



Desarrollo de una aplicación móvil, utilizando reconocimiento facial, para el control de identidad y acceso de los jugadores afiliados a la empresa Mashka fútbol, en el cantón Latacunga.

Gualsaqui Viñamagua, Jomaira Maribel y Vilca Calo, Stalin Joseph

Departamento de Eléctrica y Electrónica

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Trabajo de integración curricular, previo a la obtención del título de Tecnólogo Superior en
Redes y Telecomunicaciones

Ing. Casa Guayta, Carlos Wellington

Latacunga

25 de enero de 2023

Reporte de verificación de contenido

24/1/23, 08:42

Revisión tesis

Informe de originalidad

NOMBRE DEL CURSO
MIC PI PROFESIONAL

NOMBRE DEL ALUMNO
JOMAIRA MARIBEL GUALSAQUI VIAMAGUA

NOMBRE DEL ARCHIVO
JOMAIRA MARIBEL GUALSAQUI VIAMAGUA - Documento sin título

SE HA CREADO EL INFORME
24 ene 2023

Resumen

Fragmentos marcados	0	0 %
Fragmentos citados o entrecomillados	1	6 %

Coincidencias de la Web		
neoattack.com	1	6 %

1 fragmento

Fragmento del alumno **CITADO**

... Marín (2018) define que "es un término referente a un marco de trabajo o esquema que se establece y que se aprovecha para desarrollar y organizar un software determinado"

Mejor coincidencia en la Web

Un Framework, que se podría traducir aproximadamente como marco de trabajo, es el esquema o estructura que se establece y que se aprovecha para desarrollar y organizar un software determinado.

¿Qué es un Framework y para que sirve? - Neo Wiki - NeoAttack <https://neoattack.com/neowiki/framework/>

Casa Guayta, Carlos Welington

C. C.:0502352180



Departamento de eléctrica y electrónica

Carrera de Tecnología Superior en Redes y Telecomunicaciones.

Certificación

Certifico que el trabajo de integración curricular: **“Desarrollo de una aplicación móvil, utilizando reconocimiento facial, para el control de identidad y acceso de los jugadores afiliados a la empresa Mashka fútbol, en el cantón Latacunga.”** fue realizado por los señores **Gualsaqui Viñamagua, Jomaira Maribel y Vilca Calo, Stalin Joseph**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 25 de enero de 2023



Casa Guayta, Carlos Wellington

C. C.: 0502352180



Departamento de eléctrica y electrónica
Carrera de Tecnología Superior en Redes y Telecomunicaciones

Responsabilidad de Autoría

Nosotros, **Gualsaqui Viñamagua, Jomaira Maribel**, con cédula de ciudadanía n° 1725079063, y **Vilca Calo, Stalin Joseph**, con cédula de ciudadanía n° 0504373226 declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **Desarrollo de una aplicación móvil, utilizando reconocimiento facial, para el control de identidad y acceso de los jugadores afiliados a la empresa Mashka fútbol, en el cantón Latacunga** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 25 de enero de 2023

Gualsaqui Viñamagua, Jomaira Maribel

C.C.: 1725079063

Vilca Calo Stalin Joseph

C.C.: 0504373226



Departamento de eléctrica y electrónica

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Autorización de Publicación

Nosotros **Gualsaqui Viñamagua, Jomaira Maribel**, con cédula de ciudadanía n° 1725079063, y **Vilca Calo, Stalin Joseph**, con cédula de ciudadanía n° 0504373226, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Desarrollo de una aplicación móvil, utilizando reconocimiento facial, para el control de identidad y acceso de los jugadores afiliados a la empresa Mashka fútbol, en el cantón Latacunga** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 25 de enero de 2023

Gualsaqui Viñamagua, Jomaira Maribel

C.C.: 1725079063

Vilca Calo Stalin Joseph

C.C.: 0504373226

Dedicatoria

A Dios por su gran amor y misericordia permitiéndole vivir y disfrutar de sus grandezas cada mañana, por estar presente durante este trayecto de mi vida guiándome y ayudándome a crecer como mejor persona.

A mis padres por guiarme siempre para la toma de decisiones los cuales me han educado constantemente en cada momento de mi vida, brindándome oportunidades para ser cada día mejor profesional.

A mis profesores, por impartir día a día mis conocimientos y guiándome en valores profesionales y éticos.

Gualsaqui Viñamagua Jomaira Maribel

Dedicatoria

A Dios por su gran amor y misericordia permitiéndome vivir y disfrutar de sus grandezas cada mañana, por estar presente durante este trayecto de mi vida guiándome y ayudándome a crecer como mejor persona.

A mis padres por guiarme siempre para la toma de decisiones los cuales me han educado constantemente en cada momento de mi vida, brindándome oportunidades para ser cada día mejor profesional.

A mis profesores, por impartir día a día mis conocimientos y guiándome en valores profesionales y éticos.

Vilca Calo Stalin Joseph

Agradecimiento

A gradezco a mi director de tesis, el profesor Ing. Carlos Casa, por su apoyo y orientación durante el proceso de investigación y escritura de mi tesis. Su conocimiento y experiencia han sido invaluable para mí.

También quiero agradecer a mis compañeros de clase por su apoyo y motivación durante el proceso de investigación y escritura. Su entusiasmo y dedicación han sido una gran fuente de inspiración para mí.

Finalmente, quiero agradecer a mis padres por su apoyo incondicional durante todo el proceso. Su amor y apoyo han sido fundamentales para que yo pudiera completar mi tesis con éxito.

Gualsaqui Viñamagua Jomaira Maribel

Agradecimiento

A Dios, por regalarme la vida, la salud y llenarme de sabiduría en todos los años de estudio y así poder cumplir mis objetivos en la vida.

A mis padres, que siempre me han apoyado en cada momento de mi vida gracias a ellas eh tenido la oportunidad de formarme como profesional.

A mis hermanos y familia, por sus consejos y enseñanzas de jamás darse por vencido y cumplir cada una de las metas que me proponga.

A mis profesores, por compartir sus conocimientos impartidos en las aulas para poder aplicarlos en la vida profesional.

A mis amigos por brindarme su ayuda en mis momentos de soledad, gracias por hacerme sentir que no estoy solo gracias por el apoyo

Vilca Calo Stalin Joseph

ÍNDICE DE CONTENIDO

Carátula	1
Reporte de verificación de contenido	2
Certificación	3
Responsabilidad de Autoría	4
Autorización de Publicación	5
Dedicatoria	6
Dedicatoria	7
Agradecimiento	8
Agradecimiento	9
Índice de contenido	10
Índice de figuras	14
Índice de tablas	16
Resumen	18
Abstract	19
Capítulo I: Introducción	20
Antecedentes	20
Planteamiento de problema	21
Justificación	21
Objetivos	22
<i>Objetivo general</i>	22
<i>Objetivos específicos</i>	22
Alcance	22
Capítulo II: Marco Teórico	23
Sistemas de control	23

Control de asistencia	23
Reconocimiento Facial.....	24
Tareas del Reconocimiento Facial.....	24
Fases del Reconocimiento Facial.....	25
Redes Neuronales convolucionales.....	26
Capas convolucionales.....	28
Tecnologías de desarrollo	28
<i>Python</i>	28
<i>Pythonanywhere</i>	29
<i>Open Cv</i>	29
<i>Librería Opencv</i>	29
<i>Face Recognition</i>	30
<i>Framework</i>	30
<i>Gestión de bases de datos</i>	31
<i>Extreme Programming (XP)</i>	31
<i>El proceso de XP</i>	32
Metodología.....	34
<i>Modalidad de investigación</i>	34
<i>Modalidad de campo</i>	34
<i>Recolección de información</i>	34
<i>Procesamiento y análisis de datos</i>	35
<i>Desarrollo del Proyecto</i>	36
Capítulo III: Desarrollo y resultados.....	37
Análisis de resultados.....	37
<i>XP</i>	37

<i>Roles XP</i>	37
Método escogido	38
Análisis de herramientas de desarrollo	38
<i>Android Studio</i>	39
<i>Visual code</i>	39
<i>Python</i>	39
<i>Java</i>	39
<i>API Rest Python</i>	40
<i>JSON</i>	40
Desarrollo de la propuesta	40
Fase 1. Planificación	41
<i>Procesos generales</i>	41
<i>Historia de usuarios.</i>	41
Fase 2. Diseño	47
Fase 3. Codificación	50
<i>Arquitectura del proyecto.</i>	51
<i>Instalación y configuración de herramientas de desarrollo</i>	52
<i>Creación para el registro de rostros.</i>	53
Fase 4. Pruebas	68
<i>Prueba de aceptación</i>	68
Reconocimiento facial	73
Pruebas de la aplicación móvil.	74
Ambiente controlado	74
Características	75
<i>Conexión Internet - wifi oficina.</i>	75

<i>Luz reflejo de luz</i>	76
<i>Posición del teléfono</i>	77
<i>Cámara del teléfono - característica mpx</i>	78
Pruebas ambientes controlado	79
<i>Tiempo de respuesta</i>	79
Ambiente no controlado	86
<i>Conexión internet - datos - señal</i>	86
<i>Cuantos datos se gastos 1000MG - 450MB</i>	86
Luz del sol	86
Posición del teléfono	88
Cámara del teléfono - característica MPX	88
Pruebas ambientes no controlado	90
<i>Tiempo de respuesta</i>	90
Fase 5 lanzamiento	97
<i>Implementación del aplicativo</i>	97
Capítulo IV: Conclusiones y recomendaciones	98
Conclusiones	98
Recomendaciones	99
Bibliografía	100
Anexos	102

ÍNDICE DE FIGURAS

Figura 1 <i>Tareas de Reconocimiento Facial</i>	23
Figura 2 <i>Tareas de Reconocimiento Facial</i>	25
Figura 3 <i>Fases del Reconocimiento Facial</i>	26
Figura 4 <i>Neurona Artificial</i>	27
Figura 5 <i>Resultado de convolucional un filtro de una imagen</i>	28
Figura 6 <i>Proceso de metodología XP</i>	32
Figura 7 <i>Proceso actual que tiene la Empresa Mashka Futbol</i>	41
Figura 8 <i>Propuesta para el proceso de control de identidad Mashka Futbol</i>	41
Figura 9 <i>Arquitectura del proyecto</i>	51
Figura 10 <i>Interfaz principal de visual code</i>	52
Figura 11 <i>Instalación de librerías a través del símbolo del sistema</i>	53
Figura 12 <i>Instalación de la librería opencv</i>	54
Figura 13 <i>Instalación de la librería opencv</i>	54
Figura 14 <i>Resultados de la instalación de la librería</i>	55
Figura 15 <i>Codificación de reconocimiento de imágenes</i>	55
Figura 16 <i>Librería face recognition</i>	56
Figura 17 <i>Librería face encondings</i>	56
Figura 18 <i>Resultado de la codificación</i>	57
Figura 19 <i>Detección de imágenes para la verificación en la base de datos</i>	57
Figura 20 <i>Resultado de la codificación</i>	58
Figura 21 <i>Json permite transformar en diccionario o viceversa</i>	58
Figura 22 <i>Optimización de las líneas de código para que sea dinámico y no básico</i>	59
Figura 23 <i>Ejecución del programa lectura de las fotografías</i>	60
Figura 24 <i>Interfaz de android studio</i>	61

Figura 25	<i>Interfaz principal y lector de código QR</i>	61
Figura 26	<i>Interfaz de lectura de los datos del jugador, "Lectura del QR"</i>	62
Figura 27	<i>Interfaz de activación de cámara</i>	62
Figura 28	<i>Interfaz de reconocimiento de identidad del jugador</i>	63
Figura 29	<i>Pantalla principal del servicio de Python anyware</i>	63
Figura 30	<i>Despliegue en producción del backend, dominio para acceso de API REST</i>	64
Figura 31	<i>Configuración del entorno virtual para ejecutar el API REST</i>	64
Figura 32	<i>Instalación de librerías, clonación de proyecto git en consola</i>	65
Figura 33	<i>Las siguientes clases son principales para la administración</i>	65
Figura 34	<i>Clase que permite el reconocimiento de Código QR</i>	66
Figura 35	<i>Clase que permite el reconocimiento de Código QR</i>	66
Figura 36	<i>Clase para la toma de fotografía del rostro</i>	67
Figura 37	<i>Clase para el reconocimiento facial y verificación de resultados</i>	67
Figura 38	<i>Interfaz de reconocimiento facial Mashka Fútbol</i>	75
Figura 39	<i>Conexión adecuada para la aplicación</i>	76
Figura 40	<i>Reflejo de luz</i>	77
Figura 41	<i>Posición horizontal del teléfono</i>	78
Figura 42	<i>Características de la cámara de reconocimiento facial</i>	79
Figura 43	<i>Datos utilizados</i>	86
Figura 44	<i>Reflejo de luz</i>	87
Figura 45	<i>Posición horizontal del teléfono</i>	88
Figura 46	<i>Características del celular</i>	90

ÍNDICE DE TABLAS

Tabla 1 <i>Población y muestra</i>	34
Tabla 2 <i>Recolección de información</i>	35
Tabla 3 <i>Metodología XP</i>	38
Tabla 4 <i>Detalle de historia de usuario</i>	42
Tabla 5 <i>Historia de usuario 1</i>	43
Tabla 6 <i>Historia de usuario 2</i>	43
Tabla 7 <i>Historia de usuario 3</i>	44
Tabla 8 <i>Historia de usuario 4</i>	45
Tabla 9 <i>Interfaz administración del aplicativo usuario 5</i>	45
Tabla 10 <i>Interfaz administración del aplicativo de usuario 6</i>	46
Tabla 11 <i>Interfaz administración del aplicativo usuario 7</i>	46
Tabla 12 <i>Diseño de acceso al sistema</i>	47
Tabla 13 <i>Configuración de dispositivo</i>	48
Tabla 14 <i>Configuración de herramientas del aplicativo</i>	48
Tabla 15 <i>Conexión a la base de datos</i>	49
Tabla 16 <i>Visualización de la interfaz del aplicativo</i>	49
Tabla 17 <i>Visualización de registro de usuario</i>	50
Tabla 18 <i>Respuesta de datos jugadores</i>	50
Tabla 19 <i>Modelo para las pruebas de aceptación</i>	68
Tabla 20 <i>Prueba de aceptación usuario 1</i>	69
Tabla 21 <i>Prueba de aceptación usuario 2</i>	69
Tabla 22 <i>Prueba de aceptación usuario 3</i>	70
Tabla 23 <i>Prueba de aceptación usuario 4</i>	71
Tabla 24 <i>Prueba de aceptación usuario 5</i>	71

Tabla 25 <i>Prueba de aceptación usuario 6</i>	72
Tabla 26 <i>Prueba de aceptación usuario 7</i>	73
Tabla 27 <i>Pruebas de Tiempo de respuesta</i>	79
Tabla 28 <i>Tiempo de respuesta al reconocer a la persona</i>	83
Tabla 29 <i>Ambiente no controlado pruebas</i>	90
Tabla 30 <i>Ambiente no controlado pruebas</i>	94

Resumen

La presente monografía se enfoca en el avance de un sistema web, en cuanto al desarrollo de una aplicación móvil, utilizando Reconocimiento Facial, para el control de identidad y acceso de jugadores afiliados a la empresa Mashka futbol, la misma que no cuenta con un sistema informático que ayude a la revisión de los usuarios afiliados a la empresa. En la cual se implementará la metodología Extreme Programming (XP) , permitiendo el desarrollo del sistema en los tiempos establecidos según el cronograma presentado por el administrador. para esto el aplicativo web cuentan con funcionalidades como: registro de jugadores de forma sencilla eficaz para el proceso de guardar la información en la base de datos, detección de jugadores que estén sancionados por alguna circunstancia, ofreciendo un mejor servicio a los usuarios, el aplicativo móvil realiza el reconocimiento facial de cada usuario o jugador.

Lo más primordial del sistema es el Reconocimiento Facial puesto que verificar el acceso a jugadores afiliados a la empresa así como la inspección de los mismos que tengan dificultad en la base de datos. El sistema web se desarrollará con el lenguaje de programación Java, el cual es un lenguaje de programación orientado a objetos, que permite la creación de aplicaciones web y móviles. El uso de Java para el desarrollo de este sistema web, permitirá la creación de una aplicación móvil, que permitirá el control de identidad y acceso a una determinada área, mediante el uso de Reconocimiento Facial.

Palabras clave: programación web, aplicación móvil, metodología xp, lenguajes de programación, sistema web.

Abstract

This monograph focuses on the progress of a web system, in terms of the development of a mobile application, using Facial Recognition, for the control of identity and access of players affiliated to the company Mashka soccer, which does not have a computer system that helps the review of users affiliated to the company. In which the Extreme Programming (XP) methodology will be implemented, allowing the development of the system in the established times according to the schedule presented by the administrator. For this the web application has functionalities such as: registration of players in a simple and effective way for the process of saving the information in the database, detection of players who are sanctioned for any circumstance, offering a better service to users, the mobile application performs facial recognition of each user or player. The most primordial of the system is the Facial Recognition since to verify the access to players affiliated to the company as well as the inspection of the same that have difficulty in the database. The web system will be developed with the Java programming language, which is an object-oriented programming language that allows the creation of web and mobile applications. The use of Java for the development of this web system will allow the creation of a mobile application, which will allow the control of identity and access to a certain area, through the use of Facial Recognition.

Key words: web programming, mobile application, xp methodology, programming languages, web system.

Capítulo I

Introducción

Antecedentes

En la actualidad existen muchos aplicativos móviles que permite la gestión de asistencia de personas, los mismos funcionan bajo la gestión adecuada de los datos que a su vez está almacenados en una base datos encargada de registrar y mostrar datos al administrativo de la empresa permitiéndole controlar el acceso del personal que trabaja en dicha institución.

Contar con un sistema de control de asistencia eficiente y bien organizado de personas, puede garantizar el acceso adecuado a determinados sitios u organizaciones.

Dentro de las instituciones y organizaciones existen dispositivos como el registro de identidad y sistemas para el control de asistencia de personas como las aplicaciones de inteligencia artificial, específicamente por medio del Reconocimiento Facial que permiten gestionar adecuadamente la asistencia del personal que accede a determinado sitio.

Para llevar a cabo el Reconocimiento Facial existen dispositivos electrónicos que son utilizados con frecuencia para registrar y reconocer la identidad de personas por medio de fotografías, para posteriormente permitir el acceso a las empresas, edificios o instituciones (Lara et al., 2020).

El sistema de Reconocimiento Facial permite mejorar los procesos de seguridad tanto interna como externa de una organización, de esta manera se fomenta una imagen institucional generando perspectivas favorables, cautivando mejores procedimientos en beneficio productividad de la institución.

En la investigación Pereira et al., (2021) se ha desarrollado una aplicación web que permite verificar la utilización de la mascarilla protectora en los ambientes públicos, este proceso de detección utiliza el algoritmo de inteligencia artificial Haar Cascade que permite clasificar y organizar los rostros con y sin mascarillas protectoras, como resultado se obtiene

una aplicación permite detectar y registrar en la nube las imágenes de los rostros capturadas posibilitando un mayor análisis de estos datos utilizando técnicas de aprendizaje profundo.

Planteamiento de problema

En la empresa Mashka Fútbol, se identificó muchos problemas de acceso de usuarios, así como también inconsistencias en el control de personas ya que el personal de jugadores, que hacen uso de la plataforma de la empresa no se rige a esta cuando hacen uso de la información llevada a cabo, es decir no se cuenta con un control adecuado de las personas que ingresan hacer uso de la cancha de juego.

La falta de control en la Plataforma Mashka Fútbol provoca que los usuarios accedan al campo de juego con sanciones en su historial, también se genera mal uso de la información, así como también no se tiene un reporte diario de las personas que tienen impedimentos o sanciones en la asociación.

Es por lo que esta investigación se enfoca en el desarrollo de una aplicación móvil, utilizando Reconocimiento Facial, para el control de identidad y acceso de los jugadores afiliados a la empresa Mashka fútbol, en el cantón Latacunga.

Justificación

En la empresa Mashka Futbol se va a implementar el registro de acceso y datos de identidad que garantizará el acceso adecuado a los polideportivos, así como también se tendrá resultados sobre el registro de los jugadores que van a hacer uso de estos espacios.

La implementación del sistema para el control de acceso y control de identidad resulta de gran beneficio para la empresa Mashka Fútbol ya que concederá permiso de acceso fácil y rápido para hacer uso de las canchas deportivas además se podrá detectar la identidad de los jugadores que ingresan también de ser el caso se podrá evitar posibles suplantaciones de identidad de estos.

Objetivos

Objetivo general

- Implementar una aplicación móvil, utilizando Reconocimiento Facial, para el control de identidad y acceso de los jugadores afiliados a la empresa Mashka fútbol, en el cantón Latacunga.

Objetivos específicos

- Definir herramientas de desarrollo y herramientas complementarias para el desarrollo de la propuesta, además conocer los procesos que se realizan con los datos obtenidos por el dispositivo.
- Implementar técnicas de desarrollo de software en tecnologías de Reconocimiento Facial para el registro de identidad
- Ejecutar pruebas de funcionamiento y corrección de errores.

Alcance

El presente proyecto tiene como fin la creación de un aplicativo móvil, cumpliendo el rol de realizar el control de acceso de los miembros de la empresa Mashka futbol mediante el reconocimiento facial de cada uno de los socios administrativos y jugadores de dicha institución.

Dentro de la funcionabilidad de la aplicación se debe recalcar que con el acercamiento con la empresa, se realizara un levantamiento de requerimientos que la empresa solicitara antes de la creación del aplicativo.

Capítulo II

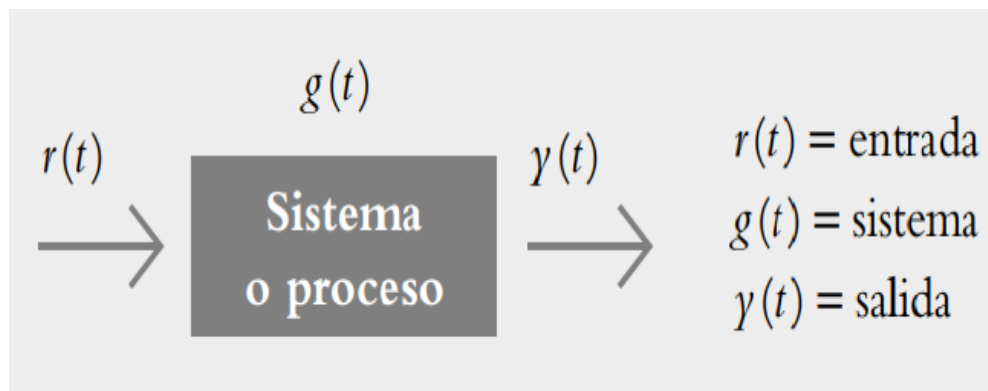
Marco Teórico

Sistemas de control

Los sistemas de control pueden llevarse a cabo mediante bloques, de modo que se integran varios elementos generando un sistema o proceso con señales de entrada $g(t)$, $r(t)$ que produce una salida $y(t)$, las entradas y la salida es una relación de causa y efecto con el sistema (Hernández, 2010).

Figura 1

Tareas de Reconocimiento Facial



Nota. En la imagen se representa el proceso que sigue las tareas de reconocimiento facial.

Tomado de (Hernández, 2010).

Control de asistencia

Se define como un conjunto de procedimientos y funciones mediante dispositivos que permiten garantizar el registro de personal en determinada organización o empresa, con la finalidad de gestionar el cumplimiento de las jornadas laborales, además permite identificar la interacción de las personas en momento determinado (Gavilánez, 2022).

Reconocimiento Facial

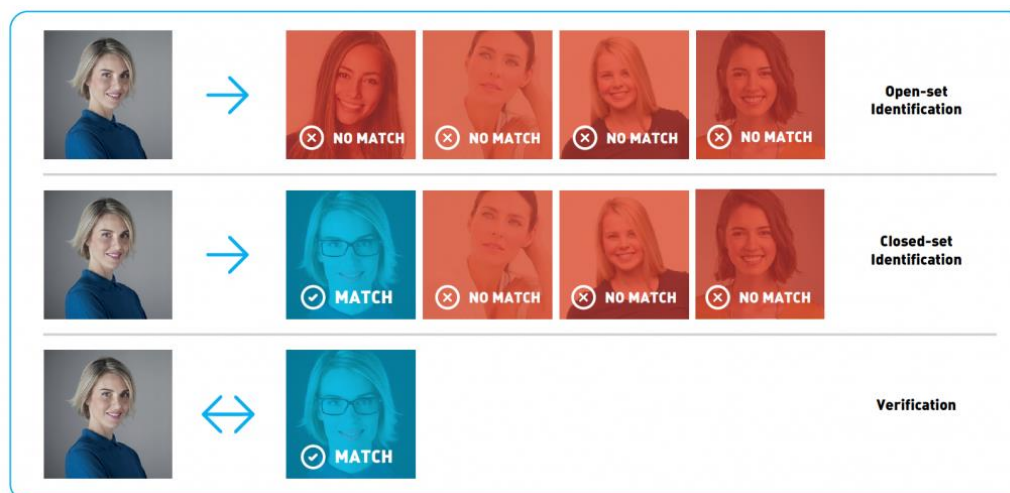
El Reconocimiento Facial se puede definir como un sistema de inteligencia artificial que permite identificar a una persona mediante la utilización de su fotografía o vídeo. Este método ha aparecido hacía varias décadas atrás, pero su acceso y utilización se ha evolucionado en los últimos años, porque actualmente se utilizan soluciones innovadoras, como aplicaciones de autenticación personal en dispositivos móviles (AWS, 2022).

Tareas del Reconocimiento Facial

El Reconocimiento Facial cumple con una serie de tareas o actividad, donde primeramente se puede detectar a una persona por medio de una fotografía o de un vídeo, también se puede identificar el género y la edad, además permite buscar a una persona en particular a partir de una base de datos de imágenes o a su vez permite comprobar si es la misma persona en diferentes imágenes (NTECHLAB, 2017).

Las tareas importantes son la identificación y verificación, para lo cual se inicia extrayendo de las imágenes de los rostros de las personas las características necesarias para el reconocimiento de caras. Los algoritmos de inteligencia artificial tienen la finalidad de identificar y encontrar el vector de características más cercano, y para la verificación utilizan un umbral simple y al final permite comparar los dos vectores para determinar si es la persona indicada.

Al ejecutar estas dos acciones se identifica a una persona entre un conjunto de imágenes de rostros o se determina que no existe esa persona de acuerdo con el conjunto de datos, esto es conocido como procedimiento de identificación de conjunto abierto (NTECHLAB, 2017).

Figura 2*Tareas de Reconocimiento Facial*

Nota. En la imagen se representa las tareas que realiza el reconocimiento facial. Tomado de (NTECHLAB, 2017).

Fases del Reconocimiento Facial

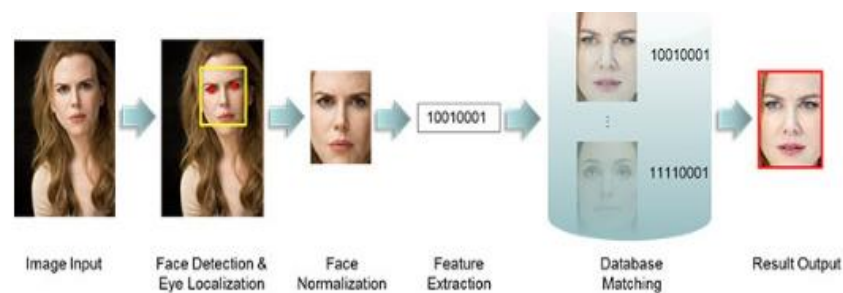
- **Detección de caras.** Existe varias imágenes de objetos que no son rostros de personas, por ende, esta fase se enfoca en localizar las caras que haya existentes en la imagen o fotografía y luego les marca con un rectángulo.
- **Alineamiento de la cara.** Conocida también como normalización porque ajusta varios elementos de la imagen como el tamaño, analítica o dimensiones con la finalidad de conseguir consistencia entre todas las imágenes que están almacenadas en la base de datos para que se puedan compararlas, permitiendo alcanzar una gran cantidad de atributos claves de la persona con la información que se encuentra en la base de datos.
- **Extracción de características.** Establece un conjunto de características que definen a una cara de una imagen determinada, permitiendo reducir a esas

características a una singularidad, dando como resultado un vector de características.

- **Reconocimiento del rostro.** Una vez obtenida la cara en el vector de características se procede a buscar en la base de datos una cara que presente las mismas características, luego se procede a comparar y obtener el reconocimiento con éxito (Gavilán, 2020)

Figura 3

Fases del Reconocimiento Facial



Nota. En la imagen se representa las fases que tiene el reconocimiento facial. Tomado de (Gavilán, 2020).

Cada una de estas fases utilizan diferentes algoritmos de inteligencia artificial, con frecuencia basados en *Deep Learning* mediante redes neuronales convolucionales que permiten hallar soluciones a problemas de control de personas en distintos lugares.

Redes Neuronales convolucionales

Es una arquitectura de red neuronal diferente que permite reducir significativamente la varianza y el número de pesos. En comparación a los algoritmos de Perceptron Multicapa (MLP), las redes neuronales convolucionales manipulan capas específicamente conectadas y no linealidades, introduciendo dos nuevos tipos de capas: capas convolucionales y agrupadas (Parada, 2021).

En la Inteligencia Artificial uno de los problemas clásicos es la identificación de imágenes, para el reconocimiento de imágenes pequeñas es suficiente las redes neuronales, sin embargo, para imágenes grandes se vuelve insuficiente la utilización de estos algoritmos.

Las redes neuronales convolucionales permiten solucionar este problema identificando algunas características espaciales de los inputs que permiten simplificar las arquitecturas de la red reduciendo, en gran medida, el número de variables de entrada. Por tanto, son especialmente útiles en problemas de visión por computador, y en particular, en el reconocimiento de objetos.

Las redes neuronales artificiales demuestran una serie de características que son parecidas a las del cerebro, es por ello por lo que son capaces de aprender de la rutina, de obtener casos anteriores y trasladar a nuevos casos, además extrae las características específicas partiendo de las entradas las cuales representan información irrelevante, la cual ofrece numerosas ventajas y además este tipo de tecnología se aplica en múltiples áreas de predicción de información.

Figura 4

Neurona Artificial



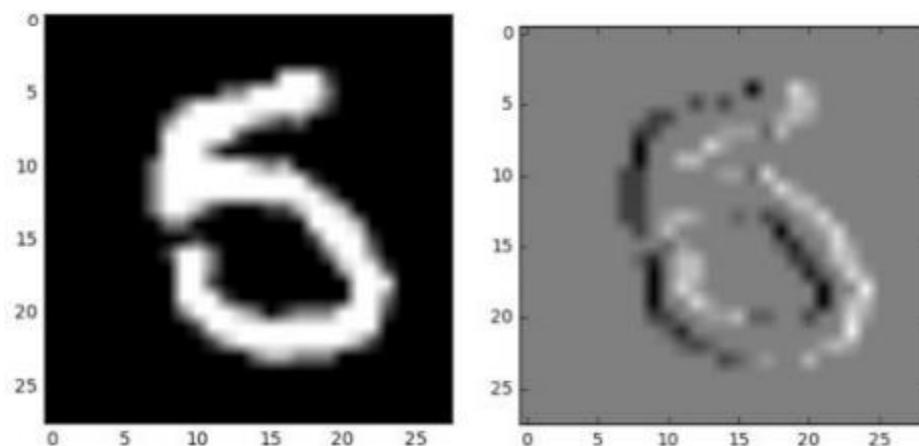
Nota. En la imagen se representa las características de la neurona artificial. Tomado de (Patiño et al, 2020).

Capas convolucionales

En inteligencia artificial las convoluciones nos ayudan extraer características, en un nivel bajo, todos los filtros pueden utilizarse para detectar las características y bordes en una imagen, y en un nivel alto, para lo cual consideran un filtro detector de borde horizontal simple, el cual genera una gran cantidad de valores negativos para entradas que el píxel izquierdo es brillante y el píxel derecho está oscuro; a la inversa, producirá grandes valores positivos para las entradas en las que el píxel izquierdo es oscuro y el píxel derecho es brillante (Inciso, 2021).

Figura 5

Resultado de convolucional un filtro de una imagen



Nota. En la imagen se representa el resultado de un filtro de una imagen. Tomado de (Inciso, 2021)

Tecnologías de desarrollo

Python.

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90 cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a Perl, pero con una sintaxis muy limpia y que favorece un código legible. Se

trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos (Gonzalez Duque, 2011).

Pythonanywhere

PythonAnywhere Proporciona un servidor para poder subir tu aplicación web a la nube. Es compatible con ´ proyectos de Django y existen dos modalidades: de pago y gratuita. La de pago tiene más ventajas y opciones como es la posibilidad de elegir el dominio, al contrario que la versión gratuita que te impone el dominio <http://yourusername.pythonanywhere.com/> (Martinez Rodriguez, 2021).

Open Cv

OpenCV (Open Source Computer Vision Library) es una librería orientada al procesamiento y análisis de imágenes. (Mínguez, 2021) Desarrollada inicialmente por Intel en 1999, ha ido ganando popularidad rápidamente, siendo hoy en día una de las más empleadas en el desarrollo de aplicaciones de visión artificial.

OpenCv es muy utilizada en grupos de investigación, organismos gubernamentales y empresas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda y Toyota. Sus ámbitos de aplicación son muchos, y entre ellos destacan:

- _ Reconocimiento facial
- _ Identificación de objetos o personas
- _ Inspección y vigilancia
- _ Juegos y controles

Librería Opencv

Las siglas Opencv provienen de los términos anglosajones “Open Source Computer Vision Library”. (Medrano, 23 de abril del 2008) Por lo tanto, Opencv es una librería de tratamiento de imágenes, destinada principalmente a aplicaciones de visión por computador en tiempo real.

Con este informe se pretende proveer de la descripción de algunas funciones de interés, así como exponer ejemplos completos y compilables de su uso, ya que esto facilitará en gran medida su posterior aplicación. También es necesario definir los principales tipos de datos que utiliza Opencv, y que están involucrados con el empleo de las funciones, la mayoría de estos tipos de datos son estructuras dinámicas.

Face Recognition

Esta librería es utilizada para reconocer y poder hacer la administración de rostros con la plataforma de desarrollo Python. Esta es una librería que se enmarca en la librería de Reconocimiento Facial Dlib, la cual que conserva un aprendizaje profundo o Deep Learning y puede alcanzar una precisión de 99,38 % (Inciso, 2021).

Framework

Un Framework Marín (2018) define que “es un término referente a un marco de trabajo o esquema que se establece y que se aprovecha para desarrollar y organizar un software determinado”. Por lo cual, a nivel de desarrollo, automatiza muchos procesos y además facilita el conjunto de la programación, siendo útil para evitar repeticiones del código para realizar funciones habituales en un rango de herramientas.

Conforme a esto, un Framework sirve para poder escribir código o desarrollar una aplicación de manera más sencilla, permitiendo una mejor organización y control de todo el código elaborado, así como una posible reutilización en el futuro. Debido a esto, garantiza una mayor productividad que los métodos convencionales y una minimización del coste al agilizar las horas de trabajo volcadas en el desarrollo. Por otra parte, su acción es algo que afecta también a los errores, minimizándolos considerablemente, y en definitiva, es algo que brinda

una ayuda general y más que considerable al programador y desarrollador, haciendo que sus labores sean mucho más sencillas.

Es importante considerar que un Framework es el esquema o estructura que se establece y que se aprovecha para desarrollar y organizar un software determinado. Esta definición, algo compleja, podría resumirse como el entorno pensado para hacer más sencilla la programación de cualquier aplicación o herramienta actual.

Gestión de bases de datos

En el estudio de Ferrer (2018) define al gestor de base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. De esta manera, está formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Todavía cabe señalar que cada base de datos se compone de una o más tablas que guarda un conjunto de datos y entre las principales características de los sistemas de base de datos se pueden mencionar: Independencia lógica y física de los datos, redundancia mínima, acceso concurrente por parte de múltiples usuarios, integridad de los datos, consultas complejas optimizadas, seguridad de acceso y auditoría, respaldo y recuperación, Acceso a través de lenguajes de programación estándar y el Sistema de Gestión.

Extreme Programming (XP)

Concebida para proyectos de pequeño y medio tamaño, prevé participación activa y recuente con el cliente (Fuentes, 2015). Describe detalladamente las prácticas de desarrollo que se van a utilizar como, por ejemplo: Lenguajes de programación, refactorización, pruebas unitarias, etc.

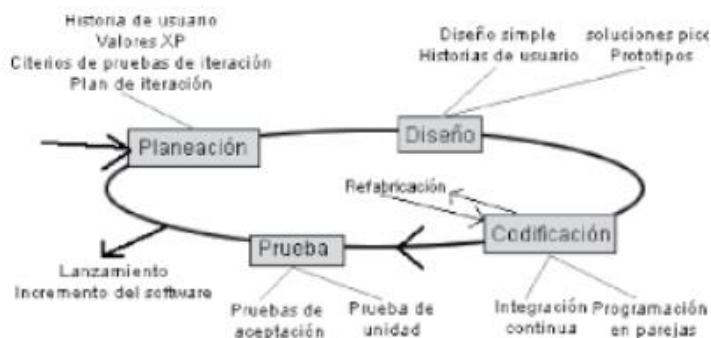
El proceso de XP

El proceso de XP al igual que RUP y MSF se presenta en fases, en XP se ejecuta en cuatro fases teniendo presente los principios y valores antes mencionados, los cuales son un eje fundamental para el correcto desarrollo de cada fase durante el ciclo.

Al igual que en las metodologías antes mencionadas (A., 1 de abril de 2011, Aprobado el 10 de junio de 2011) en el proceso XP, se observan una serie de fases que al ser concluidas dan origen a una versión del producto software, y cada versión es un ciclo, el cual hace parte del ciclo de vida del software. Al no tener más ciclos a ejecutar se entiende que los sistemas han cumplido con su objetivo, en caso contrario se deben seguir desarrollando ciclos para agregar la funcionalidad deseada. Cada fase del ciclo comprende lo siguiente:

Figura 6

Proceso de metodología XP



Nota. En la imagen se representa los procesos de la metodología XP. Tomado de (Pressman y Murrieta,2020)

Fase de planeación: Esta fase inicia con las historias de usuario que describen las características y funcionalidades del software. El cliente asigna un valor o prioridad a la historia, los desarrolladores evalúan cada historia y le asignan un costo el cual se mide en semanas de desarrollo.

Fase de diseño: El proceso de diseño debe procurar diseños simples y sencillos para facilitar el desarrollo. Se recomienda elaborar un glosario de términos y la correcta especificación de métodos y clases para facilitar posteriores modificaciones, ampliaciones o reutilización del código. Anteriormente este proceso se apoyaba en el uso de tarjetas CRC (Colaborador-Responsabilidad-Clase) la cual identifica las clases orientadas a objetos que son relevantes para el incremento del software.

Fase de codificación: En esta fase los desarrolladores deben diseñar las pruebas de unidad que ejercitarán cada historia de usuario. Después de tener las pruebas, los desarrolladores trabajarán en parejas para concentrarse en lo que debe implementarse para pasar la prueba de unidad.

Fase de pruebas: Las pruebas de unidad deben implementarse con un marco de trabajo que permita automatizarlas, con la finalidad de realizar pruebas de integración y validación diarias, esto proporcionará al equipo un indicador del progreso y revelarán a tiempo si existe alguna falla en el sistema. Las pruebas (A., 1 de abril de 2011, Aprobado el 10 de junio de 2011) tienen las siguientes características:

- Desarrollo previamente aprobado: Significa que primero se escriben las pruebas y luego el código. Las pruebas deben simular el envío de la entrada a probar y deben verificar que el resultado cumpla con las especificaciones de salida.

- Desarrollo de pruebas incremental: Los requerimientos del usuario se expresan como historias, el equipo de desarrollo evalúa cada historia y la divide en tareas. Cada una representa una característica distinta del sistema y se pueden diseñar pruebas de unidad para esa tarea.

- Participación del usuario en el desarrollo de las pruebas: El usuario ayuda a desarrollar las pruebas de aceptación, las cuales son pruebas que se implementan con los datos reales del cliente para verificar el cumplimiento real de sus necesidades.

- Uso de bancos de pruebas automatizados: Se debe usar un sistema que envíe a ejecución las pruebas automatizadas y de esta forma probar constantemente el sistema software.

Metodología

Modalidad de investigación.

La investigación es considerada bibliografía ya que se apoyará en libros, revistas, documentación sobre procesos de reconocimiento biométrico.

Modalidad de campo.

La investigación es considerada de campo porque se interactúa en el origen del problema.

Tabla 1

Población y muestra

Población	Numero	Porcentaje
Parroquias que estén afiliadas en la empresa	5	100%
Total	5	100%

Nota. Esta tabla muestra la población de la empresa con la que se trabajara.

La población total no excede a los 100 individuos por lo tanto se trabajará con el total de su población, es prescindible tener una muestra representativa.

Recolección de información

En la recolección de información se utilizará el siguiente cuestionario:

Tabla 2*Recolección de información*

Interrogantes	Explicación
¿Para qué?	Añadir valores funcionales para el control de identidad y acceso a jugadores
¿De qué personas u objetos?	Usuarios que sean afiliados de la Empresa Mashka Futbol
¿Sobre qué Aspectos?	Reconocimiento Facial. Gestión de datos obtenidos.
¿Quién, Quienes?	Investigadores: Jomaira Gualsaqui, Stalin Vilca
¿Cuándo?	Semestre académico
¿Dónde?	Empresa Mashka Futbol
¿Qué técnica de recolección?	Investigación
¿Con qué?	Libros, Revistas
¿En qué situación?	Condiciones normales.

Nota. La tabla representa la información recolectada de la empresa que se necesita para la aplicación.

Procesamiento y análisis de datos

En el procesamiento y análisis de información de los valores agregados al sistema de reconocimiento de identidad y acceso de jugadores, la misma que es analizado por el jefe de la empresa mashka futbol, quien conoce con claridad el funcionamiento del sistema existente y los procesos que se han llevado a cabo ejecutando para el control de identidad y acceso de los jugadores. Además, es quien propone los cambios que se realizaran en dicho sistema

Desarrollo del Proyecto

Las actividades que se realizarán son basadas en la metodología Xtreme Programming (XP) donde se realizará las siguientes fases:

Fase 1. Planificación

- Modelado de procesos.
- Estimación de tiempo.

Fase 2. Diseño

- Tarjetas Clase, Responsabilidad, Colaboradores Mashka Futbol

Fase 3. Codificación

- Arquitectura del proyecto
- Modelo de la base de datos
- Instalación y codificación de herramientas para el desarrollo.
- Diseño en la interfaz.
- Desarrollo de software.

Fase 4. Pruebas

- Revisión de funcionalidad del aplicativo

Fase 5. Lanzamiento

- Implementación del sistema web

Capítulo III

Desarrollo y resultados

Análisis de resultados

XP

XP es la primera metodología que impulsa las metodologías ágiles consta de cuatro variables: costo, tiempo, calidad y alcance, propone una clara distinción entre las etapas del proyecto de software.

Roles XP

Programador: Es quien produce el código del sistema y describe las pruebas unitarias.

Cliente: Escribe las historias de usuario y las pruebas funcionales para la implementación y validación.

Tester: Ayuda al cliente con las pruebas funcionales y las ejecuta regularmente.

Encargado del seguimiento: Proporciona retroalimentación al equipo. Analiza y verifica el grado de acierto de las estimaciones realizadas.

Entrenador: Provee al equipo de guías para que las prácticas XP se cumplan de manera correcta.

Consultor: Persona externa del equipo con conocimiento específico a quien se acude en caso de que surjan problemas.

Gestor: Persona que crea vínculo entre clientes y programadores

Procesos XP

El ciclo de desarrollo consiste en:

1. El cliente establece el valor de negocio.
2. El programador estima el esfuerzo necesario para la implementación.
3. El cliente define que construir de acuerdo con el tiempo y prioridades.
4. El programador construye el valor negocio.

5. Volver al paso 1

Método escogido

Tabla 3

Metodología XP.

XP	
Cultura de negocio	Operativo y colaborativo
Tamaño de proyecto	
Tamaño de equipo	1 equipo
Entorno Físico	Equipo en un mismo lugar
Modo de desarrollo	Iterativo y rápido

Nota. La tabla representa la metodología escogida para el proyecto.

Se optó a la metodología XP, debido a que esta metodología da prioridad a las necesidades del cliente permitiendo incorporarlo como si fuera un miembro del equipo, también permite modificar o adicionar funcionalidades no detectadas en etapas primeras de definición del producto esto quiere decir que permite una mejor adaptación en el proyecto.

Para el desarrollo de la metodología del proyecto actual se toman en cuenta las siguientes fases:

- Fase 1. Planificación.
- Fase 2. Diseño.
- Fase 3. Codificación.
- Fase 4. Pruebas.
- Fase 5. Lanzamiento.

Análisis de herramientas de desarrollo

Android Studio

Android es un sistema operativo al igual que Windows, Linux, etc. (Castillo, 2019) Por lo tanto, tiene el control total del dispositivo que lo contiene, así que cuando desarrollamos una aplicación, estamos desarrollando para el sistema operativo y podemos tener control de ciertos elementos que generar una experiencia más interesante y agradable para el usuario si lo hacemos de buena forma, en caso contrario podríamos estar generando unas de las peores experiencias de usuario y afectando el rendimiento del dispositivo.

Visual code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. VS Code tiene una buena integración con Git, cuenta con soporte para depuración de código, y dispone de un sinnúmero de extensiones, que básicamente te da la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación. (Flores, 2022)

Python

Python es un lenguaje de programación multiplataforma, consistente y maduro, en el cual confían con éxito las empresas y organizaciones más prestigiosas a nivel mundial, Son múltiples los campos en los que se utiliza: aplicaciones web, juegos y multimedia, interfaces gráficas, networking, aplicaciones científicas, inteligencia artificial, programación de sistemas, entre otros. (Buttu, 2014)

Java

Java proporciona mecanismos propios para especificar e implementar aplicaciones distribuidas. Java representa en este contexto una gran ventaja ya que no hay que usar librerías externas del lenguaje. Java es considerado como el lenguaje de Internet debido al soporte excelente que da al desarrollo de aplicaciones distribuidas. Además, Java incorpora

mecanismos propios de seguridad, una propiedad fundamental en las aplicaciones distribuidas basadas en Internet. (Xhafa, 2066)

API Rest Python

Existen diferentes formas de crear una API en Python, siendo las más utilizadas FastAPI y Flask. Así pues, explicaré cómo funcionan las dos, de tal forma que puedas usar la forma de crear APIs en Python que más te guste. Empecemos con FastAPI. Las API son el conjunto de protocolos o reglas que interactúan en las aplicaciones, permite a los desarrolladores acceder a los recursos de otro software sin problema. Normalmente se utilizan como herramientas públicas para software de terceros siendo estas las aplicaciones móviles. (Jauregui., 2018)

JSON

Json (javaScripts Object Notation o notación de objeto de JavaScript), se comienza a trabajar con la sintaxis del estándar, para lograr que una aplicación se comunique con un servidor y genere información aprovechable por distintas herramientas web, como navegadores, base de datos y también motores de búsqueda, con el fin de mejorar la calidad de nuestra aplicación. (Aguirre, 2020)

Desarrollo de la propuesta

La metodología para utilizarse para el desarrollo del proyecto es XP, en la que nos permite cambios repentinos en cualquier etapa de desarrollo, permite la entrega de funcionales adjudicado un producto de calidad.

Fase 1. Planificación

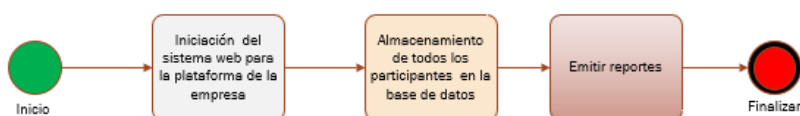
Modelo de procesos

En esta fase de proyecto se realizó a modelar los procesos que mantiene el sistema de la empresa Mashka Futbol y la propuesta para mejorar los procesos implementando nuevas tecnologías.

Procesos generales

Figura 7

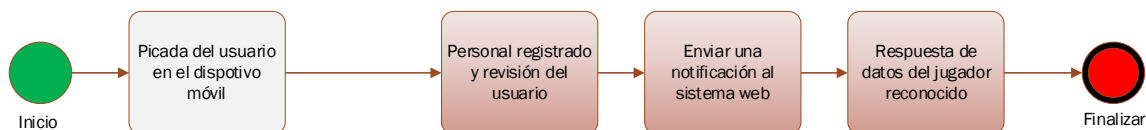
Proceso actual que tiene la Empresa Mashka Futbol



Nota. En la imagen se representa el proceso actual que tiene la empresa.

Figura 8

Propuesta para el proceso de control de identidad Mashka Futbol



Nota. En la imagen se representa una propuesta para el control de identidad.

Historia de usuarios.

En esta fase es necesario establecer historias de usuario que describan de manera clara los requerimientos del sistema a desarrollar, dichas historias deberán estar descritas en lenguaje natural para mayor comprensión.

A continuación, se describen las especificaciones que son tomadas en cuenta para las historias de usuario:

Usuario. – Persona que detalla la historia.

Número. – Número asignado a la historia.

Nombre de la historia. – Título asignado a la historia.

Prioridad de negocio. – Cuantificador que asigna la necesidad del negocio: alta, media, baja.

Riesgo de desarrollo. – Cuantificador basado en el impacto de desarrollo en el caso de fallo: alto, medio, bajo.

Iteración asignada. – Iteración asignada a la historia

Puntos estimados. – Lapso en días para concluir con la historia, días enteros y aproximados al inmediato superior.

Descripción. – Detalle en lenguaje natural en respuesta a la pregunta ¿Qué hacer?

Observaciones. – Detalles específicos para hacer que la historia se cumpla.

Programador responsable. – Persona encargada del desarrollo de la historia

Tabla 4

Detalle de historia de usuario

Historia de usuario	
Usuario:	N°:
Nombre de la historia:	
Prioridad de negocio:	Riesgo de desarrollo:
Iteración asignada:	Puntos estimados:
Descripción:	
Observaciones:	
Programador responsable:	

Nota. La tabla representa la descripción que llevara las historias de usuarios.

Historia de usuario

Se redactan las historias de usuarios que cumplirán con el modelo de los procesos redactados en el punto anterior.

Tabla 5

Historia de usuario 1.

Historia de usuario	
Usuario: administrador	N°: 1
Nombre de la historia: Acceso al sistema	
Prioridad de negocio: media	Riesgo de desarrollo: alto
Iteración asignada: 1	Puntos estimados: 2
Descripción: Pantalla que muestre el reconocimiento de código QR para la autenticación	
Observaciones: La pantalla debe contener el campo de reconocer QR	
Programador responsable: Jomaira Gualsaqui, Stalin Vilca	

Nota. La tabla representa los detalles de la historia de usuario 1.

Tabla 6

Historia de usuario 2.

Historia de usuario	
Usuario: administrador	N°: 2
Nombre de la historia: Configuración del aplicativo	
Prioridad de negocio: alta	Riesgo de desarrollo: alto

Historia de usuario

Iteración asignada: 1

Puntos estimados: 5

Descripción: Configuración del aplicativo para identificación de rostros

Historia de usuario

Observaciones: Se descargan librerías para el aplicativo, la cual es pyhton face recognition

Programador responsable: Jomaira Gualsaqui, Stalin Vilca

Nota. La tabla muestra los detalles del usuario 2.

Tabla 7

Historia de usuario 3.

Historia de usuario

Usuario: administrador

N°:3

Nombre de la historia: Configuración de herramienta del aplicativo

Prioridad de negocio: alta

Riesgo de desarrollo: alto

Iteración asignada: 2

Puntos estimados: 3

Descripción: El usuario puede tomarse una foto para la verificación de datos

Observaciones: Ninguna

Programador responsable: Jomaira Gualsaqui, Stalin Vilca

Nota. La tabla representa los detalles de usuario 3.

Tabla 8*Historia de usuario 4.*

Historia de usuario	
Usuario: administrador	N°: 4
Nombre de la historia: Conexión a la base de datos	
Prioridad de negocio: alta	Riesgo de desarrollo: alto
Iteración asignada: 2	Puntos estimados: 3
Descripción: Se utilizará la base de datos JSON ya generada	
Observaciones: Se añadirán única mente datos necesarios	
Programador responsable: Jomaira Gualsaqui, Stalin Vilca	

Nota. La tabla representa los detalles de usuario 4.**Tabla 9***Interfaz administración del aplicativo usuario 5.*

Historia de usuario	
Usuario: administrador	N°: 5
Nombre de la historia: Visualización de la interfaz del aplicativo	
Prioridad de negocio: alta	Riesgo de desarrollo: alto
Iteración asignada: 2	Puntos estimados: 3
Descripción: Se desarrolla la interfaz gráfica de cómo será el aplicativo	
Observaciones: Ninguna	
Programador responsable: Jomaira Gualsaqui, Stalin Vilca	

Nota. La tabla muestra los detalles de interfaz de administración del aplicativo de usuario 5.

Tabla 10

Interfaz administración del aplicativo de usuario 6.

Historia de usuario	
Usuario: administrador	N°: 6
Nombre de la historia: Verificación de registro de los usuarios	
Prioridad de negocio: alta	Riesgo de desarrollo: alto
Iteración asignada: 2	Puntos estimados: 3
Descripción: Se verificará a los usuarios en la base de datos posteriormente de la verificación del QR	
Observaciones: Ninguna	
Programador responsable: Jomaira Gualsaqui, Stalin Vilca	

Nota. La tabla muestra los detalles de interfaz de administración del aplicativo de usuario 6.

Tabla 11

Interfaz administración del aplicativo usuario 7.

Historia de usuario	
Usuario: administrador	N°: 7
Nombre de la historia: Respuesta de datos del jugador	
Prioridad de negocio: alta	Riesgo de desarrollo: alto
Iteración asignada: 2	Puntos estimados: 3

Historia de usuario

Descripción: El usuario puede consultar sus datos en la página luego de la detección del rostro

Observaciones: Ninguna

Programador responsable: Jomaira Gualsaqui, Stalin Vilca

Nota. La tabla muestra los detalles de interfaz de administración del aplicativo de usuario.

Fase 2. Diseño

Las tarjetas de identidad utilizadas en la empresa según los datos de los usuarios son las siguientes

Tabla 12

Diseño de acceso al sistema.

Acceso al sistema

Responsabilidad	Colaboradores Mashka Futbol
parámetros del usuario	Capa de acceso de datos
	Métodos de validación de datos

Observaciones:

Nota. La tabla muestra la descripción del acceso al sistema que se necesita para el diseño.

Tabla 13*Configuración de dispositivo*

Configuración del aplicativo	
Responsabilidad	Colaboradores Mashka Futbol
Obtener parámetros	Librerías Python y Open CV
Validar parámetros	Métodos de validación de datos
Configuración del aplicativo	
Observaciones:	

Nota. En la siguiente tabla se especifica los parámetros de la configuración de la aplicación.

Tabla 14*Configuración de herramientas del aplicativo*

Configuración de herramientas del aplicativo	
Responsabilidad	Colaboradores Mashka Futbol
Validar parámetros	Librerías Python y Open CV
	Métodos de validación de datos
Observaciones:	

Nota. En la siguiente tabla se detalla la configuración de las herramientas que contiene el aplicativo.

Tabla 15*Conexión a la base de datos*

Conexiones a la base de datos	
Responsabilidad	Colaboradores Mashka Futbol
Obtener parámetros	Librerías Python y Open CV
Validar parámetros	Métodos de validación de datos
Guardar parámetros	
Observaciones:	

Nota. En la siguiente tabla se describe la conexión a la base de datos de la app.

Tabla 16*Visualización de la interfaz del aplicativo*

Visualización de la interfaz del aplicativo	
Responsabilidad	Colaboradores Mashka Futbol
Obtener parámetros	Capa de acceso de datos
Observaciones:	

Nota. En la siguiente tabla se detalla la visualización de la interfaz gráfica del aplicativo.

Tabla 17*Visualización de registro de usuario*

Visualización de registro de usuario	
Responsabilidad	Colaboradores Mashka Futbol
Obtener parámetros	Capa de acceso de datos
Validar parámetros	Métodos de validación de datos
Guardar parámetros	
Observaciones:	

*Nota. En la siguiente tabla se describe la visualización de registro de usuario.***Tabla 18***Respuesta de datos jugadores*

Respuesta de datos jugadores	
Responsabilidad	Colaboradores Mashka Futbol
Obtener parámetros	Capa de acceso de datos
Validar parámetros	Métodos de validación de datos
Guardar parámetros	Notificación de Respuesta
Observaciones:	

*Nota. En la siguiente tabla se describe la respuesta de los datos de jugadores.***Fase 3. Codificación**

En esta fase se procede al desarrollo del sistema cumpliendo con lo expuesto con los requerimientos solicitados y de acuerdo con las iteraciones de la metodología XP.

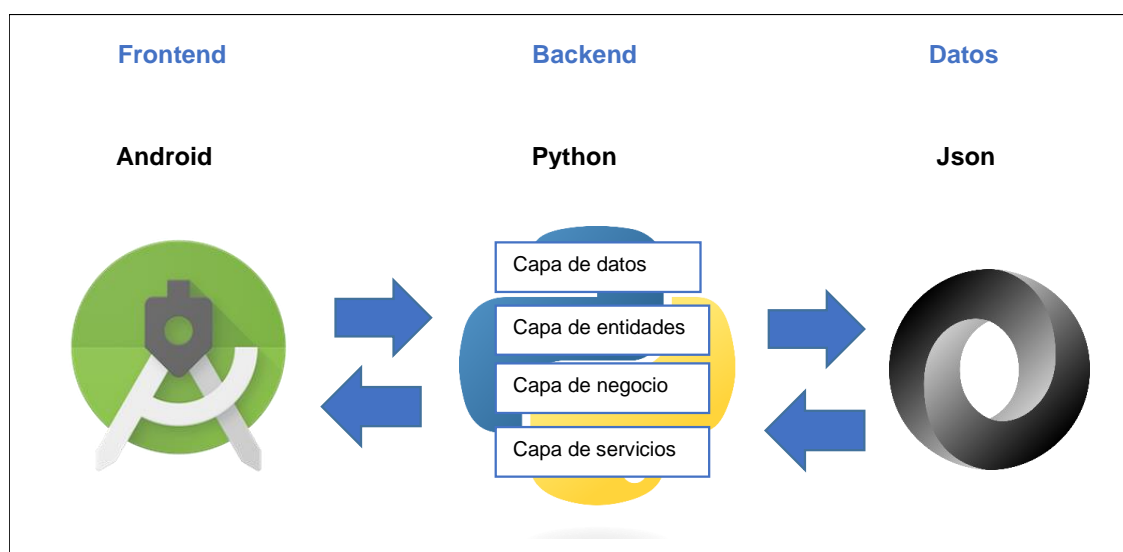
Arquitectura del proyecto.

Después de establecer las tecnologías que se van a utilizar para el desarrollo del diseño se procede a definir la arquitectura sobre la cual se va a desarrollar el sistema.

Esta es la arquitectura establecida, sobre la cual trabaja la empresa Mashka Fútbol.

Figura 9

Arquitectura del proyecto



Nota. En esta imagen se representa la arquitectura del proyecto de la aplicación.

La edificación será cliente servidor ya que la aplicación Web al tener la característica de ser adaptable tiene como clientes varios dispositivos con acceso a la red que se comunica con el servidor mediante una API para intercambiar la información.

La edificación predeterminada por parte para el desarrollo de la API en Mashka Fútbol son las siguientes capas:

- Capa de datos que permite la comunicación y gestión con la base de datos.
- Capa entidades aloja a todos los objetos (clases) que se utilizaran en la paca de negocio.

- Capa de negocio se encarga de implementar la lógica, es decir todo lo que el aplicativo web deba considerar antes de realizar la acción.
- Capa de servicios es la capa que se implementa los métodos de la capa de negocio, para crear los servicios.

Instalación y configuración de herramientas de desarrollo

Open CV

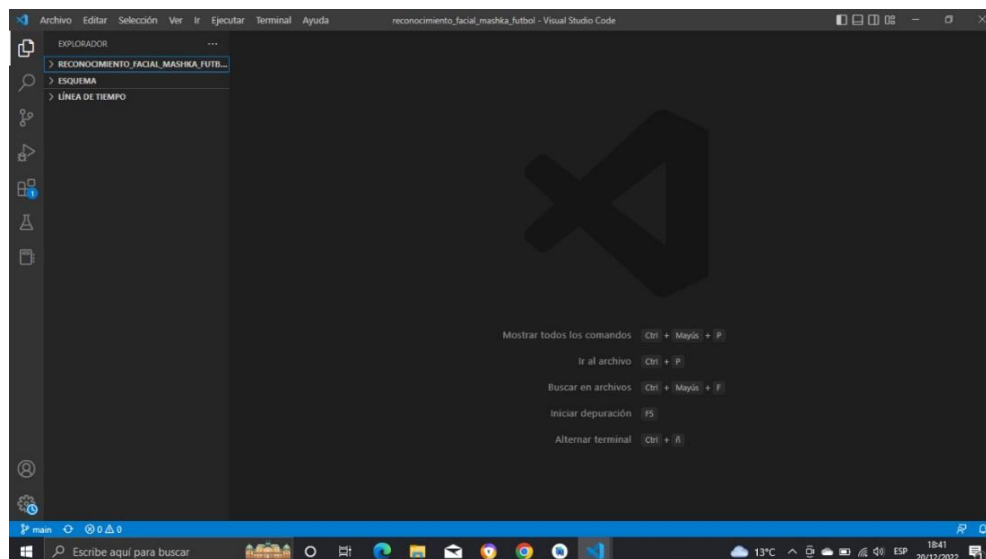
En esta herramienta se instaló sus librerías, se utilizará para la creación del reconocimiento facial.

Visual Code

Visual Studio Code permite gestionar propios atajos, así como refactorizar el código, además es de código abierto, permite la descarga de extensiones que favorece sus funcionalidades, Por lo tanto, es el editor de código para realizar el face recognition del sistema

Figura 10

Interfaz principal de visual code



Nota. En la imagen se representa la interfaz gráfica de visual code con sus funcionalidades.

Creación para el registro de rostros.

Instalación a través de cmd la librería face recognition models

Figura 11

Instalación de librerías a través del símbolo del sistema

```

Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.2130]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\HP Stalin>cd Desktop\probando_face_recognition
El sistema no puede encontrar la ruta especificada.

C:\Users\HP Stalin>python --version
Python 3.10.8

C:\Users\HP Stalin>pip install virtualenv
Collecting virtualenv
  Downloading virtualenv-20.16.5-py3-none-any.whl (8.8 MB)
----- 8.8/8.8 MB 5.3 MB/s eta 0:00:00
Collecting distlib<1,>=0.3.5
  Downloading distlib-0.3.6-py2.py3-none-any.whl (468 kB)
----- 468.5/468.5 kB 3.7 MB/s eta 0:00:00
Collecting platformdirs<3,>=2.4
  Downloading platformdirs-2.5.2-py3-none-any.whl (14 kB)
Collecting filelock<4,>=3.4.1
  Downloading filelock-3.8.0-py3-none-any.whl (10 kB)
Installing collected packages: distlib, platformdirs, filelock, virtualenv
Successfully installed distlib-0.3.6 filelock-3.8.0 platformdirs-2.5.2 virtualenv-20.16.5

[notice] A new release of pip available: 22.2.2 -> 22.3
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\HP Stalin>pip install virtualenv_

```

Nota. La figura representa los comandos que se utilizó para la instalación de la librería face reconigition models.

Instalacion mediante CMD la librería opencv-contrib-python

Figura 12

Instalación de la librería opencv

The image shows two side-by-side screenshots of Windows command prompts. The left window shows the installation of opencv-contrib-python and its dependencies (numpy, pip) in a virtual environment. The right window shows the activation of the virtual environment and the installation of the imutils library.

```

[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>
[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>pip freeze
click==8.1.3
colorama==0.4.5
dlib==19.24.0
face-recognition==1.3.0
face-recognition-models==0.3.0
numpy==1.23.4
pillow==9.2.0

[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>pip install opencv-contrib-python
Collecting opencv-contrib-python
  Using cached opencv_contrib_python-4.6.0.66-cp36-abi3-win_amd64.whl (42.5 MB)
Requirement already satisfied: numpy>=1.14.5 in c:\users\hp stalin\desktop\reconocimiento_facial\fr_venv\lib\site-packages (from opencv-contrib-python) (1.23.4)
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-4.6.0.66

[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\hp stalin\desktop\reconocimiento_facial\fr_venv\lib\site-packages (22.2.2)
Collecting pip
  Using cached pip-22.3-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.2.2
    Uninstalling pip-22.2.2:
      Successfully uninstalled pip-22.2.2
  Successfully installed pip-22.3

[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>pip freeze
click==8.1.3
colorama==0.4.5
dlib==19.24.0
face-recognition==1.3.0
face-recognition-models==0.3.0
numpy==1.23.4
opencv-contrib-python==4.6.0.66
pillow==9.2.0

[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>_
  
```

Nota. La figura representa los comandos para la instalación de la librería opencv contrib Python.

Figura 13

Instalación de la librería opencv

The image shows a single screenshot of a Windows command prompt window. It displays the installation of opencv-contrib-python and its dependencies (numpy, pip) in a virtual environment. The output shows the successful installation of opencv-contrib-python and the upgrade of pip to the latest version (22.3).

```

[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>
[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>pip freeze
click==8.1.3
colorama==0.4.5
dlib==19.24.0
face-recognition==1.3.0
face-recognition-models==0.3.0
numpy==1.23.4
pillow==9.2.0

[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>pip install opencv-contrib-python
Collecting opencv-contrib-python
  Using cached opencv_contrib_python-4.6.0.66-cp36-abi3-win_amd64.whl (42.5 MB)
Requirement already satisfied: numpy>=1.14.5 in c:\users\hp stalin\desktop\reconocimiento_facial\fr_venv\lib\site-packages (from opencv-contrib-python) (1.23.4)
Installing collected packages: opencv-contrib-python
Successfully installed opencv-contrib-python-4.6.0.66

[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>python.exe -m pip install --upgrade pip
Requirement already satisfied: pip in c:\users\hp stalin\desktop\reconocimiento_facial\fr_venv\lib\site-packages (22.2)
Collecting pip
  Using cached pip-22.3-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 22.2.2
    Uninstalling pip-22.2.2:
      Successfully uninstalled pip-22.2.2
  Successfully installed pip-22.3

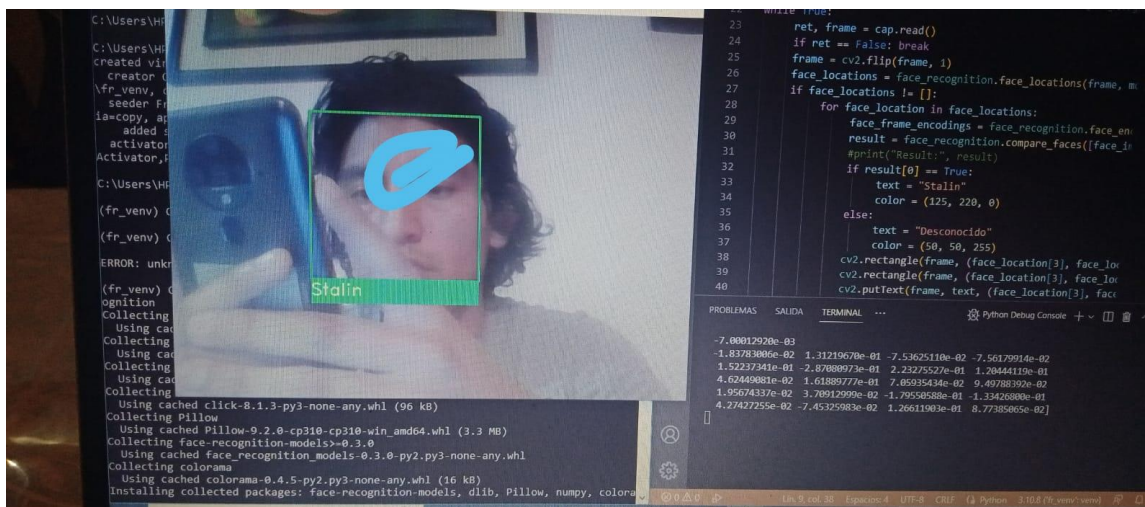
[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>
[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>pip freeze
click==8.1.3
colorama==0.4.5
dlib==19.24.0
face-recognition==1.3.0
face-recognition-models==0.3.0
numpy==1.23.4
opencv-contrib-python==4.6.0.66
pillow==9.2.0

[fr_venv] C:\Users\HP Stalin\Desktop\Reconocimiento_Facial>_
  
```

Nota. En la figura se representa la instalación de la librería opencv

Figura 14

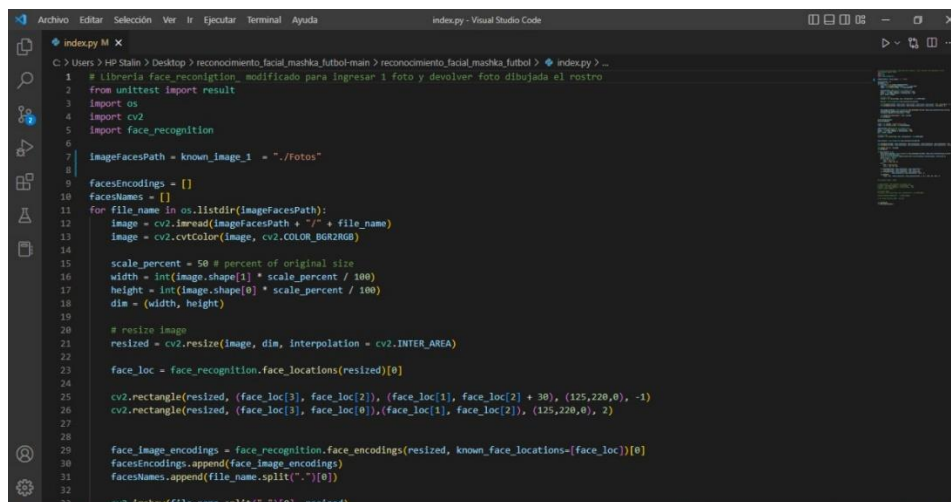
Resultados de la instalación de la librería



Nota. En la figura de muestra cómo se reconoce el rostro.

Figura 15

Codificación de reconocimiento de imágenes

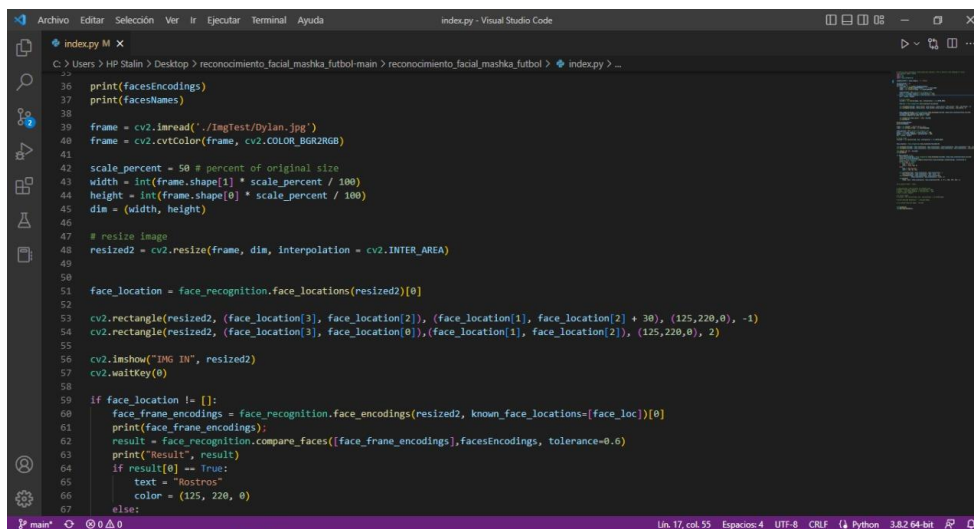


Nota. En la imagen se representa la programación de reconocimiento de imágenes.

Utilizamos **face_recognition** para detectar los rostros de imagen, de igual forma agregamos coordenadas x. y.

Figura 16

Librería face recognition



```

index.py M X
C:\Users\HP Stalin\Desktop> reconocimiento_facial_mashka_futbol-main> reconocimiento_facial_mashka_futbol > index.py > ...
36 print(facesEncodings)
37 print(facesNames)
38
39 frame = cv2.imread('./imgTest/Dylan.jpg')
40 frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
41
42 scale_percent = 50 # percent of original size
43 width = int(frame.shape[1] * scale_percent / 100)
44 height = int(frame.shape[0] * scale_percent / 100)
45 dim = (width, height)
46
47 # resize image
48 resized2 = cv2.resize(frame, dim, interpolation = cv2.INTER_AREA)
49
50
51 face_location = face_recognition.face_locations(resized2)[0]
52
53 cv2.rectangle(resized2, (face_location[3], face_location[2]), (face_location[1], face_location[2] + 30), (125,220,0), -1)
54 cv2.rectangle(resized2, (face_location[3], face_location[0]),(face_location[1], face_location[2]), (125,220,0), 2)
55
56 cv2.imshow("IMG IN", resized2)
57 cv2.waitKey(0)
58
59 if face_location != []:
60     face_frame_encodings = face_recognition.face_encodings(resized2, known_face_locations=[face_loc])[0]
61     print(face_frame_encodings)
62     result = face_recognition.compare_faces([face_frame_encodings],facesEncodings, tolerance=0.6)
63     print("Result:", result)
64     if result[0] == True:
65         text = "Rostrós"
66         color = (125, 220, 0)
67     else:

```

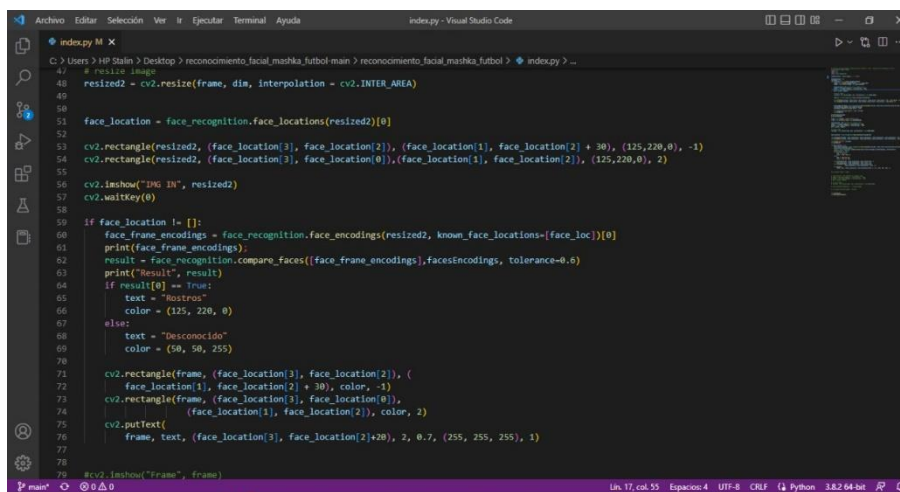
Nota. En la imagen se representa el uso y la función de la librería face recognition.

Para generar un vector de características de 128 elementos para cada rostro

face_encodings.

Figura 17

Librería face encodings



```

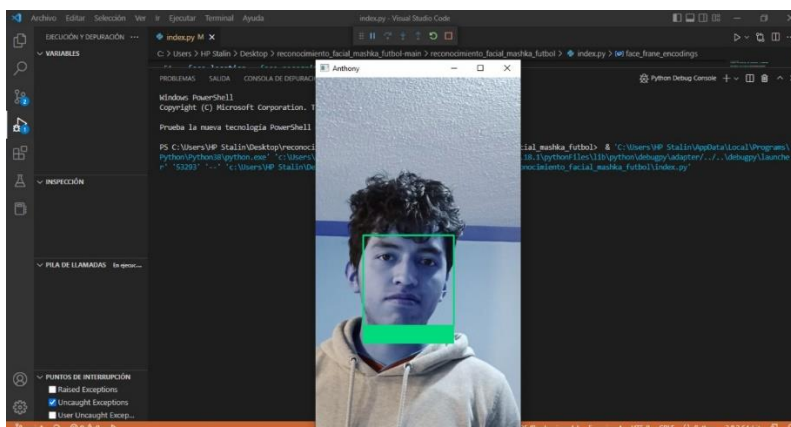
index.py - Visual Studio Code
index.py M X
C:\Users\HP Stalin\Desktop> reconocimiento_facial_mashka_futbol-main> reconocimiento_facial_mashka_futbol > index.py > ...
47 # resize image
48 resized2 = cv2.resize(frame, dim, interpolation = cv2.INTER_AREA)
49
50
51 face_location = face_recognition.face_locations(resized2)[0]
52
53 cv2.rectangle(resized2, (face_location[3], face_location[2]), (face_location[1], face_location[2] + 30), (125,220,0), -1)
54 cv2.rectangle(resized2, (face_location[3], face_location[0]),(face_location[1], face_location[2]), (125,220,0), 2)
55
56 cv2.imshow("IMG IN", resized2)
57 cv2.waitKey(0)
58
59 if face_location != []:
60     face_frame_encodings = face_recognition.face_encodings(resized2, known_face_locations=[face_loc])[0]
61     print(face_frame_encodings)
62     result = face_recognition.compare_faces([face_frame_encodings],facesEncodings, tolerance=0.6)
63     print("Result", result)
64     if result[0] == True:
65         text = "Rostrós"
66         color = (125, 220, 0)
67     else:
68         text = "Desconocido"
69         color = (50, 50, 255)
70
71     cv2.rectangle(frame, (face_location[3], face_location[2]), (
72         face_location[1], face_location[2] + 30), color, -1)
73     cv2.rectangle(frame, (face_location[3], face_location[0]),
74         (face_location[1], face_location[2]), color, 2)
75     cv2.putText(
76         frame, text, (face_location[3], face_location[2]+20), 2, 0.7, (255, 255, 255), 1)
77
78
79 #cv2.imshow("Frame", frame)

```

Nota. En la figura se muestra la función de la librería face encodings.

Figura 18

Resultado de la codificación.



Nota. En la figura se muestra el resultado de la programación, donde se observa la detección del rostro.

Detección de imágenes para ver si el resultado es verdadero o falso de los miembros que están agregados a la base de datos, en la comparación de imágenes se compara tanto los cuadros del vector de características.

Figura 19

Detección de imágenes para la verificación en la base de datos

```

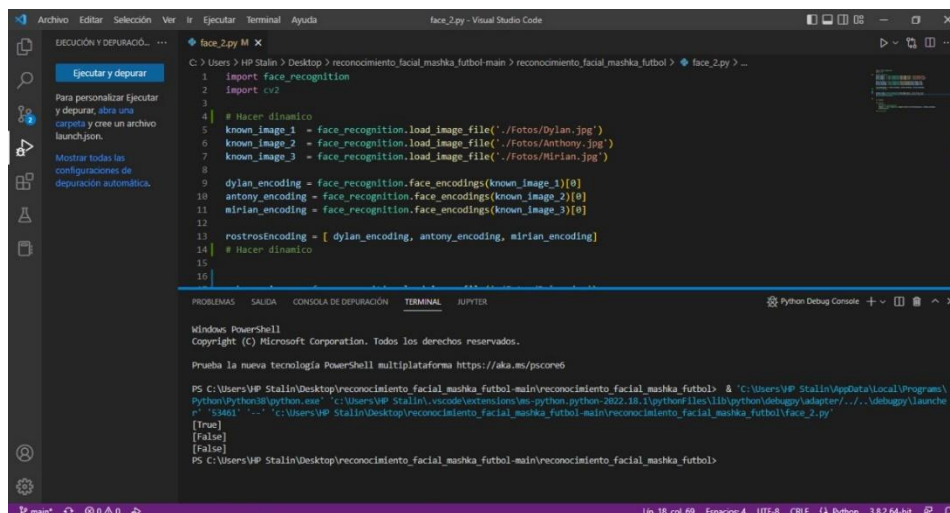
1  import face_recognition
2  import cv2
3
4  # Hacer dinámico
5  known_image_1 = face_recognition.load_image_file("../Fotos/Dylan.jpg")
6  known_image_2 = face_recognition.load_image_file("../Fotos/Anthony.jpg")
7  known_image_3 = face_recognition.load_image_file("../Fotos/Miriam.jpg")
8
9  dylan_encoding = face_recognition.face_encodings(known_image_1)[0]
10 antony_encoding = face_recognition.face_encodings(known_image_2)[0]
11 mirian_encoding = face_recognition.face_encodings(known_image_3)[0]
12
13 rostrosEncoding = [dylan_encoding, antony_encoding, mirian_encoding]
14 # Hacer dinámico
15
16 unknown_image = face_recognition.load_image_file("../Fotos/Dylan.jpg")
17 unknown_encoding = face_recognition.face_encodings(unknown_image)[0]
18 #print(unknown_encoding)
19
20
21 x = range(3)
22
23 for n in x:
24     #print(n)
25     #print(rostrosEncoding[n])
26     results = face_recognition.compare_faces([rostrosEncoding[n]], unknown_encoding)
27     print(results)
28
29 #print(results)
30

```

Nota. En la imagen se representa la codificación para la detección de imagen para la verificación en la base de datos.

Figura 20

Resultado de la codificación.



```

1 import face_recognition
2 import cv2
3
4 # Hacer dinamico
5 known_image_1 = face_recognition.load_image_file("../Fotos/Dylan.jpg")
6 known_image_2 = face_recognition.load_image_file("../Fotos/Anthony.jpg")
7 known_image_3 = face_recognition.load_image_file("../Fotos/Mirian.jpg")
8
9 dylan_encoding = face_recognition.face_encodings(known_image_1)[0]
10 antony_encoding = face_recognition.face_encodings(known_image_2)[0]
11 mirian_encoding = face_recognition.face_encodings(known_image_3)[0]
12
13 rostrosEncoding = [ dylan_encoding, antony_encoding, mirian_encoding ]
14 # Hacer dinamico
15
16

```

```

Microsoft PowerShell
Copyright (c) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol> & "C:\Users\VP Stalin\AppData\Local\Programs\Python\Python38\python.exe" "C:\Users\VP Stalin\.vscode\extensions\ms-python.python-2022.18.1\pythonFiles\lib\python\debugpy\launcher" "..." -vddbuggy/launcher" "..." "C:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol\face_2.py"
[True]
[False]
[False]
PS C:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol>

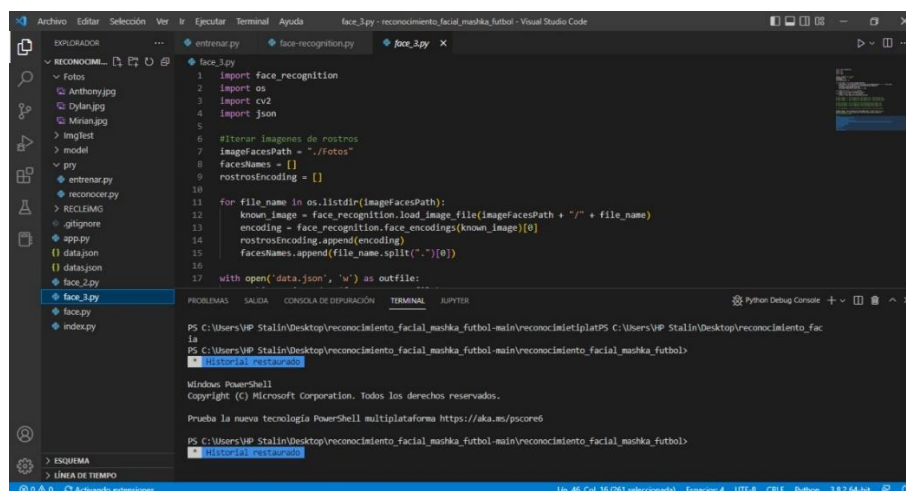
```

Nota. En la imagen se representa la ejecución de la codificación.

En Python el formato **JSON** se puede procesar gracias al paquete json, este paquete contiene el código que permite transformar los formatos json en diccionarios o viceversa.

Figura 21

Json permite transformar en diccionario o viceversa



```

1 import face_recognition
2 import os
3 import cv2
4 import json
5
6 # Iterar imagenes de rostros
7 imageFacesPath = "../Fotos"
8 facesNames = []
9 rostrosEncoding = []
10
11 for file_name in os.listdir(imageFacesPath):
12     known_image = face_recognition.load_image_file(imageFacesPath + "/" + file_name)
13     encoding = face_recognition.face_encodings(known_image)[0]
14     rostrosEncoding.append(encoding)
15     facesNames.append(file_name.split(".")[0])
16
17 with open("data.json", "w") as outfile:

```

```

PS C:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol>
PS C:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol>
Microsoft PowerShell
Copyright (c) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol>

```

Nota. En la imagen se representa la base de datos Json donde permite transformar en diccionario o viceversa.

Luego de la creación del nuevo archivo json para la base de datos que se va a utilizar, se comenzó con la optimización de las líneas de código para que así la programación sea dinámica y no sea básico a la vista.

Figura 22

Optimización de las líneas de código para que sea dinámico y no básico

```

11 for file_name in os.listdir(imageFacesPath):
12     known_image = face_recognition.load_image_file(imageFacesPath + "/" + file_name)
13     encoding = face_recognition.face_encodings(known_image)[0]
14     rostrosEncoding.append(encoding)
15     facesNames.append(file_name.split(".")[0])
16
17 with open('data.json', 'w') as outfile:
18     archivo = json.dump(facesNames,outfile)
19 with open('data_2.json', 'w') as outfile:
20     archivo = json.dump(rostrosEncoding,outfile)
21
22 # known_image_1 = face_recognition.load_image_file('./Fotos/Dylan.jpg')
23 # known_image_2 = face_recognition.load_image_file('./Fotos/Anthony.jpg')
24 # known_image_3 = face_recognition.load_image_file('./Fotos/Mirian.jpg')
25
26 # dylan_encoding = face_recognition.face_encodings(known_image_1)[0]
27 # antony_encoding = face_recognition.face_encodings(known_image_2)[0]

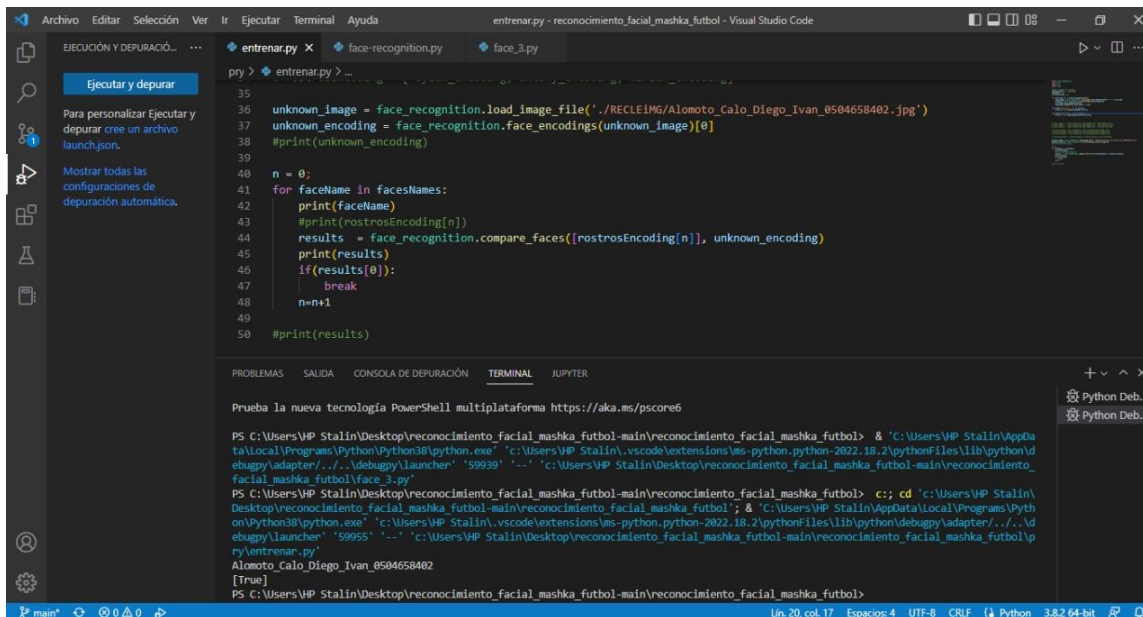
```

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure including folders like 'Fotos' and 'model', and files like 'face_3.py'. The main editor window shows the Python code for 'face_3.py'. The terminal at the bottom shows PowerShell commands and output, including a message about restoring history.

Nota. En la imagen se representa la optimización de las líneas de código para que a la vista sea dinámico y no básico.

Figura 23

Ejecución del programa lectura de las fotografías



```

entrenar.py x face-recognition.py face_3.py
Ejecución y depuración...
Ejecutar y depurar
Para personalizar Ejecutar y depurar cree un archivo launch.json.
Mostrar todas las configuraciones de depuración automática.
entrenar.py x
35
36 unknown_image = face_recognition.load_image_file("../RECLE1MG/Alomoto_Calo_Diego_Ivan_0504658402.jpg")
37 unknown_encoding = face_recognition.face_encodings(unknown_image)[0]
38 #print(unknown_encoding)
39
40 n = 0;
41 for faceName in facesNames:
42     print(faceName)
43     #print(rostrosEncoding[n])
44     results = face_recognition.compare_faces([rostrosEncoding[n]], unknown_encoding)
45     print(results)
46     if(results[0]):
47         break
48     n=n+1
49
50 #print(results)
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL JUPYTER
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6
PS C:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol> & 'C:\Users\VP Stalin\AppData\Local\Programs\Python\Python38\python.exe' 'c:\Users\VP Stalin\.vscode\extensions\ms-python.python-2022.18.2\pythonFiles\lib\python\debugpy\adapter/./../.debugpy\launcher' '59939' '--' 'c:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol\Face_3.py'
PS C:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol> c:; cd 'c:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol'; & 'C:\Users\VP Stalin\AppData\Local\Programs\Python\Python38\python.exe' 'c:\Users\VP Stalin\.vscode\extensions\ms-python.python-2022.18.2\pythonFiles\lib\python\debugpy\adapter/./../.debugpy\launcher' '59955' '--' 'c:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol\py\entrenar.py'
Alomoto_Calo_Diego_Ivan_0504658402
[True]
PS C:\Users\VP Stalin\Desktop\reconocimiento_facial_mashka_futbol-main\reconocimiento_facial_mashka_futbol>
lin. 20, col. 17 Espacios: 4 UTF-8 CRLF Python 3.8.2 64-bit

```

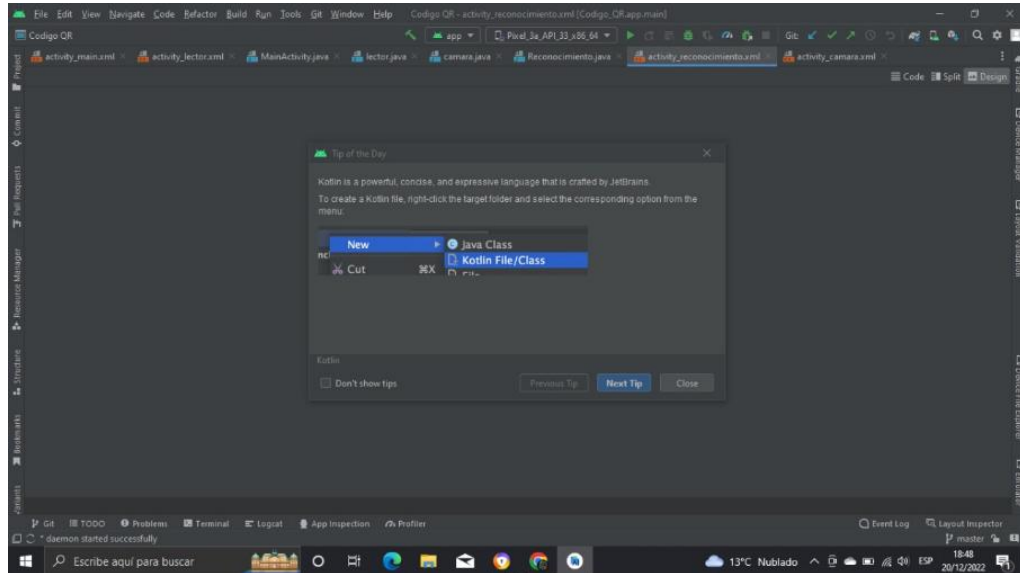
Nota. En la imagen se representa la lectura de las fotografías y comprobación de identidad.

Android Studio

Android Studio es un entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android, basado en el software JetBrains IntelliJ IDEA. Para soportar el desarrollo de aplicaciones sobre el sistema operativo Android, Android Studio utiliza un sistema de construcción basado en Gradle, emuladores, plantillas de código e integración con Github. Android Studio es el entorno de desarrollo integrado (IDE) oficial de Google para el sistema operativo Android, construido sobre el software JetBrains IntelliJ IDEA y diseñado específicamente para el desarrollo de Android. El sistema de compilación de Android es un conjunto de herramientas que se utilizan para crear, probar, ejecutar y empaquetar las aplicaciones.

Figura 24

Interfaz de android studio

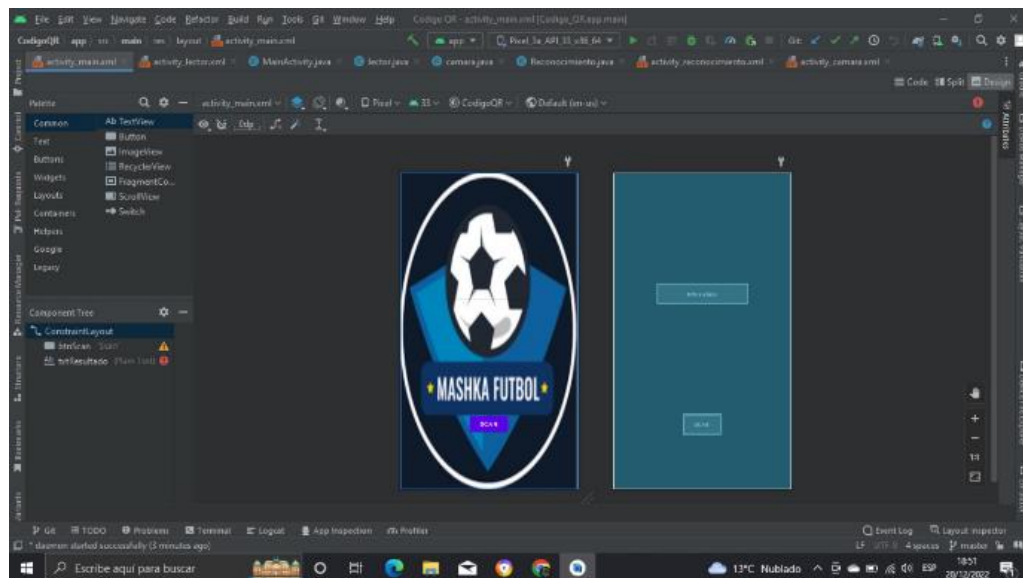


Nota. En la imagen se presenta la interfaz gráfica del software Android Studio.

Diseño de interfaz

Figura 25

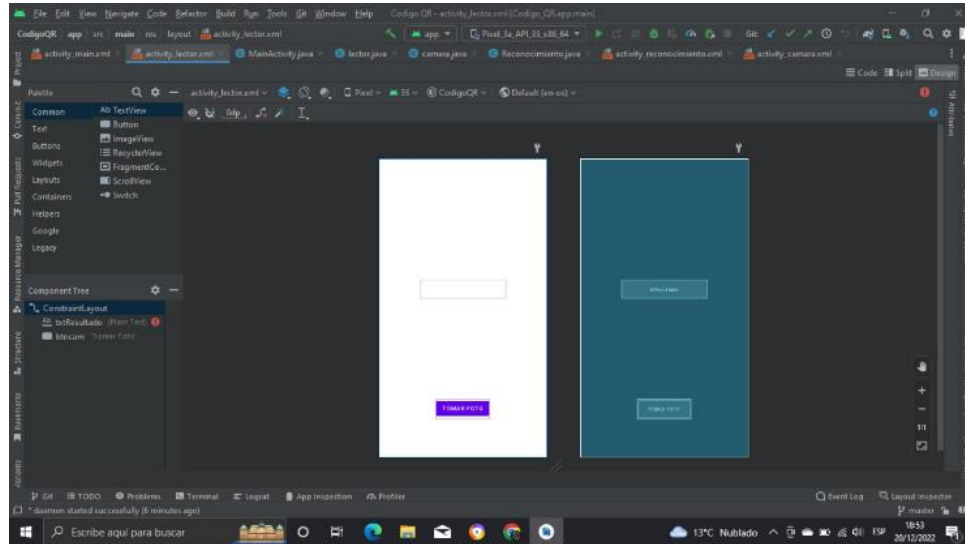
Interfaz principal y lector de código QR



Nota. En la imagen se representa el diseño del aplicativo con su respectivo logo.

Figura 26

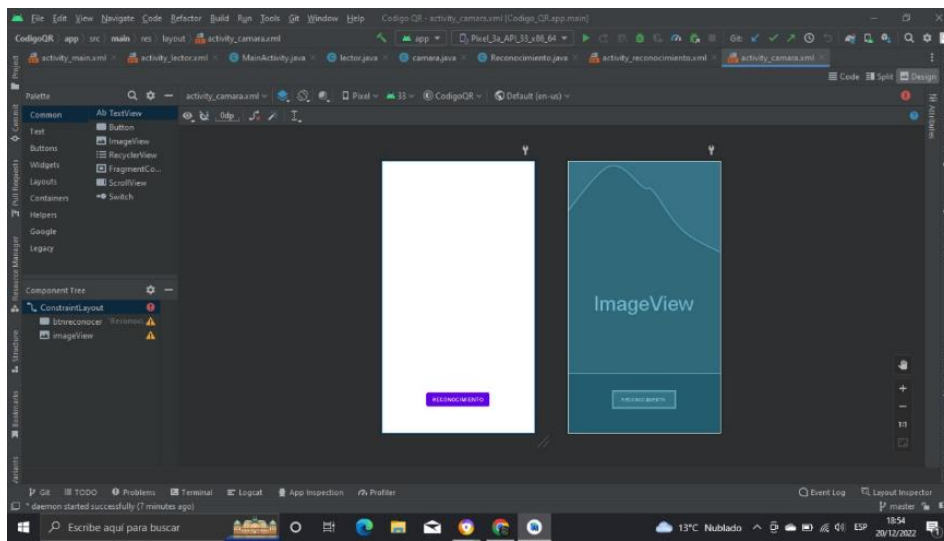
Interfaz de lectura de los datos del jugador, "Lectura del QR"



Nota. En la imagen se representa el diseño para la lectura de código QR

Figura 27

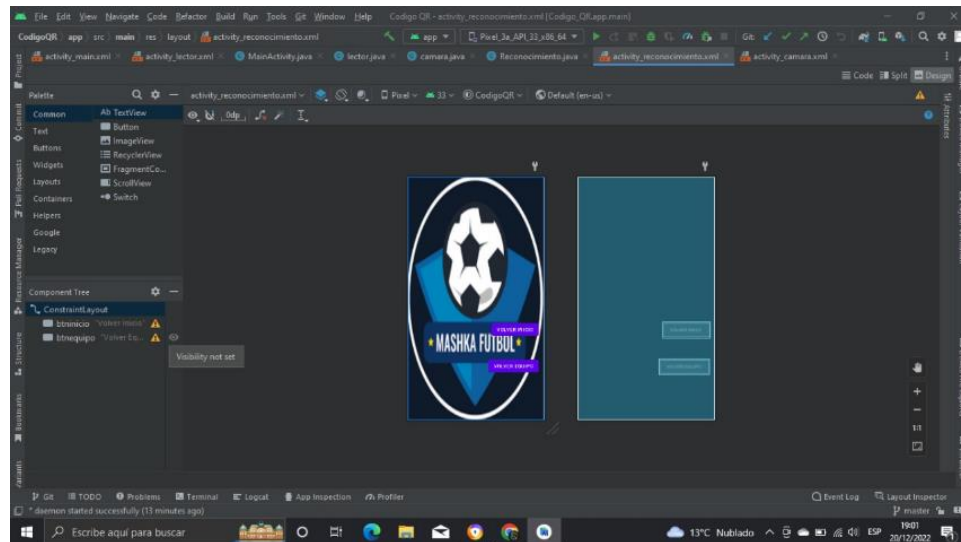
Interfaz de activación de cámara



Nota. En la imagen se representa el diseño para activar la cámara con el botón tomar foto.

Figura 28

Interfaz de reconocimiento de identidad del jugador

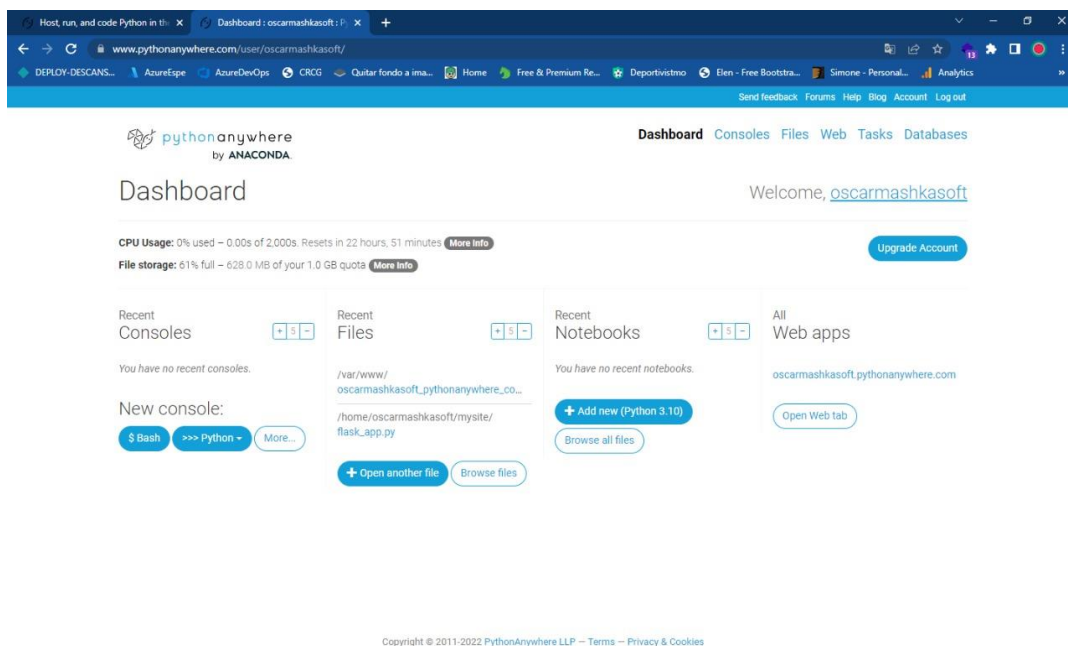


Nota. En la imagen se representa el diseño para el reconocimiento de identidad del jugador.

Creación de API Rest con Python

Figura 29

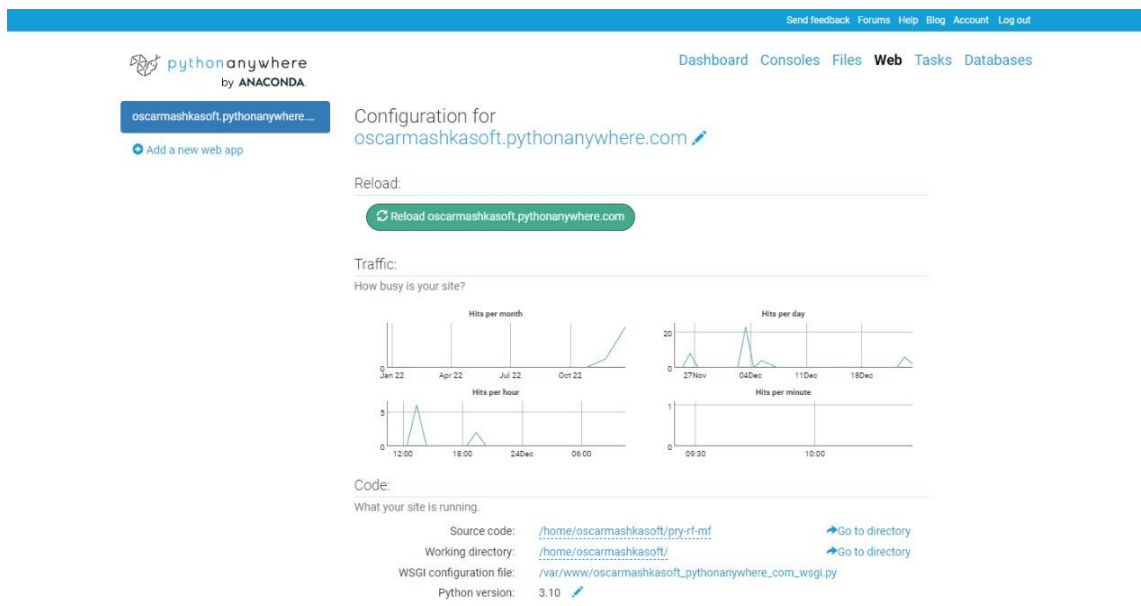
Pantalla principal del servicio de Python anyware



Nota. En la imagen se representa la interfaz principal del servicio de Python anyware.

Figura 30

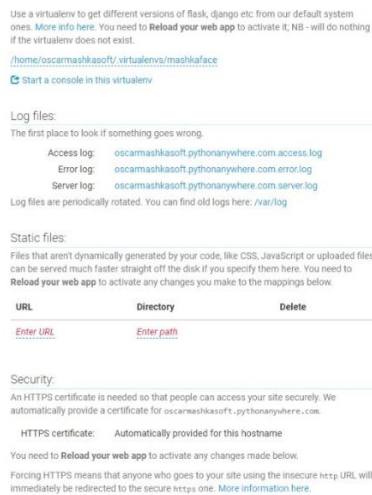
Despliegue en producción del backend, dominio para acceso de API REST hecho en Python.



Nota. En la imagen se presenta la interfaz donde existe un despliegue en producción del backend un dominio para el acceso a API REST en Python.

Figura 31

Configuración del entorno virtual para ejecutar el API REST.



Nota. En la imagen se representa la interfaz de configuración del entorno virtual para ejecutar el API REST.

Figura 32

Instalación de librerías, clonación de proyecto git en consola de servicio Python anywhere.



```

15:25 ~ $ ls
README.txt  mysite  pry-rf-mf
15:26 ~ $ cd pry-rf-mf/
15:26 ~/pry-rf-mf (main) $ ls
__pycache__  app.py  requirements.txt  src  utils
15:26 ~/pry-rf-mf (main) $ cat requirements.txt
anis8601==9.0.1
click==8.1.3
colorama==0.4.5
face-recognition==1.3.0
face-recognition-models==0.3.0
Flask==2.2.2
Flask-Cors==3.0.10
Flask-RESTful==0.3.9
importlib-metadata==5.0.0
itsdangerous==2.1.2
Jinja2==3.1.2
MarkupSafe==2.1.1
numpy==1.23.3
opencv-contrib-python==4.6.0.66
pillow==9.2.0
python-decouple==3.6
python-dotenv==0.21.0
pytz==2022.6
six==1.16.0
werkzeug==2.2.2
zip==3.10.0
15:26 ~/pry-rf-mf (main) $ cat app.py
from flask import Flask, jsonify, request
import face_recognition
from datetime import datetime
import numpy as np
from PIL import Image
import base64
import json

from utils.cronometer import Cronometer

app = Flask(__name__)

@app.route('/')
def get_movies():
    return "<div>Page MashkaSoft MashkaFutbol 1.0.0</div>" , 404

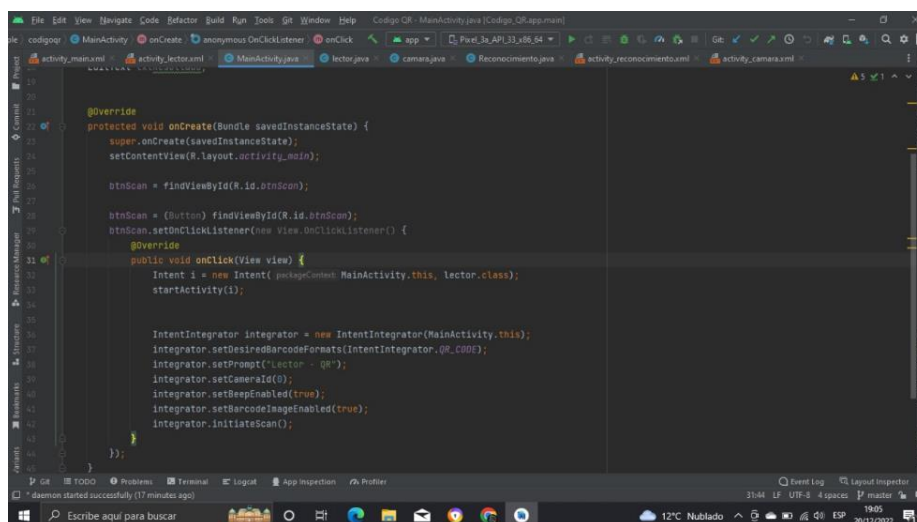
@app.route('/api/person/recognition', methods=['POST'])
def post_data():
  
```

Nota. En la imagen se representa la instalación de las librerías y la clonación del proyecto.

Creación del aplicativo móvil

Figura 33

Las siguientes clases son principales para la administración y función del aplicativo web



```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    btnScan = findViewById(R.id.btnScan);

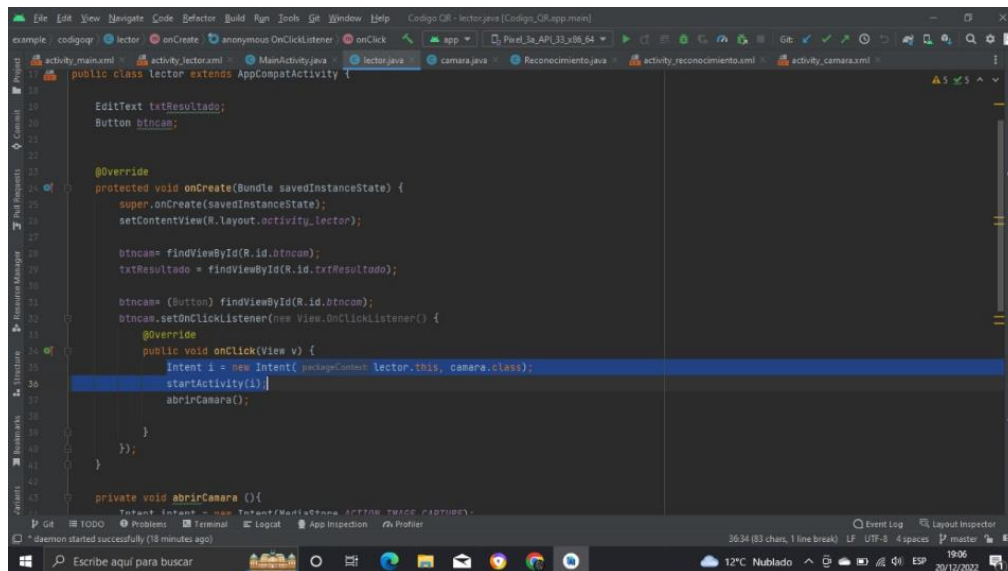
    btnScan = (Button) findViewById(R.id.btnScan);
    btnScan.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent(packageContext, Lector.class);
            startActivity(i);

            IntentIntegrator integrator = new IntentIntegrator(MainActivity.this);
            integrator.setDesiredBarcodeFormats(IntentIntegrator.QR_CODE);
            integrator.setPrompt("Lector - QR");
            integrator.setCameraId(0);
            integrator.setBeepEnabled(true);
            integrator.setBarcodeImageEnabled(true);
            integrator.initiateScan();
        }
    });
}
  
```

Nota. En la imagen se representa las clases principales para la administración y función de la aplicación.

Figura 34

Clase que permite el reconocimiento de Código QR



```

public class Lector extends AppCompatActivity {

    EditText txtResultado;
    Button btncas;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lector);

        btncas = findViewById(R.id.btncas);
        txtResultado = findViewById(R.id.txtResultado);

        btncas = (Button) findViewById(R.id.btncas);
        btncas.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i = new Intent(getApplicationContext(), Camara.class);
                startActivity(i);
                abrirCamara();
            }
        });

    }

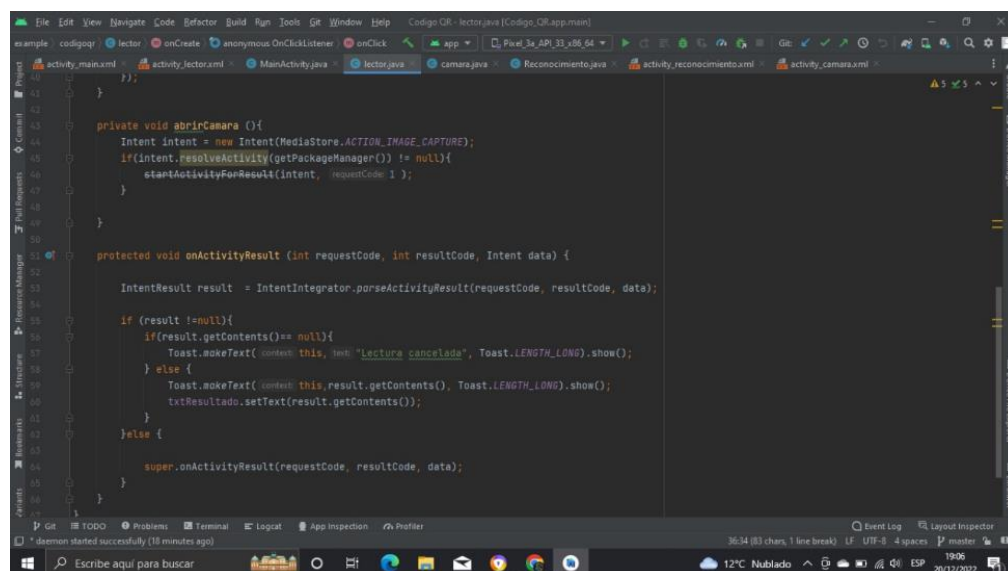
    private void abrirCamara () {
}
}

```

Nota. En la imagen se representa la clase que proporciona el reconocimiento de código QR.

Figura 35

Clase que permite el reconocimiento de Código QR



```

}

private void abrirCamara () {
    Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if(intent.resolveActivity(getPackageManager()) != null){
        startActivityForResult(intent, RequestCode 1 );
    }
}

protected void onActivityResult (int requestCode, int resultCode, Intent data) {

    IntentResult result = IntentIntegrator.parseActivityResult(requestCode, resultCode, data);

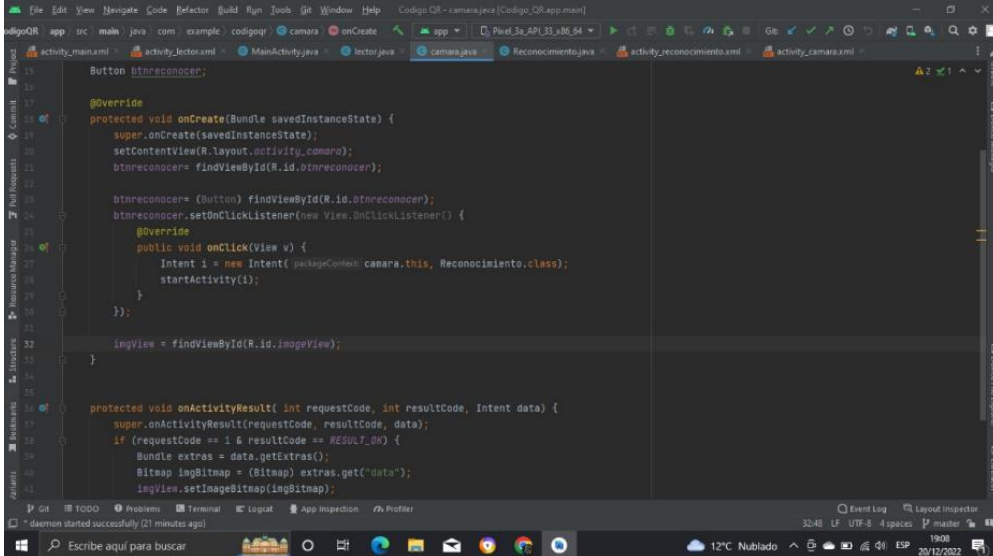
    if (result != null){
        if(result.getContents() == null){
            Toast.makeText(context, this, "Lectura cancelada", Toast.LENGTH_LONG).show();
        } else {
            Toast.makeText(context, this, result.getContents(), Toast.LENGTH_LONG).show();
            txtResultado.setText(result.getContents());
        }
    } else {
        super.onActivityResult(requestCode, resultCode, data);
    }
}
}

```

Nota. En la imagen se representa la clase que permite la identificación del código QR.

Figura 36

Clase para la toma de fotografía del rostro



```

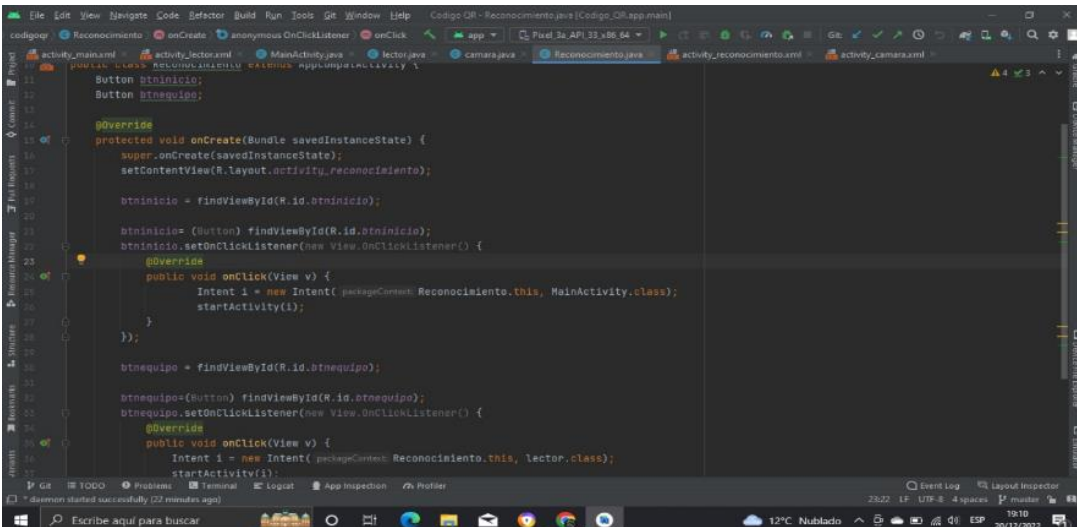
15 Button btnreconocer;
16
17 @Override
18 protected void onCreate(Bundle savedInstanceState) {
19     super.onCreate(savedInstanceState);
20     setContentView(R.layout.activity_camara);
21     btnreconocer = findViewById(R.id.btnreconocer);
22
23     btnreconocer = (Button) findViewById(R.id.btnreconocer);
24     btnreconocer.setOnClickListener(new View.OnClickListener() {
25         @Override
26         public void onClick(View v) {
27             Intent i = new Intent(packageContext.camara.this, Reconocimiento.class);
28             startActivity(i);
29         }
30     });
31
32     imageView = findViewById(R.id.imageView);
33 }
34
35
36 @Override
37 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
38     super.onActivityResult(requestCode, resultCode, data);
39     if (requestCode == 1 & resultCode == RESULT_OK) {
40         Bundle extras = data.getExtras();
41         Bitmap imgBitmap = (Bitmap) extras.get("data");
42         imageView.setImageBitmap(imgBitmap);
43     }
44 }

```

Nota. En la imagen se representa la función de la clase para la toma de fotografías.

Figura 37

Clase para el reconocimiento facial y verificación de resultados



```

11 Button btnInicio;
12 Button btnEquipo;
13
14 @Override
15 protected void onCreate(Bundle savedInstanceState) {
16     super.onCreate(savedInstanceState);
17     setContentView(R.layout.activity_reconocimiento);
18
19     btnInicio = findViewById(R.id.btnInicio);
20
21     btnInicio = (Button) findViewById(R.id.btnInicio);
22     btnInicio.setOnClickListener(new View.OnClickListener() {
23         @Override
24         public void onClick(View v) {
25             Intent i = new Intent(packageContext.Reconocimiento.this, MainActivity.class);
26             startActivity(i);
27         }
28     });
29
30     btnEquipo = findViewById(R.id.btnEquipo);
31
32     btnEquipo = (Button) findViewById(R.id.btnEquipo);
33     btnEquipo.setOnClickListener(new View.OnClickListener() {
34         @Override
35         public void onClick(View v) {
36             Intent i = new Intent(packageContext.Reconocimiento.this, Lector.class);
37             startActivity(i);
38         }
39     });
40 }

```

Nota. En la imagen se representa la clase para el reconocimiento facial y verificación de los resultados.

Fase 4. Pruebas

Prueba de aceptación

Para las pruebas se determinará el siguiente modelo.

Tabla 19

Modelo para las pruebas de aceptación

Prueba de aceptación	
Historia de usuario:	N°
Nombre:	
Prueba de aceptación:	
Condiciones de ejecución:	
Descripción:	
Pasos:	
Resultado esperado:	
Evaluación de la prueba:	

Nota. En la siguiente tabla se representa los detalles que se necesitara para las pruebas de aceptación para cada usuario.

Historia de usuario. - Número de historia de usuario a la que pertenece la prueba.

Número. – Identificador de la prueba.

Nombre. - Título otorgado a la prueba.

Condiciones de ejecución. - detalle de especificaciones para que la prueba funcione de manera correcta.

Descripción. - Detalle de especificaciones que debe cumplir la prueba de aceptación.

Pasos. – pasos que se debe seguir para que se cumpla la prueba de aceptación.

Resultado esperado. - Relato que se obtiene al realizar la prueba de aceptación.

Evaluación de la prueba. - resultado Satisfactorio- No satisfactorio.

Pruebas de aceptación

En esta fase se desarrollan las pruebas de aceptación de acuerdo con las historias de usuario establecidas.

Tabla 20

Prueba de aceptación usuario 1

Prueba de aceptación	
Historia de usuario: 5	N°1
Nombre: Acceso al sistema	
Descripción: El usuario deberá estar ingresado sus datos al sistema	
Condiciones de ejecución: El usuario deberá tener el código QR para poder validar	
Pasos: El usuario debe detectar el código Qr para verificar si pertenece a ese equipo	
Resultado esperado: El usuario accede al sistema	
Evaluación de la prueba: Satisfactorio	

Nota. En la tabla se representa los detalles para las pruebas de aceptación.

Tabla 21

Prueba de aceptación usuario 2

Prueba de aceptación	
Historia de usuario: 6	N°2
Nombre: Administrador	
Descripción: El usuario deberá visualizar que sus datos que se encuentren registrado	
Prueba de aceptación	

Prueba de aceptación

Condiciones de ejecución: El usuario deberá visualizar los detalles en la descripción del equipo

Pasos: Verificar que se encuentre en el equipo correcto mediante los datos que se visualizaran

Resultado esperado: El usuario puede verificar el equipo que pertenece

Evaluación de la prueba: Satisfactorio

Nota. En la tabla se representa la descripción para el usuario 2.

Tabla 22

Prueba de aceptación usuario 3

Prueba de aceptación

Historia de usuario: 7

N°3

Nombre: Administrador

Descripción: El usuario deberá tomarse una foto para la verificación de datos

Condiciones de ejecución: El usuario deberá tomarse un foto para verificar los datos la persona

Pasos: Dirigirse al icono u botón tomar fotografía para realizar el paso de reconocimiento

Resultado esperado: El usuario puede tomarse la foto

Evaluación de la prueba: Satisfactorio

Nota. En la tabla se representa la descripción para el usuario 3.

Tabla 23*Prueba de aceptación usuario 4*

Prueba de aceptación	
Historia de usuario: 7	N°4
Nombre: Conexión a la base de datos	
Descripción: Se utilizará la base de datos JSON	
Condiciones de ejecución: El usuario deberá agregar las fotografías a la base de datos para poder hacer el reconocimiento	
Pasos: Dirigirse hacia la base de datos que este generada agregar la fotografía y los datos del jugador	
Resultado esperado: El usuario puede agregar a más participantes	
Evaluación de la prueba: Satisfactorio	
<i>Nota.</i> En la tabla se representa la descripción para el usuario 4.	

Tabla 24*Prueba de aceptación usuario 5*

Prueba de aceptación	
Historia de usuario: 7	N°5
Nombre: Visualización de la interfaz del aplicativo	
Descripción: Se desarrolla la interfaz del aplicativo	
Condiciones de ejecución: El usuario puede verificar y manipular todos los parámetros que se crearon en el aplicativo	
Pasos: Abrir todos los pasos del aplicativo	

Prueba de aceptación

Resultado esperado: El usuario puede manipular todo el aplicativo

Evaluación de la prueba: Satisfactorio

Nota. En la tabla se representa la descripción para el usuario 5.

Tabla 25

Prueba de aceptación usuario 6

Prueba de aceptación

Historia de usuario: 8

N°6

Nombre: Verificación de registro de los usuarios

Descripción: Se verificará a los usuarios en la base de datos posterior mente de la verificación del QR

Condiciones de ejecución: El usuario deberá tomarse la foto para poder verificar la identidad

Pasos: Posterior mente de ya haberse tomado la foto tendrá que verificar que la foto este en buen estado.

Resultado esperado: El usuario puede verificar y tomarse nueva mente la foto si la fotografía está en mal estado

Evaluación de la prueba: Satisfactorio

Nota. En la tabla se representa la descripción para el usuario 6.

Tabla 26*Prueba de aceptación usuario 7*

Prueba de aceptación	
Historia de usuario: 9	N°7
Nombre: Respuesta de datos del jugador	
Descripción: El usuario puede consultar los datos en la página luego de la detección del rostro	
Condiciones de ejecución: El usuario deberá presionar el icono de reconocimiento para poder verificar su identidad	
Pasos: Posterior mente de ya haberse tomado la foto tendrá que presionar el botón de reconocimiento para verificar su identidad	
Resultado esperado: El usuario puede verificar identidad y sus datos	
Evaluación de la prueba: Satisfactorio	

Nota. En la tabla se representa la descripción para el usuario 7.

Reconocimiento facial

El reconocimiento facial es una tecnología de reconocimiento biométrico que utiliza características faciales únicas para identificar a una persona. Esta tecnología se ha utilizado en una variedad de aplicaciones, desde el control de acceso a la seguridad hasta la identificación de personas.

El reconocimiento facial se basa en la detección de características faciales únicas, como la forma de la cara, el tamaño de los ojos, la distancia entre los ojos, la forma de la nariz y la forma de la boca. Estas características se comparan con una base de datos de caras

conocidas para identificar a una persona. El reconocimiento facial también se puede utilizar para detectar el envejecimiento, el género y la expresión facial.

Pruebas de la aplicación móvil.

Para probar la aplicación de reconocimiento facial móvil, se debe realizar una prueba de usuario para verificar la precisión y la facilidad de uso. Esto implica la recopilación de datos de usuarios reales para verificar la precisión de la aplicación. Esto se puede hacer mediante la recopilación de datos de usuarios reales que usen la aplicación y luego comparar los resultados con los datos de la base de datos JSON.

Además, se debe realizar una prueba de rendimiento para verificar la velocidad de procesamiento de la aplicación. Esto se puede hacer mediante la medición de la velocidad de procesamiento de la aplicación al procesar una imagen de una cara conocida.

Ambiente controlado

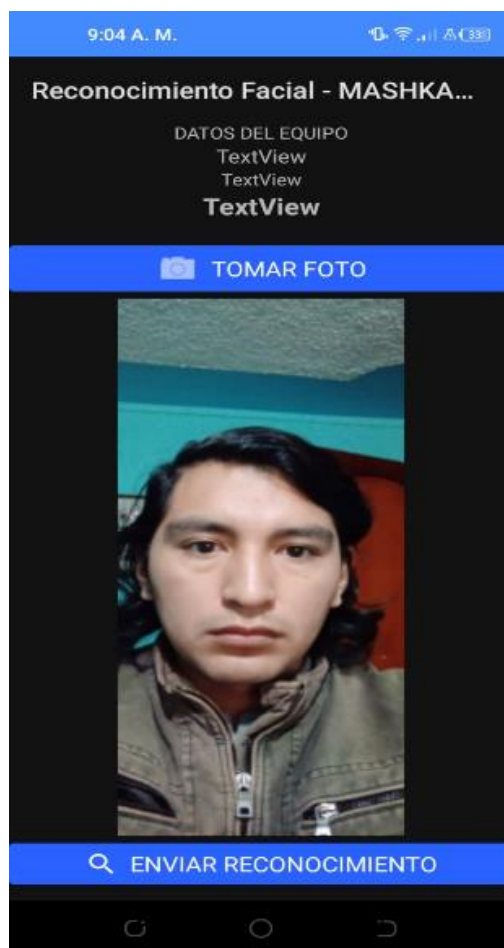
Las pruebas para el reconocimiento facial pueden controlarse mediante el uso de un entorno de pruebas controlado. Esto significa que se pueden establecer parámetros específicos para el entorno de pruebas, como la iluminación, el fondo, la distancia entre la cámara y el sujeto, el ángulo de la cámara, etc.

Esto permite probar el sistema de reconocimiento facial bajo condiciones controladas y predecibles, lo que ayuda a asegurar que el sistema funcione correctamente en un entorno real.

Además, esto también ayuda a identificar los factores que pueden afectar el rendimiento del sistema, como la iluminación, el fondo, la distancia entre la cámara y el sujeto, el ángulo de la cámara, etc. Esto nos permite mejorar el sistema de reconocimiento facial para que funcione mejor en un entorno real.

Figura 38

Interfaz de reconocimiento facial Mashka Fútbol



Nota. En la imagen se representa la interfaz donde se envía el reconocimiento del jugador.

Características

Conexión Internet - wifi oficina.

La conexión a Internet para el reconocimiento facial es una prueba para verificar si la conexión a Internet es suficiente para el reconocimiento facial. Esta prueba se realiza para asegurar que la conexión a Internet es lo suficientemente rápida y estable para que el reconocimiento facial funcione correctamente.

Esta prueba se realiza mediante la medición de la velocidad de descarga y carga de la conexión a Internet, así como la latencia de la conexión. Si la velocidad de descarga y carga es suficiente y la latencia es aceptable, entonces la conexión a Internet es adecuada para el reconocimiento facial.

Figura 39

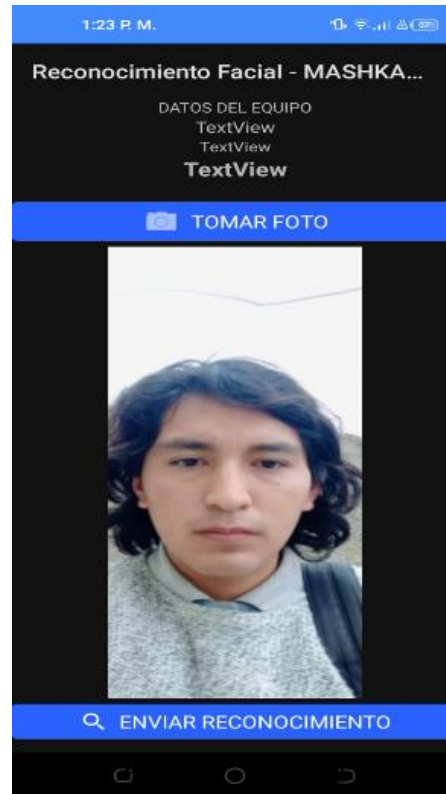
Conexión adecuada para la aplicación



Nota. En la imagen se representa la conexión adecuada a wifi y el uso de la app.

Luz reflejo de luz

El reconocimiento facial basado en el reflejo de la luz es una técnica de reconocimiento facial que utiliza una cámara para detectar la luz reflejada por la cara de una persona. Esta luz reflejada se analiza para identificar características únicas de la cara, como la forma, el tamaño y la distribución de los rasgos faciales. Esta información se compara con una base de datos de caras conocidas para determinar si la persona es una persona conocida o no. Esta técnica se utiliza a menudo en sistemas de seguridad para verificar la identidad de una persona.

Figura 40*Reflejo de luz*

Nota. En la imagen se representa la importancia que tiene el reflejo de luz.

Posición del teléfono

La prueba de posición del teléfono para el reconocimiento facial es una prueba que se realiza para asegurar que el teléfono está en la posición correcta para que el reconocimiento facial funcione correctamente. Esta prueba se realiza para asegurar que el teléfono está a la misma distancia de la cara del usuario, que el ángulo de la cámara está correcto y que el teléfono está enfocado correctamente. Esta prueba se realiza para asegurar que el teléfono está listo para el reconocimiento facial y que el usuario obtendrá los mejores resultados.

Figura 41

Posición horizontal del teléfono



Nota. En la imagen se representa la posición del teléfono y que a su vez es importante para el reconocimiento del rostro.

Cámara del teléfono - característica mpX

La cámara del teléfono MPX ofrece una variedad de características para el reconocimiento facial. Estas características incluyen:

- **Detección de rostros:** La cámara MPX puede detectar rostros en una imagen y luego ajustar la configuración de la cámara para obtener la mejor imagen posible.
- **Reconocimiento facial:** La cámara MPX puede reconocer caras individuales y luego guardar esta información para usarla en aplicaciones posteriores.

Figura 42

Características de la cámara de reconocimiento facial



Nota. En la imagen se representa las características de la cámara dependiendo del tipo de teléfono celular.

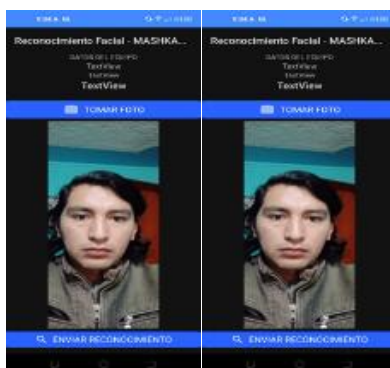
Pruebas ambientes controlado***Tiempo de respuesta*****Tabla 27**

Pruebas de Tiempo de respuesta

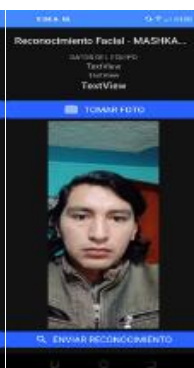
Tiempo de Respuesta al detectar la fotografía

Primera prueba	Segunda prueba	Tercera prueba	Cuarta Prueba	Quinta Prueba	Total, tiempo
----------------	----------------	----------------	---------------	---------------	---------------

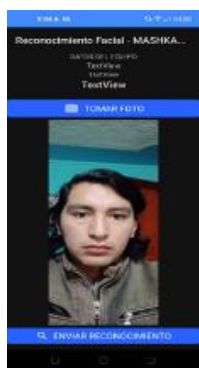
Estimación	Estimación	Estimación	Estimación	Estimación	Tiempo
------------	------------	------------	------------	------------	--------



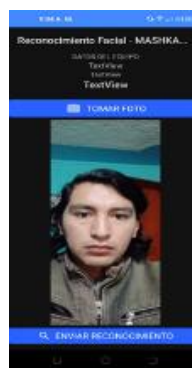
4 Seg.



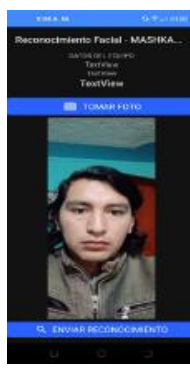
1 Seg.



1 Seg.



4 Seg.



4 Seg.

estimado
para la
identificaci
n de
fotografías
es de: 1.6

Estimación

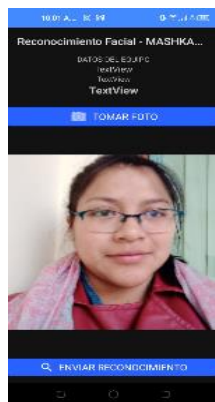
Estimación

Estimación

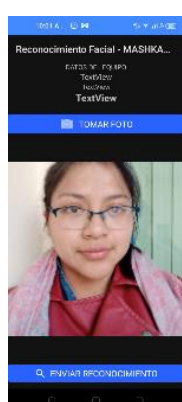
Estimación

Estimación

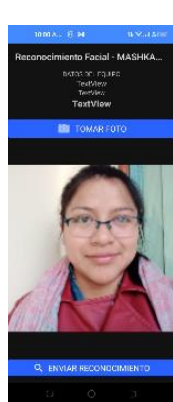
Tiempo



4. Seg.

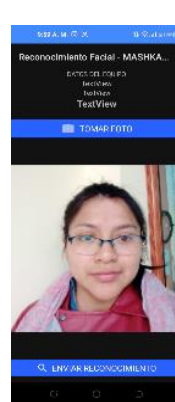


1 Seg.



Seg.

1



1 Seg.

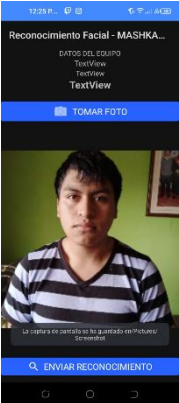
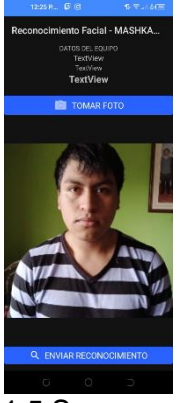
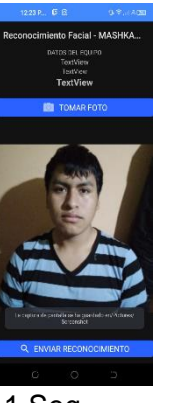

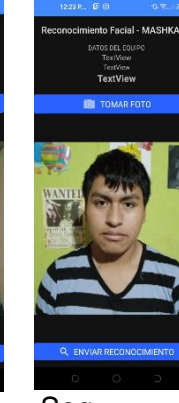


1 Seg.

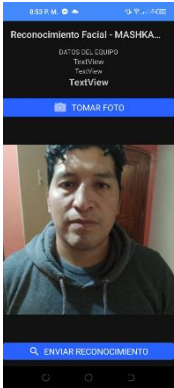
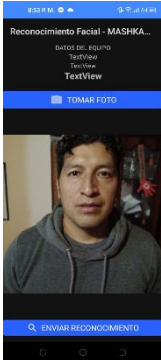
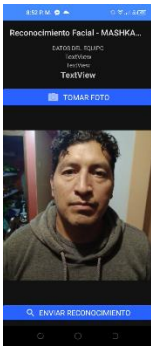
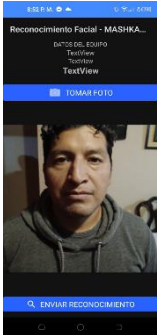

estimado
para la
identificaci
ón de
fotografías
es de: 1.6

Tiempo de Respuesta al detectar la fotografía

Estimación	Estimación	Estimación	Estimación	Estimación	Tiempo estimado para la identificación de fotografías es de:
					1.72
3 Seg.	1 Seg.	2 Seg.	1 Seg.	1.6 Seg.	

Estimación	Estimación	Estimación	Estimación	Estimación	Tiempo estimado para la identificación de fotografías es de:
					2.5
4 Seg.	1.5 Seg.	1 Seg.	1 Seg.	Seg.	2

Tiempo de Respuesta al detectar la fotografía

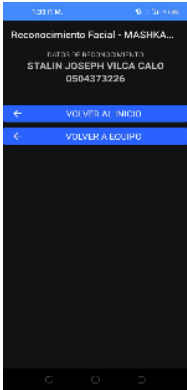
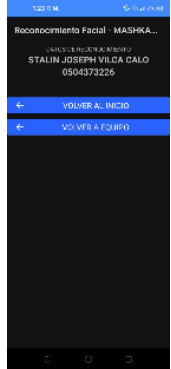
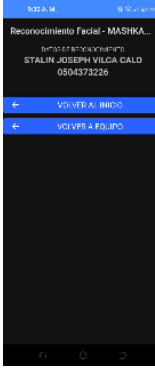
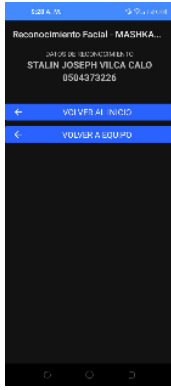
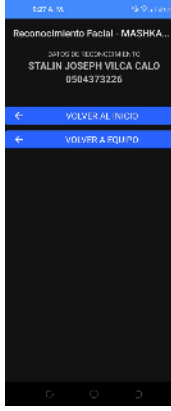
Estimación	Estimación	Estimación	Estimación	Estimación	Tiempo estimado para la identificación de fotografías es de: 1.26
					
1 Seg.	1.5 Seg.	1.8 Seg.	2 Seg.	1 Seg.	

Porcentaje total de reconocimiento fotografías: 1.63


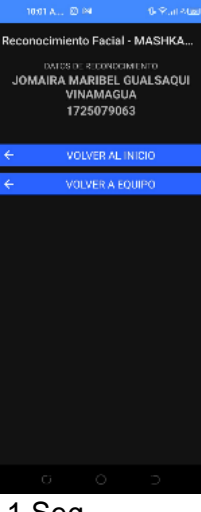
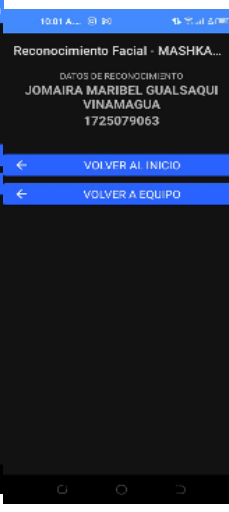
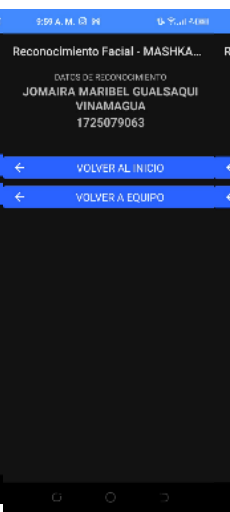
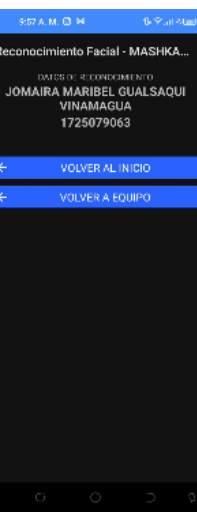
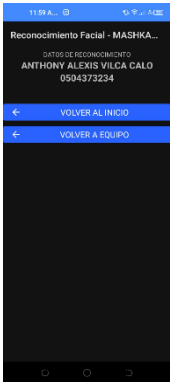
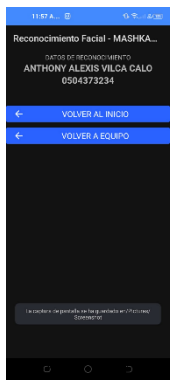
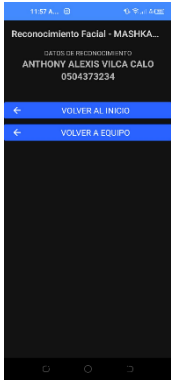
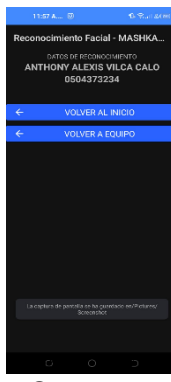
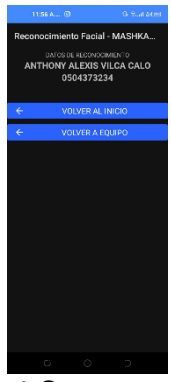
Nota. En la tabla se representa las pruebas que se realizó en el ambiente controlado con 5 usuarios.

Tabla 28

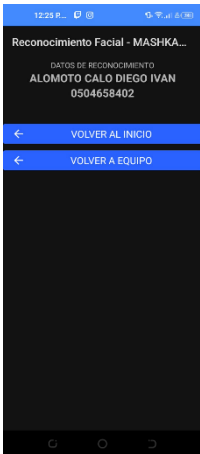
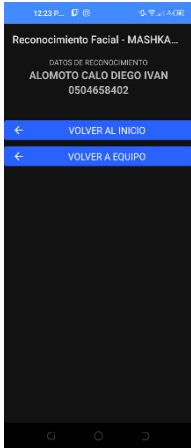
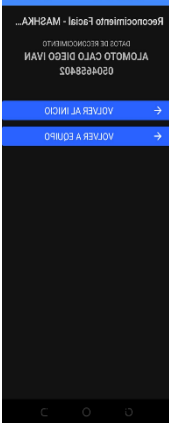
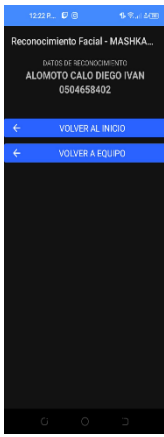
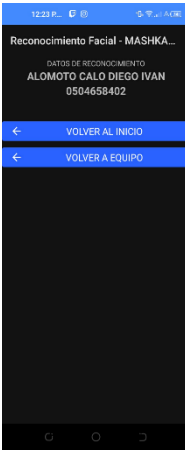
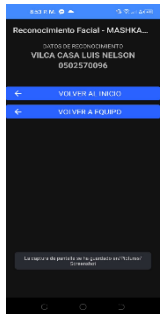
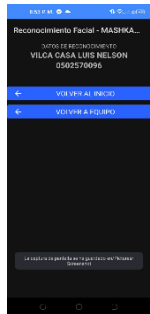
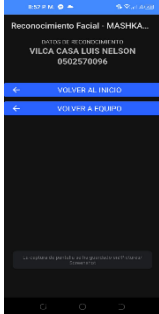
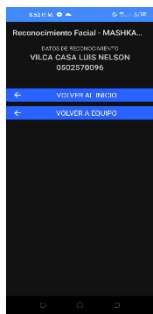
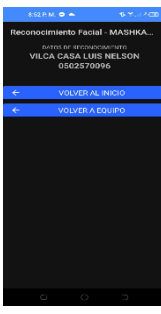
Tiempo de respuesta al reconocer a la persona

Tiempo de Respuesta al reconocer a la persona					
Primera prueba	Segunda prueba	Tercera prueba	Cuarta Prueba	Quinta Prueba	Total, tiempo
Estimación	Estimación	Estimación	Estimación	Estimación	El tiempo estimado para la identificación de fotografías es de: 1.9 segundo
					
4.5 seg.	1 seg.	n	1 seg.	2 seg.	
		1 seg.			

Tiempo de Respuesta al detectar la fotografía

Estimación	Estimación	Estimación	Estimación	Estimación	El tiempo
 <p>4.5 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	<p>El tiempo estimado para la identificación de fotografía es de: 1.7</p>
 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	<p>El tiempo estimado para la identificación de fotografía es de: 1</p>

Tiempo de Respuesta al detectar la fotografía

Estimación	Estimación	Estimación	Estimación	Estimación	
 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>Seg.</p>	<p>1</p> <p>El tiempo estimado para la identificación de fotografía es de: 1</p>
 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	 <p>1 Seg.</p>	<p>1 Seg.</p> <p>El tiempo estimado para la identificación de fotografía es de: 1</p>

Porcentaje total en la identificación de datos = 1.32

Nota. En la imagen se presenta el tiempo de respuesta en la identificación de datos.

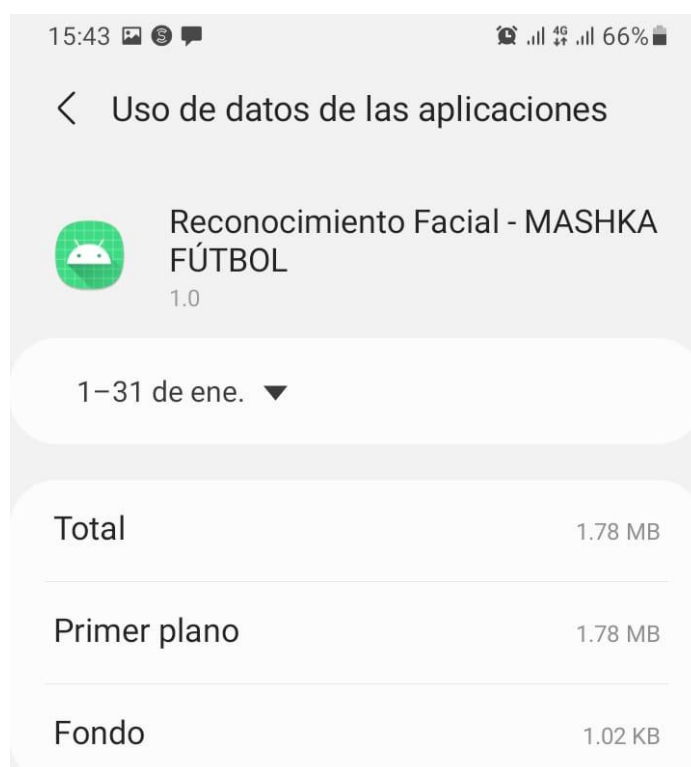
Ambiente no controlado

Conexión internet - datos - señal

Prueba de señal: Esta prueba se realiza para verificar la calidad de la señal de internet que se está utilizando para el reconocimiento facial. Se debe comprobar la velocidad de la conexión, la latencia y la fiabilidad de la señal.

Figura 43

Datos utilizados



Nota. En la imagen se representa los datos utilizados en la app.

Cuantos datos se gastos 1000MG - 450MB

Se gasto aproximadamente 1.78 MB

Luz del sol

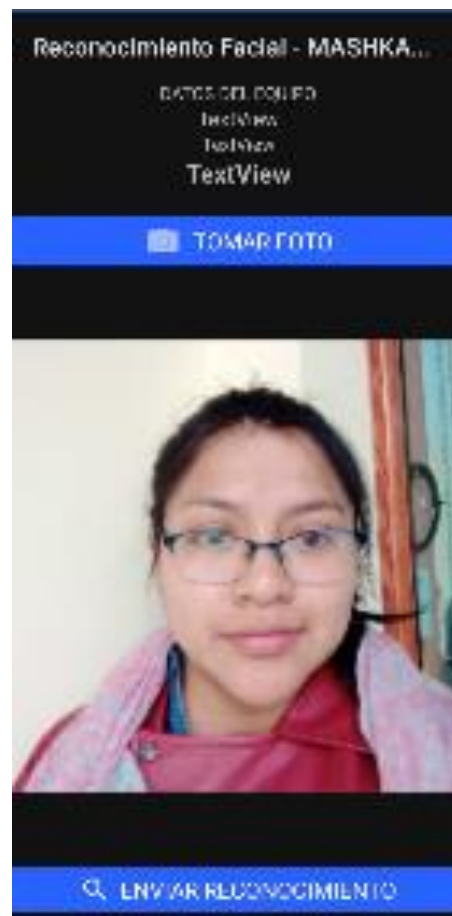
Uno de los principales problemas con el reconocimiento facial es que la luz del sol puede afectar la calidad de la imagen. Esto puede hacer que el algoritmo de reconocimiento

facial no sea capaz de detectar correctamente los rasgos faciales de una persona. Esto puede resultar en una tasa de error significativamente mayor

. Además, la luz del sol puede afectar la calidad de la imagen de tal manera que el algoritmo de reconocimiento facial no sea capaz de detectar correctamente los rasgos faciales de una persona. Esto puede resultar en una tasa de error significativamente mayor.

Figura 44

Reflejo de luz



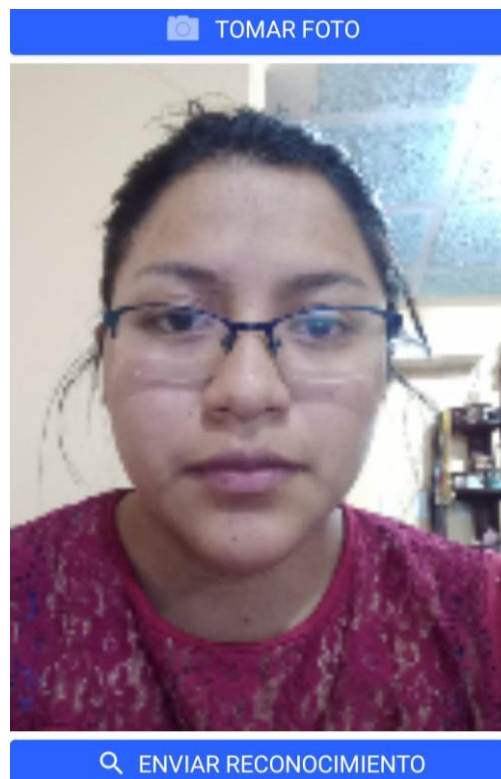
Nota. En la imagen se representa el reflejo de luz donde es importante para el reconocimiento del rostro.

Posición del teléfono

Uno de los principales escenarios en los que el reconocimiento facial no funciona es cuando el teléfono está en una posición inadecuada. Esto puede ocurrir cuando el teléfono está demasiado cerca o demasiado lejos de la cara, cuando está en un ángulo inadecuado o cuando hay una iluminación inadecuada. Esto puede hacer que el teléfono no sea capaz de detectar correctamente los rasgos faciales necesarios para el reconocimiento.

Figura 45

Posición horizontal del teléfono



Nota. En la imagen se representa la posición del teléfono móvil.

Cámara del teléfono - característica MPX

Prueba de luz: La cámara del teléfono debe ser capaz de capturar imágenes con una variedad de niveles de luz. Esto significa que el reconocimiento facial debe ser capaz de funcionar bajo diferentes condiciones de luz.

Prueba de ángulo: El reconocimiento facial debe ser capaz de detectar caras desde diferentes ángulos. Esto significa que la cámara del teléfono debe ser capaz de capturar imágenes desde diferentes ángulos.

Prueba de distancia: El reconocimiento facial debe ser capaz de detectar caras desde diferentes distancias. Esto significa que la cámara del teléfono debe ser capaz de capturar imágenes desde diferentes distancias.

Prueba de resolución: El reconocimiento facial debe ser capaz de detectar caras con una alta resolución. Esto significa que la cámara del teléfono debe ser capaz de capturar imágenes con una alta resolución.

Prueba de velocidad: El reconocimiento facial debe ser capaz de detectar caras rápidamente. Esto significa que la cámara del teléfono debe ser capaz de capturar imágenes con una velocidad rápida.

Figura 46

Características del celular



Nota. En la imagen se representa las caracterizariás de la cámara.

Pruebas ambientes no controlado

Tiempo de respuesta

Tabla 29

Ambiente no controlado pruebas

Tiempo de Respuesta al detectar la fotografía

Primera prueba	Segunda prueba	Tercera prueba	Cuarta Prueba	Quinta Prueba	Total, tiempo
----------------	----------------	----------------	---------------	---------------	---------------

Estimación 2 Seg.	Estimación 1 Seg.	Estimación 1 Seg.	Estimación 1 Seg.	Estimación 2 Seg.	Tiempo estimado
----------------------	----------------------	----------------------	----------------------	----------------------	-----------------



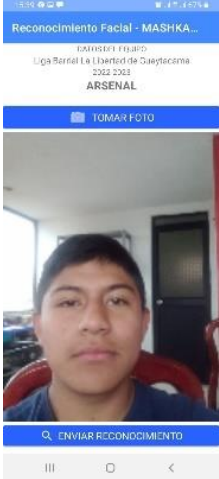
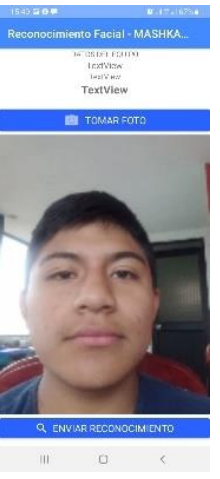
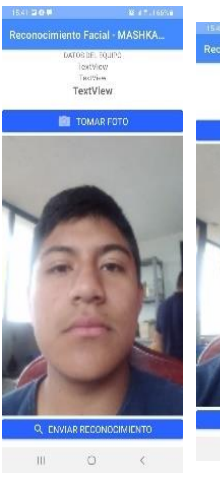
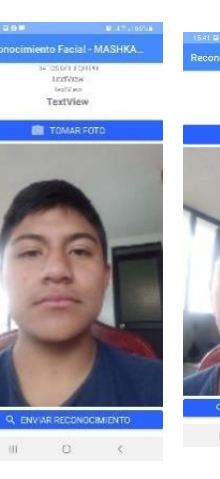
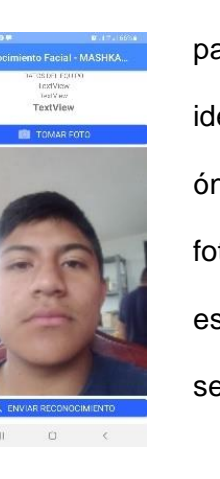
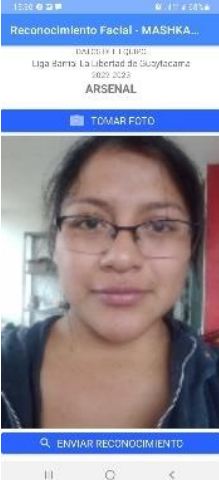
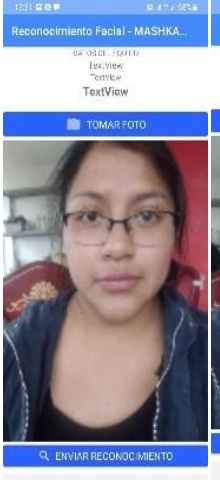

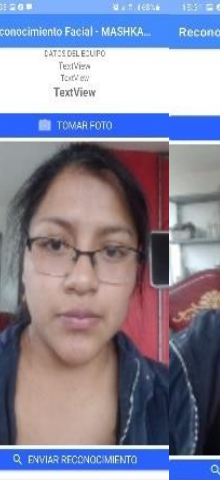
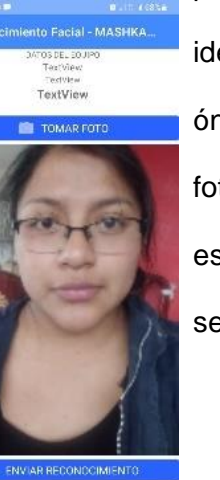
Estimación 1 Seg.	Estimación 2 Seg.	Estimación 2 Seg.	Estimación 1 Seg.	Estimación 1 Seg.	Tiempo estimado
----------------------	----------------------	----------------------	----------------------	----------------------	-----------------



para la identificación de fotografías es de: 1.4 seg.

para la identificación de fotografías es de: 1.4seg

Tiempo de Respuesta al detectar la fotografía

Estimación 2Seg.	Estimación 2Seg.	Estimación 1Seg.	Estimación 2 Seg.	Estimación 2Seg.	Tiempo estimado para la identificaci ón de fotografías es de: 1.8 seg
					
Estimación 2 Seg	Estimación 1 Seg.	Estimación 2 Seg.	Estimación 1 Seg.	Estimación 1 Seg.	Tiempo estimado para la identificaci ón de fotografías es de: 1.4 seg
					

Tiempo de Respuesta al detectar la fotografía

Estimación1	Estimación	Estimación	Estimación	Estimación	Tiempo estimado para la identificación de fotografías es de: 1.4 seg.
 <p>seg.</p>	2 Seg.	2 Seg.	2 Seg.	1 Seg.	

Porcentaje total de reconocimiento fotografías: 1.48

Nota. En la tabla se representa las pruebas que se realizó en el ambiente no controlado con 5 usuarios.

Tabla 30

Ambiente no controlado pruebas

Tiempo de Respuesta al reconocer a la persona					
Primera prueba	Segunda prueba	Tercera prueba	Cuarta Prueba	Quinta Prueba	Total, tiempo
Estimación 2 seg.	Estimación 1seg.	Estimación 1seg.	Estimación 1seg.	Estimación 2seg.	El tiempo estimado para la identificación de fotografías es de: 1.4 seg.

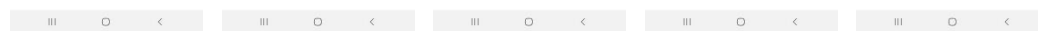
The image shows five sequential screenshots of a mobile application interface for facial recognition. Each screenshot displays the title 'Reconocimiento Facial - MASHKA...' and the identification result: 'JAIME EDILBERTO GUALSAQUI PINTA ARSENAL' with the ID number '1103705016'. Below the result, there are two blue buttons: 'VOLVER AL INICIO' (Return to Start) and 'VOLVER A EQUIPO' (Return to Team). The screenshots are arranged horizontally, showing the progression of the test results.

Tiempo de Respuesta al reconocer a la persona

Estimación 1 Seg. Estimación 2 Seg. Estimación 2 Seg. Estimación 1 Seg. Estimación 1 Seg. El tiempo estimado



para la identificación de fotografía es de: 1.4 seg.








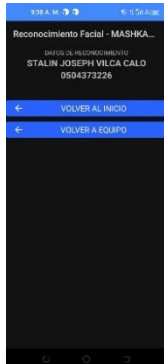




Estimación 2 Seg. Estimación 2 Seg. Estimación 1 Seg. Estimación 2 Seg. Estimación 2 Seg. El tiempo estimado



para la identificación de fotografía es de: 1.8 seg.



Tiempo de Respuesta al detectar la fotografía

Estimación 2 Seg.	Estimación 1 Seg.	Estimación 2 Seg.	Estimación 1 Seg.	Estimación 1 Seg.	El tiempo estimado para la identificación de fotografía es de: 1.4 seg
					
Estimación 1 Seg.	Estimación 1 Seg.	Estimación 2 Seg.	Estimación 2 Seg.	Estimación 1 Seg.	El tiempo estimado para la identificación de fotografía es de: 1.4 seg
					

Porcentaje total en la identificación de datos = 1.48

Nota. En la tabla se representa el tiempo de respuesta en la identificación de datos del jugador.

Fase 5 lanzamiento

Implementación del aplicativo

En la implementación del aplicativo fue realizad en la DRI en el mes de enero bajo la supervisión del ing. Oscar Toaquiza jefe de Área de Informática – Desarrollo creando un aplicativo para el sistema actual como preproducción debido a las políticas de tiene la Empresa Mashka-Futbol con respecto al aplicativo nuevo se va a incorporar a producción.

Capítulo IV

Conclusiones y recomendaciones

Conclusiones

- Identificar los procesos del aplicativo que se utilizaba en la Empresa Mashka-Futbol contribuyo a establecer los nuevos parámetros que se originaron al ejecutar las características de reconocimiento facial al aplicativo actual.
- Al utilizar el nuevo proceso generado se establecido una correcta administración de los datos obtenidos del reconocimiento , que a la vez ayuda asistir el ingreso de jugadores a la cancha de futbol es una herramienta útil para mejorar la identificación de jugadores al ingreso del polideportivo.
- La investigación ha cerca de las librerías de Python utilizadas para la gestión de identificación fue de gran ayuda para la correcta gestión y configuración en el aplicativo.
- El uso adecuado de las librerías de Python en el entorno de desarrollo permitió mejorar la identificaron de los usuarios. Esto significa que el sistema de reconocimiento puede ser más preciso y eficiente al identificar a las personas que ingresen al lugar
- La metodología XP es una excelente opción para el desarrollo del presente proyecto, ya que permitió una mayor colaboración entre el desarrollador, esto permitió trabajar de forma más eficiente y entregar el proyecto de calidad y a tiempo, logrando satisfacer los niveles de calidad establecidos por la Empresa Mashka Futbol.
- El uso del aplicativo de reconocimiento facial tiene un alcance relevante, ya que mediante estos se pretende identificar a los jugadores del polideportivo al momento de ingresar

Recomendaciones

- Es necesario analizar el software que se utilizó en el proyecto para que dicho análisis dependa de la incorporación en nuevas funciones sobre el mismo proyecto.
- En cuanto para un mejor uso del aplicativo es necesario leer el manual de usuarios donde se encuentra detallado los pasos a seguir para el funcionamiento de la aplicación.
- Se recomienda ampliar la investigación sobre las librerías de Python debido a que puede ofrecer muchas más funcionalidades que se podrían implementar en los proyectos futuros como un ejemplo la detección de sanciones a jugadores.
- En términos generales para considerar el reconocimiento facial del aplicativo tendría que ser de mayor precisión. Esto significa que el aplicativo deba ser capaz de detectar y rechazar intentos de fraude

Bibliografía

- AWS. (2022). *Datos sobre el reconocimiento facial mediante inteligencia artificial*. Obtenido de <https://aws.amazon.com/es/rekognition/the-facts-on-facial-recognition-with-artificial-intelligence/>
- Balla, P., & Jadhao, K. (2018). Sistema de seguridad de reconocimiento facial basado en IoT. *Conferencia internacional sobre ciudades inteligentes y tecnologías emergentes (ICSCET) de 2018*, 1-4. doi:10.1109/ICSCET.2018.8537344
- Gavilán, I. (2020). *Las cuatro fases del reconocimiento facial*. Obtenido de <https://ignaciogavilan.com/las-cuatro-fases-del-reconocimiento-facial/>
- Gavilánez, R. (2022). *Sistema automatizado para la gestión de asistencia y control de temperatura con reconocimiento facial de los empleados de la Empresa Eléctrica Ambato EEASA*. Ambato: Unversidad Técnica de Ambato.
- Hernández, R. (2010). *Introducción a los sistemas de control - Conceptos, aplicaciones y simulación con MATLAB*. México: Pearson.
- Inciso, J. (2021). *Desarrollo de un Sistema de Control de Acceso con Imagen y Temperatura de Trabajadores de la Empresa SEELDOIN E.I.R.L*. Lima: UTP.
- Jaramillo, D. (2021). Utilización del sistema de reconocimiento facial para preservar la seguridad ciudadana. *El Criminalista Digital*, 9,20-37, 20-37. Obtenido de <https://revistaseug.ugr.es/index.php/cridi/article/view/20899/20280>
- Jarrín, D., & Vinueza, D. (2020). *DESARROLLO DE UN SISTEMA PROTOTIPO DE PAGO DE FRUTAS A TRAVÉS DEL ENTRENAMIENTO DE REDES NEURONALES ARTIFICIALES CONVOLUCIONALES*. Quito: Escuela Politécnica Nacional.
- Lara, S., Albarracín, L., & Ponce, D. (2020). Prototipo de reconocimiento facial para mejorar el control de asistencia de estudiantes en UNIANDÉS, Quevedo. *Revista Arbitrada Interdisciplinaria KOINONIA*, 2(1), 60-72. doi:http://dx.doi.org/10.35381/r.k.v5i2.876

- Marín, R. (2018). CMS vs Framework. Diferencias, ventajas y desventajas. *INESEM*, 1.
- INTECHLAB. (30 de 05 de 2017). *Estimación de calidad de los algoritmos de reconocimiento facial*. Obtenido de <https://ntechlab.com/es/blog/2017/05/30/calidad-del-algoritmo-de-reconocimiento/#:~:text=Los%20algoritmos%20de%20reconocimiento%20facial%20est%C3%A1n%20entrenados%20en%20grandes%20conjuntos,algoritmo%20para%20cumplir%20la%20tarea.>
- Parada, P. (21 de 05 de 2021). *Qué son las Redes Neuronales Convolucionales*. Obtenido de <https://www.iebschool.com/blog/redes-neuronales-convolucionales-big-data/>
- Patiño, D., Silva, R., Munive, C., & Tobar, M. (2020). Prediction of Covid19 with the use of Random Forests Algorithm and Artificial Neural Networks. *ECUADORIAN SCIENCE JOURNAL*, 4(2), 101-110.
- Pereira, A., Donadom, T., & Oliveira, F. (2021). Aplicación de inteligencia artificial para monitorear el uso de mascarillas de protección. *Revista Científica General José María Córdova*, 19(33), 205-222.
- Terceño, A. (2017). *Análisis de un modelo predictivo basado*. Madrid: Universidad Complutense de Madrid.
- Untuña, V. (2022). *Sistema de control de acceso por medio de reconocimiento facial con uso de mascarilla y monitoreo de temperatura*. Ambato: Universidad Técnica de Ambato.
- A., O. A. (1 de abril de 2011, Aprobado el 10 de junio de 2011). Cuatro enfoques metodológicos para el desarrollo de Software RUP-MSF-XP-SCRUM. 15.
- Fuentes, J. R. (2015). *Desarrollo de Software Ágil. Extreme Programming y Scrum*. IT Campus Academy.
- Medrano, R. I. (23 de abril del 2008). Tutorial de Opencv.
- Mínguez, T. D. (2021). *Vision Artificial Aplicaciones prácticas con OPENCV- PYTHON*. Centro Español de Derechos Reprográficos: MARCOMBO,S.L.

Anexos