



## **Acta de Correcciones al Proyecto de Grado Ingeniería de Sistemas y Computación**

**Fecha: 15 Febrero 2022**

**Autores: Mauricio Cortés Díaz (cod: 8935284),  
Juan Camilo Ariza Cardona (cod: 8931625)**

**Nombre del Proyecto de Grado:**

**Aplicativo web para permitir la participación de la ciudadanía en el cuidado y conservación de los humedales del sur de Cali**

**Director: Gerardo M. Sarria M.**

Como indica el artículo 2.27 de las Directrices de Trabajo de Grado, he verificado que los estudiantes indicados arriba han implementado todas las correcciones que los Jurados del Proyecto de Grado definieron que se efectuaran, como consta en el Acta de Calificación correspondiente.

Firma de Director(a) del Proyecto de Grado

Nota de Aceptación

Aprobado por el Comité de Trabajo de Grado  
en cumplimiento de los requisitos exigidos por la  
Pontificia Universidad Javeriana para optar el  
título de Ingeniero de Sistemas y computación.



**Dr. Camilo Rocha**

Decano de la Facultad de Ingeniería



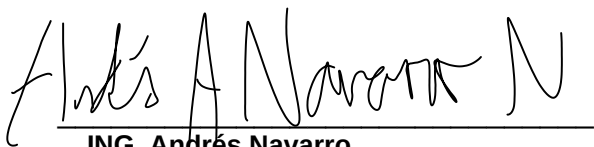
**ING. Gerardo Mauricio Sarria**

Director Carrera Ingeniería Sistemas y Computación.



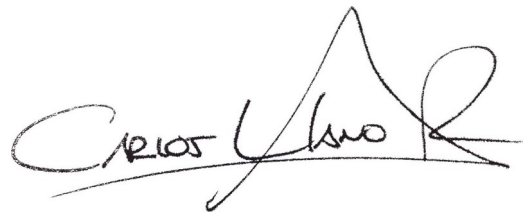
**ING. Gerardo Mauricio Sarria**

Director(a) Trabajo



**ING. Andrés Navarro**

Jurado 1



**ING. Carlos Llano**

Jurado 2

Pontificia Universidad Javeriana Cali  
Facultad de Ingeniería.  
Ingeniería de Sistemas y Computación.  
Proyecto de grado.

Aplicativo web para permitir la participación de la ciudadanía en  
el cuidado y conservación de los humedales del sur de Cali

Mauricio Cortés Díaz  
Juan Camilo Ariza Cardona

Director: Dr. Gerardo Mauricio Sarria

19 Noviembre 2021





Santiago de Cali, 19 Noviembre 2021.

Señores

**Pontificia Universidad Javeriana Cali.**

Dr. Gerardo Mauricio Sarria

Director Carrera de Ingeniería de Sistemas y Computación.

Cali.

Cordial Saludo.

Por medio de la presente me permito informarle que los estudiantes de Ingeniería de Sistemas y Computación Mauricio Cortés Diaz (cod: 8935284) y Juan Camilo Ariza Cardona(cod: 8931625) trabajan bajo mi dirección en el proyecto de grado titulado “Aplicativo web para permitir la participación de la ciudadanía en el cuidado y conservación de los humedales del sur de Cali”.

Atentamente,



---

Dr. Gerardo Mauricio Sarria

Santiago de Cali, 19 Noviembre 2021.

Señores

**Pontificia Universidad Javeriana Cali.**

Dr. Gerardo Mauricio Sarria

Director Carrera de Ingeniería de Sistemas y Computación.

Cali.

Cordial Saludo.

Nos permitimos presentar a su consideración el anteproyecto de grado titulado “Aplicativo web para permitir la participación de la ciudadanía en el cuidado y conservación de los humedales del sur de Cali” con el fin de cumplir con los requisitos exigidos por la Universidad para llevar a cabo el proyecto de grado y posteriormente optar al título de Ingeniero de Sistemas y Computación.

Al firmar aquí, damos fe que entendemos y conocemos las directrices para la presentación de trabajos de grado de la Facultad de Ingeniería aprobadas el 26 de Noviembre de 2009, donde se establecen los plazos y normas para el desarrollo del anteproyecto y del trabajo de grado.

Atentamente,



Mauricio Cortés Díaz

Código: 8935284



Juan Camilo Ariza Cardona

Código: 8931625

# Agradecimientos

A nuestros padres, al director de proyecto el profesor Gerardo Sarria, Ana María Valencia líder del proceso humedales del DAGMA y por último y por eso no menos importante al grandioso foro Stack Overflow.



# Abstract

Known as the *kidneys of the planet*, the vegetation and soils of wetlands are very useful at filtering the water that rests in them. Although this is not their only work, they are also used as wild species nests and it is one of the ecosystems with the greatest biodiversity in the world. Currently the technological resources of the city of Santiago de Cali (Colombia) are limited and there is no constant review on their care, that's why it is decided to carry out a project that ensures the preservation of some wetlands in the city, some of them are the "Humedal La Babilla", "Humedal El Retiro" and "Ecoparque Lago de las Garzas".

This project seeks to promote stronger communication between citizens and the DAGMA in which people report irregularities that occur in these areas so that after the review of the competent entities, timely actions are taken to preserve such an important ecosystem. It will be implemented in a web-application and can be accessed from any device with an internet connection both for the DAGMA and for people who wish to make a contribution. This does not seek to carry out a simple forum in which situations are commented, but will make a special communication for each event that is added to the platform.

**Key Words:** Ecosystems, Community, Web Application, Wetlands, Biodiversity.



# Resumen

Conocidos como los *riñones del planeta*, la vegetación y los suelos de los humedales son muy útiles a la hora de filtrar el agua que en estos reposa. Aunque este no es su único trabajo, en ellos también habitan especies silvestres y es uno de los ecosistemas con mayor biodiversidad del mundo. Actualmente los recursos tecnológicos de la ciudad de Santiago de Cali (Colombia) son limitados y no se cuenta con una constante revisión del cuidado de estos, es por ello que se decide realizar un proyecto que vele por la preservación de algunos humedales de la ciudad, entre ellos están el Humedal La Babilla, El Retiro y El Lago de las Garzas.

Este proyecto busca favorecer una comunicación más fuerte entre la ciudadanía y el DAGMA en el que las personas reporten irregularidades que sucedan en estas zonas para que luego de la revisión de las entidades competentes, se realicen acciones oportunas para preservar tan importante ecosistema. Será implementado en un aplicativo-web y podrá ser accedido desde cualquier dispositivo con conexión a internet tanto para el DAGMA, como para las personas que deseen realizar un aporte. Este no busca realizar un simple foro en el que se comenten situaciones, sino que realizará una comunicación especial por cada evento que sea agregado a la plataforma.

**Palabras Clave:** Ecosistemas, Comunidad, Aplicativo Web, Humedales, Biodiversidad.



# Índice general

<b>1. Descripción del Problema</b>	<b>17</b>
1.1. Planteamiento del Problema . . . . .	17
1.1.1. Formulación . . . . .	18
1.1.2. Sistematización . . . . .	18
1.2. Objetivos . . . . .	18
1.2.1. Objetivo General . . . . .	18
1.2.2. Objetivos Específicos . . . . .	18
1.3. Justificación . . . . .	19
<b>2. Marco de Referencia</b>	<b>21</b>
2.0.1. Áreas Temáticas . . . . .	21
2.0.2. Marco Teórico . . . . .	21
2.0.3. Definición de términos . . . . .	23
2.0.4. Metodología de trabajo . . . . .	24
2.0.5. Metodología Scrum . . . . .	27
2.0.6. Estructura de Desglose de Trabajo: . . . . .	30
2.0.7. Gestión de actividades . . . . .	31
2.0.8. Una SLR sobre aplicativos en pro de la conservación de ecosistemas, en especial humedales . . . . .	32
2.0.9. Preguntas de investigación . . . . .	32
2.0.10. Caracterización de dimensiones y recopilación de datos . . . . .	33
2.0.11. Conjunto final de trabajo relacionado . . . . .	33
<b>3. Desarrollo del Proyecto</b>	<b>37</b>
3.0.1. Búsqueda y comparación de herramientas . . . . .	37
3.0.2. Diagramas de casos de uso . . . . .	45
3.0.3. Prototipado y Desarrollo . . . . .	47
3.0.4. Modelo de datos: . . . . .	49
3.0.5. Arquitectura . . . . .	50
3.0.6. Listado de requisitos . . . . .	55
3.0.7. Apis externas: . . . . .	58
<b>4. Reconocimiento de imágenes</b>	<b>61</b>
4.1. Qué es el reconocimiento de imágenes . . . . .	61
4.1.1. Tareas del reconocimiento de imágenes . . . . .	61
4.1.2. Cómo funciona el reconocimiento de imágenes . . . . .	62
4.2. Convolutional neural networks . . . . .	62

4.2.1.	Capa convolucional . . . . .	63
4.2.2.	Relu Capa . . . . .	64
4.2.3.	Capa de agrupación . . . . .	64
4.2.4.	Capa completamente conectada . . . . .	65
4.3.	Por qué el reconocimiento de imágenes . . . . .	66
4.4.	Cómo construir una CNN . . . . .	67
4.5.	APIs reconocimiento de imágenes . . . . .	67
4.5.1.	Google Cloud Vision . . . . .	67
4.5.2.	IBM Watson Visual Recognition . . . . .	67
4.5.3.	Computer Vision Microsoft Azure . . . . .	68
4.5.4.	Elección API . . . . .	68
4.6.	Google Cloud Vision . . . . .	69
<b>5.</b>	<b>Pruebas</b>	<b>73</b>
5.1.	Funcionalidades a probar . . . . .	73
5.1.1.	Bloque 1 . . . . .	74
5.1.2.	Bloque 2 . . . . .	74
5.1.3.	Bloque 3 . . . . .	74
5.2.	Estrategia de pruebas . . . . .	75
5.3.	Análisis estático . . . . .	75
5.4.	Pruebas de caja blanca . . . . .	75
5.5.	Pruebas de caja negra . . . . .	75
5.6.	Pruebas de carga . . . . .	76
5.7.	Pruebas de estrés . . . . .	76
5.8.	Herramientas a usar en las pruebas no funcionales . . . . .	76
5.9.	Configuración ambiente de pruebas . . . . .	76
5.10.	Anexos pruebas . . . . .	76
<b>6.</b>	<b>Despliegue</b>	<b>79</b>
6.1.	App engine . . . . .	79
6.2.	Cloud SQL . . . . .	79
6.3.	Cloud storage . . . . .	79
<b>7.</b>	<b>Resultados</b>	<b>81</b>
7.1.	Aplicativo web . . . . .	81
7.1.1.	Aporte . . . . .	81
7.1.2.	Blog . . . . .	81
7.1.3.	Aporte detalle . . . . .	82
7.1.4.	Humedales . . . . .	82
7.1.5.	Acceso al aplicativo web . . . . .	83
7.2.	Resultados con ciudadanos . . . . .	83
7.2.1.	Análisis de resultados . . . . .	83

<b>Índice general</b>	<b>13</b>
<b>8. Conclusiones</b>	<b>91</b>
<b>Bibliografía</b>	<b>93</b>



# Introducción

Cali figura como una de las ciudades con mayor biodiversidad del planeta. Este lugar privilegiado lo debe en gran parte a su ubicación geográfica. Contar con un gradiente altitudinal que va desde los 1000 metros sobre el nivel del mar en la zona urbana y hasta los 4100 metros s.n.m. en la zona rural hace que la capital del Valle abarque una gran variedad de ecosistemas, que permiten que se convierta en el hábitat apropiado para muchas especies de fauna y flora.[dEP20]

Precisamente, entre sus mayores riquezas naturales, se encuentran los ecosistemas de humedales. Los humedales son vitales para la supervivencia humana, son uno de los entornos más productivos del mundo, y son cunas de diversidad biológica y fuentes de agua y productividad primaria de las que innumerables especies vegetales y animales dependen para subsistir.

Sin embargo, estudios demuestran que la superficie y la calidad de los humedales siguen disminuyendo en la mayoría de regiones del mundo[Ram18]. En consecuencia, los servicios de los ecosistemas que los humedales proporcionan a las plantas, animales y personas se encuentran en peligro. Cali y el Valle del Cauca no son la excepción, en las últimas décadas se ha perdido mas del 88% de estos ecosistemas. Los principales factores que amenazan la vida de los humedales se relacionan con la contaminación, la disposición de residuos sólidos y escombros, la colmatación, la desecación y drenaje, e incluso las obras de construcción en sus áreas de influencia.

El Departamento Administrativo de Gestión del Medio Ambiente (DAGMA), ente encargado de proteger estos ecosistemas, enfoca grandes esfuerzos en el mantenimiento, protección e incluso identificación de los humedales en la ciudad. A pesar de ello, los fondos no son suficiente para mantener un monitoreo periódico de los humedales de la ciudad.

Por tal razón, se propone desarrollar un aplicativo web que permita participar a la ciudadanía en el cuidado y conservación de los humedales del sur de Cali (Humedal La Babilla, El Retiro y El Lago de las Garzas) pudiendo compartir información relevante que luego sera analizada por el DAGMA para tomar las medidas ambientales correspondientes.



# Descripción del Problema

---

## 1.1. Planteamiento del Problema

Los humedales son áreas donde el agua y la tierra se juntan para formar el hogar de un gran número de especies de plantas y animales, sirven como refugio a una multitud de aves migratorias y junto a la riqueza hídrica que provee forma parte de muchos ecosistemas con una gran biodiversidad que necesitan ser protegidos y conservados.

Solo en el municipio de Santiago de Cali el DAGMA ha contabilizado 61 humedales entre públicos y privados. Siendo la mayoría de estos artificiales y concentrados en la comuna 22, sur de la ciudad [Cal17]. Estos humedales han acompañado el paisaje urbano de la ciudad durante décadas, cumpliendo funciones reguladoras como fuente hídrica y siendo símbolos de la riqueza de flora y fauna de la ciudad.

Estos ecosistemas como muchos otros no son ajenos a la acción antrópica. El crecimiento poblacional que se ha acelerado notablemente en los últimos 50 años, las formas de ocupación del territorio, unido a las prácticas socioculturales, han convertido la ciudad en un gran complejo urbano, con saldos ambientales críticos donde desechos plásticos-orgánicos ahora forman parte de los ecosistemas. Lo que pone en evidencia, no solo una tendencia negativa en cuanto al cuidado y conservación de estos ecosistemas, si no también que los humedales se siguen perdiendo, transformando o degradándose, teniendo como resultado gravísimos impactos negativos en los servicios ecosistémicos que brindan y en el deterioro o pérdida de la biodiversidad presente.

Ante esta situación distintos entes ambientales como la CVC y el DAGMA han tomado cartas en el asunto en los últimos años, realizando múltiples esfuerzos y trabajos para la recuperación y protección de estos valiosos ecosistemas reconociendo su importancia por la biodiversidad que albergan y la gran capacidad que tienen de ofrecer bienes y servicios ambientales. Sin embargo, los recursos manejados por estas entidades publicas no son lo suficientemente grandes para mantener un control y monitoreo periódico amplio para cada uno de los humedales. Con los distintos estudios se han logrado crear varios planes de manejo ambiental para algunos humedales de la comuna 22 [dGdMA17]. A pesar de estos esfuerzos, estas entidades no logran tener un alcance mayor pese a que no solamente deben tratar el tema ambiental de los humedales si no el de una gran variedad de ecosistemas como ríos, lagunas, bosques, etc. Por ello, es de vital importancia que el ciudadano también aporte información sobre posibles factores de riesgo que observa en los humedales, para

que juntos (entidad ambiental y ciudadano) se logre un apoyo en el cual los dos se vean beneficiados.

En respuesta a esto, se busca una manera en la cual el ciudadano pueda compartir información relevante y útil que tenga valor para estas entidades ambientales como el DAGMA, además de poder acceder a información relevante de distintos humedales a fin de tomar conciencia de la importancia de estos ecosistemas para la ciudad. Se plantea la solución desde un tipo de vista tecnológico a través de la cual el ciudadano pueda fácilmente proveer a estas entidades de rica información para su posterior análisis y sobre esta tomar las acciones correspondientes por la entidad.

### 1.1.1. Formulación

¿Cómo permitir la participación de la ciudadanía en el cuidado y conservación de los humedales del sur de Cali a través de un aplicativo web que envíe información a entidades ambientales (DAGMA)?

### 1.1.2. Sistematización

- ¿Qué requisitos técnicos y paquetes web serán usados en el desarrollo del aplicativo?
- ¿Cómo relacionar al usuario en el uso correcto del aplicativo web para la recolección de datos, según las necesidades del DAGMA?
- ¿Cómo implementar el aplicativo haciendo uso de los paquetes web identificados de manera que se cumplan los requisitos técnicos?
- ¿Cómo evaluar la experiencia del usuario en el aplicativo?

## 1.2. Objetivos

### 1.2.1. Objetivo General

Desarrollar un aplicativo web que permita la participación de la ciudadanía en el cuidado y conservación de los humedales del sur de Cali y el envío de información a entidades ambientales (DAGMA).

### 1.2.2. Objetivos Específicos

- Definir y adaptar los requisitos técnicos y paquetes web que serán usados en el desarrollo del aplicativo.
- Diseñar y aplicar una estrategia para relacionar al usuario en el uso correcto del aplicativo web para la recolección de datos, según las necesidades del DAGMA.
- Implementar el aplicativo web acorde a los requisitos técnicos usando los paquetes web.
- Evaluar la experiencia del usuario en uso del aplicativo.

## 1.3. Justificación

La contaminación ambiental que estamos viviendo en la actualidad provocada por la acción del hombre, está causando que muchos ecosistemas claves para el desarrollo de la fauna y flora sean transformados, por cuenta de la deforestación, la minería, construcción de obras civiles, etc. Entre estos ecosistemas afectados están los humedales, lugares donde el agua y tierra se juntan para brindar hogar a un gran número de especies de plantas y animales, sirviendo como refugio a una multitud de aves migratorias y junto a la riqueza hídrica que provee forma parte de los muchos ecosistemas que con el paso del tiempo han venido siendo afectados.

Entre las principales causas del deterioro de los humedales están la construcción de avenidas y la proliferación de urbanizaciones cerca de ellos, causando efectos como la disposición de residuos sólidos, plásticos y escombros, la colmatación, la desecación y drenaje, todo esto afectando la gran biodiversidad y riquezas de estos ecosistemas.

A pesar de la importancia de estos ecosistemas, se cree que la superficie de humedales que se perdió en todo el mundo, en los últimos 100 años, ha sido entre un 69 y un 75 %. Desafortunadamente, Colombia y especialmente el Valle del Cauca no ha estado ajena a estas tendencias tanto así que en la década de los 50 existían alrededor de 15 mil hectáreas de humedales lénticos y a finales de los años ochenta quedaban 1500 hectáreas, el 88 % de ellos habían desaparecido; este dato está registrado en una publicación hecha por The Nature Conservancy (TNC), Fundación Natura y CVC. En Cali la situación no es diferente, por ello, el DAGMA en los últimos años ha invertido más de 1.000 millones recursos propios y sobre tasa ambiental [Cal20]; enfocando grandes esfuerzos en el mantenimiento, protección e incluso identificación de los humedales en la ciudad. Este trabajo se plantea desde la necesidad que tiene el DAGMA en el cuidado y control de los humedales, dado que los recursos no alcanzan para mantener un monitoreo regular a cada uno de ellos. Es así como se busca que la propia ciudadanía se encargue de informar a esta entidad ambiental sobre posibles afectaciones que pongan en riesgo a los humedales a través un medio tecnológico.

Mediante la implementación y puesta en marcha del aplicativo web se busca que los ciudadanos compartan valiosa información que sirva para el cuidado y conservación de los humedales del Sur de Cali, esta información será recibida por el DAGMA quien se encarga de tomar las medidas ambientales correspondientes para frenar un posible deterioro de estos humedales que afecte su biodiversidad.



# Marco de Referencia

---

## 2.0.1. Áreas Temáticas

En esta sección se enuncian áreas afines al problema tratado que se encuentran en la ACM Computing Classification System (ACM CCS)[Lib15], haciendo énfasis en la ingeniería y la interacción con humanos.

- **Human-centered computing** - Human computer interaction (HCI) - HCI design and evaluation methods
- **Human-centered computing** - Ubiquitous and mobile computing - Ubiquitous and mobile devices - Mobile devices
- **Information systems** - World Wide Web - Web applications
- **Software and its engineering** - Software creation and management

## 2.0.2. Marco Teórico

En las siguientes subsecciones se especifican los conceptos relacionados con el proyecto:

### 2.0.2.1. Interacción Humano-Computador

Con el desarrollo de interfaces gráficas para manipulación directa, los sistemas informáticos se hicieron populares rápidamente. Los computadores ya no solo son utilizados por expertos y tecnólogos, sino que gracias a estos sistemas de interacción más intuitivos, un gran público tiene la oportunidad de utilizar sus dispositivos y aplicaciones de software.

Según Myers, Hollan y Cruz [MBA96], HCI se puede definir como: *El estudio sobre cómo se diseñan, implementan y usan los sistemas informáticos interactivos y la manera que influyen los ordenadores en los individuos, las organizaciones y la sociedad*

La definición de tres palabras han determinado el diseño de todo sistema basado en la interacción humano-computador:

- **Usabilidad:** Se trata de hacer que los sistemas sean fáciles de aprender y usar. Mejora la competitividad, la fidelización de los clientes y reduce los costes.
- **Seguridad:** Es la capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos.

- **Funcionalidad:** Representa la capacidad del producto software para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas.

### 2.0.2.2. Computación móvil

La Computación Móvil es una tecnología que permite la transmisión de datos, voz y video a través de una computadora o cualquier otro dispositivo inalámbrico sin tener que estar conectado a un enlace físico fijo. La principal característica de la computación móvil se basa en la portabilidad de los dispositivos y el centrarse en tecnología de uso personal.

La computación móvil implica comunicaciones móviles, hardware móvil y software móvil. Los problemas de comunicación incluyen redes ad hoc y redes de infraestructura, así como propiedades de comunicación, protocolos, formatos de datos y tecnologías concretas. El hardware incluye dispositivos móviles o componentes de dispositivos. El software móvil se ocupa de las características y requisitos de las aplicaciones móviles. Particularmente, la computación móvil es presentada a partir de sistemas de geolocalización, libros electrónicos, celulares inteligentes, computadores portátiles y consolas de juego

### 2.0.2.3. Aplicación web

Una aplicación web es una herramienta informática accesible desde cualquier navegador, bien sea a través de internet o bien a través de una red local. Gracias a esto se adapta a cualquier dispositivo, como teléfonos inteligentes o tabletas. Dado que la creación de aplicaciones nativas permiten desarrollos más complejos, también son más costosas y no siempre necesarias.

Existen diferentes tipos de aplicaciones web:

- **Aplicación web estática:**

Este tipo de *Web-App* muestra poca información y no suele cambiar mucho. Por regla general suelen estar desarrolladas en HTML y CSS, aunque también se pueden mostrar en alguna parte de la aplicación web objetos en movimiento como por ejemplo banners, GIF animados o vídeos.

- **Aplicación web dinámicas:**

Utilizan bases de datos para cargar la información y estos contenidos se actualizan cada vez que el usuario accede a la *Web-App*. Existen diferentes lenguajes de programación para el desarrollo de aplicaciones web dinámicas. Los lenguajes PHP y ASP son los más comunes porque permiten una buena estructuración del contenido.

- **Tienda virtual o comercio electrónico:**

A diferencia de una aplicación web dinámica, el desarrollo de una tienda virtual es más

---

complicado porque debe permitir pagos electrónicos a través de tarjeta de crédito, PayPal u otro método de pago. El desarrollador también deberá crear un panel de gestión para el administrador. A partir de él se subirán, actualizarán o eliminarán los productos y se podrán gestionar pedidos y pagos.

- **Aplicación web con gestor de contenidos:**

Si existe el caso de necesitar una aplicación web que exija realizar cambios continuamente, se debe recurrir a un gestor de contenidos (CMS), este puede ser modificado por el administrador y realizar las modificaciones requeridas.

#### 2.0.2.4. Gestión de proyectos de software

La gestión de proyectos de software se refiere a la rama de la gestión de proyectos dedicada a la planificación, programación, asignación de recursos, ejecución, seguimiento y entrega de proyectos de software y web.

Por ello, se tiene la gestión como un aspecto clave, dado que a lo largo del proyecto se deben establecer distintas etapas en las cuales se deben desarrollar un conjunto de actividades para dar con el avance del proyecto, estas actividades serán monitoreadas por el director del proyecto para verificar la correcta realización de estas, junto con sus sugerencias y correcciones.

- **Planificación:** esto significa armar el anteproyecto para todo el proyecto desde la ideación hasta la realización. Definirá el alcance, asignará los recursos necesarios, propondrá el cronograma, delinearé el plan de ejecución, trazará una estrategia de comunicación e indicará los pasos necesarios para las pruebas y el mantenimiento.
- **Líder:** un gerente de proyecto de software deberá reunir y liderar el equipo del proyecto, que probablemente estará formado por desarrolladores, analistas, probadores, diseñadores gráficos y redactores técnicos. Esto requiere excelentes habilidades de comunicación, personas y liderazgo.
- **Ejecución:** el gerente del proyecto participará (director Gerardo Sarria) y supervisará la ejecución exitosa de cada etapa del proyecto. Esto incluye el seguimiento del progreso, los controles frecuentes del equipo y la creación de informes de estado.
- **Mantenimiento:** la gestión de proyectos de software normalmente fomenta las pruebas constantes del producto para descubrir y corregir errores a tiempo, ajustar el producto final a las necesidades del cliente y mantener el proyecto en el objetivo. El gerente del proyecto de software es responsable de garantizar que se realicen pruebas, evaluaciones y correcciones adecuadas y consistentes.

#### 2.0.3. Definición de términos

En esta sección se realizará una especificación de algunos términos que son considerados importantes a la hora de realizar proyectos de aplicaciones web.

- **API:** La *interfaz de programación de aplicaciones* también conocida por la sigla API, permite que dos sistemas diferentes se comuniquen entre ellos, gracias a que los desarrolladores liberan funciones que pueden ser usadas por sistemas externos. Uno de sus grandes beneficios es que esta promueve la reutilización de código de terceros, lo que agiliza el proceso de desarrollo.
- **Backend:** Esta parte del desarrollo comprende todo lo que sucede detrás de lo que pueden observar los usuarios. Esto específicamente es lo que controla y da forma al *frontend*, en otras palabras lo que puede observar e interactuar. Ambos se comunican enviando y recibiendo información para ser desplegada en el aplicativo y permite el desarrollo de un sitio web dinámico.
- **Blog:** Un blog es una sección de un sitio web que consta de entradas que revisan diferentes temas en los que se resalta el punto de vista del autor. Estas se ordenan de acuerdo a su fecha de publicación o relevancia de acuerdo a algún definido. Estas publicaciones más que priorizar un dialecto formal, fomentan el toque personal y un medio de difusión reconocido.
- **Bug:** Un error de software o en inglés *Bug* es un resultado o comportamiento indeseado en un aplicativo ya sea de consola, móvil o web. Estos ocurren tanto a la hora la ejecución, como también en la compilación. Debido a estos se suelen encontrar vulnerabilidades en el sistema desarrollado y pueden ser un problema hasta para la base de datos.
- **Deployment:** Refiérase a despliegue, el proceso que incluye la secuencia de actividades necesarias para que sea posible el uso de un sistema y que este sea disponible a los usuarios. Algunas actividades de despliegue reconocidas son:
  - **Lanzamiento** (*Release*)
  - **Instalación y activación**
  - **Desinstalación**
  - **Actualización**
- **Frontend:** El frontend de un página web es lo que el usuario puede ver e interactuar, en este se precisa el resultado visual del aplicativo y prioriza la organización de los elementos en el *HTML* y los estilos en *CSS*, existen numerosos frameworks que son de utilidad a la hora de determinar la estética del sitio web, sus colores y el estilo que se desea poseer.
- **Mockups:** Son prototipos preliminares, que buscan enseñar al usuario final cómo lucirá el aplicativo sin tener que realizar una gran implementación. Estos además permiten que se conozcan un avance temprano y posibles modificaciones, antes de llegar a la etapa de desarrollo.

#### 2.0.4. Metodología de trabajo

El desarrollo ágil de software abarca un enfoque hacia la toma de decisiones en los proyectos de software, esta toma de decisiones busca un desarrollo iterativo e incremental, de manera que si existe un determinado problema, este se pueda precisar con rapidez y flexibilidad. Suponiendo una forma distinta de trabajar y organizarse, donde requisitos y soluciones evolucionen en el tiempo. Así, cada

proyecto se 'parte' en pequeñas partes que deben completarse y entregarse en pocas semanas. El motivo es desarrollar productos y servicios de alta calidad que satisfagan las necesidades cambiantes de cada uno de los clientes a una velocidad cada vez mayor.

La elección de esta metodología nació del éxito que ha tenido desde su surgimiento en los años 90, donde las estadísticas hablan por sí solas, de acuerdo al portal Open Source [Bun20] los proyectos de software usando metodologías ágiles tienen una probabilidad del 42% de éxito, mientras que usando una metodología tradicional como Cascada tiene una probabilidad del 26% de éxito, en donde el 53% de estos proyectos cambian con el tiempo y el 21% no es exitoso.

En el 2001 gracias al trabajo en conjunto de CEOs de varias compañías estadounidenses se crea el *Manifiesto Ágil* el cual especifica 4 puntos claves [Wik20]:

- **Individuos e interacciones:** La importancia y el valor de las personas antes que a los procesos y herramientas, si bien, los procesos ayudan dado que guían al trabajo y las herramientas mejoran la eficiencia, son las personas o empleados quienes colocan en marcha estos y si estos no tienen el suficiente conocimiento técnico y la actitud adecuada, no se producen resultados. Son estos procesos quienes se deben adaptar a las organizaciones, a los equipos y a las personas.
  
- **Software funcionando:** Siguiendo una metodología ágil se podrán ver distintas funcionalidades de algún proyecto de forma anticipada ya sea mediante prototipos o partes ya elaboradas, lo que ofrece una retroalimentación muy estimulante y enriquecedora al cliente de manera que este puede observar resultados tempranos y en caso de algún problema formular cambios. Por lo tanto, es difícil tener un documento con todos los requisitos bien detallados desde el comienzo del proyecto. *El software funcionando es la medida principal de progreso* [Cun01].

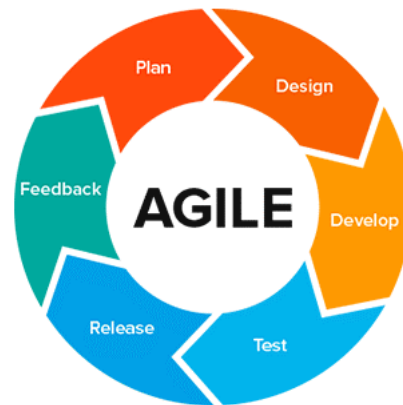


Figura 2.1: Metodología Ágil para el desarrollo de un proyecto de Software, tomado de Global Growth Agents. Disponible en <https://www.wearemarketing.com/blog/>

- **Colaboración con el cliente:** La colaboración con el cliente es de vital importancia para obtener resultados puesto que el valor del resultado no es consecuencia de haber controlado una ejecución siguiendo a procesos, sino de haber sido implementado de forma directa sobre el producto. De manera que el cliente es parte del equipo, integrándose y colaborando con él. Dejando de lado los contratos que solo son una serie de líneas que dividen responsabilidades.
- **Respuesta ante el cambio:** Como la metodología ágil surge en entornos cambiantes e inestables resulta mucho más valiosa la capacidad de respuesta que la de seguimiento y aseguramiento de planes pre-establecidos. Dado que los principales valores ágiles son la anticipación y la adaptación difieren totalmente de los métodos ortodoxos donde la planificación y control son muy fuertes para evitar desviaciones sobre algún proyecto.

#### 2.0.4.1. Metodologías Ágiles

Hoy en día, existen una multitud de metodologías ágiles, de las cuales se van a nombrar las más conocidas, una de ellas se escogerá como método de trabajo a seguir y se explicará a fondo su funcionamiento y el porqué se escogió.

- **Extreme Programming XP:** Esta metodología es bastante útil para startups o empresas que están en proceso de consolidación, dado que su principal objetivo es ayudar en las relaciones entre los clientes y empleados. La clave del éxito del Extreme Programming XP es potenciar las relaciones personales, a través del trabajo en equipo, fomentando la comunicación y eliminando los tiempos muertos [V19a]. Las principales características de esta metodología son las siguientes:
  - Desarrollo iterativo e incremental.

- Pruebas unitarias continuas.
- Programación en parejas, esta metodología recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- **Scrum:** La metodología Scrum es un marco de trabajo o framework que se utiliza dentro de equipos que manejan proyectos complejos. Es decir, se trata de una metodología de trabajo ágil que tiene como finalidad la entrega de valor en períodos cortos de tiempo y para ello se basa en tres pilares: la transparencia, inspección y adaptación. Esto permite al cliente, junto con su equipo comercial, insertar el producto en el mercado pronto, rápido y empezar a obtener ventas [E.20].
- **Kanban:** De acuerdo al portal ieb school "La estrategia Kanban conocida como "Tarjeta Visual" muy útil para los responsables de proyectos. Esta consiste en la elaboración de un cuadro o diagrama en el que se reflejan tres columnas de tareas: pendientes, en proceso o terminadas. Este cuadro debe estar al alcance de todos los miembros del equipo, evitando así la repetición de tareas o la posibilidad de que se olvide alguna de ellas. Por tanto, ayuda a mejorar la productividad y eficiencia del equipo de trabajo"[V19b].

### 2.0.5. Metodología Scrum

Scrum es un marco de trabajo que permite el desarrollo de sistemas de manera ágil, priorizando de esta manera la entrega de productos del máximo valor posible. Scrum no es una técnica para el desarrollo de productos; en cambio es una metodología como protocolo a seguir para desarrollar efectivamente. Este marco de trabajo emplea un enfoque iterativo e incremental, en el que en cada sprint se realiza una revisión de cambios y mejoras de la anterior iteración y además, garantiza de que siempre exista una versión más completa y sea totalmente funcional. Los elementos de los cuales se hace valer Scrum para el correcto desarrollo de sistemas son los roles, eventos y herramientas seguidamente descritos.

- **Roles del equipo Scrum:** El equipo cuenta con un grupo capacitado tanto en herramientas para el desarrollo del sistema, como también conocimiento del marco de trabajo a seguir para siempre priorizar la agilidad. El equipo es autosuficiente y posee los conocimientos necesarios para realizar un desarrollo sin depender de personas externas al grupo de trabajo. Los roles que componen al equipo de Scrum son los siguientes:
  - **Product Owner:** Es el responsable del proyecto. Aunque todos los participantes tomen un rol en el proyecto, la persona que tendrá la responsabilidad del trabajo es este. Él brindará a los demás integrantes del proyecto, la transparencia del product backlog, siendo él, el único que lo puede gestionar y manipular. Cabe resaltar que este es un individuo y no un conjunto de personas que trabajan en el proyecto.
  - **Development team:** Es el conjunto de personas que realizan la implementación del proyecto y además, en la realización del proyecto se entregan incrementos en sprints y se garantiza la funcionalidad del sistema entregado.

- **Scrum Master:** Es el individuo que se especializa en Scrum y asegura que los demás roles conozcan y entiendan el marco de trabajo y el flujo que se debe seguir para realizar alguna acción. Además, este también busca que personas externas al equipo, puedan conocer qué interacciones con el Equipo Scrum pueden ser útiles.
- **Eventos de Scrum:** El marco de trabajo Scrum, cuenta con eventos ya establecidos con el fin de minimizar la cantidad de reuniones no requeridas. Al comenzar un Sprint, su duración permanece constante y no se alarga o acorta.
  - **El Sprint:** El Sprint son las secciones en las que se realizará participaciones en la cronología del proyecto, estas suelen durar un mes o menos.
  - **Sprint Planning:** En este evento se realiza el análisis de los trabajos que se realizarán en el Sprint y para un Sprint con duración de 1 mes, la duración de la planificación es de 8 horas como máximo.
  - **Daily Scrum:** En este evento se realiza una reunión de máximo 15 minutos, que prioriza la aclaración de dudas y sincronización de personal. En este también se revisa el trabajo desarrollado y se realiza una proyección para lo que se obtendrá antes del siguiente Daily Scrum.
  - **Sprint Review:** Este evento ocurre precisamente antes de la finalización del Sprint y tiene como objetivo identificar el incremento del desarrollo y modificar el Product Backlog si es necesario. Su duración máxima es de 4 horas.
  - **Sprint Retrospective:** Este evento proporciona un espacio de revisión individual y ocurre entre el Sprint Planning y el Sprint Review. Su duración máxima es de 3 horas.

#### 2.0.5.1. ¿Por qué Scrum?

Se decidió escoger Scrum dada la familiaridad que se tiene con este marco de trabajo, puesto que a lo largo de la carrera se ha usado Scrum para realizar múltiples actividades y proyectos acogiéndolos al enfoque del desarrollo ágil. Por tanto, conocemos a buen nivel de detalle los roles, los eventos y las herramientas para aplicar Scrum y poder así trabajar de forma colaborativa apuntando siempre a obtener el mejor resultado posible para los proyectos.

#### 2.0.5.2. Herramientas

- **Historias de usuario (Product Backlog):** Para realizar el producto backlog del proyecto decidimos usar la herramienta Jira, la cual nos fue de gran ayuda para organizar las diferentes tareas, asignando prioridades, tiempos de inicio y fin, responsable y una descripción detallada de lo que se debía hacer para cumplir tal tarea. En la figura 2.2 se puede observar como se repartieron las actividades, junto con su nivel de prioridad y su estado.

T	Clave	Resumen	Informador	Responsable	Estado	Pr	Time to resolution	Change type
<input type="checkbox"/>	PDAG-11	Mockup	Juan Camilo Ariza	mauricio cortes	FINALIZADA	↑	16 h	II
<input type="checkbox"/>	PDAG-10	Crear vista "Nuestra Historia"	Juan Camilo Ariza	Juan Camilo Ariza	FINALIZADA	↑	16 h	II
<input type="checkbox"/>	PDAG-9	Crear vista "Programas y eventos"	mauricio cortes	Juan Camilo Ariza	ABIERTA	↑	16 h	II
<input type="checkbox"/>	PDAG-8	Incluir imágenes en galería Fauna Humedal La babilla	mauricio cortes	Juan Camilo Ariza	ABIERTA	↑	16 h	II
<input type="checkbox"/>	PDAG-6	Conectar API VISION AI (GOOGLE) al aplicativo	mauricio cortes	mauricio cortes	ABIERTA	↑	16 h	II
<input type="checkbox"/>	PDAG-5	Crear vista "Reporte a realizar"	mauricio cortes	Juan Camilo Ariza	ABIERTA	↑	16 h	II
<input type="checkbox"/>	PDAG-4	Crear modelo de base de datos	mauricio cortes	mauricio cortes	ABIERTA	↑	16 h	II
<input type="checkbox"/>	PDAG-3	Vista Home	Juan Camilo Ariza	Juan Camilo Ariza	FINALIZADA	↑	16 h	II
<input type="checkbox"/>	PDAG-1	Seleccionar frontend	mauricio cortes	mauricio cortes	FINALIZADA	↑	16 h	II

Figura 2.2: Herramienta Jira para organizar el backlog. Disponible en <https://www.atlassian.com/es/software/jira>

- Gestión de contenido:** Jira, además de permitirnos organizar el backlog también nos ofrece una gran variedad de funcionalidades para poder gestionar todas las actividades o tareas que sean necesarias mediante *widgets, dashboards, etc.* Tal como se ve en la figura 2.3, la cual muestra en la parte superior opciones para filtrar, crear paneles y aplicaciones. Así mismo, muestra un resumen de los proyectos recientes.

**Tu trabajo**

Jira Work Management está disponible para los equipos empresariales como la generación futura de Jira Core. Da la bienvenida a nuevas formas de trabajar con listas, calendarios, cronogramas, etc. [Mostrarme](#) · [Descartar](#)

**Proyectos recientes** [Ver todos los proyectos](#)

**Proyecto de Grado | Hum...**  
Gestión de asistencia

COLAS RECIENTES

All open 9

Unassigned issues 0

3 colas ▾

**Demo del proyecto de asi...**  
Gestión de asistencia

COLAS RECIENTES

All open 6

Assigned to me 1

3 colas ▾

Figura 2.3: Vista general del proyecto en Jira. Disponible en <https://humedalesurbanos.atlassian.net/>

- **Repositorio:** Para el manejo del código utilizamos GitHub (Ver figura 2.4) , donde cada vez que se realizaban cambios importantes al proyecto se hacia el correspondiente commit, asignándole su debida descripción sobre el cambio o funcionalidad agregada, siguiendo el marco de trabajo Scrum, con sprints cada semana.

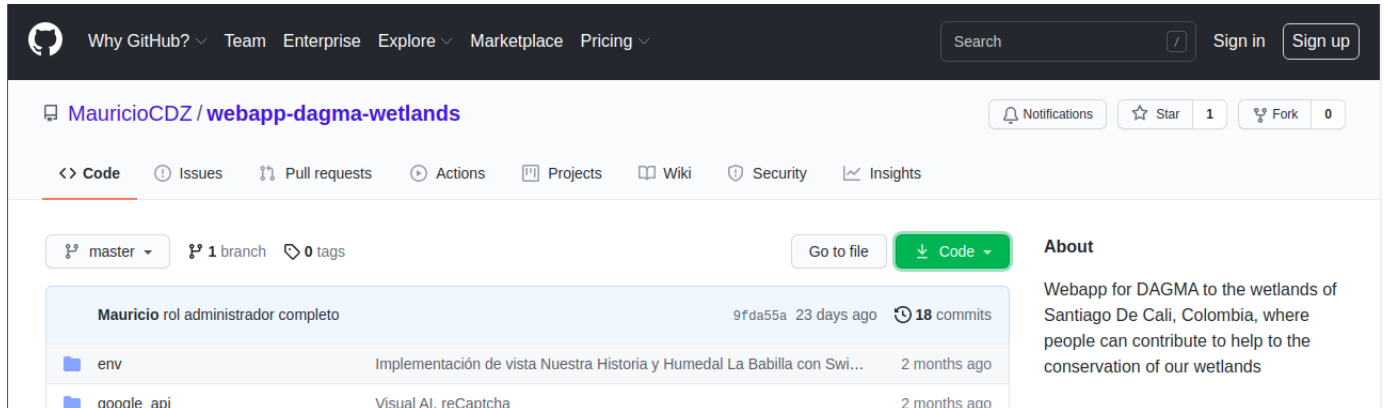


Figura 2.4: Repositorio webapp-dagma-wetlands. Disponible en <https://github.com/MauricioCDZ/webapp-dagma-wetlands>

### 2.0.6. Estructura de Desglose de Trabajo:

Llamado EDT, por sus siglas en español, estructura de desglose de trabajo; o WBS, por sus siglas en inglés, work breakdown structure; [Luc21]. Es una herramienta que permite organizar y estructurar los entregables en un proyecto de manera que puedan ser leídos en forma de actividades a realizar. El EDT debe comprenderse de manera sencilla tanto como para los participantes e interesados en él. Por ello, decidimos incluir esta estructura pues es vital para todo tipo de proyectos sean pequeños, medianos o grandes. En la figura 2.5, se puede observar a detalle la estructura que se siguió en el proyecto, desde el inicio hasta el despliegue del aplicativo.

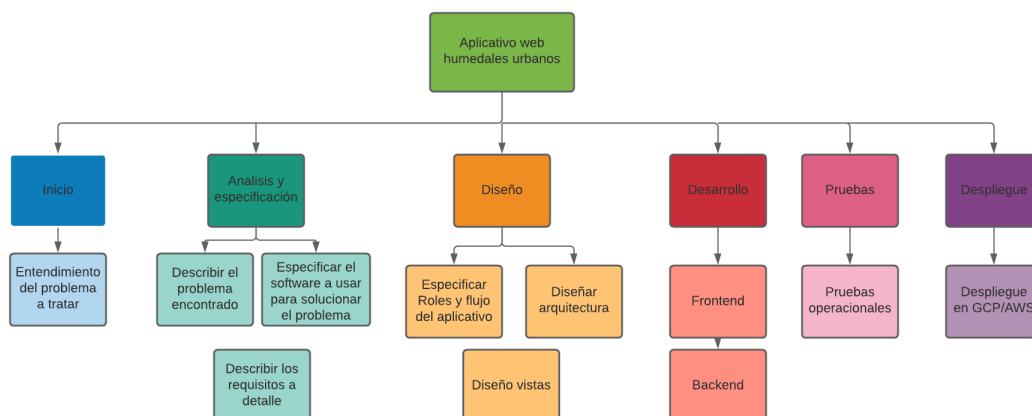


Figura 2.5: Diagrama estructura de desglose de trabajo

### 2.0.7. Gestión de actividades

Actividad		Precedencia	Duración estimada en horas	Responsables
A	Análisis del problema	...	20	Mauricio & Juan
B	Planteamiento del problema	A	15	Mauricio & Juan
C	Revisión de trabajos relacionados	B	15	Mauricio & Juan
D	Definir objetivos	C	3	Mauricio & Juan
E	Seleccionar herramientas	D	5	Mauricio & Juan
F	Requisitos Funcionales	D	7	Mauricio
G	Requisitos No Funcionales	D,E	7	Juan
H	Realización de Mockup	F	30	Mauricio & Juan
I	Selección de base de datos	D	2	Mauricio & Juan
J	Selección de frontend backend	I	3	Mauricio & Juan
K	Desarrollo de frontend	H	60	Mauricio & Juan
L	Realización modelo de base de datos	I	30	Mauricio & Juan
M	Desarrollo backend	L	40	Mauricio & Juan
N	Conexión Api Google	M	8	Mauricio
O	Conexión api reCaptcha	M	3	Juan
P	Realización de pruebas	O	20	Mauricio & Juan
Q	Despliegue en GCP/AWS	P	20	Mauricio & Juan

Cuadro 2.1: Diagrama de Gestión de actividades

### 2.0.8. Una SLR sobre aplicativos en pro de la conservación de ecosistemas, en especial humedales

El primer objetivo de esta investigación es la construcción y desarrollo de aplicativos para monitorear distintos ecosistemas, empleando diferentes técnicas y enfoques sobre el uso de esta información para el cuidado y conservación de estos. Por lo tanto, decidimos realizar una revisión sistemática de la literatura (SLR) para conocer de manera integral los diferentes aplicativos actuales en el contexto de cuidado y conservación de ecosistemas. En esta sección se presenta la metodología aplicada al desarrollo de nuestra SLR, que sigue las pautas propuestas por Kitchenham [BK07].

### 2.0.9. Preguntas de investigación

La SLR realizada aborda las siguientes preguntas de investigación:

- **PI1:** ¿Qué tecnologías se han usado en el monitoreo de distintos ecosistemas?
- **PI2:** ¿Cuáles son los riesgos que impactan en mayor grado a los humedales?
- **PI3:** ¿Qué técnicas se usan para identificar la sanidad de un humedal?
- **PI4:** ¿Cómo se utilizó la información recolectada para generar un valor ambiental en pro de la conservación?
- **PI5:** ¿Qué medio se estableció para la correcta comunicación entre ciudadanos y entidades ambientales?

#### 2.0.9.1. Proceso de búsqueda

Realizamos una búsqueda bibliográfica de artículos, journals publicados en ACM Digital Library, IEEE EXPLORE, EBSCO, IOP Science y SpringerLink. Estos fueron seleccionados debido a la afinidad con el tema y la calidad aceptable de sus publicaciones. La cadena de búsqueda utilizada fue:

*(“wetland” OR “swamp” OR “lakes” OR “forest” OR “lagoons” OR “marsh” OR “river” OR “bog” OR “bogland” OR “streams” OR “fen” OR “pond”) AND (“technology” OR “analytics” OR “QR code” OR “analysis” OR “research” OR “monitoring” OR “environmental conservation” OR “environmental care” OR “contribution” ) AND (“community” OR “people” OR “neighbor” OR “neighbour” OR “homebody” OR “tenant”) AND (“contamination” OR “sewage” OR “waste” OR “rubbish” OR “trash” OR “” OR “waste water” OR “wastewater” ) .*

Los criterios de inclusión consistieron en: i) fecha de publicación, se seleccionaron artículos publicados entre 2014 y 2020; ii) tipo de publicación, solo se seleccionaron artículos y journals; iii) el número de páginas y el idioma, excluimos las publicaciones con menos de ocho páginas y escritas en idiomas distintos del inglés o español; iv) relevancia, estudiamos títulos de artículos, abstract, resumen y verificamos la relevancia para nuestras preguntas de investigación. Además, cuando la base de datos ofreció la opción de filtrar por área, se considerará en las siguientes: ingeniería, ciencias

de la computación, ciencias ambientales, medio ambiente y ciencias de la tierra.

La primera búsqueda resultó en 2.231 artículos. Al inspeccionar el contenido de los artículos, excluimos 1854 durante la primera selección, luego otros 320 debido a la falta de relevancia o porque no fue posible identificar la información requerida por nuestra dimensión de caracterización, por ejemplo, algún ecosistema se nombraba y se decían algunas características de estos, pero el ecosistema como tal no tenía un papel mayor en la investigación por lo tanto no era el enfoque principal. Finalmente, se obtuvo un total de 57 artículos relevantes que se caracterizaron de manera integral.

### 2.0.10. Caracterización de dimensiones y recopilación de datos

Con el objetivo de una comprensión integral de los diferentes filtros que se aplicaron en la búsqueda, sus aplicaciones y las técnicas, nuestra SLR es una caracterización de aplicaciones de la ingeniería en ecosistemas, enfocándonos principalmente en los humedales y que en el contexto de este, estén involucradas las personas. Caracterizamos los artículos buscados a la luz de tres dimensiones: actividades importancia ambiental, manejo con la comunidad y análisis de datos para el monitoreo de ecosistemas. Esta subsección describe en un alto nivel los hallazgos en las dimensiones mencionadas Ver figura 2.6.

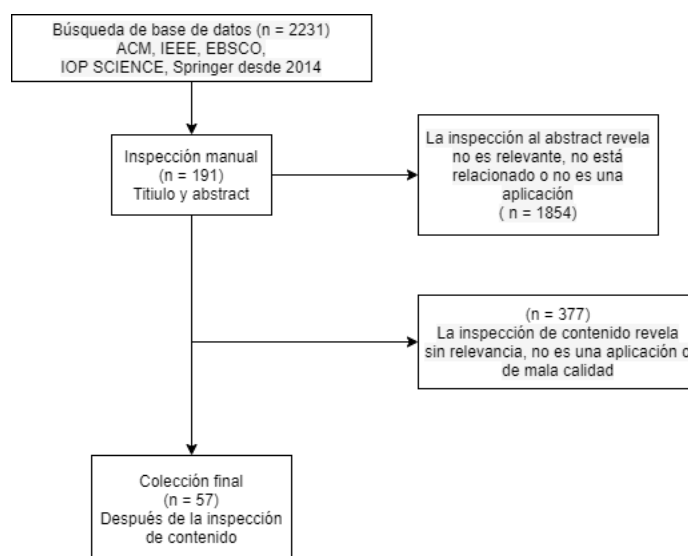


Figura 2.6: Proceso de búsqueda de artículos SLR

### 2.0.11. Conjunto final de trabajo relacionado

Con base en la extensa búsqueda que se realizó mediante SLR encontramos que no existen aportes que aborden el problema como nos lo imaginamos, dado que en un cúmulo de artículos el monitoreo de ecosistemas se realiza mediante la puesta en marcha de distintos sensores que captan

cambios en el ambiente del ecosistema. Por ejemplo, en Chunying W. [WCC<sup>+</sup>16] ellos realizaron la construcción y aplicación de un sistema de indicadores de diagnóstico de salud de humedales basado en sensación remota. Usando la tecnología de teledetección pudieron obtener parámetros relacionados con los humedales a gran escala repetidamente, lo que hizo posible una evaluación rápida y a gran escala de la salud de los ecosistemas de humedales.

Ellos seleccionaron hasta 18 indicadores que servían de diagnóstico de la salud. Luego usaron el modelo del Proceso de Jerarquía Analítica (AHP), para adquirir las ponderaciones de todos los indicadores y se aplicó este sistema de indicadores en el humedal Ruoergai en Sichuan, China. El sistema de modelo de diagnóstico se construyó a partir de los tres elementos del ecosistema de humedales, el paisaje y los factores sociales, que denominaron modelo conceptual "Factor-Paisaje-Sociedad". El sistema cuenta con 18 indicadores, incluidos 13 indicadores de estado, 5 indicadores de tendencia, en los que se pueden obtener 9 indicadores directa o indirectamente a partir de datos de teledetección, entre los cuales cabe listar los siguientes: Calidad del agua, Tendencia de cambio de la calidad del agua, Tasa de garantía de la fuente de agua, Contenido de agua del suelo, Valor de pH del suelo, Contenido de metales pesados del suelo, Índice de biodiversidad, Productividad primaria neta, Índice de idoneidad del hábitat y Tasa de cambio del área de humedales.

Por otro lado, también se estudiaron distintos artículos que hablaban del impacto del desarrollo urbano en ecosistemas, como los humedales de agua dulce. Por ejemplo, el estudio realizado por R, Carroll [CRW19] en el cual miden el impacto geoquímico del desarrollo urbano en frágiles humedales de agua dulce, en este artículo explican cómo se realizó un análisis químico en los pantanos de Blue Mountains Upland, donde se explica que el uso de materiales impermeabilizantes en las infraestructuras de los drenajes genera desafíos para los ecosistemas de agua dulce a nivel mundial. Luego de realizar un análisis profundo, se halló que algunas de las implicaciones de la urbanización de la cuenca incluyen hidrología natural alterada, erosión, invasiones de malezas y química del agua modificada. En el estudio se evaluaron 4 pantanos en un área natural y otros 4 en una zona urbana. Los pantanos urbanos presentaban una cubierta impermeable más alta y una química del agua modificada, con un pH elevado, una conductividad eléctrica e iones elevados en comparación con los pantanos naturales. Este estudio refleja que en un mundo cada vez más urbano, es necesario considerar las consecuencias ecológicas de la urbanización para orientar los enfoques de gestión futuros

Otro ejemplo bastante claro de como aparatos electrónicos como sensores son usados para monitorear ecosistemas es el esquema propuesto por Zhe-Ren Yan [YFQ<sup>+</sup>14] el cual es un esquema de monitoreo ecológico basado en una red de sensores inalámbricos en el lago Baisha de la Reserva Natural Nacional del Humedal Nanji, este texto se basa en la urgente necesidad de que haya un monitoreo a la Reserva Natural Nacional del Humedal Nanji, esto porque en la primavera y el verano, el nivel del agua sube de modo que el pueblo de Nanjishan se convierte en una isla aislada. Mientras que, durante otoño y principios de la primavera, el nivel del agua es bajo, por lo que las áreas de humedales de Nanji están expuestas principalmente. Es ahí donde los habitantes aprovechan en realizar actividades de acuicultura, como el cultivo y la recolección de peces, camarones, tornillo y mejillones. Por otro lado, esta proporciona un hábitat de invernada fundamental para una gran

cantidad de aves migratorias en las rutas de migración de Asia oriental y Australia.

Se escogió este artículo como material bibliográfico dado que, aunque se relacionen temas de electrónica y circuitos, se pueden analizar y adoptar metodologías y posturas que los autores mencionan, además de poder brindar un abrebocas a parámetros de salubridad de los humedales.

*El monitoreo del medio ambiente del lago Baisha es esencial para estudiar el flujo de energía y la circulación de materiales en los ecosistemas de humedales.* Gracias a este artículo, podemos observar cómo la población circundante a los humedales puede provocar daños en este, como por ejemplo, haciendo uso de la agricultura.

En contraste, también encontramos cómo la propia comunidad puede ayudar al cuidado de los ecosistemas, en el artículo escrito por Niva Rana M [MSR20] se afirma que existe amplia evidencia para demostrar que la participación de la comunidad se ha convertido en un requisito previo para garantizar el desarrollo y el cambio, haciendo énfasis en los proyectos exitosos dado que estos se caracterizan de manera única en términos de prioridad por parte de la comunidad; dejando claro la comprensión de las responsabilidades y competencia de las tecnologías utilizadas. Este artículo nos menciona el humedal Ramsar de la ciudad de Guwahati, India; un humedal rico en biodiversidad que tiene un paisaje foresta circundante llamado Deepor Beel. Se hace hincapié a una crisis actual de conservación debido a la rápida urbanización, los conflictos y la insignificante cooperación de las comunidades locales. Por tal razón, la investigación se enfoca en estrategias contextuales para integrar la participación comunitaria en el manejo de humedales urbanos y la conservación ambiental a través de posibilidades de derechos biológicos.

Primero identifican los problemas que están llevando al deterioro del humedal Ramsar:

- No cooperación comunitaria.
- Comunidades tribales diversas.
- Poca conciencia de los beneficios ambientales y socioculturales.
- Falta de instalaciones básicas como transporte público, empleo y medios de vida alternativos.
- Instalaciones educativas y sanitarias inadecuadas en las aldeas periféricas.
- Falta de inversión adecuada para el desarrollo del turismo sostenible y el eco-turismo.
- Enfoques contradictorios entre varias autoridades y partes interesadas (pesca, bosques, departamentos de agua, municipalidad, ONG, comunidades locales).

Por ello, se proponen una serie de soluciones de forma conjunta entre la comunidad y las autoridades para la conservación del humedal Ramsar entre las cuales están: oportunidad recreativa, mercado de turismo especial para eco-turismo, deportes acuáticos, fotografía, uso racional de los recursos naturales, oportunidades de investigación, preservación de especies e identidades culturales en peligro de extinción, desarrollo de la acuicultura, medios de vida alternativos para obtener beneficios fiscales.

En contraste al problema del Humedal Ramsar encontramos como Long H.[NHN<sup>+</sup>18] realiza una gestión inteligente y conectada de los recursos hídricos a través de las redes sociales y la participación de la comunidad. En el texto se presenta una implementación de una plataforma inteligente llamada WaterScope, esta contiene múltiples fuentes de datos, como datos sobre el nivel del agua, datos de redes sociales y artículos relacionados con el agua que posteriormente son usados para la gestión de recursos hídricos. Este además brinda un entorno en el que se permite:

- Pronosticar los niveles de agua subterránea.
- Identificar problemas relacionados con el agua.
- Compartir conocimientos y experiencia entre las partes interesadas.
- Aportar nuevos conocimientos sobre el suministro de agua y la gestión de recursos.

Esta plataforma puede ser usada como guía, para reflexionar sobre el contacto con el agua dulce que poseen los humedales y el cuidado que esta debe tener. Además de poder conectar la comunidad de interesados (stakeholders) en el cuidado y preservación de ecosistemas que no solo busquen la supervivencia de la flora y el agua, si no también de los que en ella habitan.

# Desarrollo del Proyecto

---

## 3.0.1. Búsqueda y comparación de herramientas

### 3.0.1.1. Frontend

En el desarrollo de aplicativos es una buena práctica separar la parte lógica del sistema y la interfaz de usuario que podrá ser interactuada. El backend realiza todo el procesamiento de los datos de entrada, en otras palabras gestionan las peticiones de los usuarios y las procesan para poder solucionarlas. En este sentido, la necesidad específica del frontend es recolectar los datos para que sean procesados posteriormente.

Las *Dynamic web page* (en español páginas web dinámicas) son desarrolladas con las capas anteriores, estas generan diferente información a partir de las peticiones enviadas por el cliente, permitiendo finalmente la visualización del procesamiento de la petición.

Los principales objetivos de este desarrollo consisten en la accesibilidad y el rendimiento. El primero se refiere a la compatibilidad del aplicativo, entiéndase que puedan ser accedidos desde un teléfono, tableta o computador y que se muestre no solo ordenadamente y con estilo, sino que la información sea consistente en todos los dispositivos. Por otro lado, el rendimiento es el tiempo que necesita para cargar rápidamente la información que se solicita, ya que los tiempos prolongados suele ser un problema para el usuario.

- **Bootstrap:** Es una biblioteca de código abierto diseñada por los desarrolladores de Twitter para sitios y aplicaciones web. Además de contener numerosos componentes web, este también posee extensiones de JavaScript, diferenciándola de otros frameworks que solo se enfocan el desarrollo frontend. Este además es el segundo proyecto más destacado en GitHub y por ende la comunidad de los que la utilizan es bastante amplia. En la tabla 3.1 se listan algunas de sus ventajas y desventajas al usar esta biblioteca.
- **Tailwind:** Aunque Tailwind no sea respaldado por ninguna empresa grande (llámese Google, Twitter, Facebook o Amazon), esta intenta abrirse un espacio en el área del frontend. A diferencia de los demás frameworks que buscan extenderse en la sección de escritura de estilos y la excesiva utilización de CSS, Tailwind maneja el uso de clases. Aunque se forme un sinnúmero de clases colocadas en el HTML para un solo componente web, el creador Adam Watham tiene una explicación y es que el CSS basado en utilidades es superior al CSS semántico. Es por ello que en Bootstrap 5 se buscó seguir la nueva tendencia y realizar un acoplamiento de Bootstrap a las utilidades. La tabla 3.2 muestra algunas de sus ventajas y desventajas.

Ventajas	Desventajas
No se requiere poseer el complicado conocimiento de posicionamiento de componentes web.	Si se quiere obtener un diferente resultado, es complicado a la hora de realizar modificaciones, los rellenos tienen sus colores predeterminados, las filas y los contenedores se acoplan exactamente a las mismas proporciones y no se puede realizar una gran innovación
Su compatibilidad con navegadores web es bastante amplia.	Repetitividad en la experiencia de usuario, dado que muchos de los sitios web que usados, poseen esta tecnología.
Es respaldado por Twitter y su nivel de confianza de la comunidad es fuerte (especialmente en grandes compañías)	Bootstrap contiene archivos CSS de gran tamaño que deben descargarse en el dispositivo de los usuarios, lo que puede ralentizar las vistas iniciales de la página.

Cuadro 3.1: Ventajas y desventajas Bootstrap.

- Materialize:** Otra tendencia es el Material Design que fue diseñado por Google y es ahí donde Materialize realiza su aporte en creación de componentes que tengan una paleta de colores siguiendo una rúbrica y que se cuide la estética, sin demeritar la funcionalidad y rendimiento. Aunque se vea completo, este no provee suficientes opciones de personalización. La tabla 3.3 muestra algunas de sus ventajas y desventajas.

Ventajas	Desventajas
Materialize tiene muchas opiniones sobre cómo se comportan e interactúan visualmente los elementos de UX.	A veces, Materialize no maneja correctamente ciertos anidamientos de elementos, lo que da como resultado un aspecto extraño.
Materialize es excelente en su elegante diseño de material que ofrece listo para usar.	No es compatible con algunos navegadores antiguos.
Diseño web adaptable o “responsive”	Grande / pesado.
Facil de usar	
Buena documentación	

Cuadro 3.3: Ventajas y desventajas Materialize

### Elección frontend:

Se decidió elegir para el frontend del aplicativo *Bootstrap* dado que es una biblioteca CSS muy buena para un usuario promedio o incluso principiante, ya que es muy simple de usar y tienen documentación muy detallada sobre cómo usarla. Además de ello, mantiene una amplia compatibilidad con los navegadores, lo que es de vital importancia dado que los usuarios pueden acceder desde Chrome,

Ventajas	Desventajas
Una de las principales ventajas del marco Tailwind CSS es que se evita escribir CSS por uno mismo. Lo que ahorra tiempo de diseño y desarrollo. Solo necesita aplicar clases CSS a el elemento HTML.	Con Tailwind, está aplicando clases atómicas a sus elementos HTML. Entonces, al diseñar un elemento, terminará con toneladas de clases aplicadas a sus elementos y haciendo que el archivo HTML sea difícil de leer. [cp-20a]
Con Tailwind CSS, solo necesita aplicar text-sm, text-lg o text-xl a los elementos HTML, diciendo respectivamente que desea un tamaño de fuente pequeño, grande o extra grande. De esta manera se deja de escribir cualquier número sin sentido como 15px o 18px.	Tailwind CSS no proporciona componentes con estilo por defecto como en Bootstrap o Bulma. Se necesita implementarlo desde cero. Posiblemente se puede perder tiempo para el desarrollo y diseño. Por ejemplo, Tailwind CSS no proporciona botón, barra de navegación, encabezado, etc.
Otra ventaja relacionada con estas clases predefinidas de Tailwind CSS es que se deja de ajustar el diseño propio. Debido a que Tailwind CSS ya ha definido todas las clases de CSS, no se puede especificar ningún número aleatorio para el margen, relleno, tamaño de fuente, etc.	Lleva tiempo volverse productivo con Tailwind, dado que se necesita aprender un montón de nombres de clases.

Cuadro 3.2: Ventajas y desventajas Tailwind.

Mozilla, Edge o el navegador de su preferencia. Por otra parte al usar Bootstrap, se puede usar mas tiempo en el trabajo de desarrollo, dada la curva rápida de aprendizaje de Bootstrap, y así obtener un sitio web atractivo y funcionando rápidamente.

### 3.0.1.2. Backend

El backend es la parte lógica e invisible pero fundamental en todo proyecto que incluya software web, por tanto es de vital importancia darle la relevancia que se merece. El backend se ocupa de todos los procesos que el sitio web necesita para funcionar correctamente, dado que ahí se define explícitamente toda la lógica del sitio, tomando los datos desde el frontend o la base de datos para producir algún resultado que se muestra ya sea al cliente usando la interfaz de usuario o simplemente guardando en la base de datos para un posterior uso. Estos procesos no se muestran, pero son importantes para el manejo normal del sitio.

El elegir un lenguaje de programación para el backend es muy importante para el desarrollo del proyecto. Dado que existen numerosos lenguajes, cada uno con sus ventajas y desventajas. Vamos a numerar una serie de ellos, partiendo de lo más utilizados en la actualidad para el desarrollo web, teniendo muy en cuenta características como eficiencia, compatibilidad, escalabilidad.

Al escoger un lenguaje de programación también se debe seleccionar un framework que este ofrezca, estos ofrecen un entorno pensado para hacer más sencilla la programación de cualquier aplicación o herramienta actual, puesto que ofrecen múltiples módulos y librerías que ya tienen implementados una serie de funciones por lo que se ahorra código, tiempo y esfuerzo. Es así, que el framework ayuda en gran medida a la estructuración de la lógica del aplicativo web y da una conexión a la base de datos.

A continuación se nombrarán 3 lenguajes de programación muy usados actualmente en el ámbito de desarrollo web, con una breve descripción, sus ventajas y desventajas. Junto a ellos se mencionará un framework basado en su lenguaje, con una breve descripción.

- **PHP:** Es un lenguaje de programación de código abierto orientado a objetos creado por Rasmus Lerdorf en el año 1994 [Ano20b], para desarrollar páginas y aplicaciones web. Este puede ser incrustado de manera muy flexible con HTML5. El código se ejecuta en el servidor, genera HTML y se lo envía al cliente.

Ventajas	Desventajas
Lenguaje totalmente libre y abierto	Solo se ejecuta en un servidor y se necesita un servidor web para que funcione.
Es un lenguaje multiplataforma.	Si no se configura y/o protege correctamente se dejan abiertas muchas brechas de seguridad que a la larga tendra problemas.
Los entornos de desarrollo son de rápida y fácil configuración.	Los frameworks PHP aprendieron a usar las funcionalidades integradas de PHP para evitar escribir código adicional.
Soporta el modelo vista controlador este patrón reduce el tiempo de desarrollo de cualquier sistema web, incluso tiene la opción de trabajar por separado el frontend y backend.	El uso de más funciones de los framework y las herramientas de PHP provoca un rendimiento deficiente de las aplicaciones en línea.

- **Framework PHP:** Laravel es un framework PHP de código abierto, robusto y fácil de entender. Sigue un patrón de diseño modelo-vista-controlador. Laravel reutiliza los componentes existentes de diferentes marcos, lo que ayuda a crear una aplicación web. La aplicación web así diseñada es más estructurada y pragmática. Laravel ofrece un rico conjunto de funcionalidades que incorpora las características básicas de los frameworks PHP como CodeIgniter, Yii y otros lenguajes de programación como Ruby on Rails. Laravel tiene un conjunto muy rico de características que impulsan la velocidad del desarrollo web.[Ano20c]
- **Java:** Java es un lenguaje de programación orientado a objetos, de uso general y basado en clases, diseñado para tener menores dependencias de implementación. Es una plataforma informática para el desarrollo de aplicaciones. Por lo tanto, Java es rápido, seguro y confiable.

Es ampliamente utilizado para desarrollar aplicaciones en computadoras portátiles, centros de datos, consolas de juegos, supercomputadoras científicas, teléfonos celulares, etc.[Ano20d]

Ventajas	Desventajas
Java reduce las amenazas y los riesgos de seguridad al evitar el uso de punteros explícitos. Un puntero almacena la dirección de memoria de otro valor que puede causar acceso no autorizado a la memoria.	Java consume memoria y es significativamente más lento que los lenguajes nativos como C o C ++. También es lento en comparación con otros lenguajes como C y C ++ porque cada código tiene que ser interpretado al nivel de la máquina.
Los programas Java son baratos de desarrollar y mantener, ya que estos programas dependen de una infraestructura de hardware específica para ejecutarse. Se pueden ejecutar fácilmente en cualquier máquina que reduciendo el costo adicional de mantenimiento.	Java proporciona una apariencia no tan atractiva de la GUI. Aunque hay muchos constructores de GUI en Java para crear la interfaz gráfica, todavía no son adecuados para crear una interfaz de usuario complicada.
Java es un lenguaje orientado a objetos que nos ayuda a mejorar la flexibilidad y reutilización del código. Usando el concepto de POO, podemos reutilizar fácilmente el objeto en otros programas.	Java trabaja principalmente en el almacenamiento y no se centra en la copia de seguridad de datos. Este es un gran inconveniente que le hace perder el interés y las valoraciones entre los usuarios.[Ano20a]
Java es un lenguaje multiproceso es decir , se pueden ejecutar más de un subproceso al mismo tiempo. Un hilo es la unidad más pequeña de un proceso. El subproceso múltiple ayuda a aprovechar al máximo la CPU.	Los códigos Java son detallados, lo que significa que hay muchas palabras y muchas oraciones largas y complejas que son difíciles de leer y comprender. Esto puede reducir la legibilidad del código.

- **Framework Java:** Spring Framework es uno de los frameworks de backend más robustos y versátiles. Contiene varios módulos diferentes que proporcionan a los programadores una variedad de herramientas de aplicación. Este framework utiliza los conceptos de Inyección de Dependencias e Inversión de Control para configurar los componentes de la aplicación en acoplamientos sueltos, lo que ayuda a dar a la aplicación más flexibilidad en sus respuestas.[Jim20]
- **Python:** Es un lenguaje de programación interpretado, orientado a objetos y de alto nivel. Sus estructuras de datos integradas de alto nivel, combinadas con tipado dinámico y enlace dinámico, lo hacen muy atractivo para el desarrollo rápido de aplicaciones, así como para su uso como lenguaje de scripts o para conectar componentes existentes. La sintaxis simple y fácil de aprender de Python enfatiza la legibilidad y, por lo tanto, reduce el costo de mantenimiento del programa. [cpp]

Ventajas	Desventajas
Python proporciona una amplia biblioteca que contiene código para varios propósitos como expresiones regulares, generación de documentación, pruebas unitarias, navegadores web, subprocesos, bases de datos, CGI, correo electrónico, manipulación de imágenes y más.	La ejecución línea por línea del código a menudo conduce a una ejecución lenta. La naturaleza dinámica de Python también es responsable de la velocidad lenta de Python porque tiene que hacer el trabajo adicional mientras ejecuta el código.
El uso de Python ayuda a mejorar la productividad dado la simplicidad del lenguaje y las extensas bibliotecas hacen que los programadores sean más productivos que los lenguajes como Java y C ++. Como se necesita escribir menos, se pueden hacer más cosas.	El lenguaje de programación Python usa una gran cantidad de memoria. Esto puede ser una desventaja al crear aplicaciones cuando preferimos la optimización de la memoria.
Python viene bajo la licencia de código abierto aprobada por OSI. Esto lo hace libre de usar y distribuir. Puedes descargar el código fuente, modificarlo e incluso distribuir tu versión de Python. Esto es útil para organizaciones que desean modificar algún comportamiento específico y usar su versión para el desarrollo.[cpa21]	La capa de acceso a la base de datos de Python es primitiva y poco desarrollada en comparación con las tecnologías populares como JDBC y ODBC.
Es un lenguaje interpretado dado que las declaraciones se ejecutan una a una, la depuración es más fácil que en los lenguajes compilados.	Es débil para el desarrollo móvil.

- Framework Python:** Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito.[cpd21]

#### Elección de backend:

Para la elección del backend discutimos cada uno de los 3 lenguajes anteriormente mencionados junto con su framework, analizando a detalle las ventajas y desventajas de cada uno. En cuanto a Java, luego de una investigación nos dimos cuenta que si bien en Spring se pueden realizar proyectos web de buen alcance, aspectos como la sintaxis del lenguaje, el consumo de memoria y la complejidad del framework nos hicieron descartar este framework. Por otra parte, el framework Laravel (Php) mostró grandes virtudes que nos parecieron muy importantes, puesto que la mayoría de sitios web están implementados en este lenguaje ya sea usando Laravel u otras herramientas muy conocidas como Zend o codeigniter , sin embargo, la poca

---

experiencia por parte de los integrantes del equipo en el lenguaje Php jugó un papel muy importante para descartar este framework. Por último analizamos a mismo nivel de detalle el framework Django, en este vimos características similares a Spring, salvo que Python es un lenguaje más simple que Java y por tanto la curva de aprendizaje es considerablemente menor. Además, la experiencia por parte del equipo en el conocimiento teórico y práctico en Python es superior que el de Java. Si bien, el conocimiento en Python del equipo fue un factor para escoger este lenguaje como backend en conjunto con Django, no fue el factor más importante si no, elementos especiales que observamos en Django que nos hicieron decidir por optar el uso de este. A continuación varios de estos elementos:

- Viene por defecto con algunas medidas de seguridad como para que no haya SQL Injection, no haya Cross site request forgery (CSRF) o no haya Clickjacking por JavaScript. Django se encarga de manejar todo esto de una manera muy fácil.
- Es muy escalable, algunos de los sitios más activos de la Web aprovechan la capacidad de Django para escalar de forma rápida y flexible de manera que que funcionen de forma veloz y sea estable.
- Django usa el patrón modelo MVC: Modelo-Vista-Controlador aunque lo redefine como Modelo-Vista-Template. Es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento. [cpm21]

### 3.0.1.3. Base de datos

Una base de datos es una herramienta que permite organizar datos que se relacionen en unas características y que pueda ser buscado un registro en tiempo razonable. Estas además son de gran utilidad a la hora de realizar un análisis a los datos que estas contienen, ya sea, porque se esté realizando aprendizaje o para dimensionar los conjuntos que existen y cómo estos se diferencian de otros conjuntos. Existen diferentes tipos de bases de datos, entre ellos se enunciarán los siguientes.

- **Bases de datos relacionales:** Estas bases de datos son las tradicionales y se usan para poder visualizar la información fácilmente, dado que en su mayoría las tablas están llenadas en todos sus campos. Las tablas pueden ser relacionadas con otras por medio de un enlace lógico llamado *join*. Por último el lenguaje para poder manipular este tipo de base de datos es llamado **SQL** (*Structured Query Language*).
- **Base de datos NoSQL:** En las bases de datos NoSQL, se debe su nombre a que se minimiza el uso del lenguaje SQL o se usa exclusivamente como revisión de correctitud, pero nunca como consulta. Esto se debe a que en una base datos con lenguaje SQL, los datos se encuentran distribuidos con distintos atributos, mientras que en una NoSQL, los atributos se

encuentran en una misma columna. A diferencia de las relacionales esta no permite el uso del *join*, anteriormente descrito.

- **Base de datos orientada a objetos:** Este tipo de bases de datos usa como principio la noción de objetos y clases. La principal ventaja de este tipo de base de datos es que se puede encapsular un método o atributo de un objeto, permitiendo de esta manera aumentar el nivel de protección. Además se puede usar este tipo para especificar objetos más complejos, ya que están definidos con un lenguaje de programación.
- **Base de datos de grafos:** Se basan en la noción de grafos y para poder analizar la información que contiene un registro, se debe analizar un nodo y sus conexiones. Para la analítica de grafos, se requiere el uso de una base de datos que permita el uso específico de grafos, lo que permite el análisis dimensionales por la unión de distintos nodos.

#### 3.0.1.4. Elección base de datos:

Se decidió escoger una base de datos relacional dado el volumen de datos que el aplicativo maneja, que no implica mover grandes cantidades de estos. Teniendo en cuenta que el número de tablas y relaciones no es tan elevado y de la misma manera el número de transacciones por minuto; no se hace necesario un motor NoSQL o grafos. Además, en un Sprint Review realizado junto con el personal del Dagma, se pide que el aplicativo se maneje usando una base de datos relacional y en especial una con la que laboran a menudo en este sector publico llamada MySQL, por ello se decide usar este sistema de gestión de bases de datos relacionales.

**MySQL:** MySQL es un sistema de administración de bases de datos relacionales de código abierto (RDBMS) utilizado por un gran número de aplicaciones web. Está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

### 3.0.2. Diagramas de casos de uso

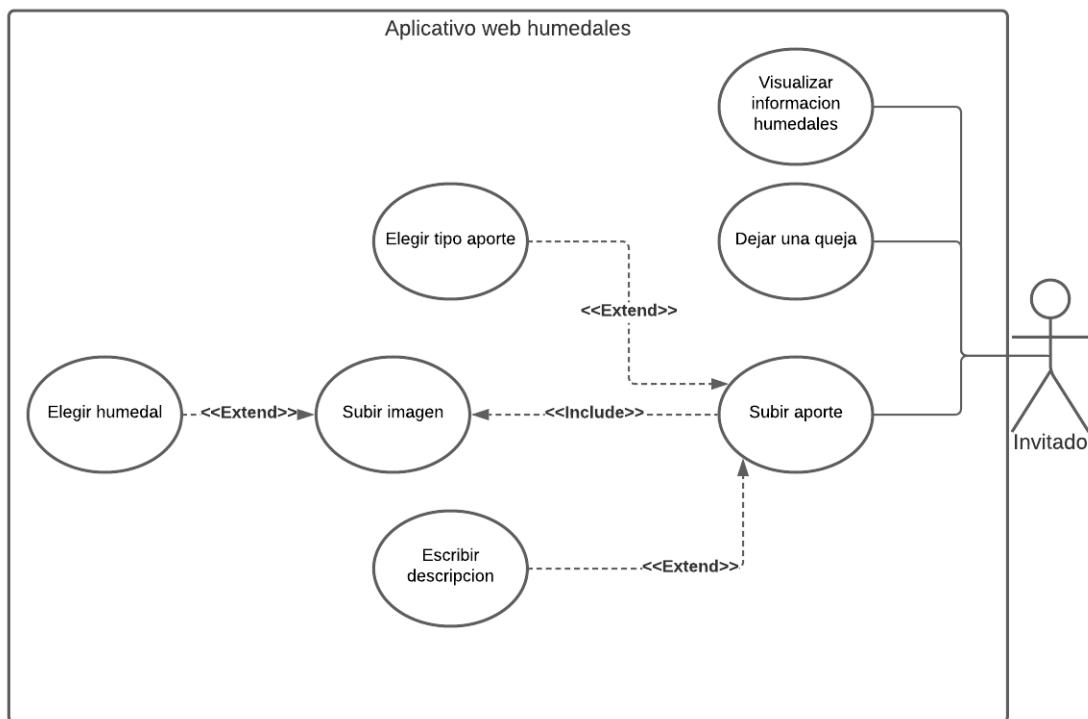


Figura 3.1: : Diagrama de Casos de uso para un usuario invitado

**Diagrama caso de uso usuario registrado:** Este diagrama extiende el anterior, de manera que un usuario registrado puede realizar las mismas acciones que un usuario invitado, más sus propias acciones.

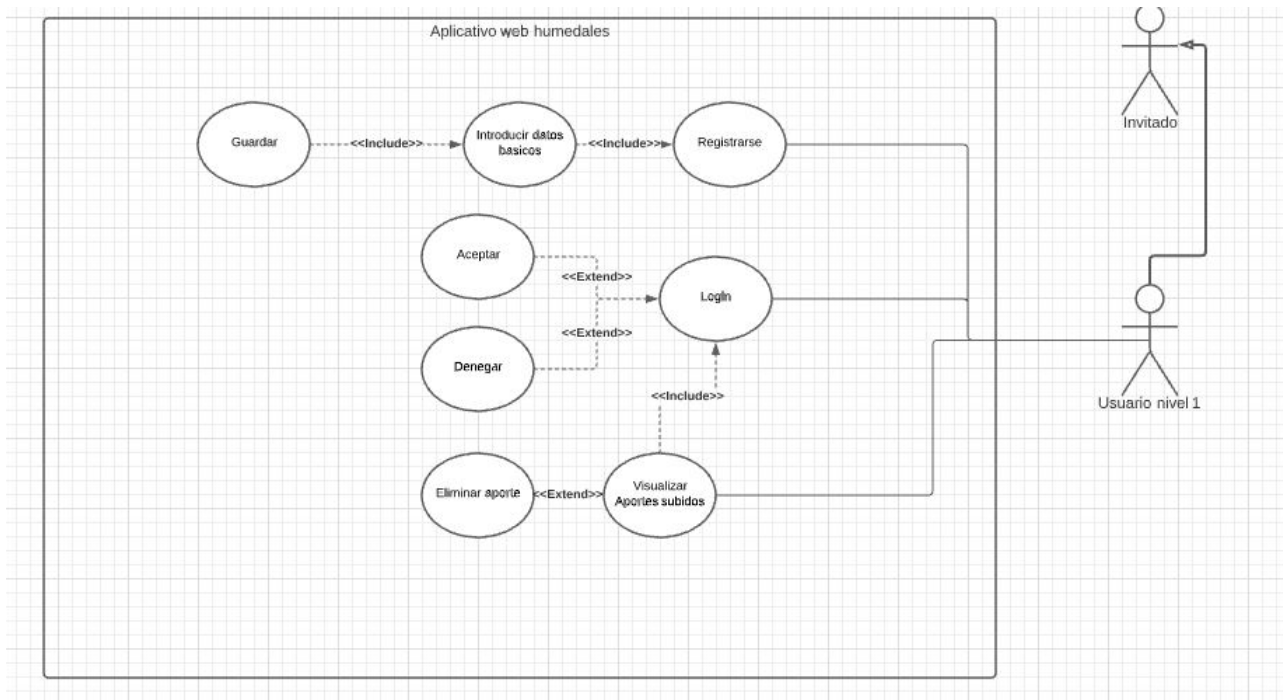


Figura 3.2: : Diagrama de Casos de uso para un usuario registrado

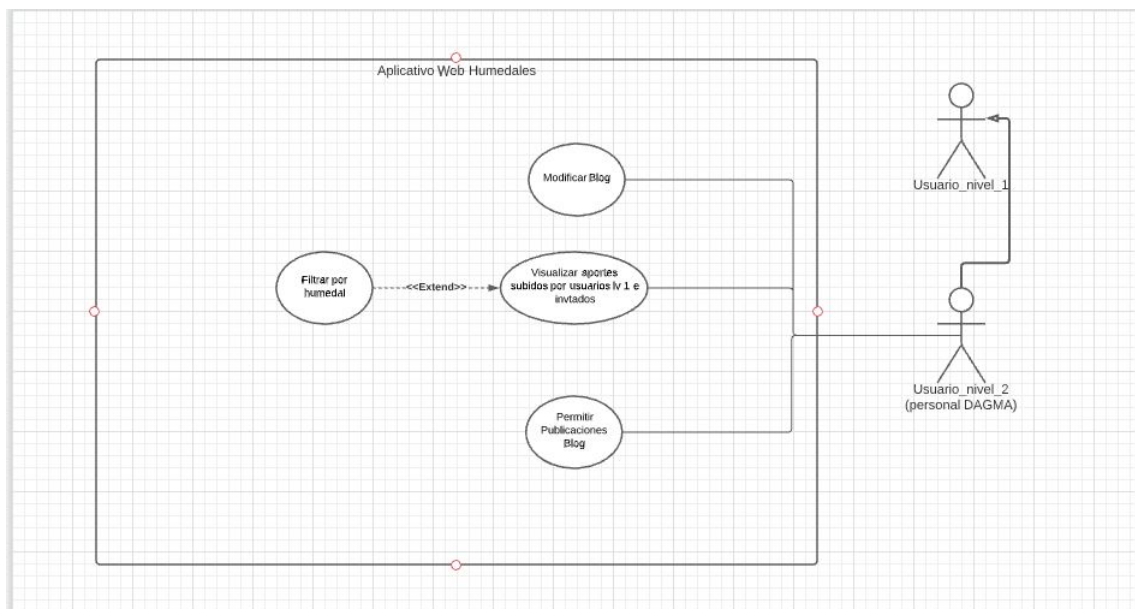


Figura 3.3: : Diagrama de Casos de uso para un usuario privilegiado(Dagma)

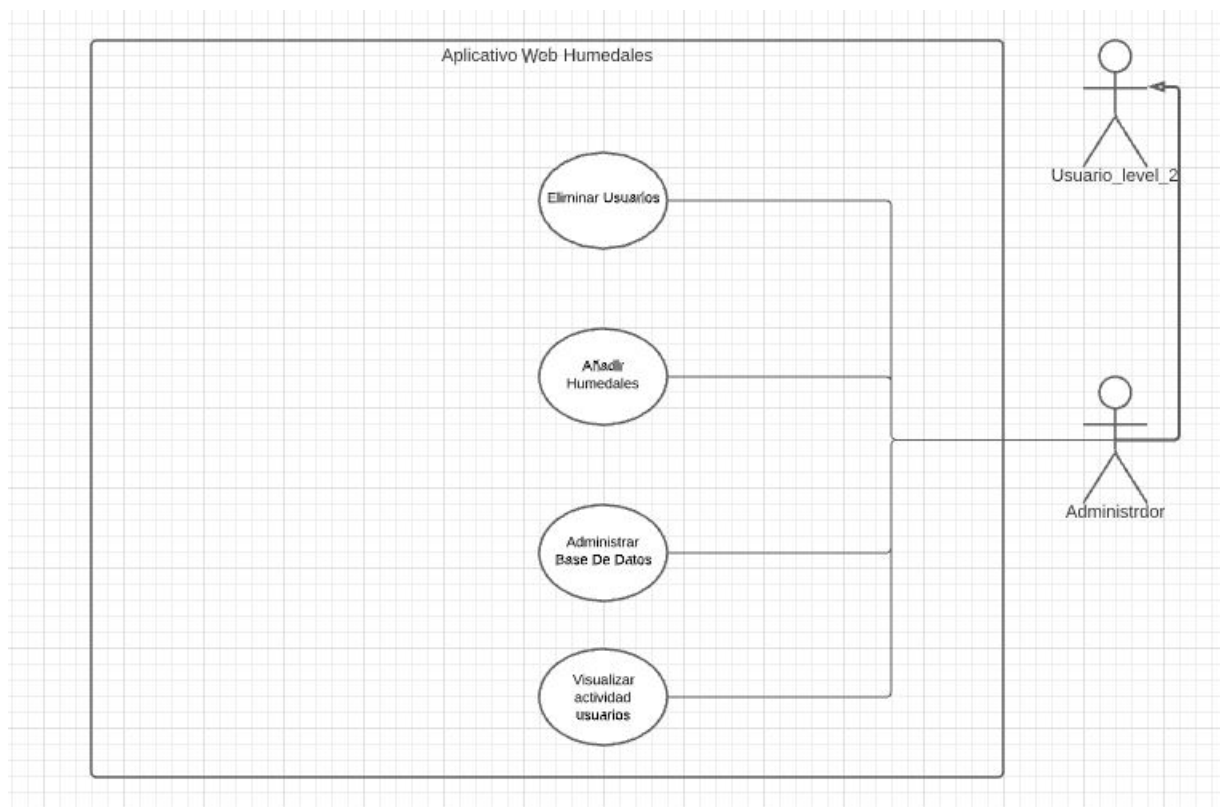


Figura 3.4: : Diagrama de Casos de uso para un administrador

### 3.0.3. Prototipado y Desarrollo

Para esta etapa del proyecto se decidió utilizar una plataforma de diseño colaborativo llamada *Marvelapp*, la cual nos sirvió en gran medida para la realización de prototipos los cuales se mostraron en su momento al director del proyecto y al DAGMA para el debido feedback. Una vez realizadas estas reuniones, se tomaron en cuenta todas las recomendaciones para ajustar aun más el diseño a los distintos requerimientos.

A continuación, capturas de pantalla de la ultima versión de prototipo realizado en *MarvelApp*, se puede consultar en <https://marvelapp.com/prototype/28990966>.



Figura 3.5: : Vista Home



Figura 3.6: : Vista menú humedales

## Humedal La Babilla

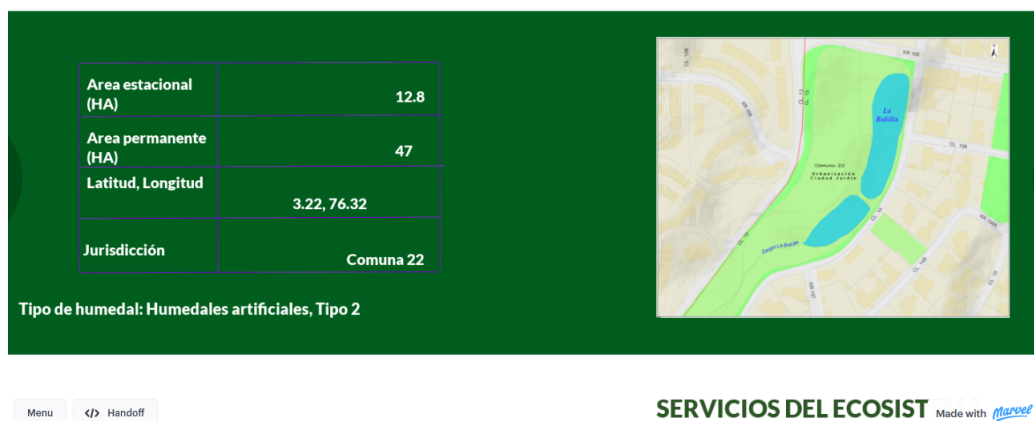


Figura 3.7: : Vista humedal

### 3.0.4. Modelo de datos:

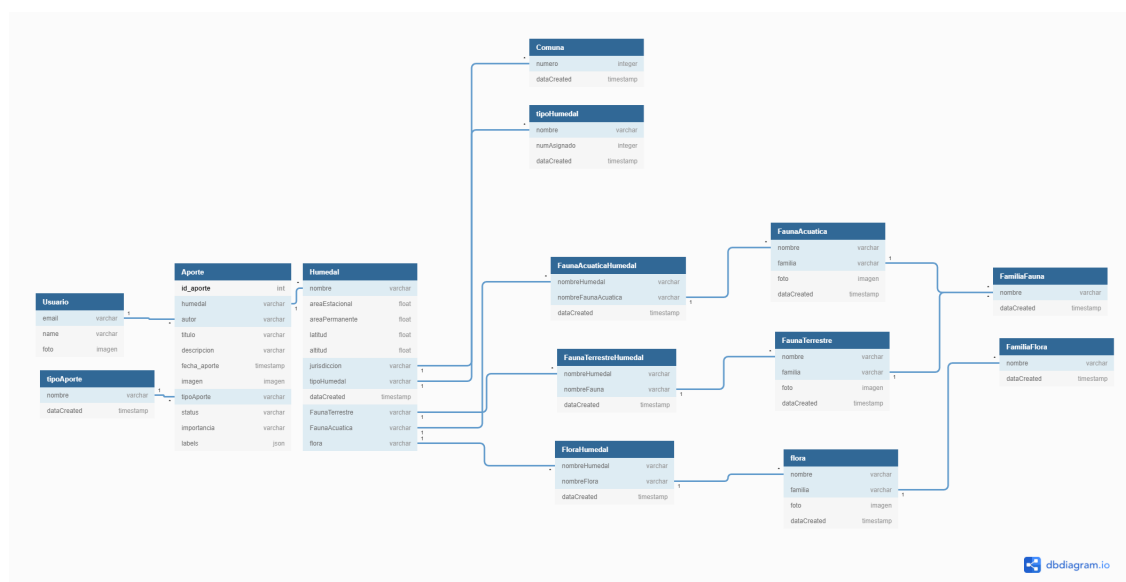


Figura 3.8: : Modelo de datos aplicativo

- Usuario:** Esta tabla es la encargada de guardar toda la información personal de los usuarios que se registren en el aplicativo; además, por defecto esta creado un usuario invitado al cual se le asignan los aportes realizados por los usuarios que no desean registrarse.

- **Aporte :** Esta tabla es la encargada de guardar toda la información del aporte realizado por un ciudadano, a su vez posee una llave foránea que lo relaciona con el humedal en cuestión.
- **Humedal:** Esta tabla guarda la información más relevante de un humedal, a su vez se relaciona mediante llaves foráneas con las tablas correspondientes a su jurisdicción, el tipo de humedal al cual pertenece, su flora, su fauna terrestre y por último su fauna acuática.
- **Fauna y Flora:** Las tablas Fauna Acuática Humedal, Fauna Terrestre Humedal y Flora Humedal guardan la respectiva información para el humedal en cuestión, a su vez se relacionan mediante una llave foránea a cada una de las especies de fauna y flora.

### 3.0.5. Arquitectura

#### 3.0.5.1. Criterios de elección

Como punto de partida en el desarrollo arquitectónico, decidimos escoger los siguientes atributos de calidad según los requisitos no-funcionales:

- Disponibilidad
- Rendimiento

Estos atributos satisfacen gran parte de los requisitos no funcionales referenciados en la tabla 3.5. Estos son específicamente los requisitos con IDs: RNF-2, RNF-4 y RNF-5, los cuales especifican los tiempos de respuesta esperados de carga luego de alguna acción realizada por el usuario. Consideramos que para este caso concreto, tener un sistema con buena disponibilidad (pocos tiempos inactividad o tiempos de inactividad programados) y tiempos de respuesta razonables (menor a los 3 segundos para la mayoría de eventos).

En los aspectos que la arquitectura final tendría puntos débiles es en el área de mantenibilidad. Si bien en los requisitos se tiene la necesidad de poder re-establecer el sistema mediante copias de seguridad automáticas (y esto siendo complemento a las decisiones de arquitectura en cuanto al atributo de disponibilidad), no fue el centro de atención a la hora de crear las arquitecturas.

#### 3.0.5.2. Arquitecturas candidatas

A continuación se adjuntan los diagramas para las 3 arquitecturas candidatas exploradas:

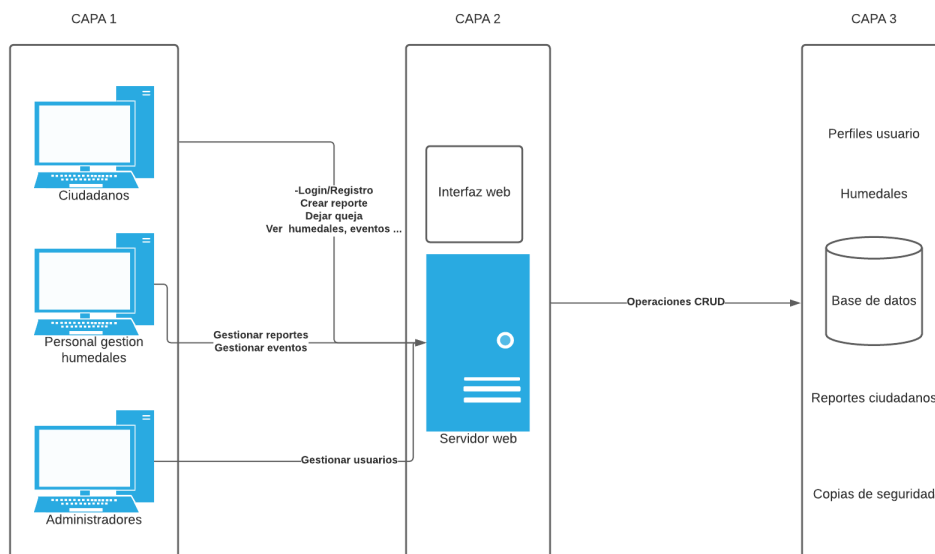


Figura 3.9: Diagrama arquitectura Cliente Servidor

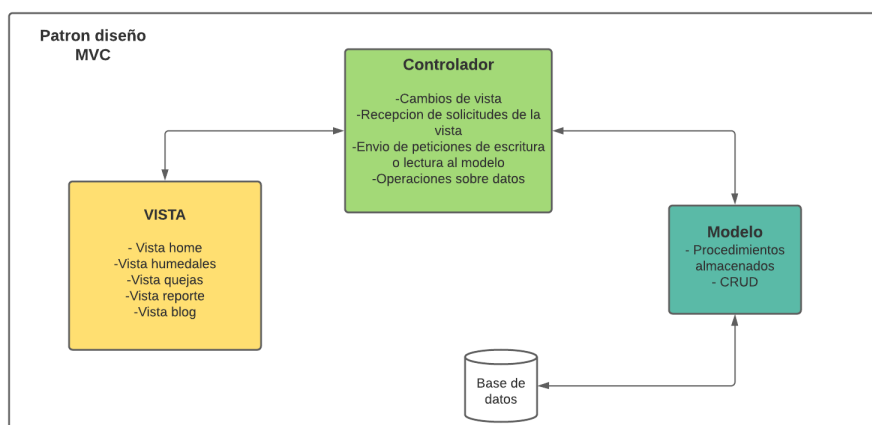


Figura 3.10: Diagrama patrón de diseño Modelo-Vista-Controlador

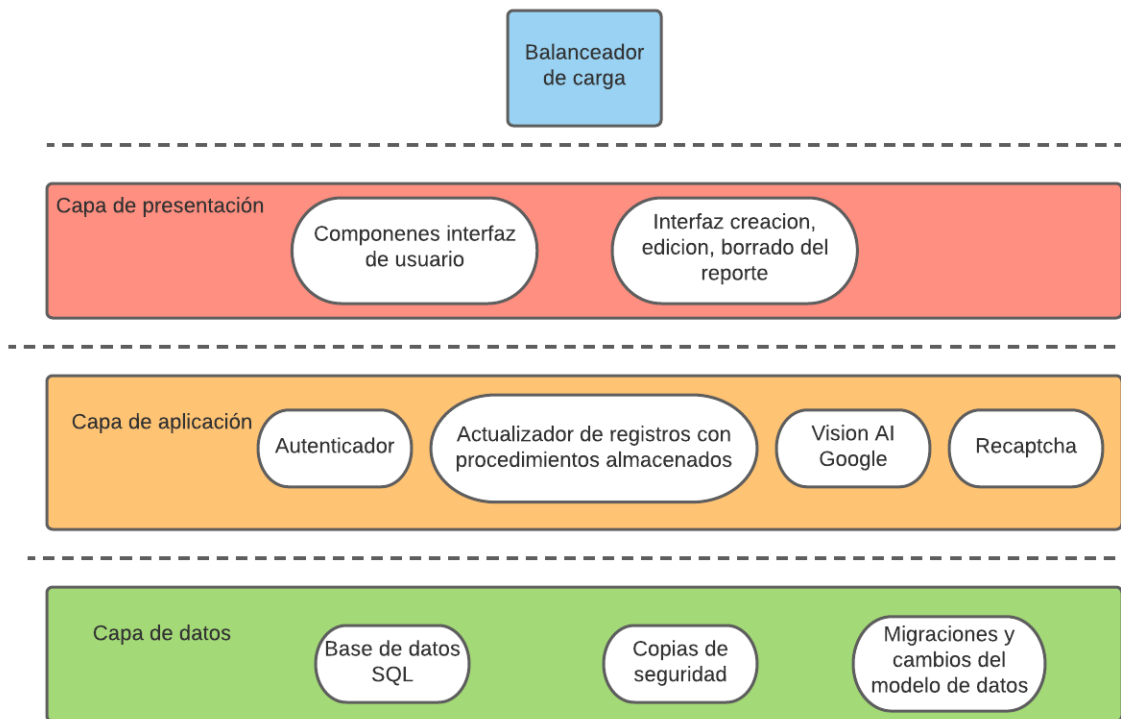


Figura 3.11: Diagrama arquitectura por capas (3 capas)

En el siguiente cuadro se realiza la comparación entre las tres arquitecturas en cuanto a sus mayores fortalezas y debilidades.

Atributo de calidad	Capas	Modelo - Vista - Controlador	Cliente - Servidor
Disponibilidad	Las aplicaciones pueden aprovechar la arquitectura modular de los sistemas habilitados usando componentes que escalan fácilmente lo que incrementa la disponibilidad.	Si se tienen componentes independientes de los demás se puede dar una solución rápida en el caso que ocurra una falla en alguno de los elementos de esta arquitectura	Si un servidor, que es el que transmite los mensajes, falla, el cliente puede buscar reconexión con otro servidor.
Rendimiento	Al tener que interactuar solamente con 3 capas, el cual es un número reducido, no habría problema en el rendimiento con esta arquitectura	Con el menor overhead de comunicación que tiene esta arquitectura y por la independencia de sus elementos se puede afirmar que su rendimiento es alto	El rendimiento puede verse afectado por factores propios de la red y fallas o retardos en las solicitudes y peticiones.

Cuadro 3.4: Comparación arquitecturas.

### 3.0.5.3. Arquitectura seleccionada

Según las deliberaciones realizadas se llegó a la conclusión de que la arquitectura por capas sería la elegida para realizar el proyecto del aplicativo web, aunque con ciertas modificaciones. Las razones por las que se escogió esta arquitectura recaen en la comparación con las otras dos arquitecturas propuestas, donde ésta nos proporciona la mayor flexibilidad en el diseño que las otras propuestas que siguen patrones más marcados. Además de eso, ésta consideramos que presenta el mejor equilibrio entre disponibilidad y rendimiento.

Brinda mejor disponibilidad por el margen de operatividad que brinda a los equipos de mantenimiento, permitiendo que el sistema pueda recuperarse rápidamente mediante el sistema de restauración mediante copias de seguridad. De esta forma los problemas con la base de datos podrían revertirse en lugar de ser resueltos de forma manual. Por otro lado, el rendimiento no está sujeto totalmente a la red sino a la comunicación global de la infraestructura y arquitectura física que se posea para el sistema.

Un aspecto que se decidió cambiar de la arquitectura por capas candidata fue la existencia de la capa de balanceo de carga. Si bien esta trae beneficios con respecto a ambos atributos escogidos, su complejidad en términos económicos a la hora de realizar el despliegue en la nube, nos hace optar por usar solamente un servidor, por lo cual no habría necesidad de usar tal balanceador de carga. A continuación se adjunta el diagrama de la arquitectura por capas con mayor nivel de detalle, al igual que las correcciones y modificaciones realizadas.

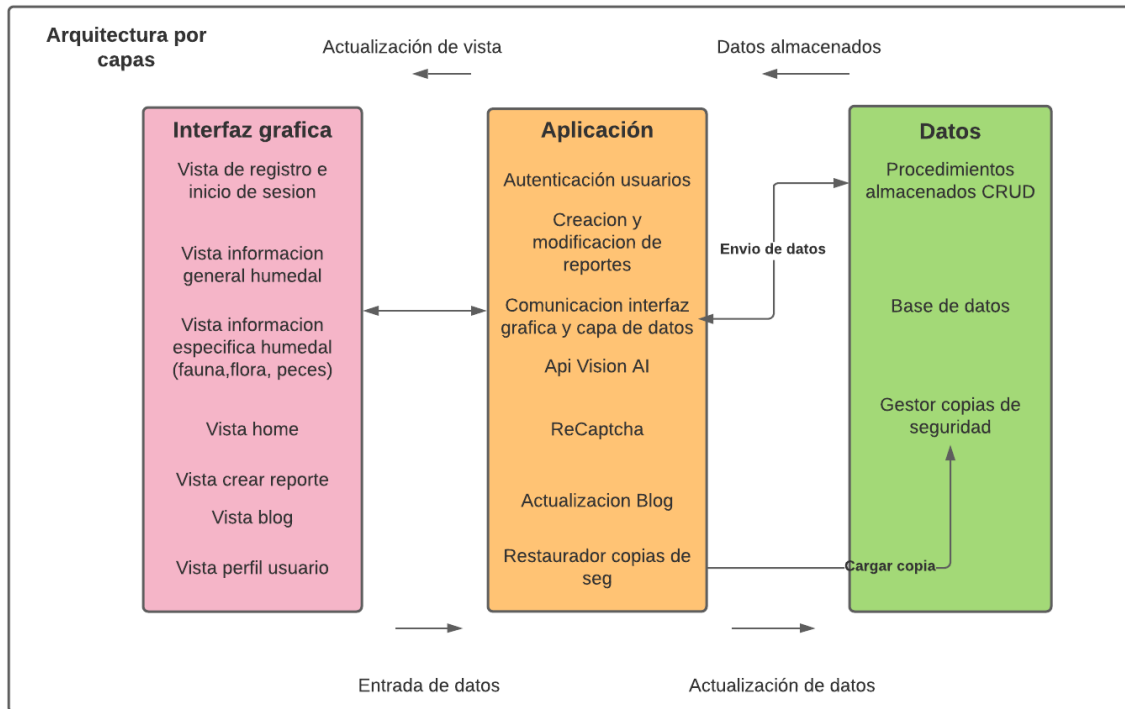


Figura 3.12: Diagrama detallado arquitectura seleccionada

Esta arquitectura cuenta con 3 capas, cada una de ellas cumpliendo un rol en específico dentro del aplicativo web:

- Capa de presentación:** La capa de presentación es la interfaz de usuario y la capa de comunicación de la aplicación, donde el usuario final interactúa con la aplicación. Su objetivo principal es mostrar información y recopilar información del usuario. Este nivel de nivel superior puede ejecutarse en un navegador web, como una aplicación de escritorio o una interfaz gráfica de usuario (GUI), por ejemplo. La capa de presentación web generalmente se desarrollan utilizando HTML, CSS y JavaScript.
- Capa aplicación:** La capa de aplicación, también conocida como nivel lógico o nivel medio, es el corazón de la aplicación. En esta capa, la información recopilada en la capa de presentación se procesa, a veces contra otra información en la capa de datos, utilizando la lógica empresarial, un conjunto específico de reglas comerciales. La capa de aplicación también puede agregar, eliminar o modificar datos en la capa de datos. La capa de aplicación se desarrolla normalmente

con Python, Java, Perl, PHP o Ruby, y se comunica con el nivel de datos mediante llamadas a la API.

- **Capa de datos:** La capa de datos, a veces llamado nivel de base de datos, nivel de acceso a datos o back-end, es donde se almacena y administra la información procesada por la aplicación. Puede ser un sistema de gestión de bases de datos relacionales como PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix o Microsoft SQL Server, o en un servidor de bases de datos NoSQL como Cassandra, CouchDB o MongoDB. [cp-20b]

Para implementar esta arquitectura en términos de código, se decide identificar cada capa mediante una conjunto de carpetas o archivos:

En la figura 3.13 se constata la estructura general del proyecto, donde las carpetas *templates,static* guardan todas las vistas o archivos que hacen parte de la capa de presentación como archivos *html,css,JavaScript*. Por otra parte, el archivo *views.py* cumple el papel de la capa de aplicación . El archivo *db.sqlite3* cumple el papel de la capa de datos, puesto que almacena toda las tablas de datos del aplicativo.

### 3.0.6. Listado de requisitos

Esta sección se dividirá en 2 partes, la primera de ellas tendrá una tabla con los requisitos funcionales los cuales declaran los servicios que proveerá el aplicativo, la segunda parte tendrá los requisitos no funcionales los cuales especifican las propiedades del aplicativo: rendimiento, seguridad, disponibilidad.

Codigo	Nombre	Descripcion
RF-1	Login	El aplicativo debe contar con un login para el acceso de usuarios.
RF-2	Perfil	Los usuarios tendrán un perfil que muestre sus datos personales: identificación, nombre, correo, foto.
RF-3	Galería	Los usuarios tendrán una galería con las imágenes aportadas por él mismo.
RF-4	Datos personales	Los datos personales del usuario serán nombre completo, fecha nacimiento, correo, C.C
RF-5	Rol invitado	Debe existir una opción de invitados para personas que no deseen registrar sus datos.
RF-6	Rol privilegios	El usuario nivel 2 (Personal Dagma) podrán realizar modificaciones al blog.
RF-7	Búsqueda	El usuario nivel 2 (personal Dagma) podra realizar una búsqueda para verificar los elementos subidos por los usuarios ó invitados.
RF-8	Permitir publicación	El usuario nivel 2 (personal Dagma) podrá permitir la publicación de elementos subidos por los usuarios nivel 1 o invitados.
RF-9	Eliminar usuarios	El administrador podrá eliminar perfiles de usuarios nivel 1 y 2.
RF-10	Rol admin	El administrador tendrá todos los permisos de los usuarios nivel 1 y 2.
RF-11	Mod publicación	El administrador y usuario nivel 2 puede modificar la descripción de las publicaciones del blog.
RF-12	Términos y cond	Para que un usuario pueda registrarse en el sistema, este debe aceptar los términos y condiciones.
RF-13	Galería invitado	El usuario que ingrese al aplicativo como invitado no podrá visualizar las imágenes subidas.
RF-14	Formato image	Los formatos para la imagen serán JPEG, JPG, PNG.
RF-15	Descripción image	Las imágenes podrán ser acompañas por una descripción.
RF-16	Elegir humedal	El usuario podrá elegir a que humedal pertenece la imagen.
RF-17	Acceder aplica	Los usuarios podrán acceder al aplicativo ingresando a un navegador con la dirección URL.

Código	Nombre	Descripción
RF-18	Info humedal	El aplicativo web contará con la información de 3 humedales.
RF-19	Pestaña humedal	Existirá una sección diferente para cada humedal (pestaña).
RF-20	Información humedal	Cada humedal contará con su nombre, ubicación, mapa, información adicional.
RF-21	Info específica humedal	Cada humedal contará con su sección específica por fauna, flora y fauna acuática.
RF-22	Footer	Cada vista debe tener en su footer enlaces a la alcaldía, teléfonos y derechos reservados
RF-23	Nav-bar	Cada vista debe tener en su nav-bar, redirecciones de las vistas relevantes según donde se encuentre (Backoffice, Frontoffice).
RF-24	Contacto	Debe existir una sección de contacto en el cual estén las direcciones de contacto de entidades ambientales (Dagma, CVC) y la alcaldía, teléfonos, correo.
RF-25	Eliminar public	El usuario lv 2 podrá eliminar una publicación realizada en el blog.
RF-26	Admin imágenes	El administrador podrá eliminar cualquier imagen subida por un usuario.
RF-27	Quejas	El aplicativo debe contar con una sección para recibir las quejas de los usuarios.
RF-28	Planes manejo	Debe existir una sección para los planes de manejo ambiental de los 3 humedales.
RF-29	Recuperar contraseña	El usuario podrá recuperar su contraseña en cualquier momento.
RF-30	Zoom imágenes	Se podrá aplicar zoom a las imágenes de las galerías de fauna, fauna acuática y flora.
RF-31	Aporte	Para poder subir un aporte se deben llenar los campos de título, descripción, tipo aporte, humedal y la imagen.
RF-32	Contenido Explícito	No se permitirán imágenes con contenido explícito.

Código	Nombre	Descripción
RNF-1	Usuarios Concurrentes	El aplicativo debe soportar 20 usuarios concurrentes.
RNF-2	Tiempo respuesta	Cuando un usuario hace clic sobre un humedal, el sistema no debe tardar más de 3 segundos en arrojar resultados.
RNF-3	Clicks	El usuario no deberá dar más de 4-5 clics para realizar un aporte (imagen) al aplicativo.
RNF-4	Disponibilidad	La base de datos del aplicativo deberá estar en funcionamiento el 90 % del tiempo.
RNF-5	Tiempo Guardado	Cuando un usuario decide enviar un aporte, el aplicativo no debe tardar más de 3 segundos en almacenar los cambios en el sistema.
RNF-6	Manual	El aplicativo debe contar con manuales de usuario estructurados adecuadamente.
RNF-7	Seguridad	El aplicativo debe encriptar las contraseñas de los usuarios

Cuadro 3.5: Listado requisitos no funcionales.

### 3.0.7. Apis externas:

Las apis hoy en día son muy utilizadas para cualquier cantidad de proyectos, dado que poseen ciertas funcionalidades especiales que se pueden adaptar a una gran cantidad de servicios, de forma económica y sencilla, agregándole gran valor a estos. Las apis permiten la comunicación entre dos aplicaciones de software a través de un conjunto de reglas. Para este aplicativo web usamos las siguientes apis:

- **ReCaptcha:** Usamos esta Api que provee Google para prevenir posibles ataques de bots los cuales podrían crear múltiples perfiles y subir gran cantidad de reportes de forma automática, lo cual podría tumbar el servidor o colocarlo muy lento. Por ello, con esta autenticación se puede evitar todo esto de manera fácil y gratis. De acuerdo con Google, reCAPTCHA es un servicio gratuito que protege su sitio del spam y el abuso. Utiliza técnicas avanzadas de análisis de riesgos para diferenciar a los humanos de los robots. [cp-21d]
- **Google Maps:** Usamos esta Api de Google para colocar el mapa embebido de cada humedal, a fin de que el usuario pueda ubicar cada uno de ellos de una forma interactiva y fácil de usar, dada la popularidad de esta Api.
- **Google Translate:** Usamos esta Api para traducir las etiquetas que nos retornaba la api Vision AI (capitulo 3) de ingles a español.

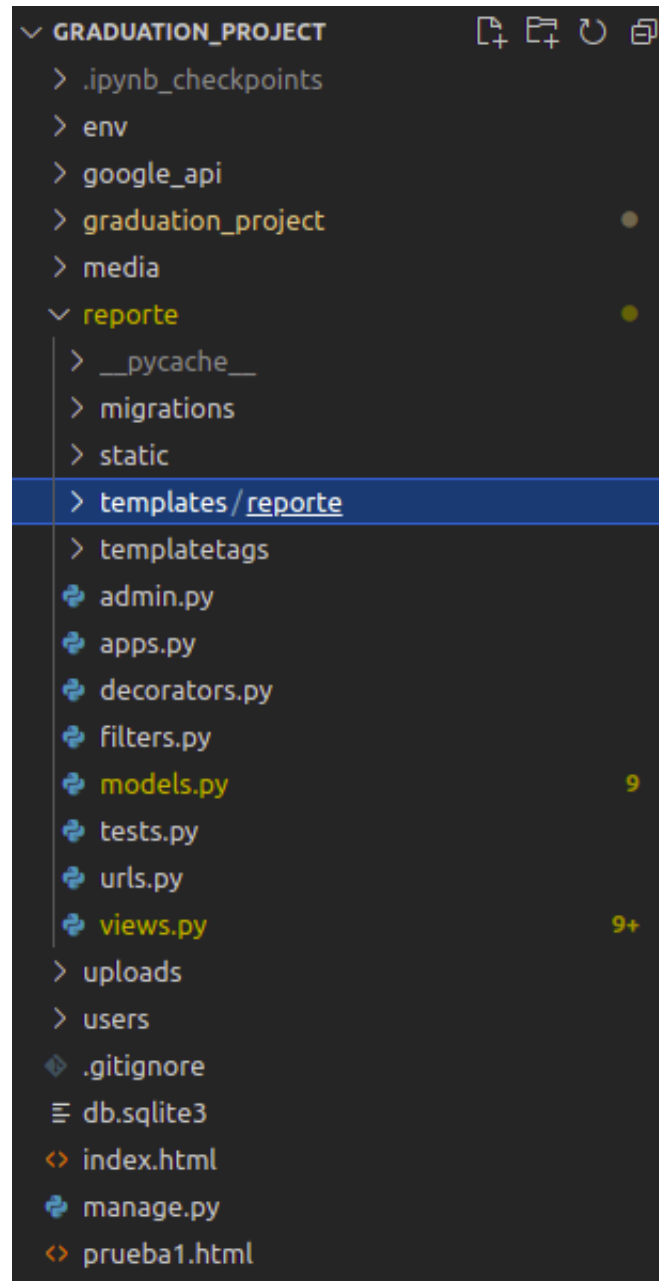


Figura 3.13: : Implementación arquitectura



# Reconocimiento de imágenes

---

## 4.1. Qué es el reconocimiento de imágenes

El reconocimiento de imágenes es la tarea de identificar objetos en alguna imagen y reconocer a que categoría pertenecen. Comúnmente, también se suele llamar reconocimiento de fotos. Cuando usualmente se ven objetos en alguna situación, como por ejemplo, en la calle mientras se camina o en el plato en un restaurante, automáticamente las personas pueden identificar que objetos son asociándolos a alguna definición, llámense carros, motos, buses, edificios, cucharas, tenedores, etc. Para las personas es una tarea muy fácil que se realiza sin darse cuenta y que tampoco requiere de plena atención. No obstante, el realizar esta misma tarea representa un gran reto para las computadoras. En la actualidad el reconocimiento de imágenes se realiza mediante el uso de inteligencia artificial y representa un problema de investigación desde hace varias décadas en el campo de la visión de la computadora (*Computer Vision*). Para ello, se utilizan varios métodos, sin embargo, se comparte la misma meta de clasificar objetos detectados en diferentes categorías.

En años pasados se ha demostrado que los métodos de aprendizaje profundo superan a técnicas anteriores de aprendizaje automático y el campo de la visión por computadora ha sido uno de los casos mas destacados. Por lo que estos métodos han logrado grandes avances expresados en mejores resultados en rendimiento (fotogramas calculados por segundo/FPS) siendo así mas flexibles en cuanto a la cantidad de información que pueden procesar por segundo los modelos de inteligencia artificial y algoritmos de aprendizaje profundo. [cp-18]

### 4.1.1. Tareas del reconocimiento de imágenes

El reconocimiento de imágenes se puede realizar con diferentes grados de precisión, de acuerdo al tipo de información o concepto requerido. De hecho, un modelo o algoritmo es capaz de detectar un elemento específico, del mismo modo que puede simplemente asignar una imagen a una categoría grande. A continuación se listarán las tareas que pueden realizar los algoritmos de reconocimiento de imágenes.

- **Clasificación:** Es la identificación de la “clase”, en otras palabras, la categoría a la que pertenece una imagen. Una imagen solo puede tener una clase.
- **Etiquetado:** También es una tarea de clasificación pero con un grado mayor de precisión. Puede reconocer la presencia de varios *conceptos* u *objetos* dentro de una imagen. Por lo tanto,

se pueden asignar una o más etiquetas a una imagen en particular.

- **Detección:** Esto puede ser necesario cuando se desea ubicar un objeto en una imagen. Una vez que se encuentra el objeto, se coloca un cuadro delimitador alrededor del objeto en cuestión.
- **Segmentación:** La segmentación de imágenes es la división de una imagen en regiones o categorías, que corresponden a diferentes objetos o partes de objetos. Cada píxel de una imagen se asigna a uno de un número de estas categorías. [cp-]

#### 4.1.2. Cómo funciona el reconocimiento de imágenes

Normalmente la tarea de reconocimiento de imágenes envuelve la creación de una red neuronal que procese individualmente los píxeles de una imagen. Estas redes se alimentan previamente con una gran cantidad de imágenes pre-etiquetadas por humanos, esto con el propósito de *enseñarle* a reconocer imágenes similares. Para ello se pasa por un proceso de filtrado de imágenes, segmentación, extracción de características y clasificación basada en reglas. El proceso desde la creación de la red neuronal hasta la clasificación de una imagen se puede sintetizar en 3 pasos comunes de algoritmos de aprendizaje. A continuación se listan:

Los modelos de reconocimiento de imágenes requieren datos de entrenamiento tales como: vídeos, fotos e imágenes. Las redes neuronales necesitan esas imágenes de entrenamiento de un conjunto de datos adquirido para crear percepciones de cómo funcionan ciertas clases. Por ejemplo, una imagen de un gato debe ser etiquetada como *Gato* o algo que se pueda entender.

Luego de ello, estas imágenes deben introducirse en una red neuronal y luego entrenarse en ellas. Por lo general, para las tareas relacionadas con las imágenes, se utilizan redes neuronales convolucionales (*convolutional neural networks CNNs*). Estas redes constan de capas convolucionales y capas de agrupación, además de capas Multiperceptron (MLP). El funcionamiento de estas redes se explicará en breve. [cp-20d]

El modelo entrenado debe probarse con imágenes que no forman parte del conjunto de datos de entrenamiento. Se realiza para determinar la usabilidad, el rendimiento y la precisión del modelo. Es por esto que, alrededor del 80 % del conjunto de datos de imágenes se usa para el entrenamiento del modelo, mientras que los datos restantes se reservan para la prueba del modelo. El rendimiento del modelo se mide en función de un conjunto de parámetros que indican el porcentaje de confianza de precisión por imagen de prueba, identificaciones incorrectas y más.

## 4.2. Convolutional neural networks

Las redes neuronales convolucionales (CNNs) son una clase de redes neuronales de aprendizaje profundo. Las CNNs representan un gran avance en el reconocimiento de imágenes, por lo que son utilizadas con mayor frecuencia para analizar imágenes y para la clasificación de estas. Sin darse cuenta, las CNNs se pueden encontrar en muchos aspectos de la vida tecnológica diaria, por ejemplo, cuando alguien sube una foto a *Facebook* con algunos de sus amigos, automáticamente esta red

social dice si desea etiquetarlos, de manera que este ya reconoció los rostros encontrados en la foto y los clasificó como alguno de sus amigos agregados en esta red social. Otro ejemplo muy relevante en este campo, son los autos autónomos, los cuales mientras son conducidos van identificando en tiempo real: señales de tránsito, semáforos, personas, cambios de carril, entre otros. Usando CNNs y otros algoritmos. También, en los campos médicos y de seguridad suelen ser usadas las CNNs. La clasificación de imágenes es el proceso de tomar una entrada (como una imagen) y generar una clase (como 'rana') o una probabilidad de que la entrada sea una clase en particular ("hay un 85 % de probabilidad de que esta entrada sea una rana"). Una persona puede mirar una imagen y darse cuenta que esta viendo un tigre acostado. Pero ¿Cómo logra una computadora aprender esto y tener un nivel de certeza alto? Gracias al uso de una red neuronal convolucional.

Según Rikiya, Mizuho Nishio y Togashi una CNN esta compuesta por: *múltiples bloques de construcción, como capas de convolución, capas de agrupación y capas completamente conectadas, y está diseñada para aprender de forma automática y adaptativa las jerarquías espaciales de características a través de un algoritmo de retro-propagación.* [Yam18] De acuerdo a esta definición y otras investigadas se saca una lista de los elementos de una CNN habitual:

- Capas convolucionales (Convolutional layers)
- Capas ReLU (ReLU layers)
- Agrupación de capas (Pooling layers)
- Una capa completamente conectada (a Fully connected layer)

Las CNN se inspiran en procesos biológicos. Se basan en una interesante investigación realizada por Hubel y Wiesel en los años 60 sobre la visión en gatos y monos [WIE61], en la cual insertaron un micro-electrodo en la corteza visual primaria de un gato parcialmente anestesiado para que no pueda moverse y posteriormente mostraron varias imágenes de líneas en diferentes ángulos. A través del micro-electrodo, encontraron que algunas neuronas se activaban de manera veloz al observar las líneas en ángulos específicos, mientras que otras neuronas respondían mejor a las líneas en diferentes ángulos. Algunas de estas neuronas respondieron a patrones claros y oscuros de manera diferente, mientras que otras neuronas respondieron para detectar movimiento en cierta dirección.

El patrón de conectividad en una CNN proviene de su investigación sobre la organización de la corteza visual. En el ojo de un mamífero, las neuronas individuales responden a los estímulos visuales solo en el campo receptivo, que es una región restringida. Los campos receptivos de diferentes regiones se superponen parcialmente de modo que se cubre todo el campo de visión. De esta manera es como funciona una CNN.

### 4.2.1. Capa convolucional

El propósito principal del paso de convolución es extraer características de la imagen de entrada. La capa convolucional es siempre el primer paso en una CNN. Dada una imagen de entrada, un

detector de características y un mapa de características. Se procede a dividir la imagen en una serie de mosaicos de algún tamaño, por ejemplo 3x3 píxeles (Ver figura 4.1), a cada mosaico se le aplica el filtro. El filtro es una matriz de números (también llamados pesos o parámetros). La profundidad del filtro, tiene que ser igual al de la profundidad de la imagen, por lo que si es una imagen a color, la profundidad del filtro sera 3. Dado proceso se hace mediante la multiplicación de matrices.

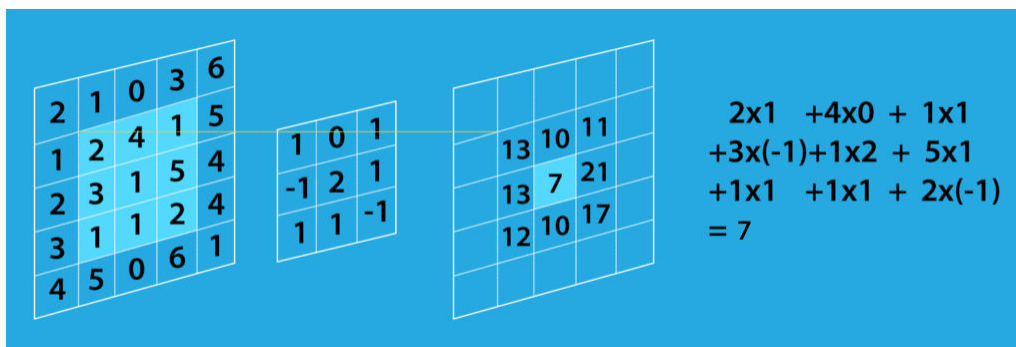


Figura 4.1: Aplicación filtro - Capa Convolutiva

La matriz resultante tiene como nombre mapa de características o mapa de activación. Se crean muchos mapas de características para de esta manera obtener la primera capa convolutiva. Permite identificar muchas características diferentes que la CNN puede usar en el proceso de aprendizaje.

#### 4.2.2. Relu Capa

En la capa ReLU (unidad lineal rectificadora o rectificadora) se aplica una función de activación en los mapas de características para aumentar la no linealidad en la red. Esto es debido a que las imágenes no son lineales, por lo que se eliminan los valores negativos de un mapa de características, colocándolos a cero. La convolución es una operación lineal con cosas como la multiplicación y suma de matrices por elementos. Se ha convertido en la función de activación predeterminada para muchos tipos de redes neuronales porque un modelo que la usa es más fácil de entrenar y, a menudo, logra un mejor rendimiento [cp-19b].

#### 4.2.3. Capa de agrupación

La capa de agrupación juega un papel muy importante en una CNN, dado que no se desea solamente poder identificar una característica específica, en un color exacto, en una ubicación exacta. Por ejemplo, un gato gris que este acostado en la parte izquierda de una imagen; no queremos que solamente se puedan identificar gatos grises acostados en la parte izquierda de una imagen, si no todos los gatos de diferentes colores, en posiciones distintas y en cualquier ubicación de la imagen. Un ejemplo de ello es la figura 4.2 la cual muestra un carro desde distintos puntos. Básicamente, se quiere que la CNN tenga *flexibilidad*. La capa de agrupación reduce progresivamente el tamaño

de la representación de entrada, permitiendo así detectar objetos en una imagen sin importar dónde se encuentren. Por otra parte, la agrupación ayuda a reducir la cantidad de parámetros requeridos y la cantidad de cálculo requerido. También ayuda a controlar el sobre-ajuste (overfitting).



Figura 4.2: Capa agrupación

Para realizar dicho proceso existen 3 formas, combinación promedio (average pooling), combinación de la suma (sum pooling) y agrupación máxima (Max pooling). Esta última, la cual es la más usada puesto que en la práctica tiende a ser mejor que las dos anteriores, funciona de la siguiente manera; la agrupación máxima se trata de obtener el valor máximo en cada punto de la imagen (Ver figura 4.3). Realizar este proceso elimina cerca del 75 % de información que no es característica. Al tomar el máximo valor de los píxeles, se contabiliza la distorsión. Si la función gira un poco hacia la izquierda o hacia la derecha o hacia otra dirección, la función agrupada será la misma, por lo que se reduce el tamaño y los parámetros. Esto es muy útil porque significa que el modelo no se ajustará demasiado a una información.

#### 4.2.4. Capa completamente conectada

De acuerdo a Borner A. [cp-19a] en este paso, se agrega una red neuronal a la CNN. El propósito principal de esta nueva red neuronal artificial es combinar las características en más atributos. Estos predecirán las clases con mayor precisión, combinando características y atributos que pueden predecir mejor las clases.

En este paso, el error se calcula y luego se propaga hacia atrás. Los detectores de pesos y características se ajustan para ayudar a optimizar el rendimiento del modelo. Luego, el proceso se repite una y otra vez, de esta manera es como la red se entrena con los datos.

El funcionamiento de las neuronas de salida de esta red, es la siguiente; por ejemplo, si se tienen

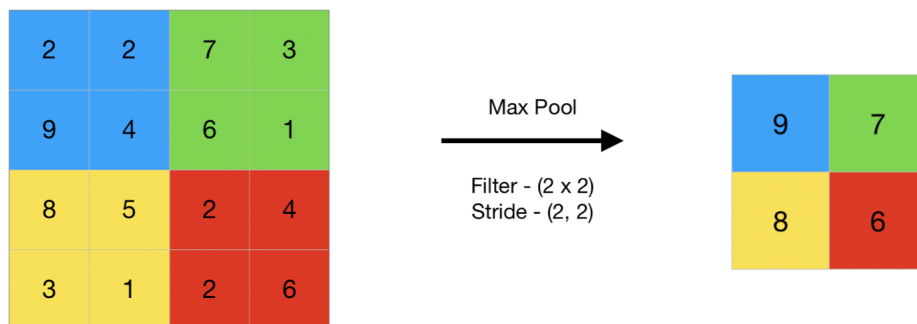


Figura 4.3: Capa agrupación-Max pool

dos clases de salida, una para un conejo y otra para un perro. Una neurona que diga “0” no está segura que la característica a la que pertenece sea la de un conejo, mientras una neurona que diga “1”, tiene la certeza suficiente para decir que esa característica pertenece a la de un conejo. En la capa completamente conectada, las neuronas leerán valores entre 0 y 1. Por lo que un valor de 0,80 significaría un 80% de certeza. Dicho esto, las neuronas de un conejo o perro están seguras cuando se identifica una o varias características, de manera que se sabe que la imagen pertenece a un conejo o perro. Por lo tanto, cada vez que se activan ciertas neuronas, la red sabe a que animal pertenece. Las neuronas del perro aprenden que por ejemplo cuando se activan ciertas neuronas como, la neurona de la “nariz grande”, la neurona del “oído flexible”, la neurona de “cola mediana”, la imagen pertenece a la de un perro. Por otro lado, la neurona del conejo aprende que cuando se activan neuronas de “Orejas largas”, “patas gruesas”, “bigotes”, la imagen pertenece a un conejo. La figura 4.4 muestra todo el proceso resumido de la CNN desde la primer capa convolucional hasta el resultado de clasificación de la imagen de un automóvil.

### 4.3. Por qué el reconocimiento de imágenes

Se decidió usar el reconocimiento de imágenes para poder identificar patrones en las imágenes subidas por los usuarios del aplicativo a fin de poder diferenciar entre imágenes que aporten valor al cuidado de los humedales y aquellas que no lo hacen. Por ejemplo, se considera que imágenes que pueden aportar valor serían las relacionadas con animales, naturaleza, contaminación, personas y cuerpos de agua. Por otro lado, imágenes relacionadas con contenido explícito tal como, pornografía, racismo y violencia no aportarían ningún valor al cuidado de los humedales, por lo cual este tipo de contenido se puede bloquear. En este último es donde entra a jugar un papel muy importante el reconocimiento de imágenes, pues se desea bloquear este tipo de contenido con el objetivo de solo guardar información con la cual la entidad de protección ambiental como el DAGMA pueda usar.

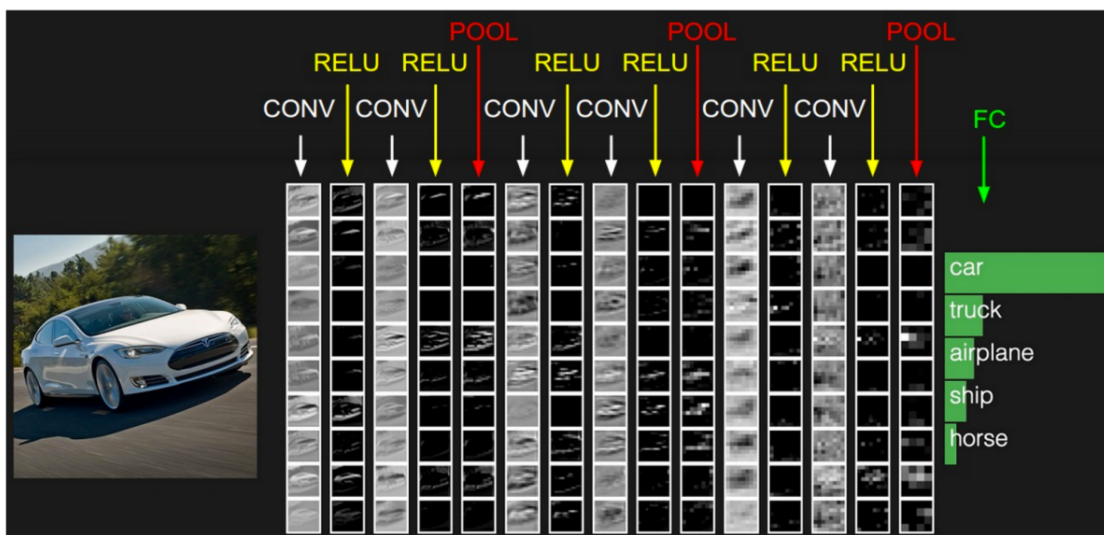


Figura 4.4: Resultado CNN

## 4.4. Cómo construir una CNN

Para construir una CNN habría que implementar cada uno de los elementos descritos anteriormente, además de poseer alguna base de datos con miles de imágenes para poder entrenar esta CNN, lo que podría ser muy costoso y conllevaría mucho tiempo. Dicho esto, recientemente se han desarrollado varias APIs que permiten que organizaciones o empresas obtengan la información que requieren en cuanto al servicio de reconocimiento de imágenes sin la necesidad de aprendizaje automático interno o experiencia en el campo de visión por computadora.

## 4.5. APIs reconocimiento de imágenes

### 4.5.1. Google Cloud Vision

De acuerdo a Google: La API de Vision ofrece modelos de aprendizaje automático preentrenados y muy potentes a través de las APIs REST y RPC. Puede asignar etiquetas a imágenes y clasificarlas de acuerdo categorías previamente definidas. Detecta objetos y caras, lee texto impreso y manuscrito, y consigue metadatos que pueden ser de utilidad para algún catalogo de imágenes.[cp-20c]

### 4.5.2. IBM Watson Visual Recognition

De acuerdo a IBM: El servicio IBM Watson Visual Recognition utiliza algoritmos de aprendizaje profundo para identificar escenas y objetos en imágenes que carga en el servicio. Puede crear y entrenar un clasificador personalizado para identificar materias que se adapten a sus necesidades.[cp-21c]

### 4.5.3. Computer Vision Microsoft Azure

Microsoft ofrece su servicio de Computer vision almacenado en su nube de Azure, características como impulsar la detectabilidad del contenido, automatizar la extracción de texto, analizar contenido de vídeo en tiempo real, procesamiento de datos visuales para etiquetar el contenido con objetos y conceptos, extraer texto, generar descripciones de imágenes, moderar contenido y reconocer el movimiento de personas en espacios físicos; hacen parte de su amplio catalogo de servicios. [cp-21a]

### 4.5.4. Elección API

Para la elección de la API, se decidió utilizar un portal web [www.g2.com](http://www.g2.com) (Ver figura 4.5) especializado en reseñas de software y servicios comerciales, para ello se comparó cada una de las APIs. Este portal las comparó de acuerdo a las siguientes 7 características en una escala de 0 a 10 : cumple con los requisitos, facilidad de uso, facilidad de administración, facilidad de instalación, facilidad de administración, calidad de soporte, facilidad para hacer negocios con la API y dirección del producto (% positiva). El puntaje en cada una de las características es basado en la calificación realizada por múltiples usuarios con experiencia verificada en la industria de la computación, software, TI, etc. La API que tuvo mejor calificación, ganando en 4 de 7 características fue Google Cloud Vision, destacándose en facilidad de uso, facilidad de instalación, calidad de soporte y facilidad para hacer negocios con la API.

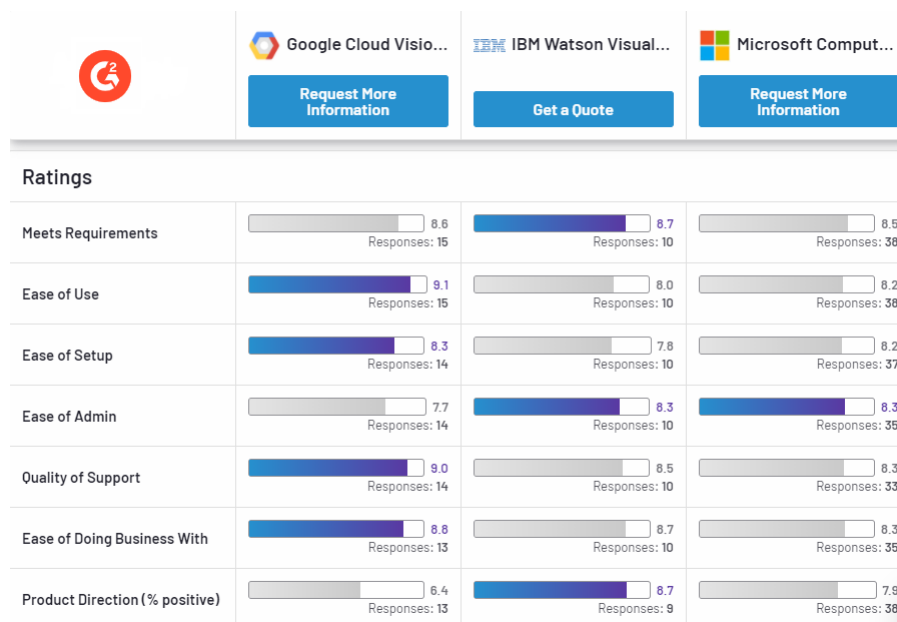


Figura 4.5: Comparación Apis

## 4.6. Google Cloud Vision

Con esta API se logra sacar etiquetas a cada imagen subida por un usuario; obteniendo así, las características más importantes de las imágenes. Estas se publican en cada aporte realizado, de manera que el mismo usuario se puede dar cuenta lo que se reconoció en él. A continuación algunas muestras.

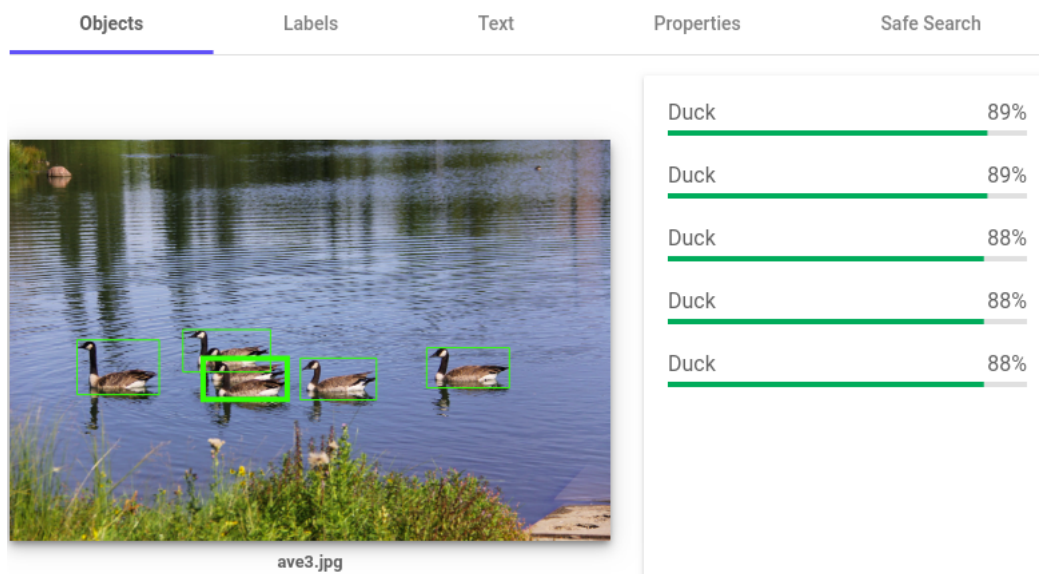


Figura 4.6: Objetos reconocidos en la imagen.

En la figura 4.6 se puede observar como la API reconoce a 5 patos, cada uno con una probabilidad de certeza de mas de 87%. Las características extraídas de la imagen se ordenan de mayor a menor según su nivel de certeza, tal como lo muestra la figura 4.7, donde se puede ver cómo se reconoce agua, pájaro, ecoregión, patos,etc; también, se observa en la sección de búsqueda de contenido explicito (Ver figura 4.8) como la imagen **no** se cataloga con este tipo de contenido puesto que ninguno de los indicadores llega a *Likely*, en español *probable*, por lo tanto esta imagen puede ser publicada en el aplicativo. En contraste, la imagen de la figura 4.9 muestra contenido explicito que la api cataloga con dos indicadores *Violence* y *Medical* con *Likely* por lo que esta imagen se bloquea y no se guarda en el aplicativo. Una vez la imagen ha sido procesada por la API y carece de contenido explicito se procede a guardar la imagen y publicarse junto con el otro contenido del aporte (Ver figura 4.10).

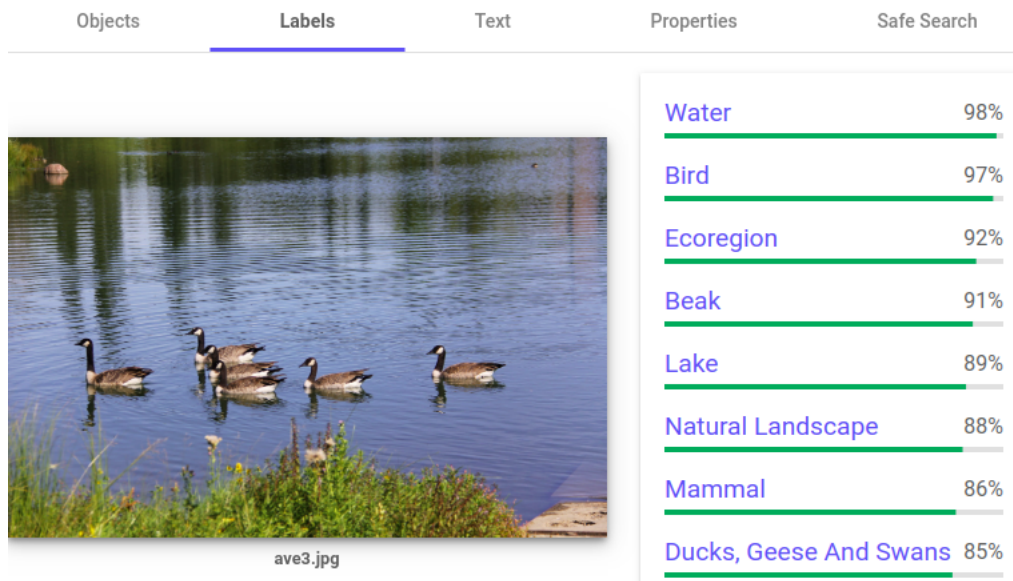


Figura 4.7: Etiquetas extraídas de la imagen.

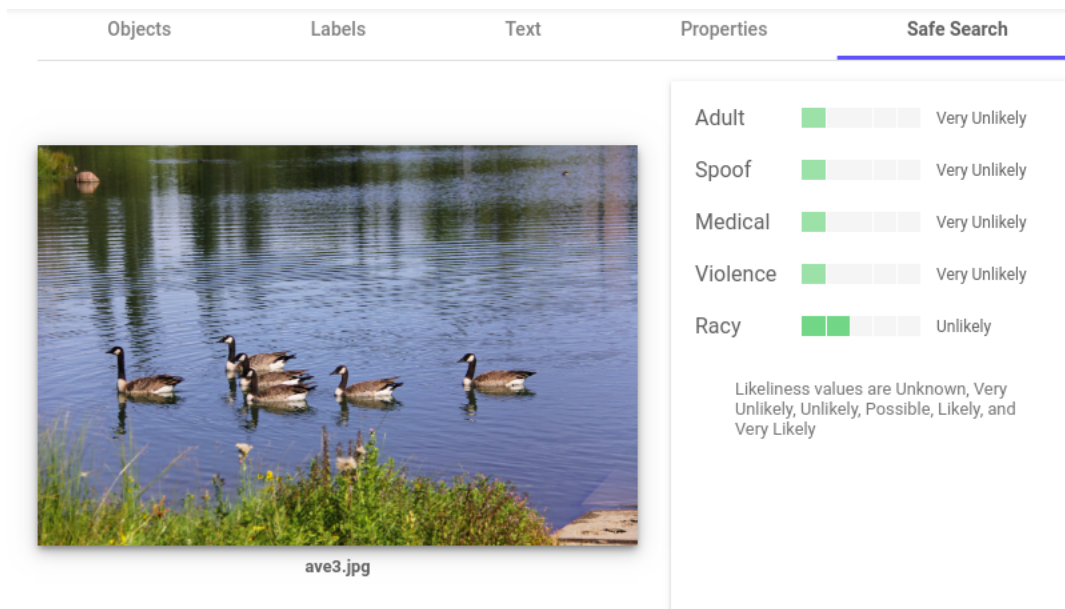


Figura 4.8: Detección de contenido explícito.

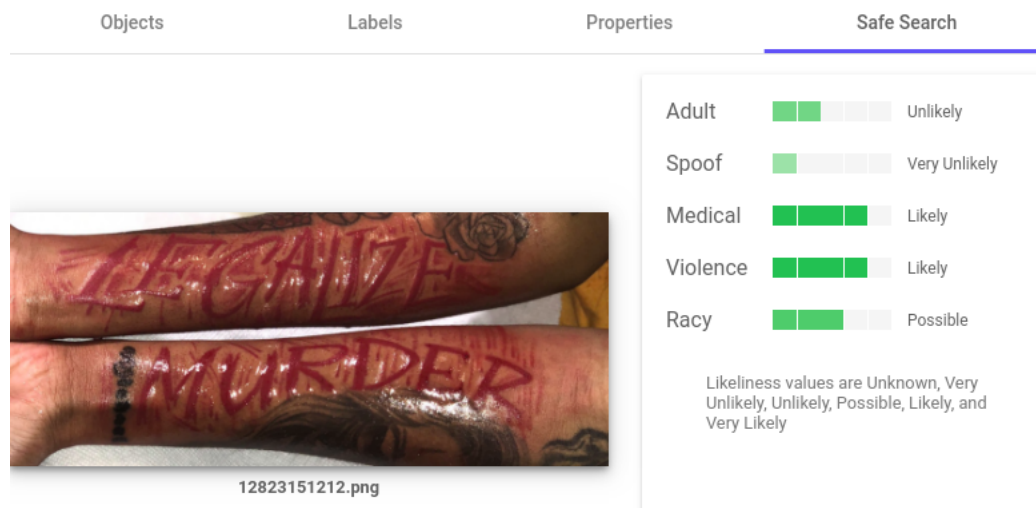


Figura 4.9: Detección contenido explícito.

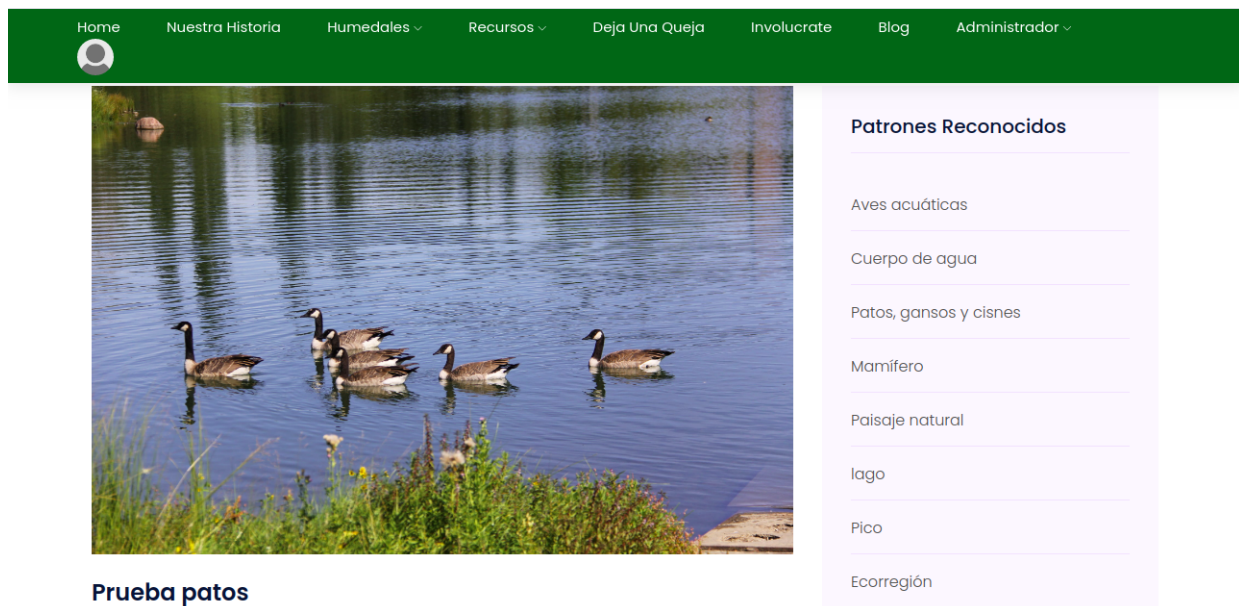


Figura 4.10: Vista aporte en el aplicativo.



## CAPÍTULO 5

# Pruebas

---

Este capítulo contiene el plan de pruebas para el aplicativo, el cual se realiza con el fin de pulir y refinar las funcionalidades implementadas en el proyecto, al igual que asegurar la calidad y mantenibilidad del código.

El aplicativo permite a cualquier ciudadano realizar un aporte que consiste en un material fotográfico acompañado de ciertos campos donde se deje evidencia de algo que este sucediendo en el humedal relacionado por ejemplo con agentes contaminantes, avistamiento de aves, actividades recreativas u otro tipo de situación. El personal de la entidad ambiental, en este caso (DAGMA), puede entrar desde el rol de administrador y gestionar cada uno de los aportes hechos por los ciudadanos a fin de encontrar situaciones en las cuales el humedal pueda estar siendo vulnerado, a su vez también puede encontrar aportes que generen valor para la entidad como los aportes relacionados con animales. Por otra parte, el administrador también tiene la opción de crear nuevos humedales, añadir imágenes a la galería de especies de cada uno de estos, gestionar usuarios y mensajes dejados por los ciudadanos.

Todas las funcionalidades que requieren intervención del usuario (ciudadano/administrador) fueron probadas. En el diseño de pruebas se completaron 78 casos de prueba funcionales y no funcionales para ser ejecutados.

Los indicadores de calidad obtenidos fueron los siguientes:

- Cobertura = 0.97
- Productividad = 0.21
- Correctitud = 0.92

Para más detalle de los indicadores resultantes consultar el *Anexo 5 - Indicadores de Calidad*.

### 5.1. Funcionalidades a probar

Todas las funcionalidades están desarrolladas como componentes individuales. Por lo tanto, se ha decidido organizarlas en 3 bloques; el primero corresponde a funcionalidades de ingreso y salida del sistema; el segundo corresponde a funcionalidades correspondientes a la creación y vista de los aportes y el uso del software por parte del ciudadano; y por último, el tercer bloque esta compuesto

por las funcionalidades propias del administrador, que se encarga del manejo de los aportes y de acciones propias de administrador de sistema como creación de humedales y gestionar usuarios.

#### 5.1.1. Bloque 1

- **1.1 Registrar:** Esta funcionalidad permite al usuario (ciudadano) tener un registro de los aportes realizados por él, además de tener un perfil asociado con sus datos.
- **1.2 Iniciar sesión:** Esta funcionalidad permite al usuario (ciudadano y administrador) acceder al aplicativo y visualizar sus aportes, por otro lado si accede y tiene rol de administrador tendrá las opciones de gestión.
- **1.3 Cerrar sesión:** Permite al usuario (sea ciudadano o administrador) regresar a la página de inicio.

#### 5.1.2. Bloque 2

- **2.1 Crear aporte:** Crea un nuevo aporte a partir de unos datos entregados, y es almacenado en la base de datos.
- **2.2 Crear mensaje contacto:** Crear un nuevo mensaje a partir de unos datos entregados, y es almacenado en la base de datos.
- **2.3 Visualizar aporte:** Permite mostrar el contenido de un aporte en pantalla de forma organizada.
- **2.4 Visualizar perfil:** Permite mostrar los datos suministrados por el usuario a la hora de realizar el registro.
- **2.5 Subir imagen de perfil:** Permite subir una imagen para colocar de perfil de usuario.

#### 5.1.3. Bloque 3

- **3.1 Gestionar aportes:** Permite eliminar, modificar y fijar(publicar en el blog) los aportes.
- **3.2 Añadir usuario:** Permite añadir un nuevo usuario ya sea como rol de ciudadano o administrador.
- **3.3 Eliminar usuario:** Permite eliminar un usuario existente.
- **3.4 Gestionar blog:** Permite visualizar cuales aportes están publicados en el blog y organizarlos según su importancia.
- **3.5 Gestionar usuarios:** Permite visualizar los usuarios registrados y el número de aportes publicados por cada uno.

- **3.6 Administrar base de datos:** Permite realizar tareas de administración manual en la base de datos mediante la consola de Django.
- **3.7 Crear humedal:** Permite crear un humedal suministrando los datos de este.

## 5.2. Estrategia de pruebas

En este plan de pruebas se contemplan pruebas tanto estáticas como dinámicas. En las pruebas estáticas se analizan características sobre cómo está escrito el código usando el servicio de Sonar-Cloud.

En las pruebas dinámicas se tienen en cuenta únicamente pruebas en el nivel de componente y de integración, y se abarcan los siguientes tipos de prueba:

- **Pruebas de caja blanca:** Donde se utilizan las pruebas y técnicas de cobertura.
- **Pruebas de caja negra:** Donde utilizan técnicas variadas para obtener casos de prueba de forma sistemática y donde se automatizan la mayoría de casos de prueba y algunos se realizaron de forma manual.
- **Pruebas de desempeño:** Donde se realizan las respectivas pruebas de carga y de estrés.

## 5.3. Análisis estático

Se usó la herramienta **sonarcloud.io** para realizar el análisis estático del software. En la primera revisión se obtuvieron 533 bugs, más de 1500 “Code Smells”, 98 “Security review” o código sensible a la seguridad, 21.6 % de código duplicado y ninguna vulnerabilidad; luego de realizar múltiples ajustes siguiendo las recomendaciones que proveyó esta herramienta, en la última revisión se obtuvieron 305 bugs, 307 “Code Smells”, 2 “Security review”, 1 % de código duplicado y ninguna vulnerabilidad. Para más detalle consultar el *Anexo No. 2*.

## 5.4. Pruebas de caja blanca

El proceso de pruebas de caja blanca se llevó a cabo en el nivel de componente. Se exploraron las coberturas sentencia y decisión con las herramientas **Pytest** y **Coverage**, se alcanzó con 38 pruebas 92 % de cobertura de sentencia y 89 % de cobertura de decisión. Para más detalle consultar el *Anexo 3*.

## 5.5. Pruebas de caja negra

Las pruebas de caja negra fueron diseñadas a nivel de integración de los módulos del sistema, ya que se prueba la concordancia entre el funcionamiento de la página desde el frontend (desde la

perspectiva del usuario) con la base de datos a tiempo real de MySQL. Para ello se usan pruebas manuales que se construyen gracias al uso de la técnica de partición de equivalencia.

## 5.6. Pruebas de carga

Se realizaron cargas que permitieron verificar el desempeño de la aplicación en condiciones normales; específicamente para las funcionalidades del login de usuarios, consulta de aportes y creación de aportes, donde se verificó que el sistema respondía bien ante un número de solicitudes normales.

## 5.7. Pruebas de estrés

Se realizaron pruebas de carga para detectar cuellos de botella, sobre todo en el proceso más crítico. Tanto la prueba de desempeño como la prueba de estrés se hicieron de manera automatizada.

## 5.8. Herramientas a usar en las pruebas no funcionales

- **Jmeter:** Debido a que el software es un aplicativo web, se decidió que utilizar esta herramienta para las pruebas de carga y estrés. Dado que proporciona distintas herramientas para poder visualizar e interpretar de buena manera los resultados de cada una de las pruebas.
- **Blazemeter:** Para capturar la secuencia de ejecución del aplicativo web y llevarlas a Jmeter en archivos con extensión jmx.

Para más detalle ver **Anexo 2**.

## 5.9. Configuración ambiente de pruebas

El servidor utilizado para la prueba es una máquina virtual con las siguientes especificaciones:

- **RAM:** 2 GB
- **Sistema operativo:** Ubuntu Server 64 Bits
- **Almacenamiento:** 10 GB SSD

La red está configurada de manera que el servicio web sea accesible a través de internet.

## 5.10. Anexos pruebas

Los anexos de este capítulo se pueden encontrar en el repositorio del proyecto en la url <https://github.com/MauricioCDZ/webapp-dagma-wetlands/tree/master/Anexos>

- Anexo No.1 Casos de prueba
- Anexo No.2 Reporte herramientas de pruebas
- Anexo No.3 Reporte pruebas de caja blanca
- Anexo No.4 Particiones de equivalencia
- Anexo No.5 Indicadores de calidad



# Despliegue

---

Para el despliegue del aplicativo web se utilizaron servicios que provee Google Cloud Platform (GCP); se utilizaron 3 de ellos, los cuales son explicados a continuación.

## 6.1. App engine

Google App Engine (GAE) es una plataforma de computación en la nube como servicio (PaaS) para desarrollar y alojar aplicaciones web en centros de datos administrados por Google. Las aplicaciones están en espacio aislado y se ejecutan en varios servidores. App Engine ofrece escalado automático para aplicaciones web por lo que a medida que aumenta la cantidad de solicitudes para una aplicación, esta asigna automáticamente más recursos para que la aplicación web maneje la demanda adicional. Si no se cuenta con un dominio propio, Google proporciona uno con la siguiente estructura, `midominio.appspot.com`. Como se está usando con una cuenta gratuita, se tiene un límite de 500 megabyte de almacenamiento permanente, la suficiente cantidad de ancho de banda y CPU para cinco millones de visitas mensuales, y si la aplicación supera estas cuotas, se pueden comprar cuotas adicionales.

## 6.2. Cloud SQL

Cloud SQL es un servicio integrado en Google Cloud Platform que facilita la configuración, gestión, almacenamiento y mantenimiento de bases de datos relacionales en la nube.

## 6.3. Cloud storage

Es el servicio para almacenar datos no estructurados: objetos o archivos de cualquier tipo y tamaño(hasta 5 TB).

En GAE se pudo desplegar fácilmente el aplicativo web, hablando en términos del backend y frontend gracias a que tiene una conexión sencilla con lenguajes como Python. Luego de ello se configuró el acceso a una base de datos MySQL almacenada en una instancia de Cloud SQL con un almacenamiento de 10 GB SSD. Por último se configuro un bucket en Cloud storage para almacenar los archivos estáticos como las imágenes de los aportes, perfiles de usuario, flora y fauna de los humedales.

En la figura 6.1 se puede observar la arquitectura con los servicios de GCP.

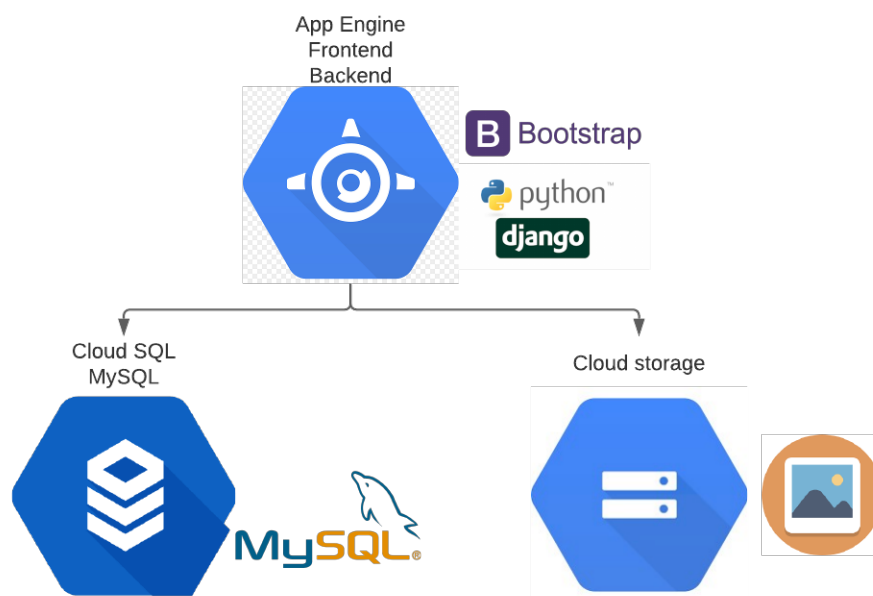


Figura 6.1: Arquitectura GCP.

# Resultados

---

A continuación se presenta el resultado final del desarrollo del proyecto, especificando los componentes de la aplicativo web.

## 7.1. Aplicativo web

En la figura 7.1 se puede observar la vista de registro y login del aplicativo.

The image shows a web interface with two main sections: 'Regístrate' (Register) and 'Login'. The 'Regístrate' section includes input fields for 'Nombre(s)', 'Email', 'Contraseña', and 'Repite la contraseña', along with a 'REGISTRARSE' button. The 'Login' section includes input fields for 'Correo' and 'Contraseña', a 'Clic Aquí' link for 'Olvidó su contraseña?', and an 'ENTRAR' button.

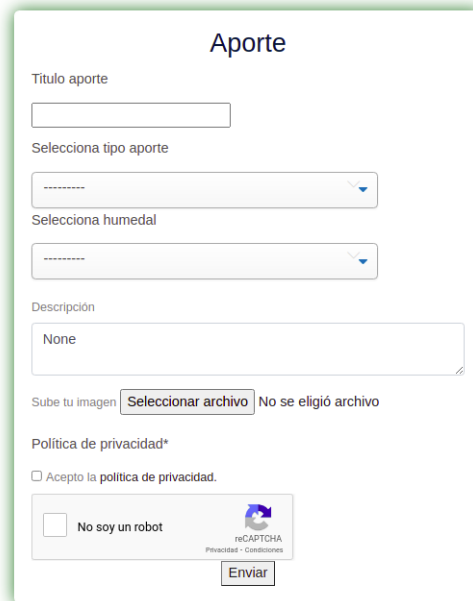
Figura 7.1: Login y registro.

### 7.1.1. Aporte

En la figura 7.2 se puede observar la vista del formulario que llenan los ciudadanos para realizar el aporte. Cada uno de los campos deben llenarse de manera correspondiente, de lo contrario no se podrá subir el aporte. Una vez se hayan llenado todos los campos, el ciudadano presiona el botón enviar.

### 7.1.2. Blog

En la figura 7.3 se puede observar el blog del aplicativo en el cual se publican los aportes más relevantes realizados por los ciudadanos. A este blog puede acceder cualquier usuario, además de que puede ver el detalle de cada aporte con solo presionar sobre el nombre de este.



The image shows a web form titled "Aporte" with the following fields and elements:

- Título aporte:** A text input field.
- Selecciona tipo aporte:** A dropdown menu with "-----" as the selected option.
- Selecciona humedal:** A dropdown menu with "-----" as the selected option.
- Descripción:** A text area containing the word "None".
- Sube tu imagen:** A button labeled "Seleccionar archivo" and the text "No se eligió archivo".
- Política de privacidad\*:** A checkbox labeled "Acepto la política de privacidad."
- reCAPTCHA:** A "No soy un robot" checkbox and the reCAPTCHA logo with the text "reCAPTCHA Privacidad - Condiciones".
- Enviar:** A submit button.

Figura 7.2: Vista aporte.

### 7.1.3. Aporte detalle

En la figura 7.4 se observa el detalle de un aporte realizado por un ciudadano, donde se detalla el humedal, la fecha, la descripción, autor y las que etiquetas que coloco la API Vision de Google a la fotografía.

### 7.1.4. Humedales

En la figura 7.5 se puede observar la vista del humedal Ecoparque las garzas, donde se especifican detalles sobre su cuerpo de agua, franja forestal, área total, jurisdicción y tipo de humedal. Además, se listan los distintos servicios que este provee así como las distintas características que lo diferencian de los otros dos humedales.

Realizando scroll hacia abajo se llega a la sección de detalle del humedal (Ver figura 7.6), donde se muestra el estudio de biodiversidad específico para este, el cual detalla el número de especies de fauna acuática, reptiles y anfibios, mastofauna, avifauna y flora. También, está el acceso a las galerías de flora, fauna y fauna acuática y por último se encuentra un mapa interactivo con la ubicación correspondiente del humedal.

### 7.1.5. Acceso al aplicativo web

Para ver las demás vistas que posee el aplicativo se puede acceder mediante la siguiente url <https://humedales-cali.uc.r.appspot.com/> o código QR (Ver figura 7.7).

## 7.2. Resultados con ciudadanos

Se probó el aplicativo web con un total de 10 ciudadanos que estaban caminando alrededor del humedal la babilla el día 15 de noviembre del 2021 entre las 10:00 AM y 12:00 PM, para realizar la prueba del aplicativo se abordó a cada uno de ellos y se explicó de breve manera el objetivo, propósito y funcionamiento del aplicativo web, enmarcado como un proyecto de grado. Se pidió probarlo subiendo un aporte con el cual quisieran dejar evidencia de algo que estuviera sucediendo en el humedal en ese momento, además de ello, se pidieron comentarios y críticas constructivas a modo de retroalimentación usando una encuesta; la encuesta contiene una serie de preguntas para evaluar la satisfacción de los ciudadanos y también un campo para comentarios y críticas; los resultados de la encuesta muestran un alto nivel de satisfacción por parte de los 10 ciudadanos.

### 7.2.1. Análisis de resultados

Gracias al uso de la encuesta se pudo evaluar el aplicativo en seis aspectos claves que consideramos importantes dada la relevancia en el cumplimiento de los requisitos funcionales, no funcionales y el correcto diseño del aplicativo con el uso de mockups. A continuación un análisis de varios de estos aspectos.

- **Velocidad de respuesta del aplicativo:** Esta pregunta se hizo con el objetivo de evaluar la experiencia del usuario en el uso del aplicativo de acuerdo a los tiempos de respuesta por cada acción que realizaba el ciudadano, entendiéndose como acción el dar clic sobre algún botón o enlace que lo llevase a un apartado del aplicativo o le permitiese subir aportes; luego de que el ciudadano navegara por varios apartados del aplicativo y realizara un aporte, se realizaba la pregunta. Siete ciudadanos respondieron que estaban “Satisfecho” y los tres restantes respondieron “Altamente satisfecho”, lo que demuestra la buena experiencia en los tiempos de respuesta desde la perspectiva del ciudadano.
- **Facilidad creación aporte:** Esta pregunta se hizo con el objetivo de evaluar la experiencia del usuario en la creación de un aporte, puesto que conocer el nivel de satisfacción del ciudadano, a su vez nos permitió evaluar el diseño de esta vista “Creación aporte” desde los mockups que se realizaron en el inicio del desarrollo del proyecto y que fueron analizados en conjunto con el equipo del DAGMA. Los resultados mostraron un alto índice de satisfacción puesto que siete ciudadanos respondieron “Altamente satisfecho” y los restantes “Satisfecho”. Mostrando que la realización de estos mockups impactaron en el favorable uso de la interfaz por parte de los ciudadanos. Sin embargo, un ciudadano en la casilla de comentarios, dio una recomendación en la que argumenta que podría mejorar la experiencia, si en lugar de buscar la imagen en

la galería, podría haber una opción en la que al dar clic se abriera la aplicación de cámara del dispositivo y así tomar la fotografía inmediatamente. Esta recomendación se toma para posible trabajo futuro relacionada al mejoramiento de la interfaz de usuario.

- **Claridad en el contenido del aplicativo:** Esta pregunta se hizo con el objetivo de evaluar la experiencia del usuario en cuanto al tamaño de los enunciados, localización de los títulos, tamaño de imágenes, etc. Siete ciudadanos mostraron estar “Satisfecho” y los tres restantes “Altamente satisfecho”. Mostrando un alto índice de satisfacción gracias al correcto diseño que se ajusta a distintos dispositivos móviles y navegadores. Por otro lado, dos ciudadanos recomendaron el uso de más imágenes relacionadas con las entidades que se encargan del cuidado de estos humedales. Esta recomendación se toma como trabajo futuro, en donde únicamente no exista información relacionada con estas entidades en la vista “Nuestra historia” sino que también en otros apartados del aplicativo.

Los resultados de la encuesta se hallan en el *Anexo No.6 Resultado encuesta* en la carpeta previamente mencionada en el capítulo de pruebas. La figura 7.8 muestra una serie de fotografías mientras se hacía la prueba con ciudadanos y algunos de sus aportes realizados.



Figura 7.3: Vista blog desde móvil.



Figura 7.4: Vista detalle aporte desde móvil.

Humedales Cali Aporta Con Tu Imagen

Home Nuestra Historia Humedales Recursos Deja Una Queja Involucrate Blog Administrador

## Ecoparque de las Garzas

Cuerpo de agua (HA)	0.8
Franja forestal protectora (HA)	3.9
Área total del humedal (HA)	4.7
Jurisdicción	Comuna 22

Tipo de humedal: Humedales artificiales, Tipo 2




### Servicios del Ecosistema


- Posee una gran riqueza florística para el territorio urbano
- Genera un área de refugio para un gran número de especies de fauna silvestre
- El humedal del Ecoparque cumple una gran función social como


Figura 7.5: Vista humedal.


Home Nuestra Historia Humedales Recursos Deja Una Queja Involucrate Blog Administrador


### Estudio de Biodiversidad (Especies)


  
**6**  
 Fauna Acuática


  
**12**  
 Reptiles y Anfibios


  
**8**  
 Mastofauna

  
**119**  
 Avifauna

  
**39**  
 Flora

  
 Flora

  
 Fauna

  
 Fauna Acuática




Figura 7.6: Vista humedal detalles.

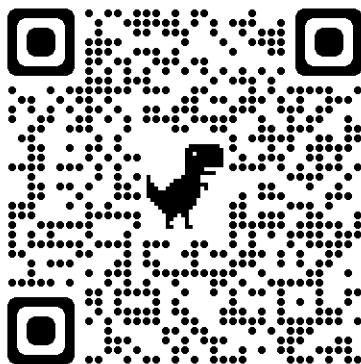


Figura 7.7: Acceso mediante código QR.



Figura 7.8: Evidencias pruebas con ciudadanos usando el aplicativo.



# Conclusiones

---

Las conclusiones generales del proyecto se dan con base en el cumplimiento de los objetivos propuestos y el alcance de estos:

- Se logró conectar a la ciudadanía con la tecnología con el fin de cuidar uno de nuestros más valiosos ecosistemas que tenemos en la ciudad, como lo son los humedales, sumado a la importancia del aporte de la propia comunidad al cuidado y conservación de estos para la generación de alertas tempranas que puedan ser revisadas por la entidad ambiental veedora en este caso DAGMA.
- Se entendió la importancia de servicios en la nube para el acceso fácil y rápido desde cualquier dispositivo con acceso a internet, apoyando el aplicativo web sobre servicios de alojamiento web, gestores de bases de datos, buckets de almacenamiento, API para el reconocimiento de imágenes, gracias al uso de Google Cloud Platform (GCP).
- Se entendió que herramientas basadas en Python como Django son de gran utilidad y fácil utilización para resolver problemas y poder satisfacer las necesidades de crear un aplicativo web.
- Se entendió a profundidad la utilización de metodologías ágiles para el desarrollo iterativo e incremental, donde los requisitos y soluciones fueron evolucionando mientras pasaba el tiempo de acuerdo a las necesidades del proyecto.
- Se continúa gestionando con el DAGMA las políticas públicas y permisos para el total despliegue del aplicativo web y así poder publicitarlo a fin de que toda la ciudadanía pueda enterarse y utilizarlo, se espera que a mitad del año 2022 este siendo desplegado totalmente.
- El aplicativo logra un apoyo importante al cuidado y conservación de los humedales a través de los aportes que la misma ciudadanía realiza; a diferencia de otros aplicativos como el de la ciudad de Bogotá llamado “Fundación Humedales Bogotá” [cp-21b] que ofrece distintos servicios como talleres virtuales, talleres ambientales, observación de aves; este no ofrece un medio para el aporte de la ciudadanía donde se pueda difundir el conocimiento de las distintas especies que habitan estos ecosistemas categorizadas como fauna terrestre, fauna acuática y flora; mientras que con este aplicativo se hace uso del contenido fotográfico e informativo de los aportes mediante la vista Blog o mediante el uso de las distintas galerías que existen para cada uno de los humedales, donde se puede apreciar con detalle las especies que habitan estos ecosistemas junto con su nombre científico.



# Bibliografía

- [Ano20a] Anonimo. *Advantages and Disadvantages of Java*. guru99.com, 2020.
- [Ano20b] Anonimo. *Historia de PHP*. php.net, 2020.
- [Ano20c] Anonimo. *The PHP Framework for Web Artisans*. Laravel.com, 2020.
- [Ano20d] Anonimo. *¿Qué es la tecnología Java y para qué la necesito?* Java.com, 2020.
- [BK07] S. Charters B. Kitchenham, B. Kitchenham. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. citeseerx, 2007.
- [Bun20] Alex Bunardzic. *Why is agile so much more successful than waterfall?* opensource.com, 2020.
- [Cal17] Alcaldía Santiago De Cali. *61 humedales entre públicos y privados hacen parte del paisaje de Santiago de Cali*. Alcaldía Santiago de Cali, 2017.
- [Cal20] Alcaldía Santiago De Cali. *Dagma realiza inversiones con recursos de sobretasa ambiental*. Departamento Administrativo de Gestión del Medio Ambiente, 2020.
- [cp-] Segmentation. <https://www.bioss.ac.uk/people/chris/ch4.pdf>, year=2018.
- [cp-18] Recent developments in deep learning for engineering applications. <https://www.hindawi.com/journals/cin/2018/7068349/>, 2018.
- [cp-19a] The complete beginner's guide to deep learning: Convolutional neural networks and image classification. <https://towardsdatascience.com/wtf-is-image-classification-8e78a8235acb>, 2019.
- [cp-19b] A gentle introduction to the rectified linear unit (relu). <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-ne> 2019.
- [cp-20a] Introducing tailwind css: Pros and cons of tailwind. <https://hackernoon.com/introducing-tailwind-css-pros-and-cons-of-tailwind-hd2k3zt2>, 2020.
- [cp-20b] Three-tier architecture. <https://www.ibm.com/cloud/learn/three-tier-architecture>, 2020.
- [cp-20c] Vision ai. <https://cloud.google.com/vision>, 2020.
- [cp-20d] What is image recognition and how it is used? <https://www.mygreatlearning.com/blog/image-recognition/>, 2020.

- [cp-21a] Computer vision. <https://azure.microsoft.com/es-es/services/cognitive-services/computer-vision/>, 2021.
- [cp-21b] Fundación humedales bogotá. <https://humedalesbogota.com/>, 2021.
- [cp-21c] Visual recognition - ibm cloud api docs. <https://cloud.ibm.com/apidocs/visual-recognition/visual-recognition-v3>, 2021.
- [cp-21d] What is recaptcha? <https://developers.google.com/recaptcha>, 2021.
- [cpa21] Python advantages and disadvantages - step in the right direction. <https://techvidvan.com/tutorials/python-advantages-and-disadvantages/>, Mar 2021.
- [cpd21] Meet django. <https://www.djangoproject.com/>, Mar 2021.
- [cpm21] Modelo-vista-controlador. <https://es.wikipedia.org/wiki/Modelo-vista-controlador>, 2021.
- [cpp] What is python? executive summary. <https://www.python.org/doc/essays/blurb/>.
- [CRW19] R Carroll, J K Reynolds, and I A Wright. *Geochemical impact of urban development on fragile freshwater wetlands*, volume 344. IOP Publishing, nov 2019.
- [Cum01] Ward Cunningham. *Manifiesto por el Desarrollo Ágil de Software*. agilemanifesto, 2001.
- [dEP20] Redacción de El País. *¿Por qué vivir en Cali es un paraíso?, los datos de nuestro tesoro natural*. elpais, 2020.
- [dGdMA17] Departamento Administrativo de Gestión del Medio Ambiente. *Humedales de Santiago de Cali*. Alcaldía Santiago de Cali, 2017.
- [E.20] Abellan E. *Scrum: qué es y cómo funciona esta metodología*. wearemarketing, 2020.
- [Jim20] Rafael Jiménez. Los 7 mejores frameworks de java de 2020. <https://openwebinars.net/blog/los-7-mejores-frameworks-de-java/>, Jun 2020.
- [Lib15] ACM Digital Library. *ACM Computing Classification System*. Association for Computing Machinery, 2015.
- [Luc21] Lucidchart. *Estructura de Desglose de Trabajo*. lucidchart, 2021.
- [MBA96] Cruz I. Myers B. A., Hollan J. *Strategic Directions in Human Computer Interaction*. ACM Computing Surveys, 1996.
- [MSR20] N. R. Mahanta, A. Samuel, and B. Rajput. *Community Participation and Bio Rights Possibilities in the Conservation of Urban Wetlands: A Study of Deepor Beel (Lake), Guwahati, India*. 2020.

- [NHN<sup>+</sup>18] Long H. Nguyen, Rattikorn Hewett, Akbar S. Namin, Nicholas Alvarez, Cristina Bradatan, and Fang Jin. *Smart and Connected Water Resource Management via Social Media and Community Engagement*. ASONAM '18. IEEE Press, 2018.
- [Ram18] Ramsar. *La importancia de los humedales*. THE RAMSAR CONVENTION SECRETARIAT, 2018.
- [V19a] Rosello V. *Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa*. iebschool, 2019.
- [V19b] Rosello V. *Las metodologías ágiles más utilizadas y sus ventajas dentro de la empresa*. iebschool, 2019.
- [WCC<sup>+</sup>16] C. Wu, C. Cao, W. Chen, R. Tian, and D. Liu. *Construction and application of the diagnostic indicator system of wetland health based on remote sensing*. 2016.
- [WIE61] D. H. HUBEL AD T. N. WIESEL. *RECEPTIVE FIELDS, BINOCULAR INTERACTION AND FUNCTIONAL ARCHITECTURE IN THE CAT'S VISUAL CORTEX*. Harvard, 1961.
- [Wik20] la enciclopedia libre Wikipedia. *Manifiesto ágil*. Wikipedia, 2020.
- [Yam18] Nishio M. Do R.K.G. Yamashita, R. *Convolutional neural networks: an overview and application in radiology*. insightsimaging, 2018.
- [YFQ<sup>+</sup>14] Z. Yan, C. Fang, S. Qi, T. Yang, D. Gao, Z. Chen, and Y. Wang. *Ecological Monitoring Scheme Based on Wireless Sensor Network in Baisha Lake of the Nanji Wetland Nation Reserve*. 2014.