



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INGENIERÍA,
INDUSTRIA Y CONSTRUCCIÓN**

CARRERA DE INGENIERÍA CIVIL

**IDENTIFICACIÓN DE PATOLOGÍAS ESTRUCTURALES
MEDIANTE PROCESAMIENTO DE IMÁGENES.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO CIVIL**

AUTORES: ADRIAN PATRICIO CASTRO YUPANGUI

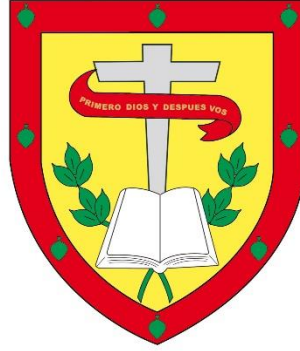
JORGE EDUARDO CONTRERAS CONTRERAS

DIRECTOR: ING. MSC. EDUARDO PALMA ZAMBRANO

CUENCA - ECUADOR

2021

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INGENIERÍA,
INDUSTRIA Y CONSTRUCCIÓN**

CARRERA DE INGENIERÍA CIVIL

**IDENTIFICACIÓN DE PATOLOGÍAS ESTRUCTURALES
MEDIANTE PROCESAMIENTO DE IMÁGENES.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO CIVIL**

AUTORES: ADRIAN PATRICIO CASTRO YUPANGUI,

JORGE EDUARDO CONTRERAS CONTRERAS

DIRECTOR: ING. MSC. EDUARDO PALMA ZAMBRANO

CUENCA - ECUADOR

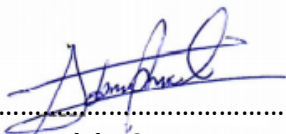
2021

DIOS, PATRIA, CULTURA Y DESARROLLO

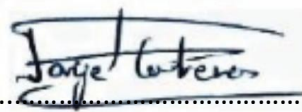
Declaratoria de Autoría y Responsabilidad

Adrián Patricio Castro Yupangui y Jorge Eduardo Contreras Contreras portadores de las cédulas de ciudadanía N° **1400891535** y N° **0106831530**. Declaramos ser los autores de la obra: **“IDENTIFICACIÓN DE PATOLOGÍAS ESTRUCTURALES MEDIANTE PROCESAMIENTO DE IMÁGENES”**, sobre la cual nos hacemos responsables sobre las opiniones, versiones e ideas expresadas. Declaramos que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximimos a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaramos finalmente que nuestra obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también nos responsabilizamos y eximimos a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, 21 de diciembre de 2021



Adrián Patricio Castro Yupangui
C.I. **1400891535**



Jorge Eduardo Contreras Contreras
C.I. **0106831530**

CERTIFICACIÓN

Yo, Ing. M.s.C Eduardo Palma Zambrano certifico que la presente tesis realizada por los Sres. Jorge Eduardo Contreras Contreras y Adrián Patricio Castro Yupangui egresados de la UNIDAD ACADÉMICA DE INGENIERÍA, INDUSTRIA Y CONSTRUCCIÓN, se desarrolló bajo mi supervisión y tutoría, siendo un trabajo elaborado de manera personal e inédito, bajo el tema “IDENTIFICACIÓN DE PATOLOGÍAS ESTRUCTURALES MEDIANTE PROCESAMIENTO DE IMÁGENES”.

Es todo cuanto puedo certificar en honor a la verdad.



Ing. M.s.C Eduardo Palma Zambrano

DIRECTOR

CERTIFICACIÓN

Yo, Ing. Elec. Juan Carlos Cobos Torres, Ph.D certifico que la presente tesis realizada por los Sres. Jorge Eduardo Contreras Contreras y Adrián Patricio Castro Yupangui egresados de la UNIDAD ACADÉMICA DE INGENIERÍA, INDUSTRIA Y CONSTRUCCIÓN, se desarrolló bajo mi supervisión y tutoría, siendo un trabajo elaborado de manera personal e inédito, bajo el tema “IDENTIFICACIÓN DE PATOLOGÍAS ESTRUCTURALES MEDIANTE PROCESAMIENTO DE IMÁGENES”.

Es todo cuanto puedo certificar en honor a la verdad.



Ing. Elec. Juan Carlos Cobos Torres, Ph.D.

ASESOR METOLÓGICO

AGRADECIMIENTO

A Dios en primer lugar por brindarnos la sabiduría para culminar con éxitos nuestros estudios universitarios.

A nuestra familia por el apoyo constante e incondicional, por todos los sacrificios y palabras de aliento que nos daban la fuerza para alcanzar nuestra meta y nunca haber perdido la fe en nosotros.

A mis profesores por haber guiado la semilla del saber, comprender, aprender y en especial a dos personas maravillosas que supieron guiarnos en el camino del conocimiento: Ingenieros Eduardo Palma y Juan Cobos gracias por ser nuestros mentores.

A nuestros amigos que durante todo el tiempo de conocernos estuvieron alentándonos a seguir adelante y no ceder ante las adversidades.

A todos quienes, de una u otra manera nos mostraron su apoyo para la realización de este documento, sin más palabras como agradecerles.

¡Gracias!

RESÚMEN

Los sistemas de reconocimiento de imágenes han recibido un fuerte impulso en la actualidad gracias al avance en la tecnología, y a su aplicación en campos muy diversos, como la ingeniería civil, así como otras áreas de la ciencia de materiales y la construcción. Este documento presenta el proceso para la identificación de patologías en estructuras de obras civiles y arquitectónicas, por medio del procesamiento de imágenes; dividida en dos tareas. La primera de ellas, la detección de similitud de una imagen, en una imagen fija. La segunda tarea, el reconocimiento que consiste en un sistema inteligente que fue entrenado con las imágenes almacenadas previamente en una base de datos. Para el caso de investigación, se implementó el método de reconocimiento con la herramienta web de Google llamada Teachable Machine, para la búsqueda por características que tenía cada patología estructural analizada. Se realizó un conjunto de pruebas sobre las bases de datos de imágenes para analizar y comprobar la eficiencia del sistema, obteniendo una precisión promedio del 91.6 % para las patologías de Desgarro laminar, Deformación por impacto y Corrosión por picadura, y una precisión promedio superior al 87.3 % para las patologías de Grietas por asentamiento diferencial, Fisuras capilares y Hongos, esto debido a que los colores en las imágenes fueron parecidos.

Palabras clave: sistema operativo, patologías estructurales, algoritmo

ABSTRACT

Image recognition systems have received a strong impulse nowadays thanks to the advance in technology, and their application in very diverse fields, such as civil engineering, as well as other areas of material science and construction. This paper presents the process for the identification of pathologies in civil and architectural structures using image processing, divided into two tasks. The first task is the detection of similarity of an image in a still image. The second task, the recognition that consists of an intelligent system that was trained with the images previously stored in a database. For the research case, the recognition method was implemented with the Google web tool called Teachable Machine, for the search by features that had each structural pathology analyzed. A set of tests were performed on the image databases to analyze and check the efficiency of the system, obtaining an average accuracy of 91.6 % for the pathologies of Laminar Tearing, Impact Deformation and Pitting Corrosion, and an average accuracy higher than 87.3 % for the pathologies of Differential Settlement Cracks, Hairline Cracks, and Fungi, this since the colors in the images were similar.

Keywords: operating system, structural pathologies, algorithm

ÍNDICE DE CONTENIDO

RESÚMEN	IV
ABSTRACT	V
CAPITULO 1	10
1. INTRODUCCIÓN	10
1.1 FORMULACIÓN DEL PROBLEMA	11
1.2 DELIMITACIÓN DEL PROBLEMA	12
1.3 JUSTIFICACIÓN	12
1.4 OBJETIVOS	13
1.4.1 <i>Objetivo general</i>	13
1.4.2 <i>Objetivos específicos</i>	13
1.5 ALCANCE	14
CAPÍTULO 2	16
2 MARCO TEÓRICO Y ESTADO DEL ARTE	16
2.1 DEFINICIÓN DE LAS PATOLOGÍAS EN LAS ESTRUCTURAS	16
2.2 MECANISMOS PARA DETERMINAR LAS PATOLOGÍAS EN ESTRUCTURAS.....	17
2.3 EVALUACIÓN DE PATOLOGÍAS EN EL ACERO Y HORMIGÓN.....	17
2.3.1 <i>Factores que intervienen en las patologías estructurales de Acero</i>	17
2.3.2 <i>Principales patologías en el Acero</i>	18
2.3.3 <i>Evaluación de las características para el análisis de patologías en acero</i>	19
2.3.4 <i>Factores que intervienen en las patologías en estructuras de Hormigón</i>	25
2.3.5 <i>Principales patologías en el hormigón</i>	25
2.3.6 <i>Evaluación de las características para el análisis de patologías en hormigón</i>	26
2.4 NORMAS Y CÓDIGOS USADAS EN LA VALORACIÓN DE LAS PATOLOGÍAS	35
2.4.1 <i>Códigos y normas de referencia.</i>	35
2.5 ANÁLISIS DEL MEDIO DE PATOLOGÍAS ESTRUCTURALES	37
2.5.1 <i>Obtención de muestras</i>	37
2.5.2 <i>Captura de imágenes</i>	37
2.5.3 <i>Base de datos</i>	37
2.5.4 <i>Entrenamiento automático y la Inteligencia Artificial</i>	38
ESTADO DEL ARTE	39
2.6 VISIÓN ARTIFICIAL.....	39
2.6.1 <i>Aplicaciones de la visión artificial</i>	41
2.6.2 <i>Etapas para el sistema de visión artificial</i>	42
2.6.3 <i>Fundamentos de la visión artificial</i>	43
2.7 HARDWARE Y SOFTWARE EN EL PROCESAMIENTO DE IMÁGENES.....	45
2.7.1 <i>Elementos para el procesamiento</i>	46
2.8 HARDWARE	46
2.9 SOFTWARE	47
2.9.1 <i>iOS de Apple</i>	47
2.9.2 <i>Swift</i>	50
2.9.3 <i>Android</i>	51
2.9.4 <i>Kotlin</i>	53

2.9.5	<i>Teachable Machine</i>	54
2.10	CLASIFICADORES DE BÚSQUEDA PARA LA APLICACIÓN	59
2.10.1	<i>Basados en Características</i>	60
2.10.2	<i>Basados en Imágenes</i>	61
CAPÍTULO 3		63
METODOLOGÍA		63
3. IMPLEMENTACIÓN DEL SISTEMA, DETECCIÓN DE PATOLOGÍAS ESTRUCTURALES PARA EL ACERO Y HORMIGÓN		65
3.1	INFORMACIÓN INICIAL.....	66
3.1.1	<i>Población</i>	66
3.1.2	<i>Entorno</i>	67
3.2	CÓDIGO DE BÚSQUEDA POR IMAGEN EN TEACHABLE MACHINE EN FORMATO JAVASCRIPT PARA EL ACERO 68	
3.3	CÓDIGO DE BÚSQUEDA POR IMAGEN EN TEACHABLE MACHINE EN FORMATO JAVASCRIPT PARA EL HORMIGÓN	70
CAPÍTULO 4		72
4. EJECUCIÓN DE LAS PRUEBAS Y ANÁLISIS DE RESULTADOS		72
4.1	PRUEBAS USANDO TEACHABLE MACHINE	72
4.2	PRUEBAS APLICANDO EL FILTRADO DE LAS IMÁGENES	75
CONCLUSIONES		80
4.3	TRABAJOS FUTUROS	81
4.4	RECOMENDACIONES	81
BIBLIOGRAFÍA		82
APÉNDICE A		87
ESPECIFICACIONES DE LA APLICACIÓN E INSTRUCTIVO DE USO		87
APÉNDICE B		94
FILTRADO DE INFORMACIÓN EN LA BASE DE DATOS		94
APÉNDICE C		97
CÓDIGO SWIFT PARA FUNCIONAMIENTO DE APLICACIÓN EN EL SISTEMA IOS		97
APÉNDICE D		117
4.5	CÓDIGO KOTLIN PARA FUNCIONAMIENTO DE APLICACIÓN EN EL SISTEMA ANDROID.....	117

INDICE DE ILUSTRACIONES

ILUSTRACIÓN 1. MODELO DE INTERACCIÓN ENTRE LA APLICACIÓN Y EL USUARIO (JONGGI HONG, 2019).....	39
ILUSTRACIÓN 2. ESQUEMA DE CLASIFICACIÓN DE IMAGEN (MALLICK, 2016).	41
ILUSTRACIÓN 3. ENCEFALOGRAMA: (A) IMAGEN ORIGINAL, (B) IMAGEN PROCESADA (MALPARTIDA, 2003).....	42
ILUSTRACIÓN 4. FALLA ESTRUCTURAL: (A) IMAGEN ORIGINAL, (B) IMAGEN PROCESADA (MALPARTIDA, 2003) .	42
ILUSTRACIÓN 5. ETAPAS DEL SISTEMA DE VISIÓN ARTIFICIAL (AUTORES).	43
ILUSTRACIÓN 6. ETAPAS FUNDAMENTALES PARA EL PROCESAMIENTO DE IMÁGENES. (AUTORES)	44
ILUSTRACIÓN 7. ARQUITECTURA DE CAPAS iOS (ARQUITECTURA iOS - TECNOLOGÍA iOS., 2021).	48
ILUSTRACIÓN 8. MODELO INTERACTIVO DE SWIFT (SWIFT - APPLE DEVELOPER, 2021).	50
ILUSTRACIÓN 9. INTERFACE DE TEACHABLE MACHINE (TEACHABLE MACHINE, 2021).....	56
ILUSTRACIÓN 10. INTRODUCCIÓN DE TIPOS DE CLASES (TEACHABLE MACHINE, 2021).	57
ILUSTRACIÓN 11. GENERACIÓN DE LA BASE DE IMÁGENES (TEACHABLE MACHINE, 2021).	57
ILUSTRACIÓN 12. CORRIDO DEL MODELO (TEACHABLE MACHINE, 2021).....	58
ILUSTRACIÓN 13. OPCIONES DE BÚSQUEDA (AUTORES)	59
ILUSTRACIÓN 14. DIAGRAMA DE FLUJO, USO DE APLICACIÓN (AUTORES).....	65
ILUSTRACIÓN 15. DIAGRAMA DE FLUJO, COMPOSICIÓN DE TEACHABLE MACHINE (AUTORES)	68
ILUSTRACIÓN 16. INTERPRETACIÓN DE PRUEBAS DEL MODELO PARA ACERO (AUTORES).....	73
ILUSTRACIÓN 17. INTERPRETACIÓN DE PRUEBAS DEL MODELO PARA HORMIGÓN (AUTORES)	74
ILUSTRACIÓN 18. PANTALLA PRINCIPAL DE LA PATOLOGÍAS APP (AUTORES)	75
ILUSTRACIÓN 19. BÚSQUEDA MEDIANTE IMÁGENES (AUTORES)	76
ILUSTRACIÓN 20. RESULTADO DE BÚSQUEDA EN PORCENTAJES (AUTORES).....	77
ILUSTRACIÓN 21. RESULTADO DE BÚSQUEDA EN PORCENTAJES (AUTORES)	78
ILUSTRACIÓN 22. RESULTADO FINAL DE LA BÚSQUEDA, DONDE SE EXPLICA LA DESCRIPCIÓN, EQUIPOS DE MEDICIÓN, TRATAMIENTOS Y TÉCNICAS DE RECONOCIMIENTO (AUTORES)	78
ILUSTRACIÓN 23: BÚSQUEDA POR MEDIO DE CARACTERÍSTICAS.....	79
ILUSTRACIÓN 24: RESULTADO DE BÚSQUEDA POR CARACTERÍSTICAS, ADEMÁS SE PRESENTA LA OPCIÓN PARA QUE EL USUARIO PUEDA DEJAR SU APORTE FOTOGRÁFICO.....	79
ILUSTRACIÓN 25. ICONO DE APLICACIÓN (AUTORES).	88
ILUSTRACIÓN 26. PANTALLA DE GENERAL DE LA APLICACIÓN (AUTORES).	88
ILUSTRACIÓN 27. ACCESO A LA BÚSQUEDA POR IMAGEN (AUTORES).....	89
ILUSTRACIÓN 28. ENLACE WEB PARA LA BÚSQUEDA POR IMAGEN (AUTORES).	89
ILUSTRACIÓN 29. APARTADO QUE PERMITE EL ACCESO DE IMÁGENES (AUTORES).	90
ILUSTRACIÓN 30. RESULTADOS DE BÚSQUEDA (AUTORES).	90
ILUSTRACIÓN 31. LISTADOS DE PATOLOGÍAS (AUTORES).....	91
ILUSTRACIÓN 32. PANTALLA DE RESULTADOS DE BÚSQUEDA (AUTORES).	91
ILUSTRACIÓN 33. PANTALLA DE CARACTERÍSTICAS (AUTORES).	92
ILUSTRACIÓN 34. SELECCIÓN DE CARACTERÍSTICAS (AUTORES).....	92
ILUSTRACIÓN 35. OPCIÓN PARA DEJAR UN APORTE (AUTORES).	93
ILUSTRACIÓN 36. PANTALLA CON MENSAJE DE CONFIRMACIÓN PARA UN APORTE (AUTORES).....	93
ILUSTRACIÓN 37. BASE DE DATOS PARA ALMACENAR LAS IMÁGENES (AUTORES).....	94
ILUSTRACIÓN 38. BASE DE DATOS PARA LOS TIPOS DE MATERIALES (AUTORES).	95
ILUSTRACIÓN 39. BASE DE DATOS QUE CONTIENE LOS RESULTADOS DE CADA TIPO DE PATOLOGÍA (AUTORES)....	96

INDICE DE GRÁFICOS

GRÁFICO 1. PATOLOGÍAS EN ESTRUCTURAS DE ACERO. (JIMENEZ, 2021).....	18
GRÁFICO 2. PATOLOGÍAS EN ESTRUCTURAS DE HORMIGÓN, (CORDERO, 2021).....	26
GRÁFICO 3. NORMAS PARA EL ESTUDIO DE LAS PATOLOGÍAS EN ACERO Y HORMIGÓN. (AUTORES).....	36
GRÁFICO 4. FASES DEL ANÁLISIS DE IMÁGENES (AUTORES).....	43
GRÁFICO 5. PROCESAMIENTO DE IMÁGENES, (GARCÍA, 2008).....	45
GRÁFICO 6. CONDICIONES DEL SISTEMA SWIFT, (INC, 2021).....	51
GRÁFICO 7. DEFINICIÓN DE SOFTWARE STACK, (KRISTEL MALAVE POLANCO, 2011).....	52
GRÁFICO 8. DETALLES DE KOTLIN, (GOOGLE, 2021).....	54
GRÁFICO 9. PROPIEDADES EN DESCRIPTORES DE CARACTERÍSTICAS, (BRAVO, 2015).....	61

INDICE DE TABLAS

TABLA 1: DESCRIPCIÓN DE PATOLOGÍAS ESTRUCTURALES. (JIMENEZ, 2021).....	24
TABLA 2. DESCRIPCIÓN DE PATOLOGÍAS ESTRUCTURALES (CORDERO, 2021).....	34
TABLA 3. ESPECIFICACIONES TÉCNICAS DEL COMPUTADOR (AUTORES).....	47
TABLA 4. CARACTERÍSTICAS DE ACCESIBILIDAD DE IOS (SEPULVEDA, 2013).	49
TABLA 5. NÚMERO DE IMÁGENES POR CLASES DE PATOLOGÍAS EN ACERO CLASIFICADAS (AUTORES).	66
TABLA 6. NÚMERO DE IMÁGENES POR CLASES DE PATOLOGÍAS EN HORMIGÓN CLASIFICADAS (AUTORES).....	67

CAPITULO 1

1. INTRODUCCIÓN

En las patologías estructurales se requiere de una comprensión avanzada, y así hacer un análisis e intervenir en sus posibles causas y su posterior reparación; es así que se propone un estudio de comparación mediante el uso de la inteligencia artificial, que en los últimos años ha demostrado tener una gran capacidad en el desarrollo y manejo de información, mediante la identificación de objetos, donde se encarga de buscar y encontrar similitudes de manera automática mediante la comparación de imágenes, puesto que, últimamente se ha visto aplicada tanto en el ámbito empresarial como en el industrial.

Se apuesta con firmeza que los resultados obtenidos en el uso del reconocimiento de características en imágenes sean cada vez más precisos, basados en procesar las mismas en una base de datos de gran volumen, con resultados capaces de superar la capacidad humana, para identificar patologías en las estructuras de obras civiles y arquitectónicas.

Al evaluar el estado de las obras de construcción, ya sean de las edificaciones de centros históricos o de construcciones vanguardistas, se puede percatar que, en algunos de los casos, ya se han realizado arreglos en sus diferentes elementos estructurales debido a que la creciente población demanda lugares más espaciosos o por el pasar de los años y la inclemencia del clima.

A los aspectos que forman parte del proceso de inspección, se la conoce como la identificación de patologías, término usado originalmente en la medicina, que, trasladándolo al ámbito de las obras de construcción, se tratan de “enfermedades” en elementos estructurales de manera local o de manera global para una edificación.

Detectar de manera correcta las patologías en las estructuras, ahorra gastos a futuro, al contrario, cuando no se hace ningún mantenimiento de la obra de construcción los gastos pueden ser mayores cuando se encuentran daños irreparables. Por lo que, en la presente investigación, se abordará la metodología y las consideraciones para obtener un algoritmo que servirá para la identificación y análisis de las patologías para el acero y hormigón.

La Inteligencia Artificial surge luego de unos trabajos publicados en la década de 1940 que no tuvieron la importancia esperada, pero luego, una publicación influyo en el trabajo de 1950 de Alan Turing, matemático británico, es entonces que nace un nuevo campo en las ciencias de la información (Victor Mendoza Lanche, 2016).

Ya con esto, se podía abrir nuevos campos de desarrollo como la robótica, la comprensión y traducción de lenguajes, el reconocimiento y aprendizaje de palabras controladas por máquinas en sistemas computacionales expertos, así como el procesamiento de imágenes, que son algunos que imitan el comportamiento humano.

En el actual trabajo de investigación de grado, se presenta una herramienta desarrollada para la identificación de patologías en estructuras de acero y hormigón, con la implementación de una aplicación móvil llamada “PATOLOGÍAS APP” que se creará para los sistemas operativos iOS y Android respectivamente.

1.1 Formulación del problema

Las edificaciones forman parte de la sociedad, estas proporcionan un valor cultural y estético a sus propietarios y a las ciudades. En general existen patologías congénitas, que es cuando las estructuras recién construidas las poseen, también existen patologías que las estructuras pueden ir adquiriendo después de la construcción, es decir, durante su vida útil.

Estos dos tipos de patologías causan efectos negativos, que en la mayoría de los casos son muy difíciles de predecir.

Estas patologías se pueden presentar en construcciones de estructuras de acero, hormigón, piedra, madera, entre otros; las cuales, hay que identificarlas lo más pronto posible, caso contrario puede afectar el rendimiento de la estructura y con el paso del tiempo, el colapso de la misma.

1.2 Delimitación del problema

En base a la problemática expuesta anteriormente, se entiende la importancia de identificar los tipos de patologías de manera local, es decir, que se centra en el análisis de un área específica del elemento estructural. Las patologías estructurales son comunes en nuestro medio, ya que se pueden presentar con diferentes características y extensiones variadas en los elementos estructurales.

El tema de este proyecto, se delimitará en analizar las patologías estructurales de acero y hormigón, utilizando la inteligencia artificial en el reconocimiento de imágenes, por medio de una aplicación móvil.

1.3 Justificación

Hoy en día las estructuras tienen un valor estético, patrimonial y lo más fundamental, que garantice la seguridad de sus ocupantes. En los últimos años, se ha sembrado la cultura de análisis y evaluación de las estructuras por medio de aplicaciones informáticas y modelos matemáticos. La continua evolución de los diferentes campos de la inteligencia artificial, las

numerosas aplicaciones que aparecen cada día. Lo que antes parecía algo futurista, se está convirtiendo en el futuro más próximo. Las empresas cada vez invierten más en este campo para solucionar de forma eficiente sus problemas. Sin lugar a duda, esto no es el trabajo del futuro, es el trabajo del presente (O. Rodríguez, 2018).

Para aprovechar los avances en investigación de la inteligencia artificial, se creará un sistema automatizado para la detección de patologías por medio del reconocimiento de imágenes, que facilitará a los profesionales de la construcción, el control y solución de las obras.

A partir de las bases de datos creadas en acero (Jimenez, 2021) y hormigón (Cordero, 2021), se desarrolló una aplicación móvil que ayudará en el reconocimiento de estas patologías, partiendo de la información recolectada y organizada anteriormente.

1.4 Objetivos

1.4.1 Objetivo general

- Desarrollar un sistema de detección de patologías en las estructuras de obras civiles, mediante el procesamiento de imágenes e inteligencia artificial, para localizar estas patologías de manera temprana.

1.4.2 Objetivos específicos

- Indagar información científica sobre el procesamiento de imágenes, enfocado a la detección de patologías en los elementos estructurales, en bases de datos

especializadas, para obtener los conocimientos necesarios y posteriormente desarrollar el sistema.

- Compilar una base de datos con imágenes de libros, artículos, revistas, entre otros, que permita el entrenamiento del algoritmo del sistema.
- Diseñar un sistema de detección de patologías que, mediante su entrenamiento, consiga la automatización y así localizar patologías estructurales.
- Realizar pruebas del sistema, a partir del análisis de una estructura específica, con imágenes capturadas en la misma y comprobar la eficiencia en la detección del tipo de fallas.

1.5 Alcance

El alcance de este estudio es realizar una base de datos de las patologías estructurales existentes en todo tipo de construcción, los materiales a los cuales se ha propuesto para realizar este estudio son el acero y hormigón.

Lo que se plantea en principio, es realizar una base de datos donde se almacenan los tipos de patologías con sus respectivas descripciones, características, equipos de medición, las técnicas de reconocimientos y los posibles tratamientos que se les pueda dar a estas enfermedades estructurales. Luego de esto, hacer un levantamiento de campo en donde consta la toma de fotografías de acuerdo a cada patología según el tipo de material.

Terminado esta base de datos, se propone realizar una aplicación sobre la identificación de patologías mediante el reconocimiento de imágenes y la inteligencia artificial, en este punto la aplicación se realizó únicamente para los materiales de “Acero” y “Hormigón” debido a que existen mayor información expuesta en la base de datos realizado por (Jimenez, 2021) y (Cordero, 2021).

Esta aplicación permite hacer una búsqueda mediante el reconocimiento de imágenes y por medio de las características de los tipos de patología de cada material. Para el reconocimiento de imágenes se está empleando una herramienta de Google llamado Teachable Machine la cual permite el trabajo del reconocimiento de imágenes.

Adicionalmente, existe una opción para que el usuario pueda compartir un aporte fotográfico de patología que será añadida a la base de datos, este aporte ayudará a que la base de datos crezca y mejore el rendimiento de dicha aplicación.

CAPÍTULO 2

2 MARCO TEÓRICO Y ESTADO DEL ARTE

En este capítulo se expone y se detalla conceptos generales de Ingeniería Civil y Programación, los cuales serán necesarios para la elaboración del sistema de reconocimiento de patologías estructurales en el acero y hormigón. De tal manera, que se pueda generar un mayor enfoque en la aplicación y el reconocimiento de las patologías.

2.1 Definición de las patologías en las estructuras

Las patologías estructurales son enfermedades anormales generadas por causas conocidas o desconocidas. También se entiende como patología estructural al estudio del comportamiento de las estructuras cuando presentan evidencias de fallas, buscando detectar sus causas y proponer acciones correctivas o su demolición (Kamintetzki, 1991).

La vulnerabilidad de las estructuras suele reflejarse a través de patologías que son visibles en las edificaciones originando múltiples efectos, desde daños y molestias para sus ocupantes, hasta grandes fallas que puede causar el colapso de la edificación (ASCE, 2018).

Una falla no necesariamente significa un colapso total, pero también puede aplicarse a una limitación, como cuando una construcción no puede desempeñarse de acuerdo con la intención original; estas patologías que se manifiestan en las edificaciones son infinitas, difícilmente se logra determinar con precisión las causas que conllevan al fallo de la estructura como tal (ASCE, 2018).

Una falla de un elemento estructural ocurre cuando esta cesa de desempeñar su función en forma satisfactoria; por lo tanto, es de gran importancia para toda construcción hacer los debidos ensayos requeridos por las normas y leyes vigentes que permitan una

buena funcionabilidad de la edificación y así brindar un ambiente de seguridad durante toda la vida útil de la estructura (Kamintetzki, 1991).

2.2 Mecanismos para determinar las patologías en estructuras

Los mecanismos empleados para determinar las patologías estructurales en los materiales de acero y hormigón, se hará mediante el uso de una base de datos (Jimenez, 2021) y (Cordero, 2021), donde se realizará un proceso de recopilación y posteriormente la selección de estas imágenes digitales. Las mismas que ayudarán a reconocer con más facilidad la posible falla o el tipo de patología existente en cualquier tipo de construcción, estas patologías pueden darse durante el proceso de construcción o con el tiempo, es decir de acuerdo a las cargas o agentes físicos o químicos a los que se haya sometido estos materiales.

2.3 Evaluación de patologías en el Acero y Hormigón

2.3.1 Factores que intervienen en las patologías estructurales de Acero

Mediante el análisis de las patologías de acero y hormigón se describen en la base de datos expuesta en (Jimenez, 2021) y (Cordero, 2021), dan a conocer los diferentes tipos de características que presenta cada patología y así poder identificar el tipo de falla existente el cualquiera de estos materiales.

Una estructura de acero que está afectada por patologías, ocasiona una disminución de su resistencia, estabilidad y durabilidad en el tiempo. Por lo que, los factores involucrados en este análisis pueden ser de manera directa como la sobrecarga, temperaturas que rebasan la resistencia, la constante humedad y la falta de mantenimiento que aumenta la presencia de elementos como la vegetación, el polvo y los desechos (Jimenez, 2021).

2.3.2 Principales patologías en el Acero

Las condiciones que dan origen a una patología son diversas, es posible establecer características específicas que contendrá una patología, llegando así a definir las patologías expresadas en el Gráfico 1:

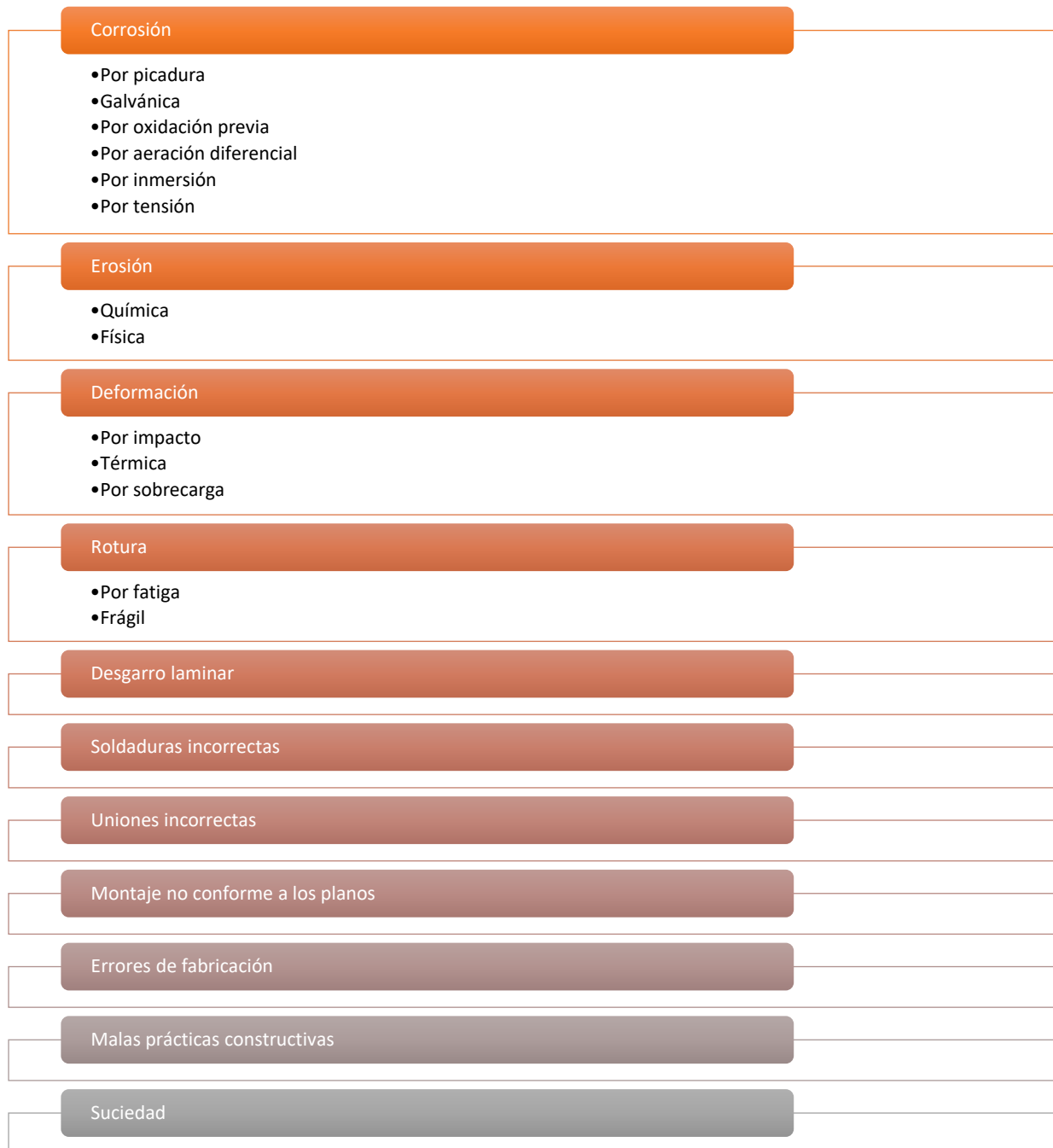


Gráfico 1. Patologías en estructuras de acero. (Jimenez, 2021)

2.3.3 Evaluación de las características para el análisis de patologías en acero

En esta evaluación de características por patologías, se analizan para cada patología que se asemeje al daño de la estructura. Algunas de estas características se pueden repetir en algunas patologías, van definidas en la Tabla 1:

Tipo de Patología	Características Principales	Equipo de Medición	Técnica para Reconocimiento	Tratamiento
Corrosión por picadura	<ul style="list-style-type: none"> - Pequeñas cavidades - Picaduras - Lágrimas de oxido - Pérdida de color - Puntos de oxido 	<ul style="list-style-type: none"> - Ultrasonido automático - Visual 	<ul style="list-style-type: none"> - Inspección por ultrasonido - Inspección visual detallada 	
Corrosión galvánica	<ul style="list-style-type: none"> - Dos materiales diferentes - Diferentes cargas eléctricas en los materiales - Uniformidad en la corrosión - Perdida de color 	<ul style="list-style-type: none"> - Ultrasonido automático - Visual 	<ul style="list-style-type: none"> - Inspección por ultrasonido - Inspección visual detallada 	Un tratamiento efectivo para este tipo de corrosión es la Galvanización.
Corrosión por oxidación previa	<ul style="list-style-type: none"> - Polvo de oxido - Oxido superficial - Uniformidad - Pérdida de color - Exposición a la intemperie 	<ul style="list-style-type: none"> - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada 	El tratamiento más adecuado para combatir la oxidación previa es la protección del elemento con pinturas protectoras. La limpieza previa es muy importante para este tratamiento.
Corrosión por aeración diferencial	<ul style="list-style-type: none"> - Solo partes específicas oxidadas - Oxido superficial - Pérdida de color - Exposición a la intemperie 	<ul style="list-style-type: none"> - Detector de Humedad - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada 	

Corrosión inmersión	por	<ul style="list-style-type: none"> - Capa de hidróxido - Carbonatación - Des uniformidad de oxido - Erosión - Presencia de agua o humedad 	<ul style="list-style-type: none"> - Detector de Humedad - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada 	La corrosión por inmersión a diferencia de las demás corrosiones, la presencia de agua o humedad es permanente por lo que la galvanización es la mejor solución Galvanización.
Corrosión tensión	por	<ul style="list-style-type: none"> - Presencia de fisuras - Des uniformidad de oxido - Esfuerzos a tensión - Costras 	<ul style="list-style-type: none"> - Ultrasonido automático - Visual 	<ul style="list-style-type: none"> - Inspección por ultrasonido - Inspección visual detallada 	La corrosión por tensión se puede tratar con diferentes métodos anticorrosivos, siempre y cuando estos supriman los factores que provocaron su inicio (tensión-fisuras).
Erosión química		<ul style="list-style-type: none"> - Desgaste del material - Reducción de secciones - Costras de oxido - Perforaciones 	<ul style="list-style-type: none"> - Flexómetro digital - Ultrasonido automático 	<ul style="list-style-type: none"> - Inspección por ultrasonido - Inspección visual detallada 	Ningún tratamiento anticorrosivo combate la erosión química, la mejor solución está en reemplazar por completo el elemento afectado, una vez realizado este procedimiento se puede aplicar pinturas protectoras para prevenir futuras afectaciones. Se debe analizar los factores que iniciaron una erosión de este tipo y erradicarlos por completo.
Suciedad		<ul style="list-style-type: none"> - Presencia de materia orgánica - Presencia de materia muerta - Capas de lodo - Exposición a la intemperie - Capas de polvo 	<ul style="list-style-type: none"> - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada 	Remoción de la capa de material orgánica e inorgánica, capa de polvo, lodo, plásticos. Se pueden aplicar las técnicas como limpieza con agua a presión.

Deformación por impacto	<ul style="list-style-type: none"> - Flechas excesivas - Pequeñas fisuras - Inicio de fatiga - Torsión 	<ul style="list-style-type: none"> - Flexómetro digital - Radiografía industrial 	<ul style="list-style-type: none"> - Inspección visual detallada - Radiografía industrial 	<p>Cuando esta deformación presenta flechas exageradas se debe reemplazar el elemento por completo. Siguiendo procedimientos de apuntalamiento estructural en todo momento hasta su reemplazo.</p>
Deformación térmica	<ul style="list-style-type: none"> - Flechas excesivas - Pérdida de color natural - Deformación general - Alabeos 	<ul style="list-style-type: none"> - Flexómetro digital - Radiografía industrial 	<ul style="list-style-type: none"> - Inspección visual detallada - Radiografía industrial 	<p>Un elemento deformado por factores térmicos debe ser reemplazado por completo ya que, pierde todas sus propiedades fisicoquímicas al momento de fluir.</p>
Deformación por sobrecarga	<ul style="list-style-type: none"> - Flechas excesivas - Fatiga - Pandeo local - Alabeos en los extremos - Torsión 	<ul style="list-style-type: none"> - Flexómetro digital - Radiografía industrial 	<ul style="list-style-type: none"> - Inspección visual detallada - Radiografía industrial 	<p>Un elemento deformado por sobrecarga generalmente debe ser reemplazado por uno de mayor resistencia.</p>
Desgarro laminar	<ul style="list-style-type: none"> - Fisuras paralelas al plano de soldadura - Perdida de color natural 	<ul style="list-style-type: none"> - Ultrasonido automático - Radiografía industrial - Líquidos penetrantes - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada - Radiografía industrial - Inspección por ultrasonido - Inspección por líquidos penetrantes 	<p>Reemplazo del elemento laminado (uniones) y proceder con métodos de soldadura calificados. Usualmente el desgarro laminar se produce por malas soldaduras, por lo que un buen procedimiento evitaría esta patología.</p>
Rotura por fatiga	<ul style="list-style-type: none"> - Perdida de sección - Fisuras profundas - Fatiga - Rotura del elemento 	<ul style="list-style-type: none"> - Ultrasonido automático - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada - Inspección por ultrasonido 	<p>Reemplazo del elemento afectado. Seguir un procedimiento que sea seguro para la estructura. Apuntar la zona afectada antes del reemplazo y no retirarlo hasta que la nueva pieza sea colocada.</p>
Rotura frágil	<ul style="list-style-type: none"> - Prolongación de fisuras - Fisuras profundas 	<ul style="list-style-type: none"> - Radiografía industria - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada - Radiografía industrial 	<p>Reemplazo del elemento afectado. Seguir un procedimiento que sea seguro para la</p>

		<ul style="list-style-type: none"> - Aceros fundidos - Rotura del elemento 			<p>estructura. Apuntar la zona afectada antes del reemplazo y no retirarlo hasta que la nueva pieza sea colocada.</p>
Erosión física		<ul style="list-style-type: none"> - Desgaste del material - Reducción de sección - Desprendimiento del material - Material carcomido 	<ul style="list-style-type: none"> - Flexómetro digital - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada 	<p>Los tratamientos termoquímicos son una muy buena alternativa para combatir la erosión física en elementos metálicos.</p>
Soldaduras incorrectas		<ul style="list-style-type: none"> - Fisuras - Burbujas internas - Falta de homogeneidad - Porosidad 	<ul style="list-style-type: none"> - Líquidos penetrantes - Radiografía industria - Ultrasonido automático 	<ul style="list-style-type: none"> - Inspección por líquidos penetrantes - Radiografía industrial - Inspección por ultrasonido 	
Uniones incorrectas		<ul style="list-style-type: none"> - Unión sin continuidad - Falta de soldadura 	<ul style="list-style-type: none"> - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada 	<p>Para este tipo de problemas, se recomienda tener una supervisión técnica y regirse a normativas constructivas desde el diseño para evitar problemas futuros.</p>
Montaje conforme a los planos	no a los	<ul style="list-style-type: none"> - Montaje erróneo - Mala posición de elementos 	<ul style="list-style-type: none"> - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada 	<p>Revisar las indicaciones propuestas dentro de los planos de diseño bajo la supervisión de personal calificado y rectificar errores. Si el modificar elementos mal posicionados implica un peligro a la estabilidad de la estructura, armar un diseño de apuntalamiento para dar seguridad durante la modificación del elemento.</p>

Errores de fabricación	<ul style="list-style-type: none"> - Erróneos espesores de solda - Abolladuras - Divergencia de espesores en el elemento - Biselados imperfectos - Divergencia en dimensiones 	<ul style="list-style-type: none"> - Flexómetro digital - Visual 	<ul style="list-style-type: none"> - Inspección visual detallada - Visual 	<p>Revisar el elemento de manera minuciosa y compararlo con las especificaciones técnicas del fabricante. En caso de que el elemento ya se encuentre ubicado dentro de la estructura, proceder a reemplazarlo por otro. Caso contrario, aplicar refuerzos metálicos, siempre y cuando el elemento soporte las cargas aplicadas.</p>
Malas prácticas constructivas	<ul style="list-style-type: none"> - Errores en normativas - Mala ejecución - Falta de criterio técnico - Procedimientos empíricos 	<ul style="list-style-type: none"> - Visual 	<ul style="list-style-type: none"> - Visual 	<p>Para evitar las malas prácticas constructivas se recomienda tener profesionales capacitados que supervisen las actividades en la construcción y rigiéndose a normas constructivas de cada país.</p>

Tabla 1: Descripción de patologías estructurales. (Jimenez, 2021)

2.3.4 Factores que intervienen en las patologías en estructuras de Hormigón

Las patologías en estructuras de hormigón que están contenidas en la base de datos de (Cordero, 2021), describen las fallas en las estructuras.

Una estructura de hormigón que está afectada por patologías, ocasiona una disminución de su resistencia, estabilidad y durabilidad en el tiempo. Por lo que, los factores involucrados en este análisis pueden ser de manera directa como la sobrecarga, temperaturas que rebasan la resistencia, la exposición a la humedad y la falta de mantenimiento.

2.3.5 Principales patologías en el hormigón

Los orígenes de las patologías pueden ser diversos, es posible establecer las clases específicas que contendrán las patologías, llegando a definir las siguientes para el hormigón

Gráfico 2:

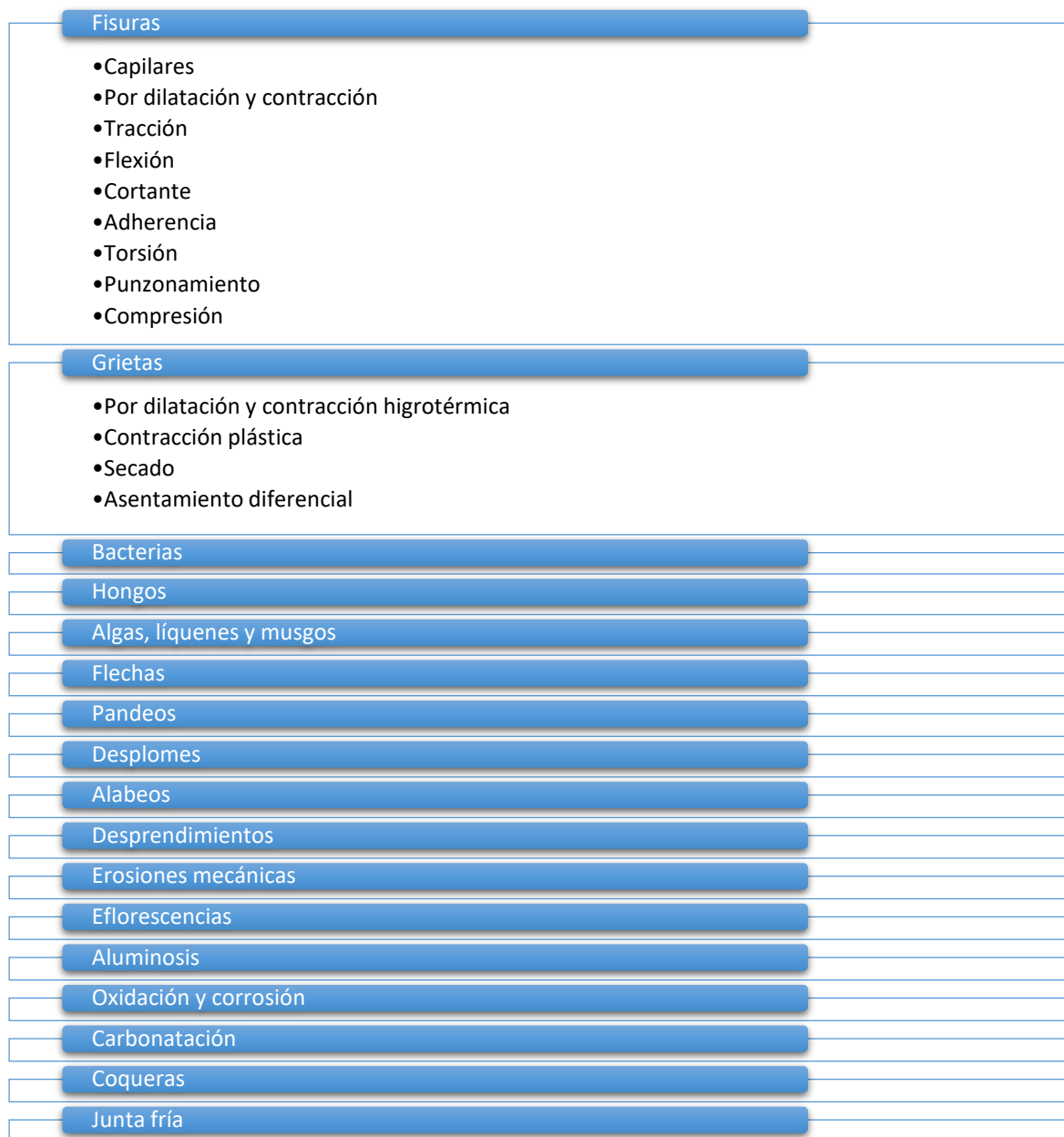


Gráfico 2. Patologías en estructuras de hormigón, (Cordero, 2021)

2.3.6 Evaluación de las características para el análisis de patologías en hormigón

En esta evaluación de características por patologías, es decir el nivel de daño y los tipos de patologías posibles que se asemejen al daño de la estructura. Algunas de estas características se pueden repetir en algunas patologías.

Las características que intervienen en el proceso corrosivo se observan en la Tabla 2.

Tipo de Patología	Características Principales	Equipo de Medición	Técnica para Reconocimiento	Tratamiento
Fisuras Capilares	<ul style="list-style-type: none"> - Fisuras delgadas. - Fisuras irregulares y no presentan un patrón determinado. 	<ul style="list-style-type: none"> - Ultrasonido automático - Visual - Humidimetro ensayo de humedad - Ultrasonido - Extensometría 	Inspección visual detallada	<p>Se puede utilizar la técnica de ocratización para fisuras menores a 2 mm.</p> <p>Se puede realizar la técnica de Inyección de fisuras y/o grietas con resina epóxica.</p>
Fisuras por dilatación y contracción.	<ul style="list-style-type: none"> - Fisuración irregular tipo piel de cocodrilo. - Patrón de mapeo y agrietamiento poco profundo. - Fisuras no mayores a 3 mm de profundidad. - Se desarrollan a edad temprana y pueden ser visibles al día siguiente del vaciado del hormigón. - Son fisuras antiestéticas. 	<ul style="list-style-type: none"> - Ultrasonido automático - Visual - Humidimetro ensayo de humedad - Ultrasonido - Extensometria 	Inspección visual detallada	Se recomienda realizar la técnica de Inyección de fisuras y/o grietas con resina epóxica o Inyección de fisuras y/o grietas con lechada o morteros con base en cemento
Bacterias	<ul style="list-style-type: none"> - Manchas en el hormigón. - Decoloración del hormigón. - Aumento de rugosidad de la superficie. - Presencia y desarrollo de colonias de cianobacterias en la superficie del hormigón. - Manchas color verde. 	<ul style="list-style-type: none"> - Visual - Humidimetro ensayo de humedad - Microscopio 	Inspección visual detallada Visual Examen de microscopio	<p>Cepillado con detergente de las superficies en donde se pueda observar colonias.</p> <p>Se recomienda la técnica de chorro de agua o granallado.</p> <p>Es muy poco usual el uso de azitromicina para estanques de hormigón debido a su costo, pero es una buena opción.</p>

Hongos	<ul style="list-style-type: none"> - Manchas de color oscuro (mohos). - Superficies más rugosas debido al desgaste producido en el hormigón. - Mayor porosidad de la superficie de contacto. 	<ul style="list-style-type: none"> - Visual - Humidímetro ensayo de humedad 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Limpieza de superficies por medio de hidrolavadoras.</p> <p>Se recomienda la técnica de chorro de agua o granallado</p> <p>Uso de protectores hidrorrepelentes</p>
Algas, líquenes y musgos	<ul style="list-style-type: none"> - Manchas de color verde. - Superficies resbalosas con mucha humedad y presencia de agua en época de invierno. - Superficies reseca y porosas en época de verano. - Se puede distinguir a simple vista el musgo y algas adheridas a la superficie del hormigón. 	<ul style="list-style-type: none"> - Microscopio - Visual 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Limpieza de superficies por medio de hidrolavadoras.</p> <p>Uso de detergentes y cepillado.</p> <p>Uso de protectores hidrorrepelentes.</p>
Flechas	<ul style="list-style-type: none"> - Se observa a simple vista los elementos estructurales horizontales y losas flectadas, siendo son más visibles estos defectos en los centros de los claros. - En etapas tempranas no se presentan fisuras hasta que los elementos comienzan a fallar. - Fisuras finas. - Fisuras verticales 	<ul style="list-style-type: none"> - Visual - Esclerómetro 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Se debe reforzar los elementos con recrecidos de hormigón por medio de la técnica hormigón o mortero proyectado.</p>
Pandeos	<ul style="list-style-type: none"> - Fisuras horizontales a las caras del elemento. - Desprendimiento de recubrimientos. 	<ul style="list-style-type: none"> - Visual - Esclerómetro 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Apuntalamiento urgente del elemento estructural.</p> <p>Aumento de recrecido de hormigón en el elemento estructural o refuerzo mediante la técnica de planchas de acero adheridas exteriormente.</p>
Desplomes	<ul style="list-style-type: none"> - Columnas desplazadas en su parte superior hacia la dirección del empuje. 	<ul style="list-style-type: none"> - Visual 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Apuntalamiento de las columnas</p>

					Reforzamiento con recrecido de hormigón técnicas de hormigón proyectado o planchas de acero adheridas exteriormente.
Alabeos		<ul style="list-style-type: none"> - Elementos estructurales girados hacia un lado. - Se observan fisuras a 45 grados, en vigas. estas fisuras nacen en la cara superior y se dirigen hacia la cara inferior. 	<ul style="list-style-type: none"> - Visual 	<p>Inspección visual detallada Visual</p>	Reforzamiento de los elementos estructurales con recrecidos de hormigón técnicas de hormigón proyectado o planchas de acero adheridas exteriormente
Grietas por dilatación y contracción higrótérmica		<ul style="list-style-type: none"> - Patrón de mapeo. - Fisuras transversales. - Fisuras largas en superficies en hormigones colocados en condiciones de viento y ambientes secos. 	<ul style="list-style-type: none"> - Visual - Humidímetro ensayo de humedad - Extensometría 	<p>Inspección visual detallada Visual</p>	Se recomienda realizar la técnica de Inyección de fisuras y/o grietas con resina epóxica o Inyección de fisuras y/o grietas con lechada o morteros con base en cemento
Grietas por contracción plástica		<ul style="list-style-type: none"> - Fisuras abiertas y horizontales en pilares y columnas. - Fisuras longitudinales abiertas en superficies horizontales. - Fisuras paralelas entre sí a distancias entre 30 y 90 cm. - Fisuras coincidentes con las barras de armado. 	<ul style="list-style-type: none"> - Visual - Humidímetro ensayo de humedad - Extensometría - Pull-out - Penetrometro 	<p>Inspección visual detallada Visual</p>	<p>Realizar pruebas de resistencia del hormigón.</p> <p>Si el hormigón no presenta disminución de resistencia realizar la técnica de Inyección de fisuras y/o grietas con resina epóxica o Inyección de fisuras y/o grietas con lechada o morteros con base en cemento</p>
Grietas por secado		<ul style="list-style-type: none"> - Grietas paralelas a corta distancia. - Grietas en dirección de las varillas de flexión en las losas. - Patrón de mapeo. - Grietas transversales. 	<ul style="list-style-type: none"> - Visual - Humidímetro ensayo de humedad - Extensómetro 	<p>Inspección visual detallada Visual</p>	El tratamiento para este tipo de fisuras se recomienda la inyección de resina epoxi.

Grietas asentamiento diferencial	por	<ul style="list-style-type: none"> - Grietas a 45 grados más abiertas en la zona más alejada del asentamiento. 	<ul style="list-style-type: none"> - Ultrasonido automático - Visual - Ultrasonido - Ondas magnéticas 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Recalzar la cimentación si es necesario, por medio de hormigón vaciado en sitio o también se puede realizar la técnica de hormigón preempacado.</p>
Fisuras tracción	por	<ul style="list-style-type: none"> - Fisuras transversales a la dirección del elemento estructural. - Seccionamiento horizontal del elemento estructural. - Si el elemento tiene exceso de refuerzo se presentan fisuras abiertas y finas en toda la longitud que se secciona. - Fisuras en uniones de columnas con mampostería. La fisura se presenta a lo largo de la columna pudiendo desprenderse la pared. 	<ul style="list-style-type: none"> - Visual - Ultrasonido - Ultrasonido automático - Pachometro-escaner - Ondas magnéticas - Pull-out 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Apuntalamiento del elemento, comprobación de resistencia en el hormigón.</p> <p>Reforzamiento de los elementos estructurales con recrecidos de hormigón técnicas de hormigón proyectado.</p>
Fisuras flexión	por	<ul style="list-style-type: none"> - La fisura empieza en la parte central del elemento estructural. - La fisura tiene una dirección vertical. - La fisura avanza hasta alcanzar el eje neutro del elemento estructural (vigas). - En columnas se presenta una fisura abierta en la cara traccionada y se cierra al alejarse de las misma. - En vigas en voladizo la fisura se genera en la parte superior del elemento en la unión de vigas o de viga-columna. - Generalmente se presentan este tipo de fisuras en estructuras aporticadas 	<ul style="list-style-type: none"> - Visual - Ultrasonido - Ultrasonido automático - Pachometro-escaner - Ondas magnéticas - Extensometria 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Apuntalamiento del elemento estructural.</p> <p>Reforzamiento de los elementos estructurales con recrecidos de hormigón técnicas de hormigón proyectado o planchas de acero adheridas exteriormente.</p>
Fisuras cortante	por	<ul style="list-style-type: none"> - Fisuras horizontales en dos caras del elemento y en las dos caras restantes se presentan fisuras a 45 o en algunos casos a 60 grados. 	<ul style="list-style-type: none"> - Visual - Ultrasonido - Ultrasonido automático - Pachometro-escaner 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Apuntalar el elemento estructural.</p> <p>Reforzamiento de los elementos estructurales con recrecidos de</p>

		<ul style="list-style-type: none"> - Desprendimiento de recubrimiento. - Las fisuras se presentan cerca de los apoyos de los elementos estructurales. 	<ul style="list-style-type: none"> - Ondas magnéticas - Extensometria 		hormigón técnicas de hormigón proyectado o planchas de acero adheridas exteriormente.
Fisuras por adherencia	por	<ul style="list-style-type: none"> - Son fisuras verticales. - Se generan desde el acero y recorren el hormigón por lo que se pierde el trabajo en conjunto de los dos materiales. - Las fisuras por adherencia también se presentan por acciones de carga, en este caso las fisuras tienen el mismo sentido de dirección de las varillas de amarre. - Corrosión de armaduras. 	<ul style="list-style-type: none"> - Visual - Ultrasonido - Ultrasonido automático - Esclerómetro 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Remover el hormigón afectado.</p> <p>Eliminación de la causa de la falla de adherencia, (corrosión, humedades, carga mediante apuntalamientos.) Realizar técnica de limpieza con chorro abrasivo</p>
Fisuras por torsión	por	<ul style="list-style-type: none"> - Fisuras a 45 grados en las caras de los elementos estructurales. - Se identifican estas fisuras ya que van en sentido contrario a las fisuras que se producen por cortante. - Tienen un aspecto similar a las del esfuerzo cortante y por tanto pueden ser confundidas con ellas, el principal criterio para distinguirlas es que en el caso del cortante las fisuras están inclinadas en el mismo sentido en las dos caras opuestas en tanto que las de torsión están inclinadas en sentidos contrarios. - Las fisuras se presentan en el perímetro del elemento estructural. - En las losas se presentan fisuras a 45 grados en los cantos de la misma. 	<ul style="list-style-type: none"> - Visual - Ultrasonido - Ultrasonido automático - Pachometro-escaner - Ondas magnéticas - Extensometria 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Apuntalamiento de los elementos estructurales.</p> <p>Compensación de cargas.</p> <p>Aumento de la sección de los elementos con recrecidos de hormigón.</p>
Fisuras por punzonamiento	por	<ul style="list-style-type: none"> - Fisuras finas, cerradas con apariencia inofensiva. - Fisuras con características muy similares a las fisuras por cortante. 	<ul style="list-style-type: none"> - Visual - Ultrasonido - Ultrasonido automático - Pachometro-escaner 	<p>Inspección visual detallada</p> <p>Visual</p>	Se recomienda realizar la técnica de anclajes de acero para reparación y refuerzo de la estructura y recrecido de hormigón por medio de

	<ul style="list-style-type: none"> - Fisuras a 45 grados en columnas y fisuras circulares alrededor de los pilares en las losas. En las losas las fisuras se presentan entre los 30 y 45 grados. - Pandeo del acero de refuerzo. - Agrietamientos en forma de cruz en muros de carga. - Desmoronamiento inclinado en vigas. 	<ul style="list-style-type: none"> - Ondas magnéticas - Extensometria - Esclerómetro - Penetrómetro 		hormigón proyectado para aumentar la sección de los elementos estructurales
Fisuras por compresión	<ul style="list-style-type: none"> - Este tipo de fisuras se presentan en dirección vertical. - Fisuras finas que dividen en dos al elemento estructural. - En ocasiones se presentan fisuras a 60 grados. - Generan desprendimiento de recubrimientos. 	<ul style="list-style-type: none"> - Visual - Ultrasonido - Ultrasonido automático - Extensometria - Esclerómetro - Penetrómetro 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Apuntalar el pilar de manera urgente.</p> <p>Reforzar el elemento por medio de hormigón proyectado o planchas de acero adheridas exteriormente</p>
Desprendimientos	<ul style="list-style-type: none"> - Desmoronamiento del hormigón. - Corrosión de las armaduras. - Hormigón de aspecto poroso. - Cambio de color del hormigón, color más claro u oscuro. 	<ul style="list-style-type: none"> - Visual - Termografía de infrarrojos - Detector de humedad - Humidimetro ensayo de humedad - Esclerómetro - Grado de corrosión 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Apuntalamiento</p> <p>Protección del hormigón si el ataque es exterior.</p> <p>Sustitución del elemento si el ataque es interior, se puede realizar las técnicas de limpieza con chorro abrasivo, raspado con agujas</p>
Erosiones mecánicas	<ul style="list-style-type: none"> - Fisuras abiertas por desgaste sin un patrón determinado. - Desprendimiento de capas. - Fisuras longitudinales. - Fisuras transversales. - Roturas en las esquinas. - Hundimientos de paños. 	<ul style="list-style-type: none"> - Visual - Esclerómetro 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Aislamiento del área deteriorada.</p> <p>Remoción del hormigón existente. técnica de rotomoldeo.</p>

Eflorescencias	<ul style="list-style-type: none"> - Manchas de color blanco o amarillo verdoso en la superficie de hormigón, mortero y materiales de construcción. - Desconche de revestimientos. - Decoloración permanente o temporal de la superficie. - Hinchazón y deformación de las superficies de revestimiento. - Capas pulverizadas en la superficie. 	<ul style="list-style-type: none"> - Visual - Humidimetro ensayo de humedad - Esclerómetro - Termografía de infrarrojos 	<p>Inspección visual detallada</p> <p>Visual</p>	<p>Para el tratamiento y prevención de las eflorescencias revisar Determinación y prevención de los niveles de eflorescencia. desde la pág. 16 a la pág. 22</p>
Aluminosis	<ul style="list-style-type: none"> - Porosidad en el hormigón. - Desprendimientos. - Pérdida de adherencia. - Pérdida de resistencia del hormigón. - Carbonatación. - Corrosión. 	<ul style="list-style-type: none"> - Visual - Ultrasonido automático - Pachometro-escaner - Esclerómetro - Penetrómetro - Termografía de infrarrojos - Grado de corrosión 	<p>Inspección visual detallada</p> <p>Visual</p> <p>Medición de diámetro de armaduras</p>	<p>Apuntalamiento del elemento.</p> <p>Sustitución del elemento afectado. Se puede realizar la técnica de Reemplazo de armaduras dañadas utilizando barras de traslape o anclajes</p>
Oxidación y corrosión	<ul style="list-style-type: none"> - Pérdida de sección de las armaduras. - Fisuración del hormigón siguiendo la dirección de las barras de armado y estribos. - Desprendimiento de las capas protectoras de la estructura. - Manchas color herrumbre en las superficies de contacto entre hormigón y acero de refuerzo 	<ul style="list-style-type: none"> - Visual - Pachometro-escaner - Esclerómetro - Grado de corrosión - Ondas magnéticas - Termografía de infrarrojos 	<p>Inspección visual detallada</p> <p>Visual</p> <p>Medición de diámetro de armaduras</p>	<p>Apuntalamiento de los elementos estructurales afectados.</p> <p>Sustitución de la armadura si está muy dañada. Técnica de Reemplazo de armaduras dañadas utilizando barras de traslape.</p>
Carbonatación	<ul style="list-style-type: none"> - Fisuras en el hormigón que empiezan en la unión de la armadura con el hormigón. - Desprendimiento de capas de protección de hormigón. - Hormigón poroso. 	<ul style="list-style-type: none"> - Visual - Humidimetro ensayo de humedad - Esclerómetro - Termografía de infrarrojos 	<p>Inspección visual detallada</p> <p>Visual</p> <p>Test de fenolftaleina</p>	<p>Sustitución de la armadura si está muy dañada. Técnica de Reemplazo de armaduras dañadas utilizando barras de traslape</p>

Coqueras	<ul style="list-style-type: none"> - Presencia de burbujas en las caras del hormigón. - El hormigón no es homogéneo. - Se puede observar los agregados del hormigón, muchas veces se observa piedras que producen taponamientos para el vertido del hormigón. - Se puede observar el acero de refuerzo. - Material segregado. - Se presentan agujeros en forma de nidos de abeja con o sin acero expuesto. 	<ul style="list-style-type: none"> - Visual - Ultrasonido - Ultrasonido automático 	Inspección visual detallada Visual	Limpieza de la superficie a tratar. Retirar todo el hormigón sucio. Protección de la armadura.
Junta fría	<ul style="list-style-type: none"> - Fisuras o grietas que toman la forma de la unión que se realiza con el hormigonado entre hormigones de diferentes edades. 	<ul style="list-style-type: none"> - Visual - Ultrasonido automático 	Inspección visual detallada Visual	Se recomienda la reparación por medio de inyección de resina epoxi, reparar o reforzar con mortero base cemento o láminas de carbono o metálicas.

Tabla 2. Descripción de patologías estructurales (Cordero, 2021)

2.4 Normas y códigos usadas en la valoración de las patologías

2.4.1 Códigos y normas de referencia.

- **AISC-360:** El código de diseño de estructuras de acero AISC-360 (American Institute of Steel Construction) recomiendan un plan de garantía de calidad que debe ser elaborado por un fiscalizador, este plan de garantía debe incluir los procesos de END que se rigen a los códigos AWS D1.1 y AWS D1.3 (Structural Welding Code Steel) según la aplicación requerida.
- **ASME-V:** Hay una sección específica del código que contiene requisitos y métodos para la aplicación de ensayos no destructivos (END), que son requisitos para otras Secciones del mismo código o referencias. Estos métodos de END están destinados a detectar imperfecciones superficiales e internas en los materiales, soldaduras, partes prefabricadas y otros componentes. Incluyen exámenes radiográficos, ultrasonido, líquidos penetrantes, partículas magnéticas, corriente de Foucault, inspección visual, prueba de fugas y emisiones acústicas.
- **AWS D1.1:** El código estructural AWS D1.1 menciona en el capítulo 6 del mismo las especificaciones para distintas técnicas No destructivas que deben cumplir las probetas a ensayar y contiene todos los requisitos para las calificaciones y responsabilidades del inspector, los criterios de aceptación para las discontinuidades y los procedimientos para el END.
- **NEC-SE-AC:** La norma NEC-SE-AC menciona las probetas que serán ensayadas y se regirán a las especificaciones ahí descritas. NEC-SE-AC de estructuras de acero menciona un plan de garantía de calidad (PGC) que debe incluir una técnica de control de calidad en estructuras metálicas mediante el uso de técnicas no destructivas.

- **NEC-SE-RE:** La norma NEC-SE-RE Menciona que ensayos no destructivos de los elementos del edificio pueden ser realizados para establecer de manera general el tipo, construcción y condición de los materiales.

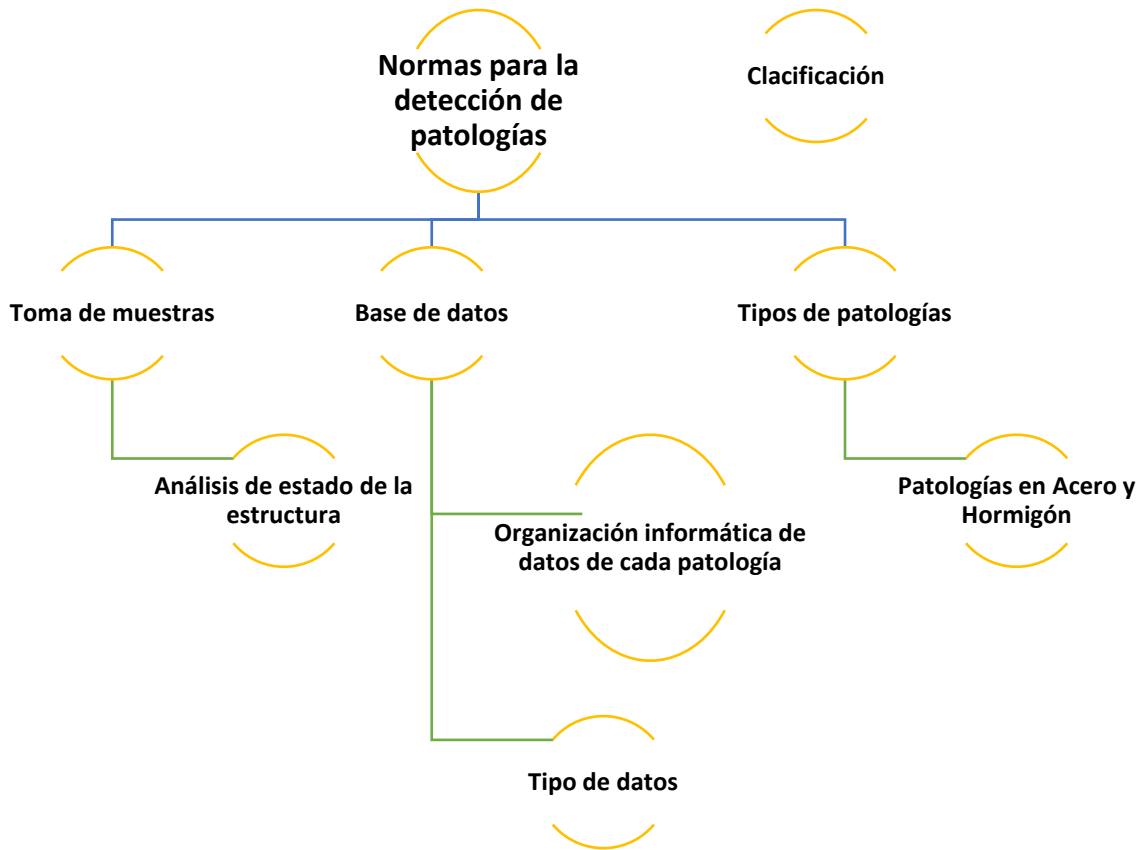


Gráfico 3. Normas para el estudio de las patologías en Acero y Hormigón. (Autores)

Se analizó y se definió que las patologías en estructuras de acero, se producen por factores diferentes como la humedad constante, la suciedad, golpes por impactos, el exceso de temperatura y la sobrecarga.

De igual manera para el hormigón, las patologías con ocasionadas por los agentes como la sobrecarga, factores de humedad, temperatura y una precaria dosificación y curado del hormigón.

2.5 Análisis del medio de Patologías Estructurales

2.5.1 Obtención de muestras

Básicamente, abarca toda la información que se puede encontrar en la documentación bibliográfica y la recopilación fotográfica de campo.

2.5.2 Captura de imágenes

Las imágenes que se capturan mediante la cámara de los dispositivos electrónicos, son in situ. Estas deben mostrar en mayor detalle la patología estructural, tomando en cuenta la luminosidad para conseguir una imagen clara. Posteriormente, son ordenadas en la base de datos; donde, se proporciona información como la procedencia, el tipo de falla a la que pertenece y el posible tratamiento que se puede dar a esta.

2.5.3 Base de datos

La base de datos que abarca los tipos de patologías en los diferentes materiales usados para la construcción, se desarrollará en la presente investigación, este como tal, parte de un proyecto en el que se proyecta generar una amplia base de datos sobre patologías estructurales. En dicha base de datos, se podrá encontrar patologías que afecten a las estructuras de acero (Jimenez, 2021), así como las patologías en el hormigón (Cordero, 2021).

El proyecto general antes mencionado, describe sobre la creación de una base de datos sobre patologías en estructuras de obra civil y arquitectónicas, aparte de su material de construcción, permitirá el desarrollo de la presente investigación, en el reconocimiento de patologías mediante procesamiento de imágenes, que se servirán del banco de imágenes generadas, para cumplir con los objetivos.

2.5.4 Entrenamiento automático y la Inteligencia Artificial

La creciente demanda en el uso de las aplicaciones para el tratamiento de información de manera automática, hacen que sean más frecuentes los sistemas de aprendizaje automático y la inteligencia artificial en la vida cotidiana de las personas (Jonggi Hong, 2019).

Con el propósito de hacer más amigable la tecnología y que, el usuario inexperto explore la enseñanza automática, se crean aplicaciones web móviles con información dinámica como se muestra en la Ilustración 1. Así es como operan los dispositivos tecnológicos para el reconocimiento de objetos por imágenes enseñables.

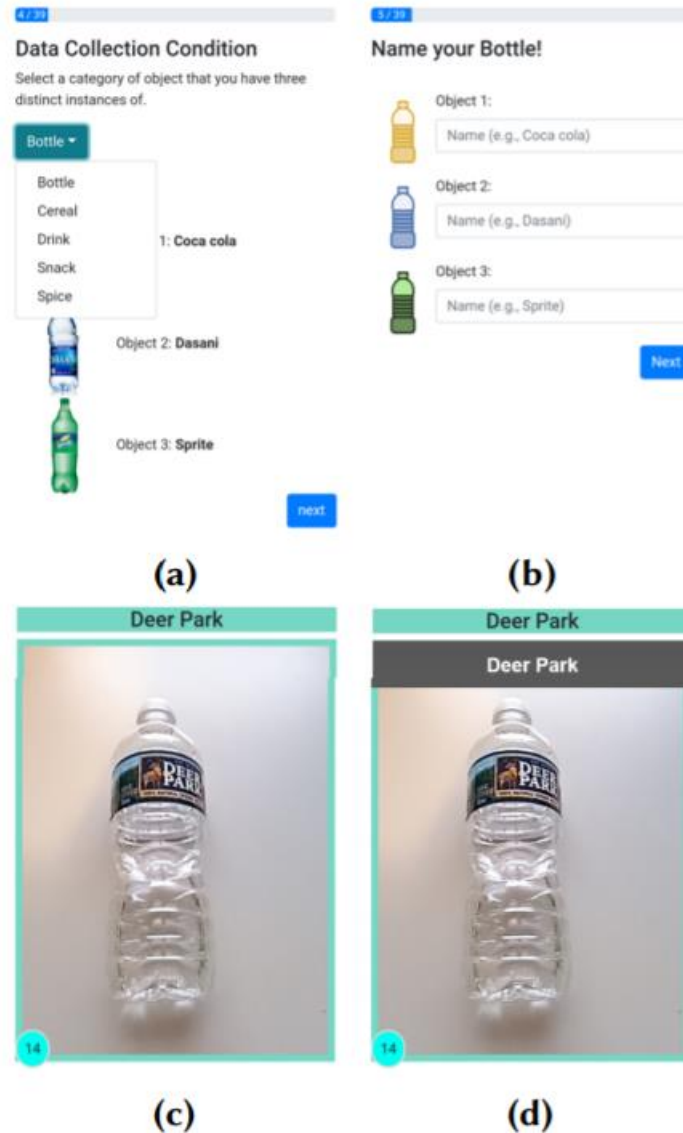


Ilustración 1. Modelo de interacción entre la aplicación y el usuario (Jonggi Hong, 2019).

ESTADO DEL ARTE

2.6 Visión Artificial

Por su nombre se puede interpretar, que tiene el objetivo de intentar imitar el proceso y la función del sistema de visión humano. La acción de diferenciar los objetos de un fondo, se asemeja a las acciones que las personas realizan en su cotidiano vivir. Es cierto que lleva

muchos años de investigación y análisis tratar que un ordenador o dispositivo sea capaz de realizar estas acciones de diferenciación, y al hacerlo que lo haga de una manera razonablemente aceptable (Platero, 2013).

El mundo tridimensional que se quiere lograr deducir con la visión artificial, contiene estructuras y propiedades variadas; no solo sus propiedades geométricas, sino también las propiedades materiales. Por sus propiedades geométricas se tiene la forma, tamaño, ubicación de objetos. Y por sus propiedades materiales se puede distinguir el color, la iluminación, textura y composición. Si en el entorno los objetos se modifican en su formación, también es necesario el cambio y actualización de los sistemas de reconocimiento visual (Malpartida, 2003).

En los últimos años, se han obtenido importantes avances en el campo de la visión por computador. Las redes neuronales profundas y puntualmente las redes neuronales convolucionales han marcado un salto en la ciencia de la inteligencia artificial. El reconocimiento de caracteres escritos, reconocimiento facial, de las señales de tráfico, la aplicación en los vehículos autónomos, entre otros, son algunos de los ejemplos que se han alcanzado (Jarrín, 2021).

La visión por computador o también llamada visión artificial (Platero, 2013), para que sea reconocida hoy en día, ha recorrido un largo camino. La mayor ventaja es la objetividad al ser empleada por un computador, ya que, al hacer una comparación con la visión humana, esta sufre limitaciones en la percepción visual (Heras, 2017). La visión artificial está simplificando esos tediosos y subjetivos procesos de control de calidad en las empresas.

La clasificación de las imágenes se la realiza mediante métodos de extracción de características, para luego por medio de un algoritmo de clasificación identificar las

características extraídas. Es importante que, antes de aplicar la metodología se realice un preproceso de la imagen a ser analizada, queda entonces normalizar los efectos de contraste y brillo además eliminar el ruido ver Ilustración 2.

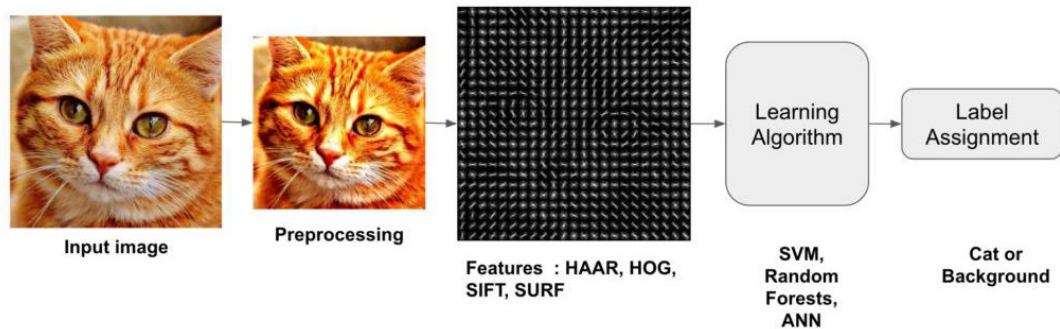


Ilustración 2. Esquema de clasificación de imagen (Mallick, 2016).

2.6.1 Aplicaciones de la visión artificial

La visión artificial cubre un gran campo de aplicaciones, en términos sencillos, permite la extracción y el análisis de información espectral (frecuencia del color e intensidad de tonos de gris), espacial (forma y posiciones) y temporal de los distintos objetos (presencia o ausencia de objetos) y dependientes del tiempo (Malpartida, 2003).

Las aplicaciones de la visión artificial son variables, citando algunas como aparecen en la Ilustración 3 e Ilustración 4. La primera, un encefalograma del cerebro de una persona y mediante el procesamiento de imágenes se logra realizar un análisis de fracturas y anomalías. Y la segunda, fallas que se presentan en una estructura, donde existen fisuras y grietas que ocasionan daños en las construcciones de obra civil.

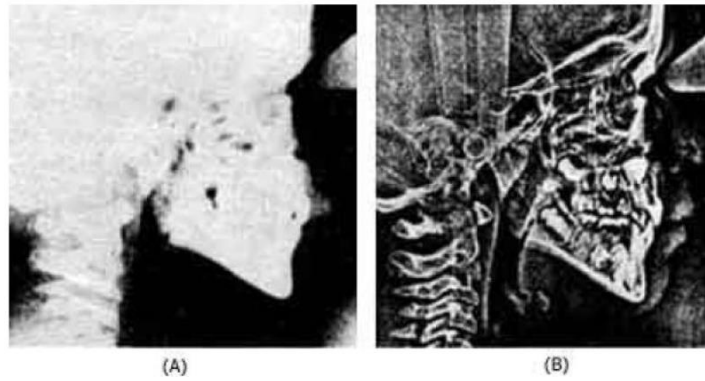


Ilustración 3. Encefalograma: (A) Imagen original, (B) Imagen Procesada (*Malpartida, 2003*).

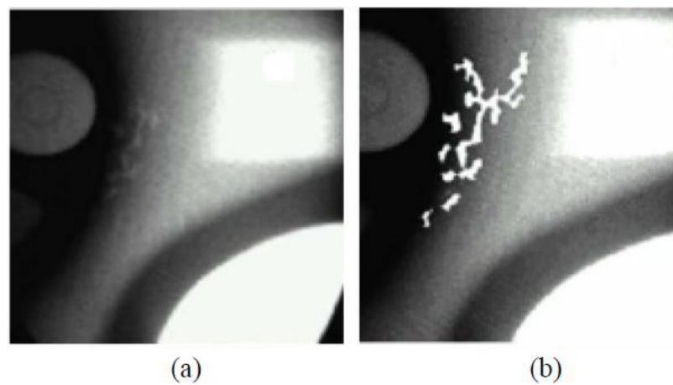


Ilustración 4. Falla estructural: (a) Imagen original, (b) Imagen Procesada (*Malpartida, 2003*).

2.6.2 Etapas para el sistema de visión artificial

Es sabido que, el ser humano y los seres vivos en general posean la capacidad de capturar la luz a través de los ojos, generando una información que al pasar por el nervio óptico hace que el cerebro la procese. Por lo tanto, la visión artificial puede llegar a imitar el funcionamiento de esta actividad y así tratar de definir las fases del análisis de las imágenes (J. F. Vélez Serrano, 2003), ver Gráfico 4. Adicionalmente, se puede ver las etapas en la Ilustración 5.

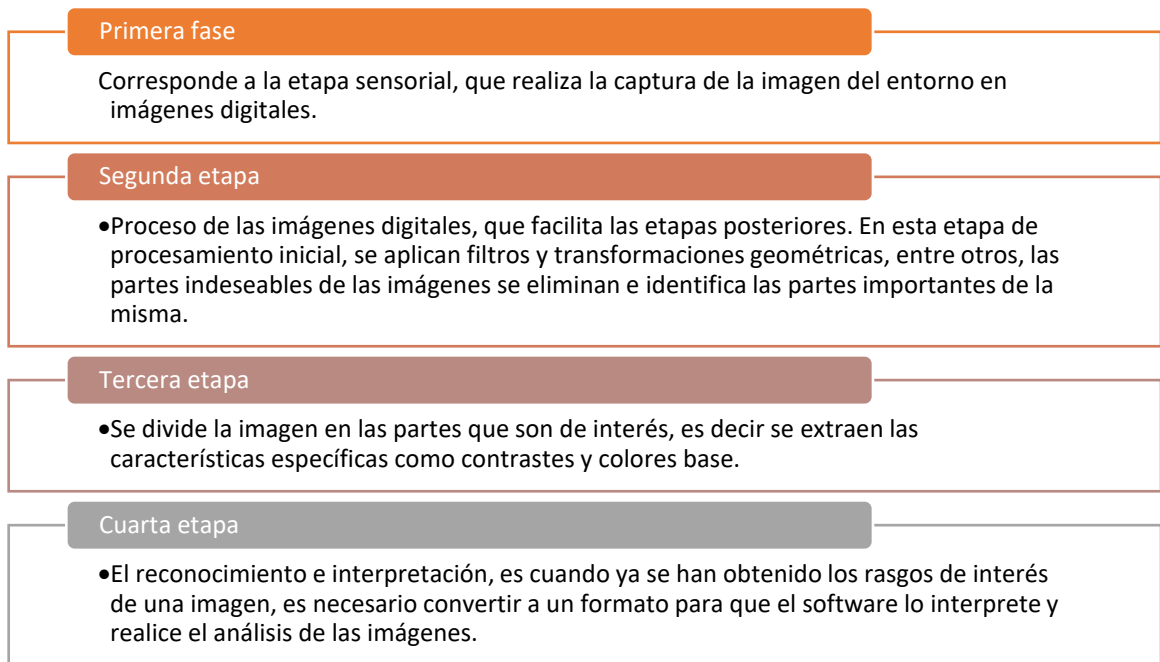


Gráfico 4. Fases del análisis de imágenes (Autores)

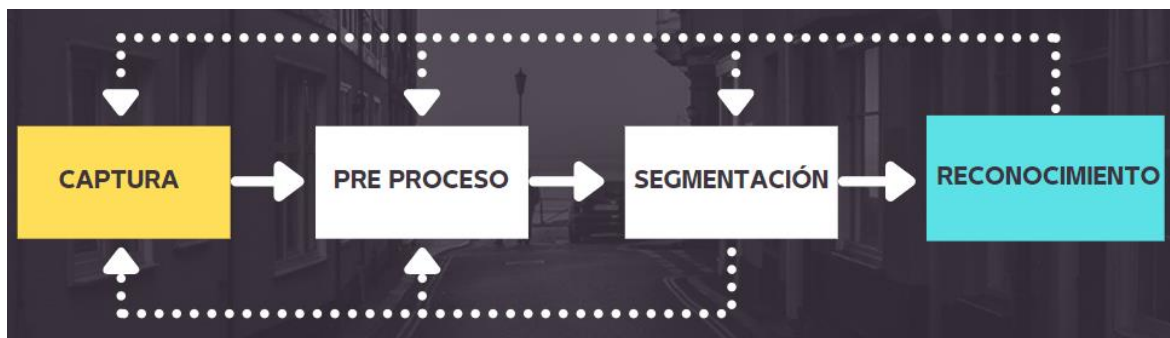


Ilustración 5. Etapas del sistema de visión artificial (Autores).

2.6.3 Fundamentos de la visión artificial

Para lograr el reconocimiento automático de las imágenes es necesario plantear el siguiente proceso, se inicia con la adquisición de la imagen del objetivo de análisis; a continuación, la imagen se procesa hasta lograr una mejor calidad y eliminar los desperfectos; como siguiente paso, se separa el objeto de interés del fondo de la imagen; luego se extraen

las características (color, textura y geometría); por último, comparar las características entre los objetos que se tienen en la base de datos y así determinar el tipo de objeto.

En la Ilustración 6, se ordenan las etapas fundamentales del procesamiento de imágenes.

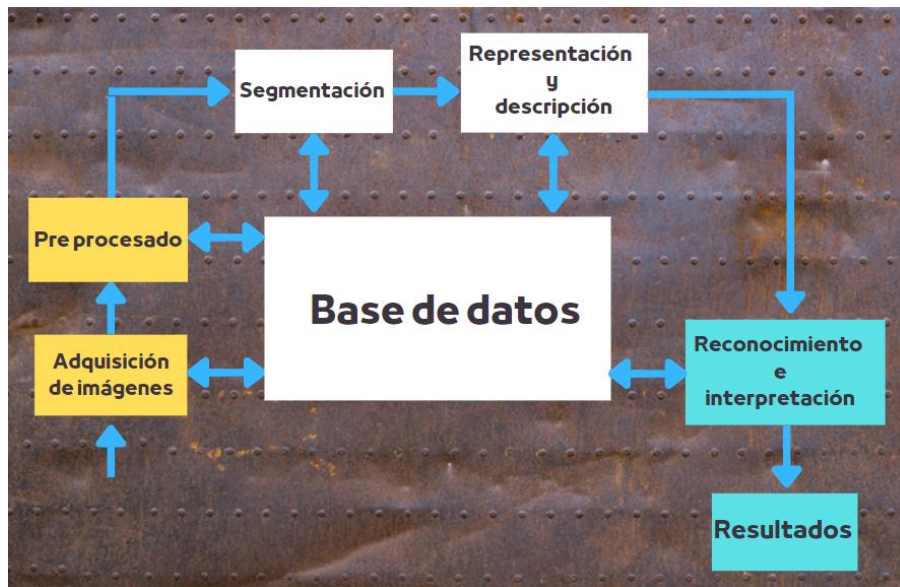


Ilustración 6. Etapas fundamentales para el procesamiento de imágenes. (Autores)

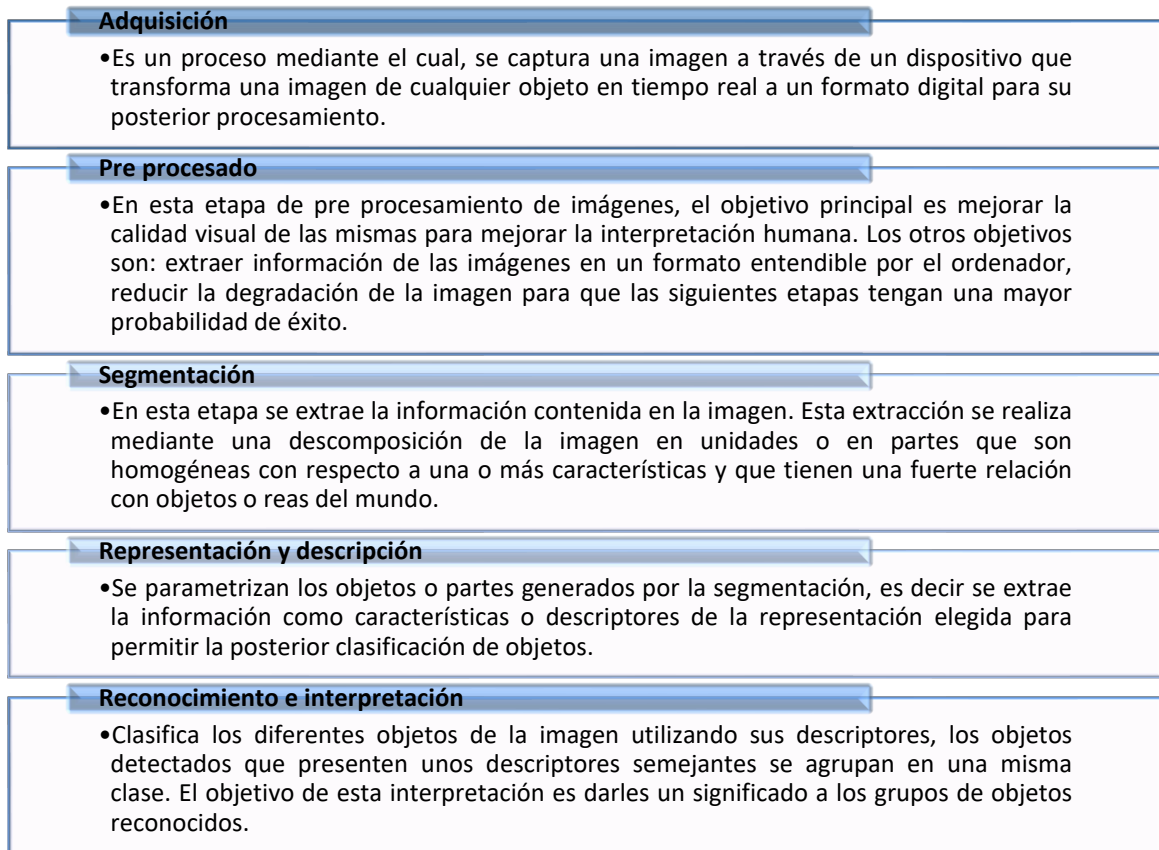


Gráfico 5. Procesamiento de imágenes, (García, 2008)

En cada una de las etapas anteriores del Gráfico 5, se requiere de un conocimiento previo como puede ser: el tamaño del objeto, la distancia a la que se encuentra, las condiciones ambientales, la inclinación, el número de objetos a reconocer en la imagen, entre otros. Toda esta información es necesaria para la utilización de determinadas técnicas y el desarrollo de algoritmos adicionales para el reconocimiento del objeto. (García, 2008)

2.7 Hardware y Software en el procesamiento de imágenes

La finalidad de la visión artificial es la percepción del mundo físico mediante la extracción de la información en imágenes y con el uso de un computador, abarcando de manera cuantificada los detalles del mundo real como: el brillo, el color, la forma que se pueden extraer de imágenes estáticas (J. F. Vélez Serrano, 2003).

Al pasar de los años, los fundamentos estudiados han permitido dar forma a la visión artificial y gracias a los avances en computadores con mayor capacidad de procesamiento se pueden obtener mejores resultados.

Procesar las imágenes de manera digital es un principio de la visión artificial, en la cual se expone a una imagen para que sea analizada y así se extraigan las características de ella, como: color, forma, textura, tamaño, entre otros. Cumpliendo así, la finalidad de implementar un algoritmo (Parker, 2010) ejecutado en un computador para esta función.

2.7.1 Elementos para el procesamiento

Hay un enorme mercado para hardware y software, con lo cual en la actualidad hay una gran variedad de sistemas para el procesamiento de las imágenes, los cuales, por sus características de oferta pueden variar en la capacidad de procesamiento, la velocidad y el costo. Este último, es muy importante a la hora de la implementación, ya que presenta las mayores limitantes a la hora de seleccionar las herramientas de hardware y software para realizar el proyecto (Yan, 2014).

Esta investigación se basa principalmente, en efectuar una herramienta de bajo costo para el procesamiento de las imágenes en la determinación de patologías estructurales que se pueden encontrar en las obras civiles, con el fin de clasificar en el tipo de material y el tipo de patología encontrada.

2.8 Hardware

Para la programación de las aplicaciones, y ya que se trata de adaptarlos a los sistemas operativos Android y iOS, es necesario el uso del computador que se detalla en la Tabla 3:

MACBOOK PRO 13”

PANTALLA	13,3 pulgadas, 2.560 x 1.600 píxeles (227 ppp) 500 nits, True Tone
PROCESADOR	Chip M1 de Apple Ocho núcleos (4 x alto rendimiento, 4 x eficiencia) Neural engine de 16 núcleos
GPU	Ocho núcleos
MEMORIA RAM	16 GB
ALMACENAMIENTO	SSD de 2 TB
PUERTOS	2 x Thunderbolt / USB 4 (USB-C) Conector de auriculares
TECLADO	Magic Keyboard, Touch Bar, Sensor Touch ID Sensor de luz ambiental
CONECTIVIDAD	Wi-Fi 6, Bluetooth 5.0
WEBCAM	FaceTime HD 720p
BATERÍA	Hasta 20 horas de reproducción de vídeo
DIMENSIONES	304,1 x 212,4 x 15,6 mm
PESO	1,4 kg
SISTEMA OPERATIVO	macOS Big Sur

Tabla 3. Especificaciones técnicas del computador (Autores)

2.9 Software

2.9.1 iOS de Apple

El sistema operativo iOS es propio de la empresa Apple Inc. Este sistema operativo fue desarrollado específicamente para iPhone (anteriormente denominado iPhone OS), y compatible con el tiempo a todos los dispositivos iPod Touch e iPad (Sepulveda, 2013).

La interfaz de usuario iOS está basada en una manipulación directa, usando gestos multitáctiles. Los elementos de control están compuestos por deslizadores, interruptores y botones, permitiendo una respuesta inmediata a las órdenes del usuario (Sepulveda, 2013).

En conjunto, el SDK de iOS, Xcode IDE y Swift by Coding dan facilidad para que los desarrolladores puedan crear aplicaciones móviles revolucionarias (Sepulveda, 2013).

La última versión del sistema operativo iOS fue lanzada en septiembre del 2020 con el iOS 14, la cual sustituye al iOS 13, con la finalidad de mejorar las expectativas del usuario, integrando cambios relevantes. IOS de deriva de MacOS que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo tipo Unix.

iOS cuenta con cuatro capas de abstracción: la capa del núcleo del sistema operativo, la capa de “Servicios Principales”, la capa de “Medios de comunicación” y la capa de “Cocoa Touch”, como se puede organizar en la Ilustración 7.

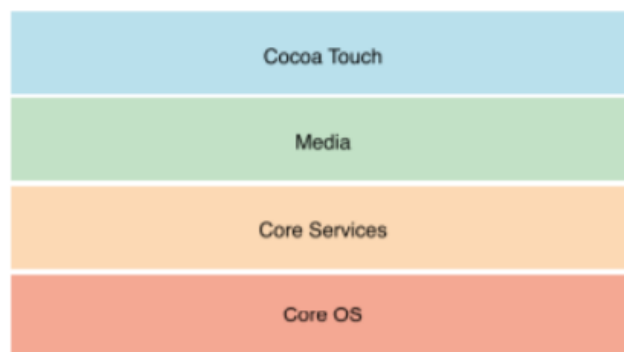


Ilustración 7. Arquitectura de capas iOS (*Arquitectura iOS - Tecnología iOS., 2021*).

Las características que presenta iOS con respecto a las características y descripción de la accesibilidad se detalla en la Tabla 4.

- **Características de accesibilidad de iOS**

Categoría	Características	Descripción
Visión	VoiceOver	Permite, a través de respuesta hablada, saber lo que está en el área de toque.
	SpeakScreen	Lee el contenido de la pantalla de aplicación actual.
	Siri	Siri es un asistente de voz, que recibe e interpreta un amplio número de comandos para realizar un variado conjunto de tareas.
	Dictado	Permite al usuario realizar entrada de texto a través de voz.
	Zoom	Funciona de manera similar a una lupa, permite ampliar el contenido de la pantalla a través de gestos táctiles.
	Escalado de fuente	Permite ajustar el tamaño del texto a las necesidades visuales del usuario.
	Alto contraste	Activa un modo en el que la pantalla utiliza colores de alto contraste.
	Colores invertidos	Permite invertir los colores (como en el caso de un negativo fotográfico) de la pantalla.
Audición	Soporte pantallas Braille	Permite utilizar pantallas Braille que le dan al usuario una respuesta basada en dicho código.
	Subtítulos	Permite agregar y visualizar pistas de subtítulos en contenidos multimedia.
	Alertas visibles y vibratorias	Permite al usuario estar al pendiente de alertas y mensajes a través de luces de notificación y vibración.
Locomoción	AssistiveTouch	Permite configurar múltiples gestos multitoque, según las necesidades del usuario.
	Switch Control	Permite navegar a través de las pantallas, por medio de los botones físicos del teléfono.
Otros	Diccionario	Muestra definiciones de términos que aparecen en la pantalla y son desconocidos para el usuario.
	API de Accesibilidad	Proporciona un conjunto de librerías para que los desarrolladores creen aplicaciones que aprovechen las características de accesibilidad del teléfono.

Tabla 4. Características de accesibilidad de iOS (Sepulveda, 2013).

2.9.2 Swift

Es un lenguaje de programación potente e intuitivo para iOS, iPadOS, macOS, tvOS y watchOS. Apareció el 2 de junio de 2014 como una creación de Apple que se enfoca en el desarrollo de aplicaciones (Owen, 2014).

Redactar un código Swift es interactivo y divertido, la sintaxis es concisa pero expresiva como se muestra en la Ilustración 8, Swift incluye también, características innovadoras que encantan a los usuarios. Como garantía el código Swift es seguro por su diseño, pero también como resultado produce software que se ejecuta en alta velocidad (Swift - Apple Developer, 2021).

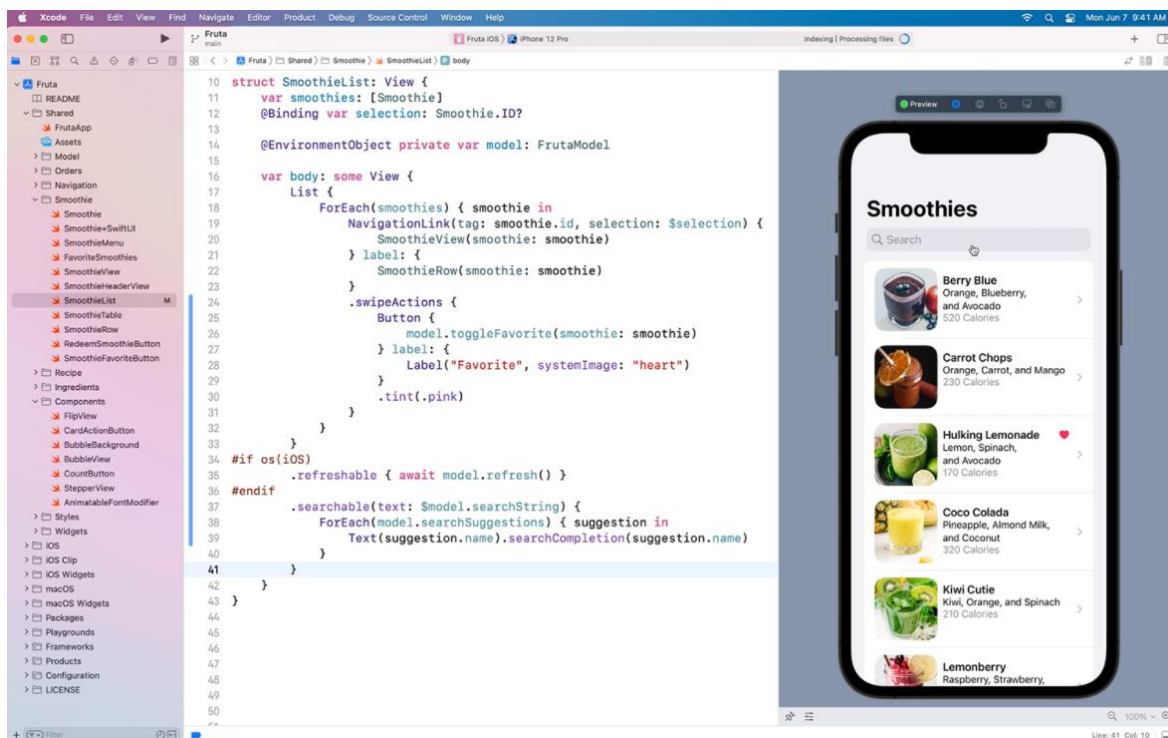


Ilustración 8. Modelo interactivo de Swift (Swift - Apple Developer, 2021).

El objetivo del proyecto Swift es crear el mejor lenguaje utilizable para usos que van desde la programación de sistemas hasta aplicaciones móviles y de escritorio, que alcanza a servicios en la nube. Swift está diseñado para facilitar la escritura y el mantenimiento

de programas correctos para el desarrollador (Inc, 2021). Para lograr este objetivo, la manera de escribir los códigos en Swift también debe ser como en el Gráfico 6:

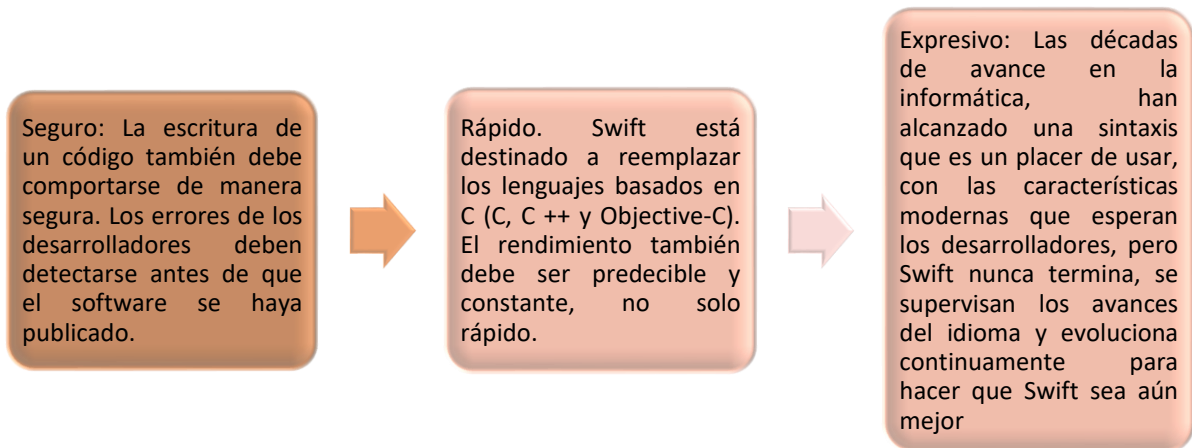


Gráfico 6. Condiciones del sistema Swift, (Inc, 2021)

2.9.3 Android

Es un sistema operativo creado para dispositivos móviles. Desarrollado por la Open Handset Alliance, la cual la dirige Google, usando diversos conjuntos de herramientas de software de código abierto para dispositivos electrónicos (Lara Cancela, 2012).

Android es un sistema operativo con una plataforma abierta para dispositivos móviles adquirido por Google y la Open Handset Alliance, Su finalidad, es satisfacer la necesidad de los operadores móviles y fabricantes de dispositivos; además, de fomentar el desarrollo de aplicaciones, cualidad que ningún otro sistema operativo incluye en sus conceptos (Kristel Malave Polanco, 2011).

Android bajo la definición de Google se considera un “software stack” o una pila de software, ya que está conformada como lo expresa en el Gráfico 7:

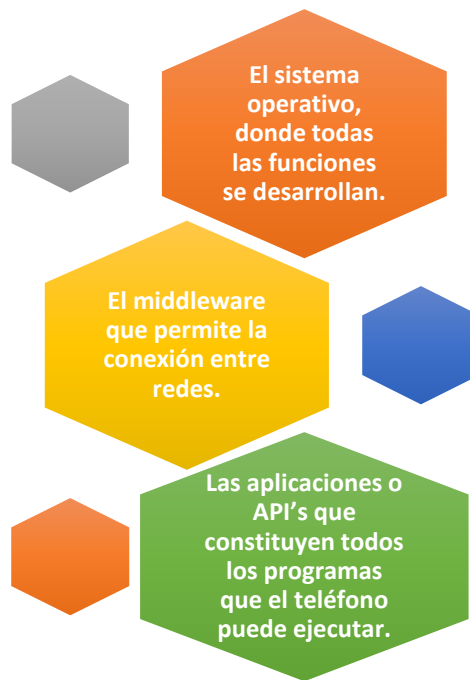


Gráfico 7. Definición de software stack, (Kristel Malave Polanco, 2011)

Su construcción tuvo como objetivo, que los desarrolladores de las aplicaciones aprovecharan al máximo las herramientas que un dispositivo.

Implementa una arquitectura en la que cualquier aplicación puede obtener acceso a las capacidades del teléfono móvil. Por ejemplo, una aplicación puede llamar una o varias de las funcionalidades básicas de los dispositivos móviles, tales como realizar llamadas, enviar mensajes de texto, o utilizar la cámara, facilitando a los desarrolladores crear experiencias más ricas y con más coherencia para los usuarios (Yeicy Juliana Molina Rivera, 2012).

Está construido sobre el kernel de Linux. Además, se utiliza una máquina personalizada virtual que fue diseñada para optimizarlos recursos de memoria y de hardware en un entorno móvil. Android es de código abierto, y además puede ser libremente ampliado para incorporar nuevas tecnologías de vanguardia que van surgiendo. La plataforma continuará evolucionando a medida que la comunidad de desarrolladores trabajando juntos puedan crear aplicaciones móviles innovadoras (Open handset alliance, 2012).

Dotada de una arquitectura interna en su plataforma Android, que engloba 4 componentes: las aplicaciones, almacén de aplicaciones, librerías y Kernel/Linux.

2.9.4 Kotlin

Es un lenguaje de programación muy utilizado por desarrolladores de Android de todo el mundo.

En Google I/O 2019, el desarrollo de Android sería cada vez más importante para Kotlin. Éste es un lenguaje de programación expresivo y conciso que permite reducir los errores de código comunes y se integra fácilmente en las apps existentes (Google, 2021).

Para apoyar el desarrollo de Android mediante Kotlin, se realiza la cofundación con Kotlin Foundation y la inversión económica con el objetivo de mejorar el rendimiento del compilador y la velocidad de compilación (Google, 2021).

Lo que los desarrolladores de aplicaciones valoran de Kotlin se resume en el Gráfico 8:



Gráfico 8. Detalles de Kotlin, (Google, 2021)

2.9.5 Teachable Machine

Es una herramienta GUI o también conocidas como interfaces gráficas de usuario o interfaces de usuario (Borja, 2008) basada en la web, para crear modelos personalizados para clasificar caracteres digitales sin una técnica de programación especializada. Su funcionamiento interpreta el aprendizaje automático, o ML que se define como un lenguaje de programación impuro (Cumming, 2005).

Teachable Machine fue creada para ayudar a estudiantes, diseñadores, profesores, entre otros, a aprender sobre el aprendizaje automático, al usar modelos de clasificación

creados por ellos mismos, al tener a su disposición tutoriales, manuales, entre otros recursos, aborda la Inteligencia Artificial en experimentos creativos (Michelle Carney, 2020).

Las personas están rodeadas de muchas facultades intelectuales para aprender nuevos procesos o disciplinas, así como también la falta de una comprensión para disciplinas particulares, Teachable Machine cuenta con numerosos casos de programación de algoritmos para ayudar al aprendizaje automático que se extienden a la vida cotidiana.

El aprendizaje por transferencia es muy útil, convirtiéndose en la técnica para el aprendizaje automático, cuando se requiere encontrar patrones y tendencias dentro de las imágenes y crear un modelo de clasificación fácil y simple en pocos segundos. Con el aprendizaje por transferencia, el usuario tiene la opción de ir añadiendo sus propios datos y reentrenar un modelo que ya fue programado (Michelle Carney, 2020).

Por ejemplo, el modelo base (mobilenet) para el apartado de imágenes en Teachable Machine, se preparó originalmente para reconocer 1000 clases distintas entre ellas (como perro, teléfono, cama, guitarra, entre otras). Los rasgos subyacentes que mobilenet usó para reconocer esas clases se pueden usar para identificar nuevas clases que el usuario defina. Esta complejidad obviamente está oculta a los usuarios, que simplemente se benefician ya que no necesitan demasiados datos y tiempo de capacitación para crear modelos que sean útiles y concretos (Michelle Carney, 2020).

El aprendizaje automático interactivo o IML (Rebecca Fiebrink, 2018), realiza este proceso al responder a las peticiones, registrando los datos de entrada del usuario para desarrollar y perfeccionar los modelos de aprendizaje automático, aprovechando los datos que los usuarios han introducido y para así tener la experiencia de resultados.

Modelo de entrenamiento para patologías: Teachable Machine dispone de una interfaz accesible, con muchas funciones para que todos los usuarios creen sus propios

modelos de clasificación de aprendizaje automático (Teachable Machine, 2021). Permite a los usuarios entrenar clasificadores para un número ilimitado de clases, que otorga la recopilación de datos junto con la respectiva clasificación, entrenamiento de modelos y la evaluación en la misma interfaz, y entrena en el dispositivo (lo que da como resultado un rendimiento más rápido y permite que el entrenamiento sea gratuito).

Para hacer un modelo de clasificación de aprendizaje automático que incluyen la (recopilación y clasificación de datos, preparación y evaluación) se presentan de izquierda a derecha para indicar la relación causal y orden de pasos para nuevos usuarios como en la Ilustración 9.

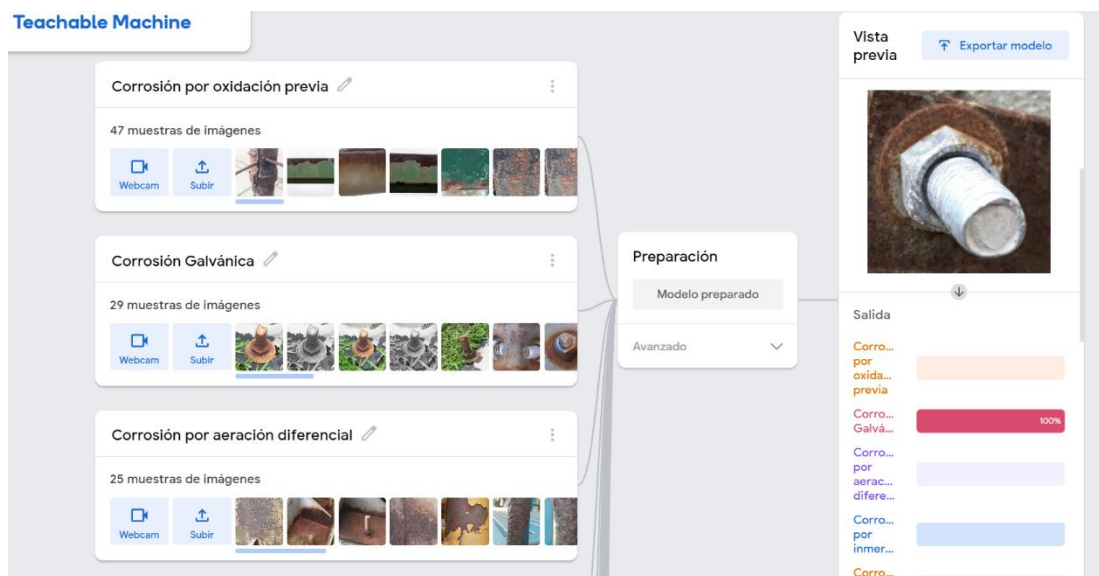


Ilustración 9. Interface de Teachable Machine (Teachable Machine, 2021).

Para crear un modelo, el usuario primero selecciona el modelo para la clasificación de imágenes como entrada, creando las clases que quieren que el modelo aprenda a clasificar como en la Ilustración 10.

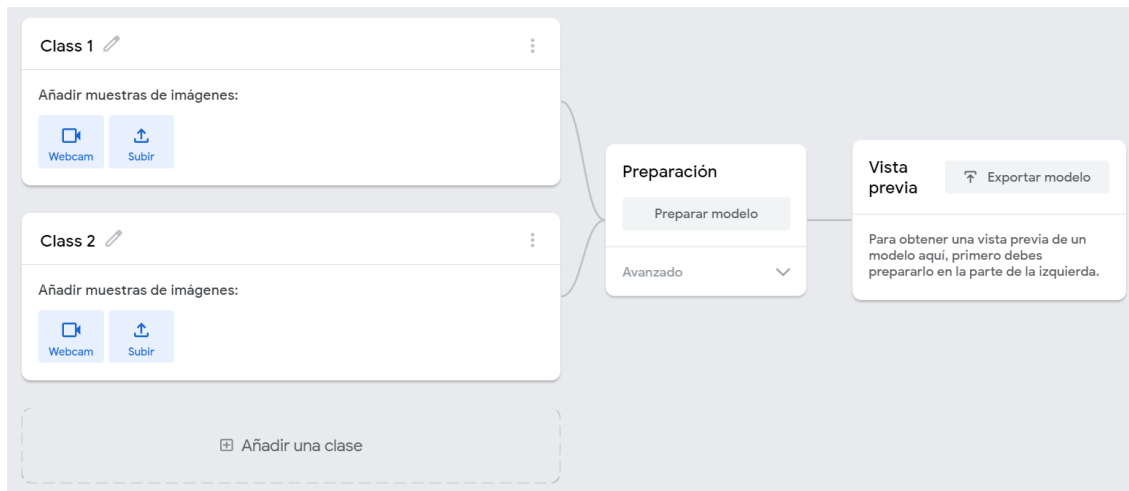


Ilustración 10. Introducción de tipos de clases (*Teachable Machine, 2021*).

A continuación, se suben las imágenes de las diferentes clases para preparar el modelo, y finalmente se prueba el modelo al comparar imágenes que tengan con las clases creadas anteriormente, como en Ilustración 11 e Ilustración 12.

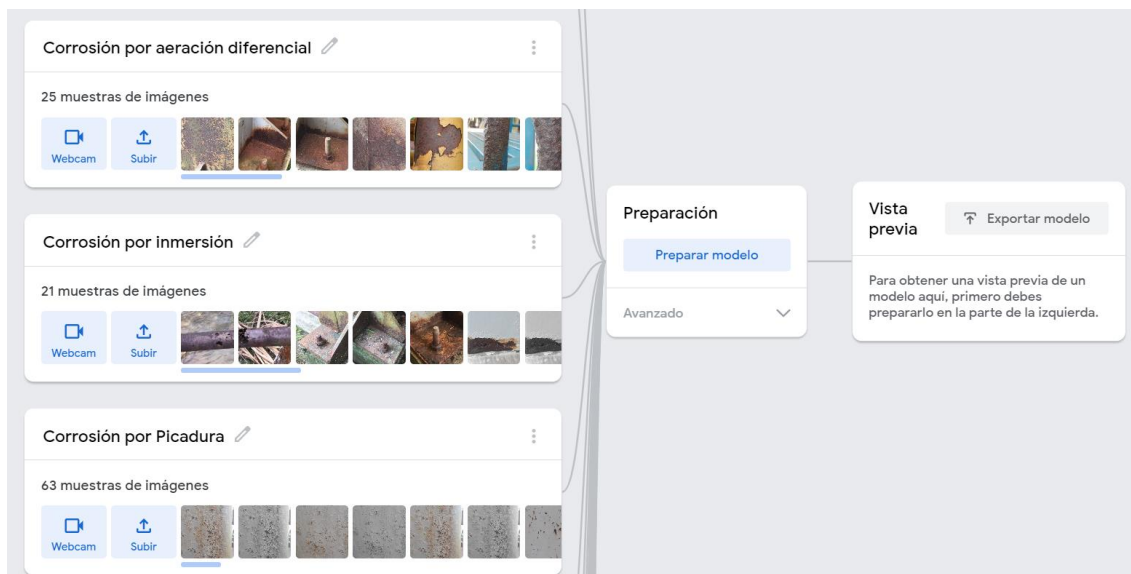


Ilustración 11. Generación de la base de imágenes (*Teachable Machine, 2021*).

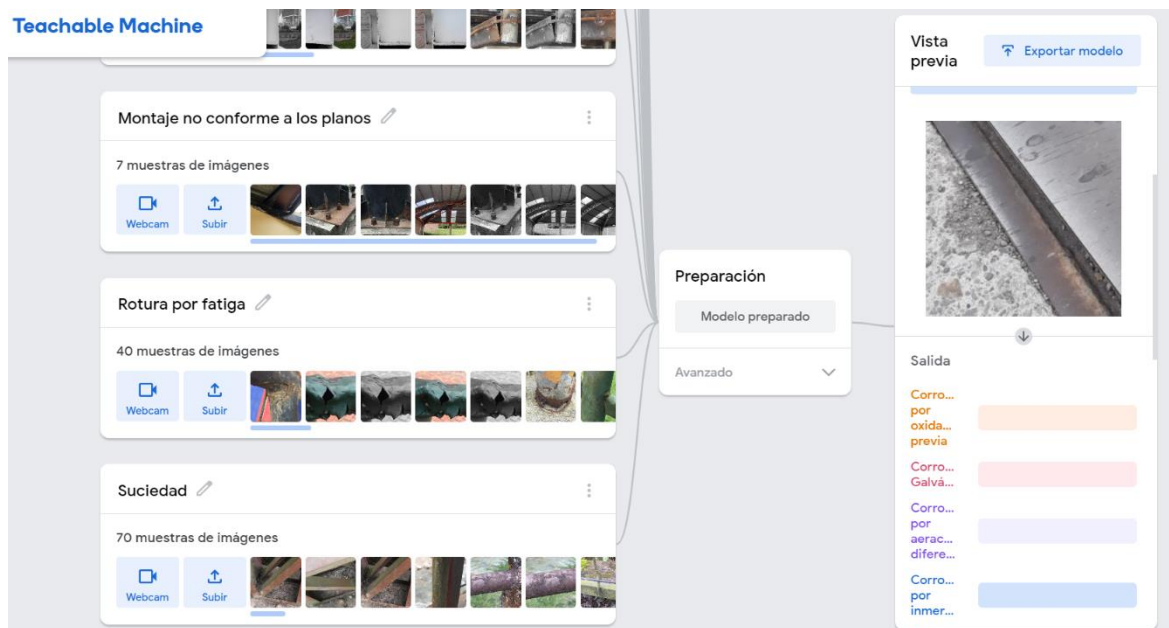


Ilustración 12. Corrido del modelo (Teachable Machine, 2021).

Teachable Machine entrena los modelos dentro del navegador del dispositivo que esté usando, con sus propios datos, sin que estos salgan de su computadora. Además, es posible descargar el modelo, usando TensorFlow.js. Esto permite, una mayor sensación de privacidad y propiedad, no es así con los sistemas que envían y procesan datos a una máquina remota, sin necesidad de preocuparse por almacenar y guardar archivos y conjuntos de datos de gran tamaño, o modelos a la nube (Michelle Carney, 2020).

TensorFlow: es una biblioteca que almacena software de código abierto, con la finalidad para facilitar el aprendizaje automático. Fue creado y habilitado por Google en el año 2015, ha conseguido una gran acogida hasta el momento. La cantidad de colaboradores es cuantiosa, con esto, la sitúan como líder en el sector del aprendizaje profundo (TensorFlow, 2021).

2.10 Clasificadores de búsqueda para la aplicación

Los clasificadores de búsqueda que se implementó en el desarrollo de la aplicación son dos métodos: imágenes y características. Estos métodos ayudan a tener mayor facilidad y eficiencia de búsqueda como se aprecia en la Ilustración 13.

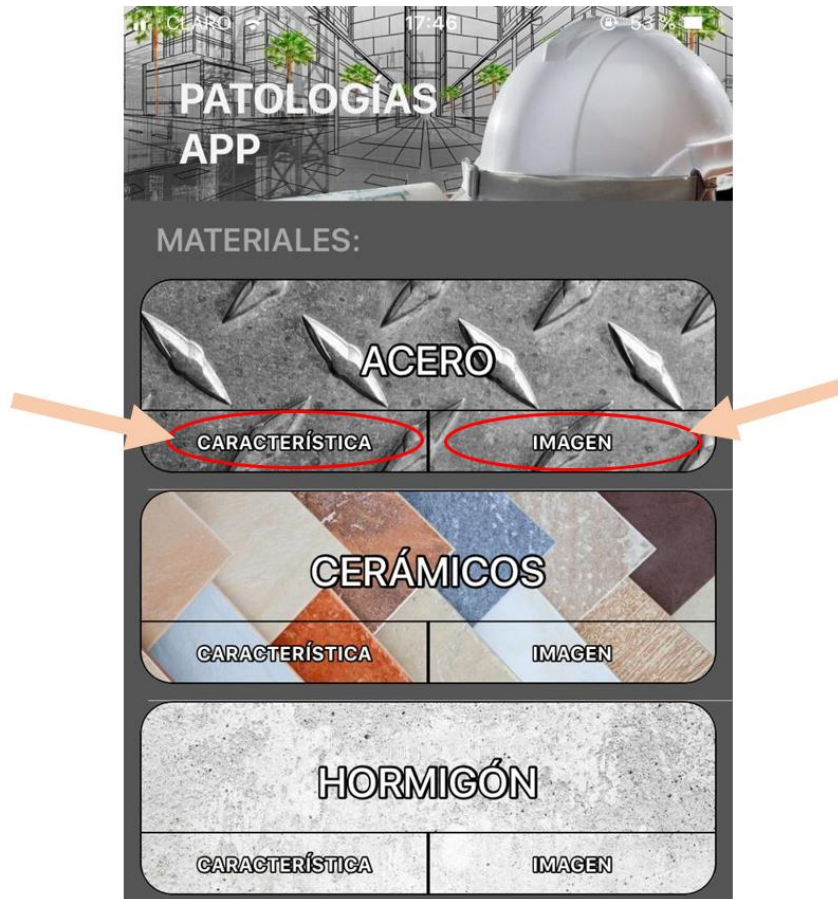


Ilustración 13. Opciones de búsqueda (Autores)

En el reconocimiento de patrones, dice que un objeto es clasificado en una de las categorías llamadas clases, usando características que describan bien las clases (Shigeo, 2001). A través de una observación y un sistema de medición se obtiene un conjunto de números que conforman el vector de medición (Pardo, 2009).

El diseño del clasificador consta de dos partes: el entrenamiento o aprendizaje que recoge las muestras de datos etiquetados y encuentra las fronteras que separan las clases. En

la gran mayoría de los casos los clasificadores presentan ciertos parámetros libres para los que, aunque se deban ajustar no existe ninguna optimización y se hace necesaria una segunda fase de validación. En esta fase, cada clasificador entrenado con una configuración diferente se valida sobre un nuevo conjunto de muestras etiquetadas (Young, 1986).

2.10.1 Basados en Características

El conjunto de características es relativamente fácil de detectar, posteriormente aquellas características que se tornen redundantes o poco descriptivas para el problema se eliminan, y, por otro lado, la no inclusión de características necesarias si sería crítica para el problema resolver (Shigeo, 2001).

La extracción de características se basa en la simplificación de información de la imagen procesada, determinando los aspectos más relevantes de las mismas, que pueden ser capaces de determinar a qué tipo de defecto pertenecen los objetos seleccionados en la segmentación previa (Bravo, 2015).

En esta etapa, a las imágenes le son extraídas un conjunto de características que describen las propiedades físicas de los objetos como pueden ser: el color, textura, área, perímetro, ancho, alto, promedio de intensidad, centroide, orientación, color, numero de hoyos, entre otros (García, 2008).

Las propiedades que se debe dar en los descriptores de características se expresan en el Gráfico 9:

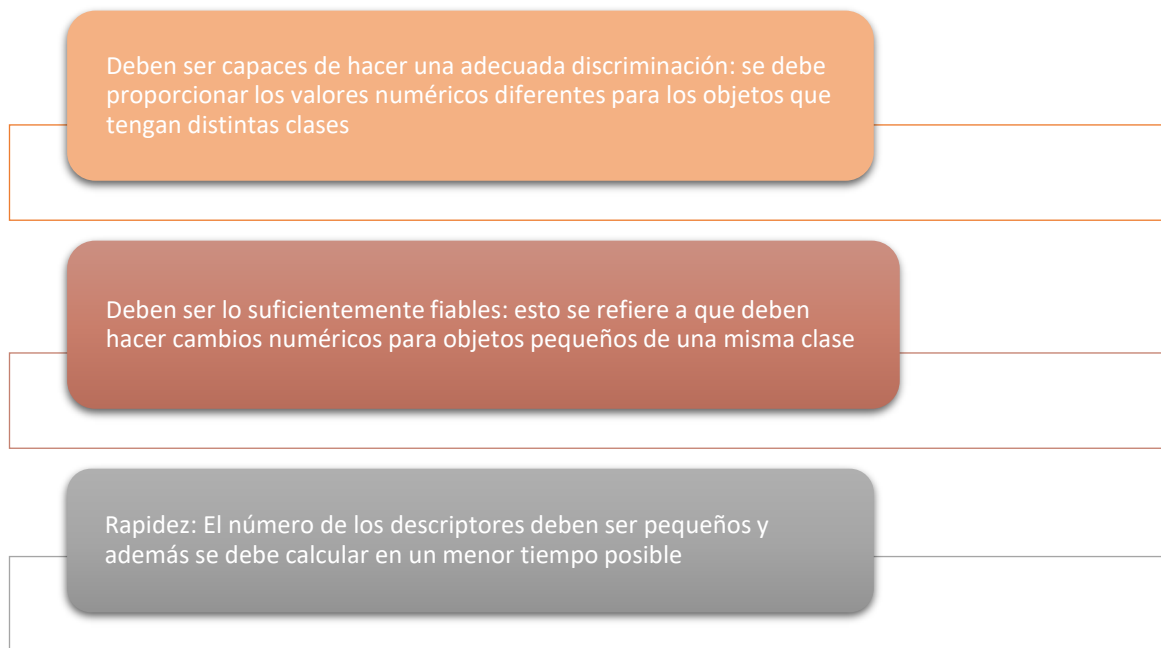


Gráfico 9. Propiedades en descriptores de características, (Bravo, 2015)

2.10.2 Basados en Imágenes

En cambio, en la búsqueda de imágenes o fotografías tomadas en campo, admite hacer una exploración en la base de datos creada en una aplicación de Google llamada Teachable Machine. Se almacenó imágenes tomadas en campo y descargadas de la web de todas las patologías estructurales existentes en los materiales de acero y hormigón, esta búsqueda dará como resultado el porcentaje de cada patología que se encuentra en la fotografía analizada, la patología con mayor porcentaje será la que se seleccione y la aplicación dará como resultado la información completa de dicha patología.

Según G. Senthil Kumar (G. Senthil Kumar, 2012), utiliza como entrada a una red neuronal artificial el valor promedio de los niveles de grises de cada una de las imágenes de corpus establecido. En cambio N. Nacereddine (N. Nacereddine, 2005), presenta una propuesta de clasificación de los principales defectos que se encuentran en inspección

radiográfica utilizando una red neuronal artificial entrenada con el algoritmo backpropagation (Zapata, 2009), trabaja con un sistema adaptativo de inferencia neuroborrosa (lógica difusa).

CAPÍTULO 3

METODOLOGÍA

Para realizar la base de datos con la información (imágenes de patologías), así como la generación de la aplicación de reconocimiento de patologías, los pasos a realizar y secuencias, así como las necesidades deberán estar bien fundamentados. Solo así, se podrá generar un desarrollo tecnológico enfocado a la identificación de patologías locales en los elementos estructurales que sea novedoso. La presente investigación está enfocada a un ámbito poco conocido hasta el momento, sirviendo de base los trabajos e investigaciones previas sobre procesamiento y el reconocimiento de imágenes, para así alcanzar los objetivos planteados.

Por tal motivo, se plantea generar una aplicación mediante un lenguaje de programación, que permita trabajar con imágenes y utilizando inteligencia artificial. Se trabajará con una base de datos, que contendrá los tipos de patologías que existen en las obras civiles. Con la base de datos, se procederá a entrenar un sistema de aprendizaje por computador, para identificar las patologías locales de una manera rápida, eficiente y automatizada. Esta base será realizada mediante la presente investigación.

Después de la recolección de información de fuentes científicas y de campo sobre las diferentes patologías que pueden afectar a las estructuras, es necesaria almacenar la información en una base de datos, que permita el desarrollo de investigaciones que complementen enfermedades en las estructuras de cualquier material mediante inteligencia artificial u otro tipo de tecnologías.

Para identificar los tipos de patologías, se consigue con la creación de un sistema informático para la detección, mediante imágenes a ser analizadas o por medio de

característica de acuerdo al tipo de patología ya que estas no son similares entre ellas, con esto, dotar de una descripción que abarque el tipo de falla, sus posibles causas y soluciones.

En la búsqueda mediante características en las aplicaciones de IOS y Android es un filtrado de información, donde se plantea facilitar al usuario a que pueda elegir algunas características que se observen y sean similares a las lesiones que se encuentren en la imagen de la patología que se está analizando. Dichas características ayudan a tener una búsqueda con más exactitud, y brinda opciones a elegir acorde a la similitud de nuestra fotografía con la enfermedad estructural.

3. Implementación del sistema, detección de patologías estructurales para el Acero y Hormigón

En la Ilustración 14, se aprecia el diagrama general de las funciones de la aplicación:

Diagrama general del funcionamiento de "PATOLOGIAS APP"

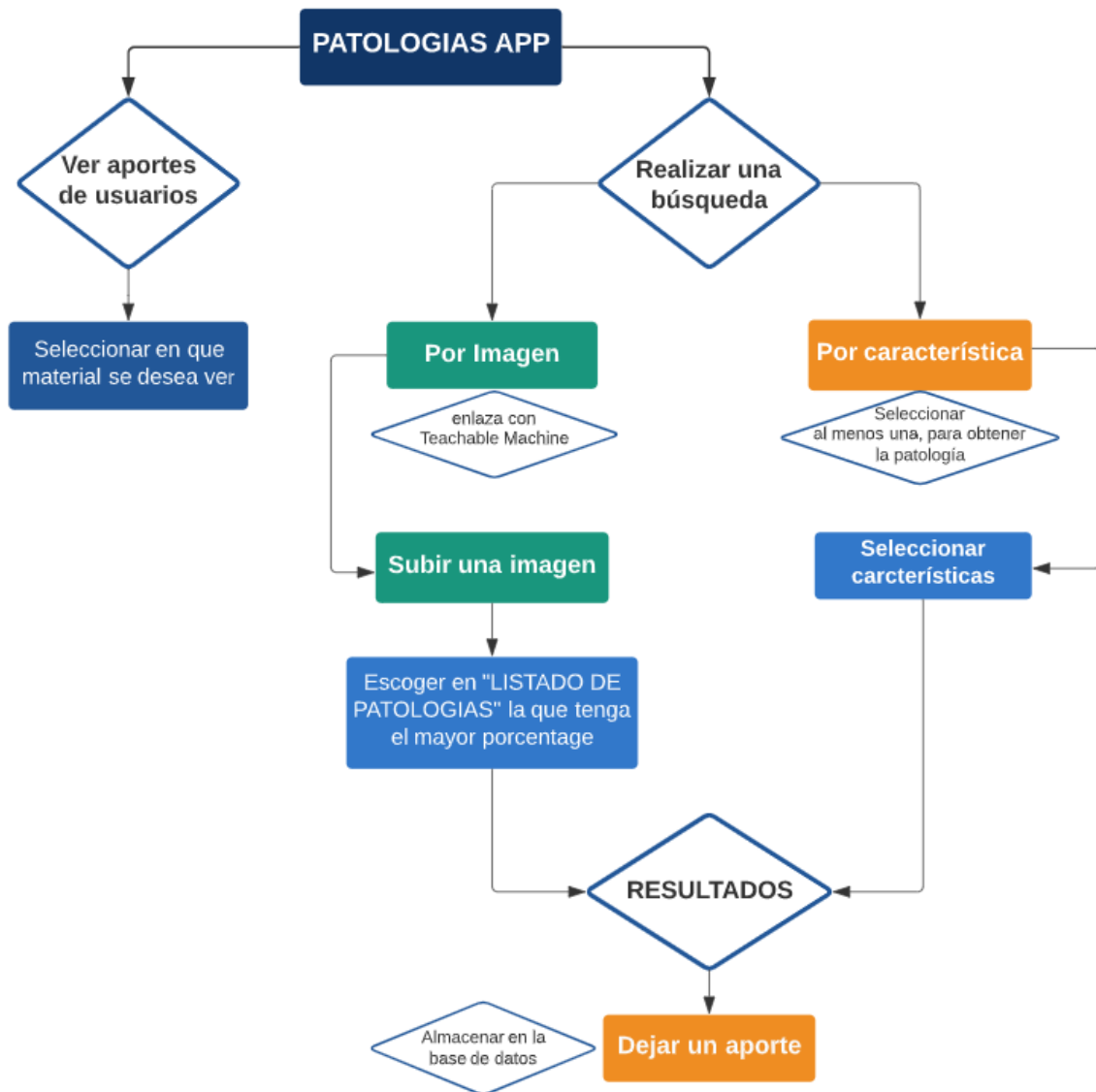


Ilustración 14. Diagrama de flujo, uso de aplicación (Autores)

3.1 Información inicial

3.1.1 Población

Como ya se expuso anteriormente, la experimentación del reconocimiento es de 20 clases o tipos de patologías en el acero y para el hormigón se tienen 28 tipos, cabe recalcar que el número de imágenes para cada clase es distinto, ya que estas se almacenaron con respecto a lo que se consiguió encontrar en campo y además lo que se encontró en las bases de datos de (Jimenez, 2021) y (Cordero, 2021) que se define en las siguientes Tabla 7 y Tabla 8 respectivamente.

Número de clase	Tipo de patología	Número de muestras	%
1	Corrosión por picadura	63	6.89
2	Corrosión galvánica	29	3.17
3	Corrosión por oxidación previa	47	5.14
4	Corrosión por aeración diferencial	25	2.73
5	Corrosión por inmersión	21	2.30
6	Corrosión por tensión	24	2.62
7	Erosión química	116	12.68
8	Suciedad	70	7.65
9	Deformación por impacto	71	7.76
10	Deformación térmica	9	0.98
11	Deformación por sobrecarga	66	7.21
12	Desgarro laminar	16	1.75
13	Rotura por fatiga	40	4.37
14	Rotura frágil	35	3.83
15	Erosión física	92	10.05
16	Soldaduras incorrectas	72	7.87
17	Uniones incorrectas	68	7.43
18	Montaje no conforme a los planos	7	0.77
19	Errores de fabricación	6	0.66
20	Malas prácticas constructivas	38	4.15
	TOTAL	915	100

Tabla 5. Número de imágenes por clases de patologías en acero clasificadas (Autores).

Número de clase	Tipo de patología	Número de muestras	%
1	Fisuras Capilares	25	4.85
2	Fisuras por dilatación y contracción	27	5.24
3	Bacterias	25	4.85

4	Hongos	30	5.83
5	Algas, líquenes y musgos	19	3.69
6	Flechas	10	1.94
7	Pandeos	3	0.58
8	Desplomes	1	0.19
9	Alabeos	3	0.58
10	Grietas por dilatación y contracción higrotérmica	37	7.18
11	Grietas por contracción plástica	7	1.36
12	Grietas por secado	5	0.97
13	Grietas por asentamiento diferencial	47	9.13
14	Fisuras por tracción	15	2.91
15	Fisuras por flexión	26	5.05
16	Fisuras por cortante	11	2.14
17	Fisuras por adherencia	1	0.19
18	Fisuras por torsión	5	0.97
19	Fisuras por punzonamiento	7	1.36
20	Fisuras por compresión	14	2.72
21	Desprendimientos	21	4.08
22	Erosiones mecánicas	18	3.50
23	Eflorescencias	60	11.65
24	Aluminosis	13	2.52
25	Oxidación y Corrosión	30	5.83
26	Carbonatación	19	3.69
27	Coqueras	29	5.63
28	Junta fría	7	1.36
TOTAL		515	100

Tabla 6. Número de imágenes por clases de patologías en Hormigón clasificadas (Autores).

3.1.2 Entorno

Los entornos de adquisición de los datos para las imágenes corresponden a la recopilación de imágenes en la web y de fotografías tomadas en campo de adquisición propia.

3.2 Código de Búsqueda por imagen en Teachable Machine en formato JavaScript para el Acero

La secuencia de programación para la búsqueda por imagen, tanto en Acero como para Hormigón son las mismas; en principio, se debe establecer las clases o tipos de patologías para cada material, estas deben estar contenidas con sus respectivas imágenes, las cuales establecerán la base de datos. Luego, se ejecuta el procesamiento y ordenamiento de clases en la preparación del modelo; y finalmente se ejecuta el código que está compuesto y ordenado para ponerlo en funcionamiento.

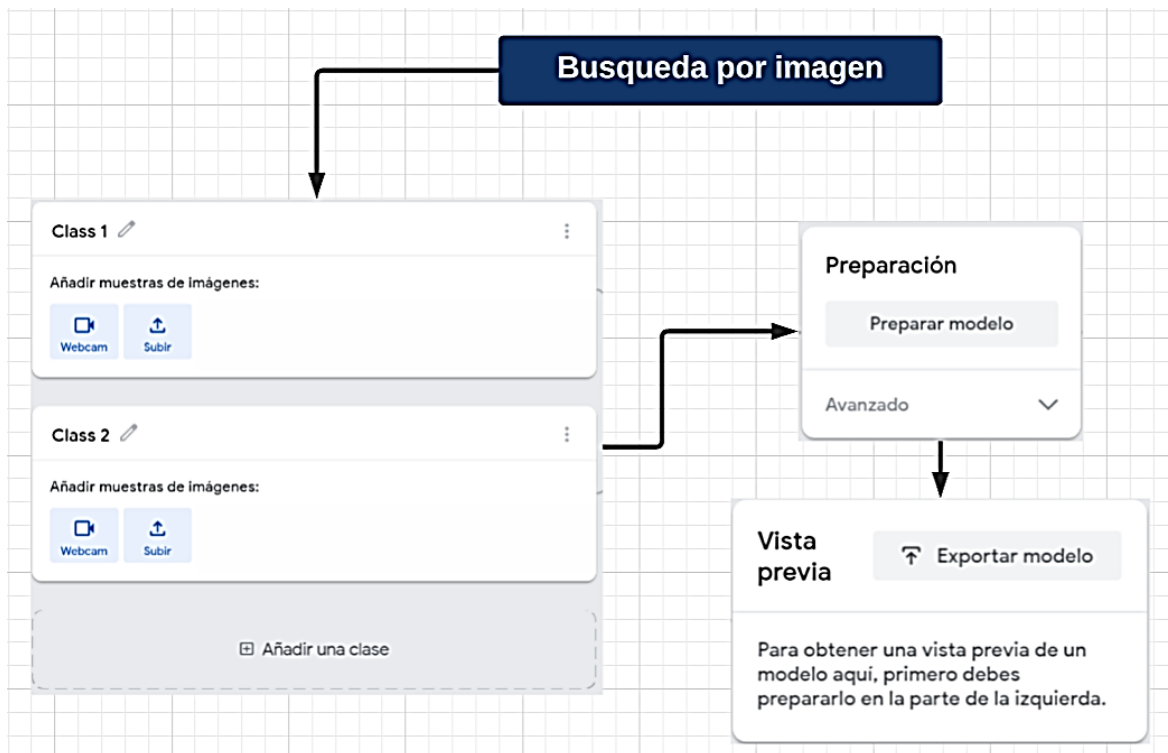


Ilustración 15. Diagrama de flujo, composición de Teachable Machine (Autores)

```
<div>Teachable Machine Image Model</div>
<button type="button" onclick="init()">Start</button>
<div id="webcam-container"></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemachine-image.min.js"></script>
```

```

<script type="text/javascript">
  // More API functions here:
  // https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image

  // the link to your model provided by Teachable Machine export panel
  const URL = "./my_model/";

  let model, webcam, labelContainer, maxPredictions;

  // Load the image model and setup the webcam
  async function init() {
    const modelURL = URL + "model.json";
    const metadataURL = URL + "metadata.json";

    // load the model and metadata
    // Refer to tmImage.loadFromFiles() in the API to support files from a file picker
    // or files from your local hard drive
    // Note: the pose library adds "tmImage" object to your window (window.tmImage)
    model = await tmImage.load(modelURL, metadataURL);
    maxPredictions = model.getTotalClasses();

    // Convenience function to setup a webcam
    const flip = true; // whether to flip the webcam
    webcam = new tmImage.Webcam(200, 200, flip); // width, height, flip
    await webcam.setup(); // request access to the webcam
    await webcam.play();
    window.requestAnimationFrame(loop);

    // append elements to the DOM
    document.getElementById("webcam-container").appendChild(webcam.canvas);
    labelContainer = document.getElementById("label-container");
    for (let i = 0; i < maxPredictions; i++) { // and class labels
      labelContainer.appendChild(document.createElement("div"));
    }
  }

  async function loop() {
    webcam.update(); // update the webcam frame
    await predict();
    window.requestAnimationFrame(loop);
  }

  // run the webcam image through the image model
  async function predict() {
    // predict can take in an image, video or canvas html element
    const prediction = await model.predict(webcam.canvas);
    for (let i = 0; i < maxPredictions; i++) {
      const classPrediction =
        prediction[i].className + ": " + prediction[i].probability.toFixed(2);
      labelContainer.childNodes[i].innerHTML = classPrediction;
    }
  }
</script>

```

3.3 Código de Búsqueda por imagen en Teachable Machine en formato JavaScript para el Hormigón

El Sistema de secuencia de programación es el mismo que se usa para material de Acero, así como en la Ilustración 15.

```
<div>Teachable Machine Image Model</div>
<button type="button" onclick="init()">Start</button>
<div id="webcam-container"></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/image@0.8/dist/teachablemachine-image.min.js"></script>
<script type="text/javascript">
  // More API functions here:
  // https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/image

  // the link to your model provided by Teachable Machine export panel
  const URL = "https://teachablemachine.withgoogle.com/models/EGskw5kOp/";

  let model, webcam, labelContainer, maxPredictions;

  // Load the image model and setup the webcam
  async function init() {
    const modelURL = URL + "model.json";
    const metadataURL = URL + "metadata.json";

    // load the model and metadata
    // Refer to tmImage.loadFromFiles() in the API to support files from a file picker
    // or files from your local hard drive
    // Note: the pose library adds "tmImage" object to your window (window.tmImage)
    model = await tmImage.load(modelURL, metadataURL);
    maxPredictions = model.getTotalClasses();

    // Convenience function to setup a webcam
    const flip = true; // whether to flip the webcam
    webcam = new tmImage.Webcam(200, 200, flip); // width, height, flip
    await webcam.setup(); // request access to the webcam
    await webcam.play();
    window.requestAnimationFrame(loop);

    // append elements to the DOM
    document.getElementById("webcam-container").appendChild(webcam.canvas);
    labelContainer = document.getElementById("label-container");
    for (let i = 0; i < maxPredictions; i++) { // and class labels
      labelContainer.appendChild(document.createElement("div"));
    }
  }

  async function loop() {
    webcam.update(); // update the webcam frame
    await predict();
    window.requestAnimationFrame(loop);
  }
}
```

```
// run the webcam image through the image model  
async function predict() {  
  // predict can take in an image, video or canvas html element  
  const prediction = await model.predict(webcam.canvas);  
  for (let i = 0; i < maxPredictions; i++) {  
    const classPrediction =  
      prediction[i].className + ": " + prediction[i].probability.toFixed(2);  
    labelContainer.childNodes[i].innerHTML = classPrediction;  
  }  
}  
</script>
```

CAPÍTULO 4

4. Ejecución de las pruebas y análisis de resultados

La aplicación cuenta con métodos de búsqueda para la identificación de patologías estructurales en acero y hormigón, una por medio de una pantalla aplicando Teachable Machine y otra mediante el filtrado de las características propias de las patologías.

4.1 Pruebas usando Teachable Machine

En este apartado se realizan pruebas de identificación para comprobar la eficacia y certeza que presenta esta herramienta. Como se muestra en la Ilustración 16 e Ilustración 17, se ensayan seis imágenes, las cuales corresponden a las patologías en Acero y Hormigón respectivamente: Resultados: para el acero se obtiene, Corrosión por picadura (a), Desgarro laminar (b) y Deformación por impacto (c). Estas aparecen con mayor porcentaje de acierto en una barra con porcentaje. De igual manera, para el hormigón se obtiene, Hongos(A1), Fisuras capilares (B1) y Grietas por asentamiento diferencial (C1).

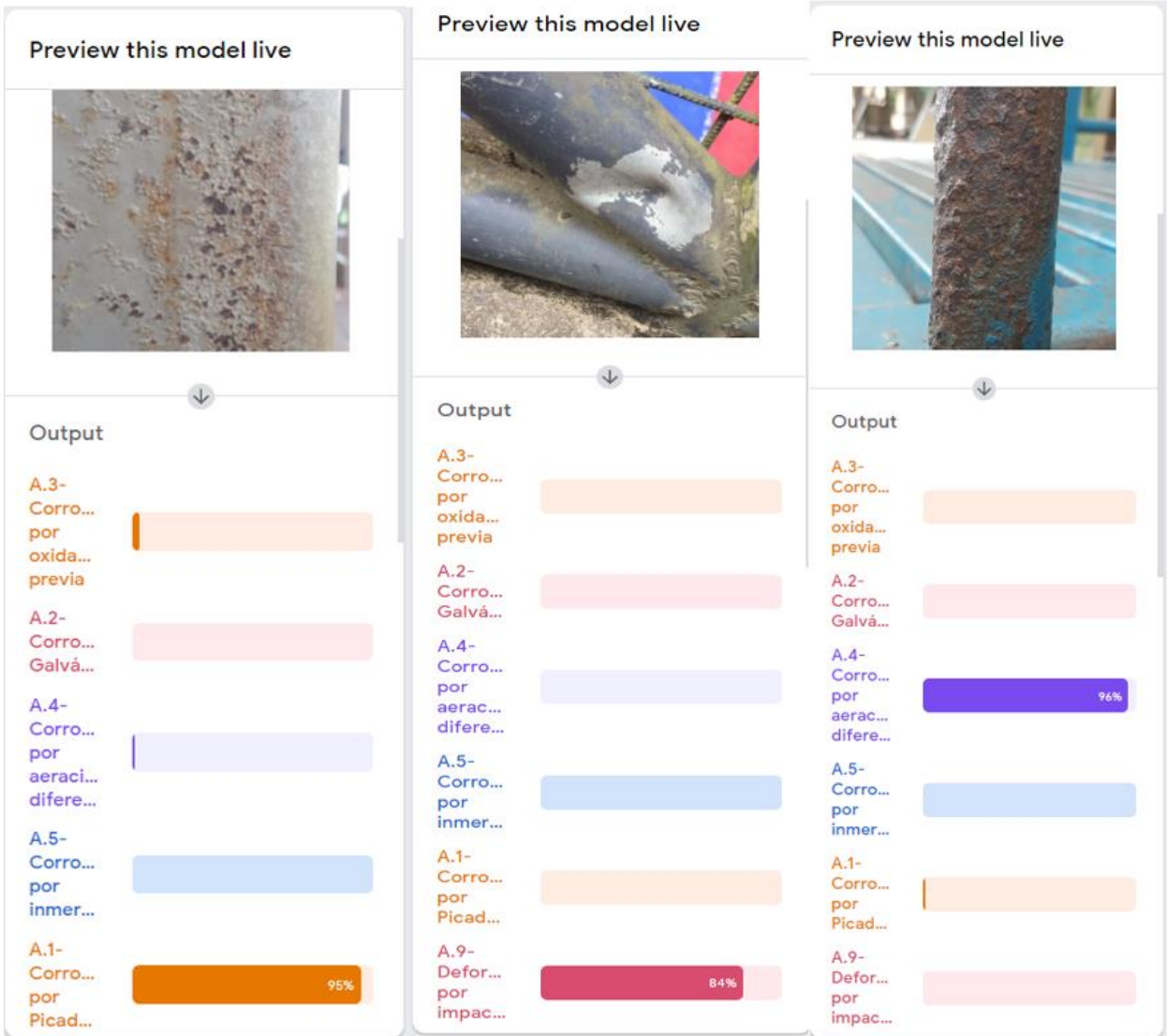


Ilustración 16. Interpretación de pruebas del modelo para Acero (Autores).

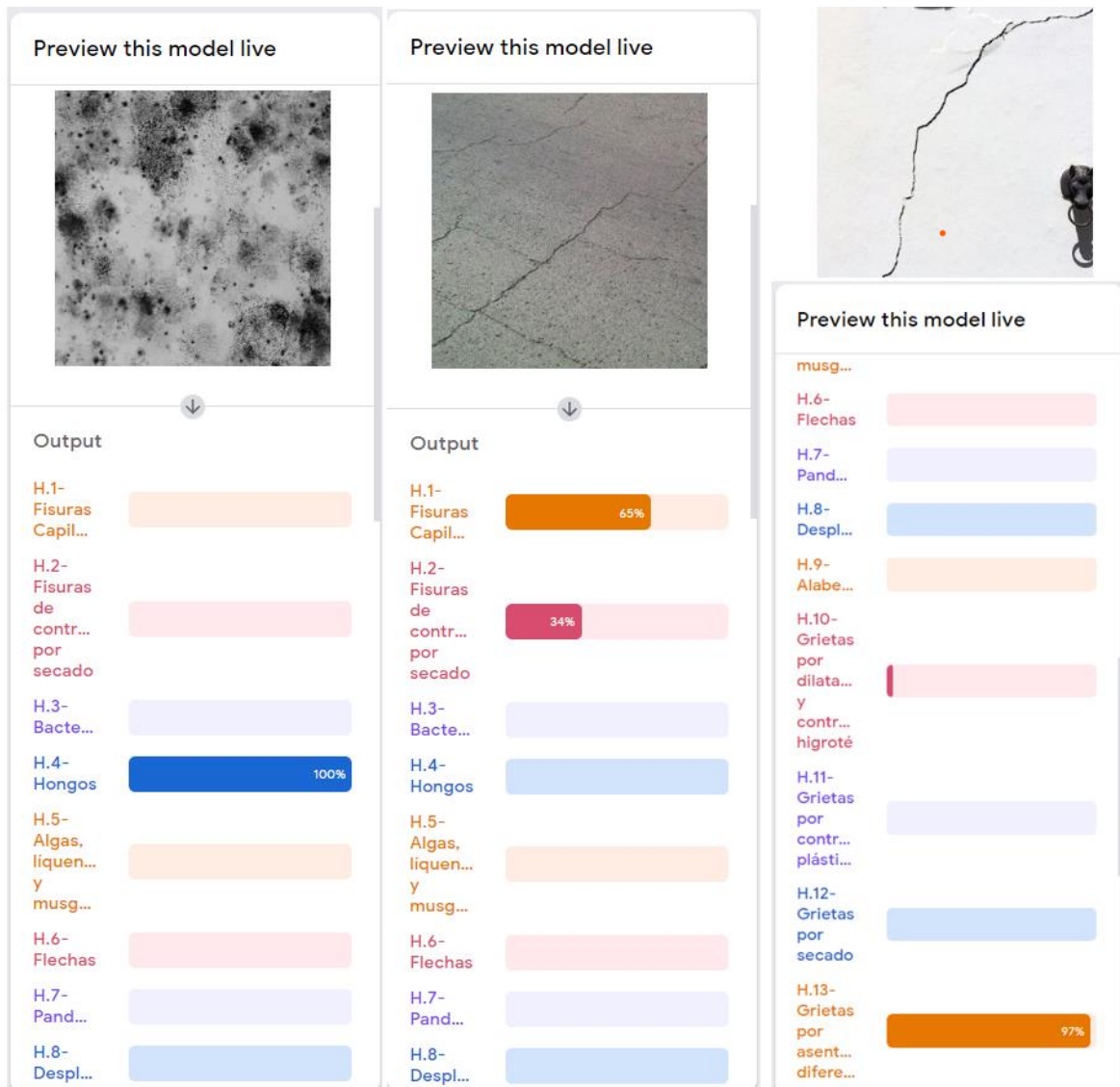


Ilustración 17. Interpretación de pruebas del modelo para Hormigón (Autores).

En algunas pruebas realizadas para la identificación del tipo de patología estructural, los colores entre imágenes resultaron parecidos, por lo que disminuye la precisión en la clasificación.

Es posible confirmar lo que afirman algunos investigadores sobre el proceso de clasificación de las imágenes basadas en los atributos de color y de forma, al plantear que diferentes imágenes pueden presentar los mismos valores de color, esto hace que no se

tengan aún métodos robustos y efectivos para que la identificación de las imágenes sea de manera precisa (Christian Montoya, 2014).

4.2 Pruebas aplicando el filtrado de las imágenes

En las primeras pruebas que se realizó en la aplicación fue mediante el manejo de Teachable Machine, donde se crea una base de datos con todas las imágenes y fotografías que se recopiló mediante un levantamiento fotográfico en campo, para lo cual fue necesaria la visita de parques y edificaciones públicas y privadas.



Ilustración 18. Pantalla principal de la Patologías App (Autores)

En la pantalla principal, como se puede apreciar en la Ilustración 18 están establecidos cuatro tipos de materiales, los cuales se puede hacer una búsqueda por medio de características de cada patología o mediante el reconocimiento artificial de imágenes por

medio del Teachable Machine. Además, se agrega una opción para ver los aportes que dejen los usuarios.

En esta segunda pantalla, como se muestra en la Ilustración 19, se presenta la búsqueda mediante imágenes, el cual se emplea el Teachable Machine donde se crea una base de datos con todas las imágenes y fotografías tomadas en campo y descargadas de la web.

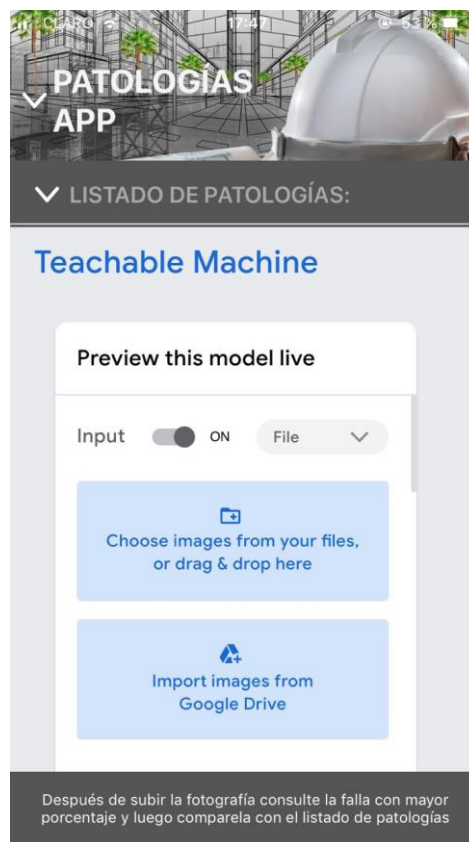


Ilustración 19. Búsqueda mediante imágenes (Autores).

En la Ilustración 20, se muestra la pantalla para la búsqueda por medio de imágenes, donde los valores de identificación aparecen en porcentajes, de acuerdo a la patología que más similitud tiene con la fotografía seleccionada. Dependiendo de la patología que tenga más similitud, se hará una selección en el listado de patologías como en la Ilustración 21, y

así obtener el resultado final de la patología con todas sus características y descripciones que se muestra en la Ilustración 22.

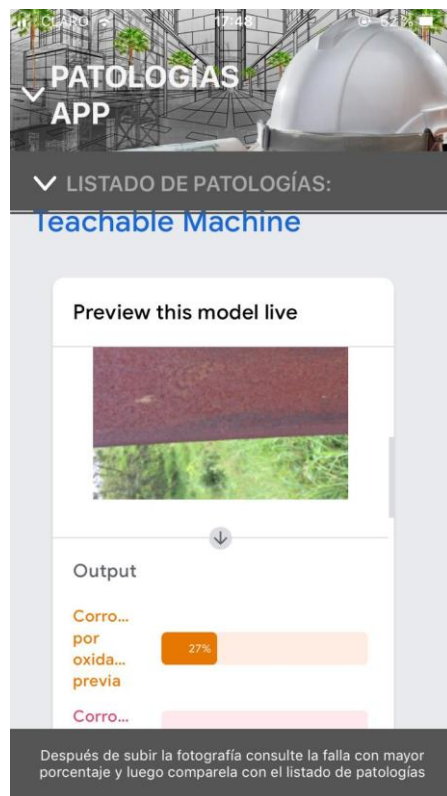


Ilustración 20. Resultado de búsqueda en porcentajes (Autores).

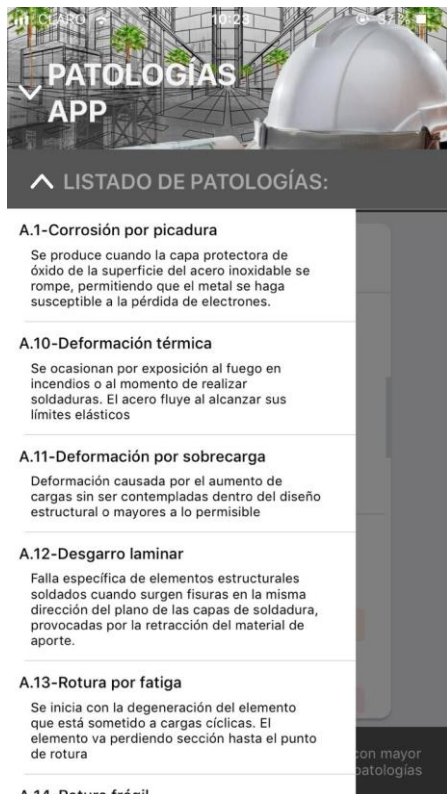


Ilustración 21. Resultado de búsqueda en porcentajes (Autores).



Ilustración 22. Resultado final de la búsqueda, donde se explica la descripción, equipos de medición, tratamientos y técnicas de reconocimiento (Autores).

El segundo método de búsqueda que se ha propuesto en la aplicación, es por medio de características propias de cada patología Ilustración 23, que se enfoca en la selección de ellas. Este método es de facilidad para que el usuario, ya que en el momento que quiera hacer algún tipo análisis de alguna patología existente, en cualquiera de estos materiales pueda determinar las características que posee y hacer la búsqueda seleccionando las características que se presenta en la aplicación.

Del mismo modo, el resultado final será semejante a la búsqueda por imágenes, es decir se presentará una breve descripción de la patología con sus características, equipo de medición, tratamiento, técnicas de reconocimiento y artículos referentes al tipo de patología que se quiera analizar.

Además de esto, la Ilustración 24 presenta la opción para que el usuario pueda dejar un aporte fotográfico, que será útil para un mejor funcionamiento y un resultado más eficiente de búsqueda. Este aporte fotográfico está condicionado a la aprobación por parte de los autores de la aplicación antes de ser agregada a la base de datos.



Ilustración 23: Búsqueda por medio de características.



Ilustración 24: Resultado de búsqueda por características, además se presenta la opción para que el usuario pueda dejar su aporte fotográfico.

CONCLUSIONES

Al final de este trabajo investigativo que fue el desarrollo de una aplicación móvil para los sistemas operativos de Android e iOS, mediante las bases de datos de Acero y Hormigón creadas anteriormente por (Jimenez, 2021) y (Cordero, 2021), se puede concluir con total seguridad, que los resultados obtenidos en campo son muy satisfactorios, con una precisión promedio de 91.6% para las patologías de Desgarro Laminar, Deformación por Impacto y Corrosión por picadura y una precisión promedio superior al 87.3% para las patologías de Grietas por asentamiento diferencial, Fisuras capilares y Hongos.

En cuanto a la investigación, la visión artificial o el procesamiento de imágenes es una herramienta informática muy poderosa y práctica, además esta herramienta puede ser muy útil dependiendo el contexto y la utilidad que se le vaya a dar. Esta es una herramienta completa, pero no tan perfecta, ya que depende mucho del tipo de escenario donde se vaya a tomar la fotografía.

Si bien la base de datos es pequeña, pero es lo suficientemente buena para mostrar resultados acertados en todos los casos que se aplicó. Es así que, en la aplicación se creó un apartado para que los usuarios, puedan aportar a la base de datos general con fotografías e imágenes. Además, esta base de datos queda abierto para futuras investigaciones, donde se desee mejorar la calidad de búsqueda e incluir búsquedas de otros tipos materiales de construcción.

En lo personal, el desarrollo de este proyecto sirvió de mucha ayuda para conocer a fondo la complejidad de este problema en la clasificación de imágenes según el tipo de patologías, es más ayudó a tener presente y conocer de mejor manera los tipos de patologías que se ocasionan en los materiales de Acero y Hormigón. Cabe recalcar que esta

investigación y la aplicación móvil, es de mucha ayuda para todos los profesionales y estudiantes interesados en el área de la Ingeniería y la Arquitectura.

4.3 Trabajos futuros

Se expresa la gran satisfacción por haber alcanzado los objetivos propuestos para la realización de este trabajo, que ha sido una tarea compleja debido al tiempo disponible para abordar este tema.

La parte que abarca esta investigación, deberá ser complementada con otros materiales de construcción como son; los materiales cerámicos y la madera. Es posible realizar una aplicación general de materiales para obras civiles, pero también se puede aplicar el mismo principio de identificación de imágenes en las patologías de los suelos como en las vías con sus tipos de materiales que están contruidos.

4.4 Recomendaciones

Una de las recomendaciones más importantes es para los usuarios que deseen utilizar esta aplicación, es que, al momento de tomar la fotografía para hacer la búsqueda en la base de datos, esta debe ser tomada y enfocada directamente a la patología, también se recomienda tener mucho en cuenta la luminosidad y el enfoque de la misma.

También se puede recomendar que de preferencia las fotografías sean tomadas en el día, debido a la mayor claridad que existe cuando una fotografía es tomada en el día, en comparación con la noche, esto nos ayudara a tener una mejor eficacia a la hora de realizar una búsqueda en nuestra base de datos.

BIBLIOGRAFÍA

Arquitectura iOS - Tecnología iOS. (20 de Agosto de 2021). Obtenido de Arquitectura iOS

- Tecnología iOS.: <https://sites.google.com/site/tecnologiaiostm/desarrollo-de-aplicaciones/arquitectura-ios>

ASCE. (2018). Obtenido de <https://civilgeeks.com/2014/08/03/que-es-una-falla-estructural/>

Borja, F. (6 de Enero de 2008). *Interfaces gráficas de usuario.* Obtenido de Wayback

Machine:

<https://web.archive.org/web/20080106075429/http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Interfaces/enlightment/index.html>

Bravo, C. F. (2015). *Inspección de soldadura MIG/MAG de piezas metálicas utilizando técnicas de visión artificial y procesamiento de imágenes.* Cuenca: Universidad Politécnica Salesiana.

Christian Montoya, J. A. (2014). *Sistema automático de reconocimiento de frutas basado en visión por computador.* Arica: SciELO.

Cordero, L. M. (2021). *Generación de base datos para la detección de patologías en hormigón.* Cuenca.

Cumming, A. (2 de Septiembre de 2005). *Introduzione a ML.* Obtenido de Wayback

Machine:

<https://web.archive.org/web/20050901195929/http://www.dcs.napier.ac.uk/course-notes/sml/manual.html>

- G. Senthil Kumar, U. N. (2012). *Vision inspection system for the identification and classification of defects in mig welding joints*. India: The International Journal of Advanced Manufacturing Technology.
- García, E. S. (2008). *Detección y clasificación de objetos dentro de un salón de clases empleando técnicas de procesamiento digital de imágenes*. Mexico: Universidad Autonoma Metropolitana.
- Google. (20 de 10 de 2021). *developers*. Obtenido de <https://developer.android.com/kotlin/first?hl=es-419>
- Heras, D. (2017). *Clasificador de Imágenes de Frutas Basado en Inteligencia artificial*. Cuenca: Universidad Católica de Cuenca.
- Inc, A. (20 de Agosto de 2021). *Swift.org*. Obtenido de Swift.org: <https://swift.org>
- J. F. Vélez Serrano, A. B. (2003). *Vision por computador* (2da ed.).
- Jarrín, S. A. (2021). *Diseño y simulación de un sistema de reconocimiento*. Quito.
- Jimenez, J. D. (2021). *Generación de base datos sobre patologías en estructuras de acero*. Cuenca.
- Jonggi Hong, K. L. (2019). *Exploring Machine Teaching for Object Recognition with the Crowd*. Scotland: CHI 2019 Late-Breaking Work.
- Kamintetzki. (1991). *Civilgeeks.com*. Obtenido de Civilgeeks.com: <https://civilgeeks.com/2014/08/03/que-es-una-falla-estructural/>
- Kristel Malave Polanco, J. L. (2011). *"Android" Google's operating system for mobile devices*. Venezuela.

- Lara Cancela, S. O. (2012). *Software de comunicaciones*. Obtenido de <http://sites.google.com/site/swcuc3m/home/android/portada>
- Mallick, S. (2016). A Brief History of Image Recognition and. Obtenido de <https://learnopencv.com/image-recognition-and-object-detection-part1/>
- Malpartida, E. A. (2003). *Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot*. Lima.
- Michelle Carney, B. W. (2020). *Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification*. Honolulu: CHI.
- N. Nacereddine, M. Z. (2005). *Weld defect detection in industrial radiography based dijital image processing*. Tunisia: Technologies Of Information And Telecommunications.
- O. Rodríguez, A. (2018). *Desarrollo de una aplicación de reconocimiento en imágenes utilizando Deep Learning con OpenCV*.
- Open handset alliance*. (2012). Obtenido de http://www.openhandsetalliance.com/android_overview.html
- Owen, W. (2 de Junio de 2014). *Apple Announces Swift, A New Programming Language for iOS and OS X*. Obtenido de TNW | Apple: <https://thenextweb.com/news/apple-announces-swift-new-programming-language-ios>
- Pardo, S. (2009). *Aplicación del modelo bag-of-words al reconocimiento de imágenes*. Madrid: Universidad Carlos III de Madrid.
- Parker, J. J. (2010). Algorithms for image processing and computer vision.

- Platero, C. (2013). Introducción a la Visión Artificial. Obtenido de http://www.elai.upm.es/webantigua/spain/Asignaturas/MIP_VisionArtificial/ApuntesVA/cap1IntroVA.pdf
- Raspberry PI, F. (2018). *Raspberry PI*. Obtenido de <https://www.raspberrypi.org/>
- Rebecca Fiebrink, M. G. (2018). *Introduction to the Special Issue on Human-Centered Machine Learning*. Londres: ACM Transactions on Interactive Intelligent Systems.
- Sepulveda, J. A. (2013). *Sistema de desarrollo para objetos de aprendizaje en dispositivos móviles con sistema operativo iOS*. Valdivia: Universidad Austral de Chile.
- Shigeo, A. (2001). *Pattern classification. Neuro-fuzzy methods and their comparison*. London: Springer-Verlang London.
- Swift - Apple Developer*. (24 de Agosto de 2021). Obtenido de Swift - Apple Developer: <https://developer.apple.com/swift/>
- Teachable Machine*. (25 de Agosto de 2021). Obtenido de Teachable Machine: <https://teachablemachine.withgoogle.com/>
- TensorFlow*. (20 de Agosto de 2021). Obtenido de TensorFlow: TensorFlow
- Victor Mendoza Lanche, E. A. (2016). *Procesamiento de Imágenes para el reconocimiento de edificios mediante*. Acapulco: Tlamati Sabiduría.
- Yan, Y. &. (2014). Large-scale image processing research cloud. In The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization.
- Yeicy Juliana Molina Rivera, J. S. (2012). *Sistema operativo Android: características y funcionalidad para dispositivos móviles*. Pereira.

Young, A. (1986). *Handbook of Pattern Recognition and Image Processing*. Michigan:
Elsevier Science.

Zapata, J. V. (2009). *An adaptive-network-based fuzzy inference system for classification
of welding defects*. Cartagena: NDT & E International.

APÉNDICE A

Especificaciones de la aplicación e instructivo de uso

Especificaciones: La aplicación está disponible en la App Store, con el nombre “PATOLOGIAS APP”.

- Tamaño: 10.6 MB
- Categoría: Educación
- Compatibilidad: Requiere iOS 12.1 o posterior. Compatible con iPhone, iPad y iPod touch.

Instrucciones de uso:

Una vez que se descarga la aplicación en el dispositivo, cuando se ha instalado correctamente, ubicamos el ícono de la aplicación “Patologías” en el menú de aplicaciones o en la pantalla del teléfono presionar sobre la aplicación para que se inicie de manera automática como en la Ilustración 26.

La pantalla principal de la aplicación es como en la Ilustración 27, en ella podemos apreciar los materiales y los tipos de búsqueda, por característica o por imagen.



Ilustración 25. Icono de aplicación (Autores).



Ilustración 26. Pantalla de general de la aplicación (Autores).

Búsqueda por imagen: al acceder por este método, saldrá un mensaje de confirmación, donde aparece la primera instrucción como en la Ilustración 28. Luego, ingresando a la pantalla de búsqueda por imagen, el enlace web de Teachable Machine dará la opción de ingresar una imagen guardada en el dispositivo y tomadas en el momento por la cámara como en Ilustración 29 y también la Ilustración 30.



Ilustración 27. Acceso a la búsqueda por imagen (Autores).

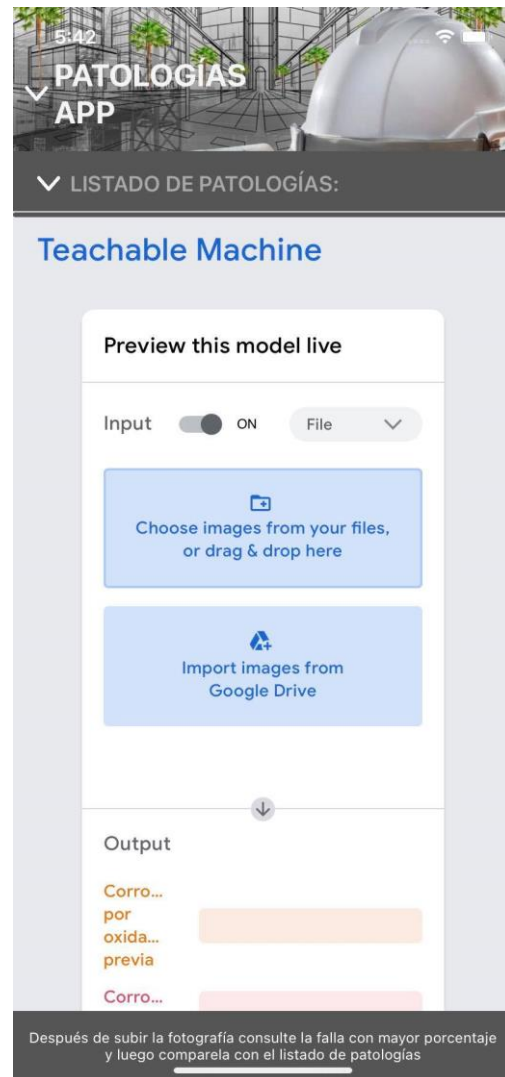


Ilustración 28. Enlace web para la búsqueda por imagen (Autores).

En la Ilustración 31, se observa los resultados de la búsqueda por imagen en forma de barras con porcentajes según la coincidencia con la base de datos, en la parte baja de la pantalla aparece una instrucción, para seleccionar del listado de patologías aquella que posee el mayor porcentaje de coincidencia como Ilustración 32.

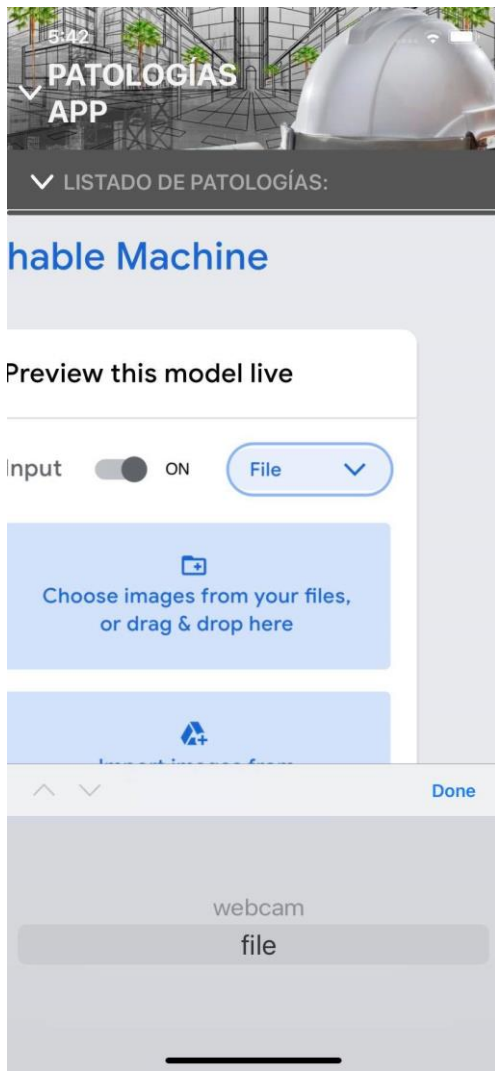


Ilustración 29. Apartado que permite el acceso de imágenes (Autores).

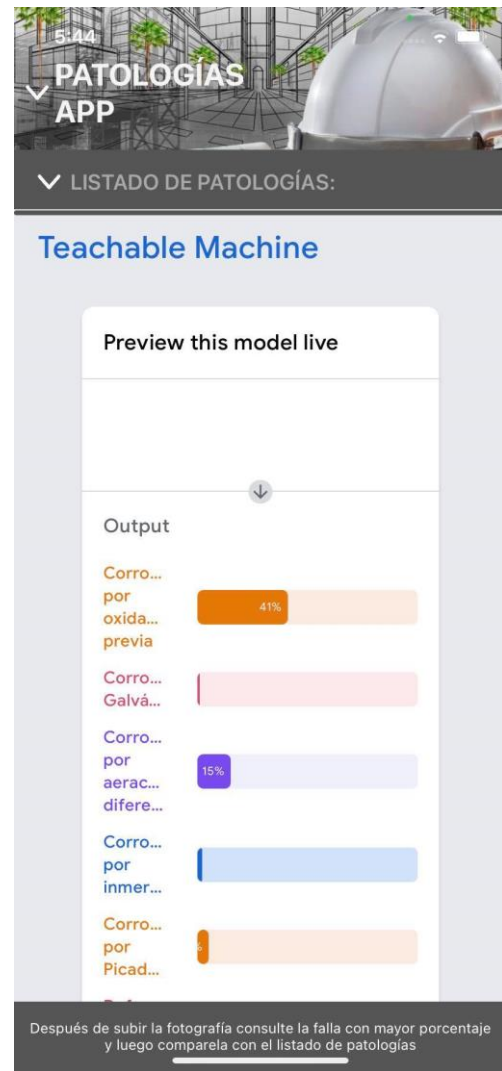


Ilustración 30. Resultados de búsqueda (Autores).

La Ilustración 32, muestra el listado de todas las patologías en el Acero que son extraídas de la base de datos, permite elegir la opción correcta.

Los resultados son un conjunto de información que es extraída de la base de datos, esta de muestra en la Ilustración 33.

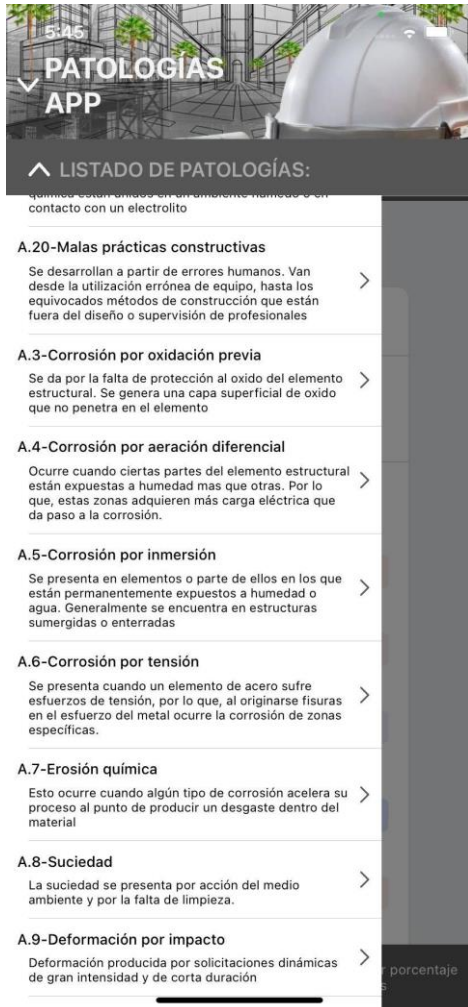


Ilustración 31. Listados de patologías (Autores).

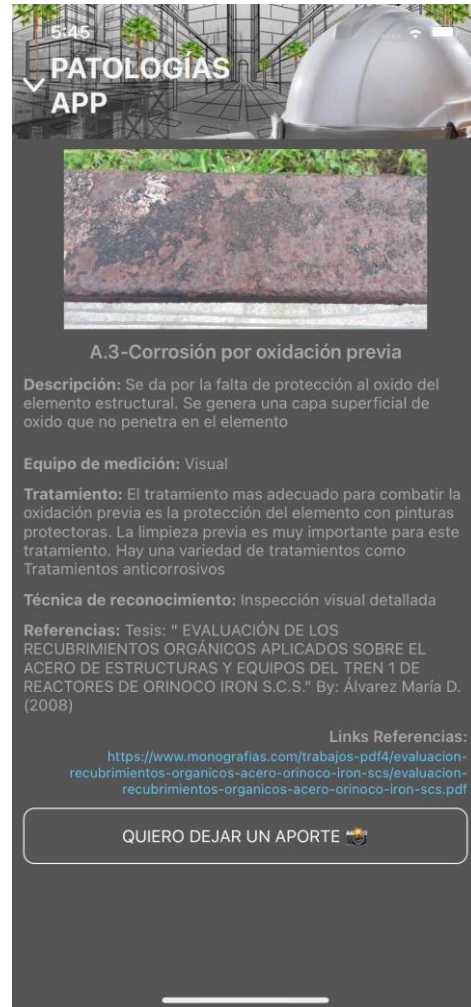


Ilustración 32. Pantalla de resultados de búsqueda (Autores).

Búsqueda por características: como se aprecia en la Ilustración 28, se puede elegir este método.

Donde, aparecerá la lista de características de todas las patologías como en la Ilustración 34, aquí se deberá seleccionar más de una opción como se muestra en la Ilustración 35 y aparecerá en la parte baja de la pantalla el enlace para ir a los resultados.



Ilustración 33. Pantalla de características (Autores).



Ilustración 34. Selección de características (Autores).

Dejar un aporte: en la pantalla de resultados, como en la Ilustración 36, al seleccionar el recuadro “QUIERO DEJAR UN APORTE” arrojará un mensaje de confirmación si queremos tomar una foto de campo o subir una imagen del dispositivo, así como se muestra en la Ilustración 36.

Luego, aparecerá un mensaje de confirmación que se guardará en una base de datos externa, como la Ilustración 37.



Ilustración 35. Opción para dejar un aporte (Autores).

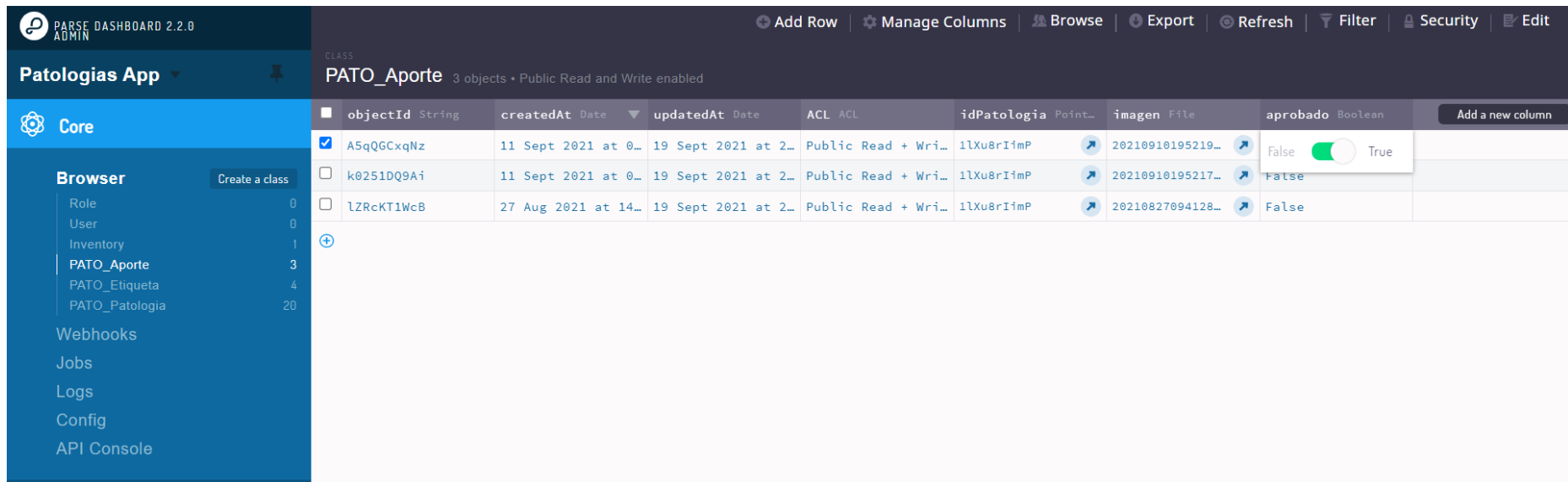


Ilustración 36. Pantalla con mensaje de confirmación para un aporte (Autores).

APÉNDICE B

Filtrado de información en la base de datos

- **Base de datos para los aportes de usuarios:** en esta etiqueta se almacenan automáticamente las imágenes compartidas de los usuarios, por parte de los autores corresponde la apreciación de las imágenes que pertenezcan a patologías y su tipo, posteriormente se hace la aprobación como se aprecia en la Ilustración 38.



The screenshot displays the Parse Dashboard interface for the 'Patologias App'. The main content area shows a table for the 'PATO_Aporte' class, which contains 3 objects. The table has the following columns: objectId (String), createdAt (Date), updatedAt (Date), ACL (ACL), idPatologia (Point), imagen (File), and aprobado (Boolean). The first row is selected, and a modal is open for the 'aprobado' field, showing a toggle switch set to 'True'.

objectId	String	createdAt	Date	updatedAt	Date	ACL	ACL	idPatologia	Point	imagen	File	aprobado	Boolean
<input checked="" type="checkbox"/>	A5qQGcxqNz	11 Sept 2021 at 0...		19 Sept 2021 at 2...		Public Read + Wri...		1lXu8IimP		20210910195219...		False	<input checked="" type="checkbox"/>
<input type="checkbox"/>	k0251DQ9Ai	11 Sept 2021 at 0...		19 Sept 2021 at 2...		Public Read + Wri...		1lXu8IimP		20210910195217...		False	<input type="checkbox"/>
<input type="checkbox"/>	lZRcKT1WcB	27 Aug 2021 at 14...		19 Sept 2021 at 2...		Public Read + Wri...		1lXu8IimP		20210827094128...		False	<input type="checkbox"/>

Ilustración 37. Base de datos para almacenar las imágenes (Autores).

- **Base de datos para la Etiqueta:** consiste en el espacio que los autores editan los tipos de materiales y sus imágenes en general en la Ilustración 39.

PARSE DASHBOARD 2.2.0 ADMIN

Patologias App

Core

Browser

Push

CLASS PATO_Etiqueta 4 objects • Public Read and Write enabled

objectId	String	createdAt	Date	updatedAt	Date	ACL	ACL	nombre	String	fondo	File
<input type="checkbox"/>	GLP05t62gY	27 Aug 2021 at 01...		9 Sept 2021 at 20...		Public Read + Wri...		ACERO		acero.jpeg	
<input type="checkbox"/>	k9g4LR5KwP	27 Aug 2021 at 01...		27 Aug 2021 at 01...		Public Read + Wri...		HORMIGÓN		hormigon.jpeg	
<input type="checkbox"/>	3MxYMkdvzD	27 Aug 2021 at 01...		27 Aug 2021 at 02...		Public Read + Wri...		CERÁMICOS		ceramicos.jpeg	
<input type="checkbox"/>	cS43ellCzK	27 Aug 2021 at 01...		27 Aug 2021 at 01...		Public Read + Wri...		MADERA		madera.jpeg	

Ilustración 38. Base de datos para los tipos de materiales (Autores).

- Base de datos que contiene la información que aparece en los resultados para la aplicación: la etiqueta contiene la descripción de cada patología, las características, imágenes representativas que son una por cada patología, la referencia documentada y el tratamiento, Ilustración 40.

urlReferencia	equipoMedición	imagenes	nombre	ACL	características	descripción	updatedAt	IdEtiqueta	tratamiento	tecnicaReconocim	referencia	imagen	created
(1): http://revis...	Ultrasonido autom...	https://drive.goo...	A.1-Corrosión por...	Public Read + Wri...	*Pequeñas cavidad...	Se produce cuando...	27 Aug 2021 at 02...	0LP05t02gY	Existen algunos t...	Inspección por ul...	*Artículo 1 "Dete...	A 1 Corrosion ...	27 Aug
https://www.monog...	Visual	https://drive.goo...	A.3-Corrosión por...	Public Read + Wri...	*Polvo de oxido *	Se da por la falt...	27 Aug 2021 at 02...	0LP05t02gY	El tratamiento ma...	Inspección visual...	Tesis: " EVALUACI...	A 3 Corrosion ...	27 Aug
https://www.scieL...	Ultrasonido autom...	https://drive.goo...	A.2-Corrosión gal...	Public Read + Wri...	*Dos materiales d...	Ocurre cuando dos...	27 Aug 2021 at 02...	0LP05t02gY	Un tratamiento ef...	Inspección por ul...	Artículo: "Introd...	A 2 Corrosion ...	27 Aug
@article{melcher...	Detector de humed...	https://drive.goo...	A.5-Corrosión por...	Public Read + Wri...	*Capa de hidróxid...	Se presenta en el...	27 Aug 2021 at 02...	0LP05t02gY	La corrosión por ...	Inspección visual...	Artículo: " Infl...	A 5 Corrosion ...	27 Aug
http://monografia...	Detector de humed...	https://drive.goo...	A.4-Corrosión por...	Public Read + Wri...	*Solo partes espe...	Ocurre cuando cie...	27 Aug 2021 at 03...	0LP05t02gY	La corrosión por ...	Inspección visual...	Artículo: " LOS P...	A 4 Corrosion ...	27 Aug
@phdthesis{sole2...	Ultrasonido autom...	https://drive.goo...	A.6-Corrosión por...	Public Read + Wri...	*Presencia de fis...	Se presenta cuand...	27 Aug 2021 at 02...	0LP05t02gY	La corrosión por ...	Inspección por ul...	Artículo: "Proced...	A 6 Corrosion ...	27 Aug
@ARTICLE {, autho...	Flexómetro digita...	https://drive.goo...	A.7-Erosión quími...	Public Read + Wri...	*Desgaste del mat...	Esto ocurre cuand...	27 Aug 2021 at 02...	0LP05t02gY	Ningún tratamient...	Inspección por ul...	Artículo: "Corros...	A 7 Erosion Qu...	27 Aug
@ARTICLE {, autho...	Visual	https://drive.goo...	A.8-Suciedad	Public Read + Wri...	*Presencia de mat...	La suciedad se pr...	27 Aug 2021 at 02...	0LP05t02gY	Remoción de la ca...	Inspección visual...	Artículo: "Proble...	A 8 Suciedad.j...	27 Aug
@ARTICLE {, autho...	Flexómetro digita...	https://drive.goo...	A.9-Deformación p...	Public Read + Wri...	*Flechas excesiva...	Deformación produ...	27 Aug 2021 at 02...	0LP05t02gY	Cuando esta defor...	Inspección visual...	Artículo: " DIAGN...	A 9 Deformacio...	27 Aug
@ARTICLE {, autho...	Flexómetro digita...	https://drive.goo...	A.10-Deformación ...	Public Read + Wri...	*Flechas excesiva...	Se ocasionan por ...	27 Aug 2021 at 02...	0LP05t02gY	Un elemento defor...	Inspección visual...	Artículo: "LA VUL...	A 10 Deformaci...	27 Aug
@ARTICLE {, autho...	Flexómetro digita...	https://drive.goo...	A.11-Deformación ...	Public Read + Wri...	*Flechas excesiva...	Deformación causa...	27 Aug 2021 at 02...	0LP05t02gY	Un elemento defor...	Inspección visual...	Artículo: " AN EX...	A 11 Deformaci...	27 Aug
@article{feier201...	Líquidos penetran...	https://drive.goo...	A.12-Desgarro lam...	Public Read + Wri...	*Fisuras paralela...	Falla específica ...	27 Aug 2021 at 02...	0LP05t02gY	Reemplazo del ele...	Inspección por lí...	Artículo: "Lamell...	A 12 Desgarro ...	27 Aug
@nproceedings{ma...	Ultrasonido autom...	https://drive.goo...	A.13-Rotura por f...	Public Read + Wri...	*Pérdida de secci...	Se inicia con la ...	27 Aug 2021 at 02...	0LP05t02gY	Reemplazo del ele...	Inspección por ul...	Artículo: "Reduci...	A 13 Rotura po...	27 Aug
@article{yan2020f...	Visualradiografía...	https://drive.goo...	A.14-Rotura frági...	Public Read + Wri...	*Prolongación de ...	Se produce a lo l...	27 Aug 2021 at 02...	0LP05t02gY	Reemplazo del ele...	Inspección visual...	Artículo: "Fatigu...	A 14 Rotura fr...	27 Aug
@article{sanchez2...	Flexómetro digita...	https://drive.goo...	A.15-Erosión fisi...	Public Read + Wri...	*Desgaste del mat...	Es la acción dest...	27 Aug 2021 at 03...	0LP05t02gY	Los tratamientos ...	Inspección visual...	Artículo: "Ident...	A 15 Erosion f...	27 Aug
@ARTICLE {, autho...	Líquidos penetran...	https://drive.goo...	A.16-Soldaduras f...	Public Read + Wri...	*Fisuras *Burbuja...	Es la mala unión ...	27 Aug 2021 at 02...	0LP05t02gY	Un tratamiento ad...	Inspección por lí...	Artículo: "EVALUA...	A 16 Soldadura...	27 Aug
@article{jimenez2...	Visual	https://drive.goo...	A.17-Uniones inco...	Public Read + Wri...	*Unión sin contin...	Forma incorrecta ...	27 Aug 2021 at 02...	0LP05t02gY	Para este tipo de...	Inspección visual...	Artículo: "Compor...	A 17 Uniones i...	27 Aug
Artículo 1: http:	Visual	https://drive.goo...	A.18-Montaje no c...	Public Read + Wri...	*Montaje erróneo ...	Equívocación en l...	27 Aug 2021 at 02...	0LP05t02gY	Revisar las índic...	Inspección visual...	Artículo 1: "IDEN...	A 18 Montaje n...	27 Aug
@mastersthesi{h...	Flexómetro digita...	https://drive.goo...	Errores de fabric...	Public Read + Wri...	*Erróneos espesor...	Errores en la fab...	9 Sept 2021 at 20...	0LP05t02gY	Revisar el elemen...	Inspección visual...	Artículo: "PATOLO...	A 19 Errores d...	27 Aug
@mastersthesi{ve...	Visual	https://drive.goo...	A.20-Malas prácti...	Public Read + Wri...	*Errores en norma...	Se desarrollan a ...	27 Aug 2021 at 02...	0LP05t02gY	Para evitar las m...	Visual	Artículo: "IDENTI...	A 20 Malas pra...	27 Aug

Ilustración 39. Base de datos que contiene los resultados de cada tipo de patología (Autores).

APÉNDICE C

Código swift para funcionamiento de aplicación en el sistema iOS

- **Búsqueda por medio del procesamiento de imágenes, usando la herramienta de Google “Teachable Machine”**

En esta pantalla que muestra la aplicación, el usuario puede ingresar una fotografía que tenga en la memoria del dispositivo, así mismo se permite la captura de fotografías en el instante de la inspección, adicionalmente ofrece ciertos mensajes de su uso.

```
//  
// NavegadorVC.swift  
// Patologias  
//  
// Created on 11/8/21.  
//  
  
import UIKit  
import WebKit  
import Parse  
import RKDropdownAlert  
  
class NavegadorVC: UIViewController, UIWebViewDelegate, WKUIDelegate, WKNavigationDelegate,  
UITableViewDelegate, UITableViewDataSource {  
  
    @IBAction func accionRegresar(_ sender: Any) {  
        self.dismiss(animated: true, completion: {  
        })  
    }  
  
    @IBOutlet weak var vOscura: UIView!  
    let localNotification = UINotificationFeedbackGenerator()  
    @IBOutlet weak var tvPatologias: UITableView!  
    var arrayPatologia:[Patologia] = []  
    var cargadoPatologia:Bool = false  
    let appDelegate = UIApplication.shared.delegate as! AppDelegate  
    @IBOutlet weak var pvCarga: UIProgressView!  
    var myTimer = Timer()  
    var cargarBlanco = false  
    var progressWK : UIProgressView!  
    @IBOutlet weak var wkWebView: WKWebView!  
    var theBool = false  
    var textoLight = false  
    var observer: NSKeyValueObservation?  
    var mostrandoPatos = false  
    @IBOutlet weak var btnChevronPatos: UIButton!  
  
    struct idEtiqueta {  
        static var seleccionada:String!  
    }  
}
```

```

@IBAction func muestraOcultasPatos(_ sender: UIButton) {

    if arrayPatologia.count > 0 {
        if mostrandoPatos {
            mostrandoPatos = false
            tvPatologias.isHidden = true
            vOscura.isHidden = true
            btnChevronPatos.setImage(UIImage(named: "down.png"), for: .normal)
        }
        else {
            mostrandoPatos = true
            tvPatologias.isHidden = false
            vOscura.isHidden = false
            btnChevronPatos.setImage(UIImage(named: "up.png"), for: .normal)
        }
    }
    else {
        RKDDropdownAlert.title("Error", message: "No existen patologias", backgroundColor: UIColor.red,
        textColor: UIColor.white, time: 3)
        self.localNotification.notificationOccurred(.error)
    }
}

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return cargadoPatologia ? arrayPatologia.count : 1
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "CeldaPatologia") as! celdaPatologiaTVC
    if cargadoPatologia {
        let pato = arrayPatologia[indexPath.row]
        cell.lblNombre.text = pato.nombre
        cell.lblDescripcion.text = pato.descripcion
    }
    return cell
}

func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
    tableView.deselectRow(at: indexPath, animated: true)
    PatologiasVC.patoLocal.seleccionada = arrayPatologia[indexPath.row]
    let patosVC = self.storyboard?.instantiateViewController(withIdentifier: "identificadorPatologiasVC")
as! PatologiasVC
    self.present(patosVC, animated: true, completion: {
    })
}

override func viewDidLoadSubviews() {

}

@objc func AccionOcultarListado(){
    muestraOcultasPatos(btnChevronPatos)
}

override func viewDidLoad() {
    super.viewDidLoad()

    tvPatologias.register(UINib(nibName: "celdaPatologia", bundle: nil), forCellReuseIdentifier:
"CeldaPatologia")
}

```



```

        let carac = object["caracteristicas"] as! String
        let pato = Patologia(objectId: object.objectId!, nombre: nombre, descripcion: desc,
idEtiqueta: idEtiqueta, referencia: ref, urlReferencia: urlRef, equipoMedicion: equ, tecnicaReconocimiento:
tec, caracteristicas: carac, tratamiento: trat, prevencion: prev, imagenes: imgs)
        self.arrayPatologia.append(pato)
    }
    self.patologiasDescargadas()
}
}
else {
    print("Error al descargar las patologias. \!(error!)")
}
}
}
}

func patologiasDescargadas(){
    arrayPatologia.sort(by: { ($0.nombre) < ($1.nombre)})
    cargadoPatologia = true
    tvPatologias.reloadData()
}

@available (iOS 11, *)
deinit {
    // Later You Can Remove Observer
    self.observer?.invalidate()
}

@objc func cambiaValorBlanco(){
    cargarBlanco = true
}

func webView(_ webView: WKWebView, didFail navigation: WKNavigation!, withError error: Error) {
    print("elErrorNav")
    print(error)
}

func webViewWebContentProcessDidTerminate(_ webView: WKWebView) {
    print("WebContentProcessDidTerminate")
}

func webView(_ webView: WKWebView, didFailProvisionalNavigation navigation: WKNavigation!,
withError error: Error) {
    print("didFailProvisionalNavigation")
    print(error)
}

func webView(_ webView: WKWebView, didStartProvisionalNavigation navigation: WKNavigation!) {
    print("didStartProvisionalNavigation")
}

func webView(_ webView: WKWebView, didReceiveServerRedirectForProvisionalNavigation
navigation: WKNavigation!) {
    print("didReceiveServerRedirectForProvisionalNavigation")
}

override func viewDidDisappear(_ animated: Bool) {

```



```

class PatologiasVC: UIViewController, UITableViewDelegate, UITableViewDataSource,
descripcionProtocol, UINavigationControllerDelegate, UIScrollViewDelegate,
UIImagePickerControllerDelegate, UIDocumentPickerDelegate {

    func agregarAporte(btn: UIButton) {
        let alert = UIAlertController(title: "Seleccionar fuente", message: "Que deseas hacer?", preferredStyle:
UIAlertController.Style.actionSheet)
        let cancelAction = UIAlertAction(title: "Cancelar", style: .cancel) { (action:UIAlertAction!) in }
        let tomarAction = UIAlertAction(title: "Tomar Foto", style: .default) { (action:UIAlertAction!) in
            self.imagePicker.sourceType = .camera
            self.present(self.imagePicker, animated: true, completion: nil)
        }
        let seleccionarAction = UIAlertAction(title: "Seleccionar del carrete", style: .default) {
(action:UIAlertAction!) in
            self.imagePicker.sourceType = .photoLibrary
            self.present(self.imagePicker, animated: true, completion: nil)
        }
        alert.addAction(tomarAction)
        alert.addAction(seleccionarAction)
        alert.addAction(cancelAction)
        alert.popoverPresentationController?.sourceView = self.view
        self.present(alert, animated: true, completion: nil)
    }

    // MARK UIImagePickerControllerDelegate

    func imagePickerController(_ picker: UIImagePickerController, didFinishPickingMediaWithInfo info:
[UIImagePickerController.InfoKey : Any]) {
        // Local variable inserted by Swift 4.2 migrator.
        let info = convertFromUIImagePickerControllerInfoKeyDictionary(info)

        imagePicker.dismiss(animated: true, completion: nil)
        var img =
(info[convertFromUIImagePickerControllerInfoKey(UIImagePickerController.InfoKey.originalImage)] as?
UIImage)!
        var datos = img.jpegData(compressionQuality: 0.0)!
        print(datos.count)

        if datos.count>250000{
            img = img.resizeWithPercentage(percentage: 0.35)!
            datos = img.jpegData(compressionQuality: 0.0)!
            print(datos.count)
            self.agregarFotoEnParse(imagen: img, calidad: 0.0)
        }
        else{
            var calidad = CGFloat(1.0)
            datos = img.jpegData(compressionQuality: calidad)!
            while datos.count>200000 && calidad >= 0.0{
                calidad = calidad - 0.1
                datos = img.jpegData(compressionQuality: calidad)!
            }
            self.agregarFotoEnParse(imagen: img, calidad: calidad)
        }
    }

    func agregarFotoEnParse(imagen:UIImage, calidad: CGFloat){

        let dateFormatter = DateFormatter()
        dateFormatter.dateFormat = "yyyyMMddHHmmss"
    }
}

```

```

let datos = imagen.jpegData(compressionQuality: calidad)!
let aporte = PFObjeto(className:"PATO_Aporte")
aporte["idPatologia"] = PFObjeto(withoutDataWithClassName: "PATO_Patologia",
objectId:patoLocal.seleccionada.objectId)
aporte["aprobado"] = false
aporte["imagen"] = PFFileObject(name:"\\(dateFormatter.string(from:Date())).jpg", data: datos)
aporte.saveInBackground {
    (success: Bool, error: Error?) in
    if (success) {
        let alert = UIAlertController(title: "Gracias por tu aporte", message: "Recibimos tu aporte, este será
visible una vez que lo hayamos aprobado", preferredStyle: UIAlertController.Style.alert)
        alert.addAction(UIAlertAction(title: "Aceptar", style: UIAlertAction.Style.default, handler: nil))
        self.present(alert, animated: true, completion: nil)
    } else {
        let alert = UIAlertController(title: "Error al grabar la foto", message: "\\(error!) ", preferredStyle:
UIAlertController.Style.alert)
        alert.addAction(UIAlertAction(title: "Aceptar", style: UIAlertAction.Style.default, handler: nil))
        alert.popoverPresentationController?.sourceView = self.view
        alert.popoverPresentationController?.sourceRect = CGRect(x: self.view.center.x, y:
self.view.center.y, width: 64, height: 64)
        self.present(alert, animated: true, completion: nil)
    }
}
}

var imagePicker: UIImagePickerController!

func verFoto(btn: UIButton) {
    var images = [SKPhotoProtocol]()
    let cell = tvDescripcion.cellForRow(at: IndexPath(row: 0, section: 0)) as! celdaDescripcionPato
    let arrayArchivosImagen = [cell.ivImagen.image]
    for pic in arrayArchivosImagen {
        let photo = SKPhoto.photoWithImage(pic!)
        photo.caption = patoLocal.seleccionada.nombre
        images.append(photo)
    }
    SKPhotoBrowserOptions.displayBackAndForwardButton = false
    let browser = SKPhotoBrowser(photos: images)
    browser.initializePageIndex(0)
    present(browser, animated: true, completion: {})
}

struct patoLocal {
    static var seleccionada:Patologia!
}

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return 1
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier:"CeldaDescripcionPato") as!
celdaDescripcionPato

    if patoLocal.seleccionada.imagen == nil {
        if (patoLocal.seleccionada.imagenes != nil) {
            if let url = URL(string: patoLocal.seleccionada.imagenes!) {

```

```

        let task = URLSession.shared.dataTask(with: url) { data, response, error in
            guard let data = data, error == nil else { return }

            DispatchQueue.main.async { /// execute on main thread
                cell.ivImagen.image = UIImage(data: data)
            }
        }
        task.resume()
    }
}
else{
    //no tiene imagen la pato!
}
}
cell.delegate = self
cell.lblNombre.text = patoLocal.seleccionada.nombre
cell.btnAporte.setTitle(patoLocal.seleccionada.objectId, for: .disabled)

let attributsBold = [NSAttributedString.Key.font : UIFont.systemFont(ofSize: 14, weight: .bold)]
let attributsNormal = [NSAttributedString.Key.font : UIFont.systemFont(ofSize: 14, weight: .regular)]
var boldStringPart = NSMutableAttributedString(string: "Descripción: ", attributes:attributsBold)
var attributedString = NSMutableAttributedString(string:
"\(patoLocal.seleccionada.descripcion.trimmingCharacters(in: .whitespacesAndNewlines))",
attributes:attributsNormal)
    boldStringPart.append(attributedString)
    cell.lblDescripcion.attributedString = boldStringPart

    boldStringPart = NSMutableAttributedString(string: "Referencias: ", attributes:attributsBold)
    attributedString = NSMutableAttributedString(string:
"\(patoLocal.seleccionada.referencia.trimmingCharacters(in: .whitespacesAndNewlines))",
attributes:attributsNormal)
    boldStringPart.append(attributedString)
    cell.lblReferencias.attributedString = boldStringPart

    cell.lblUriReferencias.text = "\((patoLocal.seleccionada.urlReferencia.trimmingCharacters(in:
.whitespacesAndNewlines))"

    cell.lblUriReferencias.handleURLTap { laUrl in
        UIApplication.shared.open(laUrl)
    }
    cell.btnAporte.layer.borderWidth = 1.0
    cell.btnAporte.layer.borderColor = UIColor.white.cgColor
    cell.btnAporte.layer.cornerRadius = 10
    cell.btnAporte.layer.masksToBounds = true

    if patoLocal.seleccionada.prevencion != nil {
        boldStringPart = NSMutableAttributedString(string: "Prevencion: ", attributes:attributsBold)
        attributedString = NSMutableAttributedString(string:
"\(patoLocal.seleccionada.prevencion!.trimmingCharacters(in: .whitespacesAndNewlines))",
attributes:attributsNormal)
        boldStringPart.append(attributedString)
        cell.lblPrevencion.attributedString = boldStringPart
    }
    else {
        cell.lblPrevencion.text = ""
    }
    if patoLocal.seleccionada.tratamiento != nil {
        boldStringPart = NSMutableAttributedString(string: "Tratamiento: ", attributes:attributsBold)

```

```

        attributedString = NSMutableAttributedString(string:
"\(patoLocal.seleccionada.tratamiento!.trimmingCharacters(in: .whitespacesAndNewlines))",
attributes:attributsNormal)
        boldStringPart.append(attributedString)
        cell.lblTratamiento.attributedString = boldStringPart
    }
    else {
        cell.lblTratamiento.text = ""
    }

    boldStringPart = NSMutableAttributedString(string: "Equipo de medición: ", attributes:attributsBold)
    attributedString = NSMutableAttributedString(string:
"\(patoLocal.seleccionada.equipoMedicion.trimmingCharacters(in: .whitespacesAndNewlines))",
attributes:attributsNormal)
    boldStringPart.append(attributedString)
    cell.lblEquipoMedicion.attributedString = boldStringPart

    boldStringPart = NSMutableAttributedString(string: "Técnica de reconocimiento: ",
attributes:attributsBold)
    attributedString = NSMutableAttributedString(string:
"\(patoLocal.seleccionada.tecnicaReconocimiento.trimmingCharacters(in: .whitespacesAndNewlines))",
attributes:attributsNormal)
    boldStringPart.append(attributedString)
    cell.lblTecnicaReconocimiento.attributedString = boldStringPart

    return cell
}

@IBOutlet weak var tvDescripcion: UITableView!

@IBAction func accionRegresar(_ sender: Any) {
    self.dismiss(animated: true, completion: {
    })
}

override func viewDidLoad() {
    super.viewDidLoad()

    // Do any additional setup after loading the view.
    tvDescripcion.register(UINib(nibName: "celdaDescripcionPato", bundle: nil), forCellReuseIdentifier:
"CeldaDescripcionPato")
    tvDescripcion.tableView.footerView = UIView(frame: CGRect.zero)
    imagePicker = UIImagePickerController()
    imagePicker.delegate = self
}

/*
// MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    // Get the new view controller using segue.destination.
    // Pass the selected object to the new view controller.
}
*/
}

```

```

// Helper function inserted by Swift 4.2 migrator.
fileprivate func convertFromUIImagePickerControllerInfoKeyDictionary(_ input:
[UIImagePickerController.InfoKey: Any]) -> [String: Any] {
    return Dictionary(uniqueKeysWithValues: input.map {key, value in (key.rawValue, value)})
}

// Helper function inserted by Swift 4.2 migrator.
fileprivate func convertFromUIImagePickerControllerInfoKey(_ input: UIImagePickerController.InfoKey) -
> String {
    return input.rawValue
}

extension UIImage {

    func isEqualToImage(image: UIImage) -> Bool {
        let data1: NSData = self.pngData()! as NSData
        let data2: NSData = image.pngData()! as NSData
        return data1.isEqual(data2)
    }

    func resizeWithPercentage(percentage: CGFloat) -> UIImage? {
        let imageView = UIImageView(frame: CGRect(origin: .zero, size: CGSize(width: size.width *
percentage, height: size.height * percentage)))
        imageView.contentMode = .scaleAspectFit
        imageView.image = self
        UIGraphicsBeginImageContextWithOptions(imageView.bounds.size, false, scale)
        guard let context = UIGraphicsGetCurrentContext() else { return nil }
        imageView.layer.render(in: context)
        guard let result = UIGraphicsGetImageFromCurrentImageContext() else { return nil }
        UIGraphicsEndImageContext()
        return result
    }

    func resizeWithWidth(width: CGFloat) -> UIImage? {
        let imageView = UIImageView(frame: CGRect(origin: .zero, size: CGSize(width: width, height:
CGFloat(ceil(width/size.width * size.height))))
        imageView.contentMode = .scaleAspectFit
        imageView.image = self
        UIGraphicsBeginImageContextWithOptions(imageView.bounds.size, false, scale)
        guard let context = UIGraphicsGetCurrentContext() else { return nil }
        imageView.layer.render(in: context)
        guard let result = UIGraphicsGetImageFromCurrentImageContext() else { return nil }
        UIGraphicsEndImageContext()
        return result
    }
}
}

```

- **Búsqueda por características**

Permite a los usuarios de la aplicación, que previamente se haya realizado una inspección visual de la patología, ingresar ciertas características propias para obtener los

resultados precisos. Esta búsqueda facilita y permite obtener resultados de acuerdo a las características que seleccione el usuario.

Código de pantalla:

```
//
// CaracteristicasVC.swift
// Patologias
//
// Created on 11/8/21.
//

import UIKit
import Parse
import RKDropdownAlert

class CaracteristicasVC: UIViewController, UITableViewDelegate, UITableViewDataSource {

    struct idEtiqueta {
        static var seleccionada:String!
    }

    @IBAction func accionMostrar(_ sender: UIButton) {

        if arrayPatosResultantes.count > 0 {
            if arrayPatosResultantes.count == 1 {
                PatologiasVC.patoLocal.seleccionada = arrayPatosResultantes.first
                let patologiasVC = self.storyboard?.instantiateViewController(withIdentifier:
"identificadorPatologiasVC") as! PatologiasVC
                self.present(patologiasVC, animated: true, completion: { })
            }
            else {
                let alert = UIAlertController(title: "Se encontraron varias patologías", message: "Cual deseas
visualizar?", preferredStyle: UIAlertController.Style.actionSheet)
                let cancelAction = UIAlertAction(title: "Cancelar", style: .cancel) { (action:UIAlertAction!) in }
                for pato in arrayPatosResultantes {
                    let accion = UIAlertAction(title: pato.nombre, style: .default) { (action:UIAlertAction!) in
                        PatologiasVC.patoLocal.seleccionada = pato
                        let patologiasVC = self.storyboard?.instantiateViewController(withIdentifier:
"identificadorPatologiasVC") as! PatologiasVC
                        self.present(patologiasVC, animated: true, completion: { })
                    }
                    alert.addAction(accion)
                }
                alert.addAction(cancelAction)
                alert.popoverPresentationController?.sourceView = self.view
                self.present(alert, animated: true, completion: nil)
            }
        }
    }

    @IBOutlet weak var btnRight: UIButton!
    let localNotification = UINotificationFeedbackGenerator()
    @IBOutlet weak var tvCaracteristicas: UITableView!
    var arrayPatologias:[Patologia] = []
    var arrayCaracteristicas:[String] = []
    var arrayPatosResultantes:[Patologia] = []
}
```

```

var cargadoCaracteristicas = false
@IBOutlet weak var lblResultados: UILabel!
var arrIndSeleccionados:[Int] = []

@IBAction func accionRegresar(_ sender: Any) {
    self.dismiss(animated: true, completion: {
    })
}

func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return arrayCaracteristicas.count
}

func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier:"CeldaCaracteristica") as!
celdaCaracteristicaTVC
    if cargadoCaracteristicas {
        let carac = arrayCaracteristicas[indexPath.row]
        cell.lblCaracteristica.text = carac
        if arrIndSeleccionados.contains(indexPath.row) {
            cell.contentView.backgroundColor = UIColor.lightGray
        }
        else {
            cell.contentView.backgroundColor = UIColor.white
        }
    }
    return cell
}

func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

    if !arrIndSeleccionados.contains(indexPath.row) {
        if (arrIndSeleccionados.count == 5) {
            RKDropDownAlert.title("Error", message: "Puedes seleccionar máximo 5 características",
background-color: UIColor.red, textColor: UIColor.white, time: 3)
            self.localNotification.notificationOccurred(.error)
        }
        else{
            arrIndSeleccionados.append(indexPath.row)
        }
    }
    else {
        arrIndSeleccionados.remove(at: arrIndSeleccionados.firstIndex(of: indexPath.row)!)
    }
    tvCaracteristicas.reloadData()
    if arrIndSeleccionados.count > 0 {
        self.generarResultados()
    }
    else {
        lblResultados.text = "Selecciona al menos dos características"
        btnRight.isHidden = true
    }
}

func generarResultados(){

    arrayPatosResultantes.removeAll()
    arrayPatosResultantes = []
    var arrCaracSeleccionadas:[String] = []

```

```

var i = 0
for car in arrayCaracteristicas {
    if arrIndSeleccionados.contains(i) {
        arrCaracSeleccionadas.append(car)
    }
    i+=1
}

for pato in arrayPatologias {
    var contieneCar = true
    for car in arrCaracSeleccionadas {
        if contieneCar {
            if !patO.caracteristicas.contains(car) {
                contieneCar = false
            }
        }
    }
    if contieneCar {
        arrayPatosResultantes.append(pato)
    }
}

for pato in arrayPatosResultantes {
    print(pato.nombre)
}

if arrayPatosResultantes.count > 0 {
    var texto = "Se ha(n) encontrado \(arrayPatosResultantes.count) coincidencia(s)"
    if arrIndSeleccionados.count < 2 {
        texto = "Se ha encontrado \(arrayPatosResultantes.count) coincidencia"
        texto = texto + "\nSelecciona al menos dos características"
    }
    lblResultados.text = texto
    btnRight.isHidden = false
}
else {
    lblResultados.text = "No se encontraron coincidencias"
    btnRight.isHidden = true
}
}

override func viewDidLoad() {
    super.viewDidLoad()

    tvCaracteristicas.register(UINib(nibName: "celdaCaracteristica", bundle: nil), forCellReuseIdentifier:
"CeldaCaracteristica")
    tvCaracteristicas.tableFooterView = UIView(frame: CGRect.zero)
    descargarPatologias()

    // Do any additional setup after loading the view.
}

func descargarPatologias(){

    let query = PFQuery(className:"PATO_Patologia")
    query.whereKey("idEtiqueta", equalTo: PFObject(withoutDataWithClassName:"PATO_Etiqueta",
objectId:idEtiqueta.seleccionada))
    query.findObjectsInBackground { (objects: [PFObject]?, error: Error?) in

```

```

DispatchQueue.main.async {
    if error == nil {
        if (objects?.count)! > 0 {
            let cantPatos = objects?.count
            for object in objects! {

                let idEtiqueta = (object["idEtiqueta"] as? PFOBJECT)?.objectId!
                let nombre = object["nombre"] as! String
                let desc = object["descripcion"] as! String
                let ref = object["referencia"] as! String
                let urlRef = object["urlReferencia"] as! String
                let equ = object["equipoMedicion"] as! String
                let tec = object["tecnicaReconocimiento"] as! String
                let trat = object["tratamiento"] as? String
                let prev = object["prevencion"] as? String
                var imgs:String? = nil
                if (object["imagen"] != nil) {
                    imgs = (object["imagen"] as? PFFileObject)?.url!
                }
                let carac = object["caracteristicas"] as! String
                let pato = Patologia(objectId: object.objectId!, nombre: nombre, descripcion: desc,
idEtiqueta: idEtiqueta, referencia: ref, urlReferencia: urlRef, equipoMedicion: equ, tecnicaReconocimiento:
tec, caracteristicas: carac, tratamiento: trat, prevencion: prev, imagenes: imgs)
                self.arrayPatologias.append(pato)
            }
            self.patologiasDescargadas()
        }
    }
    else {
        print("Error al descargar las patologias. \((error!)\)")
    }
}
}

func patologiasDescargadas(){
    arrayPatologias.sort(by: { ($0.nombre) < ($1.nombre)})
    if arrayPatologias.count>0 {
        lblResultados.text = "Selecciona al menos dos características"
    }
    cargarCaracteristicas()
}

func cargarCaracteristicas(){
    for pato in arrayPatologias {
        let todasLasCarac = pato.caracteristicas
        let arrCaracteristicas:[String]! = todasLasCarac.split{$0 == "*"}.map(String.init)
        for carac in arrCaracteristicas {
            let c = carac.trimmingCharacters(in: .whitespacesAndNewlines)
            if !arrayCaracteristicas.contains(c) {
                arrayCaracteristicas.append(c)
            }
        }
    }
    arrayCaracteristicas.sort(by: { $0 < $1 })
    cargadoCaracteristicas = true
    tvCaracteristicas.reloadData()
}
}

```

```

/*
// MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    // Get the new view controller using segue.destination.
    // Pass the selected object to the new view controller.
}
*/
}

```

- **Resultados y descripción del tipo de patología**

En este apartado permite observar los resultados de las búsquedas mediante imágenes y características, en los resultados de búsqueda se apreciarán las descripciones de las características, los equipos de medición, los tratamientos y las técnicas de reconocimiento. Cualquiera que sea el modo de búsqueda los resultados serán los mismos.

Código de pantalla

```

//
// celdaDescripcionPato.swift
// Patologias
//
// Created on 12/8/21.
//

import UIKit
import ActiveLabel

protocol descripcionProtocol : NSObjectProtocol {
    func verFoto(btn: UIButton) -> Void
    func agregarAporte(btn: UIButton) -> Void
}

class celdaDescripcionPato: UITableViewCell {

    weak var delegate: descripcionProtocol?
    @IBOutlet weak var btnAporte: UIButton!
    @IBOutlet weak var ivImagen: UIImageView!
    @IBOutlet weak var lblNombre: UILabel!
    @IBOutlet weak var lblDescripcion: UILabel!
    @IBOutlet weak var lblReferencias: UILabel!
    @IBOutlet weak var lblUrlReferencias: ActiveLabel!
    @IBOutlet weak var lblEquipoMedicion: UILabel!

```

```

@IBOutlet weak var lblTratamiento: UILabel!
@IBOutlet weak var lblPrevencion: UILabel!
@IBOutlet weak var lblTecnicaReconocimiento: UILabel!
override func awakeFromNib() {
    super.awakeFromNib()
    // Initialization code
}

override func setSelected(_ selected: Bool, animated: Bool) {
    super.setSelected(selected, animated: animated)

    // Configure the view for the selected state
}
@IBAction func agregarAporte(_ sender: UIButton) {
    if delegate?.responds(to: Selector(("agregarAporte:"))) != nil {
        delegate?.agregarAporte(btn:sender)
    }
}

@IBAction func verFoto(_ sender: UIButton) {
    if delegate?.responds(to: Selector(("verFoto:"))) != nil {
        delegate?.verFoto(btn:sender)
    }
}
}

```

- **Aportes de los usuarios**

En este apartado se crea una opción para que, el usuario tenga la posibilidad de subir imágenes que estén relacionadas con los tipos de patologías, es decir el usuario puede subir fotografías o imágenes que crea convenientes según el tipo de patología, esto ayudara aumentar la base de datos y posteriormente tener mejores resultados de búsqueda.

Código de funciones de pantalla:

```

// AportesVC.swift
// Patologias
//
// Created by Jorge Contreras, Adrian Castro on 23/8/21.
//

import UIKit
import Parse
import SKPhotoBrowser

class AportesVC: UIViewController, UICollectionViewDelegate, UICollectionViewDataSource, UICollectionViewDelegateFlowLayout {

```

```

var arrayArrayAportes:[[Aporte]] = []
var todosLosAportes:[Aporte] = []
var todasLasPatos:[String] = []
var cargadoAportes = false
@IBOutlet weak var cvAportes: UICollectionView!
var dateFormatter = DateFormatter()

@IBAction func accionCerrar(_ sender: UIButton) {
    self.dismiss(animated: true, completion: {
    })
}

func collectionView(_ collectionView: UICollectionView, numberOfItemsInSection section: Int) -> Int {
    return cargadoAportes ? arrayArrayAportes[section].count : 0
}

func collectionView(_ collectionView: UICollectionView, didSelectItemAt indexPath: IndexPath) {
    var images = [SKPhotoProtocol]()
    let image = arrayArrayAportes[indexPath.section][indexPath.row].imagen != nil ?
UIImage(data:arrayArrayAportes[indexPath.section][indexPath.row].imagen!) : nil
    let arrayArchivosImagen = [image]
    if image != nil {
        for pic in arrayArchivosImagen {
            let photo = SKPhoto.photoWithImage(pic!)
            let fechaFormateada:String =
dateFormatter.string(from:arrayArrayAportes[indexPath.section][indexPath.row].fecha).uppercased().replaci
ngOccurrences(of: ".", with: "/")
            photo.caption = "Aporte: \(fechaFormateada)"
            images.append(photo)
        }
    }
    SKPhotoBrowserOptions.displayBackAndForwardButton = false
    let browser = SKPhotoBrowser(photos: images)
    browser.initializePageIndex(0)
    present(browser, animated: true, completion: {})
}

func numberOfSections(in collectionView: UICollectionView) -> Int {
    return cargadoAportes ? arrayArrayAportes.count : 1
}

func collectionView(_ collectionView: UICollectionView, layout collectionViewLayout:
UICollectionViewLayout, referenceSizeForHeaderInSection section: Int) -> CGSize {
    return CGSize(width: collectionView.frame.width, height: 40)
}

func collectionView(_ collectionView: UICollectionView, cellForItemAt indexPath: IndexPath) ->
UICollectionViewCell {
    let cell = collectionView.dequeueReusableCell(withReuseIdentifier: "CeldaAporte", for: indexPath) as!
CeldaAporte
    if cargadoAportes {
        let aporte = arrayArrayAportes[indexPath.section][indexPath.row]
        if (aporte.imagen != nil) {
            cell.ivAporte.image = UIImage(data:(aporte.imagen!))
        }
        let fechaFormateada:String =
dateFormatter.string(from:aporte.fecha).uppercased().replacingOccurrences(of: ".", with: "/")
        cell.lblFechaAporte.text = "Aporte: \(fechaFormateada)"
        cell.contentView.layer.borderWidth = 1.0
    }
}

```

```

        cell.contentView.layer.borderColor = UIColor.black.cgColor
        //cell.contentView.layer.cornerRadius = 20
        cell.contentView.layer.masksToBounds = true
    }
    return cell
}

func collectionView(_ collectionView: UICollectionView, viewForSupplementaryElementOfKind kind:
String, at indexPath: IndexPath) -> UICollectionViewCell {
    if kind == UICollectionView.elementKindSectionHeader {
        let sectionHeader = collectionView.dequeueReusableSupplementaryView(ofKind: kind,
withReuseIdentifier: "header", for: indexPath) as! SectionHeader
        sectionHeader.label.text = cargadoAportes ? arrayArrayAportes[indexPath.section].first?.nombre :
"CARGANDO..."
        return sectionHeader
    } else { //No footer in this case but can add option for that
        return UICollectionViewCell()
    }
}

override func viewDidLoad() {
    super.viewDidLoad()

    // Do any additional setup after loading the view.
    dateFormatter.dateFormat = "dd.MMM,yyyy"
    cvAportes.register(UINib(nibName: "celdaAporte", bundle: nil), forCellWithReuseIdentifier:
"CeldaAporte")
    self.cvAportes.register(SectionHeader.self, forSupplementaryViewOfKind:
UICollectionView.elementKindSectionHeader, withReuseIdentifier: "header")
    descargarAportes()
}

func descargarAportes(){

    let query = PFQuery(className:"PATO_Aporte")
    //query.whereKey("idEtiqueta", equalTo: PFObjeto(withoutDataWithClassName:"PATO_Etiqueta",
objectId:idEtiqueta.seleccionada))
    query.whereKey("aprobado", equalTo: true)
    query.includeKey("idPatologia")
    query.findObjectsInBackground { (objects: [PFObjeto]?, error: Error?) in

        DispatchQueue.main.async {
            if error == nil {
                if (objects?.count)! > 0 {
                    let cantAportes = objects?.count
                    for object in objects! {
                        print(object)
                        let pato = (object["idPatologia"] as! PFObjeto)
                        let idPatologia = (object["idPatologia"] as! PFObjeto).objectId!
                        if !(self.todasLasPatos.contains(idPatologia)) {
                            self.todasLasPatos.append(idPatologia)
                        }
                        let nombre = pato["nombre"] as! String
                        let fecha = object.createdAt
                        let aporte = Aporte(objectId: object.objectId!, idPatologia: idPatologia, nombre: nombre,
fecha: fecha!)
                        if aporte.imagen == nil {
                            if (object["imagen"] != nil) {
                                (object["imagen"] as? PFFileObjeto)?.getDataInBackground({ (data:Data?,
imgError:Error?) in

```

```

        DispatchQueue.main.async {
            if imgError == nil && data != nil{
                aporte.setImagen(imagen: data!)
                self.cvAportes.reloadData()
            }
        }
    })
}
else{
    //no tiene fondo la etiqueta!
}
}
self.todosLosAportes.append(aporte)
}
self.aportesDescargados()
}
}
else {
    print("Error al descargar las aportes. \((error!)\)")
}
}
}
}

func aportesDescargados(){

    for idPato in todasLasPatos {
        var aportesPato:[Aporte] = []
        for aporte in todosLosAportes {
            if aporte.idPatologia == idPato {
                aportesPato.append(aporte)
            }
        }
        arrayArrayAportes.append(aportesPato)
    }
    cargadoAportes = true
    cvAportes.reloadData()
}

/*
// MARK: - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation
override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
    // Get the new view controller using segue.destination.
    // Pass the selected object to the new view controller.
}
*/
}

class SectionHeader: UICollectionViewReusableView {
    var label: UILabel = {
        let label: UILabel = UILabel()
        label.textColor = .white
        label.font = UIFont.systemFont(ofSize: 15, weight: .semibold)
        label.sizeToFit()
        return label
    }()
}

```

```
override init(frame: CGRect) {
    super.init(frame: frame)

    addSubview(label)

    label.translatesAutoresizingMaskIntoConstraints = false
    label.topAnchor.constraint(equalTo: self.topAnchor, constant: 10).isActive = true
    label.leftAnchor.constraint(equalTo: self.leftAnchor, constant: 10).isActive = true
    label.rightAnchor.constraint(equalTo: self.rightAnchor).isActive = true
}

required init?(coder aDecoder: NSCoder) {
    fatalError("init(coder:) has not been implemented")
}
}
```

APÉNDICE D

4.5 Código kotlin para funcionamiento de aplicación en el sistema Android

- **Ver aportes de usuarios**

Se presenta el código que muestra en una pantalla, las imágenes que dejaron los aportes al realizar la búsqueda de manera exitosa, estas de reflejan de manera ordenada y por tipo de patología.

```
package com.chapps.patologias

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.Button
import android.widget.ImageView
import android.widget.TextView
import androidx.appcompat.app.AlertDialog
import androidx.recyclerview.widget.GridLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.bumptech.glide.Glide
import com.chapps.patologias.adapters.GridItemAdapter
import com.chapps.patologias.base.GridRVContainer
import com.stfalcon.imageviewer.StfalconImageViewer
import com.stfalcon.imageviewer.loader.ImageLoader
import java.util.*

class AportesActivity : AppCompatActivity() {
    private var listCompleto = mutableListOf<GridRVContainer>()
    private var listTodosLosAportes = mutableListOf<aporte>()
    private var listEtiquetas = mutableListOf<etiqueta>()
    private var listEtiquetasNombres = mutableListOf<String>()
    private lateinit var rvAportes: RecyclerView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_aportes)

        rvAportes = findViewById(R.id.rv_aportes)
        findViewById<Button>(R.id.btn_cerrar_aportes).setOnClickListener { finish() }
        utilidad().descargarEtiquetas() { result ->
            for (obj in result) {
                listEtiquetas.add(obj)
            }
            listEtiquetas.sortByDescending { it.nombre }
            listEtiquetas.reverse()
        }
    }
}
```

```

    for (item in listEtiquetas){
        listEtiquetasNombres.add(item.nombre)
    }
}
findViewById<Button>(R.id.btn_seleccionar_material).setOnClickListener {
    val builder = AlertDialog.Builder(this, R.style.AlertDialogTheme)
    builder.setTitle("Que aportes deseas visualizar?")
    builder.setItems(listEtiquetasNombres.toArray()) { dialog, which ->
        utilidad().descargarAportesDe(listEtiquetas[which].objectId) { result ->
            findViewById<TextView>(R.id.txt_material).text = listEtiquetasNombres[which]
            listTodosLosAportes = mutableListOf()
            listCompleto = mutableListOf()
            for (obj in result) {
                listTodosLosAportes.add(obj)
            }
            listTodosLosAportes.sortBy { it.nombre }
            aportesDescargados()
        }
    }
    val dialog = builder.create()
    dialog.show()
}
}

fun aportesDescargados(){
    if (listTodosLosAportes.isEmpty()){
        rvAportes.visibility = View.GONE
        findViewById<TextView>(R.id.txt_cargando).text = "No se encontró ningún aporte"
    }
    else{
        findViewById<TextView>(R.id.txt_cargando).text = ""
        rvAportes.visibility = View.VISIBLE
        var ultNombre = listTodosLosAportes.first().nombre
        val obj = GridRVContainer(null, true, ultNombre)
        listCompleto.add(obj)
        for (apo in listTodosLosAportes){
            var currNombre = apo.nombre
            if (currNombre != ultNombre) {
                val obj = GridRVContainer(null, true, currNombre)
                listCompleto.add(obj)
                ultNombre = currNombre
            }
            val obj = GridRVContainer(apo, false, null)
            listCompleto.add(obj)
        }

        val adapterGrid = GridItemAdapter(listCompleto) { item, pos ->
            if (item.myObject != null) {
                var prov = java.util.ArrayList(Arrays.asList(""))
                StfalconImageViewer.Builder<String>(
                    this@AportesActivity,
                    prov,
                    ImageLoader<String?> { imageView, image ->
                        Glide.with(this@AportesActivity).load(item.myObject!!.urlImagen).into(imageView)
                    }).show()
            }
        }
        val gridLayout = GridLayoutManager(this, 2)
        rvAportes.layoutManager = gridLayout
        rvAportes.adapter = adapterGrid
    }
}

```



```

private val INPUT_FILE_REQUEST_CODE = 1
private val FILECHOOSER_RESULTCODE = 1

@Throws(IOException::class)
private fun createImageFile(): File? {
    // Create an image file name
    val timeStamp: String = SimpleDateFormat("yyyyMMdd_HHmmss").format(Date())
    val imageFileName = "JPEG_" + timeStamp + "_"
    val storageDir = Environment.getExternalStoragePublicDirectory(
        Environment.DIRECTORY_PICTURES
    )
    return File.createTempFile(
        imageFileName, /* prefix */
        ".jpg", /* suffix */
        storageDir /* directory */
    )
}

override fun onShowFileChooser(view: WebView?, filePath: ValueCallback<Array<Uri>>?,
fileChooserParams: FileChooserParams?
): Boolean {
    // Double check that we don't have any existing callbacks
    if (mFilePathCallback != null) {
        mFilePathCallback!!.onReceiveValue(null)
    }

    mFilePathCallback = filePath
    var takePictureIntent: Intent? = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
    if (takePictureIntent!!.resolveActivity(Navegador().applicationContext.packageManager) != null) {
        // Create the File where the photo should go
        var photoFile: File? = null
        try {
            photoFile = createImageFile()
            takePictureIntent.putExtra("PhotoPath", mCameraPhotoPath)
        } catch (ex: IOException) {
            // Error occurred while creating the File
            Log.e("ErrorCreatingFile", "Unable to create Image File", ex)
        }

        // Continue only if the File was successfully created
        if (photoFile != null) {
            mCameraPhotoPath = "file:" + photoFile.getAbsolutePath()
            takePictureIntent.putExtra(
                MediaStore.EXTRA_OUTPUT,
                Uri.fromFile(photoFile)
            )
        } else {
            takePictureIntent = null
        }
    }

    val contentSelectionIntent = Intent(Intent.ACTION_GET_CONTENT)
    contentSelectionIntent.addCategory(Intent.CATEGORY_OPENABLE)
    contentSelectionIntent.type = "image/*"
    val intentArray: Array<Intent?>
    intentArray = takePictureIntent?.let { arrayOf(it) } ?: arrayOfNulls(0)
    val chooserIntent = Intent(Intent.ACTION_CHOOSER)
    chooserIntent.putExtra(Intent.EXTRA_INTENT, contentSelectionIntent)
    chooserIntent.putExtra(Intent.EXTRA_TITLE, "Image Chooser")
    chooserIntent.putExtra(Intent.EXTRA_INITIAL_INTENTS, intentArray)
    Navegador().startActivityForResult(chooserIntent, INPUT_FILE_REQUEST_CODE)
}

```

```

return true
}

// openFileChooser for Android 3.0+
fun openFileChooser(uploadMsg: ValueCallback<Uri>?, acceptType: String?) {
    mUploadMessage = uploadMsg
    // Create AndroidExampleFolder at sdcard
    // Create AndroidExampleFolder at sdcard
    val imageStorageDir = File(
        Environment.getExternalStoragePublicDirectory(
            Environment.DIRECTORY_PICTURES
        ), "AndroidExampleFolder"
    )
    if (!imageStorageDir.exists()) {
        // Create AndroidExampleFolder at sdcard
        imageStorageDir.mkdirs()
    }

    // Create camera captured image file path and name
    val file = File("" + imageStorageDir + File.separator.toString() + "IMG_" +
        System.currentTimeMillis().toString() + ".jpg")
    mCapturedImageURI = Uri.fromFile(file)

    // Camera capture image intent
    val captureIntent = Intent(
        MediaStore.ACTION_IMAGE_CAPTURE
    )
    captureIntent.putExtra(MediaStore.EXTRA_OUTPUT, mCapturedImageURI)
    val i = Intent(Intent.ACTION_GET_CONTENT)
    i.addCategory(Intent.CATEGORY_OPENABLE)
    i.type = "image/*"

    // Create file chooser intent
    val chooserIntent = Intent.createChooser(i, "Image Chooser")

    // Set camera intent to file chooser
    chooserIntent.putExtra(
        Intent.EXTRA_INITIAL_INTENTS, arrayOf<Parcelable>(captureIntent)
    )

    // On select image call onActivityResult method of activity
    Navegador().startActivityForResult(chooserIntent, FILECHOOSER_RESULTCODE)
}

// openFileChooser for Android < 3.0
fun openFileChooser(uploadMsg: ValueCallback<Uri>?) {
    openFileChooser(uploadMsg, "")
}

//openFileChooser for other Android versions
fun openFileChooser(
    uploadMsg: ValueCallback<Uri>?,
    acceptType: String?,
    capture: String?
) {
    openFileChooser(uploadMsg, acceptType)
}
}

```

- **Busqueda por características**

Esta basada en enlazar la base de datos general, que mediante la filtración de características propias de las patologías muestra el posible resultado de búsqueda.

```
package com.chapps.patologias

import android.annotation.SuppressLint
import android.content.Intent
import android.graphics.Color
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.view.View
import android.widget.Button
import android.widget.ImageView
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.recyclerview.selection.SelectionTracker
import androidx.recyclerview.selection.StableIdKeyProvider
import androidx.recyclerview.selection.StorageStrategy
import androidx.recyclerview.widget.DividerItemDecoration
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.chapps.patologias.adapters.CaracteristicaRecyclerAdapter
import com.chapps.patologias.adapters.MyLookup

class CaracteristicasActivity : AppCompatActivity() {

    var listPatologias = mutableListOf<patologia>()
    var listCaracteristicas = mutableListOf<String>()
    var listPatosResultantes = mutableListOf<patologia>()
    var listPatosResNombres:MutableList<String> = ArrayList()
    var listIndSeleccionados = mutableListOf<Int>()
    private var tracker: SelectionTracker<Long>? = null
    private lateinit var txtResultados:TextView
    private lateinit var ivFlechaResultados:ImageView
    private var eti:etiqueta? = null

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_caracteristicas)

        ivFlechaResultados = findViewById<ImageView>(R.id.iv_flecha_resultados)
        ivFlechaResultados.setColorFilter(Color.argb(255, 255, 255, 255))
        findViewById<Button>(R.id.btn_cerrar_aportes).setOnClickListener { finish() }
        txtResultados = findViewById<TextView>(R.id.txt_resultados)

        eti = intent.extras?.get("EXTRA_ETIQUETA") as etiqueta

        utilidad().descargarPatologias(eti!!.objectId) { result ->
            for (obj in result) {
                listPatologias.add(obj)
            }
        }
    }
}
```



```

my_rv.layoutManager = LinearLayoutManager(this)
my_rv.setHasFixedSize(true)

val adapter = CaracteristicaRecyclerAdapter(listCaracteristicas, this)
my_rv.adapter = adapter

tracker = SelectionTracker.Builder<Long>(
    "selection-1",
    my_rv,
    StableIdKeyProvider(my_rv),
    MyLookup(my_rv),
    StorageStrategy.createLongStorage()
).withSelectionPredicate(object : SelectionTracker.SelectionPredicate<Long>(){
    override fun canSelectMultiple():Boolean {
        return true
    }
    override fun canSetStateForKey(key: Long, nextState: Boolean): Boolean {
        if(nextState && tracker!!.selection.size() >= 5) { // 5 - max selection size
            Toast.makeText(applicationContext, "Selecciona máximo 5 características",
Toast.LENGTH_SHORT).show()
            return false // Can't select when 5 items selected
        }
        return true // When selection size < 5 - allow selection
    }
    override fun canSetStateAtPosition(position: Int, nextState: Boolean): Boolean {
        return true
    }
}).build()
adapter.setTracker(tracker)

tracker?.addObserver(
    object: SelectionTracker.SelectionObserver<Long>() {
        override fun onSelectionChanged() {
            val nItems:Int? = tracker?.selection?.size()

            if(nItems!=null && nItems > 0) {
                if (!listIndSeleccionados.isEmpty()) {
                    listIndSeleccionados = mutableListOf<Int>()
                }
                for (indice in tracker!!.selection) {
                    listIndSeleccionados.add(indice.toInt())
                }
                listIndSeleccionados.sortBy { it }
                if (listIndSeleccionados.size>0) {
                    generarResultados()
                }
            } else{
                txtResultados.text = "Selecciona al menos dos características"
                ivFlechaResultados.visibility = View.GONE
            }
        }

        } else {
            // Reset color and title to default values
            txtResultados.text = "Selecciona al menos dos características"
            ivFlechaResultados.visibility = View.GONE
            if (!listPatosResultantes.isEmpty()) {
                listPatosResultantes = mutableListOf()
                listPatosResNombres = mutableListOf()
            }
        }
    }
)

```

```

    }
    })

    findViewById<RecyclerView>(R.id.rv_caracteristicas).addItemDecoration(
        DividerItemDecoration(findViewById<RecyclerView>(R.id.rv_caracteristicas).getContext(),
            DividerItemDecoration.VERTICAL)
    )
}

fun generarResultados(){
    if (!listPatosResultantes.isEmpty()) {
        listPatosResultantes = mutableListOf<patologia>()
        listPatosResNombres = ArrayList()
    }
    var listCaracSeleccionadas = mutableListOf<String>()
    var i = 0
    for (car in listCaracteristicas) {
        if (listIndSeleccionados.contains(i)){
            listCaracSeleccionadas.add(car)
        }
        i+=1
    }

    for (pato in listPatologias) {
        var contieneCar = true
        for (car in listCaracSeleccionadas) {
            if (contieneCar) {
                if (!pato.caracteristicas.contains(car)){
                    contieneCar = false
                }
            }
        }
        if (contieneCar) {
            listPatosResultantes.add(pato)
        }
    }

    for (pato in listPatosResultantes) {
        if (!listPatosResNombres.contains(pato.nombre)){
            Log.d("DEBUG", "${pato.nombre}")
            listPatosResNombres.add(pato.nombre)
        }
    }

    if (listPatosResultantes.size > 0) {
        var texto = "Se ha(n) encontrado ${listPatosResultantes.size} coincidencia(s)"
        if (listIndSeleccionados.size < 2) {
            texto = "Se ha(n) encontrado ${listPatosResultantes.size} coincidencia"
            texto += "\nSelecciona al menos dos características"
        }
        txtResultados.text = texto
        ivFlechaResultados.visibility = View.VISIBLE
    }
    else {
        txtResultados.text = "No se encontraron coincidencias"
        ivFlechaResultados.visibility = View.GONE
    }
}
}

```

```

override fun onSaveInstanceState(outState: Bundle) {
    super.onSaveInstanceState(outState)

    if(outState != null)
        tracker?.onSaveInstanceState(outState)
    }
}

```

- **Dejar Aporte voluntario**

Usa la cámara y memoria del dispositivo, que permite subir las imágenes para luego almacenarlas en una base de datos externa, esta estará visible todo el tiempo en la aplicación.

```

package com.chapps.patologias

import android.Manifest
import android.R.attr
import android.app.Activity
import android.content.Intent
import android.content.pm.PackageManager
import android.graphics.Bitmap
import android.graphics.BitmapFactory
import android.media.Image
import android.net.Uri
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.os.Environment
import android.provider.MediaStore
import android.text.SpannedString
import android.util.Log
import android.view.View
import android.widget.Button
import android.widget.ImageView
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import androidx.core.app.ActivityCompat
import androidx.core.content.ContentProviderCompat.requireContext
import androidx.core.content.ContextCompat
import androidx.core.content.FileProvider
import androidx.core.text.bold
import androidx.core.text.buildSpannedString
import com.bumptech.glide.Glide
import com.denzcoskun.imageslider.ImageSlider
import com.denzcoskun.imageslider.interfaces.ItemClickListener
import com.denzcoskun.imageslider.models.SlideModel
import com.parse.ParseFile
import com.parse.ParseUser
import com.parse.SaveCallback
import com.stfalcon.imageviewer.StfalconImageViewer
import com.stfalcon.imageviewer.loader.ImageLoader

```

```

import java.io.ByteArrayOutputStream
import java.text.DateFormat
import java.text.SimpleDateFormat
import java.util.*
import kotlin.collections.ArrayList
import com.parse.ParseObject
import java.io.File
import java.io.IOException
import androidx.core.app.ActivityCompat.startActivityForResult
import androidx.core.app.ActivityCompat.startActivityForResult
import androidx.core.app.ActivityCompat.startActivityForResult

import android.content.ContentValues
import android.R.attr.thumbnail
import android.graphics.Matrix
import android.hardware.camera2.CameraCharacteristics
import android.hardware.camera2.CaptureRequest
import android.media.ExifInterface
import java.lang.Exception

class PatologiaActivity : AppCompatActivity() {

    private val pickImage = 100
    private var imageUri: Uri? = null
    private var pato:patologia? = null
    private val REQUEST_PERMISSION = 200
    private val REQUEST_IMAGE_CAPTURE = 1
    lateinit var currentPhotoPath: String

    @Throws(IOException::class)
    private fun createCapturedPhoto(): File {
        val timestamp: String = SimpleDateFormat("yyyyMMdd-HHmms", Locale.US).format(Date())
        val storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES)
        return File.createTempFile("PHOTO_${timestamp}", ".jpg", storageDir).apply {
            currentPhotoPath = absolutePath
        }
    }

    override fun onResume() {
        super.onResume()
        checkCameraPermission()
    }

    private fun openCamera() {

        val values = ContentValues()
        values.put(MediaStore.Images.Media.TITLE, "New Picture")
        values.put(MediaStore.Images.Media.DESCRPTION, "From your Camera")
        imageUri = contentResolver.insert(
            MediaStore.Images.Media.EXTERNAL_CONTENT_URI, values
        )
        val intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)
        intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri)
        startActivityForResult(intent, REQUEST_IMAGE_CAPTURE)

    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

```

```

setContent(R.layout.activity_patologia)

pato = intent.extras?.get("EXTRA_PATOLOGIA") as patologia
utilidad().descargarImgsPato(pato!!.objectId) { result ->
    val imageList = ArrayList<SlideModel>() // Create image list
    for (obj in result) {
        imageList.add(SlideModel(obj))
    }
    val imageSlider = findViewById<ImageSlider>(R.id.image_slider)
    imageSlider.setImageList(imageList)
    imageSlider.setItemClickListener(object : ItemClickListener {
        override fun onItemSelected(position: Int) {
            // You can listen here
            var prov = java.util.ArrayList(Arrays.asList(""))
            StfalconImageViewer.Builder<String>(
                this@PatologiaActivity,
                prov,
                ImageLoader<String?> { imageView, image ->
                    Glide.with(this@PatologiaActivity).load(result[position]).into(imageView)
                }).show()
        }
    })
}

val nombre = findViewById<TextView>(R.id.txt_pato_nombre)
val carac = findViewById<TextView>(R.id.txt_pato_caracteristicas)
val descrip = findViewById<TextView>(R.id.txt_pato_descripcion)
val tratam = findViewById<TextView>(R.id.txt_pato_tratamiento)
val equi = findViewById<TextView>(R.id.txt_pato_equi_med)
val tec = findViewById<TextView>(R.id.txt_pato_tec_reconocimiento)
val ref = findViewById<TextView>(R.id.txt_pato_referencias)
val lRef = findViewById<TextView>(R.id.txt_pato_links_ref)
nombre.text = pato!!.nombre
val stringCar: SpannedString = buildSpannedString { bold { append("Características: ") }
    append("${pato!!.caracteristicas}") }
carac.text = stringCar
val stringDes: SpannedString = buildSpannedString { bold { append("Descripción: ") }
    append("${pato!!.descripcion}") }
descrip.text = stringDes
val stringTrat: SpannedString = buildSpannedString { bold { append("Tratamiento: ") }
    append("${pato!!.tratamiento}") }
tratam.text = stringTrat
val stringEqui: SpannedString = buildSpannedString { bold { append("Equipo de medición: ") }
    append("${pato!!.equipoMedicion}") }
equi.text = stringEqui
val stringTec: SpannedString = buildSpannedString { bold { append("Técnica de reconocimiento: ") }
    append("${pato!!.tecnicaReconocimiento}") }
tec.text = stringTec
val stringRef: SpannedString = buildSpannedString { bold { append("Referencias: ") }
    append("${pato!!.referencia}") }
ref.text = stringRef
val stringLRef: SpannedString = buildSpannedString { bold { append("Links Referencias: ") }
    append("\n${pato!!.urlReferencia}") }
lRef.text = stringLRef

val btnCerrar = findViewById<Button>(R.id.btn_cerrar_patologia)
btnCerrar.setOnClickListener { finish() }

val btnAportar = findViewById<Button>(R.id.btn_dejar_aporte)
btnAportar.setOnClickListener {

```

```

val listOpciones = listOf<String>("Cámara","Galería")
val builder = androidx.appcompat.app.AlertDialog.Builder(this, R.style.AlertDialogTheme)
builder.setTitle("Selecciona una fuente para dejar el aporte")
builder.setItems(listOpciones.toArray()) { dialog, which ->
    when (which) {
        0 -> openCamera()
        1 -> {
            val gallery =
                Intent(Intent.ACTION_PICK, MediaStore.Images.Media.INTERNAL_CONTENT_URI)
            startActivityForResult(gallery, pickImage)
        }
    }
}
builder.setCancelable(true)
val dialog = builder.create()
dialog.show()

}

}

private fun checkCameraPermission() {
    if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) !=
        PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(this, arrayOf(Manifest.permission.CAMERA),
            REQUEST_PERMISSION)
    }
}

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    Log.i("onActivityResult: ", "requestCode=" + requestCode + ", resultCode=" + resultCode + ", data=" +
data);
    //callbackManager.onActivityResult(requestCode, resultCode, data)
    super.onActivityResult(requestCode, resultCode, data)

    if (resultCode == RESULT_OK && requestCode == REQUEST_IMAGE_CAPTURE) {
        if (requestCode == REQUEST_IMAGE_CAPTURE) {

            try {
                val btM = MediaStore.Images.Media.getBitmap(
                    contentResolver, imageUri
                )
                Log.d("DEBUG", "btm.byteCount: ${btM.byteCount}")
                val baComp = getCompressedBitmapData(btM, 360000, 1000)
                val cbtMComp = BitmapFactory.decodeByteArray(baComp, 0, baComp!!.size)
                Log.d("DEBUG", "cbtMComp.byteCount: ${cbtMComp.byteCount}")

                val dFNombre: DateFormat = SimpleDateFormat("yyyyMMddHHmmss")
                val randomNum = (0..9).random()

                val aporte = ParseObject("PATO_Aporte")
                aporte.put("idPatologia", ParseObject.createWithoutData("PATO_Patologia", pato!!.objectId))
                aporte.put("idEtiqueta", ParseObject.createWithoutData("PATO_Etiqueta", pato!!.idEtiqueta))
                aporte.put("aprobado", false)
                aporte.put("imagen", ParseFile("${dFNombre.format(Date())}$randomNum.jpg",baComp))
                aporte.saveInBackground(SaveCallback { e ->
                    // Handle success or failure here
                    if (e == null){
                        //Log.i("FutbolEc", "Current user actualizado")
                    }
                })
            }
        }
    }
}

```



```

    }
    else{
        Toast.makeText(this, "", Toast.LENGTH_LONG).show()
        val builder = androidx.appcompat.app.AlertDialog.Builder(this, R.style.AlertDialogTheme)
        builder.setTitle("Error al grabar la foto")
        builder.setMessage("Error: Se produjo una excepción al grabar el aporte")
            .setCancelable(true)
            .setPositiveButton("Aceptar") { dialog, id -> }
        val alert = builder.create()
        alert.show()
    }
})

}
}
}

//COMPRIMIR IMAGEN:
fun getCompressedBitmapData(bitmap: Bitmap, maxFileSize: Int, maxDimensions: Int): ByteArray? {
    val resizedBitmap: Bitmap
    resizedBitmap = if (bitmap.width > maxDimensions || bitmap.height > maxDimensions) {
        getResizedBitmap(
            bitmap,
            maxDimensions
        )
    } else {
        bitmap
    }
    var bitmapData = getByteArray(resizedBitmap)
    while (bitmapData.size > maxFileSize) {
        bitmapData = getByteArray(resizedBitmap)
    }
    return bitmapData
}

fun getResizedBitmap(image: Bitmap, maxSize: Int): Bitmap {
    var width = image.width
    var height = image.height
    val bitmapRatio = width.toFloat() / height.toFloat()
    if (bitmapRatio > 1) {
        width = maxSize
        height = (width / bitmapRatio).toInt()
    } else {
        height = maxSize
        width = (height * bitmapRatio).toInt()
    }
    return Bitmap.createScaledBitmap(image, width, height, true)
}

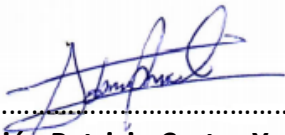
private fun getByteArray(bitmap: Bitmap): ByteArray {
    val bos = ByteArrayOutputStream()
    bitmap.compress(Bitmap.CompressFormat.JPEG, 80, bos)
    return bos.toByteArray()
}
}

```

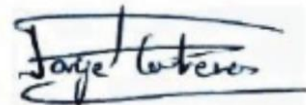
AUTORIZACION DE PUBLICACION EN EL REPOSITORIO INSTITUCIONAL

Nosotros, **Adrián Patricio Castro Yupangui y Jorge Eduardo Contreras Contreras** portadores de las cédulas de ciudadanía N.º **1400891535** y N.º **0106831530**. En calidad de autores y titulares de los derechos patrimoniales del trabajo de titulación “Proceso de consolidación urbana en las áreas de influencia inmediata del Cantón Chilla” de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconocemos a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Así mismo; autorizamos a la Universidad para que realice la publicación de este trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cuenca, 21 de diciembre de 2021



.....
Adrián Patricio Castro Yupangui
C.I. **1400891535**



.....
Jorge Eduardo Contreras Contreras
C.I. **0106831530**