



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TITULO DEL TRABAJO DE TITULACIÓN

Desarrollar una aplicación móvil de visión por computador para el reconocimiento de prendas de vestir de las personas con discapacidad visual.

AUTOR

Soriano Roca Kerly Elisa

PROYECTO DE UNIDAD DE INTEGRACIÓN CURRICULAR

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

Ing. Jaime Orozco Iguasnia, Mgt

La libertad, Ecuador

Año 2024



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino, Mgt.
DIRECTOR DE LA CARRERA

Ing. Jaime Orozco Iguasnia, Mgt.
TUTOR

Ing. Carlos Sánchez León, Mgt.
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez, Mgt.
DOCENTE GUÍA LIC



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **SORIANO ROCA KERLY ELISA**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 9 días del mes de diciembre del año 2024

TUTOR

A handwritten signature in blue ink, appearing to read "Jaime Orozco Iguasnia", is written over a horizontal line.

Ing. Jaime Orozco Iguasnia, Mgt



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
DECLARACIÓN DE RESPONSABILIDAD**

Yo, SORIANO ROCA KERLY ELISA

DECLARO QUE:

El trabajo de Titulación, Desarrollar una aplicación móvil de visión por computador para el reconocimiento de prendas de vestir de las personas con discapacidad visual. previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 9 días del mes de diciembre del año 2024

EL AUTOR

K.E.R

SORIANO ROCA KERLY ELISA



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA**

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Desarrollar una aplicación móvil de visión por computador para el reconocimiento de prendas de vestir de las personas con discapacidad visual, presentado por el estudiante, SORIANO ROCA KERLY ELISA fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 5%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



TUTOR

Ing. Jaime Orozco Iguasnia, Mgt



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, SORIANO ROCA KERLY ELISA

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, a los 9 días del mes de diciembre del año 2021

EL AUTOR

K.E.R

SORIANO ROCA KERLY ELISA

AGRADECIMIENTO

Quiero expresar mi más profundo agradecimiento a todas las personas que me han apoyado durante este importante recorrido. A mi madre, Roca Yagual Rosa Elvira, quien ha estado siempre presente, motivándome a seguir adelante, consolándome en los momentos más difíciles y celebrando conmigo cada logro alcanzado. Su amor incondicional y su fortaleza han sido mi mayor fuente de inspiración.

A mi abuelo, Roca Orrala Julio Víctor, cuyo amor y cuidado inigualables me acompañan siempre. Gracias por esperar mi regreso cada día que salía a la universidad, por preocuparte por mi brindarme ese sincero y profundo amor de padre que tanto valoro.

A mis hermanos, Tíos y amigos por ser las personas en quienes siempre puedo confiar. Gracias por su apoyo constantes, finalmente, mi especial agradecimiento a los docentes, por su paciencia, dedicación y enseñanza a lo largo de este proceso. Sus conocimientos y guían han sido fundamental para culminar este largo camino.

Kerly Elisa Soriano Roca

DEDICATORIA

Dedico este trabajo a mi madre, Roca Yagual Rosa Elvira, por ser mi pilar, inspiración y mi constante apoyo por no dejarme renunciar cuando le decía que yo no podía más. Sus palabras de aliento, su amor incondicional y su fe en mí han sido mi fortaleza en los momentos más difíciles.

A mi abuelo, Roca Orrala Julio Víctor, quien con su cariño inquebrantable y su consejo me ha demostrado su amor verdadero. Este logro es también suyo, por todo lo que me has enseñado y por ser siempre un padre para mí.

A mis amigos que he tenido la fortuna de conocer en este camino universitario, con quienes hemos compartido momentos de aprendizaje, apoyo mutuo y crecimiento. Juntos hemos superado desafíos sin dejar espacio para la envidia ni el egoísmo, construyendo lazos de compañerismo basados en la sinceridad y el respeto. Este logro también es suyo, porque siempre estuvieron allí, brindándome su apoyo.

Kerly Elisa Soriano Roca

ÍNDICE GENERAL

TITULO DEL TRABAJO DE TITULACIÓN	I
TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD.....	IV
DECLARO QUE:.....	IV
CERTIFICACIÓN DE ANTIPLAGIO.....	V
AUTORIZACIÓN.....	VI
AGRADECIMIENTO.....	VII
DEDICATORIA.....	VIII
ÍNDICE GENERAL.....	IX
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS.....	XIII
RESUMEN.....	XVII
ABSTRACT.....	XVIII
INTRODUCCIÓN	2
CAPÍTULO 1	3
1. FUNDAMENTACION	3
1.1 ANTECEDENTES.....	3
1.2 DESCRIPCIÓN DEL PROYECTO.....	6
1.3 OBJETIVOS DEL PROYECTO.....	8
1.4 JUSTIFICACIÓN.....	9
1.5 ALCANCE DEL PROYECTO	12

1.6	METODOLOGÍA DEL PROYECTO	14
1.6.1	METODOLOGIA DE LA INVESTIGACIÓN.....	14
1.6.2	BENEFICIARIOS DEL PROYECTO.....	15
1.6.3	VARIABLES	15
1.6.4	ANALISIS DE RECOLECCIÓN DE DATOS.....	15
1.7	METODOLOGÍA DE DESARROLLO.....	16
CAPITULO II		19
1	PROPUESTA	19
2.1	MARCO CONTEXTUAL	19
2.2	MARCO CONCEPTUAL.....	20
2.3	MARCO TEÓRICO.....	29
2.4	REQUERIMIENTOS.....	31
2.4.1	REQUERIMIENTO FUNCIONALES	31
2.4.2	REQUERIMIENTO NO FUNCIONALES	33
2.5	COMPONENTE DE LA PROPUESTA	34
2.5.1	ARQUITECTURA DEL SISTEMA.....	34
2.5.2	INSTUMENTOS DE DESARROLLO.....	35
2.5.3	CREACION DEL DATASET Y ETIQUETADOS PARA ENTRENAR LA RED NEURONAL CONVOLUCIONAL CNN.....	35
2.5.4	PROCEDIMIENTO DE ENTRENAMIENTO Y SUS RESULTADOS	36
2.5.5	ALGORITMO DE DETECCION DE COLORES	42
2.5.6	ALGORITMO DE DETECCIÓN DE PRENDAS E INTEGRACIÓN DEL ALGORITMO PARA DETECCIÓN DE COLORES.....	43

2.5.7 CREACION DEL API PARA LA CONEXIÓN A LA BASE Y SU FUNCIONES PARA LA PAGINA WEB Y APLICATIVO MÓVIL.	44
2.5.8 DIAGRAMA DE CASO DE USO.....	62
2.5.9 MODELO DE DATOS	64
2.6 DISEÑO DE INTERFAZ.....	64
2.7 PRUEBAS.....	69
2.8 RESULTADOS.....	77
CONCLUSIONES	82
RECOMENDACIONES	83
BIBLIOGRAFÍA.....	84

ÍNDICE DE TABLAS

Tabla 1. Población beneficiaria del proyecto	15
Tabla 2 Requerimiento Funcional Del sistema web.....	32
Tabla 3 Requerimiento Funcional del aplicativo móvil.....	33
Tabla 4 Requerimiento no funcional del sistema web y aplicativo móvil	34
Tabla 5 Prueba de Inicio de sesión del sistema web	71
Tabla 6 Prueba de registro de usuario	72
Tabla 7 Registro de prendas pruebas.....	73
Tabla 8 Registro de dispositivos	74
Tabla 9 Detecciones en tiempo real	76
Tabla 10 Entrenamiento del modelo	77
Tabla 11 resultado de entrenamiento	78
Tabla 12 Tablas de Resultado de prueba.....	79
Tabla 13. Encuesta	100

ÍNDICE DE FIGURAS

Fig. 1. Fases de la metodología incremental para el sistema web	17
Fig. 2. Fases de la metodología incremental para aplicativo móvil	18
Fig. 3. Mapa de ubicación: UNIDAD EDUCATIVA “MANUELA ESPEJO”. ...	19
Fig. 4. Arquitectura del sistema cliente-servidor	34
Fig. 5. Imágenes del dataset y su etiquetado.....	36
Fig. 6. Conversión al formato YOLO	36
Fig. 7. Código de entrenamiento YOLO.....	37
Fig. 8. F1-Confidence curva.....	37
Fig. 9. Imagen de la precisión	38
Fig. 10. Imagen de la precisión y sensibilidad	38
Fig. 11. Imagen de la confianza y sensibilidad	39
Fig. 12. Imagen de la métrica de confusión	39
Fig. 13. Imagen de la métrica de confusión	40
Fig. 14. Análisis de Datos	41
Fig. 15. Matriz de dispersión.....	41
Fig. 16 resultado por época	42
Fig. 17. Algoritmo para la detección de colores	42
Fig. 18 .Función para identificar el nombre del color	43
Fig. 19. Función para la combinación de colores.....	43
Fig. 20. Primera parte del algoritmo de detección de prendas.....	44
Fig. 21. Segunda parte del algoritmo de detección de prendas	44
Fig. 22. Conexión del API con cada método.....	44

Fig. 23. Conexión del API con la base de datos.....	45
Fig. 24. Código de listar usuarios.....	45
Fig. 25. Código de Inicio de sesión en Visual Studio Code.....	46
Fig. 26. Método POST para el registro de usuarios	46
Fig. 27. Código de Registro en Visual Studio Code	47
Fig. 28. Método POST para verificar código de validación.....	47
Fig. 29.Código de verificación en Visual Studio Code.....	48
Fig. 30.Función de encriptar contraseña.	48
Fig. 31. Método POST para solicitar código de validación	49
Fig. 32. Función para enviar código de validación al correo	49
Fig. 33. Código de solicitar y enviar código al correo en Visual Studio Code.....	50
Fig. 34. Método de verificación de código de validación	50
Fig. 35. Método de verificación de código en Visual Studio Code	50
Fig. 36. Método de POST para la actualización de contraseña.....	51
Fig. 37. Código de actualización de contraseña en Visual Studio Code.....	51
Fig. 38. Librerías	52
Fig. 39. Primera parte del código de registro	53
Fig. 40. Segunda parte del código de registro.....	53
Fig. 41. Tercera parte del código de registro.....	53
Fig. 42. Cuarta parte del código de registro	54
Fig. 43.Cuarta parte del código de registro	54
Fig. 44.. Código de registro de prendas en Visual Studio Code	55
Fig. 45. Primera parte del código de aumento de datos	55

Fig. 46. Segunda parte del código de aumento de datos	56
Fig. 47. Tercera parte del código de aumento de datos.....	56
Fig. 48. Cuarta parte del código de aumento de datos	57
Fig. 49. Código de Actualización del dataset en Visual Studio Code.....	57
Fig. 50. Método de conteo y conversión de datos.....	58
Fig. 51. Método GET para lista prendas	58
Fig. 52. Código de listar prendas en Visual Studio Code	59
Fig. 53. Método GET para lista prendas pendientes	59
Fig. 54. Código de listar prendas pendientes en Visual Studio Code	60
Fig. 55. Método POST para registra el dispositivo	60
Fig. 56. Método para registra el dispositivo en Flutter	61
Fig. 57. Método para realizar detecciones usando la interfaz flutter	61
Fig. 58. Dependencias del archivo pubspec.Yaml	62
Fig. 59. Dependencias Del Archivo Androidmanifest.Xml.....	62
Fig. 60. Diagrama de caso de uso para el sistema web	63
Fig. 61. Diagrama de caso de uso del aplicativo móvil.....	63
Fig. 62. Base de Datos del sistema web y aplicativo móvil	64
Fig. 63. Interfaz de inicio de sesión del sistema web	64
Fig. 64. Interfaz de registro de usuario del sistema web.	65
Fig. 65. Interfaz de restablecer contraseña del sistema web.	66
Fig. 66. Interfaz de registro de prendas del sistema web	67
Fig. 67. Interfaz para listar prendas registradas	67
Fig. 68. Interfaz de inicio del dispositivo móvil.	68

Fig. 69. Detección de prendas y sus colores	68
Fig. 70. Detección de prendas, color y combinación	69
Fig. 71 Interfaz Login.....	92
Fig. 72. Interfaz principal	93
Fig. 73. Interfaz de listar prendas	94
Fig. 74. Interfaz principal del dispositivo móvil	94
Fig. 75. Interfaz principal del Detector de prendas	95
Fig. 76. Vista de Descarga	96
Fig. 77. Paso 2 para instalar el aplicativo móvil	97
Fig. 78. Paso 3 para instalar el aplicativo móvil	97
Fig. 79. Paso 4 para instalar el aplicativo móvil	98
Fig. 80. Paso 5 para instalar el aplicativo móvil	98
Fig. 81. Paso 6 para instalar el aplicativo móvil	98
Fig. 82. Paso 7 para instalar el aplicativo móvil	99

RESUMEN

El proyecto tiene como objetivo desarrollar una aplicación móvil para identificar patrones de prendas con sus respectivos colores, así como un sistema web que se permita subir imágenes de ropa, esta información alimentará el dataset del sistema, orientado a personas con problemas de reconocimientos de objetos y distinción de colores. El software cuenta con algoritmo de visión computacional que procesarán las imágenes capturadas en tiempo real desde el dispositivo móvil, el proceso de entrenamiento de la red neuronal convolucional YOLOv8, se inició recolectando y llevando a cabo un etiquetado manual de imágenes de vestimentas, esta información fue utilizada para crear el dataset personalizado, con los datos listos, se continuó con el entrenamiento de la red neuronal, finalizando esta parte se integró el modelo entrenado con el algoritmo de visión por computador junto con el algoritmo de identificador de colores KMeans, la integración de este algoritmo facilita el reconocimiento de prendas y colores en tiempo real. Otras herramientas utilizadas para la creación de estos softwares son PyCharm Community Edition que es entorno de código Python, que facilito la creación del API más la implementación de los procesos de entrenamiento y funcionalidades del sistema. Además, se usó flutter como framework de diseño de la interfaz del aplicativo móvil y ionic, Angular el sitio web conectado con flask el cual es un enrutamiento para interconectar el API con las interfases. El método de desarrollo que se eligió es incremental ya que tiene 4 fase: requisito, diseño, desarrollo y pruebas, el objetivo de aplicativo móvil es que la persona puede usarlo fácilmente por lo tanto su interfaz es amigable e intuitiva tiene funcionalidades fáciles de majear debido al problema visual que tiene los usuarios. Finalizando el desarrollo de sistemas, se procedió a realizar pruebas para evaluar la precisión de reconocimiento del aplicativo móvil cada detección realizada tiene un nivel de confianza de 90% con esta información se puede validar que el sistema está detectando de manera eficaz.

Palabras claves: Visión por computador, persona con discapacidad visual, reconocimiento de prendas.

ABSTRACT

The project aims to develop a mobile application to identify patterns of clothing with their respective colors, as well as a web system that allows uploading images of clothes, this information will feed the dataset of the system, oriented to people with problems of object recognition and colors distinction. The software has computer vision algorithms that will process the images captured in real time from the mobile device, the training process of the YOLOv8 convolutional neural network began by collecting and carrying out a manual labelling of images of clothing, this information was used to create the personalized dataset, With the data ready, we continued with the training of the neural network, finishing this part we integrated the trained model with the computer vision algorithm together with the KMeans colors identification algorithm, the integration of this algorithm facilitates the recognition of garments and colors in real time. Other tools used for the creation of these software's are PyCharm Community Edition, which is a Python code environment, which facilitated the creation of the API plus the implementation of the training processes and functionalities of the system. In addition, flutter was used as a framework to design the interface of the mobile application and ionic, Angular the website connected with flask which is a routing to interconnect the API with the interfaces. The development method chosen is incremental as it has 4 phases: requirement, design, development and testing, the objective of the mobile application is that the person can use it easily, therefore its interface is friendly and intuitive, it has easy to use functionalities due to the visual problem that the users have. At the end of the system development, tests were carried out to evaluate the recognition accuracy of the mobile application, each detection carried out has a confidence level of 90%, with this information it can be validated that the system is detecting effectively.

Keywords: Computer vision, visually impaired person, clothing recognition.

INTRODUCCIÓN

Los problemas de visuales dependiendo el tipo de discapacidad visual, afecta a las personas en realizar actividades diarias de manera autónoma. Uno de los tipos de problemas visual que se presentan es el monocromatismo, la personas con este trastorno tiene problemas para distinguir color y hasta objetos dado a que perciben el mundo en una escala de grises blanco y negro, esto es un obstáculo para actividades como la de vestirse, leer, escribir incluso caminar.

El desarrollo de este sistema de reconocimiento de prendas desde un aplicativo móvil en tiempo real más el diseño de un sitio web que facilite el subir imágenes que actualicen el dataset dirigido a las personas que tenga dificultades para identificar prendas y reconocer colores. Para llevar a cabo este desarrollo se divide en dos capítulos primero se realiza un estudio para profundizar el tema, detallar las funcionalidades con la que contará el aplicativo y sitio web y mediante un banco de pregunta dirigido a lo docente identificar a los beneficiaros. Como segunda sección se aborda el uso de técnica de visión computacional, además se explica la forma de recopilación y etiquetado de imágenes para el dataset, también se permite conocer la red neuronal a utilizar para el entrenamiento de detección y reconocimiento de vestimenta, así como implementar algoritmos para identificar colores. El propósito de crea este aplicativo es que las persona con problemas de reconocimiento de colores y objeto, pueda utilizarla para identifica sus prendas de vestir con sus respectivas tonalidades así puedan lucir de una mejor manera.

El desarrollar este software es ofrecer una solución practica y accesible a las personas con monocromatismo, es decir que puedan vestirse de una forma autónoma si la necesidad de alguien que le ayude por lo tanto el aplicativo tiene una interfaz amigable e intuitiva, cuenta con una retroalimentación auditiva y textual sobre reconocimiento de lo que capta la cámara, esto permite a los usuarios realizar una selección de ropa según su gusto permitiendo lucir bien, este aplicativo móvil no solo busca innova en el ámbito de las tecnologías de asistencia sino ofrecer a lo usuario una herramienta que le ayuden en sus proceso de vestirse.

CAPÍTULO 1

1. FUNDAMENTACION

1.1 ANTECEDENTES

En la provincia de Santa Elena, el consejo Nacional del Ecuador informa que un 8.47% de la población tiene discapacidad visual [1]. Los problemas de visión tienen diversas causas como errores refractivos (miopía, hipermetropía, astigmatismo), cataratas, retinopatía diabética (daños en los vasos sanguíneos) y glaucoma (afecta el nervio óptico) [2]. Estas condiciones que limitan severamente la independencia y participación de las personas en actividades cotidianas como la lectura, la escritura, la navegación espacial e incluso en realizar tareas sencillas como vestirse, limitando su calidad de vida y su interacción en la sociedad [3].

Una condición visual que algunas personas padecen es el monocromatismo o daltonismo, son trastornos en los que las personas son incapaces de identificar colores, lo que hace que perciban el mundo en diferentes tonos de gris. Esta incapacidad para distinguir colores genera una serie de dificultades en la vida diaria, y una de ellas es la dificultad para vestirse de manera independiente, ya que no puede diferenciar el color de las prendas ni combinarlas. Este problema visual junto con otro tipo de discapacidad necesita soluciones tecnológicas que permitan no solo mejorar la accesibilidad, sino también ayuden en procesos básicos como el de vestirse [4].

En las últimas décadas los avances tecnológicos de visión por computador más el aprendizaje autónomo ha abierto oportunidades en crear sistemas para reconocer objetos e imágenes con precisión. Estos avances no solo se limitan al ámbito académico o industrial, sino que también han sido integrados en diferentes campos como por ejemplo la conducción autónoma, la salud y el entretenimiento, donde su uso es para automatizar procesos, optimizar tareas empresariales o desarrollar herramientas de aprendizaje. Uno de los sectores que tiene beneficio con este avance son la producción de herramientas para ayudar a una persona con algún problema de discapacidad, donde la tecnología crea soluciones para personas con discapacidades sensoriales [5].

La tecnología en la educación inclusiva sigue teniendo limitaciones debido a los pocos recursos que puede existir, sin embargo, esto no quiere decir que no existan herramientas que aporte en la educación solo que implementarlas requiere recurso que las instituciones no pueda cumplir. La unidad educativa Manuela Espejo, es especializada en la educación de persona con discapacidad, los estudiantes que forma parte de esta institución poseen diversas discapacidades. Sin embargo, esta unidad educativa no cuenta con tecnología de apoyo especialmente para estudiantes con discapacidad visual. La Tecnología de visión por computador podría ser utilizada para crear herramientas que permitan ser utilizadas para realizar tareas sencillas, permitiendo que los estudiantes con problema visual o con monocromatismo adquiera mayor autonomía en sus actividades cotidianas, como el de reconocer colores de prendas de vestir [6].

Realizar investigaciones sobre la tecnología de visión por computador y el aprendizaje automático, es fundamental para el desarrollo de aplicaciones o sistemas que permitan identificar prendas de vestir. Un caso ilustrativo es el proyecto que se desarrolló en el año 2020 en la universidad de Chile titulado “Detección de prendas de vestir utilizando modelos de detección de objetos basado en Deep Learning” en el que se analizaron cinco modelos de detección de objetos que son Yolov3, RetinaNet, Faster R-CNN, Mask R-CNN y TridentNet. Este estudio utilizó los conjuntos de datos DeepFashion2 y ModaNet, con la finalidad de entrenar los modelos seleccionados para identificar cuál es mejor para la detección con precisión y qué dataset se recomienda para alimentar el dataset [7].

DeepFashion2 y ModaNet son conjuntos de datos para tareas de visión por computadora el cual permite clasificar, segmentar y detectar patrones de prendas, para realizar este proyecto se creó una máquina virtual en la nube para mejorar el rendimiento. Antes del entrenamiento ajustaron los datasets para cada modelo una vez obtenidos los datos se procedió a entrenar después de finalizar se evaluó los cinco los resultados obtenidos como respuesta que TridentNet tiene un mejor desempeño de reconocimiento usando el dataset ModaNet ya que tiene clases de vestimentas más generales que DeepFashion2 [7].

Por otro lado, en la universidad politécnica de Valencia con el tema “Desarrollo de un sistema para identificar prendas en imágenes a través redes Yolov8 personalizada” en el año 2023 este proyecto se explica como entrenar yolov8 con un datasets personalizado el cual cuenta con 10 clases cada una de ellas tiene 537 imágenes, el proceso de entrenamiento se realiza en Google Colab, el dataset debe estar en formato Yolo con sus respectivas etiquetas y anotaciones, también explica la conexión con flask ya que diseñó una interfaz intuitiva para interactuar con el usuario, este sistema sirve para que el usuario suba imágenes el cual será evaluado por el modelo para hacer las respectivas detecciones [8].

Además, en la universidad politécnica de Catalunya BarcelonaTech se realizó el siguiente proyecto “Desarrollo de una inteligencia artificial para la detección en tiempo real de prendas de ropa en personas basada en tipo y color”

Además, en la universidad politécnica de Catalunya BarcelonaTech se realizó un proyecto el cual consistió en desarrollar una inteligencia artificial para la identificación en tiempo real de prendas de vestir. Se utilizaron técnicas de cuantización para determinar el color, al ser un proyecto el cual se implementó para una empresa se realizó un análisis de presupuesto, se seleccionaron varios datasets para que el modelo entrenado pueda tener una mejor precisión en lo que es detección, también se llevó a cabo una investigación de los modelos Yolo con el propósito de seleccionar uno adecuado para hacer predicciones en tiempo real, otra herramienta que se utilizó para el desarrollo es LANACCESS esta sirvió para crear y desplegar la aplicación [9].

Estos proyectos demuestran como las tecnologías de visión por computador y el aprendizaje automático puede servir para desarrollar herramientas para que realicen tareas que sean útiles para el ser humano. Los avances en estos campos podrían ser directamente aplicados para el desarrollo de un aplicativo que pueda utilizar un estudiante en la unidad educativa Manuela Espejo, especialmente aquellos que poseen un grado de visión avanzado o que tienen monocromatismo.

1.2 DESCRIPCIÓN DEL PROYECTO

El proyecto tiene como finalidad crear un aplicativo móvil dirigido para a la unidad Educativa Manuela Espejo, con el fin de asistir en el proceso de vestimenta de las personas con problemas visuales o monocromatismo. Esta aplicación debe capturar imágenes en tiempo real para reconocer las prendas más sus colores, permitiendo a los usuarios mejorar su independencia al momento de vestirse.

El desarrollo del aplicativo se basa en utilizar redes neuronales convolucionales (CNN) para el realiza un entrenamiento que permitiendo reconocer patrones de prendas, además se debe crear un algoritmo de visión por computador que sirva para procesar las imágenes capturadas con el propósito de identifica no solo las prendas si no sus respectivos colores.

El sistema contará con una página web donde los usuarios podrá subir imágenes individuales de prendas las cuales servirá para alimentar el dataset de modelo entrenado. Esta plataforma permitirá que le administrado use los datos almacenados en el dataset para que el modelo entrenado pueda usar esta nueva información para un reentrenamiento, lo que garantiza una mejor continua en la precisión del reconocimiento.

Este software debe tener un diseño amigable y fácil de usar tanto para el sistema web como para el aplicativo móvil, se utilizará colores grises ya que esto brindará una claridad en el diseño para que las persona con problema visuales y monocromatismo. No tenga problema en su uso. Al integrar el reconocimiento de prendas y colores, la aplicación brindar a los usuarios una descripción de prendas, tanto en texto como en audio, mejorando así la autonomía de las personas con discapacidad visual, El sistema web y aplicativo móvil de reconocimiento de prendas con sus colores cuenta con los siguientes módulos de desarrollo los cuales se describirá a continuación.

PÁGINA WEB

Módulo de Inicio de Sesión:

- Ingreso de credenciales (correo electrónico y contraseña)
- Opción para recupera las contraseñas.

- Enlace para ir a la sección de registro de nuevo usuarios

Módulo de Registro de Usuarios:

- Ingreso de datos del usuario.
- Validar datos
- Código de verificación
- Validar campos
- Registro de los nuevos usuarios
- Almacenar los datos ingresados

Módulo de Recuperación de Contraseña:

- Ingreso de correo electrónico
- Código de verificación
- Ingreso de contraseña nueva y confirmación.
- Actualiza contraseña

Módulo Principal de la página:

- Sección de Registro de prendas
- Menú para ir a listar prendas
- Cerrar Sección

Módulo de Registro de prendas:

- Ingresar una foto de la prenda
- Ingresar el nombre la prenda
- Se extraen coordenada
- Almacenamos datos de las prendas
- Actualización del dataset

Módulo de listar:

- Listar prendas y sus coordenadas
- Listar prendas pendientes

APLICATIVO MÓVIL

Módulo de Inicio de sesión

- Icono de inicio para extraer información del dispositivo
- Almacena los datos
- Contador para el seguimiento del uso del aplicativo

Módulo de detección de prendas

- Interfaz de cámara para capturar imagen
- Activar o desactiva el proceso de detección
- Opción para combinación de prendas
- Opción para utiliza cámara Frontal/Trasera
- Mostrar Resultado
- Audio para leer los resultados

El proyecto contribuye con la línea de investigación de Desarrollo de software vinculada con la sublínea de desarrollo de algoritmos y visión artificial, utilizando librerías de visión por computador más el entrenamiento de una red neuronal convolucional CNN para el reconocimiento de prendas integrando por algoritmos que permita identificar colores esto garantiza que la descripción del resultado de detección en tiempo real sea correcta.

1.3 OBJETIVOS DEL PROYECTO

OBJETIVO GENERAL:

Desarrollar una aplicación móvil para reconocimiento de prendas de vestir con sus respectivos colores, que facilite el proceso de selección y combinación de ropa para personas con Monocromatismo.

OBJETIVO ESPECIFICOS:

- Investigar algoritmos de visión por computadora para la detección y reconocimiento de prendas de vestir con sus colores.
- Desarrollar una página web que permita actualizar el dataset del modelo entrenado mediante la carga de imágenes.

- Integrar algoritmos de aprendizaje automático, como redes neuronales convolucionales (CNN) y métodos de detección de características, para reconocer patrones de prendas de vestir en la aplicación móvil.
- Realizar pruebas para evaluar la precisión y la confiabilidad del sistema en diferentes condiciones de iluminación, variaciones en la disposición de las prendas de vestir.

1.4 JUSTIFICACIÓN

El monocromatismo conocido también como acromatopsia es una discapacidad visual que no distingue colores ya que solo se puede percibir los colores en tonos grises, blanco y negro. Las personas que padecen de esta enfermedad enfrentan desafíos en su vida diaria, que incluye dificultades para ver en lugares con poca iluminación, alta sensibilidad a la luz solar y visión reducida en general. Esta causa influye en su proceso de realizar actividades cotidianas, como leer, pintar y vestirse ya que tiene poca capacidad de reconocer prendas con sus colores. Esta condición no solo afecta la autonomía de estos individuos, sino también sus participaciones en diversas actividades cotidianas y su integración social [10].

Los problemas de discapacidad visual los cuales incluye trastorno relacionado con la percepción de colores como la discromatopsia, el daltonismo, hemiacromatopsia entre otras, son afecciones que no permiten reconocer ciertos colores a diferencia del monocromatismo el cual no se distingue ningún color, estas pueden ser causadas por factores genéticos, hereditario de tipo autosómico o se pueden desarrollarse por consecuencia de la diabetes o glaucomas para estos problemas no existe tratamiento adecuado además que pueden avanzar provocando una ceguera total en la persona dejando a los afectados en una situación de vulnerabilidad [11].

La tecnología ha evolucionado sirviendo de aportes para desarrollar herramientas que sirvan de utilidad para el ser humano, existen aplicaciones como Color Blind Pal que están dirigidas a personas con daltonismo el cual permite distinguir solo colores está disponible tanto para Android y IOS otro ejemplo es Color ID Free es una herramienta móvil similar a la anterior la diferencia que solo está disponible para IOS, existen otros ejemplos de aplicaciones similares sin embargo no ayudan mucho

cuando una persona con problema de monocromatismo ya que además de no poder distinguir colores alguna tiene complicaciones para reconocer objeto. La falta de estas herramientas que ayuden a este tipo de usuario es ausente y esto provoca frustración porque depende de otros factores para poder llevar una vida normal [12].

Desarrollar un aplicativo móvil que utilice tecnologías de visión por computador y redes neuronales convolucionales (CNN) para reconocer prendas de vestir con sus respectivos colores, que sirva de ayuda en el proceso de vestimenta de las personas con discapacidad visual o monocromatismo es un avance significativo en la tecnología, ya que los usuario no necesitara de terceras persona que le indique las prenda con su colores si no que lo puede hacer de forma autónoma usando el aplicativo esto permitirá que puede seleccionar las ropa de acuerdo a su gusto. Este software contiene algoritmos que procesa imagen dando a conocer en una descripción el tipo de prenda con sus colores el cual es proyectada a través de una retroalimentación auditiva.

La finalidad de este proyecto es el desarrolla de una aplicación móvil que se utilizara en dispositivos Android dirigido a estudiante con discapacidad visual o monocromatismo de la unidad educativa “MANUELA ESPEJO”. La aplicación tiene el objetivo de reconocer prendas de vestir con sus respetivos colores proporcionado información a través de descripción textual y auditiva sobre el resultado además que contara con una interfaz amigable y fácil de utilizar, adaptándose a la persona con problema visual para que su proceso de uso se sencillo.

Este proyecto también tiene como aporte al desarrollo del componente cuatro de proyecto de vinculación, el cual busca fomentar la inclusión social mediante el uso seguro y responsable de la tecnología de información y comunicación. La aplicación móvil está dirigida especialmente a persona con discapacidad visual o monocromatismo, donde usa la tiflotecnología para asistirles en tareas cotidianas como el proceso de vestirse si depende de terceras personas. De esta forma mejora la autonomía e inclusión permitiéndoles que pueda vestirse de manera independiente.

Además, el proyecto se alinea con los objetivos del plan de creación de oportunidades 2021-2015, donde se promueve el acceso de tecnologías digitales inclusiva y fomenta la participación de personas con discapacidad visual, se encuentra fundamentado en los siguientes ejes:

Eje financiero

Objetivo 4: Asegurar una administración sostenible y transparente de las finanzas públicas [13].

Eje social

Política 5.5: incrementa la conexión digital y el acceso de la población a tecnología emergentes [13].

Indicadores territoriales

Pol 5.1

C3. Esto permite fomentar programa de protección social que no son contributiva en la provincia que tiene mayor pobreza y desnutrición [13].

Pol 5.4

A4. Fortalecer la conexión y la utilización de las TIC como medio para optimizar el acceso a otros servicios [13].

Con base en todo esto argumento se pretende realizar un aplicativo móvil Android dirigido para estudiante con problemas visual o monocromatismo de la unidad educativa Manuela Espejo que sea capaz de reconocer el patrones de prendas de vestir con su respetivo colores brindando mediante descripción textual y auditiva los resultado de la captura realizada, además de contar con una sistema web donde el usuario puede subir prenda de su closet para actualizar el dataset, donde el administrado se encargara de revisarlo proceder a reentrenar el modelo para que su aprendizaje sea más preciso en reconocimiento.

1.5 ALCANCE DEL PROYECTO

La finalidad del "Desarrollo de una aplicación móvil de visión por computadora para el reconocimiento de prendas de vestir de personas con discapacidad visual" es tener un aplicativo que sirva para reconocer prendas de vestir con sus respectivos colores para los estudiantes de la unidad educativa Manuela Espejo especialmente aquello que tenga dificultad para reconocer patrones de ropa con sus respectivos colores y puedan utilizarlo para proceso de vestimenta sin la necesidad de que alguien más le ayude en su selección.

Al contar con mucho diseño de vestimenta para crear el dataset del aplicativo se recopilará imágenes de prendas básica y sencilla el cual tendrá sus etiquetas como prueba, las clases usadas para el entrenamiento serán cuatro el cual tendrá alrededor de 600 imágenes cada una de ellas.

El desarrollo del proyecto cuenta con ocho módulos que se deben implementar esto está dividido en seis módulos para el sistema web y dos para el aplicativo móvil. El primer módulo del sistema web es el ingreso de sección de usuarios si ya está registrado puede usar sus credenciales para acceder a la plataforma, además que ofrecen opciones de recuperación de contraseñas en caso de olvido también cuenta con un enlace a la sección de registro para aquellos usuarios que use este sistema por primera vez. El diseño está creado con colores en tonos grises con el propósito de brindar una vista interfaz amigable para las personas con trastorno relacionado con la percepción de colores así puede visualizar los componentes que tiene en esta sección.

En el módulo de registro se ingresan los datos de los nuevos usuarios para poder usar el sistema web, estos datos proporcionados serán validados con el propósito de asegurar que la información ingresada sea correcta, la verificación se realiza mediante un código que se envía al correo ingresado por el usuario esto confirmará la autenticidad del registro. El sistema verifica si el código ingresado es correcto crear la cuenta almacenando los datos en la MySQL caso contrario no permite registro ni creación de cuenta por lo tanto no podrá iniciar la sección en la interfaz de inicio.

Para aquellos usuarios que haya olvidado sus contraseñas cuenta con el módulo de recuperación de contraseña el cual permitirá registrar una nueva contraseña. En esta parte se debe ingresar el correo de la cuenta del usuario para verificar si existe. Una vez confirmado este proceso se envía el código de verificación a dicho correo para poder continuar con el restablecimiento de contraseña. Finalmente se solicita al usuario ingresar nueva contraseña, confirmar la contraseña actual para actualizar de esta manera poder seguir usando el sistema.

El sistema cuenta con el módulo principal el cual su diseño es en tonos grises. Esto es por la condición de los usuarios a quienes van dirigidos en esta sección. Tenemos un menú lateral el cual no dirige a la lista de prendas registradas y cierra la sección. En la página principal también tenemos la parte de registrar prendas, cuenta con tres botones que nos permiten tomar foto, registrar y actualizar el dataset del modelo.

El módulo de registrar prendas, el usuario puede cargar imágenes de su guardarropa de manera individual, automáticamente se extraen las coordenadas de segmentación de los patrones de las prendas, se ingresan los nombres de las clases para después registrar. Una vez terminado el proceso, el usuario dará clic en el icono de actualizar dataset, el cual automáticamente genera un aumento de información con las imágenes proporcionadas para después enviar todo al dataset y se actualice el modelo así como continuar su autoaprendizaje.

En la sección de listar se puede visualizar las imágenes ingresadas de cada usuario, más las coordenadas extraídas por codificación interna de manera automática. También cuenta con otra sección la cual se lista aquellas imágenes que no fueron reconocidas por el modelo. Esto es para permitir que solo vayan los datos correctos al dataset. Por lo tanto se usa el mismo modelo para hacer segmentaciones de las imágenes que se ingresan, al realizar este proceso quiere decir que la foto en el apartado pendiente no se toma en cuenta, solo que el administrador la revise y verifique esta información.

El propósito del sistema web es que el usuario pueda ingresar sus propias fotos de su guardarropa, esto es como un personalizador del aplicativo. Todos los datos son almacenados en el dataset internamente, abre el código que verifique si el dataset fue actualizado. Si está en lo correcto ejecuta los comandos para usar el dataset y reentrenar el modelo del aplicativo con los datos nuevos.

Los módulos del aplicativo móvil son el de iniciar sesión aquí se extraer información del celular además que tiene un contador interno esto permite saber cuántas veces el usuario usa el aplicativo no se pide ingresar datos por el problema de visión de los estudiantes, al terminar el proceso se da pase a la siguiente sección.

La siguiente sección es la parte del módulo de detección de prendas donde ofrecen una interfaz intuitiva de la cámara permitiendo a los usuarios capturar en tiempo real imágenes para detectar, el usuario cuenta con dos opciones el de detener e iniciar la detección normal es decir solo reconocer patrones de prenda con su colores la otra opción es combinación aquí no solo tendrá la información de reconocer como la anterior sino que además podrás saber cómo combinar la prenda, el aplicativo permite usar la cámara frontal o trasera del dispositivo, la detecciones se realiza en tiempo real dando los resultado de manera textual y auditiva, proporcionando una experiencia accesible y eficiente para el uso del aplicación.

1.6 METODOLOGÍA DEL PROYECTO

1.6.1 METODOLOGIA DE LA INVESTIGACIÓN

La metodología de investigación usada en este proyecto es de tipo exploratoria ya que hay varios sistemas de reconocimiento de prendas por lo que será revisadas con el propósito de recolectar información para tener un enfoque claro sobre los procesos a seguir para el desarrollo del aplicativo móvil [14].

La investigación es exploratoria ya que le proyecto implica la creación y aplicación de una tecnología específica, debido a que se debe crea un aplicativo el cual necesita utilizar algoritmo de visión por computador y rede neuronales que permita realizar el proceso de reconocimiento de patrones de prendas de vestir, por tal motivo se debe hacer una investigación para seleccionar algoritmo y herramienta para preparar el entorno de desarrollo adecuadamente con el fin de crear el diseño del software de forma correcta.

Otra metodología de investigación que se utiliza es la experimental ya que este método implica realizas prueba, ajuste y evaluar cómo funciona el aplicativo móvil desde el celular. Esta prueba no sirve para mejorar el diseño o la red neural entrenada con el objetivo de mejora su precisión en el reconocimiento más examinar

como se ejecuta el dispositivo en varios ambientes. Con esta metodología se busca no solo darle solución al problema que padece las personas con monocromatismo, sino también garantiza que sea accesible, intuitiva y practica para que se utilice sin ningún problema [15].

1.6.2 BENEFICIARIOS DEL PROYECTO.

Se realizo una recolección de datos mediante la encuesta realizada a los docentes de la U.E Manuela Espejo para identificar el total de estudiantes que tiene discapacidad visual específicamente problema para identifica colores u objetos, mediante el análisis de los resultados se determinó los beneficiarios del proyecto.

BENEFICIARIOS	TIPOS	CANTIDAD
UNIDAD EDUCATIVA MANUERLA ESPEJO	ESTUDIANTES	5

Tabla 1. Población beneficiaria del proyecto

Este grupo de personas representa la población con las que se explorara los resultados que la aplicación demuestre, la selección de este grupo se especificó en el problema de visión que tiene.

1.6.3 VARIABLES

Variable Independiente: Uso de la aplicación móvil de reconocimiento de prendas de vestir.

Variable Dependiente: Precisión en el reconocimiento de prendas de vestir

1.6.4 ANALISIS DE RECOLECCIÓN DE DATOS

La encuesta es una técnica que permite recolectar información para después realizar un análisis detallado. Por ser una forma sencilla y fácil de utilizar se la implemento en el proyecto, mediante una serie de pregunta a los docentes de la U.E Manuela Espejo datos que se usaron identifica a los beneficiarios y opiniones sobre la aplicación a desarrollar.

La encuesta realizada a los docentes de la unidad educativa “Manuela Espejo” fue para conocer el número de estudiante que tiene discapacidad visual y de este grupo saber quiénes tiene problema para reconocer objeto e identificar colores. Esta información es importante ya que se determina el número de beneficiario a los que va dirigido el proyecto.

Otro punto se logró obtener una vez finalizada la encuesta es si los docentes creen que un aplicativo que ayude en el proceso de vestimenta para aquello que no puede reconocer prendas ni colores sería beneficioso, también las ventajas que puede tener implementan este tipo de aplicativos, un parte fundamental de la encuesta son la características y funcionalidades con la que creen que deba contar el aplicativo para que los estudiantes puedan utilizarlo si problemas.

1.7 METODOLOGÍA DE DESARROLLO

El proyecto tiene como objetivo crea un aplicativo móvil que permita reconocer prendas de vestir con sus respectivos colores para tener una desarrollo y planificación correcta se utiliza la metodología de desarrollo incremental debido a su parecido con la modelo casada su diferencia es que este método se le puede agregar una funcionalidad nueva en cada eta si es necesario [16]. Las fases de este modelo permitirán tener un desarrollo ordenado por lo tanto se dividen en cuatro etapas que son las siguiente:

Fase de requisitos: En esta etapa, se lleva a cabo una investigación profunda sobre las necesidades específicas de los usuarios con monocromatismo o discapacidad visual. Esto incluye comprender cómo perciben el mundo y los colores, lo cual es necesario para diseñar una interfaz accesible y funcional. Se analizan los desafíos visuales a los que se enfrentan estos usuarios y se definen las funcionalidades del aplicativo, asegurando que la interfaz sea intuitiva y fácil de usar.

Fase de diseño: En esta fase se diseña la interfaz del aplicativo más las funcionalidades de como esta va a esta conformada. Usamos un prototipo para visualiza su arquitectura esto permitirá realizar ajustes y mejora la usabilidad. Además de especificar lo requisitos técnicos como el hardware y software que se necesite para un correcto funcionamiento.

Fase de desarrollo: En esta fase se lleva a cabo la creación del aplicativo donde se prepara todo el entorno para desarrollarlo, lo primero que debe hacer es crear la data con foto de prendas esto sirve para entrenar la red neural convolucional, crear el algoritmo de visión por computador para procesar las imágenes una vez realizada la codificación del aplicativo se establece proceso de control de calidad para asegura que cada función cumpla con lo que se ha planificado.

Fase de prueba: Esta es la fase final del desarrollo aquí se realizará prueba de cómo funciona cada componente del aplicativo con el objetivo de comprobar que este correctamente elaborado. Esta aplicación móvil debe ser capaz de reconocer prendas de vestir con sus respectivos colores como posee tres componentes cada uno debe estar bien configurado por lo tanto estas pruebas exhaustivas que se realiza sirven para hacer ajustes hasta que todo el aplicativo cumpla con los requisitos planteando desde el inicio.

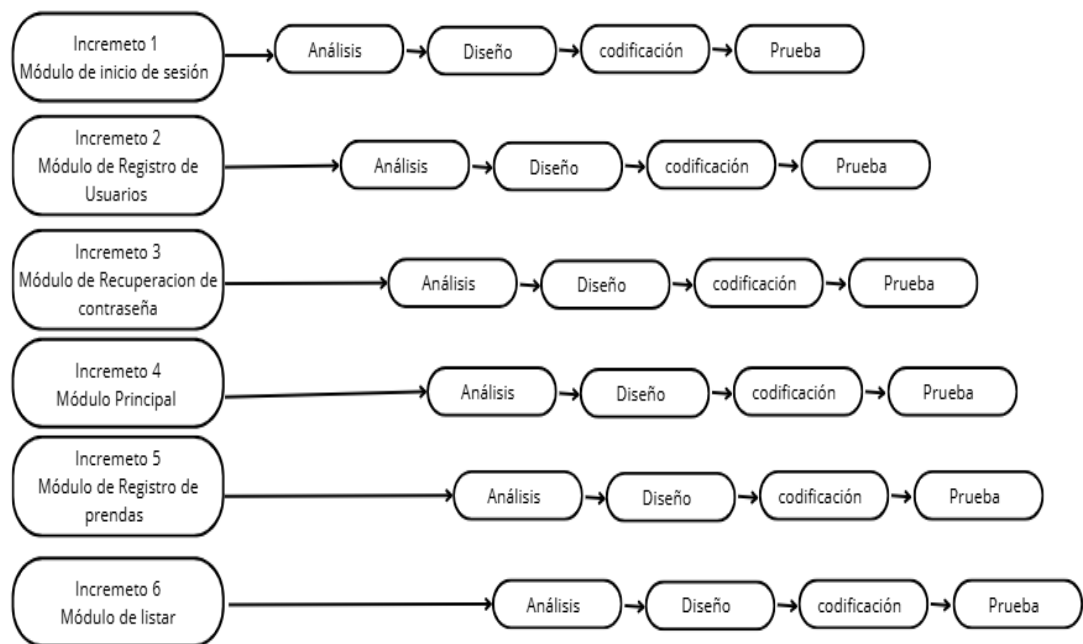


Fig. 1. Fases de la metodología incremental para el sistema web

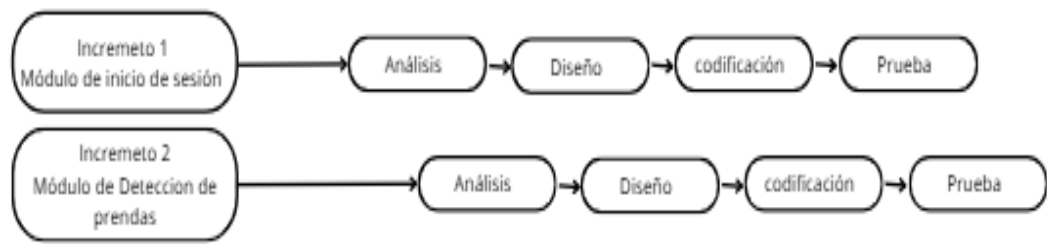


Fig. 2. Fases de la metodología incremental para aplicativo móvil

CAPITULO II

1 PROPUESTA

2.1 MARCO CONTEXTUAL

La Unidad Educativa “Manuela Espejo” este situado en la provincia de Santa Elena Cantón La Libertad, en el barrio 28 de mayo, calle 13 y 14 Av. 16. Es un centro educativo urbano del Ecuador, ubicado en la Zona 5 geográficamente, cuenta con modalidad presencial en las jornada matutina y vespertina [17].

La institución educativa se destaca por tener un compromiso con la inclusión ya que trabajan con estudiantes que posee algún tipo de discapacidad enriqueciendo así las capacidades cognitivas, sociales y emocionales de lo cada uno de ellos, ofrece algunos servicios especializados como terapias físicas, psicológica, de lenguaje entre otros. Además, que tiene una enseñanza multimodal donde se adaptan técnicas que favorezca la integración y participación en proceso educativo [17].

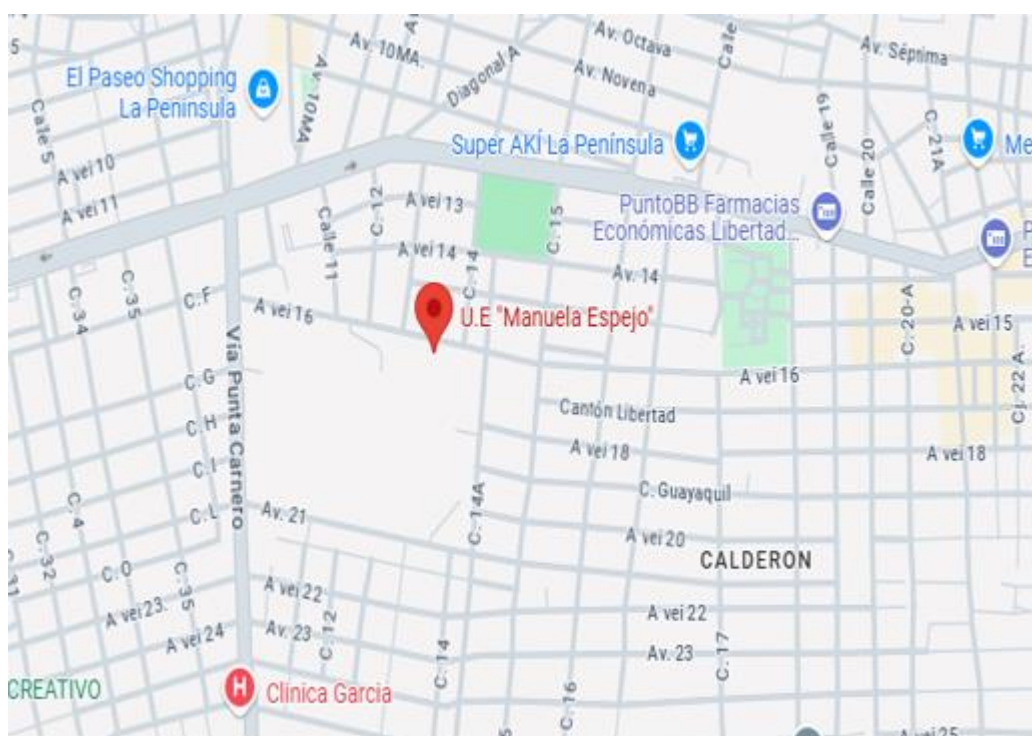


Fig. 3. Mapa de ubicación: UNIDAD EDUCATIVA “MANUELA ESPEJO”.

2.2 MARCO CONCEPTUAL

2.2.1 Inteligencia Artificial (IA):

La inteligencia artificial es una disciplina de la tecnología de la información que se encarga de desarrollar sistemas capaces de realizar o automatizar tareas que requiera de inteligencia humana. Esta inteligencia artificial permite que los aplicativos desarrollados puedan aprender y mejorar su capacidad de aprendizaje esto se logra integrando algoritmos de aprendizaje automático que se encarga de analizar grandes conjuntos de datos y ajustan sus parámetros para mejorar el rendimiento de los sistemas consiguiendo así prototipo que aprenda por sí solo [18].

2.2.1 Visión Por Computador:

La tecnología de visión por computador son disciplinas que combina conocimiento de informática, matemática y procesamiento de imágenes esto permite crear o diseñar sistemas que procesen imágenes y analicen la información visual de manera automática. Esta sirve para crear aplicativos o sistemas que capturen imágenes en tiempo real para procesar y realizar detecciones o reconocimientos logrando así poder usar esta información para analizar. Los algoritmos y técnicas específicas permiten usar esta tecnología para ajustar y que funciones en dispositivos móviles o cámara web [19].

2.2.2 Algoritmos De Detección De Objetos:

Los algoritmos de detección de objetos son métodos de visión por computadora que sirven para reconocer objetos en imágenes o videos. Estos algoritmos permiten conocer la posición y tamaño de las cosas dentro de las imágenes usando coordenadas o cajas delimitadoras. Son fundamentales en sistemas para reconocer prendas u otros tipos de objetos donde se extrae información como características, colores, textura y otra cosa más [20].

Los algoritmos de detección de objetos permiten integrar modelos CNN para mejorar su precisión en el reconocimiento de algún tipo de objeto permitiendo así crear sistemas capaces de procesar imágenes con precisión dando resultados favorables en su funcionamiento alguno de estos algoritmos son:

- **SSD:** Estos son algoritmos de visión artificial que permite detectar y localizar objetos dentro de imágenes o videos de forma rápida y precisa. Se distingue por su capacidad de detectar en una sola pasada, son ideales para hacer detecciones en tiempo real. Su entrenamiento es más largo debido a su complejidad ya que requiere una preparación más detallada y recursos computacionales profesional [21].
- **RestinaNet:** Este es otro modelo de detección de objeto que utiliza focal los para tener una mejor precisión en reconocer objetos pequeños y difíciles, más si en la detección existe desbalance entre el fondo y lo objetos esto algoritmo puede ser útil y eficaz ya que combina la velocidad de los modelos de una sola etapa con un rendimiento superior para objeto miniaturas [22].

2.2.3 Redes Neuronales convolucionales CNN

Esta son tipo de red neuronal que usa tareas de visión por computado para ayudar en la clasificación de imágenes, segmentación y detección de objeto. Aquí se integra los algoritmos de visión por computador y algoritmos de reconocimiento para el procesamiento de imagen ya sea foto o en tiempo real el cual permite que el prototipo aprenda automáticamente característica que se genera a partir de grandes cantidades de datos etiquetado para entrenamiento. Esto permite crea aplicaciones que no necesite una programación manual para reconocer patrones si no que es capaz de aprender a partir de los datos ingresado [23].

- **Yolo:** este es un modelo de detección de objetos en imágenes tiene una rapidez y eficaz en realizar reconocimiento a diferencia de otros modelos permite crear aplicaciones que pueda procesar imágenes en tiempo real, además que su entrenamiento no necesita de mucho recurso ya que cuenta con varias versiones que son adaptable a diferentes software o hardware que posee el usuario y son adaptable es decir que se puede utilizar para crear modelo de reconocimiento diversos [24].
- **Faster R-CNN:** este es otro ejemplo de redes neuronales CNN que permite hacer detecciones de objetos y aunque tiene dos etapas la primera utiliza Red de propuesta de regiones esto genera posibles regiones de los objetos y la segunda etapa clasifica y ajustan para determinar la ubicación del objeto.

A tener estas dos capas su precisión es más confiable y alta sin embargo para hacer detecciones en tiempo real suele ser lenta por lo tanto no son recomendables para estos tipos de tareas [25].

- **Mask R-CNN:** esta es una extensión de Fast R-CNN aparte de hacer detecciones permite usar el modelo para realizar segmentaciones y extraer máscaras, se los puede usar para extraer las coordenadas de la imagen procesada sin embargo para realizar detecciones y segmentación en tiempo real no son adecuados ya que el procesamiento es mucho más lento, además que se requiere un procesamiento de entrenamiento mucho más largo y su configuración es más complicada [26].

2.2.4 Transferencia de Aprendizaje (Transfer Learning)

La transferencia de aprendizaje es una técnica que permite transmitir los conocimientos adquiridos mediante un conjunto de datos grandes para resolver problemas. Este método de aprendizaje automático autoriza que el modelo entrenado previamente pueda seguir su autoaprendizaje usando un conjunto de datos más pequeños o especializado. En los sistemas de visión por computadora como el reconocimiento de prendas usar esta técnica es útil, ya que trabajar con un conjunto de datos específico usar modelo preentrenado como Yolo, ResNet y ajustarlo usando fine-tuned con datos propios, optimizando el proceso y disminuyendo la cantidad de datos requeridos como el tiempo de entrenamiento [27].

2.2.5 Aprendizaje Profundo (Deep Learning)

El aprendizaje profundo es una rama que trabaja con redes neuronales profundas esto ayuda a resolver tareas complejas de análisis y clasificación, esto las hace diferentes a otras redes de aprendizaje automático, las redes neuronales convolucionales CNN son un ejemplo de aprendizaje profundo ya que tienen la capacidad de aprender características, se las utilizan especialmente para tareas de visión por computadora, donde se pueden clasificar, segmentar y detectar objetos. El uso de redes profundas permite que el sistema aprenda a reconocer patrones de manera precisa sin tener que usar etiquetado manual, el objetivo es mejorar la precisión de reconocimiento [28].

2.2.6 Segmentación de Imágenes

La segmentación de imágenes son proceso de visión por computador el cual consiste en dividir las fotos en segmento pequeño para después analizar esto permite identifica y aislar las regiones de interés para recopilar datos el cual se pueda usar para el reconocimiento de objeto. Yolo o Mask R-CNN usa algoritmos de segmentación ya que generar máscaras especifica de las imágenes especialmente para identificar color o texturas, son técnicas que mejora la precisión y eficiencia en las detecciones más si es usado en tiempo real [29].

2.2.7 Detección en Tiempo Real

Las detecciones en tiempo real son fundamental en aplicaciones que trabajan por visión de computadora donde se busca proporcionar una experiencia interactiva en detecciones, esto permite procesar imágenes instantáneamente a media que se hacen captura proporcionando respuesta al instante sobre el reconocimiento. Los algoritmos como Yolo y SSD son capaces de realizar detecciones en tiempo real donde su proceso es rápido y preciso la diferencia es que Yolo permite trabajar con dataset pequeño proporcionado nos un entrenamiento más rápido a diferencia de SSD el cual necesita un conjunto de dato más grande. Esto son modelos ideales para desarrollar aplicaciones móviles donde el proceso de imagen con velocidad sea rápido y sin retrasos [30].

2.2.8 Reconocimiento de Patrones

El reconocimiento de patrones es una técnica que se utiliza para identificar patrones o características en datos, que pueden ser imágenes, sonidos, o incluso señales. En visión por computadora, el reconocimiento de patrones es esencial para tareas como la identificación de objetos, colores, y texturas en las imágenes. Para este procedimiento se utilizan tecnología de machine Learning consiste en que las computadoras puedan aprender a clasifica, identifica, reconocer los objetos de manera autónoma [31], hay tres clasificaciones principales que son:

- **Clasificación supervisada:** Aquí se utilizan modelos que ya este entrenado de manera correcta para realizar cualquier tipo de clasificación o para saber el comportamiento de los objetos a identificar [32].

- **Clasificación no supervisada:** Este es un método el cual no se necesita etiquetas predefinidas o resultado conocido para identificar patrones. Es una técnica exploratoria el cual consiste en agrupar datos similares en base a características [32].
- **Clasificación parcialmente supervisada:** Es una técnica que se caracteriza por combinar métodos supervisado y no supervisado, se la puede emplear cuando los dato son mixto es decir solo una parte dispone de etiqueta y el resto no el propósito es aprovechar todos los datos tenga o no etiquetas sea utilizado para que modelo a entrena sea preciso en su entrenamiento y reconocimiento [32].

2.2.9 YOLO

Yolo es una familia de redes neuronales convolucional que se la puede emplear para la detección de objetos en tiempo real, imágenes o videos. Se lo diferencia de otras redes neuronales ya que requiere de múltiples etapas para el procesamiento en tiempo real mientras que Yolo solo utiliza una sola capa para aborda de manera eficaz la predijo ubicando lo objeto y sus clases en una sola pasada esto permite que el modelo sea más rápido que otros métodos tradicionales [33].

YOLO tiene varias versiones como YOLOv2, YOLOv3, YOLOv4, YOLOv5 y YOLOv8, cada versión ha mejorado en lo que es precisión y velocidad para abordar desafíos como la detección de objeto más pequeños, para detecciones de objeto en diferentes escalas además que puede ser entrenado según la necesidad del usuario [34].

Una de las versiones reciente de esta familia de redes neuronales convolucionales es YOLOv8 ofrece una mejora a comparación de su otras versiones, como utiliza redes neuronales convolucionales permite extraer las coordenadas y característica de las imágenes, este red neuronal se destaca por tener la capacidad de trabajar con modelo preentrenado aplicando técnica de transferencia de aprendizaje esto facilita que el usuario no tenga la necesidad de disponer de grandes volúmenes de datos ya que puede crear conjuntos de información personalizada y adaptar al modelo para realizar tareas específicas facilitando el proceso de entrenamiento mejorando la eficiencia [35].

Una de las características que posee YOLOv8 es que trabaja con framework como PyTorch que es una técnica de aprendizaje profundo muy popular, facilita la creación, el ajuste y el entrenamiento ofreciendo una gran flexibilidad y fácil de uso del modelo además que trabajar con esta versión brinda un informe detallado de los resultados del entrenamiento donde se visualiza las matrices y eficaz de modelo entrenado [36].

2.2.10 Dataset

Los dataset son conjunto de datos que están estructurado de manera correcta para su procesamiento, análisis o uso según la aplicación específicas. La información que contenga el dataset varía según la necesidad del usuario. En contexto al proyecto el dataset tiene una recopilación de imágenes su construcción se basa en una recolección de imágenes de internet y foto tomada por el administrado, este conjunto de datos es guardad en el Big Data igual que sus etiquetas servirá para el entrenamiento de la red neuronal para el reconocimiento de patrones de prendas de vestir.

Los dataset personalizado el proceso de creación suele se largó y cansado ya que se debe recoleta la información necesaria, aunque actualmente ya hay dataset con datos para entrenar redes neuronales, sin embargo, contienen grandes cantidad de foto el cual ajustarlo para usarlo en un modelo específico puede ocupar recurso del computador que tal vez el usuario no posee uno de los ejemplo tenemos DeepFashnion2 Dataset este contiene 491,000 imágenes diversas con trece categorías de prendas con sus respectivas etiquetas sin embargo debido a la gran cantidad de información, puede presentar retos como su procesamiento y ajuste de dato el cual se requiere de técnicas de preprocesamiento que requieren de tiempo y recursos, sin no se aplica la técnicas adecuadamente los datos procesados puede estar con error retasando más el procesos de entrenamiento [37].

2.2.11 Backend

El backend es una parte fundamental e invisible de un aplicativo o página web, permite tener un funcionamiento correcto y seguro, se caracteriza por gestionar la lógica del negocio y el procesamiento de datos, asegurando que la aplicación se

opere de manera eficaz y confiables. Esto se encarga de interactuar con la base de datos ya que permite almacenar, recupera y lista todo el contenido, los lenguajes de programación para el desarrollo de backend son Python, JavaScript, Ruby, Java y PHP esto permite crear servicios y APIs que actúan como un puente de conexión entre la base de datos y la aplicación o sitio web, administrando correctamente el procesamiento de la información y presentarlo de forma adecuada al usuario [38].

2.2.12 Python

Python es un lenguaje de programación el cual tiene varias versiones que permite crea codificaciones el cual sirva para páginas web o análisis de datos, trabajar con inteligencia artificial, automatizar proceso o crear scripting. Este lenguaje de programación es de alto nivel ya que se integra con todo tipo de sistemas y aumenta la velocidad del desarrollo al contar con bibliotecas profesionales ajustándose no solo a los sistemas de programación, sino que además se adapta a sistemas operativos computacionales como Windows, macOS, Linux y Unix [39].

2.2.13 APIs

API es un conjunto de reglas que permite la interacción entre sitio web o aplicaciones además que trabaja con lenguaje de programación en Python creado un entorno profesional. Esto permite integran y trabajar con algoritmo de detección de objeto, librerías de visión por computador y almacenar la información en una base de datos. Son fundamentas ya que actúan como intermediario facilitando que el sistema creado pueda acceder a funcionalidades o datos del sistema sin necesidad de componente interno.

2.2.14 Visual Studio Code

Es un editor de codificación gratuito, soporta diferente lenguaje de programación para el diseño de interfaz de página web o aplicaciones móviles permitiendo que se convierta en una de las herramientas más populares. Ofrece característica como depuración, control de versiones, además de al trabajar con varios lenguajes se crea entorno personalizado evitando saturaciones en el sistema operativo del usuario.

2.2.15 phpMyAdmin

phpMyAdmin es un administrador de base de datos MySQL de una manera sencilla y con una interfaz amigable, permite realizar tareas para creación, modificación, actualización, eliminación, registro y ejecutar consulta a la base de datos desde un conector como un servidor o un API. Se lo utiliza comúnmente en entornos de desarrollo para sistemas web o móviles para la administración de datos de MySQL [40].

2.2.16 XAMPP

Es un software gratuito que permite enviar solicitudes su objetivo principal es realizar gestiones a la base de datos MySQL desde un servidor, es una herramienta fácil de instalar con una interfaz que sirve para autorizar la habilitación de puertos con el propósito de realizar la conectividad de los APIs con la base de datos [41].

2.2.17 Framework

Es un conjunto de bibliotecas o herramientas que permiten desarrollar aplicaciones gracias a la estructura que tienen para construir servidores o APIs, un ejemplo de este framework es Flask este permite crear solicitudes en Python controlando así la interacción entre la interfaz con la base de datos u otro componente dando una mejor estructuración al proyecto además de que permite trabajar con bibliotecas de visión por computadora y modelos entrenados para usarlos según la necesidad o tareas específicas [42].

2.2.18 Flutter

Flutter es un entorno de desarrollo compatible con Google es utilizado para crear interfaces de aplicaciones móviles o web, su lenguaje de programación es Dart tiene un alto rendimiento y es altamente personalizado. Es una herramienta conocida por su rapidez y capacidad en compilar aplicaciones para diversas plataformas, su fácil uso ejecuta aplicaciones móviles de manera rápida además se puede conectar con los APIs para crear una conectividad e interacción de las funcionalidades que desean tener el sistema a desarrollar [43].

2.2.19 OpenCV

OpenCV son bibliotecas libres de uso para trabajar con algoritmos de visión y aprendizaje automático, esto permite el procesamiento de imágenes o videos para

realizar extracciones de característica como la detección de objeto y analizar en tiempo real. Es compatible con varios lenguajes de programación como Java y Python se los puede integrar en varios proyectos que necesita realizar tareas como el reconocimiento o detecciones en tiempo real [44].

2.2.20 Algoritmo de Colores

Este algoritmo es un conjunto de procesos que manipulan e identifican colores en una imagen. Se puede usar en lenguaje como Python a través de librerías como RGB, CMYK, HSL, KMeans, estos algoritmos permiten ajustar la saturación o el brillo de las fotos para realizar una correcta detección de colores, se crea un diccionario de colores ya que al identificar colores dentro de una imagen los resultados son códigos. Al utilizar una lista de los códigos de colores se facilita la visualización de la respuesta del proceso [45].

2.2.21 PyCharm Community Edition 2023.3.5

PyCharm Community Edition es un aplicativo gratuito que permite crear un entorno de desarrollo Python para el fácil uso de instalación de librerías o bibliotecas evitando saturaciones en el sistema operativo del computador. Es popular ya que soporta proyectos grandes como el uso de modelos entrenados para detecciones de objeto y librerías como OpenCV, esto facilita una estructuración profesional para la elaboración de proyectos de desarrollo para aplicaciones móviles y web [46].

2.2.22 Interfaz de Usuario Adaptativa para Personas con Discapacidad Visual

Este proyecto trata del desarrollo de un aplicativo móvil y un sistema web para personas con discapacidad visual, dirigidas a aquellas que tienen trastornos como monocromatismo, una condición en la cual la persona solo puede ver en tonos grises, negro y blanco, por lo tanto, se necesita diseñar una interfaz con estos tonos con alto contraste para asegurar que los elementos visuales sean fáciles de distinguir. Esto asegura que los usuarios puedan tener una facilidad de navegación en la plataforma o aplicativo, además se implementan otras medidas accesibles como usar audio para leer los resultados de la detección realizada en tiempo real [47].

2.3 MARCO TEÓRICO

2.3.1 Sistema de reconocimiento de objetos móviles para personas con discapacidad visual.

Los sistemas que sirven para el reconocimiento de objetos móviles diseñados especialmente para asistir a personas con problemas visuales tienen enfoque principal en el utilizar redes neuronales convolucionales en aplicaciones móviles, esto permite identificar objetos en tiempo real desde el dispositivo a través de la cámara. En la investigación resalta la importancia de la precisión en el reconocimiento de objetos, especialmente cuando se debe proporcionar informes a las personas con discapacidad visual. Otro factor importante es que los sistemas creados para procesar y responder las captura en tiempo real deben ser de manera rápida, fluida y efectiva, además de que los diseños e interfaces tiene que ser accesibles para que el usuario pueda interactuar sin ningún problema [47].

Los sistemas de reconocimiento de objetos cuentan con un entorno controlado, aunque existen desafíos cuando se trabaja con escenarios complejos, aun así, deben ser capaces de adaptarse a distintos tipos de objetos y entorno si perder la precisión, sus respuestas de proceso deben ser rápidas para que las personas con discapacidad visual puedan interactuar con su entorno de forma autónoma. La importancia de mejora los sistemas de detección de objeto son importante además que no solo esta parte es fundamental, sino que también la usabilidad y accesibilidad de las aplicaciones esto garantizaran que su uso sea fácil para los usuarios [47].

2.3.2 Avances recientes en tecnologías asistidas basada en aplicaciones móviles para personas con discapacidad visual

La revisión sobre los avances tecnológico para asistir a personas con discapacidad visual basada en dispositivo móviles es importante porque se refleja la parte esencial o recurso de estas aplicaciones además de que se enfoca en problema a resolver para persona con discapacidad visual. Hay aplicaciones o sistema que utilizan el procesamiento de imagen, como el uso de redes neuronales para esto proceso, donde el fin o propósito de esto sistemas es mejora la autonomía de los usuarios, proporcionando herramientas que le permiten interactuar con su entorno

usando técnica como navegación por voz, la detección en tiempo real entre otras opciones según la funcionalidad del aplicativo [48].

Las tecnologías son importantes ya que permite personalizar o diseñar sistemas dependiendo la necesidad de las personas con problema visuales. Esto debe ser adaptable y garantizar su facilidad de uso, abordando así las dificultades que aún persisten en la integración de todas estas herramientas garantizando su eficiencia y más si es en tiempo real. Las aplicaciones móviles como asistentes han evolucionado en mucho factor como la adaptabilidad, accesibilidad y operación con la intención de mejorar sus funciones sea precisa [48].

2.3.3 Evaluación de sistemas de reconocimiento de objetos para personas con discapacidad visual.

Los sistemas de reconocimiento de objeto diseñado para tareas específicas que sirven de asistencia a persona con discapacidad visual debe ser evaluados según los criterios principales como la precisión, la velocidad en realizar detecciones en tiempo real y facilidad de uso. La precisión en reconocimiento de objeto es fundamental ya que un sistema no identifique de forma correcta es riesgoso ya que pone en peligro a los usuarios, además que la prueba para cada aplicación debe ser importante ya que se aseguran de que el aplicativo puede procesar imágenes rápidas con un grado de precisión brindando seguridad en su uso para persona con problema visuales [49].

2.3.4 Desarrollo de un sistema de asistencia para persona con discapacidad visual en un ambiente estructurado empleado visión artificial.

El desarrollo de sistemas de asistencias para persona con problema visuales utilizando algoritmo de visión artificial es una mejora en la navegación de los usuarios. Este sistema utiliza entorno el cual consiste en utilizar la cámara del dispositivo para hacer detecciones en tiempo real y proporcionar sus respuestas de forma auditiva guiando así a las personas con discapacidad visual. Esta tecnología permite el reconocimiento de objeto y obstáculos de un entorno cerrado o abierto, brindando información detallada para facilitar la movilidad de los individuos, gracias a esto se puede crear una interacción segura y eficiente mejorando la autonomía, los

avances tecnológicos son herramientas fundamentales para el diseño de aplicaciones que permiten realizar tareas específicas creando asistentes capaces de ayudar a personas con discapacidad visual [50].

2.3.5 Herramientas para ayudar a detectar las características de la ropa mediante imágenes.

La creación de herramienta que asistan a persona con discapacidad visual en tareas específicas son indispensable, una de las dificultades que alguna persona tiene es vestirse, gracias a los avances y el uso de redes neuronales convolucionales se a creado herramienta que permite identificar prendas de vestir y sus colores mediante el procesamiento de imágenes, proporcionando una descripción que facilita la selección de ropa. Las aplicaciones que emplean redes neuronales para la detección de patrones integran algoritmos KMeans para extraer colores de estos reconocimientos su capacidad de reconocer es dependiendo la clase y colores el cual fue creado el aplicativo, esto sistema mejora la autonomía de las personas con discapacidad visual al permitirles tomar decisiones informada sobre su vestimenta [51].

2.4 REQUERIMIENTOS

2.4.1 REQUERIMIENTO FUNCIONALES

A continuación, se presenta los requisitos funcionales para la ejecución del aplicativo móvil y sistema web.

Nº	Descripción del Requerimiento del sitio web
1	El sitio web permite el registro de correo y clave para los usuarios esto permite que pueda acceder usando sus credenciales
2	Se necesita un correo y contraseña para iniciar sesión, si hay nuevo usuario puede registrarse para poder usar el sistema web.
3	El sitio web permite subir imágenes para alimentar el dataset del modelo entrenado

4	Al cargar la imagen de las prendas del usuario automáticamente se usa el modelo entrenado para hacer segmentaciones y extraer coordenadas. Datos que se visualizan en el apartado de listar.
5	Se registra las prendas antes de actualizar el dataset ya que una al actualizar este realiza un aumento de datos ajustando según el brillo y saturación.
6	Una vez actualizado el dataset este creará automáticamente el formato adecuado para el proceso de reentrenamiento del modelo con los nuevos datos.
7	Internamente se ejecuta comando para reentrenar el modelo permitiendo que este se alimente con los datos del usuario dando como resultado datos del proceso ejecutado.

Tabla 2 Requerimiento Funcional Del sistema web

N°	Descripción del Requerimiento del aplicativo móvil
1	El aplicativo cuenta con una interfaz el cual tiene un botón para iniciar sesión aquí se extraen automáticamente datos de dispositivo.
2	Una vez iniciada sesión se dirige a la ventana del aplicativo el cual tiene la cámara para hacer detecciones en tiempo real
3	El aplicativo abrirá la cámara y mediante audio comunicará el enfoque de la prenda para realizar la detección.
4	La aplicación tiene dos opciones iniciar detección el cual solo proporcionará los resultados de la prenda y color.
5	La segunda opción del aplicativo es combinación este botón permite hacer la detección y dar como respuesta la prenda color y con que puedes combinar.

6	En la parte superior cuenta con un icono el cual permite cambiar la cámara es decir que se puede usar cámara frontal/trasera
7	Los resultados de cada opción no solo estarán disponibles textualmente, sino que también es auditiva.

Tabla 3 Requerimiento Funcional del aplicativo móvil

2.4.2 REQUERIMIENTO NO FUNCIONALES

A continuación, se presenta los requisitos no funcionales para la ejecución del aplicativo móvil y sistema web.

Nº	Descripción del Requerimiento del sitio web y Móvil
1	Ambos sistemas usaran el modelo entrenado para la segmentación y detección de los patrones de prendas
2	Las interfaces del sistema web y el aplicativo móvil tiene que ser sencilla para que el usuario pueda interactuar sin problema
3	El desarrollo del API contara con un backend para mantener una comunicación en el servicio tanto del aplicativo como sitio web.
4	El lenguaje de programación a utilizar será en un entorno Python configurado en PyCharm Community Edition.
5	Se registra las prendas antes de actualizar el dataset ya que una al actualizar este realiza un aumento de datos ajustando según el brillo y saturación.
6	Se utilizará labelme para realizar el etiquetado de las imágenes que esta en el dataset el cual servirá para realizar el entrenamiento.
7	Para el entrenamiento se utilizará YOLOv8 el cual permite usar modelo preentrenado y ajustarlo para una tarea específica.

8	Se uso librerías como ultralytics, OpenCV, KMeans para la detección de prendas y colores.
9	Se utilizará XAMPP para gestionar la base de datos MySQL
10	phpMyAdmin servirá para crea la base de datos y gestionar las consultas
11	Android Studio servirá para desarrolla el aplicativo móvil el cual se lo realizará con un lenguaje Dart de flutter
12	Se utiliza visual Studio para el desarrollo del sitio web en lenguaje de programación angular.

Tabla 4 Requerimiento no funcional del sistema web y aplicativo móvil

2.5 COMPONENTE DE LA PROPUESTA

2.5.1 ARQUITECTURA DEL SISTEMA

CLIENTE - SERVIDOR

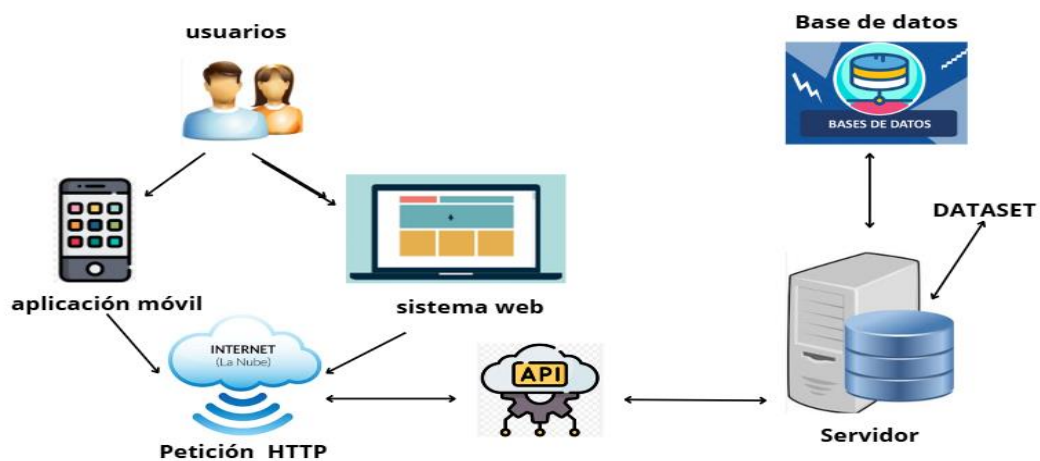


Fig. 4. Arquitectura del sistema cliente-servidor

La arquitectura que se utilizó es cliente y servidor ya que facilita la interacción de los usuarios (clientes) con los servidores a través de una comunicación organizada y eficiente, donde se realizan solicitudes de servicios (Request) desde el aplicativo móvil o sistema web al servidor, este se encarga de procesar la solicitud y devuelve una respuesta (Response) [52].

El uso de esta arquitectura facilita un control eficaz sobre las solicitudes realizadas y enviadas al servidor permitiendo una gestión y optimización de los servicios proporcionados. Además, que ofrece una capa de seguridad, garantizando que la información solicitada sea procesada adecuadamente evitando problemas como accesos no autorizados o datos mal gestionados. Estos servidores están diseñados para manejar múltiples solicitudes de manera eficiente permitiendo que los usuarios interactúen de forma confiable y fluida [52].

2.5.2 INSTRUMENTOS DE DESARROLLO.

El desarrollo del proyecto los instrumentos utilizados fueron los siguientes:

Equipo de Hardware:

- Una laptop con un procesador 11th Gen Intel(R) Core (TM) i5-1135G7 @ 2.40GHz 2.40 GHz y 8 GB de memoria RAM
- Un celular Galaxy A03s modelo SM-A037M/DS con 4GB de memoria RAM, tiene un procesador MediaTek Helio P35, con triple cámara principal de 13 MP + 2MP +2MP y cuenta con una cámara frontal de 5MP su versión es un Android 11.

Equipo Software:

- PyCharm Community Edition 2023.3.5
- Android Studio
- Flutter 3.24.3
- Python 3.8
- Librerías
- Visual Studio Code

2.5.3 CREACION DEL DATASET Y ETIQUETADOS PARA ENTRENAR LA RED NEURONAL CONVOLUCIONAL CNN.

La recopilación de datos para la creación del dataset fue tomar fotografías de diversas prendas de ropa y una búsqueda en línea para obtener más imágenes, el dataset creado cuenta con cuatro clases blusa, camisa, buso y shorts esta tiene alrededor de 500 imágenes cada una de ella. Se utilizó la herramienta Labelme para

el etiquetado de cada imagen, ya que permite marcar las áreas de interés con caja delimitadora y asignarle la clase correspondientes, completado el proceso se usó el comando `labelme2yolo` para convertir las anotaciones al formato YOLO, el cual es el estándar para entrenar la red neuronal convolucional.

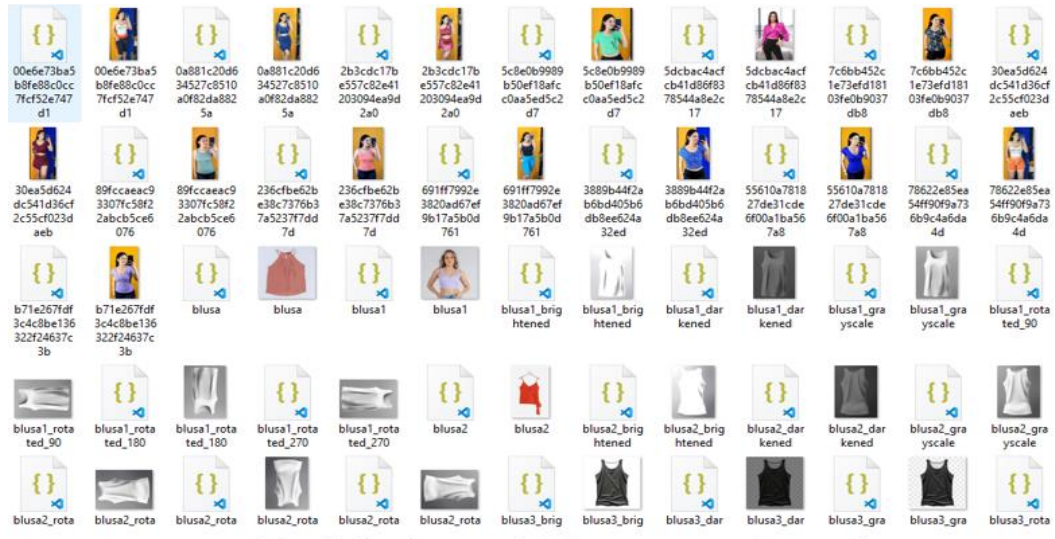


Fig. 5. Imágenes del dataset y su etiquetado

```

Converting cm4 - copia (4).json for train ...
Converting WIN_20241020_23_54_35_Pro - copia.json for train ...
Converting WIN_20241020_23_54_52_Pro - copia.json for train ...
Converting cm16 - copia (4).json for train ...
Converting cm18 - copia (3).json for train ...
Converting cm5 - copia.json for train ...
Converting Camista5_brightened.json for train ...
Converting cm4.json for train ...
Converting camisaCasual2s - copia (2).json for train ...
Converting WIN_20241020_23_54_13_Pro - copia (2).json for train ...
Converting cms - copia (3).json for train ...
Converting cm20 - copia.json for train ...
Converting WIN_20241020_23_54_45_Pro - copia (2).json for train ...
Converting Camiseta17_grayscale.json for train ...

```

Fig. 6. Conversión al formato YOLO

2.5.4 PROCEDIMIENTO DE ENTRENAMIENTO Y SUS RESULTADOS

El proceso de entrenamiento se configuro el siguiente código que consiste en especificar el número de épocas (30), la ubicación del archivo dataset en su formato correcto, como YOLO usa modelo preentrenado para no entrenar desde cero el modelo base utilizado es `yolov8n-seg.pt`, se ajusta el tamaño del batch. Durante el

procesamiento, el modelo se ajustó sus parámetros para mejorar la precisión en la detección y segmentación de las prendas.

```
def entrenar_yolo(json_dir):
    try:
        print("Iniciando el entrenamiento de YOLO...")
        dataset_yaml_path = os.path.join(json_dir, 'dataset.yaml')

        # Ejecutar el entrenamiento de YOLO
        command = f'yolo task=segment mode=train epochs=30 data={dataset_yaml_path} model=yolov8n-seg.pt imgsz=416 batch=4'
        with open('yolo_training_log.txt', 'w') as log_file:
            subprocess.run(command, shell=True, check=True, stdout=log_file, stderr=log_file)
        print("Entrenamiento de YOLO completado. Revisa el archivo 'yolo_training_log.txt' para detalles.")

    except subprocess.CalledProcessError as e:
        print(f"Error al ejecutar el entrenamiento de YOLO: {e}")
```

Fig. 7. Código de entrenamiento YOLO

Una vez terminado la fase de entrenamiento se crear automáticamente una carpeta el cual contiene todos los resultados del modelo entrenado esta carpeta tiene el modelo con el nombre best.pt que está listo para usar en la aplicación para detecciones y en la página web para realizar las segmentaciones. A continuación, se describe cada explicación los resultados del procedimiento.

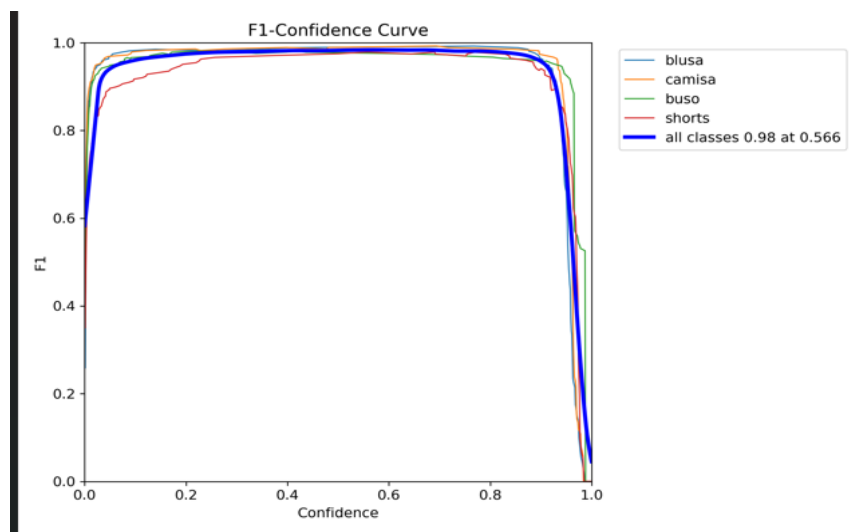


Fig. 8. F1-Confidence curva

La figura 8 muestra la relación entre el F1-score que es una métrica para saber el problema de clasificación el cual se refiere a la precisión y sensibilidad, además que describe el nivel de confianza del modelo para cada clase. El resultado global de modelo entrenado es de 0.98 esto significa que el rendimiento del modelo es perfecto en termino de precisión, es decir que el modelo realizara muy pocas

predicciones incorrectas logrando identificar los patrones de prenda de manera eficaz.

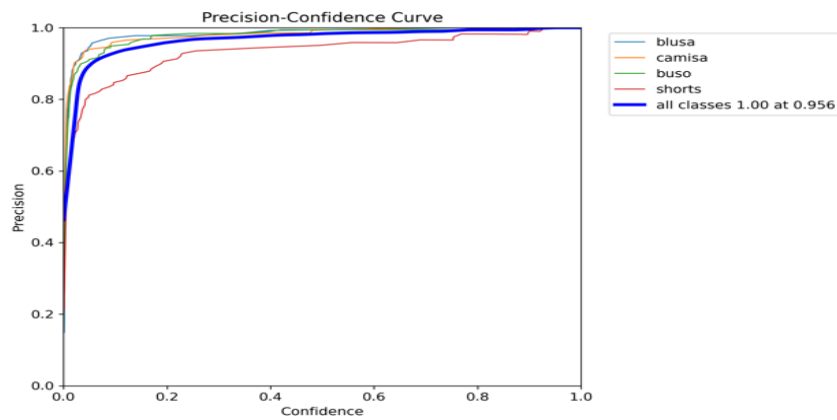


Fig. 9. Imagen de la precisión

En esta grafica de la figura nueve se muestra la curva de precisión en función de la confianza donde la línea azul refleja la precisión promedio para todas las clases el cual alcanza un valor optimo de 1.00 cuando el nivel de confianza es 0.956. Esto significa que al ajustar el umbral de confianza a este valor el modelo tendrá una máxima precisión, aunque con este umbral alto se corre el riesgo de afectar la sensibilidad del modelo por lo que se recomienda probar umbral hasta encontrar la adecuada para realizar detecciones.

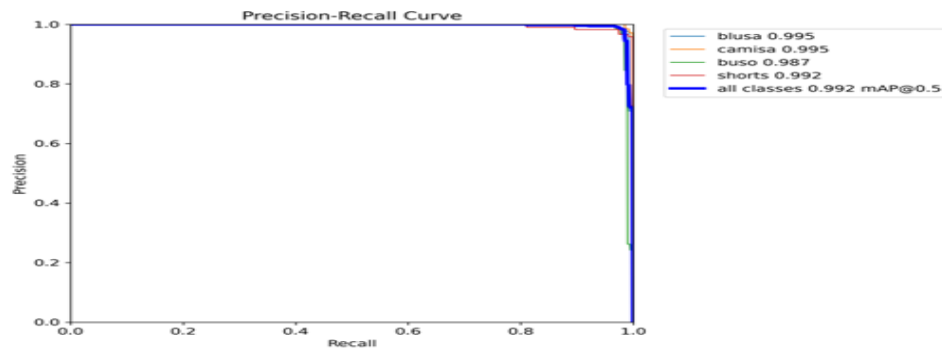


Fig. 10. Imagen de la precisión y sensibilidad

En la gráfica de la figura 10 se muestra que el modelo entrenado tiene un desempeño bueno para realizar la clasificación de las clases con la que se entrenó, el valor [mAP@0.5](#) representa el promedio de la precisión esto permite tener un desempeño adecuado para clase lo que significa que las detecciones se realizasen de manera efectiva.

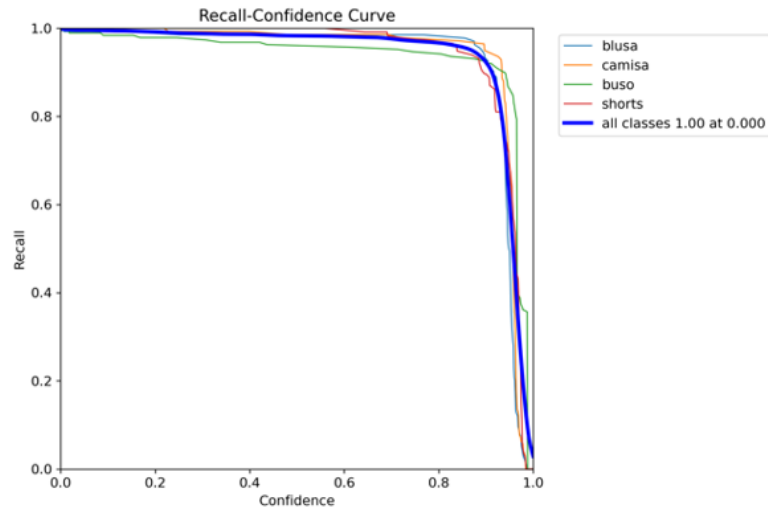


Fig. 11. Imagen de la confianza y sensibilidad

La grafica de la figura 11 muestra un alto porcentaje de sensibilidad lo que significa identificar correctamente la mayoría de los ejemplos de clase además tiene una alta confianza esto brinda seguridad en la realizar predicciones. Esto es favorable porque da a conocer que incluso con un nivel medio de confianza el modelo es bueno para reconocer.

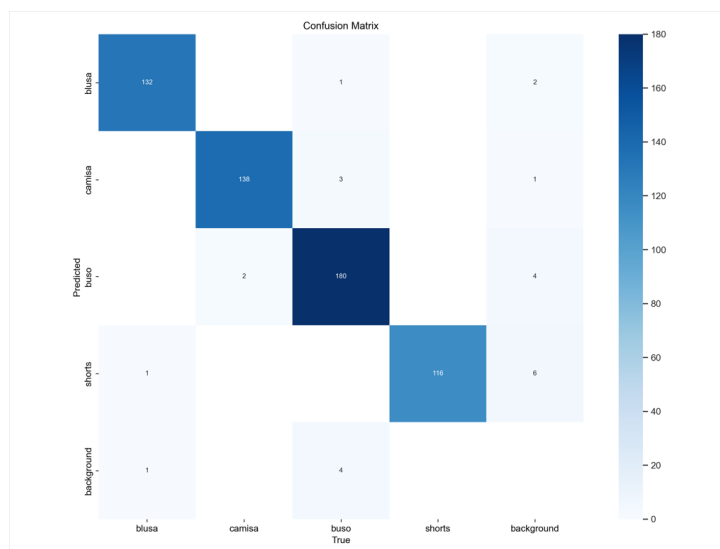


Fig. 12. Imagen de la métrica de confusión

La figura 12 muestra los resultados de la matriz de confusión esta es una herramienta valiosa que permite evaluar el desempeño del modelo entrenado. Se puede obtener una visión precisa de las fortalezas y debilidades del modelo. Como respuesta obtenida muestra que tiene un buen desempeño en la clasificación de prendas, más con las clases camisa y busos. Sin embargo, en algunas pruebas puede confundirse con otra clase con fondos, pero la probabilidad es baja.

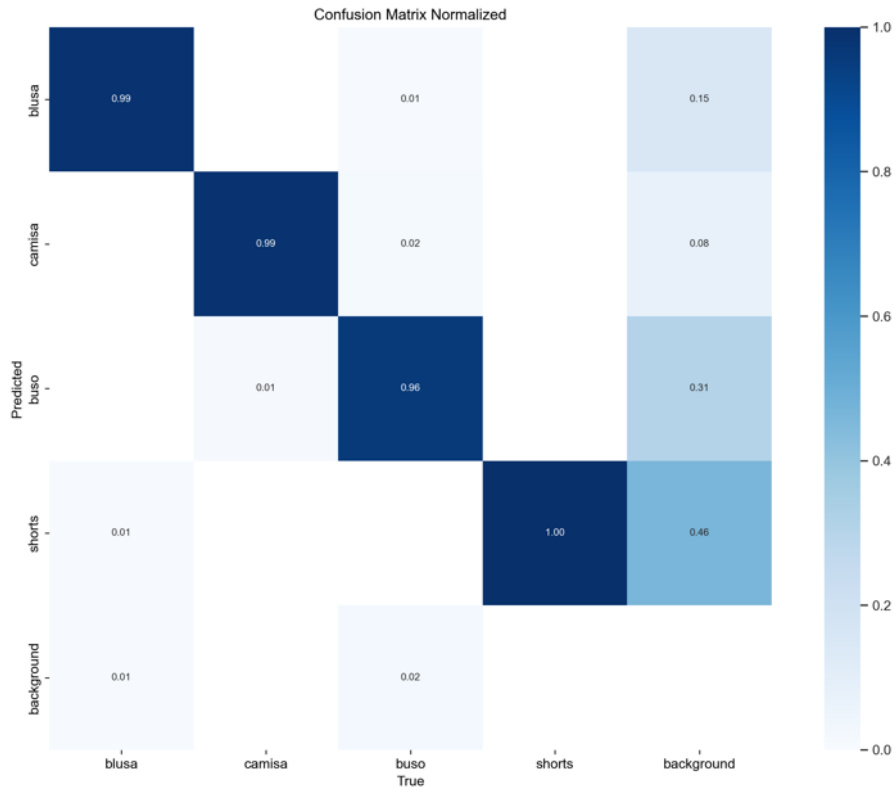


Fig. 13. Imagen de la métrica de confusión

La figura 13 muestra los resultados de la matriz de confusión normalizada en esta parte se muestra la cantidad de datos que este clasificado correctamente e incorrecta como se puede apreciar las clases camisa y buso tiene un buen desempeño a diferencia de shorts tiene un desempeño un poco bajo sin embargo no es impedimento para que el modelo no reconozca esta clase.

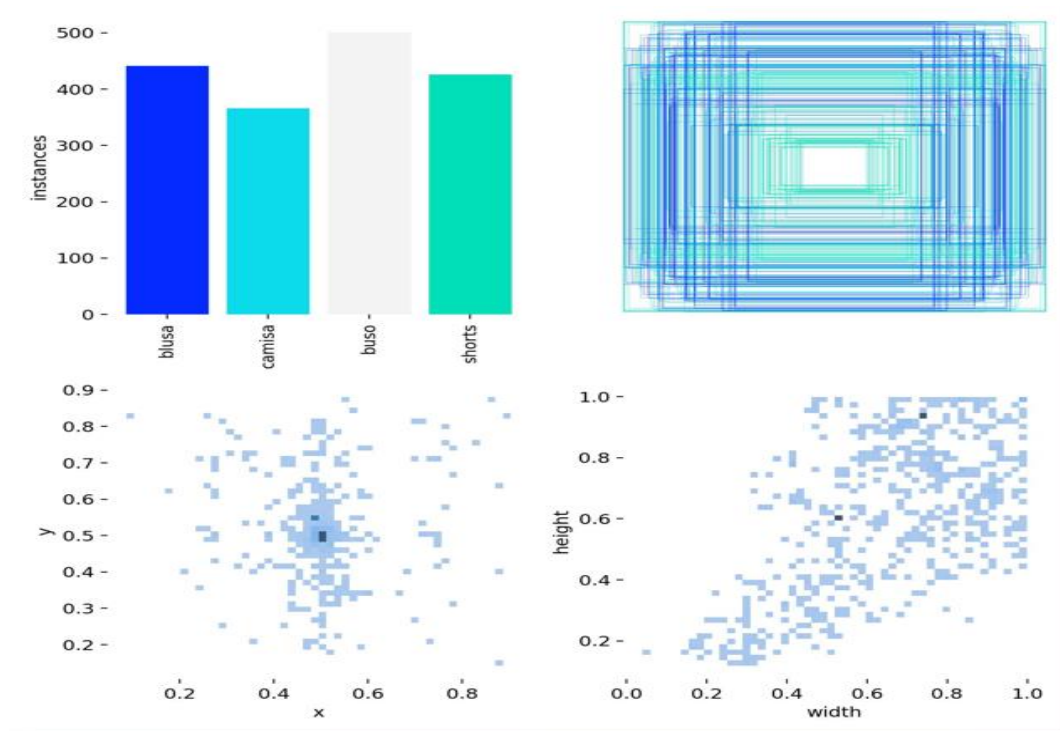


Fig. 14. Análisis de Datos

El gráfico de la figura 15 ofrece una visión general de la clase tomando en cuenta las posiciones y los tamaños de las cajas delimitadoras, aquí se muestra que el modelo tiene un buen desempeño para el reconocimiento de prendas además que demuestra que las clases esta equilibrada algo que es favorable para el aprendizaje del modelo entrenado.

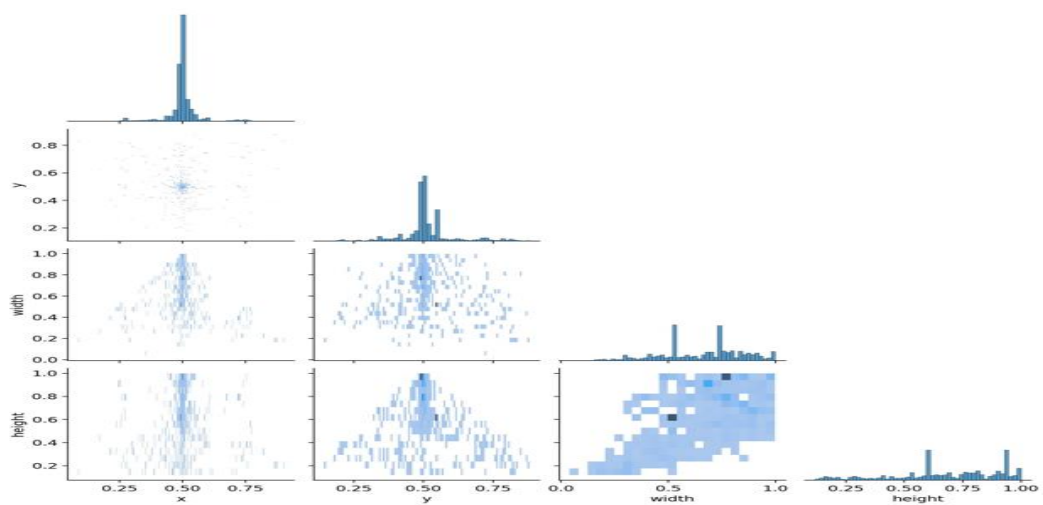


Fig. 15. Matriz de dispersión

El modelo entrenado tiene una buena distribución y concentración en la figura 16 se puede visualizar que los resultados son buenos, ya que no se observan distribuciones atípicas ni correlacionales que pueda afectar el desempeño.

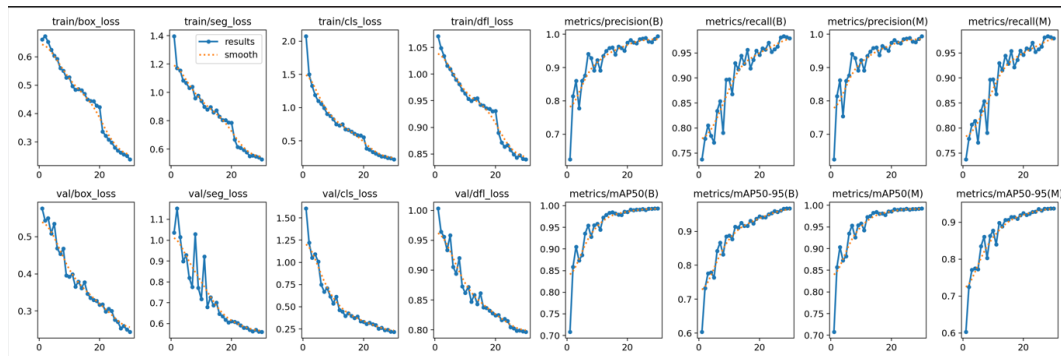


Fig. 16 resultado por época

2.5.5 ALGORITMO DE DETECCIÓN DE COLORES

El algoritmo de detección de colores toma una región específica de la imagen, como el proceso de detección de prendas genera una caja delimitada esta es extraída y convertida al formato RGB, el cual es un modelo de colores, se usó algoritmo de KMeans para agrupar los colores dominante en la imagen permitiendo calcular el color más predominante. Los resultados se muestran en el formato RGB el cual es transformado a un nombre de color utilizando las bibliotecas webcolors, además que en base al color detectado se usa código de combinación el cual se usa un diccionario de combinaciones predefinidas.

```
def get_dominant_color_kmeans(image, bbox, n_clusters=3):
    x1, y1, x2, y2 = bbox
    # Extraer la región de la imagen delimitada por la caja
    region = image[int(y1):int(y2), int(x1):int(x2)]
    # Convertir la región a formato RGB
    region_rgb = cv2.cvtColor(region, cv2.COLOR_BGR2RGB)
    # Aplanar la imagen en una lista de píxeles RGB
    pixels = region_rgb.reshape(-1, 3)
    # Aplicar KMeans para encontrar el color dominante
    kmeans = KMeans(n_clusters=n_clusters, n_init=10, random_state=42)
    kmeans.fit(pixels)
    # Contar la frecuencia de cada cluster
    counts = np.bincount(kmeans.labels_)
    dominant_color = kmeans.cluster_centers_[np.argmax(counts)]
    return tuple(map(int, dominant_color))
```

Fig. 17. Algoritmo para la detección de colores

```

def get_color_name(rgb):
    try:
        color_name = webcolors.rgb_to_name(rgb)
    except ValueError:
        min_distance = float('inf')
        color_name = "desconocido"
        css3_colors = {
            "azul": (240, 248, 255),
            "blanco": (169, 169, 169),
            "rojo": (220, 20, 60),
            "negro": (0, 0, 0),
            "verde_oscuro": (0, 100, 0),
        }
        for name, hex_rgb in css3_colors.items():
            distance = np.sqrt((hex_rgb[0] - rgb[0]) ** 2 + (hex_rgb[1] - rgb[1]) ** 2 + (hex_rgb[2] - rgb[2]) ** 2)
            if distance < min_distance:
                min_distance = distance
                color_name = name
        return color_name.lower()

```

Fig. 18. Función para identificar el nombre del color

```

colores_combinaciones = {
    "negro": ["blanco", "gris", "rojo", "amarillo", "azul", "verde"],
    "rojo": ["blanco", "negro", "gris", "amarillo", "azul"],
    "verde": ["blanco", "negro", "gris", "amarillo", "azul"],
    "azul": ["blanco", "negro", "gris", "amarillo", "rojo"],
    "amarillo": ["negro", "blanco", "rojo", "azul", "gris"],
    "blanco": ["negro", "rojo", "verde", "amarillo", "azul"],
}

1 usage
def get_color_combinations(color_name):
    return colores_combinaciones.get(color_name, ["Ninguna combinación sugerida"])

```

Fig. 19. Función para la combinación de colores

2.5.6 Algoritmo de detección de prendas e integración del algoritmo para detección de colores.

La detección de prendas se la realiza usando el modelo entrenado llamado best.pt este modelo permitirá realizar el procesamiento de la imagen, identificar las regiones donde se encuentra la prenda. La detección incluye la información sobre el nombre de la clase, la confianza, precisión y las dimensiones. Una vez identificado la prenda se integra el sistema relacionara cada detección con el color predominante y la combinación. Esta función está en un método POST para utilizarla en el aplicativo al iniciar las detecciones.

```
@detecciones_bp.route(rule='/detectar1', methods=['POST'])
def detectar():
    global ultimo_tiempo_deteccion
    try:
        if 'imagen' not in request.files:
            return jsonify({'message': "No se ha enviado ninguna imagen."}), 400
        archivo_imagen = request.files['imagen']
        npimg = np.frombuffer(archivo_imagen.read(), np.uint8)
        img = cv2.imdecode(npimg, cv2.IMREAD_COLOR)
        if img is None:
            return jsonify({'message': "Error al decodificar la imagen."}), 400
        resultados = modelo(img)
        detecciones = []
        for result in resultados:
            for box in result.boxes:
                x1, y1, x2, y2 = box.xyxy[0].numpy()
                conf = box.conf[0].numpy()
                cls = int(box.cls[0].numpy())
                if conf > 0.5: # Umbral de confianza
                    deteccion = {
                        "clase": modelo.names[cls], "confianza": float(conf), "x": int(x1), "y": int(y1), "width": int(x2 - x1), "height": int(y2 - y1)
                    }

```

Fig. 20. Primera parte del algoritmo de detección de prendas

```

                cls = int(box.cls[0].numpy())
                if conf > 0.5: # Umbral de confianza
                    deteccion = {
                        "clase": modelo.names[cls], "confianza": float(conf), "x": int(x1), "y": int(y1), "width": int(x2 - x1), "height": int(y2 - y1)
                    }
                    detecciones.append(deteccion)
                    dominant_color = get_dominant_color_kmeans(img, bbox=[x1, y1, x2, y2])
                    color_name = get_color_name(dominant_color)
                    color_combinations = get_color_combinations(color_name)
                    deteccion["color_predominante"] = color_name
                    deteccion["color_rgb"] = dominant_color
                    deteccion["combinaciones"] = color_combinations
                    cv2.rectangle(img, (int(x1), int(y1)), (int(x2), int(y2)), (255, 0, 0), 2)
                    cv2.putText(img, text=f'{modelo.names[cls]}: {conf:.2f}', org=(int(x1), int(y1) - 10),
                                cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.5, color=(255, 0, 0), thickness=2)
                    cv2.putText(img, text=f'{color_name}', org=(int(x1), int(y1) + 10),
                                cv2.FONT_HERSHEY_SIMPLEX, fontScale=0.5, color=(255, 0, 0), thickness=2)

# Guardar la imagen con las detecciones y colores
detecciones_path = 'detecciones.jpg'
cv2.imwrite(detecciones_path, img)
return jsonify(detecciones), 200
except Exception as e:
    print(f'Error al procesar la imagen: {str(e)}') # Imprime el error para depurar
    return jsonify({'message': f'Error al procesar la imagen: {str(e)}'}), 500

```

Fig. 21. Segunda parte del algoritmo de detección de prendas

2.5.7 CREACION DEL API PARA LA CONEXIÓN A LA BASE Y SU FUNCIONES PARA LA PAGINA WEB Y APLICATIVO MÓVIL.

El API está dividido en secciones la conexión principal se usa flask de Python para agregar cada una los métodos creados y sus funciones.

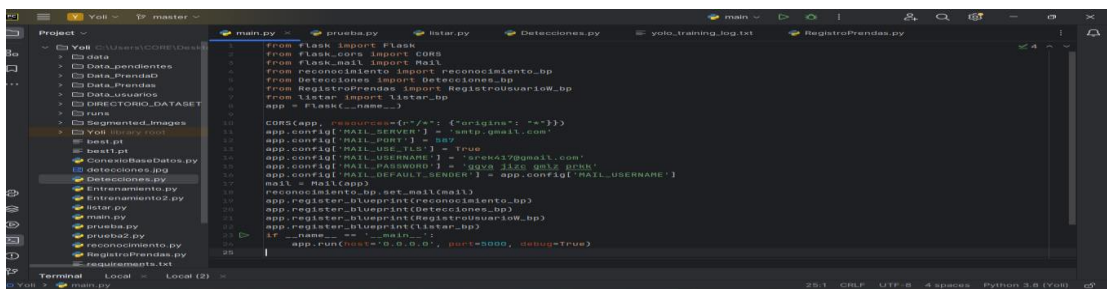
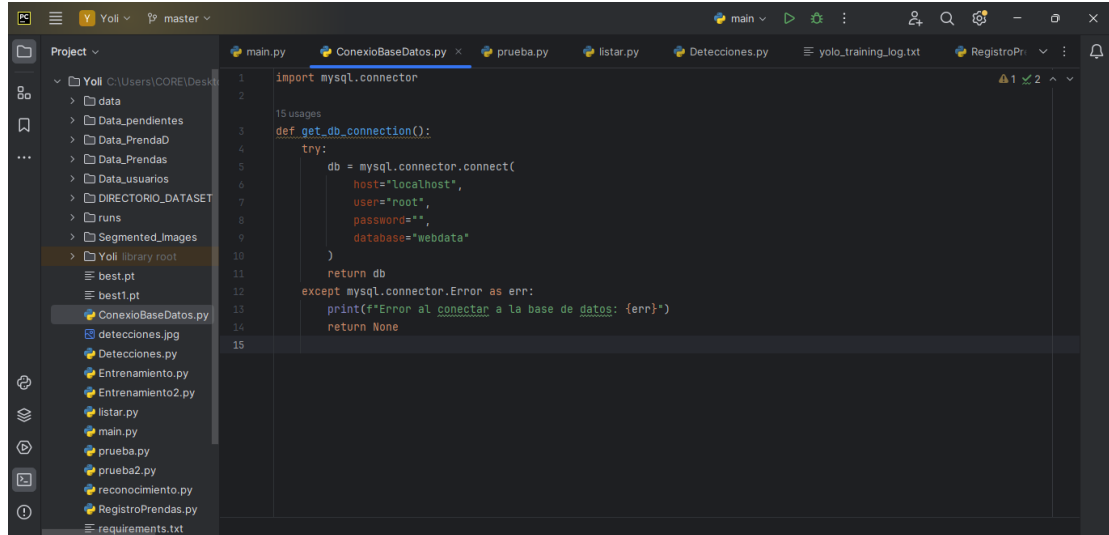


Fig. 22. Conexión del API con cada método

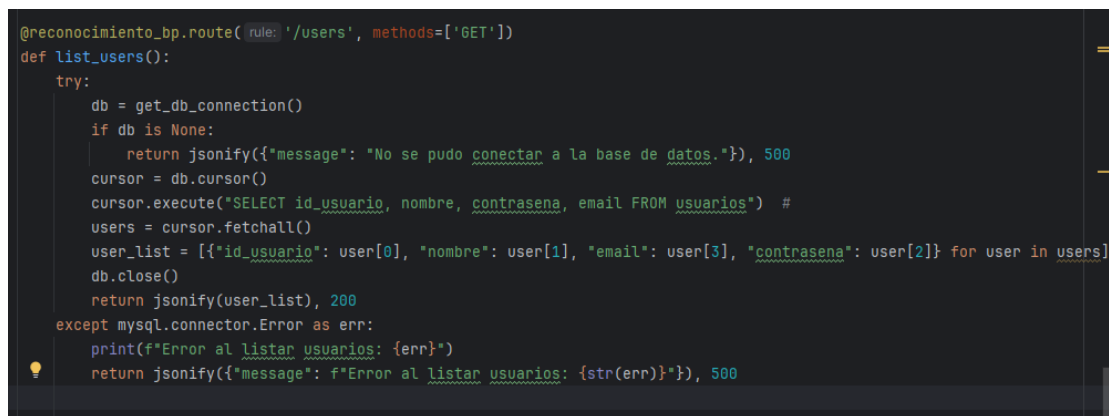
Esta parte del código se visualiza la conexión a la base de datos, para lo cual se debe importar las librerías `mysql.connector`, esto permite que los usuarios realice consultas o registro, estando conectado con la API.



```
1 import mysql.connector
2
3 def get_db_connection():
4     try:
5         db = mysql.connector.connect(
6             host="localhost",
7             user="root",
8             password="",
9             database="webdata"
10        )
11        return db
12    except mysql.connector.Error as err:
13        print(f"Error al conectar a la base de datos: {err}")
14        return None
15
```

Fig. 23. Conexión del API con la base de datos

La siguiente imagen muestra los métodos de iniciar sesión en el sitio web, para esto se permite realizar la consulta a la base de datos para validar y verificar el ingreso de los usuarios, si los datos son correctos ingresa a la interfaz correspondiente caso contrario se puede dirigir a la parte de registro. Para realizar la verificación de los datos registrado los listamos para después comprar con los datos ingresado.



```
@reconocimiento_bp.route(rule='/users', methods=['GET'])
def list_users():
    try:
        db = get_db_connection()
        if db is None:
            return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
        cursor = db.cursor()
        cursor.execute("SELECT id_usuario, nombre, contrasena, email FROM usuarios") #
        users = cursor.fetchall()
        user_list = [{"id_usuario": user[0], "nombre": user[1], "email": user[3], "contrasena": user[2]} for user in users]
        db.close()
        return jsonify(user_list), 200
    except mysql.connector.Error as err:
        print(f"Error al listar usuarios: {err}")
        return jsonify({"message": f"Error al listar usuarios: {str(err)}"}, 500
```

Fig. 24. Código de listar usuarios

Este método es utilizado para ser llamado en la sección de apartado del sistema web creado con codificación angular en un entorno de Visual Studio Code, a continuación, se muestra la imagen donde se ha utilizado el método y como se realiza la comprobación.

```

src > app > paginas > principal > TS principal.page.ts > PrincipalPage > IniciarSeccion
11 export class PrincipalPage implements OnInit {
23   loadUsers() {
24     this.http.get(`${this.serGeneral.getBaseUrl()}/users`)
25       .subscribe((response: any) => {
26         this.usuarios = response;
27         console.log("Lista de usuarios cargada:", this.usuarios);
28       }, (error) => {
29         console.error("Error al cargar usuarios:", error);
30       });
31   }
32   IniciarSeccion(event: Event) {
33     event.preventDefault();
34     const user = this.usuarios.find(u => u.email === this.nombre);
35     console.log("consulta", user);
36     if (user) {
37       console.log("Usuario encontrado:", user);
38       if (user.contrasena === this.contrasena) {
39         console.log("Inicio de sesión exitoso.", user);
40         localStorage.setItem('id_usuario', user.id_usuario);
41         this.router.navigate(['/home']);
42       } else {
43         console.warn("Contraseña incorrecta.", user);
44         alert("Contraseña incorrecta. Intenta nuevamente.");
45       }
46     } else {
47       console.warn("Usuario no registrado.");
48       alert("Usuario no registrado. Por favor, regístrate.");
49     }
50   }
51 }

```

Fig. 25. Código de Inicio de sesión en Visual Studio Code

El siguiente código permite registra los usuarios para esto se involucran varias funciones las cuales permite verificar si los correos ingresados son válidos, esto se logra enviando código de verificación a los correos ingresados para el proceso de registro si lo código son válidos registra caso contrario no permite continuar el proceso.

```

@reconocimiento_bp.route('/register', methods=['POST'])
def register_user():
    data = request.json
    print("Datos recibidos:", data)
    if 'nombre' not in data or 'email' not in data or 'contrasena' not in data:
        return jsonify({"message": "Faltan campos requeridos."}), 400
    nombre = data['nombre']
    email = data['email']
    contrasena = data['contrasena']
    try:
        db = get_db_connection()
        if db is None:
            return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
        cursor = db.cursor()
        cursor.execute("SELECT email FROM usuarios WHERE LOWER(email) = LOWER(%)", (email,))
        existing_user = cursor.fetchone()
        if existing_user and existing_user[0] == email:
            return jsonify({"message": "Correo electrónico ya registrado."}), 400
        encrypted_password = encrypt_password(contrasena)
        verification_code = str(random.randint(100000, 999999))
        temp_users[email] = {'nombre': nombre, 'contrasena': encrypted_password, 'verification_code': verification_code}
        send_verification_code(email, verification_code)
        return jsonify({"message": "Se ha enviado un código de verificación."}), 201
    except mysql.connector.Error as err:
        print(f"Error al registrar usuario: {err}")
    register_user()

```

Fig. 26. Método POST para el registro de usuarios

```

65 async registerUser(event: Event) {
66     event.preventDefault();
67
68     const userData = {
69         nombre: this.nombre,
70         email: this.email,
71         contraseña: this.contrasena
72     };
73     console.log('Datos a enviar:', userData);
74     const url = `${this.serGeneral.getBaseUrl()}/register`;
75
76     this.http.post(url, userData).subscribe(
77         async (response: any) => {
78             await this.presentVerificationPrompt();
79         },
80         async (error) => {
81             const errorMessage = error.error?.message || 'Ocurrió un error durante el registro.';
82             await this.presentAlert('Error', errorMessage);
83             console.error(error);
84         }
85     );
86 }
87

```

Fig. 27. Código de Registro en Visual Studio Code

```

@reconocimiento_bp.route(rule: '/verify', methods=['POST'])
def verify_code():
    data = request.json
    print("Datos de verificación:", data)
    if 'email' not in data or 'codigo' not in data:
        return jsonify({"message": "Faltan campos requeridos."}), 400
    email = data['email']
    codigo = data['codigo']
    try:
        if email in temp_users and temp_users[email]['verification_code'] == codigo:
            db = get_db_connection()
            if db is None:
                return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
            cursor = db.cursor()
            sql = "INSERT INTO usuarios (nombre, email, contraseña) VALUES (%s, %s, %s)"
            values = (temp_users[email]['nombre'], email, temp_users[email]['contrasena'])
            cursor.execute(sql, values)
            db.commit()
            del temp_users[email]
            db.close()
            return jsonify({"message": "Usuario registrado exitosamente."}), 201
        else:
            return jsonify({"message": "Código de verificación incorrecto."}), 400

```

Fig. 28. Método POST para verificar código de validación


```

async verifyCode() {
  const verifyUrl = `${this.serGeneral.getBaseUrl()}/verify`;
  const verificationData = {
    email: this.email,
    codigo: this.codigoVerificacion
  };

  this.http.post(verifyUrl, verificationData).subscribe(
    async (response: any) => {
      await this.presentAlert('Éxito', 'Código de verificación correcto. Usuario registrado exitosamente.');
```

Fig. 29. Código de verificación en Visual Studio Code

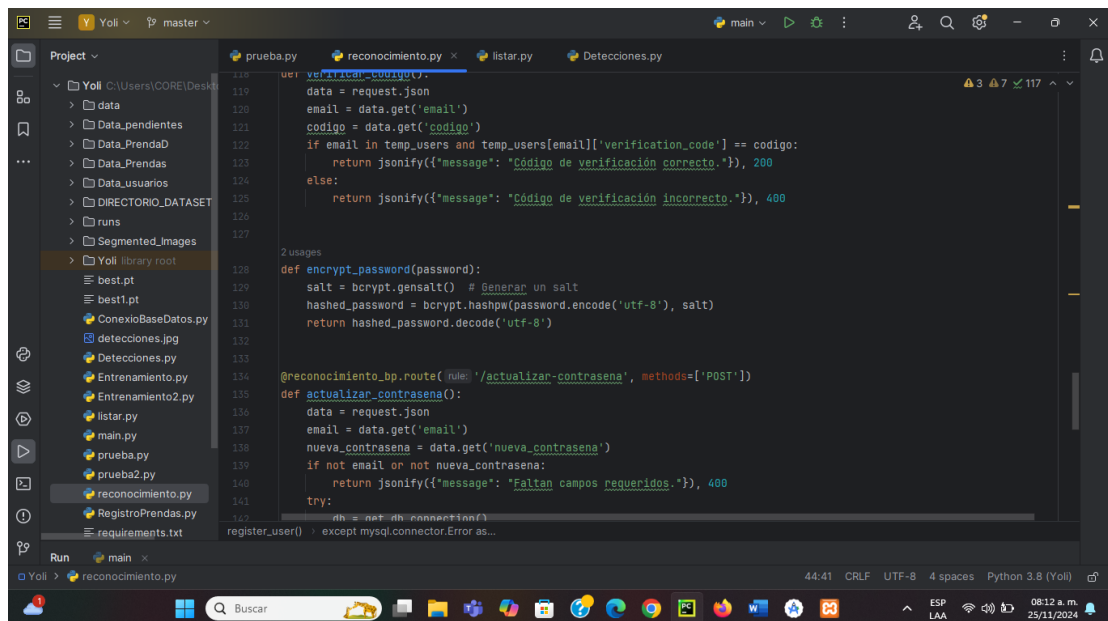


Fig. 30. Función de encriptar contraseña.

En el sistema web hay opciones para actualizar contraseña el cual usa los métodos de validación de código a los correos como método de seguridad para poder hacer cambio si es que al usuario se le olvido sus credenciales además usa la función de encriptado de contraseña para poder enviar a la base de datos para estos proceso se usa el método POST actualizar-contraseña.

```

@reconocimiento_bp.route(rule: '/solicitar-codigo', methods=['POST'])
def solicitar_codigo():
    data = request.json
    email = data.get('email')
    if not email:
        return jsonify({"message": "Falta el campo de correo electrónico."}), 400
    try:
        db = get_db_connection()
        if db is None:
            return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
        cursor = db.cursor()
        cursor.execute("SELECT email FROM usuarios WHERE LOWER(email) = LOWER(%s)", (email,))
        existing_user = cursor.fetchone()
        if not existing_user:
            return jsonify({"message": "Correo electrónico no registrado."}), 400
            verification_code = str(random.randint(a=100000, b=999999))
            temp_users[email] = {'verification_code': verification_code}
            send_verification_code(email, verification_code)
            db.close()
            return jsonify({"message": "Se ha enviado un código de verificación."}), 200
    except mysql.connector.Error as err:
        print(f"Error al verificar el correo electrónico: {err}")
        return jsonify({"message": f"Error al verificar el correo electrónico: {str(err)}"}), 500

```

Fig. 31. Método POST para solicitar código de validación

```

2 usages
def send_verification_code(email, code):
    try:
        msg = Message(subject="Código de Verificación",
                      sender="srek417@gmail.com",
                      recipients=[email])
        msg.body = f"Tu código de verificación es: {code}"
        mail.send(msg)
        print(f"Código de verificación enviado a {email}")
    except Exception as e:
        print(f"Error al enviar correo: {e}")
        raise e

```

Fig. 32. Función para enviar código de validación al correo

```

async solicitarCodigo(event: Event) {
  const url = `${this.serGeneral.getBaseUrl()}/solicitar-codigo`;
  const data = { email: this.email };

  this.http.post(url, data).subscribe(
    async () => {
      await this.presentAlert('Éxito', 'Se ha enviado un código de verificación a tu correo.');
```

Fig. 33. Código de solicitar y enviar código al correo en Visual Studio Code

```

@reconocimiento_bp.route(rule: '/verificar-codigo', methods=['POST'])
def verificar_codigo():
  data = request.json
  email = data.get('email')
  codigo = data.get('codigo')
  if email in temp_users and temp_users[email]['verification_code'] == codigo:
    return jsonify({"message": "Código de verificación correcto."}), 200
  else:
    return jsonify({"message": "Código de verificación incorrecto."}), 400
```

Fig. 34. Método de verificación de código de validación

```

async verifyCode() {
  const verifyUrl = `${this.serGeneral.getBaseUrl()}/verificar-codigo`;
  const verificationData = {
    email: this.email,
    codigo: this.codigoVerificacion,
  };

  this.http.post(verifyUrl, verificationData).subscribe(
    async () => {
      await this.presentAlert('Éxito', 'Código de verificación correcto. Ingrese tu nueva contraseña.');
```

Fig. 35. Método de verificación de código en Visual Studio Code

```

@reconocimiento_bp.route(rule: '/actualizar-contrasena', methods=['POST'])
def actualizar_contrasena():
    data = request.json
    email = data.get('email')
    nueva_contrasena = data.get('nueva_contrasena')
    if not email or not nueva_contrasena:
        return jsonify({"message": "Faltan campos requeridos."}), 400
    try:
        db = get_db_connection()
        if db is None:
            return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
        cursor = db.cursor()
        encrypted_password = encrypt_password(nueva_contrasena)
        sql = "UPDATE usuarios SET contrasena = %s WHERE LOWER(email) = LOWER(%s)"
        cursor.execute(sql, (encrypted_password, email))
        db.commit()
        if email in temp_users:
            del temp_users[email]
        db.close()
        return jsonify({"message": "Contraseña actualizada exitosamente."}), 200
    except mysql.connector.Error as err:
        print(f"Error al actualizar la contraseña: {err}")
        return jsonify({"message": f"Error al actualizar la contraseña: {str(err)}"}),

```

Fig. 36. Método de POST para la actualización de contraseña

```

async updatePassword() {
    if (this.nuevaContrasena !== this.confirmarContrasena) {
        await this.presentAlert('Error', 'Las contraseñas no coinciden. Por favor, intentalo de nuevo.');
```

```

        return;
    }
    const updateUrl = `${this.serGeneral.getBaseUrl()}/actualizar-contrasena`;
    const data = {
        email: this.email,
        nueva_contrasena: this.nuevaContrasena,
    };

    this.http.post(updateUrl, data).subscribe(
        async () => {
            await this.presentAlert('Éxito', 'Contraseña actualizada exitosamente.');
```

```

            this.email = '';
            this.codigoVerificacion = '';
            this.nuevaContrasena = '';
            this.confirmarContrasena = '';
            this.codigoVerificado = false;
            setTimeout(() => {
                location.reload();
            }, 1000);
        },
        async (error) => {
            const errorMessage = error.error?.message || 'Ocurrió un error al actualizar la contraseña.';
            await this.presentAlert('Error', errorMessage);
        }
    );
}

```

Fig. 37. Código de actualización de contraseña en Visual Studio Code

Para todo esto proceso se requiere de librerías importada en Python para brinda un correcto funcionamiento.

```
from flask import Blueprint, request, jsonify
from flask_cors import CORS
import mysql.connector
from flask_mail import Message
import random
import mysql.connector
from ConexionBaseDatos import get_db_connection
reconocimiento_bp = Blueprint(name: 'reconocimiento', __name__)
CORS(reconocimiento_bp)
temp_users = {}
mail = None

2 usages (1 dynamic)
def set_mail(mail_instance):
    global mail
    mail = mail_instance

reconocimiento_bp.set_mail = set_mail
```

Fig. 38. Librerías

Como el sistema debe tener un aprendizaje automático se requiere de un método para registrar las prendas con sus respectivas clases. Para este procedimiento se realizó un método POST de registro de prendas. La función no solo permite registro, sino que además usa el modelo entrenado anteriormente para realizar segmentaciones es decir que si las imágenes ingresadas serán procesadas para extraer coordenada que se almacenara en un formato JSON. Este algoritmo permite darle un nombre único a la imagen y al JSON, esta información se enviará a la base de datos y a una carpeta general llamada Data_usuarios, si por una razón se sube imágenes que no tenga que ver con patrones de prendas el modelo lo detectará y registrar como pendiente estos como método de seguridad para que el dataset del modelo entrenado solo tenga datos de vestimentas para usarlo en un reentrenamiento con las nuevas informaciones así no se verá afectado su aprendizaje.

```

25
26
27 @RegistroUsuarioW_bp.route('/registerPrendas', methods=['POST'])
28 def registrar_prendas():
29     image_file = request.files.get('image')
30     id_usuario = request.form.get('usuario')
31     nombre_prenda = request.form.get('etiqueta').lower()
32
33     if not id_usuario or not image_file:
34         return jsonify({"message": "Faltan campos requeridos."}), 400
35     img = Image.open(image_file).convert('RGB')
36     img_np = np.array(img)
37     img_cv2 = cv2.cvtColor(img_np, cv2.COLOR_RGB2BGR)
38     results = modelo(img_cv2)
39     json_data = {
40         "version": "5.4.1",
41         "flags": {},
42         "shapes": [],
43         "imagePath": image_file.filename,
44         "imageData": None
45     }
46     coordenadas_unicas = []
47     mask_image = img_cv2.copy()
48     deteccion_realizada = False
49     for result in results:
50         # Verificar si result.masks no es None
51         if result.masks is not None:

```

Fig. 39. Primera parte del código de registro

```

49     for result in results:
50         if result.masks is not None:
51             masks = result.masks.xy
52             labels = result.names
53             predicted_classes = result.boxes.cls
54             for idx, mask in enumerate(masks):
55                 class_id = int(predicted_classes[idx])
56                 label = labels[class_id]
57                 mask_points = mask.tolist()
58                 if mask_points not in coordenadas_unicas:
59                     coordenadas_unicas.append(mask_points)
60                     json_data["shapes"].append({
61                         "label": nombre_prenda,
62                         "points": mask_points,
63                         "group_id": None,
64                         "description": "",
65                         "shape_type": "polygon",
66                         "flags": {},
67                         "mask": None
68                     })
69                 mask_points = np.array(mask_points, dtype=np.int32)
70                 cv2.polylines(mask_image, [pts[mask_points]], isClosed=True, color=(0, 255, 0), thickness=3)
71
72             deteccion_realizada = True # Se detectaron objetos
73         else:
74             print("No se detectaron máscaras en esta imagen.")
75
76     if not deteccion_realizada:
77         registrar_prendas()

```

Fig. 40. Segunda parte del código de registro

```

73         print("No se detectaron máscaras en esta imagen.")
74     else:
75         print("No se detectaron objetos en la imagen, se marcará como pendiente.")
76
77     try:
78         directorio_pendientes = r"C:\Users\CORE\Desktop\APP\Yoli\Data_pendientes"
79         if not os.path.exists(directorio_pendientes):
80             os.makedirs(directorio_pendientes)
81             ruta_completa_usuario = os.path.join(directorio_pendientes, f"pendiente_{uuid.uuid4()}.jpg")
82             img.save(ruta_completa_usuario)
83             db = get_db_connection()
84             if db is None:
85                 return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
86             cursor = db.cursor()
87             sql_pendiente = """
88             INSERT INTO pendientes (id_usuario, nombre_prenda, ruta_imagen, clase)
89             VALUES (%s, %s, %s, %s)
90             """
91             cursor.execute(sql_pendiente, (id_usuario, nombre_prenda, ruta_completa_usuario, nombre_prenda))
92             db.commit()
93             db.close() # Cerrar la conexión a DB
94             return jsonify({"message": "No se detectaron prendas en la imagen. Prenda marcada como pendiente."}), 201
95     except Exception as e:

```

Fig. 41. Tercera parte del código de registro

```

104 db.close() # Cerrar la conexión a DB
105 return jsonify({"message": "No se detectaron prendas en la imagen. Prenda marcada como pendiente."}), 201
106
107 except Exception as e:
108     print(f"Error al registrar la prenda como pendiente: {e}")
109     return jsonify({"message": f"Error al registrar la prenda como pendiente: {e}"}), 500
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Fig. 42. Cuarta parte del código de registro

```

112 img.save(ruta_completa_usuario)
113 cv2.imwrite(ruta_segmentada, mask_image)
114 with open(ruta_json_usuario, 'w') as f:
115     json.dump(json_data, f, indent=4)
116 cursor.execute("SELECT id_clase FROM clases_prendas WHERE nombre_clase = %s", (nombre_prenda,))
117 result = cursor.fetchone()
118 if not result:
119     cursor.execute("INSERT INTO clases_prendas (nombre_clase) VALUES (%s)", (nombre_prenda,))
120     db.commit()
121     Nclase = cursor.lastrowid
122 else:
123     Nclase = result[0]
124 sql_prenda = """
125 INSERT INTO prendas_subidas (id_usuario, nombre_prenda, ruta_imagen, nombre_json, ruta_json, id_clase)
126 VALUES (%s, %s, %s, %s, %s, %s)
127 """
128 values = (id_usuario, nombre_prenda, ruta_completa_usuario, nombre_json, ruta_json_usuario, Nclase)
129 cursor.execute(sql_prenda, values)
130 db.commit()
131 db.close()
132 return jsonify({"message": "Prenda registrada exitosamente"}), 201
133 except Exception as e:
134     print(f"Error al registrar prendas: {e}")
135     if db:
136         db.close() # Asegurarnos de cerrar la conexión a la base de datos en caso de error
137     return jsonify({"message": f"Error al registrar prendas: {e}"}), 500
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Fig. 43. Cuarta parte del código de registro

```

async REGISTRAR(event: Event) {
  event.preventDefault();
  const imageBlob = this.convertirImagenABase64Blob(this.selectedImages);
  const formData = new FormData();
  formData.append('etique', this.ETIQUETA);
  formData.append('Usuario', this.id_usuario);
  formData.append('image', imageBlob, 'imagen.jpg');
  if (this.coordenadas) {
    formData.append('coordenadas', JSON.stringify(this.coordenadas));
  }
  this.http.post(`${this.serGeneral.getBaseUrl()}/registerPrendas`, formData).subscribe(
    async (response: any) => {
      if(response.message.includes('pendiente')){
        await this.presentAlert('Registro Exitoso', 'Prenda pendiente de registrar.');
```

Fig. 44.. Código de registro de prendas en Visual Studio Code

Otro método que hay y se usa en el sitio web es la opción actualizar dataset, este función usa el método GET este método permite hace un balance de dato aumentando es decir que los usuario suben sus prendas el método realizar un aumento por cada clase registrada alrededor de 50 imágenes esto permite que al dataset se envíen más datos ajustado el brillo, contraste y saturación favoreciendo al aprendizaje del modelo, terminado el proceso toda la imágenes se depositara en el dataset principal para permitir el reentrenamiento del modelo.

```

@tostar_bp.route(rule='/BalancearDataset/<int:id_usuario>', methods=['GET'])
def balancear_dataset(id_usuario):
  print(f"listando las prendas del usuario con id: {id_usuario} para balancear el dataset.")
  if not id_usuario:
    return jsonify({"message": "No se proporcionó un identificador de usuario."}), 400
  try:
    db = get_db_connection()
    if db is None:
      return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
    cursor = db.cursor()
    cursor.execute(
      "SELECT nombre_prenda, ruta_imagen, ruta_json FROM prendas_subidas WHERE id_usuario = %s", (id_usuario,)
    )
    prendas = cursor.fetchall()
    print(f"Resultados obtenidos: {prendas}")
    rutas_imagenes_por_clase = {}
    for nombre_prenda, ruta_imagen, ruta_json in prendas:
      if nombre_prenda not in rutas_imagenes_por_clase:
        rutas_imagenes_por_clase[nombre_prenda] = []
        rutas_imagenes_por_clase[nombre_prenda].append((ruta_imagen, ruta_json))
      imagen_original_destino = os.path.join('C:\\Users\\CORE\\Desktop\\APP\\Yoli\\Data_Prenda\\dataset',
        os.path.basename(ruta_imagen))
      if not os.path.exists(imagen_original_destino):
        shutil.copy(ruta_imagen, imagen_original_destino)
        print(f"Imagen original guardada en {imagen_original_destino}")

```

Fig. 45. Primera parte del código de aumento de datos


```

        print(f"Imagen original guardada en {imagen_original_destino}")
    else:
        print(f"La imagen {imagen_original_destino} ya existe. No se copia.")
    json_destino = os.path.join('C:\\Users\\CORE\\Desktop\\APP\\Yoli\\Data_PrendaD\\dataset',
                                os.path.basename(ruta_json))
    if not os.path.exists(json_destino):
        shutil.copy(ruta_json, json_destino)
        print(f"Archivo JSON guardado en {json_destino}")
    else:
        print(f"El archivo JSON {json_destino} ya existe. No se copia.")
for nombre_prenda, rutas_clase in rutas_imagenes_por_clase.items():
    imagenes_a_generar = 50 - len(rutas_clase)
    if imagenes_a_generar > 0:
        for _ in range(imagenes_a_generar):
            ruta_imagen, ruta_json = random.choice(rutas_clase)
            img = Image.open(ruta_imagen)
            rnd_brightness = random.uniform(a: 0.8, b: 1.2)
            rnd_contrast = random.uniform(a: 0.8, b: 1.2)
            rnd_saturation = random.uniform(a: 0.8, b: 1.2)
            rnd_hue = random.uniform(a: 0, b: 0.5)
            color_trans = transforms.ColorJitter(
                brightness=rnd_brightness,
                contrast=rnd_contrast,
                saturation=rnd_saturation,
                hue=rnd_hue
            )

```

Fig. 46. Segunda parte del código de aumento de datos

```

            hue=rnd_hue
        )
        augmented_img = color_trans(img)
        aug_image_id = f"{uuid.uuid4()}"
        aug_image_filename = f"{aug_image_id}.jpg"
        aug_json_filename = f"{aug_image_id}.json"
        aug_image_path = os.path.join('C:\\Users\\CORE\\Desktop\\APP\\Yoli\\Data_PrendaD\\dataset',
                                        aug_image_filename)
        aug_json_path = os.path.join('C:\\Users\\CORE\\Desktop\\APP\\Yoli\\Data_PrendaD\\dataset',
                                        aug_json_filename)

        if not os.path.exists(aug_image_path):
            augmented_img.save(aug_image_path)
            print(f"Imagen aumentada guardada en {aug_image_path}")
        else:
            print(f"La imagen aumentada {aug_image_path} ya existe. No se guarda.")

        if not os.path.exists(aug_json_path):
            shutil.copy(ruta_json, aug_json_path)
            print(f"Etiqueta en {aug_json_path} guardada.")
        else:
            print(f"El archivo JSON aumentada {aug_json_path} ya existe. No se copia.")

    db.close()
    print("Aumento de datos completado y guardado en el dataset.")
    json_dir = 'C:\\Users\\CORE\\Desktop\\APP\\Yoli\\Data_PrendaD\\dataset'
    threading.Thread(target=ejecutar_labelme2yolo, args=(json_dir,)).start()

```

Fig. 47. Tercera parte del código de aumento de datos

```

        augmented_img.save(aug_image_path)
        print(f"Imagen aumentada guardada en {aug_image_path}")
    else:
        print(f"La imagen aumentada {aug_image_path} ya existe. No se guarda.")

    if not os.path.exists(aug_json_path):
        shutil.copy(ruta_json, aug_json_path)
        print(f"Etiqueta en {aug_json_path} guardada.")
    else:
        print(f"El archivo JSON aumentada {aug_json_path} ya existe. No se copia.")

    db.close()
    print("Aumento de datos completado y guardado en el dataset.")
    json_dir = 'C:\\Users\\CORE\\Desktop\\APP\\Yoli\\Data_PrendaD\\dataset'
    threading.Thread(target=ejecutar_labelme2yolo, args=(json_dir,)).start()
    return jsonify({"message": "Dataset balanceado exitosamente."}), 200
except mysql.connector.Error as err:
    print(f"Error al acceder a la base de datos: {err}")
    return jsonify({"message": f"Error al acceder a la base de datos: {err}"}), 500

```

Fig. 48. Cuarta parte del código de aumento de datos

```

async CREAM_DATASET() {
    if (this.id_usuario) {
        const loading = await this.loadingController.create({
            message: 'Por favor, espere mientras se crea el dataset. Esto puede tardar un minuto. No cierre la sesión.'
        });
        await loading.present();

        try {
            const response: any = await this.http.get(`${this.serGeneral.getBaseUrl()}/BalancearDataset/${this.id_usuario}`);
            if (response && response.message) {
                console.log(response.message);
                await this.presentAlert('Éxito', 'El dataset se ha creado y balanceado exitosamente.');
```

Fig. 49. Código de Actualización del dataset en Visual Studio Code

Internamente abran dos funciones que se ejecutaran en segundo plano cada vez que se actualice el dataset se ejecutaran para el aprendizaje automático del modelo. Primero se realiza un conteo, si las clases cuenta con 500 imágenes o más se la usan para reentrenar el modelo si hay nuevas clases y tiene pocas imágenes esta no se tomará en cuenta hasta que se recopilen la cantidad establecida, esto permite un equilibrio en el dataset para que no se vea afectado su aprendizaje ya que se

recomienda que cada clase tenga 500 imágenes. Terminando el conteo se ejecuta la función para darle el formato adecuado para después entrenar.

```
def ejecutar_labelme2yolo(json_dir):
    try:
        clase_imagenes = {}
        for root, dirs, files in os.walk(json_dir):
            for file in files:
                if file.endswith(".json"):
                    json_path = os.path.join(root, file)
                    with open(json_path, 'r') as f:
                        data = json.load(f)
                        for shape in data['shapes']:
                            clase = shape['label']
                            if clase not in clase_imagenes:
                                clase_imagenes[clase] = 0
                            clase_imagenes[clase] += 1
        clases_validas = {clase: count for clase, count in clase_imagenes.items() if count >= 500}
        print(f"Clases válidas para convertir a YOLO: {clases_validas}")
        if clases_validas:
            command = f'Labelme2yolo --json_dir {json_dir}'
            subprocess.run(command, shell=True, check=True)
            yolo_dataset_dir = os.path.join(json_dir, 'YOLODataset')
            if os.path.exists(yolo_dataset_dir):
                entrenar_yolo(yolo_dataset_dir)
            else:
                print("Error: La carpeta de datos YOLO no se ha generado correctamente.")
        else:
            print("No se encontraron clases válidas para la conversión.")
```

Fig. 50. Método de conteo y conversión de datos

Otro método que cuenta el API son lo método de listado en esta se muestra cada imagen que es el usuario registro con su coordenada y también la que esta como pendiente estos datos pendientes se revisasen directamente desde el entorno del administrado para verifica si son o no válidas para el aprendizaje de modelo.

```
@listar_bp.route(rule='/listarPrendas/<int:id_usuario>', methods=['GET'])
def listar_prendas_por_usuario(id_usuario):
    try:
        db = get_db_connection()
        if db is None:
            return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
        cursor = db.cursor()
        cursor.execute(
            "SELECT id_prenda, id_usuario, nombre_prenda, ruta_imagen, ruta_json FROM prendas_subidas WHERE id_usuario = %s"
            (id_usuario,)
        )
        resultados = cursor.fetchall()
        prendas = []
        for row in resultados:
            prenda = {
                "id_prenda": row[0], "id_usuario": row[1], "nombre_prenda": row[2],
                "ruta_imagen": row[3], "ruta_json": row[4]
            }
            prendas.append(prenda)
        db.close()
        return jsonify(prendas), 200
    except Exception as e:
        print(f"Error al listar prendas: {e}")
        return jsonify({"message": f"Error al listar prendas: {str(e)}"}), 500
```

Fig. 51. Método GET para lista prendas

```

async CARGAR_IMAGENES_POR_USUARIO() {
  if (this.id_usuario) {
    try {
      const response: any = await this.http.get(`${this.serGeneral.getBaseUrl()}/listarPrendas/${this.id_usuario}`);
      this.prendas = await Promise.all(response.map(async (prenda: Prenda) => {
        if (!prenda.ruta_json) {
          console.error('ruta_json no está definida para:', prenda);
          return null;
        }
        console.log('Fetching data for:', prenda.ruta_json);
        try {
          const datosResponse: any = await this.http.get(`${this.serGeneral.getBaseUrl()}/datosPrenda/${prenda.ruta_json}`);
          console.log('Datos recibidos:', datosResponse);
          return {
            ...prenda,
            datos: datosResponse
          };
        } catch (datosError) {
          console.error('Error al cargar datos de prenda:', datosError);
          return {
            ...prenda,
            datos: null
          };
        }
      }));
      this.prendas = this.prendas.filter(prenda => prenda !== null);
      console.log('Prendas cargadas para el usuario:', this.prendas);
    } catch (error) {}
    const err = error as HttpResponse;
    console.error('Error al cargar prendas por usuario:', err);
    if (err.status === 404) {

```

Fig. 52. Código de listar prendas en Visual Studio Code

```

@listar_bp.route(rule: '/listarPrendasPendientes/<int:id_usuario>', methods=['GET'])
def listar_prendas_pendientes(id_usuario):
    print(f"listando prendas pendientes para el usuario: {id_usuario}")
    try:
        db = get_db_connection()
        if db is None:
            return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
        cursor = db.cursor()
        cursor.execute(
            "SELECT id_pendiente, id_usuario, nombre_prenda, ruta_imagen, clase FROM pendientes WHERE id_usuario = %s",
            (id_usuario,)
        )
        resultados = cursor.fetchall()
        prendas_pendientes = []
        for row in resultados:
            prenda = {
                "id_pendiente": row[0], "id_usuario": row[1], "nombre_prenda": row[2], "ruta_imagen": row[3],
                "clase": row[4]
            }
            prendas_pendientes.append(prenda)
        db.close()
        return jsonify(prendas_pendientes), 200
    except Exception as e:
        print(f"Error al listar prendas pendientes para el usuario {id_usuario}: {e}")
        return jsonify({"message": f"Error al listar prendas pendientes: {str(e)}"}), 500

```

Fig. 53. Método GET para lista prendas pendientes

```

async CARGAR_PRADNAS_PENDIENTES() {
  try {
    const response: any = await this.http.get(`${this.serGeneral.getBaseUrl()}/listarPrendasPendientes/${this.id_us
    this.prendasPendientes = response;
  } catch (error) {
    console.error('Error al cargar prendas pendientes:', error);
  }
}

```

Fig. 54. Código de listar prendas pendientes en Visual Studio Code

Otro de los servicios que cuenta el API es el método POST registro de dispositivo este método permite iniciar sesión en el aplicativo móvil el cual extraerá los datos de celular los cuales son Android ID que es el identificador que tiene cada dispositivo, además del modelo y marca cuenta con un contador que permite saber el uso del aplicativo. Porque se obtiene esta información para el monitoreo de la aplicación evitando así que el usuario tenga que usar correo y contraseña ya que esto permite una interacción sencilla para el uso de sistema móvil.

```

def registrar_dispositivo():
    data = request.get_json()
    marca = data.get('marca')
    modelo = data.get('modelo')
    serie = data.get('identificadorUnico')
    if not marca or not modelo or not serie:
        return jsonify({"message": "Faltan campos requeridos."}), 400
    db = get_db_connection()
    if db is None:
        return jsonify({"message": "No se pudo conectar a la base de datos."}), 500
    cursor = None
    try:
        cursor = db.cursor()
        cursor.execute("SELECT id, contador FROM dispositivos WHERE marca = %s AND modelo = %s AND serie = %s", (marca, modelo, serie))
        dispositivo = cursor.fetchone()
        if dispositivo:
            nuevo_contador = dispositivo[1] + 1
            cursor.execute("UPDATE dispositivos SET contador = %s WHERE id = %s", (nuevo_contador, dispositivo[0]))
        else:
            cursor.execute("INSERT INTO dispositivos (marca, modelo, serie, contador) VALUES (%s, %s, %s, %s)", (marca, modelo, serie, nuevo_contador))
        db.commit()
        return jsonify({"message": "Dispositivo registrado correctamente."}), 201
    except mysql.connector.Error as e:
        print(f"Error al registrar el dispositivo: {str(e)}")
        return jsonify({"message": "Error al registrar el dispositivo."}), 500

```

Fig. 55. Método POST para registra el dispositivo

En el siguiente apartado se muestra la codificación en flutter donde se utiliza la URL que tiene el API para el correcto funcionamiento del aplicativo móvil para el registro de dispositivo que permite iniciar sesión y da paso a la interfaz principal para hacer detecciones en tiempo real.

```

Future<void> registrarDispositivo(BuildContext context) async {
  DeviceInfoPlugin deviceInfo = DeviceInfoPlugin();
  String marca = "Desconocido";
  String modelo = "Desconocido";
  String identificadorUnico = "Desconocido";
  try {
    if (defaultTargetPlatform == TargetPlatform.android) {
      AndroidDeviceInfo androidInfo = await deviceInfo.androidInfo;
      marca = androidInfo.manufacturer ?? "Desconocido";
      modelo = androidInfo.model ?? "Desconocido";
      final platform = LocalPlatform();
      if (platform.isAndroid) {
        identificadorUnico = androidInfo.id ?? "Desconocido";
      } else if (defaultTargetPlatform == TargetPlatform.iOS) {
        IosDeviceInfo iosInfo = await deviceInfo.iosInfo;
        marca = iosInfo.name ?? "Desconocido";
        modelo = iosInfo.model ?? "Desconocido";
        identificadorUnico = iosInfo.identifierForVendor ?? "Desconocido";
      } final body = jsonEncode({
        "marca": marca,
        "modelo": modelo,
        "identificadorUnico": identificadorUnico,
      }); final response = await http.post(
        Uri.parse('http://192.168.0.10:5000/registrar_dispositivo'),
        headers: {'Content-Type': 'application/json'},
        body: body,
      );
    }
  } catch (e) {
    print(e);
  }
}

```

Fig. 56. Método para registra el dispositivo en Flutter

```

Future<void> _sendImageToServer(XFile imageFile) async {
  if (_ultimoTiempoEnvio != null && DateTime.now().difference(_ultimoTiempoEnvio!) < _intervaloEspera) {
    print("Por favor, espera antes de realizar otra detección.");
    return;
  }
  var request = http.MultipartRequest('POST', Uri.parse('http://192.168.0.10:5000/detectar1'));
  request.files.add(await http.MultipartFile.fromPath('imagen', imageFile.path));
  var response = await request.send();
  if (response.statusCode == 200) {
    var responseData = await http.Response.fromStream(response);
    var detections = jsonDecode(responseData.body);
    setState(() {
      _detections = ListMap<String, dynamic>.from(detections);
    });
    if (_detections.isNotEmpty) {
      var detection = _detections.first;
      String textToSpeak = _getDetectionResponse(detection);
      _detectionText = textToSpeak;
      _speak(textToSpeak);
    }
    _ultimoTiempoEnvio = DateTime.now();
  } else {
    print("error en el proceso.");
  }
}

```

Fig. 57. Método para realizar detecciones usando la interfaz flutter

El entorno configurado permite realizar detecciones en tiempo real desde el celular para el reconocimiento de prendas con sus colores y combinación. Para lograr todo el funcionamiento se le debe dar permiso en el archivo ANDROIDMANIFEST.XML e importar dependencia en el archivo PUBSPEC.YAML a continuación se visualiza la gráfica con la información de esto archivos.

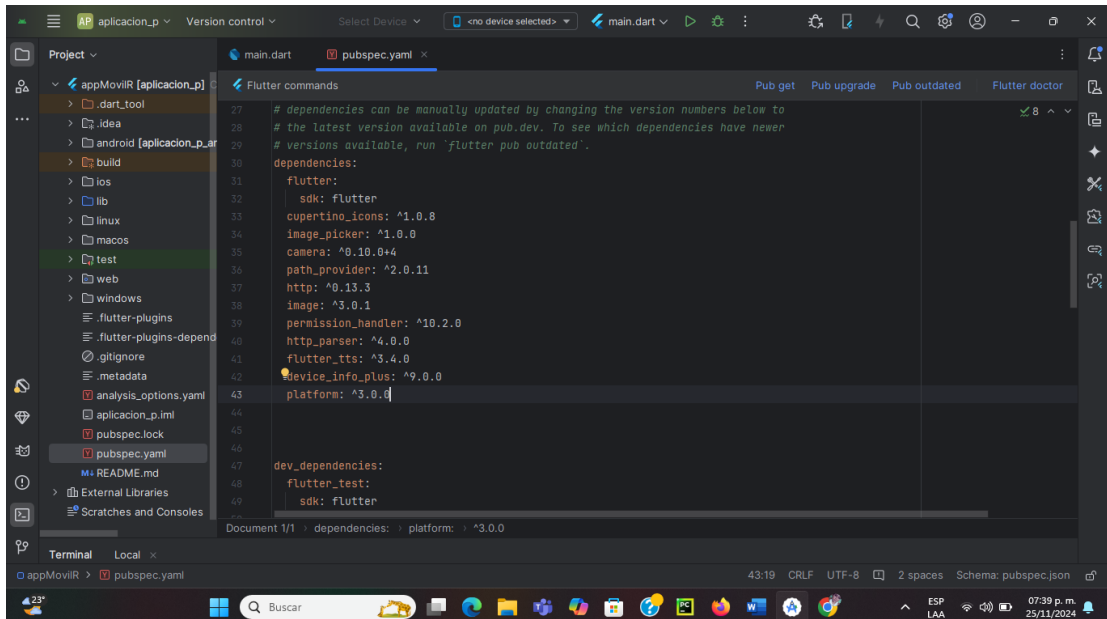


Fig. 58. Dependencias del archivo pubspec.Yaml

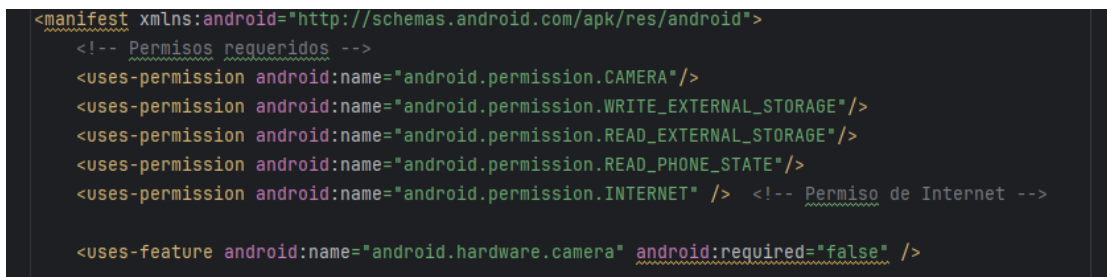


Fig. 59. Dependencias Del Archivo Androidmanifest.Xml

Esta son las configuraciones principales de sistema para reconocimiento de prendas y alimentar el dataset esto permite implementa método de autoaprendizaje para que el modelo mejores su precisión en reconocer prendas sin dificultad alguna.

2.5.8 DIAGRAMA DE CASO DE USO

Los diagramas de caso de uso son diagramas de comportamientos que permite representar los procesos de un sistema. Nos son lenguaje de programación es un lenguaje del modelo es decir que explica cómo está planificado el sistema tanto para usuario y administrado. Esto hace referente a la estructura el cual incluye todos los elementos de sistema y como está relacionado entre sí [53].

Las funciones del sistema representan el punto de vista del usuario que tiene como nombre actor en UML, esto no significa que debe ser un ser humano, sino que es el rol también se le atribuye al sistema externo que accede a otro sistema, en poca palabra es la relación entre un actor y sus requisito o expectativa del sistema, representado la acciones a la que puede acceder y a la que no tiene permiso [53].

Diagrama de caso de uso general para los usuarios y administrador

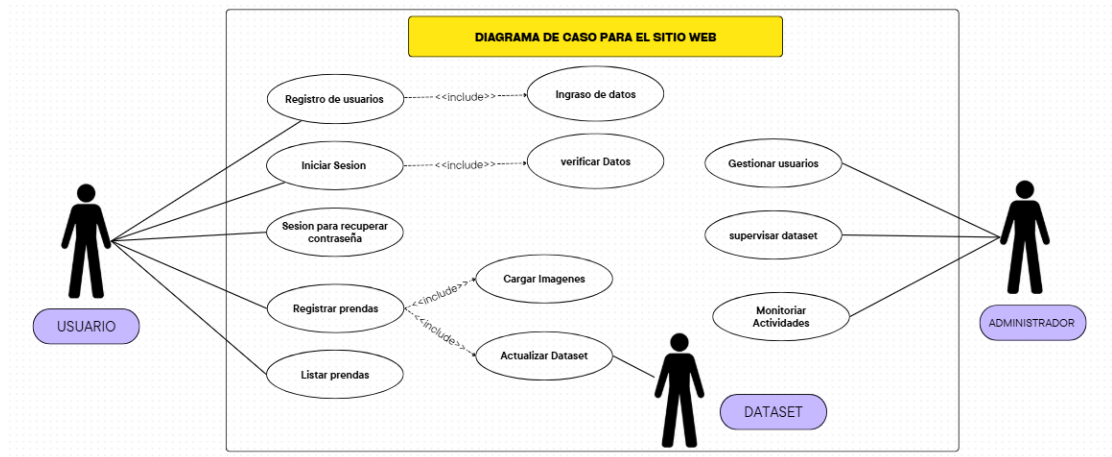


Fig. 60. Diagrama de caso de uso para el sistema web

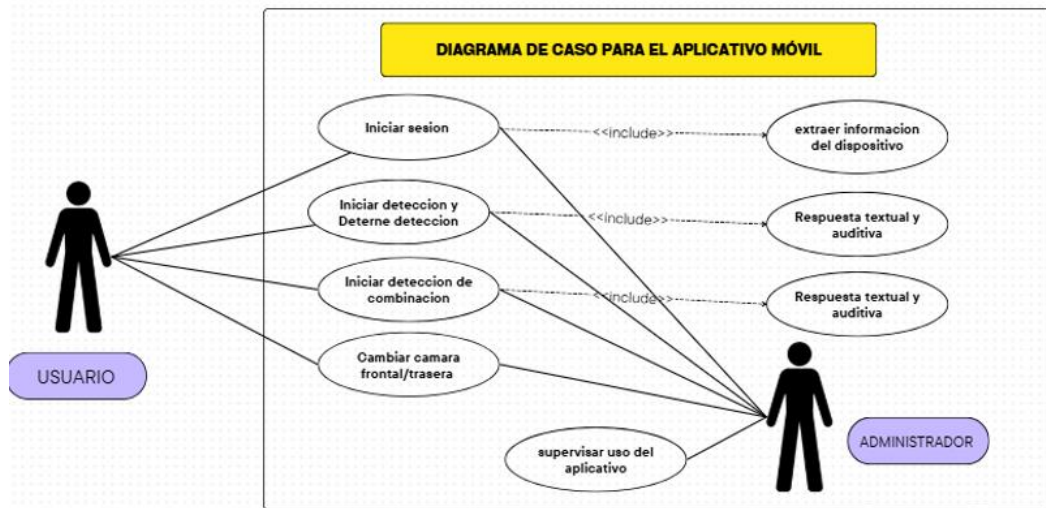


Fig. 61. Diagrama de caso de uso del aplicativo móvil

2.5.9 MODELO DE DATOS

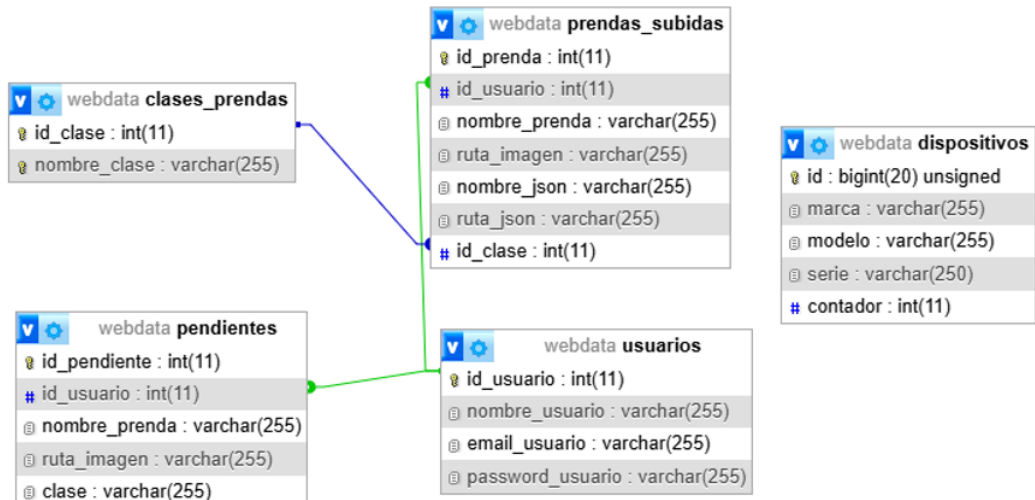


Fig. 62. Base de Datos del sistema web y aplicativo móvil

2.6 DISEÑO DE INTERFAZ

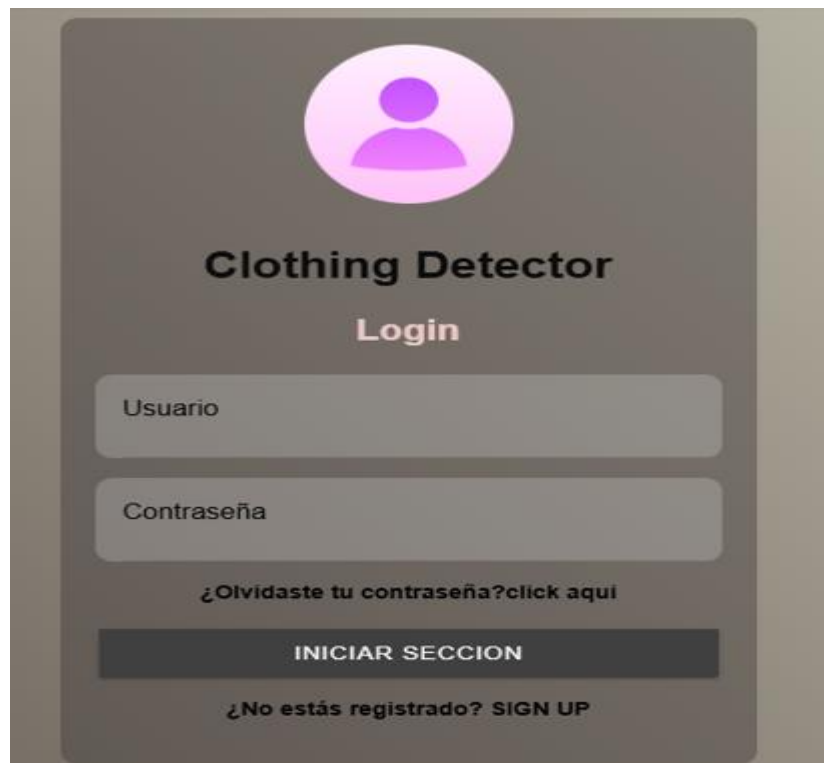


Fig. 63. Interfaz de inicio de sesión del sistema web

La interfaz de inicio de la página web cuenta con la parte de login el cual sirve para ingresar credenciales para permitir ir a apartado de registro, cuenta también con la opción para restablecer contraseña y para el apartado registro. Los colores

utilizados en la interfaz son tonalidades grises ya que esto permitirá que la persona con problema para reconocer colores pueda navegar si dificulta.

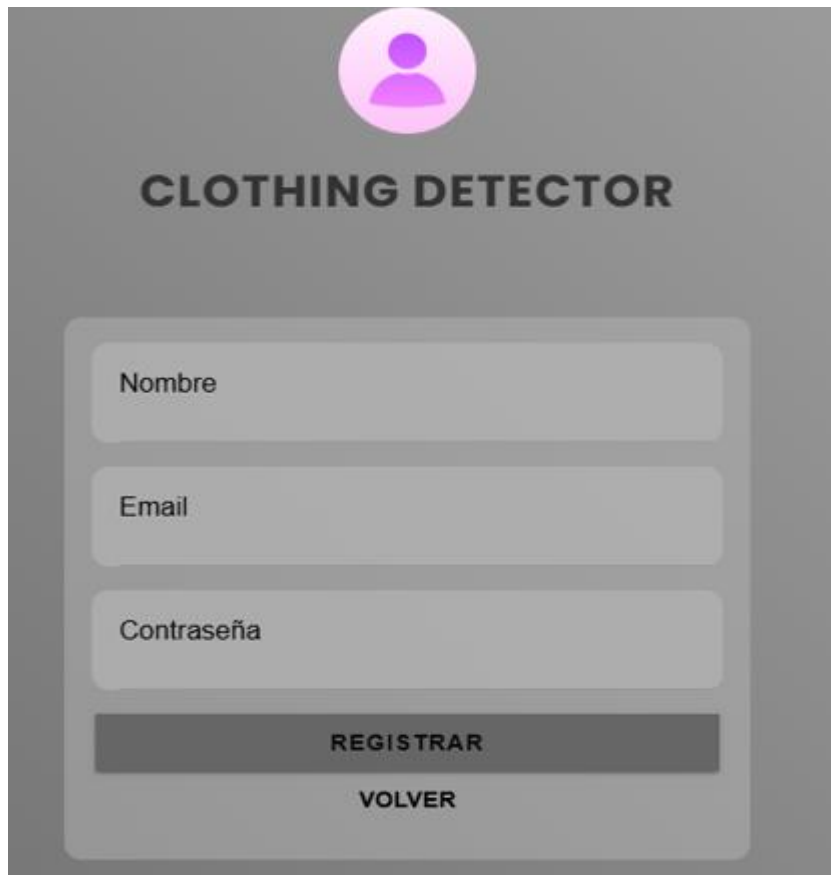
The image shows a user registration interface for a web system. At the top center, there is a circular icon containing a stylized human figure. Below this icon, the text "CLOTHING DETECTOR" is displayed in a bold, uppercase font. The main registration area consists of three stacked input fields, each with a label: "Nombre" (Name), "Email", and "Contraseña" (Password). Below these fields are two buttons: "REGISTRAR" (Register) and "VOLVER" (Return), both in uppercase letters. The entire interface is set against a dark gray background.

Fig. 64. Interfaz de registro de usuario del sistema web.

La siguiente interfaz es la de registra usuario en esta parte se debe ingresa la información solicitada que son Nombre, Email, Contraseña. Antes de procesar el registro se enviará un código de verificación al usuario el cual debe registrarlo para validad su correo, finalizando el procesamiento de validación se procede a registrar los datos en la base.

**Restablecer
Contraseña**

Ingrese su correo Electrónico

elisa_sr@outlook.es

SOLICITAR CÓDIGO

Ingrese tu nueva contraseña

Vuelva a Digitar la contraseña

ACTUALIZAR CONTRASEÑA

VOLVER

Fig. 65. Interfaz de restablecer contraseña del sistema web.

En esta sección restablecer contraseña permite a los usuarios hacer una actualización de la contraseña si por algún motivo se olvidó de la misma. Aquí se solicitará el correo para verificar si está registrado, si es correcto se enviará un código al correo para después procesar la actualización de contraseñas.



Fig. 66. Interfaz de registro de prendas del sistema web

La siguiente interfaz es de la página principal aquí el usuario puede registrar la prenda usando la opción tomar foto, la imagen debe tener el nombre el cual debe ser ingresado en nombre de prenda o clase, una vez este campo tenga contenido se procede a registrar prendas, aquí el proceso interno extraerá coordenadas de la segmentación a la imagen almacenará los datos en un JSON, terminado el proceso de registrar prendas el usuario puede actualizar el dataset aquí hay un aumento de datos automático esto se recomienda para que la foto se ajuste según el brillo y saturación. El aumento de datos finaliza enviando todas las imágenes al dataset principal de modelo.



Fig. 67. Interfaz para listar prendas registradas

La siguiente interfaz permite visualizar todas las prendas que el usuario ha ingresado, esto permite también visualizar si por un caso uno de los patrones de

vestimenta ingresado por el usuario no fue registro se le mostrara como pendiente esto significa que es el administrado que accedera a esta información para revisarla y poder verificar si es una información validad para alimentar el dataset.

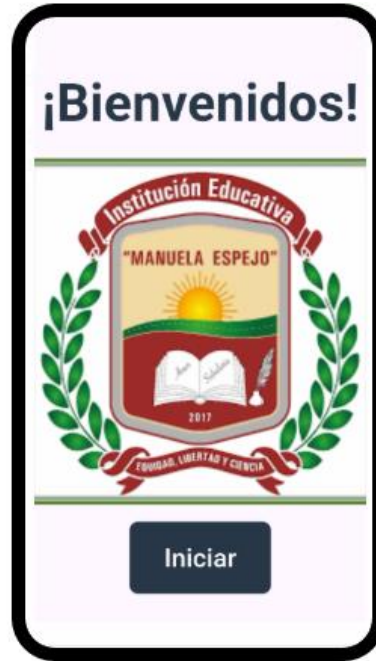


Fig. 68. Interfaz de inicio del dispositivo móvil.

Esta es la interfaz del aplicativo móvil el cual al iniciar automáticamente extrae datos del dispositivo eso permite una facilidad de uso para los usuarios que tiene problemas visuales permitiendo tener un monitoreo de los dispositivos donde se instala el aplicativo más las veces de uso.

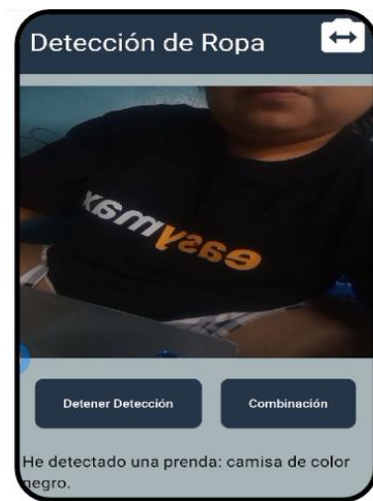


Fig. 69. Detección de prendas y sus colores

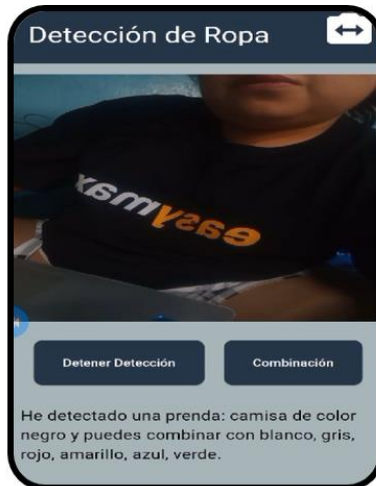


Fig. 70. Detección de prendas, color y combinación

Este es el diseño de la parte principal del aplicativo, el cual está diseñado para iniciar detecciones y combinación es decir que el usuario puede iniciar y detener el proceso de captura en tiempo real obteniendo como repuesta la descripción textual y auditiva, la opción de combinar brinda una retroalimentación de cuáles son el tono recomendable para usar la prenda reconocida, además está la parte superior el cual tiene un icono que permite cambiar la cámara por frontal o trasera.

2.7 PRUEBAS

Las pruebas que se realizaron fueron con el objetivo de verificar que el sistema web y aplicativo móvil funcione adecuadamente, esto permite encontrar posibles fallos y corregirlos para garantizar un funcionamiento correcto de cada componente que tiene el sistema de reconocimiento de prendas con sus respectivos colores.

Prueba N° 1: Inicio de sesión en el sistema web	
Objetivo	Comprobar que el ingreso de los usuarios al sitio web sea de manera correcta.
Descripción	El usuario debe ingresar las credenciales que son email y

	contraseña para acceder a la interfaz principal.
Rol	Usuario-Administrado
Caso N° 1: Verifica que las credenciales de inicio de sesión sean correctas	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Correo electrónico • Contraseña 	<ul style="list-style-type: none"> • El sistema web valida que el correo esta registrado y que la contraseña sea correcta • El sistema web muestra un mensaje exitoso y permite dirigirse a la siguiente interfaz • Muestra las funcionalidades disponibles del sistema web
Caso N° 2: Usuarios con credenciales incorrecta para iniciar sesión	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Correo electrónico • contraseña 	El sistema web mostrara un mensaje el cual dice que el correo o la contraseña son incorrectas.
Caso N° 3: Usuario usa campo vacío para iniciar sesión	
Ninguno	El sistema web muestra un mensaje que por favor rellene los campo con la información solicitad.
Resultados de la prueba	
Resultados	Evaluación

El sistema web permite un inicio de sesión con credenciales correctas	Exitoso
---	---------

Tabla 5 Prueba de Inicio de sesión del sistema web

Prueba N°2: Registro de usuario para el sistema web	
Objetivo	Comprobar que los datos ingresados del usuario en el sitio web se guarden correctamente.
Descripción	El usuario ingresa la información correspondiente para acceder a la interfaz siguiente.
Rol	Usuario
Caso N°1: Registro de usuarios	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Nombre • Email • Contraseña 	<ul style="list-style-type: none"> • El sistema web debe validar los datos ingresados por el usuario. • En la base de datos se deben ingresar los datos y permitir el inicio de sesión con los nuevos registros.
Caso N°2: Registros con información incompleta de los usuarios	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Nombre • Contraseña 	Si la información ingresada está incompleta no se registrarán los datos y se visualizará un mensaje de error al registrarse vuelva a intentarlo.

Caso N°3: Información vacía al registrar usuario	
Ninguno	El sistema web pedirá que los campos estén completos para poder guardar.
Resultados de la prueba	
Resultados	Evaluación
El sistema web permite realizar el registro de forma correcta con la información y credenciales de usuario.	Exitoso

Tabla 6 Prueba de registro de usuario

Prueba N°3: Registro de prendas en el sitio web	
Objetivo	verificar que se pueda agregar prendas mediante el sitio web con su respectiva información y guardarlas en el dataset
Descripción	El usuario registrará las prendas subidas, estas se almacenarán en la base de datos y permitirán actualizar el dataset del modelo entrenado.
Rol	Usuario-Administrador
Caso N°1: Registro de prendas	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Nombre • Imagen • Coordenadas 	<ul style="list-style-type: none"> • El sistema web debe cargar la imagen. • Se debe registrar el nombre de la prenda.

	<ul style="list-style-type: none"> Automáticamente se hace segmentaciones y se extraen coordenadas
Caso N°2: Registros prendas incompleto	
Datos de entrada	Datos de salida
Nombre Imágenes Coordenadas	Si los datos son ingresados, pero no se extrajeron coordenada las imágenes se guardarán como pendientes.
Caso N°3: Información vacía	
Ninguno	El sistema web no permitirá registrar nada
Resultados de la prueba	
Resultados	Evaluación
El sistema web permite registro de prenda después de registrar permitirá la opción de actualizar dataset el cual enviará solo las prendas registradas.	Exitoso

Tabla 7 Registro de prendas pruebas

Prueba N°4: Iniciar sesión en aplicativo móvil	
Objetivo	Extraer datos del dispositivo para el respectivo uso del aplicativo móvil
Descripción	Al dar Clic en el botón iniciar automáticamente se extraerá la información del dispositivo.

Rol	Usuario
Caso N°1: Registro información del dispositivo	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Marca • Modelo • Identificador 	<ul style="list-style-type: none"> • El aplicativo móvil extraerá esta información automáticamente. • Se almacenará en la base de datos esta información. • Permite el acceso a la siguiente interfaz
Caso N°2: Registros incompleta de la información	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Marca • Modelo • Identificador 	Si una de esta información no es extraída adecuadamente no se permite el acceso.
Caso N°3: Información vacía	
Ninguno	El sistema no registrar el dispositivo ni se le permitirá el acceso a la siguiente interfaz.
Resultados de la prueba	
Resultados	Evaluación
El aplicativo móvil permite registrar el dispositivo y dar acceso a las funcionalidades.	Exitoso

Tabla 8 Registro de dispositivos

Prueba N°5: Iniciar Detecciones en tiempo real desde el aplicativo móvil	
Objetivo	Realiza detecciones en tiempo real proporcionando una descripción de su resultado en forma textual y auditiva.
Descripción	El aplicativo móvil permite procesar imágenes en tiempo real brindando una descripción de los resultados de prendas con su respectivo color.
Rol	Usuario
Caso N°1: Iniciar captura en tiempo real	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Procesar imágenes • Resultados 	<ul style="list-style-type: none"> • El aplicativo móvil permite procesar imagen en tiempo real. • Sus resultados se muestran en forma textual y auditiva
Caso N°2: Iniciar captura en tiempo real sin mostrar nada	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Procesar imágenes • Resultados 	Si el usuario no muestra imagen y no inicia las detecciones no se brindará resultados.
Resultados de la prueba	
Resultados	Evaluación
El aplicativo permite hacer detecciones en tiempo real una vez se inicie las	Exitoso

detecciones brindando una respuesta textual y auditiva.	
---	--

Tabla 9 Detecciones en tiempo real

Prueba N°6: Entrenamiento en segundo plano después de actualizar el dataset	
Objetivo	Realizar una actualización del modelo usando los datos ingresados de forma correcta en el sistema web.
Descripción	Una vez ingresado los datos desde el sistema web y realizado la respectiva actualización del dataset en segundo plano se ejecuta el reentrenamiento del modelo.
Rol	Administrador
Caso N°1: Entrenamiento del modelo con el formato adecuado	
Datos de entrada	Datos de salida
<ul style="list-style-type: none"> • Imágenes almacenadas en el dataset • Clase mayor a 500 • Archivos TXT en formato YOLO 	<ul style="list-style-type: none"> • Una vez obtenido la actualización del dataset con los datos correspondiente se procede a realizar un conteo de imágenes por clase para poder reentrenar el modelo.
Caso N°2: Entrenamiento del modelo con datos incorrecto	
Datos de entrada	Datos de salida

<ul style="list-style-type: none"> • Imágenes • Archivos con formato incorrecto • Clase incompleta 	Si el dataset tiene un formato incorrecto dará error en el entrenamiento, además de que si el conteo de clase es incompleto no las tomara en cuenta hasta que supere las 500.
Resultados de la prueba	
Resultados	Evaluación
El sistema procesa correctamente el reentrenamiento con los nuevos datos.	Exitoso

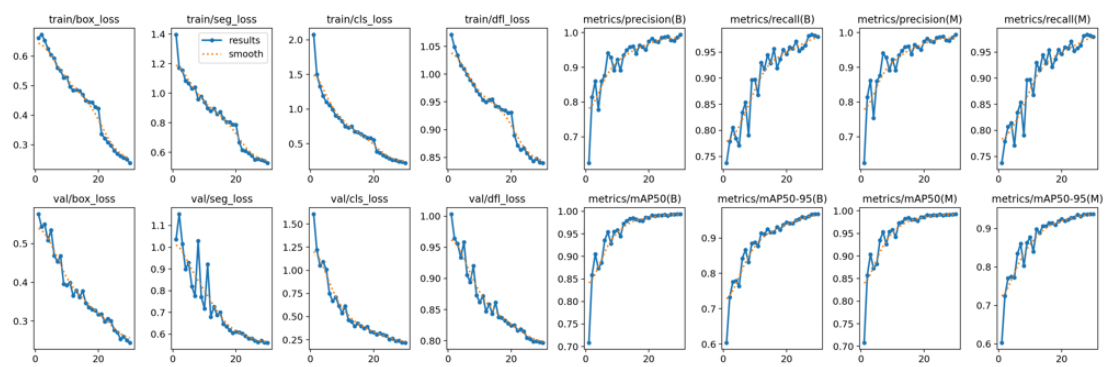
Tabla 10 Entrenamiento del modelo

2.8 RESULTADOS

las pruebas realizadas se realizaron con el propósito de verificar que el sistema web funciones adecuadamente para el ingreso de nuevo patrones de vestimenta, para esto se realizó un preentrenamiento de un modelo para segmentar y extraer coordenadas de la nueva imagen subidas por el usuario. Ajustando el umbral del modelo se pudo comprobar que el modelo puede reconocer en imagen patrone nuevo, es decir la clase entrenada son blusa, camisa, buso y short, pero se puede reconocer hasta prendas como pantalones y añadirlo en una nueva clase, sin embargo, si la nueva clase tiene pocas imágenes no se tomara en cuenta hasta que su cantidad de información supere las 600 fotos esto permite un mejor aprendizaje del modelo.

Otra funcionalidad que se comprobó son las del aplicativo móvil en realizar las detecciones en tiempo real con el propósito de determina si el modelo hace las detecciones de manera correcta, para ello se hizo prueba de reconocimiento de varios tipos de prendas en diferente sitio e iluminaciones y de esta formar validar que todo esté en perfecto estado.

Resultado del modelo entrenado con las prendas ingresadas desde el sitio web



RESULTADOS		
PATRONES DE PRENDAS	cantidad	Resultados del entrenamiento
Shorts	631	95.86
buso	619	97.35
camisa	623	98.01
blusa	629	97.20

Tabla 11 resultado de entrenamiento

Fig. 71. Resultado general de entrenamiento desde la página web

Los resultados muestran como disminuye consistentemente las pérdidas de entrenamiento y validación, esto indica que el modelo está aprendiendo de manera efectiva. En cuanto a las métricas, se observa un incremento continuo en la precisión y el recall tanto para las clases, reflejando así un buen desempeño en las detecciones. Adema las métricas de mAP50 y mAP50-95 se ven que han mejorado superando casi un 95% lo que significa que tiene una excelente capacidad para realizar detecciones brindando resultados positivos en el respectivo procesamiento.

Resultado de las detecciones realizada desde el dispositivo móvil

Clase	Exitoso		fallos		No ejecuta	
	Total, de imágenes	%	Total, de imágenes	%	Total, de imágenes	%
Blusas	8	100	0	0	0	0
Shorts	8	100	0	0	0	0
Camisa	8	98	2	2	0	0
Buso	8	100	0	0	0	0

Tabla 12 Tablas de Resultado de prueba

Para comprobar el funcionamiento del aplicativo se seleccionaron prendas de las clases ingresada de diferentes colores, esto permite conocer si el aplicativo funciona adecuadamente, cuando se usa el dispositivo para captura imágenes en tiempo real y enviarlas al servidor cuya respuesta se devuelve y se proyecta en texto/audio.

En la primera clase que corresponde a 8 prendas de blusa cuyos colores son negro, blanco, azul y roja en diferente modelo sus resultados demostraron el 100% de precisión y un total de 0% de fallo y 0% de no ejecución esto demuestra que el aplicativo funciona de manera correcta.

En la segunda clase que corresponde a 8 prendas de shorts cuyos colores son negro, blanco, azul y roja en diferente modelo sus resultados demostraron el 100% de precisión y un total de 0% de fallo y 0% de no ejecución esto demuestra que el aplicativo funciona de manera correcta.

En la tercera clase que corresponde a 8 de camisa cuyos colores son negro, blanco, azul y roja en diferente modelo sus resultados demostraron el 98% de precisión y un total de 2% de fallo esto se debe a la iluminación y 0% de no ejecución esto demuestra que el aplicativo funciona de manera correcta.

En la cuarta clase que corresponde a 8 de buso cuyos colores son negro, blanco, azul y roja en diferente modelo sus resultados demostraron el 100% de precisión y un total de 0% y 0% de no ejecución esto demuestra que el aplicativo funciona de manera correcta.

Se realizó una encuesta dirigida a lo docente de la U.E Manuela Espejo a un total de 8 docentes, el banco de pregunta se realizó con la finalidad de obtener información de cuanto estudiante tiene problema de visión específicamente en reconocer colores o objetos, otra pregunta se originó con el fin de conocer la percepción de los docentes sobre el uso del aplicativo y los posibles beneficios que puede surgir con la propuesta aplicada.

La encuesta se encuentra en [Anexo 4]. Este banco de pregunta realizada a 8 docente de la U.E “Manuela Espejo” permitió conocer que 5 estudiante tiene problemas para reconocer objeto y otro con dificultad para identificar colore, además un 75% de los docentes considera que un aplicativo móvil para el reconocimiento de patrones de prendas y colores sería beneficioso para los estudiantes, ya que mejoraría su autonomía en el proceso de vestirse.

Otra de la pregunta cuya respuesta obtenida fue que el 100% de los docentes cree que la tecnología móvil podría mejorar la accesibilidad de los estudiantes con discapacidad mientras en el aula.

En la quinta pregunta con un 100% de los docentes están dispuestos a integrar la aplicación en sus clases, siempre y cuando sea fácil de utilizar. Además de que creen que el aplicativo puede fomentar la independencia y confianza de los estudiantes en el proceso de vestirse.

Como resultado de la séptima pregunta el 75% de los docentes opina que el aplicativo debe ser compatible con dispositivo de uso frecuentes el 12.5% opina que es importante, pero no imprescindible siempre y cuando funciones en lo dispositivo más utilizado y un 12.5% opino que es algo relevante, aunque no es prioritario si cumple con su propósito.

Una de las características adicionales que los docentes creen que el aplicativo debe tener con un 87.5% es que tenga un diseño de interfaz amigable y fácil de navegar,

ya que esto con un 100% en la opinión de lo docente creen que beneficiaría a fomentar la autonomía. Otra característica con un 8735% es que el aplicativo debe reconocer y con un 12.5% debe describir sus detecciones, como última pregunta uno de los desafíos que pueden enfrentar lo estudiante al usar el aplicativo con un 75% es la conexión a internet y 25% es la baja precisión en el reconocimiento de prendas y colores.

CONCLUSIONES

El proyecto se finalizó desarrollando el aplicativo móvil para el reconocimiento de prendas de vestir con sus respectivos colores, el cual captura imágenes en tiempo real y realiza detecciones brindando como respuesta textual/auditiva sobre los resultados, va dirigido a la persona con dificultad de reconocer objetos o identificar colores.

Se realizó una investigación sobre los algoritmos de visión por computador, lo que facilitó el procesamiento de las imágenes desde dispositivo móvil, las librerías utilizadas fueron CV2 y ultralytics, que se implementaron en el API del sistema, el cual procesa todas las solicitudes desde la interfaz con este proceso se logró cumplir con uno de los objetivos de este proyecto, brindando la capacidad de detectar y procesar las imágenes de manera eficiente.

Además, se desarrolló un sistema web el cual permite que los usuarios suban imágenes para la actualización del dataset. Cuando las imágenes son cargadas desde la interfaz automáticamente se realiza segmentaciones de esta, extrayendo coordenadas las cuales se almacena en un archivo JSON. Esta información se enviará al dataset principal y en segundo plano se ejecuta dos comandos el cual permite transformar esto datos en el formato YOLO y reentrenar el modelo.

La integración del modelo YOLOv8, el cual es una red neuronal convolucional, facilitando que las imágenes procesan desde el aplicativo móvil sea reconocidas enviado como resultado la detección del patrón de prendas, el color y la combinación. Esto facilitó las detecciones en tiempo real desde el aplicativo móvil, logrando una buena precisión en el reconocimiento de prendas con sus respectivos colores.

Como finalización de todo el sistema desarrollado se evaluó la precisión de las detecciones en diferentes condiciones de iluminación cumpliendo con un resultado del 98% de precisión en reconocer las prendas. El resultado obtenido confirma que tanto el sistema web como el aplicativo móvil cumple con la funcionalidad planteada en el inicio de proyecto, ofreciendo una herramienta que ayude a las personas con problemas visuales en el proceso de vestirse de forma individual.

RECOMENDACIONES

- ❖ Para el uso de aplicativo móvil se realizó un manual de usuario el cual es disponible de manera detallada para el funcionamiento respectivo además que cuenta con una instrucción clara para el uso de este.
- ❖ Cuando el usuario utilice sistema web cuenta con opciones para tomar foto o subir desde la galería de la PC. Esta debe ser individuales como se utiliza un modelo preentrenado la cantidad de imágenes de las clases ya registradas no se limitará sin embargo el modelo puede llegar a reconocer patrones nuevos, se les recomienda utilizar alrededor de 600 imágenes como mínimo para reentrenar el modelo con clase nueva.
- ❖ Para el uso de dispositivo móvil el usuario cuenta con cámara trasera y delantera para realizar detecciones además si la condición de iluminación no es buena se usa el flash para ayudar a capturar las imágenes de forma correcta para brindar una respuesta precisa de cuál es el color y prenda detectada.
- ❖ Con el propósito de que ambos sistemas se puedan ejecutar de manera correcta es necesario darle permiso a la cámara de dispositivo además de que el usuario debe contar con acceso a internet.

BIBLIOGRAFÍA

- [1] M. D. Salud, «Estadísticas de Discapacidad,» 2022. [En línea]. Available: <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>. [Último acceso: 1 05 2024].
- [2] O. M. d. l. Salud, «Organización Mundial de la Salud" (OMS),» 2024. [En línea]. Available: [https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment#:~:text=Entre%20estos%201000%20millones%20de,y%20la%20retinopat%C3%ADa%20diab%C3%A9tica%20\(3%2C](https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment#:~:text=Entre%20estos%201000%20millones%20de,y%20la%20retinopat%C3%ADa%20diab%C3%A9tica%20(3%2C). [Último acceso: 1 5 2024].
- [3] «rededuca,» [En línea]. Available: <https://www.rededuca.net/blog/educacion-y-docencia/discapacidad-visual>.
- [4] «blueconemonochromacy,» [En línea]. Available: <https://www.blueconemonochromacy.org/es/monocromatismo-de-conos-azules/>.
- [5] M. -. Mathematics., «"Computer Vision and Machine Learning",» 2023. [En línea]. Available: https://www.mdpi.com/journal/mathematics/special_issues/Comput_Vision_Mach_Learn. [Último acceso: 10 5 2024].
- [6] «escuelasecuador,» [En línea]. Available: https://www.escuelasecuador.com/escuela-de-educacion-basica-manuela-espejo-santa-elena-la-libertad-24h00256#google_vignette.
- [7] S. R. sepulveda Osses, «Repositorio Academico de la universidad de chile,» 2020. [En línea]. Available: <https://repositorio.uchile.cl/handle/2250/176546>.

- [8] G. Parés Marí, «Universitat Politècnica de València,» [En línea]. Available: <https://m.riunet.upv.es/bitstream/handle/10251/195577/Pares%20-%20Desarrollo%20de%20un%20sistema%20de%20deteccion%20de%20prendas%20de%20ropa%20en%20imagenes%20de%20moda%20mediante%20la%20....pdf?sequence=1&isAllowed=y>.
- [9] A. E. Boada, «universitat politècnica de catalunya,» 01 07 2024. [En línea]. Available: <https://upcommons.upc.edu/bitstream/handle/2117/412627/189503.pdf;jsessionid=8D40A78E438CCD48ED45763192112D77?sequence=2>.
- [10] «PuntoDIS,» [En línea]. Available: https://puntodis.com/featured_item/discapacidad-visual/. [Último acceso: 10 5 2024].
- [11] «optica avenida,» 4 enero 2021. [En línea]. Available: <https://opticasavenida.com/como-tratar-la-acromatopsia/#:~:text=Tratamientos%20del%20monocromatismo,ayudan%20a%20una%20mejor%20visi%C3%B3n..>
- [12] «Generacion Anahuac,» [En línea]. Available: <https://www.anahuac.mx/generacion-anahuac/la-influencia-de-la-tecnologia-en-nuestra-vida-cotidiana#:~:text=La%20tecnolog%C3%ADa%20bien%20aplicada%20nos,distancias%20con%20amistades%20o%20familiares..>
- [13] g. d. ecuador, «plan de creacion de oportunidades,» [En línea]. Available: <https://www.planificacion.gob.ec/wp-content/uploads/2021/09/Plan-de-Creacio%CC%81n-de-Oportunidades-2021-2025-Aprobado.pdf>.
- [14] «Questionpro,» [En línea]. Available: <https://www.questionpro.com/blog/es/investigacion-exploratoria/>.

- [15] E. d. E. Significados, «Enciclopedia Significados.» [En línea]. Available: <https://www.significados.com/investigacion-experimental/>.
- [16] Valtx, «Valtx.» [En línea]. Available: Cita completa: [3] . Valtx, “Metodologías de desarrollo de software: ¿Qué son y para qué sirven? - Valtx.” [Online]. Available: <https://www.valtx.pe/blog/metodologias-para-el-desarrollo-de-software-que-son-y-para-que-sirven#:~:text=Una%20metodolog%C3%ADa%20de%20desarrollo%20de%20software%20que%20sirven%20para%20qu%C3%A9%20sirven,una%20metodolog%C3%ADa%20de%20desarrollo%20de%20software%20que%20sirven%20para%20qu%C3%A9%20sirven,una%20metodolog%C3%ADa%20de%20desarrollo%20de%20software%20que%20sirven%20para%20qu%C3%A9%20sirven>.
- [17] N. a. found, «ESCUELA DE EDUCACION BASICA MANUELA ESPEJO,» [En línea]. Available: <https://www.escuelasecuador.com/escuela-de-educacion-basica-manuela-espejo-santa-elena-la-libertad-24h00256>.
- [18] «unesco,» [En línea]. Available: <https://www.unesco.org/es/artificial-intelligence>.
- [19] «edsrobotics,» [En línea]. Available: <https://www.edsrobotics.com/blog/vision-computador-que-es/>.
- [20] «ISID,» [En línea]. Available: <https://isid.com/es/tecnologia/object-detection/>.
- [21] «Analytuc Vidhya,» [En línea]. Available: https://www-analyticsvidhya-com.translate.goog/blog/2023/11/real-time-object-detection-with-ssds-single-shot-multibox-detectors/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=rq.
- [22] «edsrobotics,» [En línea]. Available: <https://www.edsrobotics.com/blog/vision-computador-que-es/>.
- [23] «ibm,» [En línea]. Available: <https://www.ibm.com/es-es/topics/convolutional-neural-networks>.

- [24] «datacamp,» [En línea]. Available: <https://www.datacamp.com/es/blog/yolo-object-detection-explained>.
- [25] «ibm,» [En línea]. Available: <https://www.ibm.com/docs/es/visual-insights?topic=model-faster-cnn>.
- [26] «GoogleCloud,» [En línea]. Available: <https://cloud.google.com/tpu/docs/tutorials/mask-rcnn-2.x?hl=es-419#:~:text=Mask%20RCNN%20es%20una%20red,de%20segmentaci%C3%B3n%20para%20cada%20uno..>
- [27] «DataScientest,» [En línea]. Available: <https://datascientest.com/es/que-es-el-transfer-learning#:~:text=%E2%80%9CEl%20Transfer%20Learning%2C%20o%20aprendizaje,problemas%20para%20resolver%20otros%20problemas.%E2%80%9D>.
- [28] «IBM,» [En línea]. Available: <https://www.ibm.com/mx-es/topics/deep-learning>.
- [29] «CaseGuard,» [En línea]. Available: <https://caseguard.com/es/articles/que-es-la-segmentacion-de-imagenes-cuales-son-sus-aplicaciones/>.
- [30] «paperswithcode,» [En línea]. Available: <https://paperswithcode.com/task/real-time-object-detection>.
- [31] «Ceupe,» [En línea]. Available: <https://www.ceupe.com/blog/reconocimiento-de-patrones.html>.
- [32] «ArcMap,» [En línea]. Available: <https://desktop.arcgis.com/es/arcmap/latest/extensions/spatial-analyst/image-classification/what-is-image-classification->

Una%20aplicaci%C3%B3n%20web,muy%20popular%20basado%20en%20PHP..

- [41] «Apache Friends,» [En línea]. Available: <https://www.apachefriends.org/es/index.html>.
- [42] «rockcontent,» [En línea]. Available: <https://rockcontent.com/es/blog/framework/>.
- [43] «AWS,» [En línea]. Available: <https://aws.amazon.com/es/what-is/flutter/>.
- [44] H. Rodríguez, «Crehana,» 28 abril 2021. [En línea]. Available: <https://www.crehana.com/blog/transformacion-digital/que-es-opencv/>.
- [45] I. J. Martínez, «TIN ACADEMIA,» 31 06 2023. [En línea]. Available: <https://academia.tinoreste.com/deteccion-de-objetos-mediante-color/>.
- [46] M. Escalante, 3 agosto 2023. [En línea]. Available: https://abcxperts.com/que-es-pycharm-y-su-comparacion-con-otros-ides/?srsltid=AfmBOoph8_UGo3WT8RxSFV4bmwgXiPrfCsTNbaz2iLGKOTZeT3yAzGx.
- [47] «Ciencialatina.org,» [En línea]. Available: <https://ciencialatina.org/index.php/cienciala/article/view/9118/13600..>
- [48] «datascientest,» 2024. [En línea]. Available: <https://datascientest.com/es/convolutional-neural-network-es>.
- [49] M. D. L. INVESTIGACIÓN, «revistaalergia.mx,» [En línea]. Available: <https://revistaalergia.mx/ojs/index.php/ram/article/view/199/350>.
- [50] G. Velasco Ramos, «repositoriodspace,» 2021. [En línea]. Available: <http://repositoriodspace.unipamplona.edu.co/jspui/handle/20.500.12744/4331>.

- [51] «disposit digital de document de la UAB,» 2021. [En línea]. Available: <https://ddd.uab.cat/record/248494>. [Último acceso: 2024].
- [52] «Daemon4,» 23 julio 2024. [En línea]. Available: <https://www.daemon4.com/empresa/noticias/arquitectura-cliente-servidor/#:~:text=La%20estructura%20cliente%2Dservidor%20se,eficiente%20de%20las%20aplicaciones%20modernas..>
- [53] IONOS, «Equipo editorial de IONOS,» 24 07 2020. [En línea]. Available: <https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/diagrama-de-casos-de-uso/>.

ANEXOS

ANEXO 1. Manual del sistema web para el usuario

El siguiente Manual explica el uso de sistema web para subir imágenes de prendas y actualizar el dataset del modelo entrenado.

1. Objetivo

Este manual tiene como propósito explicar el uso adecuado del sistema web además de explicar las funcionalidades de tiene.

2. Inicio de sesión del sistema web

Paso 1: El usuario debe ingresar las credenciales la cuales son en usuario él es correo electrónico y contraseña, si en caso el usuario no esta registrado debe ir a la sección de registro dando clic en SIGN UP y si el usuario olvido su contraseña debe hacer clic en ¿olvidaste tu contraseña? Para actualizarla y poder iniciar sesión sin problema.



Fig. 71 Interfaz Login

1. Registro de prendas en el sistema web

Paso 1: El usuario una vez iniciado sesión visualizará la pantalla principal del sistema web, sirve para registrar imágenes y actualizar el dataset, está cuenta con 3 opciones:



Fig. 72. Interfaz principal

- **Opción 1:** Tomar foto esto abrirá una ventana de la cámara el cual tiene dos opciones tomar foto o subir imágenes. Cuando las imágenes se cargan automáticamente se extraen coordenada si la imagen fue detectada correctamente.
- **Opción 2:** Registro de prenda el usuario debe llena el campo nombre de prenda o clase y debe haber cargado la imagen correctamente caso contrario no permite el registro mostrando un mensaje campos incompleto.
- **Opción 3:** actualización del dataset al presionar esta opción todas las imágenes registradas se envían al dataset principal y en segundo plano se empieza un reentrenamiento del modelo usado las nuevas imágenes.

Paso 2: El sistema cuenta con un menú en la parte superior este permite ir a la opción de listar donde se puede visualizar las imágenes registradas y la opción dos que cerrar sesión



Fig. 73. Interfaz de listar prendas

ANEXO 2. Manual del aplicativo móvil

El siguiente Manual explica el uso del aplicativo móvil para realizar detecciones de prenda en tiempo real

Objetivo

Este manual tiene como propósito explicar el uso adecuado del aplicativo móvil además de explicar las funcionalidades de tiene y como usarlas.

1. Inicio de sesión del aplicativo móvil

En la pantalla principal de la aplicación solo cuenta con un botón iniciar, este permite extraer automáticamente información del dispositivo para permitir el ingreso a la siguiente parte.



Fig. 74. Interfaz principal del dispositivo móvil

2. Pantalla principal del dispositivo móvil para realizar detecciones

Paso 1: El aplicativo en la pantalla principal se visualiza la cámara para realizar detecciones esta cuenta con tres opciones que son las siguiente:

Opción 1: El aplicativo móvil tiene un icono en la parte superior que permite cambiar de cámara esto permite que el usuario pueda usar la cámara frontal/trasera

Opción 2: La aplicación cuenta con un botón de iniciar detecciones este permite empezar las detecciones en tiempo real para después visualizar los resultados, si se vuelve a presionar detiene las detecciones.

Opción 3: El otro botón es combinación el cual visualiza permite hacer detecciones igual que el otro botón la diferencia es que permite tener como resultado sugerencia de combinación.

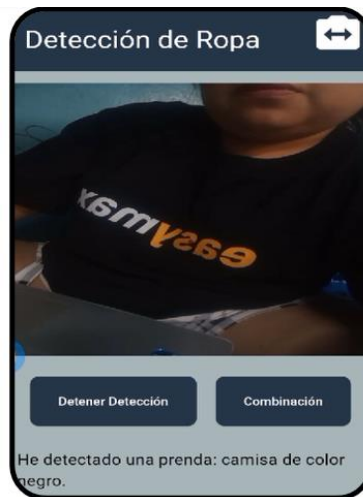


Fig. 75. Interfaz principal del Detector de prendas

Anexo 3 Manual de descarga del aplicativo

La aplicación se encuentra en un drive con acceso público para que los usuarios pueda instalarlos.

Pasos para la descarga e instalación.

Paso 1: Pueden descargar el aplicativo móvil usando el siguiente enlace:

https://drive.google.com/file/d/1wqj70_-J0qKfhRdqBiAasO-3O-xcDaHg/view?usp=sharing

Paso 2: Una vez ingresado al enlace le aparecerá la siguiente opción para la descarga del aplicativo.

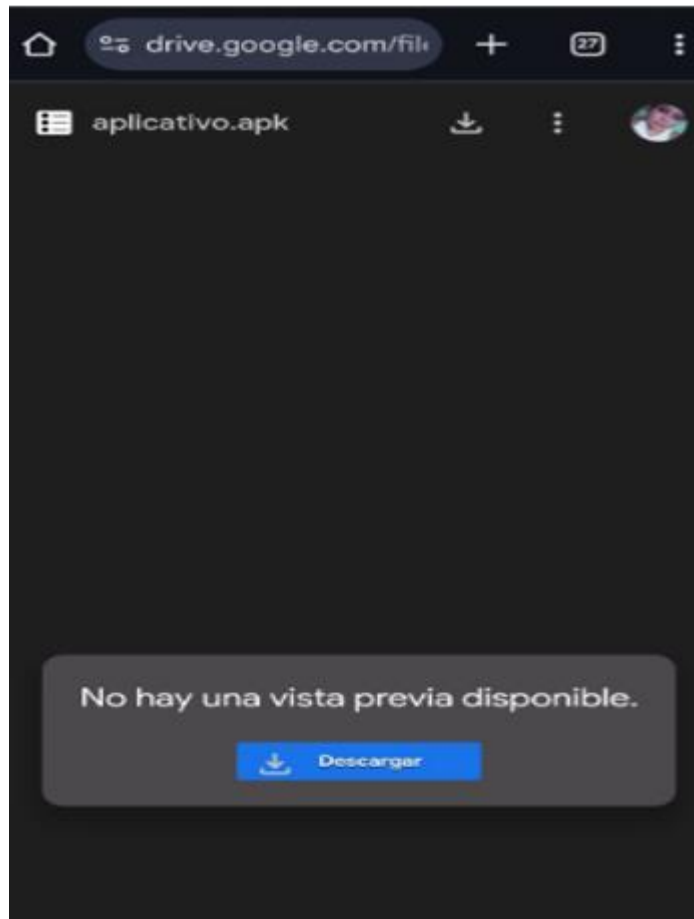


Fig. 76. Vista de Descarga

Paso 3: Seleccionamos Descargar y se abre una ventana el cual dice que Google drive no puede analizar este archivo se omite el mensaje dando clic en “Descargar de todos modos”.

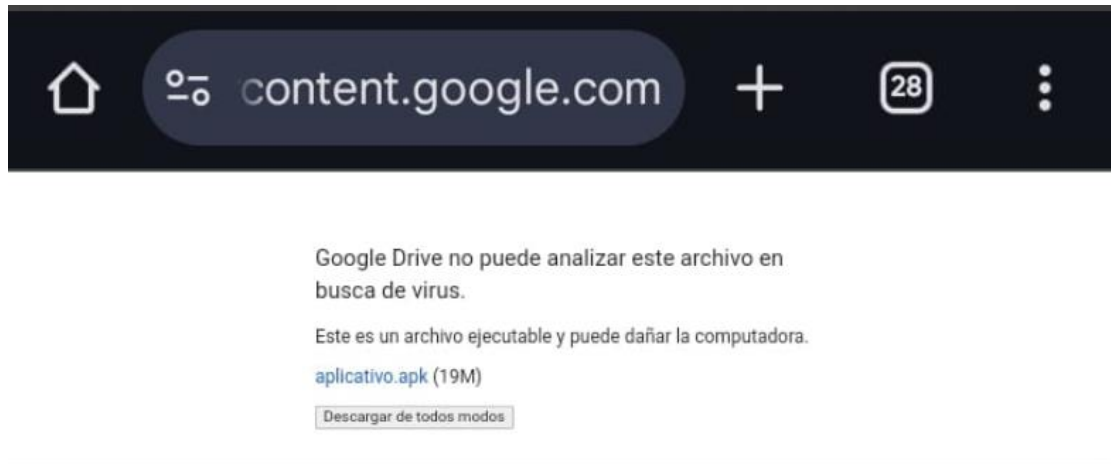


Fig. 77. Paso 2 para instalar el aplicativo móvil

Paso 4: confirmamos la descarga en “Descargar de todos modos”

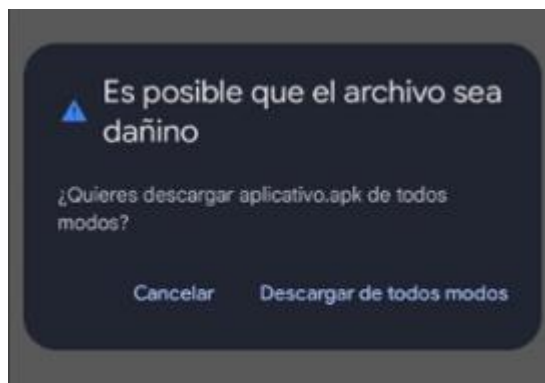


Fig. 78. Paso 3 para instalar el aplicativo móvil

Paso 5: Damos clic en abrir.

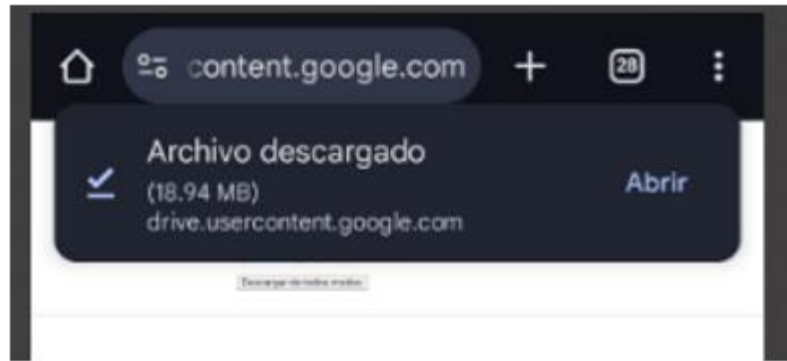


Fig. 79. Paso 4 para instalar el aplicativo móvil

Paso 6: Al dar Clic en abrir nos dirige a la carpeta donde está el aplicativo descargado.



Fig. 80. Paso 5 para instalar el aplicativo móvil

Paso 7: Damos clic y se visualiza la ventana de instalación donde se debe presionar “Instalar”.

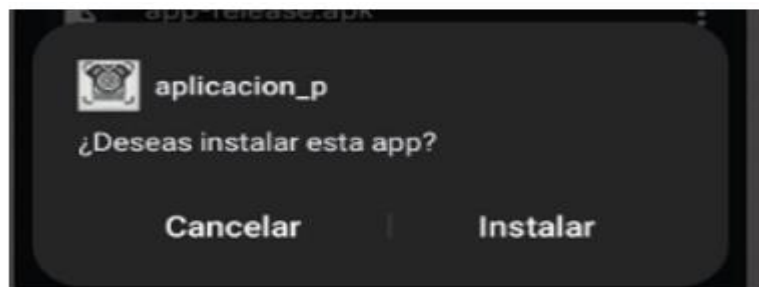


Fig. 81. Paso 6 para instalar el aplicativo móvil

Paso 8: Una vez iniciado el proceso de instalación se debe espera hasta que salga una ventana que permita abrir la aplicación esto significa que el aplicativo se instaló correctamente.

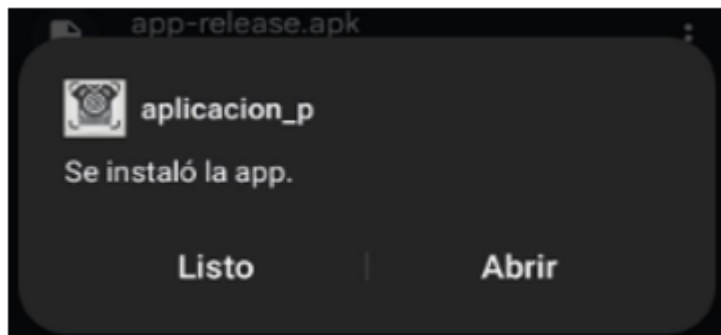


Fig. 82. Paso 7 para instalar el aplicativo móvil

Anexo 4 Formato de la encuesta dirigida a los docentes de la U.E Manuela Espejo

Objetivo	Identifica el número de estudiante con discapacidad visual en la U.E Manuela Espejo y conocer la opinión de los docentes sobre el uso un aplicativo móvil para el reconocimiento de patrones y colores de prenda de vestir.
1	¿En su curso existen estudiante con discapacidad visual?
2	Si su respuesta fue si, ¿cuántos de ellos tienen dificultad para reconocer objeto y distinguir colores?
3	¿Cree que el uso de una aplicación móvil que reconozca patrones y colores de prendas de vestir sería beneficioso para estudiantes con discapacidad visual?
4	¿Qué tan útil considera que sería la tecnología móvil para mejorar la accesibilidad de los estudiantes con discapacidad visual en el aula?
5	¿Estaría dispuesto a integrar el uso de esta aplicación en sus clases para apoyar a estudiantes con discapacidad visual?

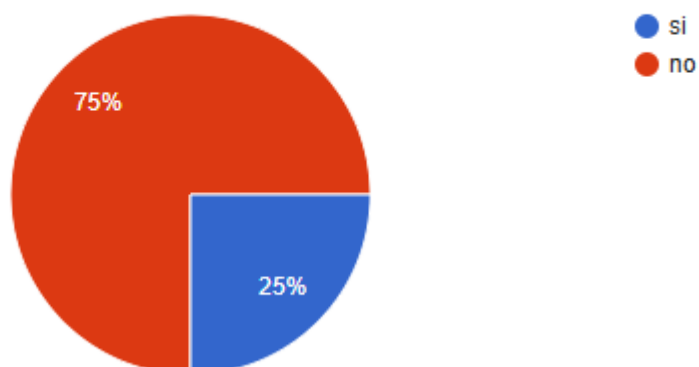
6	¿Cuál sería el mayor beneficio de usar una aplicación de reconocimiento de patrones y colores en el aula?
7	¿Qué tan importante es para usted que la aplicación sea compatible con dispositivos de uso común en el aula (teléfonos móviles, tabletas, etc.)?
8	¿Qué características adicionales consideraría útiles para mejorar la aplicación y adaptarla mejor a las necesidades de los estudiantes con discapacidad visual?
9	¿Qué beneficios cree que tendría una aplicación que ayude a los estudiantes con discapacidad visual a reconocer sus prendas de vestir?
10	¿Qué tipo de funcionalidad le parecería más importante en una aplicación de reconocimiento de prendas de vestir?
11	¿Cuáles cree que son los principales desafíos en el uso de una aplicación móvil para estudiantes con problema visual?

Tabla 13. Encuesta

Anexo 4 Resultados de la encuesta dirigida a los docentes de la U.E Manuela Espejo

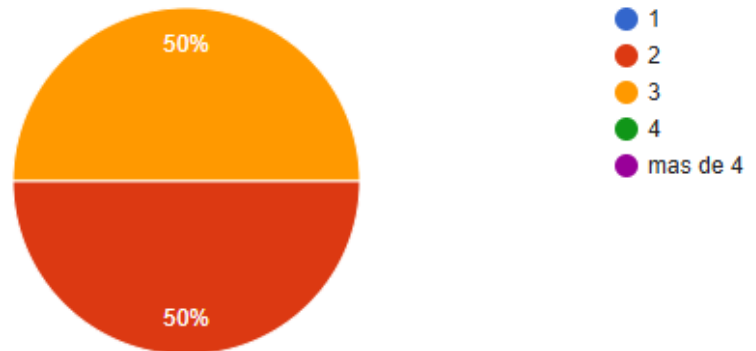
1. ¿En su curso existen estudiante con discapacidad visual?

8 respuestas



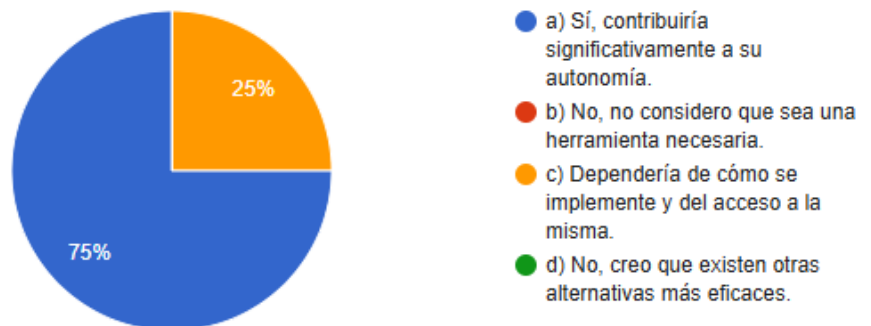
2. Si su respuesta fue si, ¿cuántos de ellos tienen dificultad para reconocer objetos y distinguir colores?

2 respuestas



3. ¿Cree que el uso de una aplicación móvil que reconozca patrones y colores de prendas de vestir sería beneficioso para estudiantes con discapacidad visual?

8 respuestas



4. ¿Qué tan útil considera que sería la tecnología móvil para mejorar la accesibilidad de los estudiantes con discapacidad visual en el aula?

8 respuestas



- a) Mucho, ya que puede facilitar significativamente su vida diaria.
- b) Algo, pero su impacto dependerá de cómo se implemente.
- c) Poco, ya que no se basa tanto en tecnología.
- d) Nada, no creo que sea una herramienta relevante.

5. ¿Estaría dispuesto a integrar el uso de esta aplicación en sus clases para apoyar a estudiantes con discapacidad visual?

8 respuestas



- a) Sí, siempre que sea fácil de usar.
- b) No, prefiero otros métodos tradicionales.
- c) Depende de si la aplicación es compatible con los dispositivos de la escuela.
- d) No, ya tengo herramientas suficientes para apoyar a mis estudiantes.

6. ¿Cuál sería el mayor beneficio de usar una aplicación de reconocimiento de patrones y colores en el aula?

8 respuestas

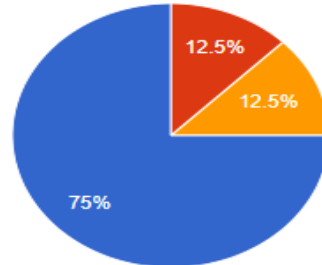


- a) Fomentar la independencia y confianza de los estudiantes.
- b) Ayuda a los docentes a ahorrar tiempo en el aula.
- c) Introducir nuevas formas de enseñanza más dinámicas.
- d) No hay ninguna ventaja significativa para los estudiantes.

7. ¿Cuál sería el mayor beneficio de usar una aplicación de reconocimiento de patrones y colores en el aula? ¿Qué tan importante es para usted que la

aplicación sea compatible con dispositivos de uso común en el aula (teléfonos móviles, tabletas, etc.)?

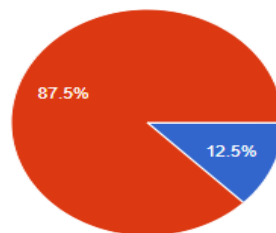
8 respuestas



- a) Es fundamental que sea compatible con todos los dispositivos de uso frecuente.
- b) Es importante, pero no imprescindible si funciona en los dispositivos más utilizados.
- c) Es algo relevante, aunque no es prioritario si cumple su propósito.
- d) No es importante, ya que los estudiantes pueden adaptarse...

8. ¿Qué características adicionales consideraría útiles para mejorar la aplicación y adaptarla mejor a las necesidades de los estudiantes con discapacidad visual?

8 respuestas



- a) Integración con tecnologías de asistencia como lectores de pantalla o asistentes de voz.
- b) Diseño de una interfaz más accesible y fácil de navegar.
- c) Opciones de personalización para adaptarse a las necesidades individuales de cada estudiante.
- d) Ninguna, considero que la aplicación es suficiente en su estado actual.

9. ¿Qué beneficios cree que tendría una aplicación que ayude a los estudiantes con discapacidad visual a reconocer sus prendas de vestir?

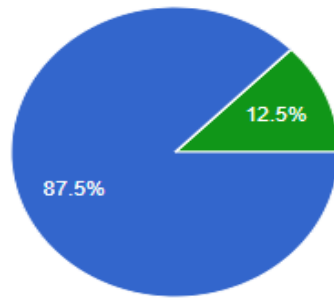
8 respuestas



- a) Fomentar la autonomía
- b) Mejorar la inclusión social
- c) Aumentar la confianza personal de los estudiantes
- d) Todas las anteriores

10. ¿Qué tipo de funcionalidad le parecería más importante en una aplicación de reconocimiento de prendas de vestir?

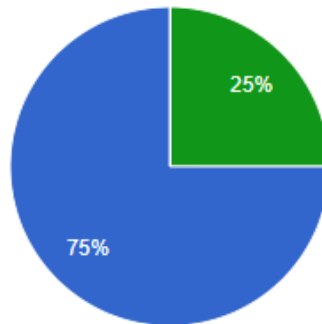
8 respuestas



- a) Reconocimiento y descripción de colores.
- b) Identificación de patrones, estampados o texturas.
- c) Asistencia para combinar colores y estilos.
- d) Descripción de la prenda, como tipo (camisa, pantalón).

11. ¿Cuáles cree que son los principales desafíos en el uso de una aplicación móvil para estudiantes con problema visual

8 respuestas



- a) Baja precisión en el reconocimiento de colores o objetos.
- b) Falta de compatibilidad con tecnologías de asistencia.
- c) Interfaces difíciles de usar para personas con discapacidad visual.
- e) Necesidad de conexión constante a internet.

