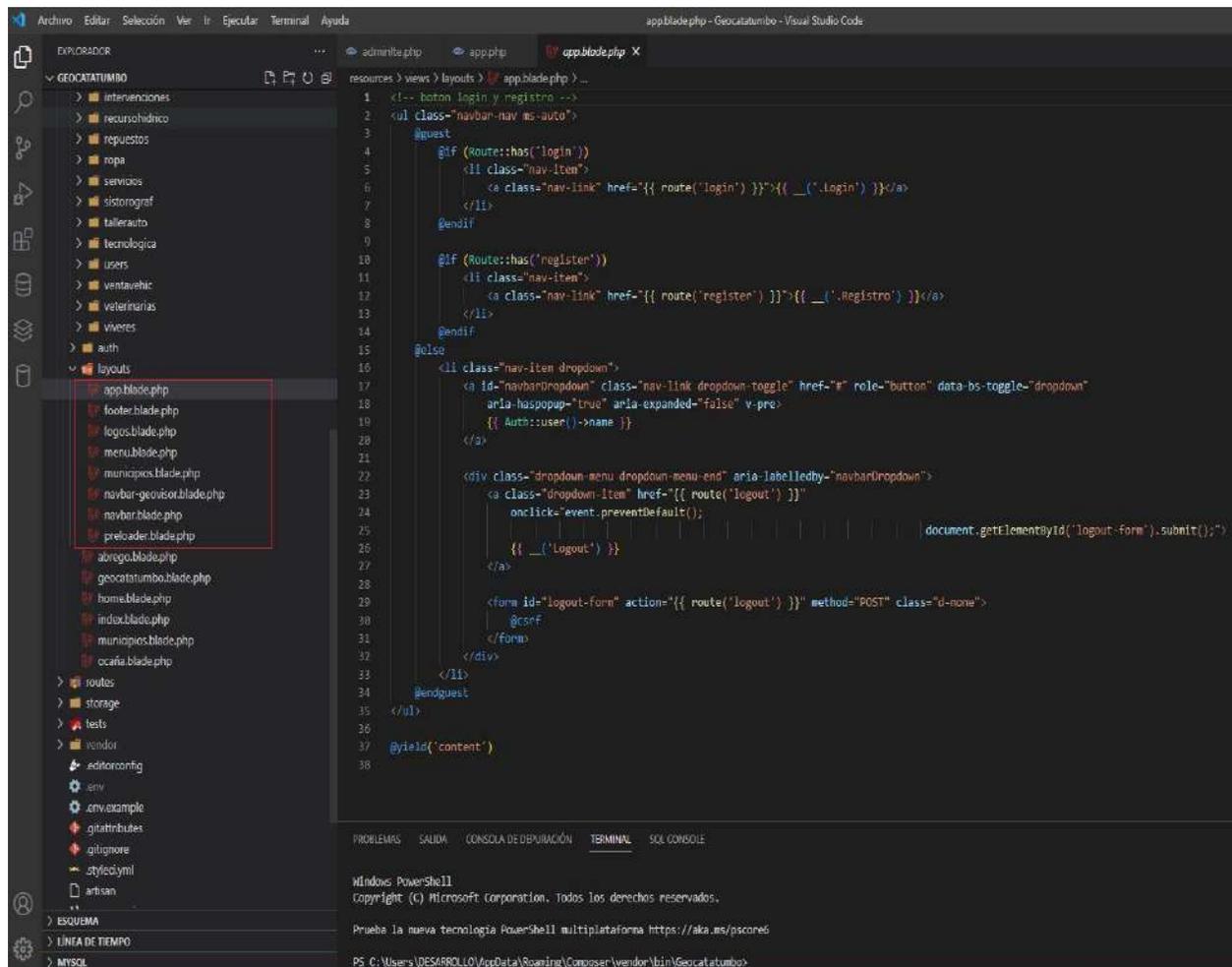
```

19
20
21 Route::get('/', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
22 Route::get('/geocatatumbo', [App\Http\Controllers\HomeController::class, 'geocatatumbo'])->name('geocatatumbo');
23 Route::get('/municipios', [App\Http\Controllers\HomeController::class, 'municipios'])->name('municipios');
24
25 //mapas
26 Route::get('/ocaña', [App\Http\Controllers\HomeController::class, 'ocaña'])->name('ocaña');
27 Route::get('/abrego', [App\Http\Controllers\HomeController::class, 'abrego'])->name('abrego');
28
29
30 //pedir iniciar sesion al entrar a home
31 Route::get('/home', function(){
32     return view('home');
33 });
34
35 //si se habilita esta ruta no se tiene seguridad ya que cualquiera puede entrar a una ruta sin login
36 // Route::get('/home', [App\Http\Controllers\HomeController::class, 'home'])->name('home');
37
38
39 Route::get('/admin/users/index', [App\Http\Controllers\Admin\UsersController::class, 'index'])->name('admin.users.index');
40 Route::post('/admin/users/store', [App\Http\Controllers\Admin\UsersController::class, 'store'])->name('admin.users.store');
41 Route::post('/admin/users/{postId}/update', [App\Http\Controllers\Admin\UsersController::class, 'update'])->name('admin.users.update');
42 Route::delete('/admin/users/{postId}/delete', [App\Http\Controllers\Admin\UsersController::class, 'delete'])->name('admin.users.delete');
43
44 Route::get('/admin/arbolados/index', [App\Http\Controllers\Admin\ArboladosController::class, 'index'])->name('admin.arbolados.index');
45 Route::post('/admin/arbolados/store', [App\Http\Controllers\Admin\ArboladosController::class, 'store'])->name('admin.arbolados.store');
46 Route::post('/admin/arbolados/{postId}/update', [App\Http\Controllers\Admin\ArboladosController::class, 'update'])->name('admin.arbolados.update');
47 Route::delete('/admin/arbolados/{postId}/delete', [App\Http\Controllers\Admin\ArboladosController::class, 'delete'])->name('admin.arbolados.delete');
48
49 Route::get('/admin/recursohidrico/index', [App\Http\Controllers\Admin\RecursohidricoController::class, 'index'])->name('admin.recursohidrico.index');
50 Route::post('/admin/recursohidrico/store', [App\Http\Controllers\Admin\RecursohidricoController::class, 'store'])->name('admin.recursohidrico.store');
51 Route::post('/admin/recursohidrico/{postId}/update', [App\Http\Controllers\Admin\RecursohidricoController::class, 'update'])->name('admin.recursohidrico.update');
52 Route::delete('/admin/recursohidrico/{postId}/delete', [App\Http\Controllers\Admin\RecursohidricoController::class, 'delete'])->name('admin.recursohidrico.delete');
53
54 Route::get('/admin/intervenciones/index', [App\Http\Controllers\Admin\IntervencionesController::class, 'index'])->name('admin.intervenciones.index');
55 Route::post('/admin/intervenciones/store', [App\Http\Controllers\Admin\IntervencionesController::class, 'store'])->name('admin.intervenciones.store');
56 Route::post('/admin/intervenciones/{postId}/update', [App\Http\Controllers\Admin\IntervencionesController::class, 'update'])->name('admin.intervenciones.update');
57 Route::delete('/admin/intervenciones/{postId}/delete', [App\Http\Controllers\Admin\IntervencionesController::class, 'delete'])->name('admin.intervenciones.delete');
58
59 Route::get('/admin/funcionamientos/index', [App\Http\Controllers\Admin\FuncionamientosController::class, 'index'])->name('admin.funcionamientos.index');
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

En la programación se busca hacer uso de las buenas prácticas para que el software sea fácil, rápido y seguro de desarrollar. En algunas ocasiones es necesario reutilizar código para no crear las mismas líneas de código en todas las vistas, para ello se crea un componente y se hace el llamado de este en la vista que se quiera visualizar. A continuación, se puede observar los componentes como el footer, logos, menú, municipios, navbar y preloader.

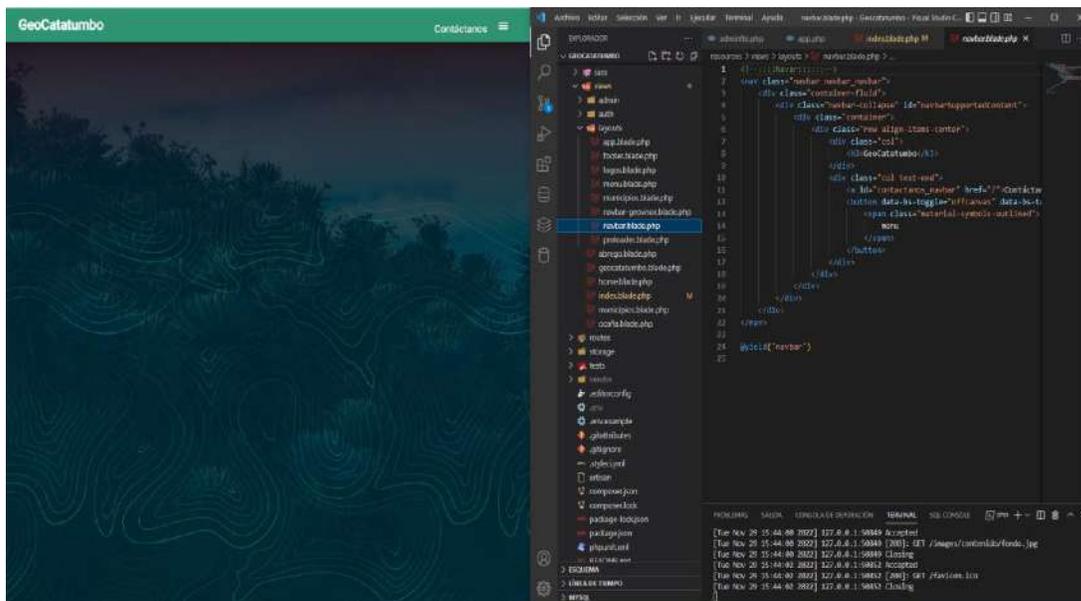
Figura 33. Componentes.



- **Actividad 6.** Desarrollar las funcionalidades requeridas para la interacción del usuario con el Geo portal.

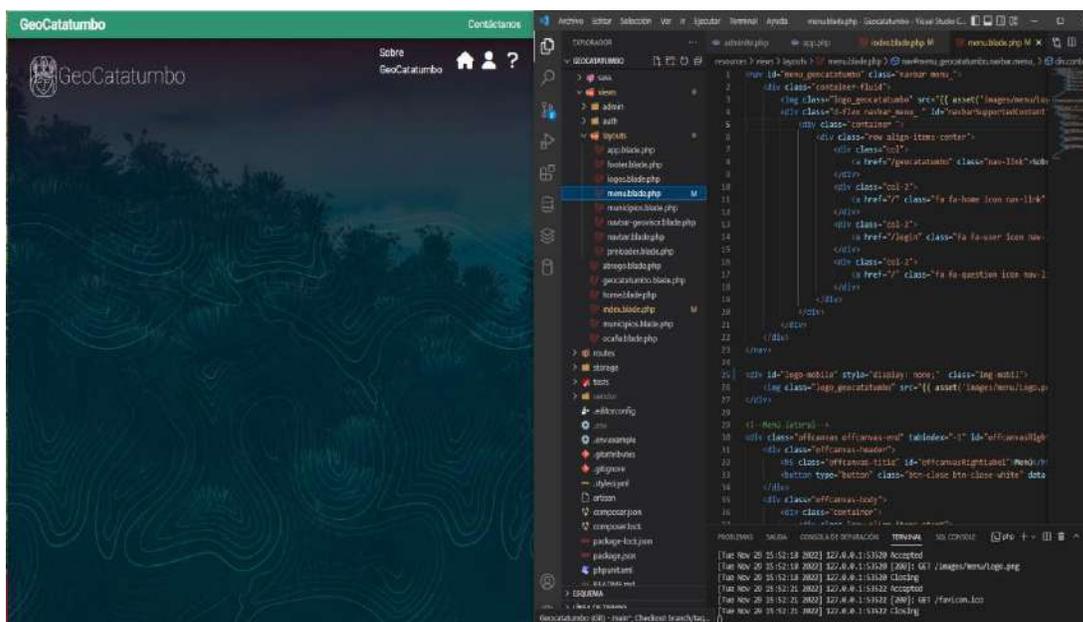
Todo sitio web debe contar con un encabezado, donde se muestre el nombre de este y un apartado para visualizar los contactos e información relevante para el usuario.

Figura 34. Navbar o banner.



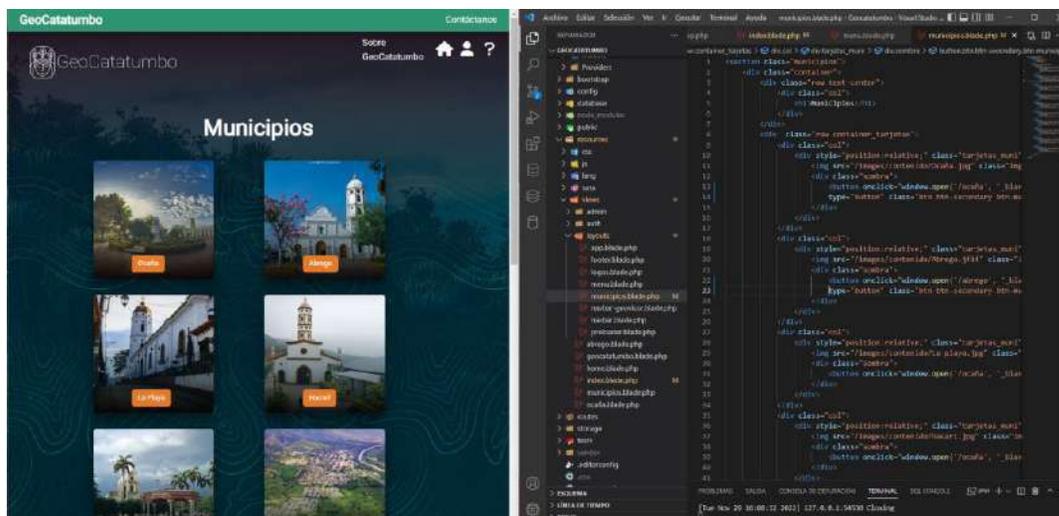
Se creó un menú el cual contiene el logo de geo Catatumbo y los iconos que redirigen a las diferentes vistas como información, inicio, login y ayuda.

Figura 35. Barra Menú.



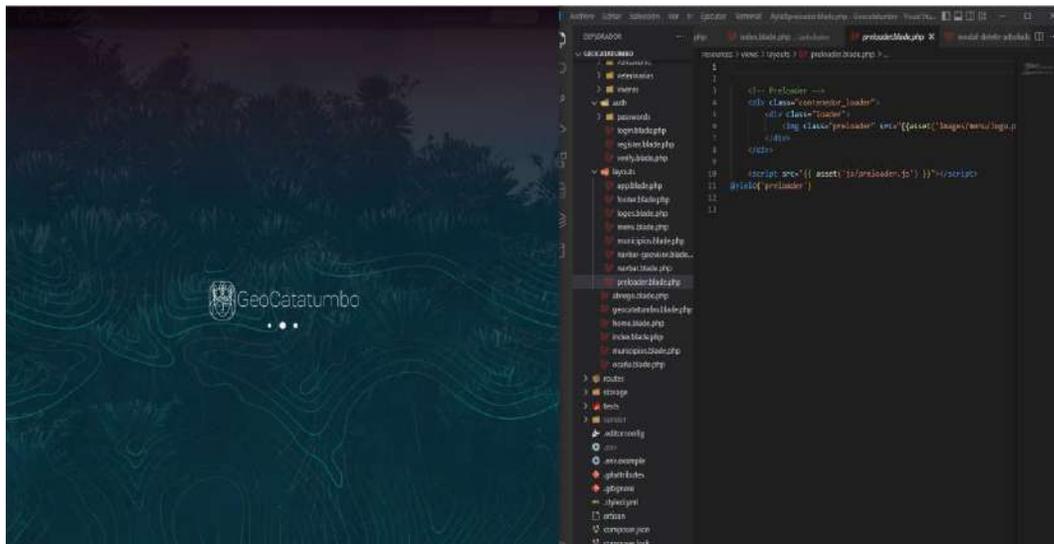
En este apartado se crearon once tarjetas las cuales contienen la imagen de cada municipio de norte Santander, estas permiten redirigir hacia el Geovisor generando una vista de cada mapa con la descripción de las zonas, barrios, puntos de Georreferenciación entre otros.

Figura 36. Tarjetas con Municipios.



Antes de cargar cada mapa en el Geovisor se creó una imagen de carga para indicar al usuario que debe esperar un pequeño lapso de tiempo mientras se carga el mapa de esta forma poder visualizar todas sus funciones.

Figura 37. Preload (Imagen de carga).



El Geovisor contiene un menú con el nombre del geoportal, un apartado de contactos, un botón que permite redirigir hacia la vista anterior, además de ello se muestra el mapa del municipio seleccionado con su descripción y los puntos de geolocalización.

Figura 38. Geovisor con Mapa Ocaña, descripción y puntos de georreferenciación.

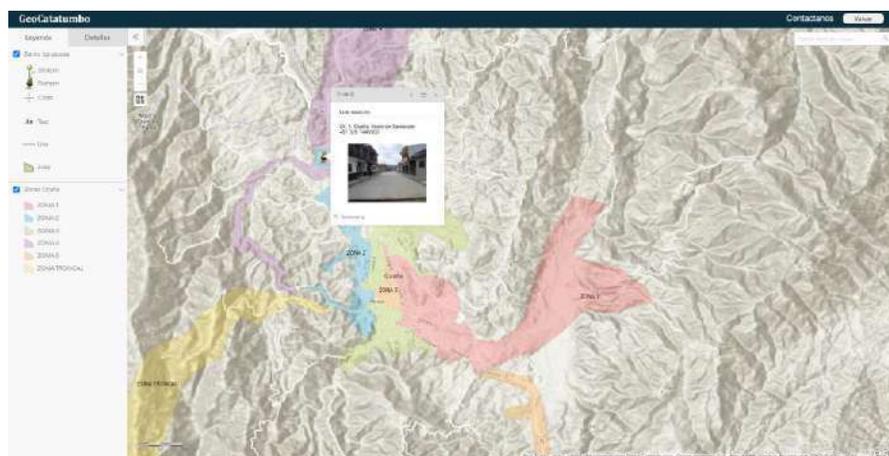
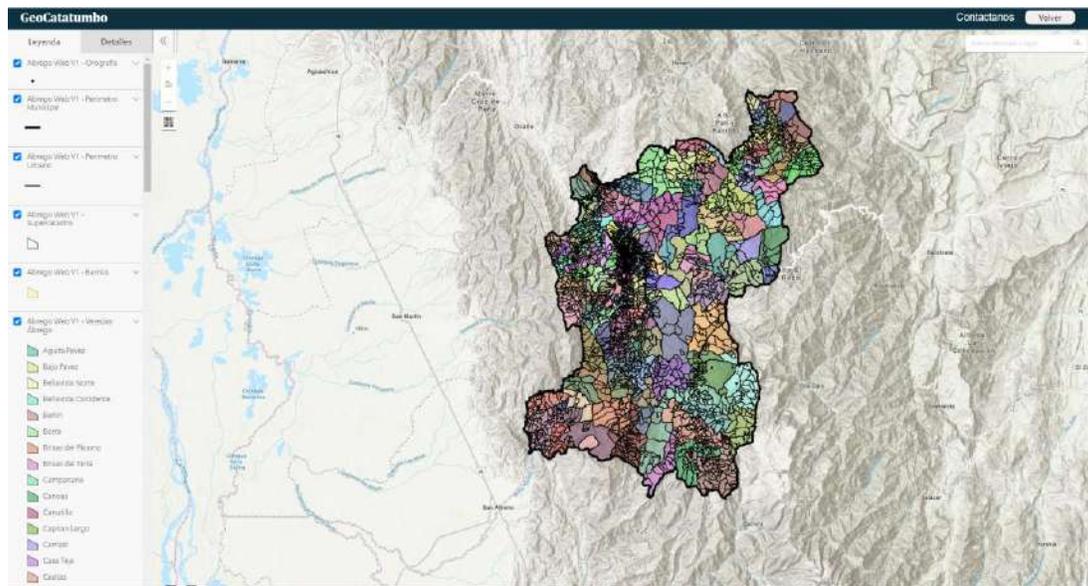


Figura 39. Geovisor con Mapa Abrego y descripción de los Barrios.



El menú lateral derecho permite redirigir hacia las diferentes secciones del geoportal.

Figura 40. Menú Lateral Derecho.



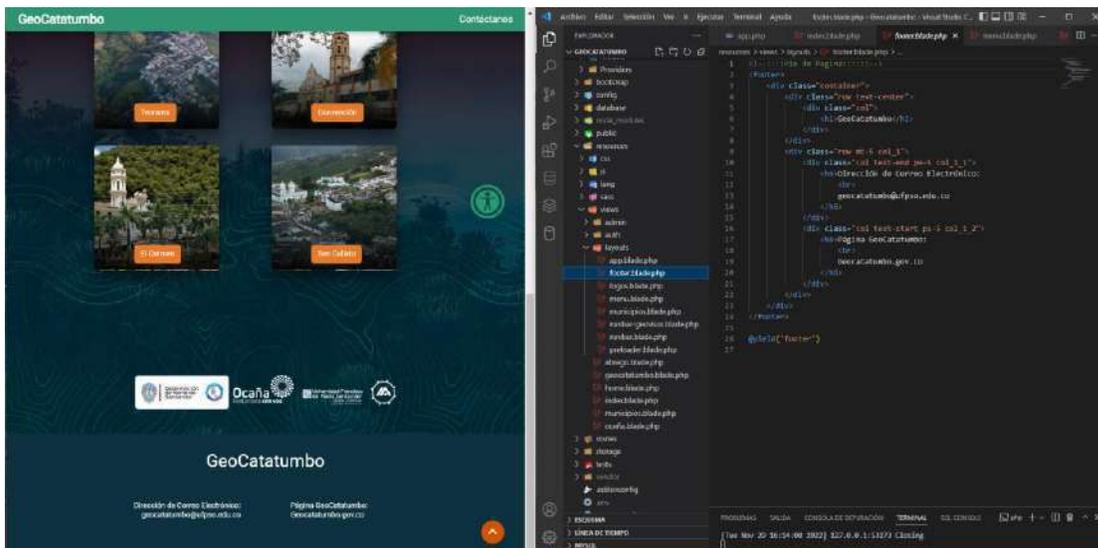
El botón de accesibilidad contiene una serie de gadgets los cuales permiten que el usuario pueda cambiar el contraste de la pantalla, tamaño de letra, espaciado, saturación entre otros.

Figura 41. Botón Accesibilidad.



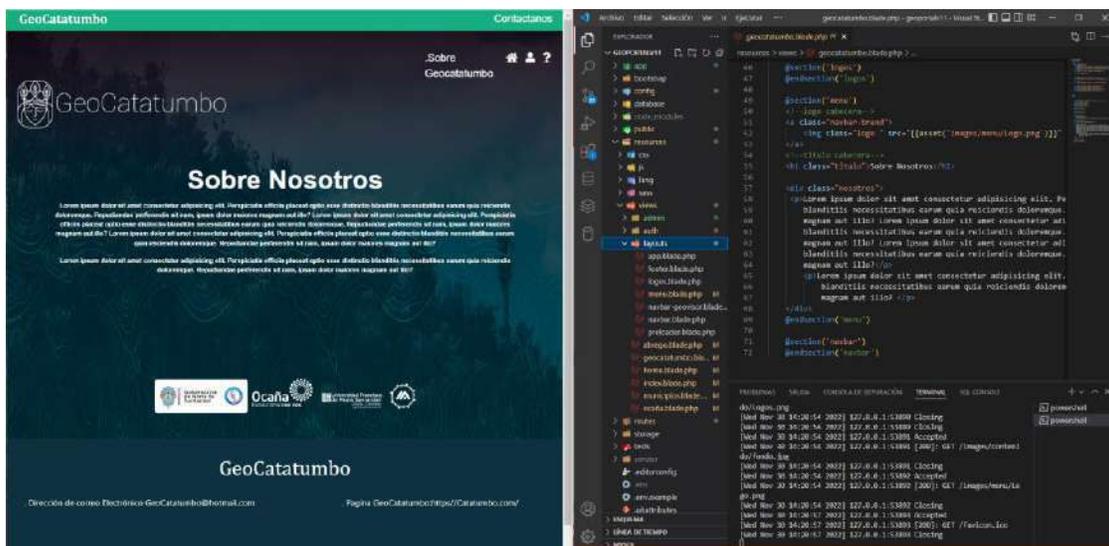
Figura 42. Logos pie página.

Figura 43. Footer (Pie de página).



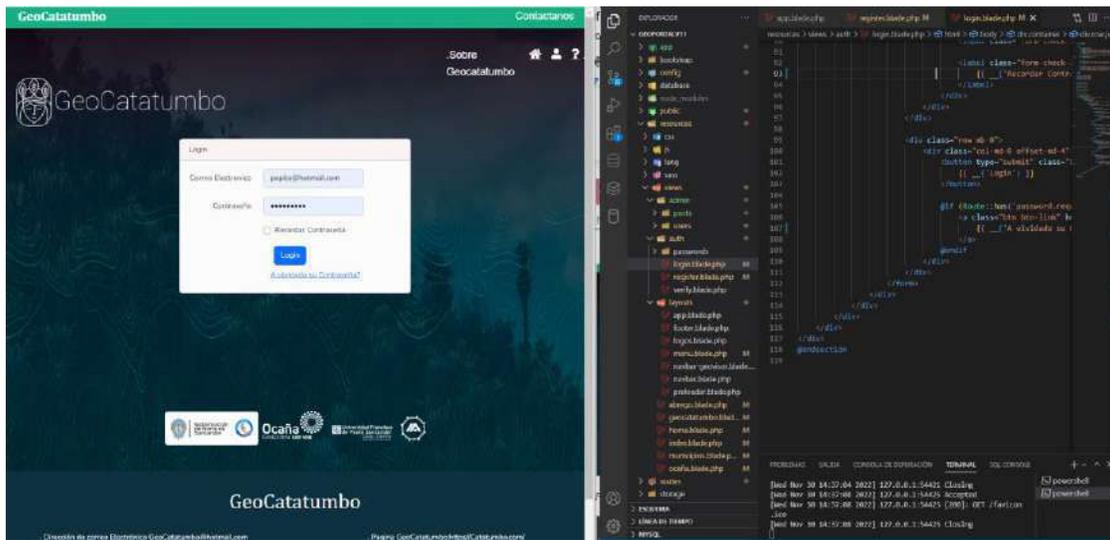
El apartado de Geocatatumbo contiene información sobre “nosotros”, el cual permite que el usuario tenga un amplio conocimiento sobre el centro de desarrollo CEDIT y la UFPSO.

Figura 44. Sobre Geocatatumbo.



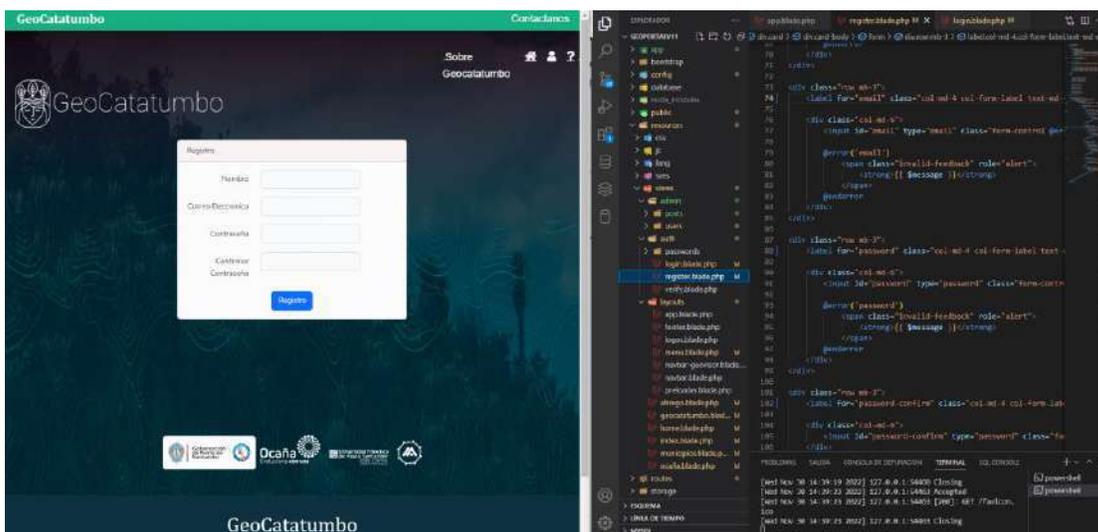
El login permite restringir el acceso para los usuarios que no estén registrados en la base de datos, de esta forma se puede tener un control de quienes ingresan al geoportal.

Figura 45. Formulario de Login.



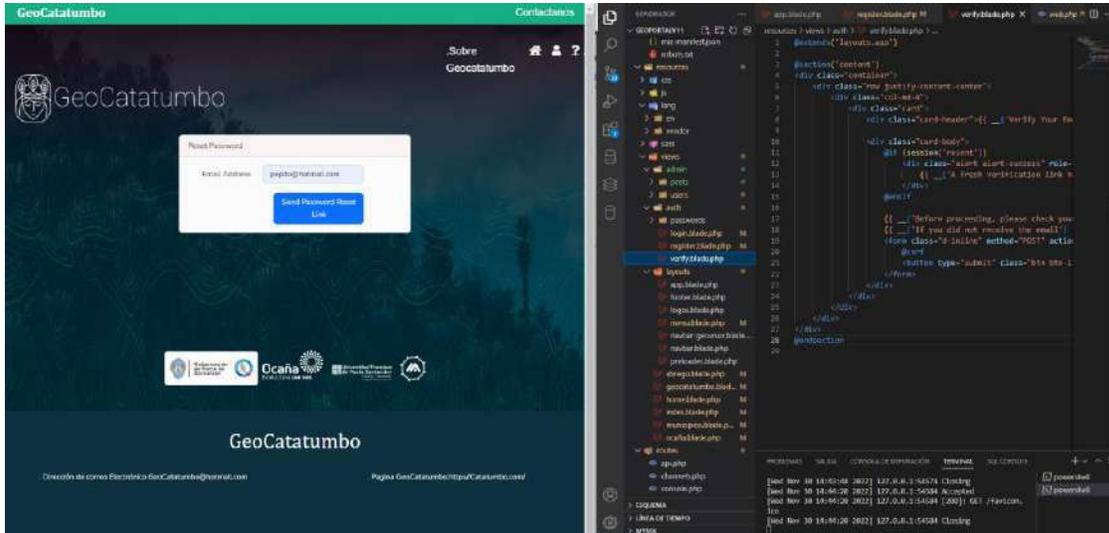
El Formulario de registro permite que los usuarios creen su cuenta en el geoportal para poder acceder a este.

Figura 46. Formulario de Registro.



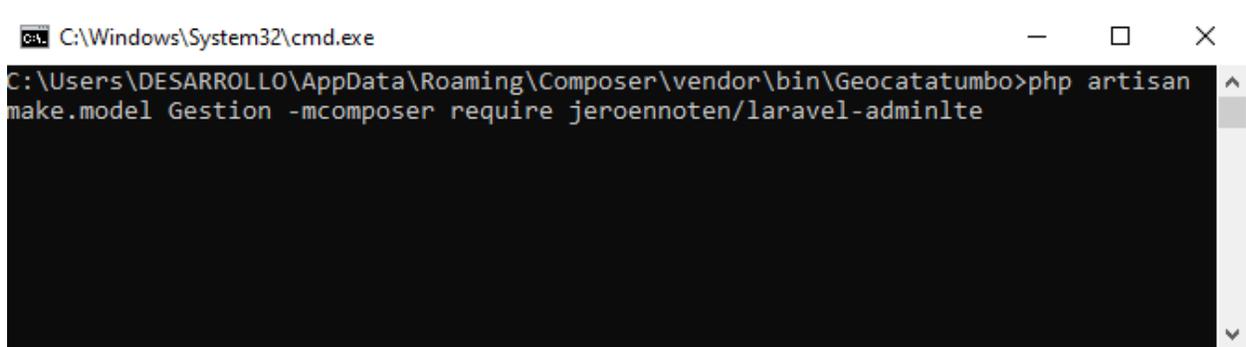
En caso de olvidar la contraseña hay una opción que permite recuperar la contraseña enviando un mensaje al correo del usuario con el que creó la cuenta en el geoportal.

Figura 47. Formulario de Recuperación de contraseña.



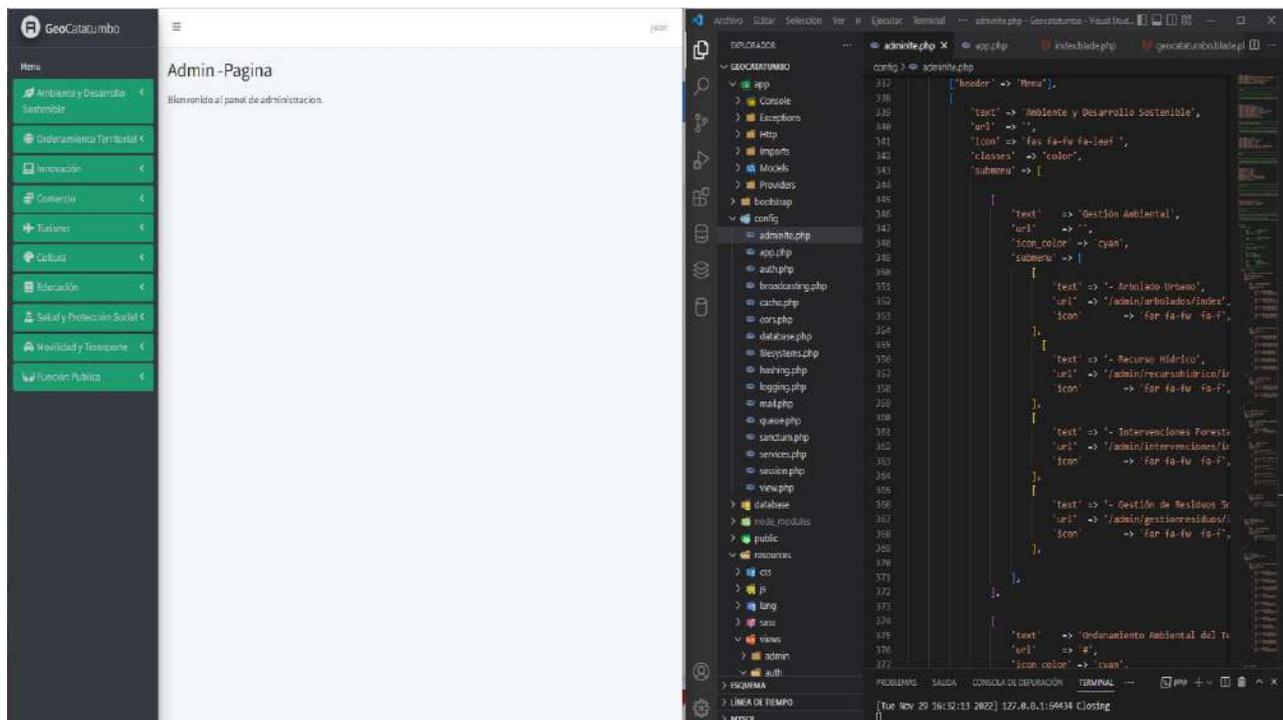
Para la creación del panel de administración se ejecuta la siguiente línea de código en la consola cmd, la cual genera una interfaz del panel administración.

Figura 48. Creación Panel administración.



El panel de administración contiene un menú lateral izquierdo en el que se describen todas las categorías de las tablas que contienen la información de los puntos de georreferenciación en cada mapa.

Figura 49. Panel administración.



El panel de administración contiene un menú con una serie de categorías las cuales tienen un submenú y a su vez cada apartado que contiene una tabla con la información que tendrá cada punto de georreferenciación de los mapas de cada municipio de norte Santander.

Figura 50. Tabla Arbolado Urbano.

The image shows two side-by-side screenshots. The left screenshot displays the 'Crear' form for 'Arbolado Urbano' in the GeoCataluña application. The form has a table header with columns: 'Intervención', 'Fecha de Intervención', 'Estado del Individuo', 'Ubicación Geográfica', and 'Acciones'. Below the header, there are input fields for 'idInt', 'valor', 'estado', and 'idUbic'. The right screenshot shows the PostgreSQL database schema for the 'Arbolado Urbano' table. The schema includes columns: 'idInt', 'valor', 'estado', 'idUbic', and 'acciones'. The schema also shows the table's structure, including the 'acciones' column and the 'idInt' column.

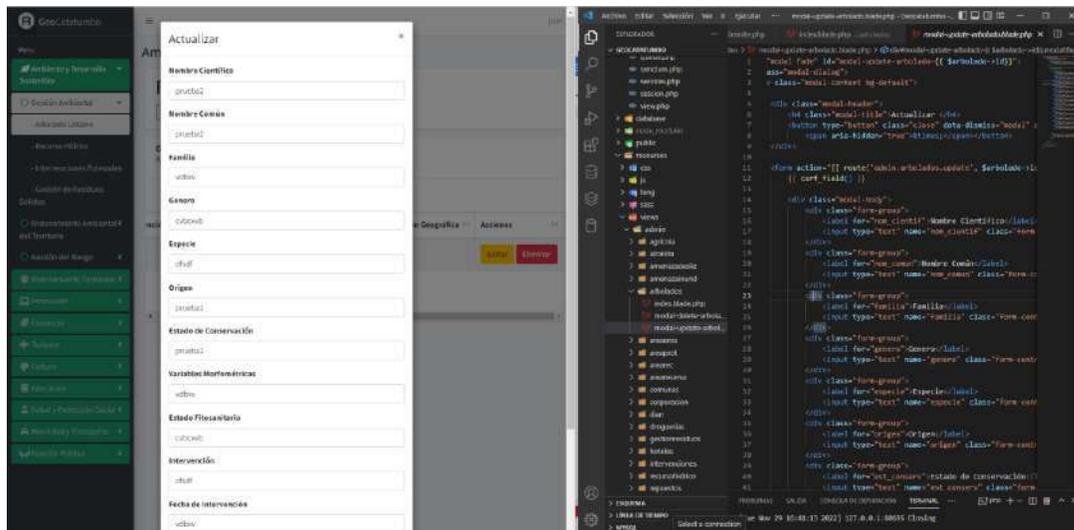
El formulario de registro de datos contiene el nombre de cada columna de la tabla que se encuentra almacenada en la base de datos postgres.

Figura 51. Formulario para Crear Datos.

The image shows two side-by-side screenshots. The left screenshot displays the 'Crear' form for 'Arbolado Urbano' in the GeoCataluña application. The form has a table header with columns: 'Nombre Científico', 'Nombre Común', 'Familia', 'Genero', 'Especie', 'Origen', 'Estado de Conservación', 'Variables Morfológicas', 'Estado Filogenético', 'Intervención', and 'Fecha de Intervención'. Below the header, there are input fields for 'nombreCientifico', 'nombreComun', 'familia', 'genero', 'especie', 'origen', 'estadoConservacion', 'variablesMorfologicas', 'estadoFilogenetico', 'intervencion', and 'fechaIntervencion'. The right screenshot shows the PostgreSQL database schema for the 'Arbolado Urbano' table. The schema includes columns: 'nombreCientifico', 'nombreComun', 'familia', 'genero', 'especie', 'origen', 'estadoConservacion', 'variablesMorfologicas', 'estadoFilogenetico', 'intervencion', and 'fechaIntervencion'. The schema also shows the table's structure, including the 'acciones' column and the 'idInt' column.

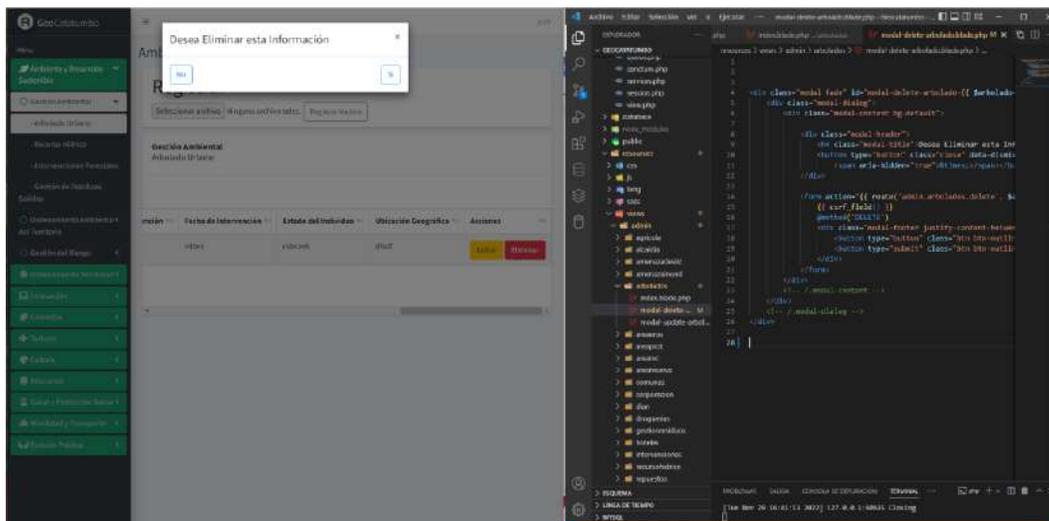
El formulario de actualizar datos contiene el nombre de cada columna de la tabla que se encuentra almacenada en la base de datos postgres.

Figura 52. Formulario para Editar Datos.



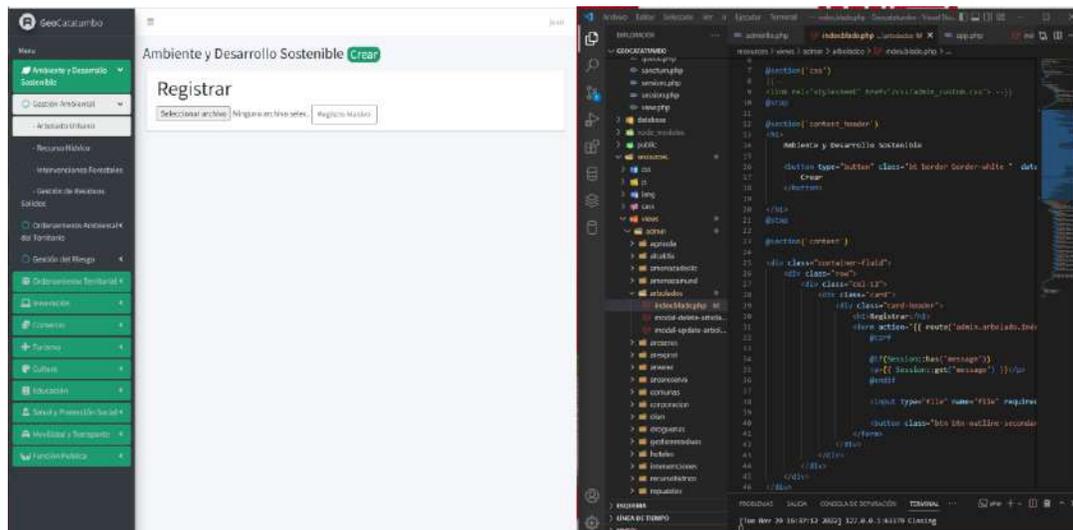
El formulario eliminar cuenta con una validación de datos para confirmar si el usuario desea eliminar la información o desea cancelar su petición.

Figura 53. Formulario para Eliminar Datos.



El registro masivo permite realizar la carga de una tabla de Excel con la información que se quiera cargar sin la necesidad de crear uno a uno cada dato que contendrá la tabla.

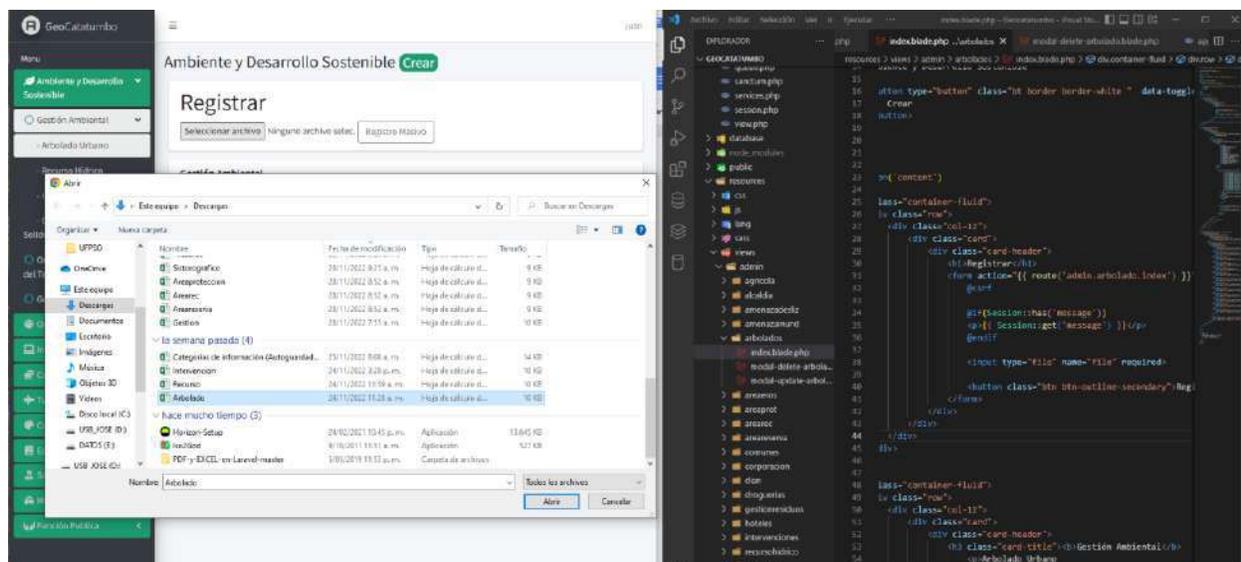
Figura 54. Formulario Registro Masivo.



El archivo Excel debe contener la información de cada columna de la tabla para poder

cargarlo.

Figura 55. Selección Archivo Registro Masivo.



Durante el desarrollo del geoportal se realizaron diferentes funcionalidades las cuales permiten que este tenga una interfaz agradable al usuario, las cuales se lograron haciendo uso de lenguajes de programación como PHP, JVS y HTML. Además de esto es necesario hacer uso de patrones de diseño los cuales permiten que el código se mantenga ordenado, sin que se repitan líneas de código y optimizándolo ya que de esta forma será más eficaz.

3.1.3 Objetivo 3. Realizar pruebas de los cambios implementados en el Geoportal que permitan verificar el correcto funcionamiento del mismo.

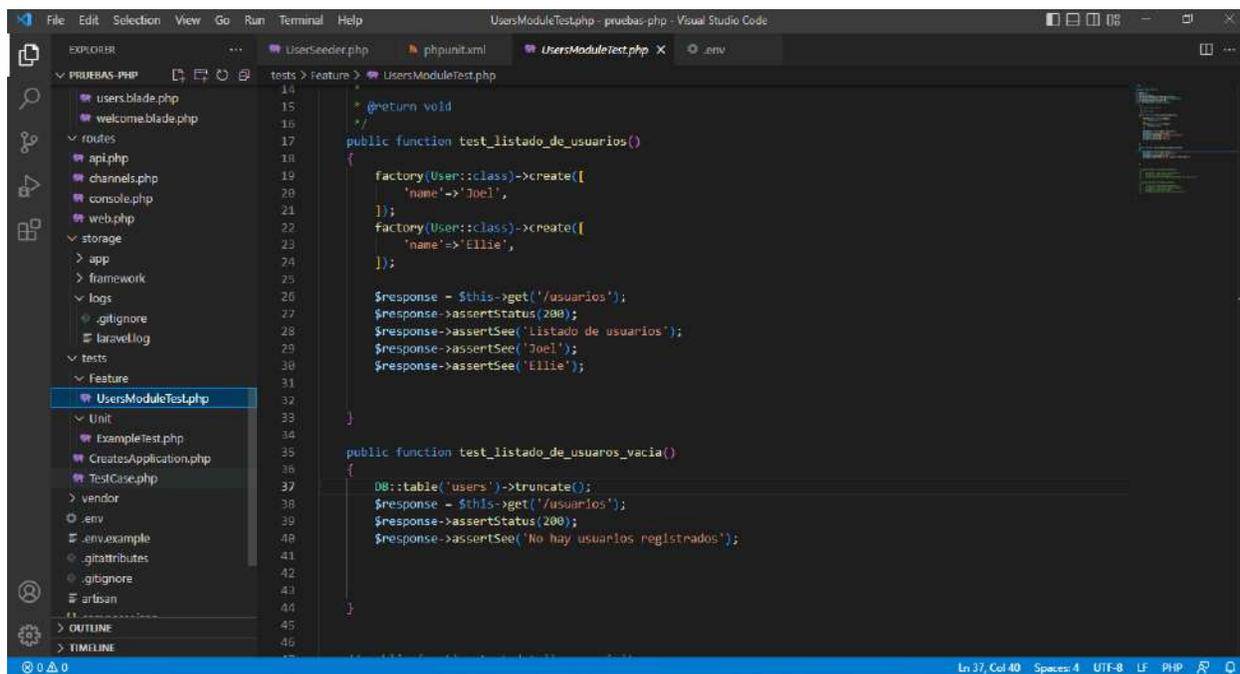
- **Actividad 7.** Probar las distintas funcionalidades del Geoportal y solucionar posibles errores.

Las pruebas de un software permiten determinar que un proyecto esté funcionando correctamente y que puede seguir funcionando, cuando se trabaja en un proyecto independientemente de cuál sea el tipo de framework y se están introduciendo cambios es

posible que sin querer se pueda dañar alguna parte del proyecto.

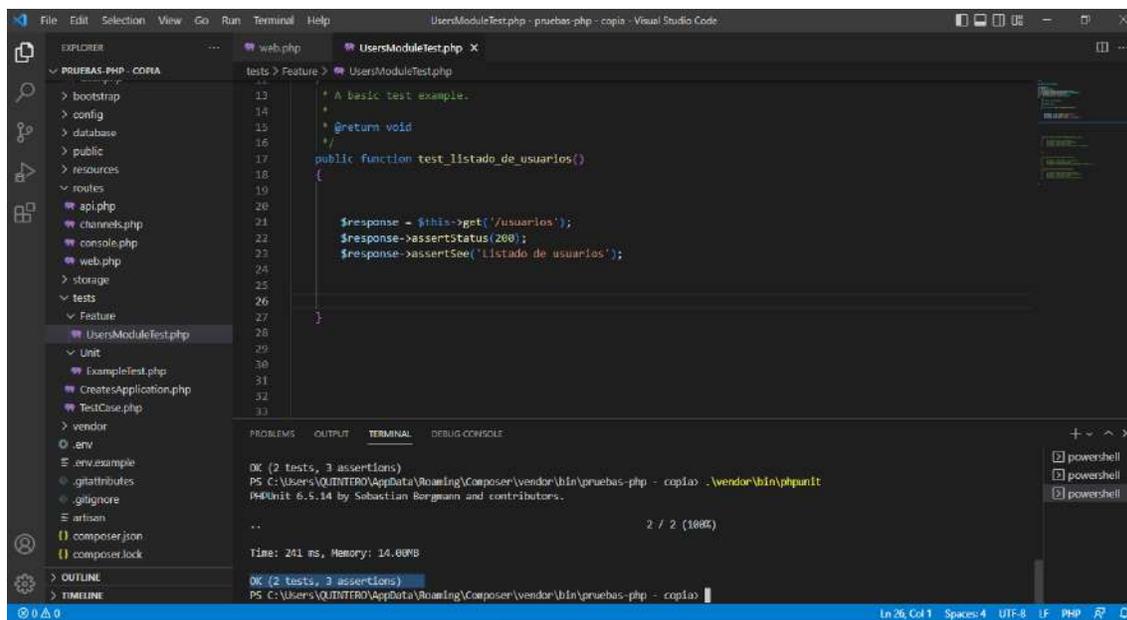
Para encontrar este tipo de errores, Laravel cuenta con un componente llamado `laravel.log` que permite visualizar el error y la fecha exacta en la que se generó, por otro lado, Laravel permite realizar pruebas del proyecto de manera automatizada. Laravel incluye en el directorio principal del proyecto, un directorio llamado `tests` o pruebas, en este directorio se escribe el código que se encarga de probar el código del resto del proyecto, este se encuentra separado en dos subdirectorios el directorio `Feature` donde se van a escribir las pruebas que van a simular peticiones `Http` al servidor y `Unit` donde se van a escribir las pruebas que van a probar partes individuales del proyecto, como las clases y métodos.

Figura 56. Directorio Feature y Unit.



Cuando se ejecutan las pruebas Php.unit se recorren todas las clases dentro del directorio tests que tengan el sufijo test y se ejecutan las pruebas en cada uno de estos métodos. En la carpeta Feature se tiene una prueba llamada UserModuleTest que va a ejecutar la prueba test_listado_de_usuarios, el código dentro de la prueba realiza una petición http a la Url de usuarios mediante el controlador, donde hace referencia a la función index y luego va a comprobar que al ejecutar la prueba se obtiene el estado Http 200, lo que indica que todo salió bien, que la Url existe y cargo correctamente.

En caso de que la Url no concida con la Url del test, al ejecutar las pruebas se obtiene un error donde dice que se espera un código 200 y recibió un código 404, lo que quiere decir que la url ya no existe.

Figura 57. Test o prueba.

The screenshot shows the Visual Studio Code interface. The Explorer on the left shows a project structure with a 'tests' directory containing a 'Feature' directory, which in turn contains 'UsersModuleTest.php'. The main editor displays the code for 'UsersModuleTest.php' with the following content:

```
13 * A basic test example.  
14 *  
15 * @return void  
16 */  
17 public function test_listado_de_usuarios()  
18 {  
19  
20  
21 $response = $this->get('/usuarios');  
22 $response->assertStatus(200);  
23 $response->assertSee('listado de usuarios');  
24  
25  
26 }  
27  
28  
29  
30  
31  
32  
33
```

The bottom panel shows the terminal output of the test execution:

```
OK (2 tests, 3 assertions)  
PS C:\Users\QUINTERO\AppData\Roaming\Composer\vendor\bin\pruebas-php - copia> .\vendor\bin\phpunit  
PHPUnit 6.5.14 by Sebastian Bergmann and contributors.  
  
..                                     2 / 2 (100%)  
  
Time: 241 ms, Memory: 14.60MB  
OK (2 tests, 3 assertions)  
PS C:\Users\QUINTERO\AppData\Roaming\Composer\vendor\bin\pruebas-php - copia>
```

En el archivo web.php se encuentra la ruta /usuarios que debe coincidir con la ruta que se encuentra en el test de prueba. A las rutas se les puede agregar una función que devuelve una cadena de texto o también se puede hacer referencia al controlador el cual redirige a la función index y devuelve una respuesta a la vista.

Figura 58. Ruta /usuarios.

```

1 <?php
2
3
4
5 Route::get('/', function () {
6     return 'Rome';
7 });
8
9 Route::get('/usuarios', function () {
10    return 'Usuarios';
11 });
12 Route::get('/usuarios/{id}', 'UserController@show')
13     ->where('id', '[0-9]+');
14
15
16 Route::get('/usuarios/nuevo', 'UserController@create');
17
18
19 Route::get('/saludo/{name}/{nickname}', 'WelcomeUserController');

```

```

[Thu Feb 2 12:56:38 2023] 127.0.0.1:62163 Accepted
[Thu Feb 2 12:56:38 2023] 127.0.0.1:62163 Closing
[Thu Feb 2 12:56:38 2023] 127.0.0.1:62162 [200]: GET /favicon.ico
[Thu Feb 2 12:56:38 2023] 127.0.0.1:62162 Closing
[Thu Feb 2 12:56:55 2023] 127.0.0.1:62165 Accepted
[Thu Feb 2 12:56:55 2023] 127.0.0.1:62166 Accepted
[Thu Feb 2 12:56:55 2023] 127.0.0.1:62165 Closing
[Thu Feb 2 12:56:55 2023] 127.0.0.1:62166 [200]: GET /favicon.ico
[Thu Feb 2 12:56:55 2023] 127.0.0.1:62166 Closing

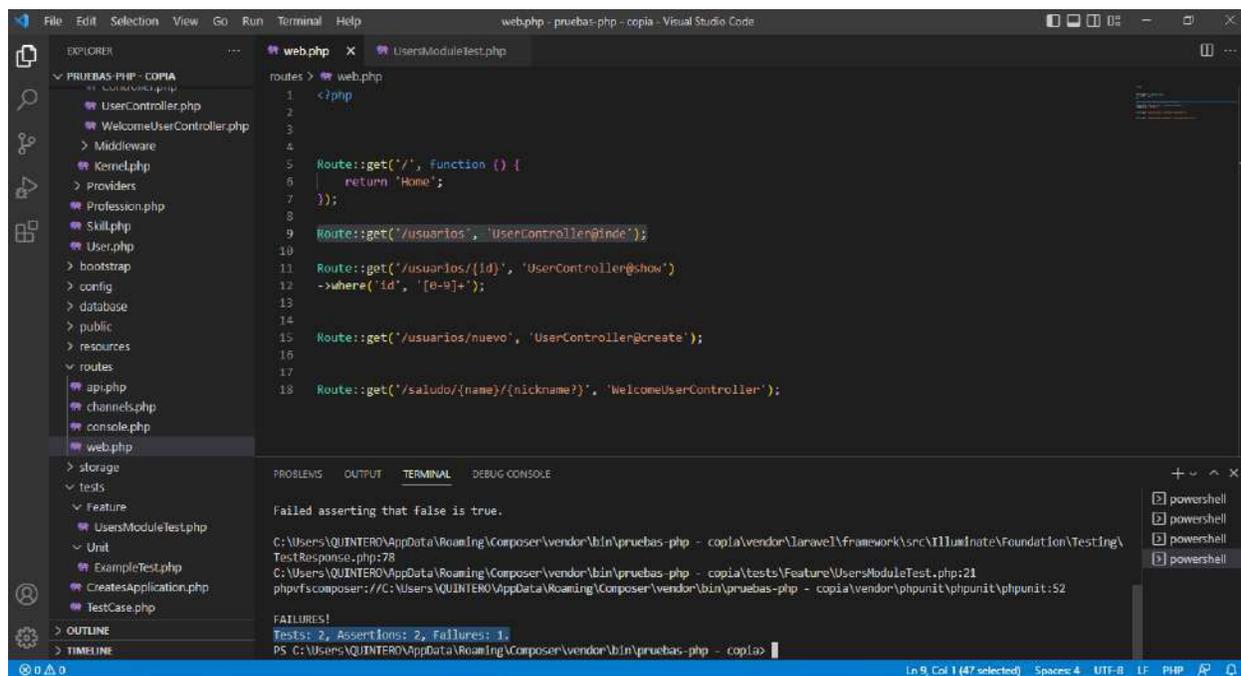
```

En este caso se puede observar que esta retornando la cadena de texto con el nombre “usuarios”.

Figura 59. Cadena de texto en pantalla.



Si se modifica la ruta y se reemplaza el nombre índice por inde, se presenta un error al cargar la ruta ya que esta no es válida.

Figura 60. Error al ejecutar el Test.

Para este caso se van a crear dos usuarios en la base de datos “Joel” y “Ellie”, para ello se crean los métodos crear indicando la columna “name” que contiene los nombres de los usuarios y se realiza la migración hacia la base de datos.

Figura 61. Métodos crear usuario.

The screenshot shows the Visual Studio Code editor with the file `UserSeeder.php` open. The code defines a `factory` method for the `User` class, which is used to create users with specific attributes. The terminal output shows the successful execution of the migration command, including the creation of tables and the insertion of data.

```

29     'is_admin' => true,
30 ];
31
32     factory(User::class) -> create([
33         'profession_id' => $professionId
34     ]);
35
36     factory(User::class) -> create([
37         'name' => 'Joel',
38     ]);
39
40     factory(User::class) -> create([
41         'name' => 'Ellie',
42     ]);
43
44     factory(User::class, 48) -> create();
45 }

```

```

PS C:\Users\WJUNTERO\AppData\Roaming\Composer\vendor\bin\pruebas-php> php artisan migrate:refresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2014_11_15_000000_create_users_table
Migrated: 2014_11_15_000000_create_users_table
Migrating: 2023_01_27_164738_create_professions_table
Migrated: 2023_01_27_164738_create_professions_table
Migrating: 2023_01_27_185130_add_profession_id_to_users
Migrated: 2023_01_27_185130_add_profession_id_to_users
Migrating: 2023_01_30_181143_create_skills_table
Migrated: 2023_01_30_181143_create_skills_table
PS C:\Users\WJUNTERO\AppData\Roaming\Composer\vendor\bin\pruebas-php>

```

A continuación, se realiza la consulta a la tabla `users`, mediante `select from` para confirmar que los usuarios fueron creados correctamente.

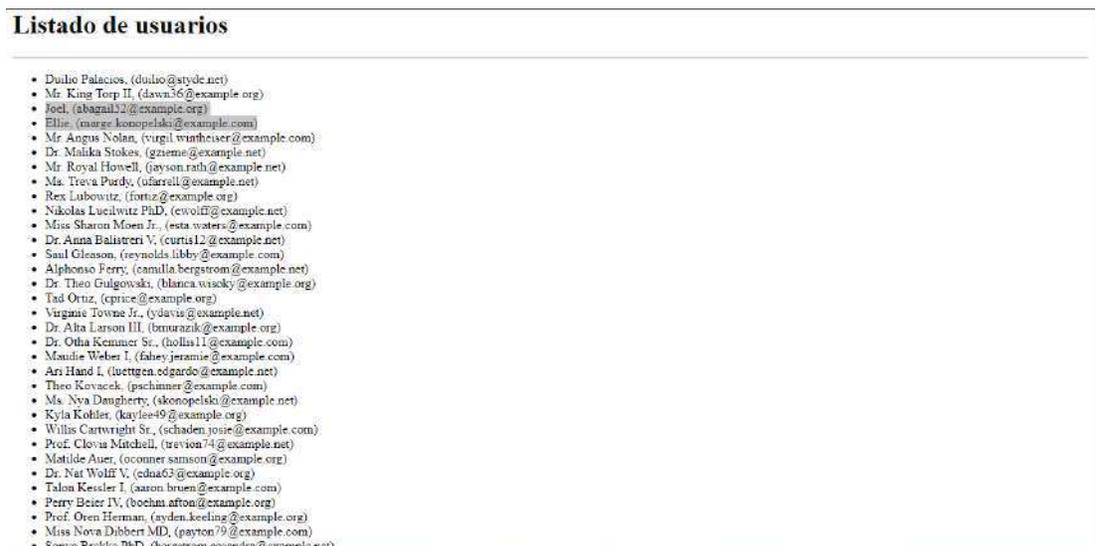
Figura 62. Usuarios creados en la base de datos.

The screenshot shows the pgAdmin interface with a SQL query executed against a PostgreSQL database. The query is `select * from users`. The results are displayed in a table with columns `id`, `name`, `email`, and `password`.

id	name	email	password
1	Dulio Palacios	dulio@etoye.net	\$2y\$10\$uFKqE8oe8NU3iPzwUJ01/dRfYhwR6hgnZuLZL874pE
2	Dr Lynn Schulist IV	anyang@example.com	\$2y\$10\$TKh8H1.P1Qx37YgCzwk8.K/N1Wq4H9cc0qdVFYg787
3	Joel	enrique4@example.org	\$2y\$10\$TKh8H1.P1Qx37YgCzwk8.K/N1Wq4H9cc0qdVFYg787
4	Ellie	ookeefe@example.com	\$2y\$10\$TKh8H1.P1Qx37YgCzwk8.K/N1Wq4H9cc0qdVFYg787
5	Mrs. Abbey Renner	hrogahn@example.net	\$2y\$10\$TKh8H1.P1Qx37YgCzwk8.K/N1Wq4H9cc0qdVFYg787
6	Mrs. Janessa Howe	kasaulka.elizabeth@example.org	\$2y\$10\$TKh8H1.P1Qx37YgCzwk8.K/N1Wq4H9cc0qdVFYg787
7	Dr Archibald Gmeier	baum.arigae@example.org	\$2y\$10\$TKh8H1.P1Qx37YgCzwk8.K/N1Wq4H9cc0qdVFYg787
8	Zane Emard	croste05@example.org	\$2y\$10\$TKh8H1.P1Qx37YgCzwk8.K/N1Wq4H9cc0qdVFYg787
9	Jassen VenKraeden	stiles.phoos@example.org	\$2y\$10\$TKh8H1.P1Qx37YgCzwk8.K/N1Wq4H9cc0qdVFYg787

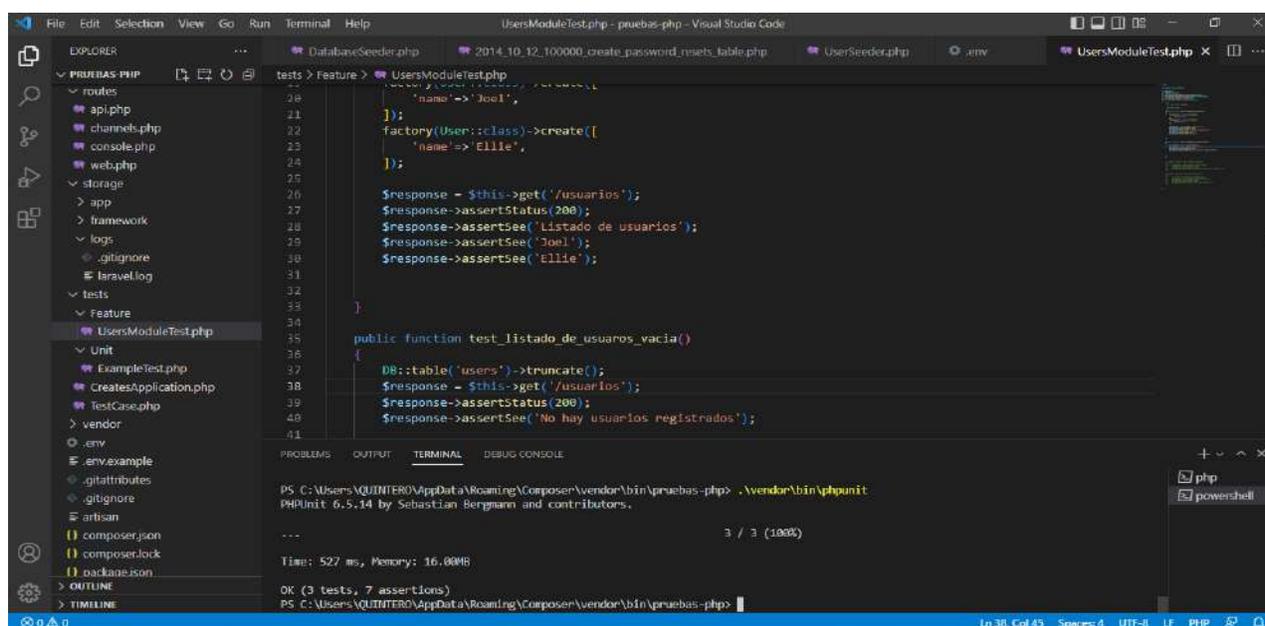
En la siguiente imagen se pueden visualizar los usuarios creados en pantalla mediante la ruta /usuarios.

Figura 63. Usuarios en pantalla.



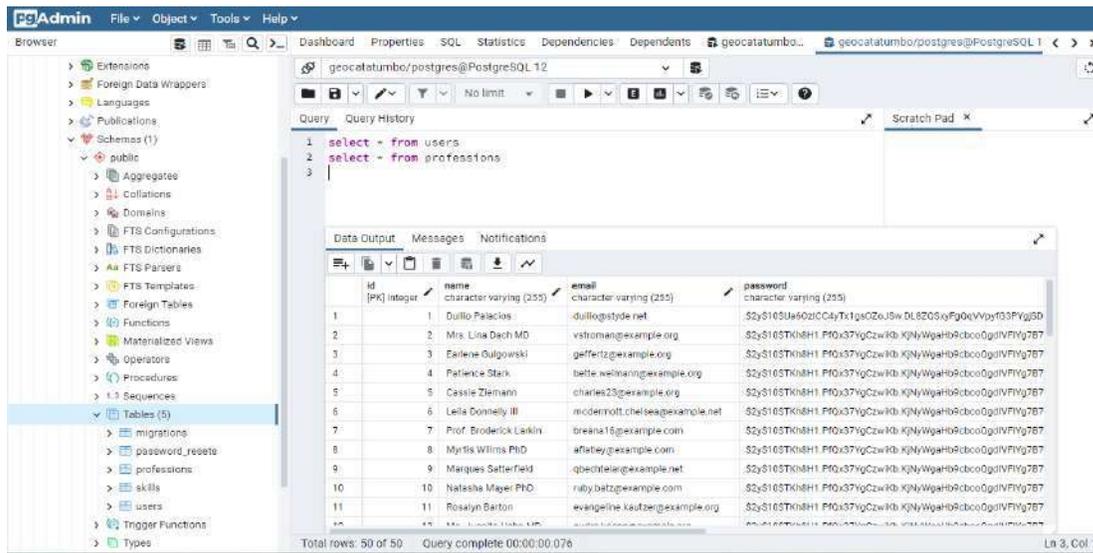
Si se realiza una prueba para buscar los usuarios creados en la base de datos, se observa que las pruebas son correctas.

Figura 64. Test usuarios creados.



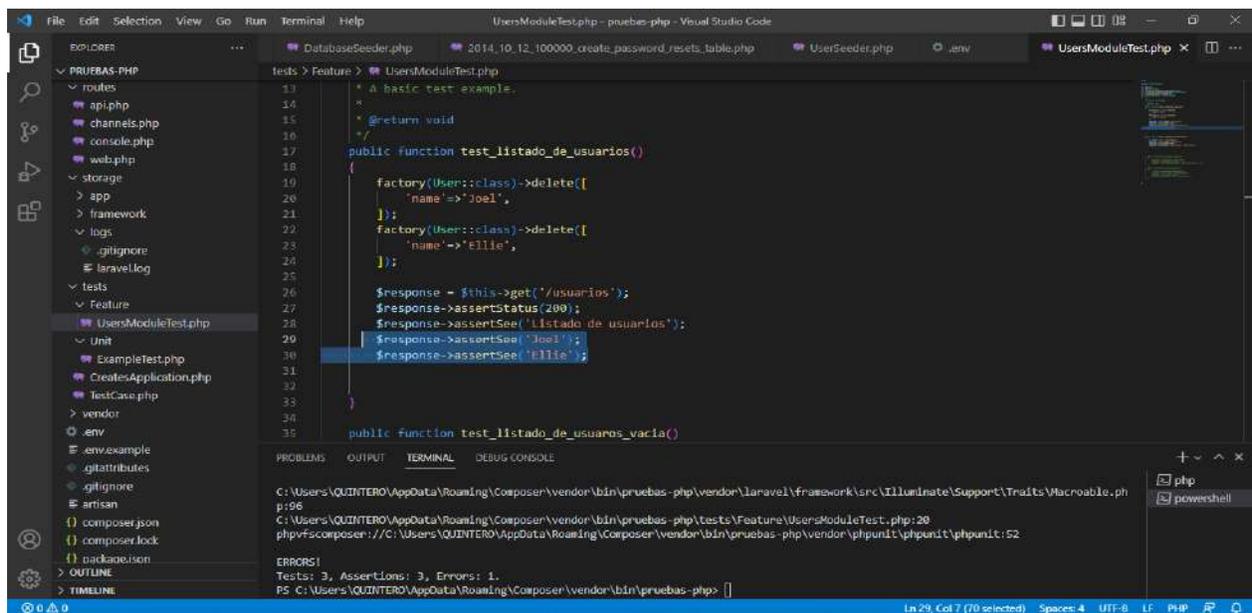
Si, por el contrario, se eliminan los usuarios y se realiza el test, este genera un error debido a que ya no se encuentran registrados en la base de datos.

Figura 65. Usuarios eliminados.



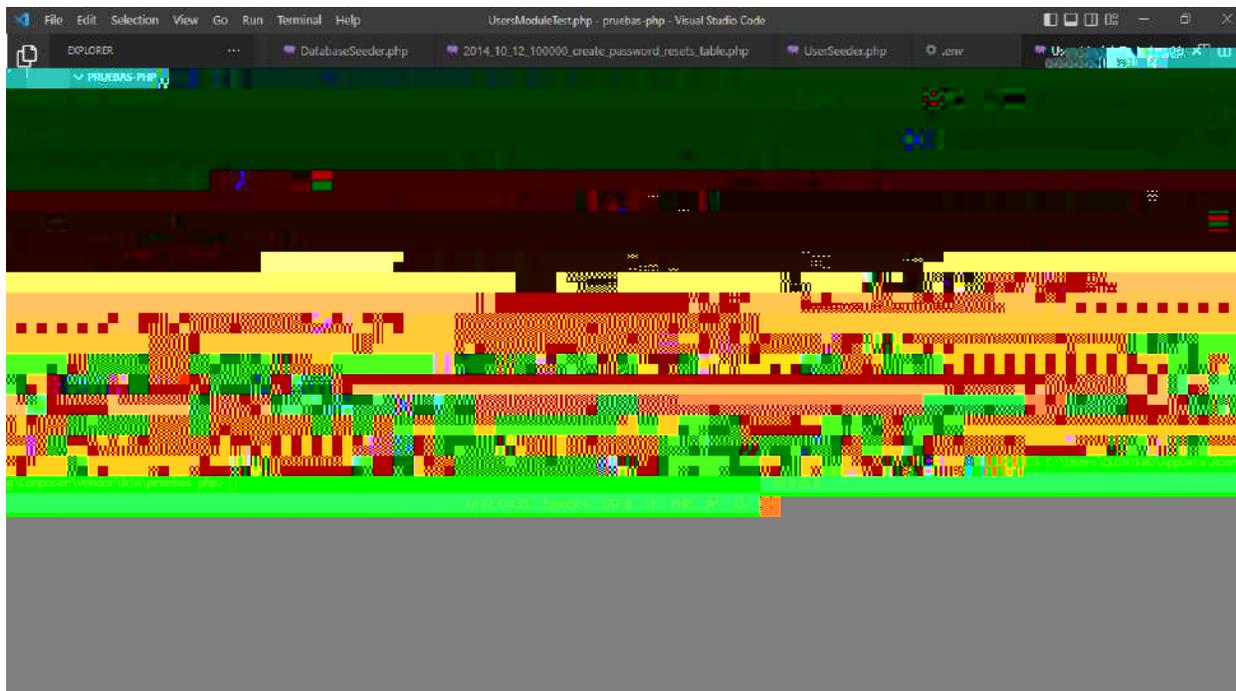
A continuación, se realiza el test con la búsqueda de los usuarios en la base de datos, lo que genera un error ya que no se encuentran registrados.

Figura 66. Búsqueda de usuarios.



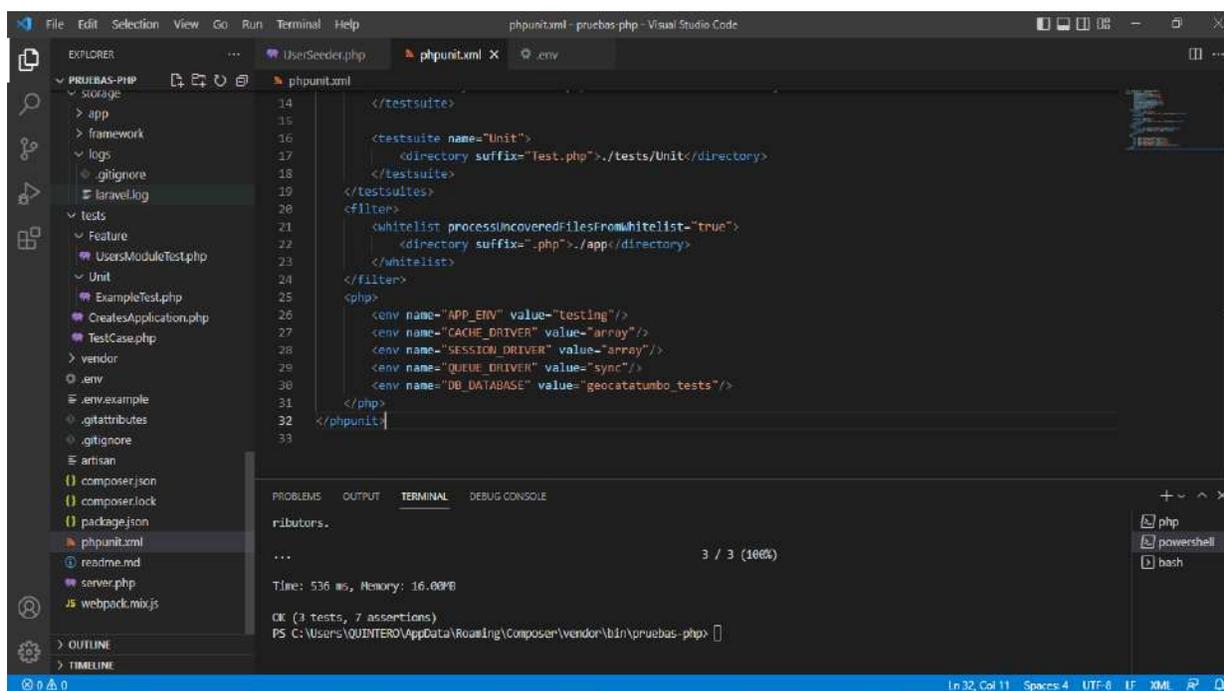
En caso de comentar las líneas de código con los usuarios “Joel ” y “Ellie” las pruebas van a pasar porque se le está indicando a la prueba que estos usuarios ya no se encuentran registrados en la base de datos.

Figura 67. Líneas de código con usuarios comentadas.



Como buena práctica se aconseja crear una base de datos dedicada para el entorno de pruebas automatizadas y otra base de datos dedicada para el entorno de bases locales para evitar que un entorno afecte al otro, ya que al realizar las pruebas si se tiene información almacenada esta se puede perder.

Figura 68. Base de datos dedicada.



Actividad 8. Verificar el correcto funcionamiento.

Para comprobar que todo funciona correctamente se realizaron pruebas de registro de usuario, inicio de sesión, visor de mapas, redirección de vistas, el crud de cada tabla y la carga masiva de documentos Excel.

Por otro lado, debido a que muchas personas utilizan dispositivos móviles fue necesario adaptar el diseño para poder visualizarlo en los diferentes dispositivos electrónicos por lo que se decidió quitar el menú lateral derecho ya que este impide que se adapte a dispositivos móviles como celulares, tablets y portátiles.

Además de ello se cambiaron algunos estilos css, por el framework bootstrap para crear las tarjetas y los iconos del menú ya que este se adapta a dispositivos móviles por defecto y permite crear una mejor interfaz agradable al usuario.

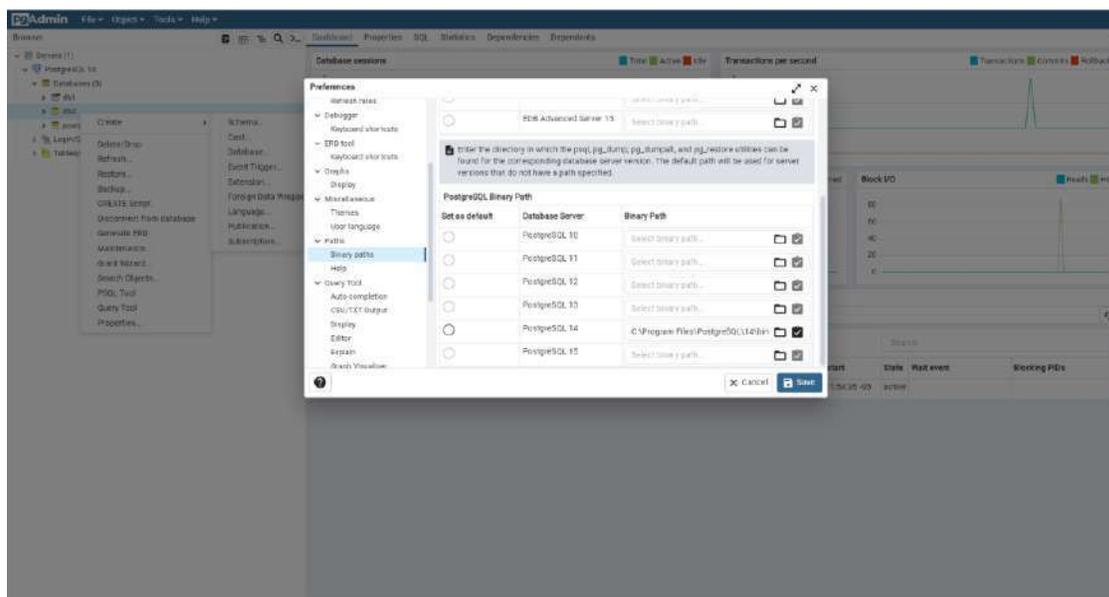
Figura 69. Geoportal sin Menú lateral derecho.



Para poder visualizar el geoportal online se compró el dominio de Geo Catatumbo donde se alojó el software, al cargar los archivos se presenta el error de acceso a los usuarios y al panel de administración por lo que fue necesario realizar el backup de la base de datos y subirlo al servidor.

Al generar el backup se presentó un error ya que postgres no contaba con una ruta por lo que fue necesario agregarla para poder generar el backup y poder subirlo al dominio Geo Catatumbo.

Figura 70. Creación de ruta para crear backup en postgres.



Realizar las pruebas de un software es de gran importancia porque este permite ir evaluando el comportamiento de las funciones, métodos, y rutas, lo que indica que el software esta funcionando y si se ingresa un error en el código se procede a corregirlo, lo que garantiza que el software no va a presentar fallas más adelante. Al inicio puede ser un poco tedioso realizar las pruebas, es por esto que algunas personas deciden ir realizando las pruebas al software y otras lo hacen cuando ya el software está terminado.

4. Diagnóstico final.

Para la creación del geoportal se empezó trabajando con el framework de Angular, pero debido a que se presentaron varios inconvenientes de la conexión con la base de datos se decidió cambiar al framework de Laravel el cual cuenta con una estructura similar a la de angular.

Este software permite que la población de Norte de Santander pueda realizar búsquedas de los diferentes sitios estratégicos, los cuales se pueden localizar en los mapas de cada municipio, haciendo uso de dispositivos electrónicos como celulares, tablets y computadores, ya que este cuenta con diseño responsive el cual se adapta a estos, permitiendo que las diferentes personas que no cuentan con un computador puedan acceder y obtener información en el geoportal.

5. Conclusiones.

Para dar cumplimiento al primer objetivo se realizó un estudio de las diferentes tecnologías con las cuales trabajar, lo que llevo a que se utilizara el framework de Laravel ya que ofrece una mejor facilidad para crear la conexión con la base de datos postgres, el cual hace uso de lenguajes como php, html, jvs y bootstrap, permitiendo crear las diferentes funcionalidades del geoportal.

En el desarrollo del geoportal se hizo uso de diferentes lenguajes de programación los cuales vienen incluidos con el framework Laravel, estos permitieron crear diferentes tipos de funcionalidades y patrones de diseño brindando al usuario una interfaz amigable e intuitiva.

Por último, se realizaron las pruebas unitarias y funcionales del software para garantizar que este funciona correctamente.

6. Recomendaciones

Se recomienda al Centro de Innovación y Tecnológica (CEDIT) de la universidad Francisco de Paula Santander Ocaña continuar realizando este tipo de proyectos que benefician a los estudiantes, brindando la oportunidad de realizar las pasantías, formándose como profesionales, adquiriendo conocimientos y fomentando el desarrollo de proyectos para el uso de la comunidad.

El geoportal es un proyecto amplio y escalable por lo que se recomienda continuar con su creación y ampliar la información suministrada.

Referencias

Angular.<https://angular.io/docs>

Administrar extensiones en Visual Studio Code.pdf. (s. f.). Obtenido de documentación visual studio code

<https://code.visualstudio.com/docs/editor/extension-marketplace>

Centro de Desarrollo e Innovación Tecnológica (CEDIT) de la Universidad Francisco de Paula Santander Ocaña.

<https://cedit.ufpso.edu.co/>

Cardona, V. M., Cleves, G. J., & Barrero, J. G. (s. f.). ¿QUÉ ES UN GEOPORTAL Y CÓMO SE CREA?.10.

Composer. (s.f.). Composer. Obtenido de A Dependency Manager for PHP: <https://getcomposer.org/>

Capas—Ayuda de ArcGIS Online Documentación.pdf. (s. f.).

Crud.pdf. (s. f.). Obtenido de Revista Cubana de Ciencias Informáticas.

<https://www.redalyc.org/pdf/3783/378343671007.pdf>

Cabrera, L. (2019, 2 de diciembre). Angular con MySQL y PHP - Tutorial [Video]. YouTube. <https://youtu.be/08t6bwCaG5E>

D, A. (2022). Tareas esenciales y mejores prácticas para configurar ArcGIS Online. 13.

Extensiones de VS Code para PHP y Laravel – Styde.net.pdf. (s. f.).

Obtenido de Styde

<https://styde.net/extensions-de-vs-code-para-php-y-laravel/>

Gauchat, J. D. (s. f.). El gran libro de HTML5, CSS3 y Javascript. 377.

Kong, W. (2016). Extending Faceted Search to the Open-Domain Web.

ACM SIGIR Forum, 50(1), 90-91.

<https://doi.org/10.1145/2964797.2964814>

Geoportal Medellín. <https://www.medellin.gov.co/geomedellin/>

Gaceta Oficial Digital, miércoles 10 de octubre de 2012. (2012). 85.

Laravel-Clase2.pdf. (s. f.).

Laravel-Clase1.pdf. (s. f.).

Lez, D. N. G. (s. f.). Responsive web design: Diseño multidispositivo para mejorar la experiencia de usuario. 10.

Ley 1915 del 12 de julio de 2018.pdf. (s. f.). Obtenido del artículo El congreso de Colombia

<http://derechodeautor.gov.co:8080/documents/10181/182597/LEY+1915+DEL+12+DE+JULIO+DE+2018.pdf/e29d68a7-1250-4204-a0dd-6c511130d912>

Ley_1273_2009.pdf. (s. f.).

Ley565_2000.pdf. (s. f.).

Laravel. <https://laravel.com/docs/9.x>

Mendoza Vázquez, I. (2011). Definición de un framework para aplicaciones web con navegación sensible a concerns [Magíster en Ingeniería de Software, Universidad Nacional de La Plata].

<https://doi.org/10.35537/10915/4192>

Modal.pdf. (s. f.). Modal. Obtenido de Codea:

<https://codea.app/blog/ventana-modal>

Prada, J. Á., Higuera, T. F. A., & Gallardo, R. J. (s. f.). Determinación Y Análisis De Las Causas De Accidentalidad En La Ciudad De Ocaña, Norte De Santander.

Reichard, K. (1998). Algunos de los dolores de cabeza que lo acompañan. 6.

Rivera, A. L. (s. f.). República de Colombia. Gobierno Nacional. 2.

Ruiz, T. (2021, 26 de enero). Curso MEAN - Angular, Node, Express y MongoDB en 2 horas [Video]. YouTube. <https://youtu.be/aLbDBaFYoLY>

Ruiz, T. (2021, 8 de mayo). Angular Material Tutorial [Video]. YouTube. <https://youtu.be/rWOWTVSMfPw>

Tait, M. G. (2005). Implementación de geoportales: Aplicaciones de SIG distribuidos. 15.

UFPSO <https://ufpso.edu.co/>

Wohner, C., Peterseil, J., Poursanidis, D., Kliment, T., Wilson, M., Mirtl, M., & Chrysoulakis, N. (2019). DEIMS-SDR – A web portal to document research sites and their associated data. *Ecological Informatics*, 51, 15-24. <https://doi.org/10.1016/j.ecoinf.2019.01.005>

Apéndice

Apéndice A Reunión inicio proyecto

Figura 71. Reunión inicio proyecto.



Apéndice B Reunión para realizar los mock ups

Figura 72. Reunión para realizar los mockups.



Apéndice

Apéndice C Constancia continuidad del proyecto Geocatambo.

Figura 73. Constancia de continuidad del proyecto geo Catatumbo.


Universidad Francisco de Paula Santander
 Ocaña - Colombia
 NIT 900 103 130 01

EL DIRECTOR DEL CENTRO DE DESARROLLO INNOVACIÓN TECNOLÓGICA "CEDIT"

HACE CONSTAR QUE:

Los estudiantes **MAYTE ANGÉLICA HERNÁNDEZ PALLARES** identificada con Cédula de Ciudadanía 1.091.678.758 con código 191380, del programa de Ingeniería de Sistemas y **ELVER QUINTERO AYALA** identificado con Cédula de Ciudadanía 1.091.665.590 con código 191509 del programa de Ingeniería de Sistemas realizaron sus pasantías durante el II semestre del año 2022 en el **CENTRO DE DESARROLLO E INNOVACIÓN TECNOLÓGICA "CEDIT"** dependencia de la **UNIVERSIDAD FRANCISCO DE PAULA SANTANDER SECCIONAL OCAÑA**, desempeñando cargos como Analista de Requerimientos y Desarrollo Web respectivamente; dentro del **Proyecto GEOCATATUMBO**, el cual nace de la necesidad de contar con un Sistema que permita el acceso, visualización, y descarga de información relacionada a los municipios que conforman la Región del Catatumbo.

Así mismo, cabe mencionar que desde el **CENTRO DE DESARROLLO E INNOVACIÓN TECNOLÓGICA "CEDIT"** se da continuidad en dicho proyecto, para inicio del año 2023 se van a seguir gestando alianzas para el manejo de datos abiertos con diferentes actores del orden municipal y departamental (Asomunicipios, Alcaldías, Gobernación, Corporación Autónoma Regional, entre otros) que harán parte de este megaproyecto para la Región.

Para constancia se firma a solicitud del interesado a los 13 días del mes de diciembre de 2022.



WILMAR ALIRIO GONZALEZ PEINADO
 Director, Centro de Desarrollo e Innovación Tecnológica "CEDIT"
 UFPSO

Transcriptor: Gricela Peñaranda Rojas



Via Acolsure, Sede el Algodonal, Ocaña, Colombia - Código postal: 546552
 Línea gratuita nacional: 01 8000 121 022 - PBX: (+57) (7) 569 00 88 - Fax Ext. 104
 info@ufpso.edu.co - www.ufpso.edu.co

