



Interface de visualización de señales de video generadas por un modelo de re-identificación de personas, utilizando características faciales y silueta corporal, a través de dos cámaras de video, a través de técnicas avanzadas de visión por computadora.

Chiluisa García, Jhonatan Javier y Fernández Bustos, Erick Alejandro

Departamento de Ciencias de Computación

Carrea de Ingeniería en Software

Trabajo de integración curricular, previo a la obtención del título de Ingeniero de Software

Ing. Jacho Guanoluisa, Nancy del Pilar

22 de agosto del 2023

Latacunga



Plagiarism and AI Content Detection Report

Tesina - Interfaz de re-identificación (...)

Scan details

Scan time:
August 23th, 2023 at 16:55 UTC

Total Pages:
90

Total Words:
22269

Plagiarism Detection



Types of plagiarism		Words
Identical	0.5%	113
Minor Changes	0.2%	37
Paraphrased	3%	678
Omitted Words	15%	3347

AI Content Detection



Text coverage		Words
AI text	5.6%	694
Human text	94.4%	40334

[Learn more](#)

Alerts: (1)

Cross Language: Same Document Language

Submitted language and cross-language text are the same language. No credits were used.

2/5 Severity



Plagiarism Results: (22)

T-EPEL-SOF-0089.pdf

0.9%

<https://repositorio.espe.edu.ec/bitstream/21000/35731/2/e...>

aevizute@espe.edu.ec

1 Carátula Sistema de re-identificación de personas a través de las características faciales y soft-biométricas el color de la vestimen...

Un recorrido por Scrum - Iniciativa Social

0.8%

<https://www.iniciativasocial.net/?p=1496>

Iniciativa Social ¡Bienvenido al blo...

Certified by

About this report
help.copyleaks.com

copyleaks.com

Ing. Jacho Guanoluisa, Nancy Del Pilar

C. C. 0501963086



Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Certificación

Certifico que el trabajo de integración curricular: **“Interfaz de visualización de señales de video generadas por un modelo de re-identificación de personas, utilizando características faciales y soft-biométricas como la textura, color y silueta corporal, a través de dos cámaras de video, a través de técnicas avanzadas de visión por computadora”** fue realizado por los señores **Fernández Bustos, Erick Alejandro** y **Chiluisa Garcia, Jhonatan Javier**, 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 analizada 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, 23 de agosto del 2023

Ing. Jacho Guanoluisa, Nancy Del Pilar

C. C. 0501963086



Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Responsabilidad de Autoría

Nosotros, **Fernández Bustos, Erick Alejandro y Chiluisa Garcia, Jhonatan Javier**, con cédulas de ciudadanía n° 0503814063 y 0550160295, declaramos que el contenido, ideas y criterios del trabajo de integración curricular: **Interfaz de visualización de señales de video generadas por un modelo de re-identificación de personas, utilizando características faciales y soft-biométricas como la textura, color y silueta corporal, a través de dos cámaras de video, a través de técnicas avanzadas de visión por computadora** es de mi/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, 23 de agosto del 2023

.....
Fernández Bustos, Erick Alejandro

C.C.: 0503814063

.....
Chiluisa Garcia, Jhonatan Javier

C.C.: 0550160295



Departamento de Ciencias de la Computación

Carrera de Ingeniería de Software

Autorización de Publicación

Nosotros, **Fernández Bustos, Erick Alejandro y Chiluisa Garcia, Jhonatan Javier**, con cédulas de ciudadanía n° 0503814063 y 0550160295, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Interfaz de visualización de señales de video generadas por un modelo de re-identificación de personas, utilizando características faciales y soft-biométricas como la textura, color y silueta corporal, a través de dos cámaras de video, a través de técnicas avanzadas de visión por computadora** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi/nuestra responsabilidad

Latacunga, 23 de agosto del 2023

.....
Fernández Bustos, Erick Alejandro

C.C.: 0503814063

.....
Chiluisa Garcia, Jhonatan Javier

C.C.: 0550160295

Dedicatoria

Este proyecto está dedicado a todas las personas que han brindado su apoyo incondicional a lo largo de este camino. Especialmente, a mi madre Fanny Yolanda Garcia Osorio y a mi padre Edwin Patricio Chiluisa Pazuña, quienes han sido pilares fundamentales para mi desarrollo personal y profesional, a mis hermanos por estar a mi lado en los momentos tanto buenos como difíciles. A mis amigos y amigas, les agradezco por sus palabras de aliento, las cuales me han proporcionado la fuerza necesaria para culminar esta etapa académica.

No puedo dejar de recordar y honrar a todos mis abuelitos que están en el cielo, quienes cada día me envían bendiciones y cuidan de mí.

También quiero expresar sincero agradecimiento al Ing. José Carrillo y la Ing. Nancy Jacho por su invaluable orientación, sabiduría y paciencia a lo largo de este proceso de investigación. El haber atendido a sus sugerencias y recomendaciones han sido fundamentales para la conclusión exitosa de este proyecto. Verdaderamente, espero que todo el sacrificio y dedicación invertidos en esta investigación contribuyan de manera significativa a la comunidad científica.

Jhonatan Javier Chiluisa Garcia

Ecuador, agosto 2023

Agradecimiento

Quiero comenzar expresando mi profundo agradecimiento Dios por haberme brindado la fuerza y la determinación necesarias para concluir con éxito este proyecto y lograr mi sueño de graduarme como profesional. También, deseo expresar mi más sincero agradecimiento a mis padres, Fanny Yolanda Garcia Osorio y Edwin Patricio Chiluisa Pazuña por haber proporcionado todo lo necesario para mi bienestar a lo largo de mi vida. Su apoyo incondicional ha sido la piedra angular que me ha permitido superar los obstáculos que se han presentado en mi camino.

Por último, agradezco a todas las personas que, de una forma u otra, han contribuido al desarrollo de este proyecto de investigación. El apoyo que me han brindado, ya sea a través de su colaboración o su aliento, han sido fundamentales para convertir este proyecto en una realidad.

Jhonatan Javier Chiluisa Garcia

Ecuador, agosto 2023

Dedicatoria

Con profundo cariño y gratitud dedico este trabajo a mis queridos padres, Luis Fernández y Soña Bustos, cuyo amor incondicional y sacrificios han sido mi inspiración constante, su confianza en mí ha sido la brújula que me ha guiado a lo largo de este camino académico. Cada paso que he dado, cada obstáculo que he superado, ha sido impulsado por su constante apoyo y aliento.

A Jenny Segovia, mi novia, por creer en mí cuando dudaba de mí mismo y por ser mi mayor fuente de inspiración. Ha sido mi compañera constante, mi confidente y mi mayor alegría. Este trabajo no solo es un logro académico, sino también un reflejo de nuestro amor y apoyo mutuo. Cada logro que alcance en la vida estará vinculado a ti.

Para mis queridas abuelitas, Teresa Laica y Mariana Laguaquiza, este logro es un tributo a la fuerza, el amor y la sabiduría que han tejido a lo largo de mi vida. Sus historias, consejos y cariño han sido la base sobre la cual he construido este camino académico.

Desde los días de pupitre, a mi querido amigo Mendelson Albán por sus bromas, su inquebrantable optimismo, sus palabras de aliento en los momentos difíciles, su apoyo y su capacidad para escuchar sin juzgar, han sido un confort que nunca dejare de apreciar.

Erick Alejandro Fernández Bustos

Ecuador, agosto 2023

Agradecimiento

Agradezco a todos los docentes de la carrera de Ingeniería de Software de la Universidad de las Fuerzas Armadas ESPE sede Latacunga, especialmente al ingeniero José Carrillo e ingeniera Nancy Jacho, estimados profesores universitarios, por su dedicación y experiencia como educadores. Sus lecciones, consejos y mentoría han sido un faro de conocimiento que ha guiado mi camino académico y personal.

A mis queridos padres, abuelitos, hermanos, tíos, tías, primos, primas, demás familia allegada, a mi novia y amigos, en este punto de mi vida, me siento profundamente agradecido por el amor, el apoyo y la fortaleza que siempre me han brindado. Su presencia ha sido un pilar inquebrantable en cada paso de mi camino, y vuestra generosidad y cariño han sido un regalo invaluable.

Erick Alejandro Fernández Bustos

Ecuador, agosto 2023

ÍNDICE DE CONTENIDO

Carátula.....	1
Reporte de verificación de contenido	2
Certificación	3
Responsabilidad de autoría	4
Autorización de publicación	5
Dedicatoria.....	6
Agradecimiento.....	7
Dedicatoria.....	8
Agradecimiento.....	9
Índice de contenido.....	10
Índice de figuras	13
Índice de tablas	16
Resumen	18
Abstract.....	19
Capítulo I: Introducción	20
Propósito y contextualización del tema	20
Justificación del interés de la investigación	22
Objetivos Generales y Específicos	23
<i>Objetivo General</i>	24
<i>Objetivos Específicos</i>	24
Metodología	24
Capítulo II: Marco Teórico	27

Interfaces de software para la visualización de sistemas de re-identificación de personas	27
Principios de diseño para el diseño de interfaces gráficas de usuario para la visualización de sistemas de re-identificación de personas.....	31
Metodología para el diseño de interfaces gráficas de usuario para la visualización de sistemas de re-identificación de personas	37
Lenguaje para desarrollar una interfaz gráfica de usuario que permita visualizar sistemas de re-identificación humana	43
Herramienta de creación de interfaces gráficas de usuario para sistemas de re-identificación de personas.	44
Herramienta de codificación para el desarrollo de una interfaz gráfica de usuario para la visualización de sistemas de re-identificación de personas	45
Características Biométricas	47
Características Soft-biométricas	47
Técnicas de Visión por Computador	50
Modelos y Algoritmos de Machine Learning	53
Fusión de características	56
Metodología para el desarrollo de la interfaz gráfica para la visualización de sistemas de re-identificación de personas	59
Capítulo III: Implementación del sistema	61
Análisis y diseño del sistema.....	63
<i>Análisis del sistema</i>	63
<i>Roles Scrum</i>	63
<i>Historias de Usuario</i>	64

<i>Product Backlog del proyecto</i>	66
<i>Herramientas software para el desarrollo de la interfaz</i>	67
<i>Ambiente controlado para la implementación de la Interfaz de visualización para un sistema de Re-identificación de personas</i>	69
<i>Definición e implementación de técnicas de visualización y optimización de recursos para captar modelos de re-identificación de personas</i>	72
<i>Sprint 01: Diseño de una interfaz multiplataforma, fácil de usar e intuitiva</i>	73
<i>Sprint 02: Visualización de señales de video generadas por un sistema de cámaras de video</i>	80
<i>Sprint 03: Visualización en tiempo real de la re-identificación de personas mediante el uso de características faciales y soft-biométricas, como la textura</i>	98
Capítulo IV: Validación del sistema	116
Marco de evaluación.....	116
Aplicación de métricas de usabilidad y experiencia de usuario	116
Aplicación de test de escala de usabilidad del sistema (SUS)	124
Capítulo V: Conclusiones	127
Bibliografía	129
Anexos	137

ÍNDICE DE FIGURAS

Figura 1 <i>Interfaz de línea de comandos Símbolo del sistema</i>	28
Figura 2 <i>Interfaz de realidad virtual</i>	29
Figura 3 <i>Interfaz de realidad aumentada enfocada a la educación</i>	30
Figura 4 <i>GUI de configuración (Windows 10)</i>	31
Figura 5 <i>Sitio web Coolors</i>	33
Figura 6 <i>Plantilla de un sistema de re-identificación de personas</i>	34
Figura 7 <i>Comparación de las tipografías serif y no serif</i>	35
Figura 8 <i>Qt Designer</i>	45
Figura 9 <i>Visual Studio Code</i>	46
Figura 10 <i>Representación del modelo holístico</i>	48
Figura 11 <i>Imagen de captura de rostro y su imagen con la proyección definida por Eigenfaces</i>	49
Figura 12 <i>Fisherfaces</i>	49
Figura 13 <i>Representación del modelo no holístico</i>	50
Figura 14 <i>Clasificador de imágenes utilizado para clasificar equipos de construcción</i>	51
Figura 15 <i>Detector de objetos usado para detectar una cámara</i>	52
Figura 16 <i>Seguidor de objetos utilizado para el seguimiento de automóviles</i>	53
Figura 17 <i>Reconocimiento facial mediante el método Haar Cascade</i>	54
Figura 18 <i>Transformación de imagen a LBP y su respectivo histograma</i>	55
Figura 19 <i>Representación de la arquitectura de YOLO</i>	56
Figura 20 <i>Identificación facial y captura de imagen</i>	56
Figura 21 <i>Mascaras corporales para la comparación con LBPH</i>	57
Figura 22 <i>Reconocimiento de persona</i>	58

Figura 23 Plano 2D del Bloque A, segundo piso de la Universidad de las Fuerzas Armadas ESPE sede Latacunga	71
Figura 24 Representación 3D de la ubicación de las camaras para capturar videos de prueba	72
Figura 25 Burndown Chart – Historia de Usuario 01	76
Figura 26 Implementación del diseño de la interfaz de vizualización	78
Figura 27 Bosquejo inicial de la interfaz de vizualización	79
Figura 28 Diseño de la interfaz de vizualización en Figma	80
Figura 29 Burndown Chart – Historia de Usuario 02	84
Figura 30 Conjunto de carpetas e imágenes generadas por el sistema de re-identificación de personas	85
Figura 31 Burndown Chart – Historia de Usuario 03	89
Figura 32 Visualización de las señales de video, coincidencias y descriptores en tiempo real	90
Figura 33 Conjunto de videos de prueba	91
Figura 34 Diseño para las entradas de video	92
Figura 35 Codificación para la visualización de las señales de video generadas por las dos cámaras de video	93
Figura 36 Código para extraer las rutas de las imágenes que se usan en la interfaz.....	95
Figura 37 Diseño para las coincidencias y descriptores.....	96
Figura 38 Código para actualizar las imágenes de las coincidencias y descriptores.....	97
Figura 39 Burndown Chart – Historia de Usuario 04	102
Figura 40 Burndown Chart – Historia de Usuario 05	106
Figura 41 Burndown Chart – Historia de Usuario 06	110

Figura 42 <i>Interfaz de visualización de señales de video generadas por un modelo de re-identificación a través de características faciales y soft-biométricas, como la textura</i>	111
Figura 43 <i>Visualización de la re-identificación facial y soft-biométrica</i>	113
Figura 44 <i>Visualización de información adicional para la identificación de una persona</i>	114
Figura 45 <i>Activación de la alerta al identificar una persona</i>	115
Figura 46 <i>Resultados representados en tasa de éxito al momento de realizar una tarea</i>	119
Figura 47 <i>Resultados de la medición del tiempo en que los usuarios logran un objetivo con la aplicación</i>	121
Figura 48 <i>Resultados de la puntuación del cuestionario de satisfacción</i>	123

ÍNDICE DE TABLAS

Tabla 1 <i>Test de SUS.</i>	40
Tabla 2 <i>Escala de Likert para evaluación de SUS</i>	41
Tabla 3 <i>Roles Scrum</i>	63
Tabla 4 <i>Historias de Usuario</i>	65
Tabla 5 <i>Product Backlog del Proyecto</i>	67
Tabla 6 <i>Herramientas de desarrollo</i>	68
Tabla 7 <i>Hardware necesario para implementar la interfaz</i>	73
Tabla 8 <i>Historia de usuario para la creación de la interfaz de visualización</i>	74
Tabla 9 <i>Sprint Backlog 01</i>	75
Tabla 10 <i>Historia de usuario para la visualización de señales de video generadas por cámaras de video</i>	82
Tabla 11 <i>Sprint Backlog 02</i>	83
Tabla 13 <i>Sprint Backlog 03</i>	88
Tabla 14 <i>Historia de usuario para la visualización en tiempo real de la re-identificación de personas, utilizando características faciales y soft-biométricas, como la textura.</i>	99
Tabla 15 <i>Sprint Backlog 04</i>	100
Tabla 16 <i>Historia de usuario para la visualización en tiempo real de información adicional en la re-identificación de personas</i>	103
Tabla 17 <i>Sprint Backlog 05</i>	105
Tabla 18 <i>Historia de usuario para la visualización de alertas automáticas al momento de identificar una persona</i>	107
Tabla 19 <i>Sprint Backlog 06</i>	108

Tabla 20 <i>Cuestionario para medir la eficacia</i>	117
Tabla 21 <i>Resultados de la aplicación de cuestionario de eficacia a los diferentes usuarios</i>	118
Tabla 22 <i>Cuestionario para medir la eficiencia</i>	120
Tabla 23 <i>Resultados en segundos de la aplicación de cuestionario de eficiencia a los diferentes usuarios</i>	120
Tabla 24 <i>Cuestionario para medir la satisfacción</i>	122
Tabla 25 <i>Resultados de la aplicación de cuestionario de satisfacción a los diferentes usuarios</i>	122
Tabla 26 <i>Resultados de la aplicación del test de SUS a 10 usuarios</i>	125

Resumen

En respuesta a la creciente necesidad de mejorar la seguridad y la identificación precisa de individuos, el presente proyecto aborda el desarrollo e implementación de una interfaz gráfica de usuario que permite la visualización de señales de video generadas por un avanzado modelo de re-identificación de personas que utiliza características fáciles y soft-biométricas como la textura. Esta investigación aprovecha técnicas avanzadas de procesamiento de imágenes y computación para lograr una identificación robusta y precisas en entornos controlados de vigilancia utilizando simultáneamente y en tiempo real dos cámaras de video. Se utiliza la metodología Scrum para el análisis y desarrollo de la interfaz de usuario, con el fin de adaptarse a los cambios de los requisitos, garantizar entregas progresivas y cumplir con los plazos establecidos. Las pruebas de la interfaz se llevaron a cabo en la sede Latacunga de la Universidad de las Fuerzas Armadas ESPE. Durante estas pruebas, se capturaron señales de video de las cámaras ubicadas en uno de los pasillos de la institución. La interfaz muestra la entrada de dos cámaras: la cámara 1 para el reconocimiento facial y la cámara 2 para la identificación de la parte posterior de la persona usando características soft-biométricas como la textura de la ropa. También presenta conjuntos de imágenes que representan coincidencias y descriptores que genera el sistema de re-identificación de personas. La validación de la interfaz tuvo como propósito cumplir con los requerimientos y potenciar la interacción y experiencia del usuario con el sistema de re-identificación. Esta mejora contribuye a lograr un seguimiento e identificación eficaces de personas, permitiendo una acción ágil por parte del personal de seguridad.

Palabras clave: interfaz gráfica, re-identificación de personas, características faciales, características soft-biométricas, reconocimiento facial.

Abstract

In response to the growing need for improved security and accurate identification of individuals, this project addresses the development and implementation of a graphical user interface that enables the display of video signals generated by an advanced person re-identification model that utilizes easy and soft-biometric features such as texture. This research leverages advanced image processing and computational techniques to achieve robust and accurate identification in controlled surveillance environments using two video cameras simultaneously and in real time. Scrum methodology is used for the analysis and development of the user interface, in order to adapt to changing requirements, ensure progressive deliveries and meet established deadlines. The interface tests were carried out at the Latacunga headquarters of the Universidad de las Fuerzas Armadas ESPE. During these tests, video signals were captured from cameras located in one of the corridors of the institution. The interface shows the input from two cameras: camera 1 for facial recognition and camera 2 for identification of the back of the person using soft-biometric features such as clothing texture. It also presents sets of images representing matches and descriptors generated by the person re-identification system. The interface validation was intended to meet the requirements and enhance the user's interaction and experience with the re-identification system. This improvement contributes to achieve an effective tracking and identification of persons, allowing an agile action by security personnel.

Keywords: graphical interface, person re-identification, facial features, soft-biometric features, face recognition.

Capítulo I

Introducción

Propósito y contextualización del tema

En los últimos años, se ha popularizado el uso de los sistemas de videovigilancia que funcionan con cámaras de vídeo (de gama baja, media o alta) colocadas en lugares estratégicos para extraer información importante, (Lignia Tuapanta & Pichucho Tandalla, 2023). Dicha información debe presentarse de forma intuitiva mediante una interfaz de visualización de señales de vídeo.

Según estudios recientes, este tipo de sistemas se ha utilizado en la medicina, el transporte, el deporte, ciudades inteligentes, búsqueda forense multicámara (Manosalvas Salazar, 2017; Liao et al., 2020). La videovigilancia se utiliza hoy en día en contextos abiertos como cerrados, abarcando una amplia gama de lugares como aeropuertos, cárceles, escuelas, carreteras, barrios residenciales, así como edificios comerciales y oficinas.

En un sistema de videovigilancia convencional gestionado por el personal operativo, resulta poco práctico examinar todas las escenas debido al excesivo volumen de flujos de video. Por consiguiente, es esencial disponer de sistemas automatizados que realicen este análisis en tiempo real, lo que permite una vigilancia continua y aumenta considerablemente la eficacia de la protección contra sucesos que podrían poner en peligro a las personas.

Actualmente, los sistemas de videovigilancia permiten la re-identificación de personas, que puede describirse como el reconocimiento de un individuo en diferentes ubicaciones a través de una red de cámaras que no se superponen (Camps et al., 2017; Ren et al., 2020).

Las características faciales y soft-biométricas como la textura son dos aspectos fundamentales para la identificación de personas, ya que proporcionan información única y distintiva. Sin embargo, procesar y visualizar estas señales de video generadas por modelos de re-identificación de personas requiere una interfaz de usuario versátil con patrones de diseño limpios y diseño responsivo.

Generalmente, los sistemas de videovigilancia son monitoreados por los agentes de control los cuales se encargan de observar varias cámaras de seguridad las 24 horas de los 365 días del año. Un ejemplo real es el personal operativo de las salas de videovigilancia del ECU 911 que están a cargo del sistema interconectado de cámaras de videovigilancias desplegadas en varias ciudades del Ecuador para aumentar la seguridad de la población.

Es posible que los agentes de control no estén al tanto de todas las circunstancias que tienen lugar en los distintos escenarios. Si detectan alguna anomalía en una escena, su atención se desvía de lo que está sucediendo en las demás, disminuyendo así la visibilidad global. “Esto conlleva al inadecuado uso de la información y la ineficacia en el monitoreo” (Muñoz Flórez, 2017, pág. 7).

El implementar una interfaz de visualización bien diseñada, que integre todas las fases de la re-identificación de personas, proporciona una forma intuitiva y eficiente de acceder a la información relevante (Ren et al., 2020). Permite a los operadores o usuarios del sistema visualizar de manera rápida y clara el proceso de detección y los resultados de la re-identificación, como las coincidencias de personas o descriptores. Esto facilita la toma de decisiones y agiliza las respuestas en situaciones de seguridad o monitoreo.

El sistema sobre el cual se va a implementar la interfaz de usuario comprende dos importantes fases: la identificación de personas, mediante el rostro, obtenido por la cámara 1, y la detección y Re-

Identificación de personas mediante las características soft-biométricas, textura, color y silueta corporal, identificada por la cámara 2, en donde no se detecta el rostro.

Debido a que el sistema está dividido por fases, la interfaz de usuario puede utilizar hilos de procesamiento (thread) con el objetivo de mejorar la capacidad de respuesta en la re-identificación y la experiencia del agente de control al interactuar con el sistema. Los mismos también pueden ser usados para actualizar la interfaz de manera asíncrona mientras se realiza tareas en segundo plano.

La interfaz también se divide en diferentes partes para visualizar de forma eficiente la entrada de vídeo de la cámara 1, las coincidencias y los descriptores con sus metadatos asociados, como el nombre, el índice de precisión del modelo, y la salida de vídeo de la cámara 2, donde se ve la re-identificación de la persona.

A la luz de lo anterior, se sugiere el desarrollo de una interfaz de visualización para las señales de vídeo generadas por un modelo de re-identificación humana. En la interfaz se utilizarán dos cámaras de vídeo y elementos biométricos blandos como la textura y el reconocimiento facial. Además, la interfaz permite conocer visualmente las predicciones del modelo y el proceso de re-identificación de individuos.

Justificación del interés de la investigación

Los operadores y analistas de seguridad podrán identificar y seguir a personas de interés con mayor rapidez y precisión gracias al desarrollo de una interfaz de visualización sencilla y clara. Una interfaz de visualización acelera la eficacia operativa y reduce la curva de aprendizaje durante la re-identificación y el seguimiento de personas utilizando dos cámaras de vídeo mediante la explotación de las señales visuales producidas por el modelo de re-identificación.

Diseñar una interfaz de visualización adecuada para los sistemas de reidentificación humana que se basan en rasgos faciales y biométricos blandos como la textura tiene una serie de usos beneficiosos y prácticos. Reducen los errores humanos cuando se utilizan en lugares que requieren un alto nivel de seguridad, como aeropuertos, estaciones de tren y zonas comunes donde se reúne mucha gente. Pueden ser útiles para gestionar multitudes en grandes eventos, encontrar sospechosos en investigaciones criminales e identificar sospechosos.

Este estudio pretende desarrollar una interfaz visual para las señales de vídeo producidas por un modelo de re-identificación humana. Este modelo utiliza características faciales y biométricas blandas como la textura, la silueta y el color, y funciona a partir de un circuito cerrado de cámaras de vídeo.

Para alcanzar el objetivo, se tiene previsto utilizar hilos de procesamiento (Thread) con la finalidad de consumir eficazmente las señales generadas por el sistema de Re-ID, mejorando así la eficacia y la capacidad de respuesta de la interfaz.

La capacidad de gestionar y analizar varios flujos de video simultáneamente utilizando Threads, permite la presentación en tiempo real de numerosas señales de video producidas por el modelo de re-identificación, lo que se traduce en una mejora sustancial del seguimiento y la detección de personas. Además, este proyecto no sólo cumplirá su objetivo principal, sino que también ofrecerá consejos útiles para el desarrollo de otros proyectos centrados o relacionados con las interfaces de visualización en sistemas de re-identificación de personas.

Objetivos Generales y Específicos

A continuación, se presentan los objetivos generales y específicos de esta actividad de integración curricular.

Objetivo General

Desarrollar una interface de visualización de señales de video generadas por un modelo de re-identificación de personas, utilizando características faciales, soft-biométricas como la textura, color, silueta y técnicas avanzadas por computadora a través de dos cámaras de video.

Objetivos Específicos

- Conocer y analizar el estado del arte sobre las técnicas de visualización, optimización de recursos e interfaces de software para la visualización de sistemas de re-identificación de personas.
- Implementar una interface de visualización de señales de video generadas por modelos de re-identificación de personas, utilizando las características faciales y soft-biométricas como la textura, usando técnicas avanzadas de visión por computadora.
- Validar los resultados, corregir errores y mejorar la usabilidad de la Interfaz para un sistema de re-identificación de personas.
- Redactar el escrito de la tesina para constatar el proceso de desarrollo de la interfaz de visualización para un sistema de re-identificación de personas.

Metodología

Este trabajo tiene como objetivo primordial, el desarrollo de una interface de visualización de señales de video que es generado por un modelo de re-identificación de personas, utilizando características faciales y soft-biométricas como la textura, a través de dos cámaras de video, con ayuda de técnicas avanzadas de visión por computadora.

Para lograr este propósito, es fundamental emplear una metodología que se ajuste a los objetivos planteados para esta investigación. En este sentido, se opta por utilizar el enfoque sistemático,

que permite la búsqueda y revisión exhaustiva de la bibliografía académica pertinente al tema de investigación, sentado así la base de los conocimientos teóricos iniciales.

Con ese fin, se empieza con el análisis de las técnicas de visualización y optimización de recursos existentes para mostrar señales de video generadas por los modelos de re-identificación de personas, después de esto, se puede continuar con el estudio y análisis de las diferentes interfaces de software que tengan relación con la visualización de sistemas de re-identificación (Re-ID).

Investigar y revisar artículos académicos que se encuentran en bases de datos reconocidas como Scopus, Web of Science e IEEE Xplore puede ayudarnos a recopilar los datos necesarios. Este enfoque contribuirá significativamente a fortalecer la comprensión de la temática de investigación, proporcionando una base sólida de conocimiento respaldada por fuentes científicas de alta calidad.

A continuación, se hará un diseño o mockup de la interface usando herramientas software que ayuden con el diseño de interfaces graficas como lo es Figma. Este diseño debe tener secciones para poder distribuir de mejor manera los componentes que tendrá la interface en toda la ventana principal.

Posteriormente, se realiza un análisis y revisión de código fuente del sistema al cual vamos a integrar la interface, de esta manera, se logrará entender el flujo de trabajo del sistema, librerías usadas, entorno de desarrollo y lenguaje de programación utilizado. Después, de entender cómo funciona el sistema, se indagará y probará una librería o framework que se adapte al sistema y con la cual se pueda desarrollar aplicaciones gráficas de usuario (GUI).

Por último, para visualizar los datos y las normas de validación, la interfaz se conecta a varios sistemas multicámara de re-identificación de personas. Esto se utilizará para evaluar la interfaz de

visualización con el fin de identificar cualquier fallo que conduzca a errores y, en función de las conclusiones, mejorar el rendimiento de la supervisión de la re-identificación de personas.

Capítulo II

Marco teórico

En este capítulo se analizarán en profundidad los términos relacionados con la comprensión de una aplicación informática con interfaz de visualización de señales de vídeo producidas por un modelo de re-identificación de personas. Se incluirán temas como, las diferentes interfaces de software existentes, principios de diseño de interfaces, una metodología de diseño de interfaces centrada en el usuario, principios de usabilidad, escala de usabilidad del sistema, experiencia de usuario y sus elementos.

En cuanto a la escritura de código de la interfaz de software, se presenta un lenguaje de programación frecuentemente usado en la actualidad, que tiene una herramienta de diseño que facilita la creación de interfaces. Todo esto es posible gracias a un editor de código, que es parte fundamental de este capítulo.

Para comprender el funcionamiento la interfaz internamente, se define características biométricas y soft-biométricas, técnicas holísticas y no holísticas, técnicas de Visión por Computador, modelos y algoritmos de Machine Learning, y la fusión de todas estas características para lograr la re-identificación.

Finalmente, en lo que respecta a la metodología de desarrollo de software, se usa Scrum y se detalla brevemente en esta sección.

Interfaces de software para la visualización de sistemas de re-identificación de personas

Una Interfaz de software es un conjunto de componentes de programas de computador que muestran información al usuario y le brindan la posibilidad de interactuar tanto con la información,

como con el propio equipo computacional. Se compone principalmente de dos elementos: la entrada y la salida, (Albornoz, 2014).

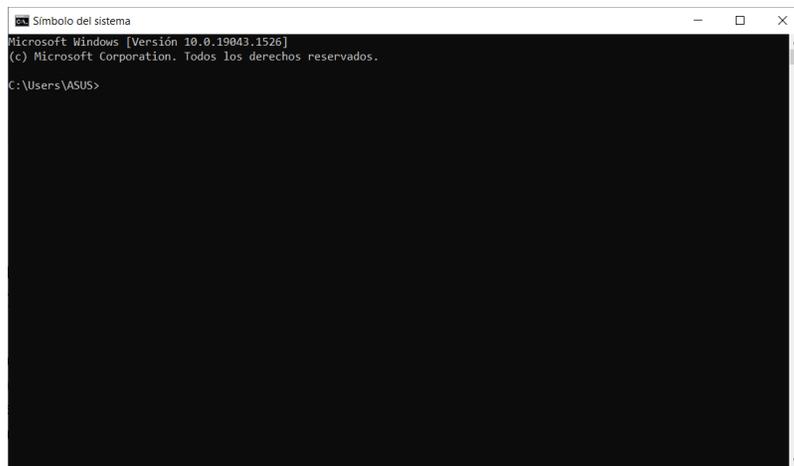
En la siguiente sección, se detallan los diferentes tipos interfaces de software en las que es posible implementar un sistema de re-identificación de personas.

Interfaz de línea de comando

Las interfaces de línea de comando se basan en el ingreso de texto. El modo principal de interactuar con las CLI es a través del texto. Los usuarios introducen comandos, que son líneas de texto que siguen una sintaxis específica, y reciben una respuesta del comando en forma de líneas de texto. (Sampath et al., 2021).

Figura 1

Interfaz de línea de comandos Símbolo del sistema.



Interfaz de realidad virtual

Una Interfaz de realidad virtual ofrece la posibilidad de sumergir a los usuarios en un entorno tridimensional donde pueden observar y participar en interacciones en entornos remotos o compartidos

en tres dimensiones Figura 2. Esto puede facilitar una mayor conciencia de la situación y una interacción más fluida, (Wonsick & Padir, 2020).

Figura 2

Interfaz de realidad virtual.



Nota. Recuperado de (Haya, 2019).

Interfaz de realidad aumentada

Una interfaz de realidad aumentada (RA) es aquella que muestra gráficos interactivos generados por computadora directamente en objetos reales. En las interfaces de realidad aumentada, se agregan imágenes u objetos virtuales en 3D que se superponen a objetos reales para complementarlos. Las aplicaciones de AR suelen utilizar pantallas portátiles o montadas en la cabeza para mostrar los gráficos generados por computadora en el entorno del usuario, (Poupyrev et al., 2002) (Billinghurst et al., 2005).

Figura 3

Interfaz de realidad aumentada enfocada a la educación.



Nota. Recuperado de (Reina, 2021).

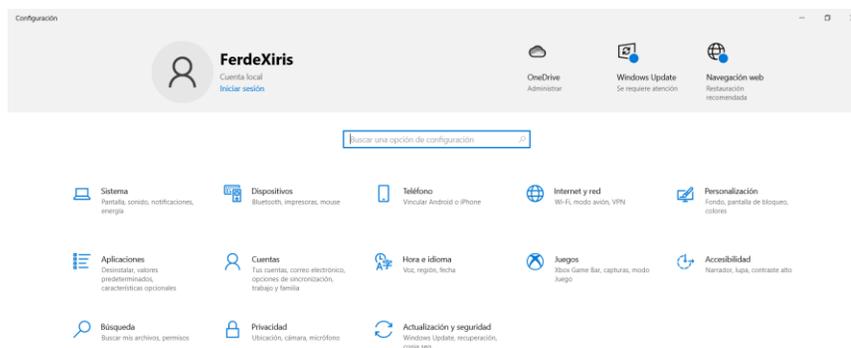
Interfaz gráfica de usuario

Para facilitar la interacción usuario-sistema, una interfaz gráfica de usuario (GUI) utiliza las capacidades gráficas de los ordenadores, y al mismo tiempo oculta al usuario las particularidades del lenguaje de programación. (Martinez, 2011).

Un gráfico de interfaz de usuario utiliza elementos gráficos como herramienta de comunicación, Figura 4. Los widgets, que son elementos gráficos, suelen activarse cuando un usuario hace clic en ellos con el ratón u otra herramienta de seguridad (Jansen, 1998).

Figura 4

GUI de configuración (Windows 10).



Al utilizar una interfaz gráfica de usuario correctamente diseñada, aumenta la interacción entre los usuarios y el sistema de acuerdo con las normas de usabilidad especificadas en la norma ISO-9241-11 (Bevan et al., 2015). Este tipo de interfaces, por su apariencia, facilitan la navegación y la comprensión de los resultados. En consecuencia, para este proyecto se decidió desarrollar una interfaz gráfica de usuario que permitiera visualizar un sistema de re-identificación.

Es necesario tener en cuenta algunos de los conceptos de diseño que intervienen en la producción de una interfaz gráfica, a continuación, se explican algunos de estos conceptos.

Principios de diseño para el diseño de interfaces gráficas de usuario para la visualización de sistemas de re-identificación de personas

La visualización de los sistemas de re-identificación de personas depende en gran medida del diseño de las interfaces gráficas de usuario. Se requiere una interfaz gráfica de usuario intuitiva, fácil de usar y estéticamente agradable. Para lograrlo, se utilizan diversos conceptos de diseño para mejorar la

usabilidad de la interfaz y maximizar la experiencia del usuario, entre los que se incluyen los que se enumeran a continuación.

Anticipación

Una interfaz gráfica de usuario debe anticiparse a las necesidades y mostrar en pantalla lo que el usuario necesita para interactuar con el sistema, en lugar de exigirle que busque la solución (Gómez, 2000).

Autonomía

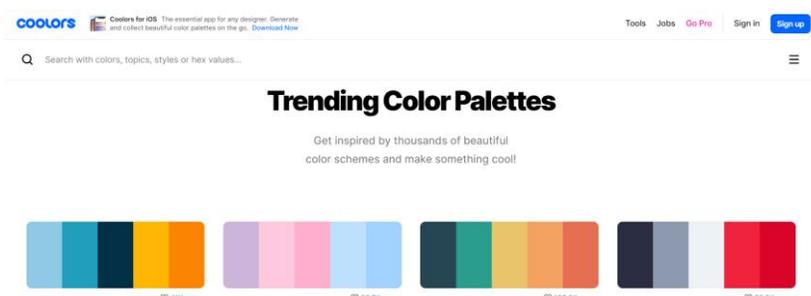
La interfaz gráfica de usuario debe ser tal que el usuario pueda aprender a utilizar el programa rápidamente (Gómez, 2000).

Visualización del color

Los colores influyen en muchos aspectos de nuestra vida. Utilice una paleta de colores limitada al crear interfaces gráficas de usuario, como se ve en la Figura 5. Los colores apagados, delicados y complementarios funcionan bien en este contexto. Los colores apagados, delicados y complementarios funcionan bien en este contexto. El uso de paletas de colores confiere a la interfaz gráfica de usuario uniformidad, regularidad y un aspecto profesional (Romero, 2012).

Figura 5

Sitio web Colors.



Nota. Recuperado de (Bianchi, 2023).

Hay que tener en cuenta a los usuarios que tienen problemas para ver los colores. Es esencial distribuir los componentes en una pantalla cuando se presenta mucha información para que el usuario pueda verlo todo con claridad y sin sentirse abrumado (Albornoz, 2014).

Curva de aprendizaje

Lo ideal es que la curva de aprendizaje sea lo más baja posible, para que incluso los usuarios no cualificados puedan utilizar la interfaz sin dificultad (Gómez, 2000).

Reducción de latencia

Cuando se utiliza la multitarea, se puede ahorrar el tiempo transcurrido entre un estímulo y su reacción. Las técnicas multitarea permiten al usuario seguir trabajando mientras las tareas de transmisión y procesamiento de datos se realizan en segundo plano (Gómez, 2000).

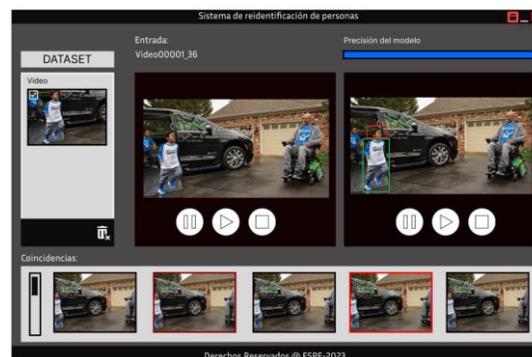
Disposición de los elementos de la interfaz

Para diseñar una plantilla que permita personalizar los datos de los cuadrantes de la interfaz gráfica de usuario, es importante tener en cuenta la distribución por edades, los intereses, el nivel de

educación y los campos de estudio pertinentes del público objetivo. Los puntos clave de enfoque se eligen en función de lo que se quiera destacar en el área de diapositivas, así como en el material que allí se muestra, (Romero, 2012).

Figura 6

Plantilla de un sistema de re-identificación de personas.



En Occidente, donde la información se lee de arriba abajo y de izquierda a derecha, la estructura de la información, debe tener en cuenta la cultura imperante. Esto determina el orden preferido de posición de los elementos principales de la interfaz gráfica de usuario. Se aconseja preservar la coherencia estructural en el diseño de la interfaz, para organizar todo el material, lo que puede hacerse utilizando un diseño de plantilla prefabricados, (Romero, 2012).

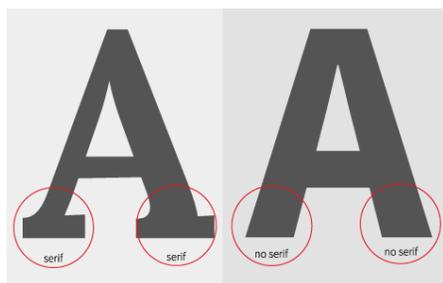
Tipografía

Debe ser posible leer el tipo y el tamaño de la fuente, y el texto debe ser de un color distinto al del fondo (lo ideal es utilizar letras negras sobre fondos claros). Se recomienda utilizar fuentes sin remates para mejorar la legibilidad en pantalla, para reducir el impacto del pixelado y proporcionar un aspecto más nítido en la interfaz gráfica de usuario Figura 7, (Romero, 2012).

Al utilizar el tipo de letra Arial en un tamaño de 20 a 25 puntos, mejora la legibilidad del texto de la interfaz y seleccionar un tamaño de letra más grande y llamativo para los encabezados, un tamaño como el mencionado ofrece una mejor legibilidad. La coherencia de la interfaz puede verse afectada si se utilizan más de tres tipos y tamaños de letra (Romero, 2012) (Albornoz, 2014).

Figura 7

Comparación de las tipografías serif y no serif.



Nota. Recuperado de (Valenzuela, 2016).

Imágenes

Las imágenes son necesarias en una interfaz gráfica de usuario para comparar y evaluar el reconocimiento facial y corporal, que permite la identificación de personas. Al ofrecer una representación visual completa y precisa de las personas, el uso de imágenes en una interfaz gráfica de usuario mejora significativamente la precisión y la eficacia de los resultados.

Porcentaje de confianza

Al mostrar el porcentaje de confianza en la interfaz gráfica de usuario, el objetivo es proporcionar al usuario una opinión imparcial sobre el grado de confianza de una identificación. Permite evaluar y comprobar la corrección del sistema y comprender cómo se han generado los resultados. El

nivel de confianza de cada resultado da una idea clara de la precisión de la identificación que se ha logrado.

Visualización de cámaras

La ventaja de utilizar la visualización de la cámara en tiempo real en la interfaz gráfica de usuario es que el proceso de re-identificación puede completarse de forma rápida y automática. Esto implica que los usuarios pueden identificar con rapidez y precisión a las personas observando su entorno en tiempo real, lo que resulta crucial en situaciones que exigen una respuesta rápida.

Resultados

La representación visual del resultado de la re-identificación mediante una imagen del individuo reconocido permite al usuario confirmar o verificar con mayor precisión y facilidad la identidad de una persona.

Familiaridad del usuario

Los elementos manejados por el sistema deben ser pertinentes para la tarea en cuestión, y el lenguaje y los elementos visuales utilizados en la interfaz gráfica de usuario deben ser sencillos para que el usuario los entienda (Albornoz, 2014).

Uniformidad de la interfaz gráfica de usuario

Se tarda menos en aprender a utilizar los elementos de la interfaz porque todos tienen que seguir la misma estructura (Albornoz, 2014).

Mínima sorpresa

En la interfaz gráfica de usuario deben evitarse comportamientos inesperados que puedan molestar al usuario y hacerle perder el interés de utilizar la aplicación (Albornoz, 2014).

Metodología para el diseño de interfaces gráficas de usuario para la visualización de sistemas de re-identificación de personas

La realización de tareas, la productividad y el placer del usuario pueden mejorar con una interfaz gráfica de usuario bien diseñada. Es crucial seguir los métodos que se muestran a continuación para crear un diseño GUI adecuado.

Diseño centrado en el Usuario

El diseño centrado en el usuario es una técnica que implica a los usuarios en el centro de atención para la creación de interfaces. Con el fin de aumentar la comprensión de las demandas de los usuarios, se repiten iterativamente las etapas de diseño y evaluación (Sastoque et al., 2016).

1. **Análisis:** La identificación de los usuarios de la interfaz gráfica de usuario, sus demandas y el entorno en el que se desarrollarán las actividades se llevan a cabo durante la fase de análisis. En esta fase se describen los objetivos, las dificultades, las tareas, la recogida de información y las acciones que se prevé que lleven a cabo los usuarios al utilizar la interfaz gráfica de usuario propuesta.
2. **Diseño:** Los criterios esenciales para el diseño se describen detalladamente durante esta fase, en la que también se prioriza cada componente según su importancia. Se realizan prototipos utilizando los requisitos durante las fases necesarias del proceso de diseño, lo que permite evaluar la solución propuesta.

3. **Evaluación:** La etapa de evaluación se enfoca primordialmente en la participación de los usuarios para evaluar los prototipos creados en la fase previa. Su propósito es desarrollar un prototipo de interfaz gráfica de usuario que siga los principios del diseño centrado en el usuario, sometiéndolo a pruebas con usuarios y adaptándolo en base a sus aportes a través de iteraciones.
4. **Implementación:** La implementación es el paso en el que la solución prevista se construye y desarrolla de forma concreta a partir de los resultados de las fases de análisis, diseño y evaluación.
5. **Despliegue:** En esta última fase del proceso de diseño, el producto se devuelve al usuario final para que lo utilice.

Principios de usabilidad

La usabilidad es una característica de una interfaz gráfica de usuario que permite la representación eficaz de un sistema de re-identificación de personas, y se refiere a lo bien que el usuario encuentra la interfaz para utilizarla. La calidad de la interacción del usuario con la interfaz gráfica de usuario se conoce como usabilidad (Sastoque et al., 2016).

Como resultado, se han definido las siguientes características de usabilidad:

1. **Eficacia:** Es la precisión e integridad con la que los usuarios logran determinados objetivos. La calidad de las soluciones y la tasa de éxito son ejemplos de indicadores de eficacia (Frøkjær et al., 2000).

2. **Eficiencia:** Es la relación entre la precisión, el esfuerzo y los recursos empleados por los usuarios para alcanzar determinados objetivos. La eficiencia puede medirse observando los tiempos de finalización y aprendizaje de las tareas (Frøkjær et al., 2000).
3. **Satisfacción:** La comodidad y los comentarios positivos de los clientes sobre el uso del sistema son factores clave de la satisfacción (Frøkjær et al., 2000).

También hay que tener en cuenta los siguientes principios de usabilidad al momento de desarrollar:

- **Enfoque temprano en Usuarios y Tareas:** Comprender a los usuarios y las tareas que se llevarán a cabo utilizando la interfaz es crucial para darse cuenta de que un ser humano posee una variedad de rasgos cognitivos, conductuales y físicos, todos los cuales afectan a las tareas que ejecutará (Sastoque et al., 2016).
- **Mediciones Empíricas:** Como se indica en el proceso de evaluación del diseño centrado en el usuario, los usuarios deben participar en las evaluaciones de los prototipos. Para evaluar a los usuarios, recopilar datos y cuantificar la usabilidad empírica, deben estar conectados a la tarea en un contexto real (Sastoque et al., 2016).
- **Diseño Iterativo:** Al crear una interfaz gráfica de usuario, es fundamental seguir las etapas del diseño centrado en el usuario, iterando sobre las fases de diseño y evaluación tantas veces como sea necesario. (Sastoque et al., 2016).

System Usability Scale (SUS). Escala de usabilidad del sistema.

System Usability Scale (SUS) ofrece un indicador rápido de la calidad, se utiliza con frecuencia para evaluar la usabilidad de los sistemas. Diez preguntas componen la encuesta SUS, que se muestra en

la Tabla 1, cada pregunta utiliza una escala Likert para ser evaluada, que se describe en la Tabla 2 (Grier et al., 2013).

Tabla 1

Test de SUS.

Preguntas	
Q1	Me gustaría utilizar este sistema con más frecuencia
Q2	El sistema me pareció muy complejo
Q3	El sistema me pareció fácil de usar.
Q4	Necesito el apoyo de un técnico para poder utilizar este sistema.
Q5	Las distintas funciones de este sistema estaban bien integradas.
Q6	Existe demasiada incoherencia en este sistema.
Q7	La mayoría de la gente aprendería a utilizar este sistema rápidamente.
Q8	El sistema me pareció muy complicado de utilizar.
Q9	Me sentí muy seguro utilizando el sistema.
Q10	Necesito aprender muchas cosas antes de utilizar este sistema.

Tabla 2*Escala de Likert para evaluación de SUS.*

N°	Enunciado	Valor
1	Totalmente en desacuerdo.	1
2	En desacuerdo.	2
3	Ni de acuerdo ni en desacuerdo.	3
4	De acuerdo.	4
5	Totalmente de acuerdo.	5

El cálculo de la puntuación del Sistema de Usabilidad (SUS), se realiza sumando las contribuciones de cada puntuación de los elementos, en donde Q_n es el número de pregunta y E_n es el valor de la posición de la escala de Likert en respuesta a la pregunta (Brooke, 1995):

Para los elementos $Q_{impares}$, la contribución se calcula restando 1 al valor de la posición en la escala $E_{impares}$. Dado que hay 5 elementos en esta categoría, se restan 5 puntos a la sumatoria de $E_{impares}$, lo que se resume en la siguiente fórmula, en donde $R_{impares}$ es el resultado:

$$R_{impares} = \sum_{E_{impares}} x - 5$$

$$R_{impares} = (E_1 + E_3 + E_5 + E_7 + E_9) - 5$$

De diferente manera, para cada una de Q_{pares} , la contribución equivale a la resta de 5 menos el valor de la posición en la escala E_{pares} . Dado que hay cinco elementos en esta categoría, la suma total

resulta en 25 y se resta la sumatoria total de los valores de las posiciones en la escala E_{pares} . La fórmula equivalente es la siguiente en donde R_{pares} es el resultado:

$$R_{pares} = 25 - \sum x_{E_{pares}}$$

$$R_{pares} = 25 - (E_2 + E_4 + E_6 + E_8 + E_{10})$$

Finalmente, se procede a multiplicar la suma de los resultados por 2,5 para obtener el valor total del puntaje de System Usability (SU) que determina el nivel de satisfacción de la interfaz, de la siguiente manera:

$$SU = (R_{impares} + R_{pares}) \cdot 2.5$$

Una puntuación de 85 o más se considera una gran usabilidad, mientras que una puntuación inferior a 70 se considera una usabilidad insatisfactoria, (Grier et al., 2013).

Experiencia de Usuario UX (User Xperience)

Experiencia de usuario (UX) es el término utilizado para describir las sensaciones, respuestas y experiencias que tiene una persona cuando interactúa con una aplicación informática. El modelo UX Honeycomb, que representa gráficamente un panel con siete celdas, se ha utilizado para ayudar en la creación de estas experiencias. Cada una de las siete celdas del modelo comprende los numerosos componentes y elementos esenciales de una experiencia de usuario positiva (Sastoque et al., 2016). A continuación, se listan estos elementos principales.

- **Útil:** El diseño debe ofrecer una funcionalidad totalmente nueva y mejor que las alternativas existentes para que el usuario alcance sus objetivos reales (Sastoque et al., 2016).

- **Deseable:** La respuesta debe darse de forma que el usuario se sienta a gusto con la solución sugerida (Sastoque et al., 2016).
- **Accesible:** Un amplio abanico de personas debe poder utilizar el diseño. La curva de aprendizaje debe ser lo más corta posible con respecto a los conceptos y soluciones existentes (Sastoque et al., 2016).
- **Creíble:** Los elementos de diseño deben inspirar confianza y fe en la solución propuesta (Sastoque et al., 2016).
- **Alcanzable:** La disposición de los componentes de diseño debe facilitar a los usuarios su búsqueda y utilización. A pesar de los obstáculos para el éxito y la productividad, los usuarios deben poder utilizar la solución de inmediato hasta que se hayan habituado completamente a ella (Sastoque et al., 2016).
- **Utilizable:** El diseño debe funcionar correctamente y carecer de fallos o defectos tecnológicos graves (Sastoque et al., 2016).
- **Valioso:** El diseño debe aportar ventajas y ser ventajoso para el usuario. Para estar satisfecho, el usuario buscará el beneficio de la aplicación (Sastoque et al., 2016).

Lenguaje para desarrollar una interfaz gráfica de usuario que permita visualizar sistemas de re-identificación humana

En el ámbito de la visión por ordenador, la re-identificación de seres humanos se está desarrollando rápidamente. En los proyectos complicados de este sector, es esencial elegir el lenguaje de programación adecuado para crear una interfaz gráfica de usuario.

Python

El amplio ecosistema de módulos y frameworks de Python, que facilita la creación de interfaces gráficas de usuario dinámicas y atractivas, es una ventaja definitiva. Python es fácil de aprender y de resolver problemas durante el proceso de desarrollo gracias a su amplia comunidad de desarrolladores y a su abundante documentación. Los desarrolladores disponen de numerosos foros de debate, tutoriales y ejemplos de código como recursos complementarios (González Duque, 2011).

Herramienta de creación de interfaces gráficas de usuario para sistemas de re-identificación de personas.

Para crear una interfaz gráfica de usuario es necesario utilizar herramientas de desarrollo de software especializadas. Estas herramientas facilitan la manipulación de los elementos gráficos necesarios para el funcionamiento de la interfaz. La herramienta utilizada en este proyecto se describe en la sección siguiente.

PyQt

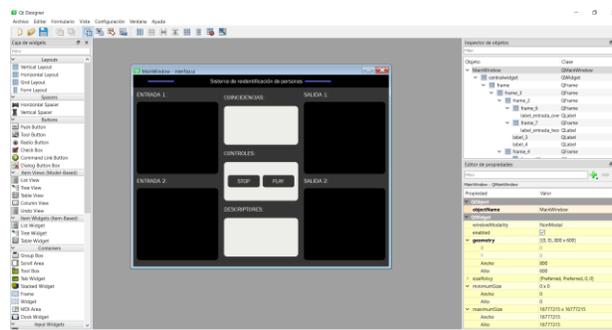
El framework PyQt ofrece una selección de herramientas para desarrollar interfaces de usuario capaces de conectarse a bases de datos SQL, manejar multimedia, realizar gráficos 2D y 3D y llevar a cabo otras muchas funciones. PyQt proporciona todas las herramientas necesarias para crear aplicaciones completas y dinámicas con interfaces de usuario de última generación (J. Willman, 2021).

Hay situaciones en las que se requiere una solución más rápida y eficaz, aunque la construcción manual de la interfaz de usuario ofrezca a los diseñadores un mayor control sobre el proceso de diseño. En este sentido, Qt proporciona una interfaz excelente para crear ventanas primarias, widgets y cuadros de diálogo. Especialmente útil es Qt Designer Figura 8, un conjunto de herramientas de desarrollo gráfico

que proporciona una gran selección de widgets y herramientas para el desarrollo de interfaces gráficas de usuario. Construir y configurar aplicaciones Qt o PyQt es rápido y fácil gracias a la capacidad de arrastrar y soltar de Qt Designer (J. M. Willman, 2022).

Figura 8

Qt Designer.



Debido a la disponibilidad de Qt Designer, se eligió PyQt como herramienta para crear interfaces gráficas de usuario para la visualización de sistemas de reidentificación. La inclusión de Qt Designer, una potente herramienta de desarrollo gráfico que facilita el diseño y la personalización de interfaces de usuario, es una de las características distintivas de PyQt5.

Herramienta de codificación para el desarrollo de una interfaz gráfica de usuario para la visualización de sistemas de re-identificación de personas

Las herramientas de codificación son necesarias para la producción de programas e interfaces gráficas de usuario. Estas herramientas ayudan a los programadores a crear, modificar, probar y depurar el código de creación de interfaces.

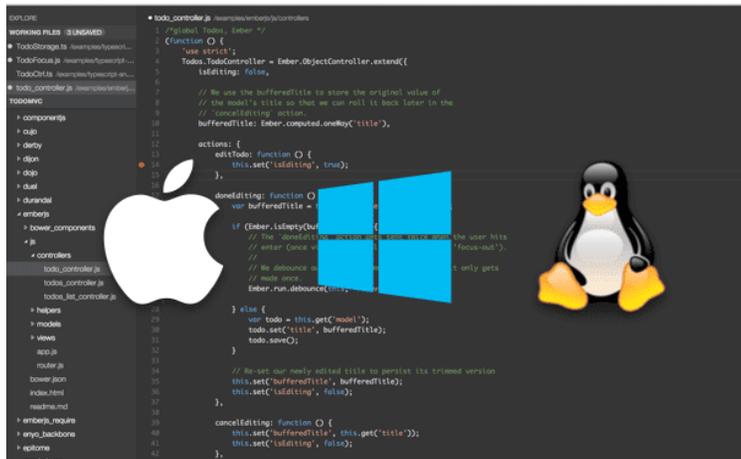
Visual Studio Code

La función Intellisense, que agiliza el código ofreciendo sugerencias gramaticales específicas para cada lenguaje de programación, es uno de los aspectos más útiles del editor Visual Studio Code (Microsoft, 2023).

El editor de código para crear la interfaz gráfica de usuario en la visualización de los sistemas de reidentificación de personas fue elegido como Visual Studio Figura 9. Su selección se basa en factores como la facilidad de uso, la experiencia de usuario (UX) ideal y una amplia gama de herramientas de codificación.

Figura 9

Visual Studio Code.



Nota. Recuperado de (Microsoft, 2023).

Características Biométricas

Los rasgos físicos conocidos como biométricos son distintivos de cada individuo. Utilizar rasgos biométricos prácticamente imposibles de imitar tiene una serie de ventajas y un alto nivel de seguridad (Alejandra, 2016)

La tasa de identificación de un individuo desconocido es de aproximadamente el 92%, según un informe del Instituto Nacional de Normas y Tecnología (NIST) sobre diversas evaluaciones biométricas de la metodología de reconocimiento facial de 2010, que se considera el método más preciso para el reconocimiento facial (Kaur et al., 2020).

Características Soft-biométricas

Los rasgos biométricos blandos son atributos físicos o de comportamiento tangibles y observables. Algunos ejemplos comunes de rasgos biométricos blandos son la altura, el peso, las huellas dactilares, el color del pelo, el color de los ojos, la etnia, el color de la piel, la ropa, la forma de andar, las cicatrices y los tatuajes, pero carecen del carácter distintivo y la permanencia necesarios para distinguir suficientemente a dos personas (Reid et al., 2013).

Estos sencillos aspectos biométricos blandos son especialmente adecuados para las difíciles situaciones que se plantean en la reidentificación de personas, en las que hay que tener en cuenta cuestiones como la adquisición de rasgos a distancia, la discreción en cuanto al conocimiento de los observados, la baja resolución y calidad de las imágenes, los cambios de iluminación, las características visuales de las distintas cámaras, el uso del zoom y las variaciones de perspectiva, entre otras dificultades. Utilización del aspecto de la prenda como sustituto de los aspectos biométricos blandos tradicionales (Alejandra, 2016) (Satta, 2013).

Técnicas Holísticas

Las zonas faciales o corporales enteras se utilizan como entrada en un sistema de re-identificación que utiliza metodologías holísticas, también conocidas como modelos holísticos. Imagen 1. Los enfoques holísticos indican que los descriptores utilizados para identificar las cualidades de cada imagen son globales, es decir, que se tiene en cuenta todo el rostro o el cuerpo (Llamas, 2017).

Figura 10

Representación del modelo holístico.



Nota. Recuperado de (Ljupco, 2020).

Los métodos holísticos como Eigenfaces y Fisherfaces se emplean con frecuencia en la reidentificación. Estos métodos no holísticos identifican utilizando la región facial.

Eigenfaces

Eigenfaces es un método integral para identificar rostros utilizando vectores propios. Este método se enmarca dentro de los enfoques de visión por ordenador que sólo utilizan la región facial para identificar a las personas. Utiliza un método basado en la apariencia denominado Eigenfaces, que trata de codificar y comparar imágenes faciales individuales de forma holística utilizando la varianza encontrada en un conjunto de fotos faciales (Lata et al., 2009).

Figura 11

Imagen de captura de rostro y su imagen con la proyección definida por Eigenfaces



Nota. Recuperado de (Turk & Pentland, 1991).

Una de las ventajas de utilizar Eigenfaces es que emplea una estrategia de reducción de la dimensionalidad que permite representar los rostros en un espacio de características más pequeño. Esto facilita el procesamiento y aumenta la eficacia del algoritmo.

Fisherfaces

Mediante el análisis discriminante lineal (LDA), la técnica Fisherfaces integral encuentra y extrae los elementos más cruciales de una imagen que ayudan a distinguir los rostros. La técnica Fisherfaces holística trata de encontrar los vectores que mejor distinguen los datos mostrados en la Figura 3 para reducir la dimensionalidad de las matrices de representación de las imágenes (Encalada & Carlos, 2015).

Figura 12

Fisherfaces.



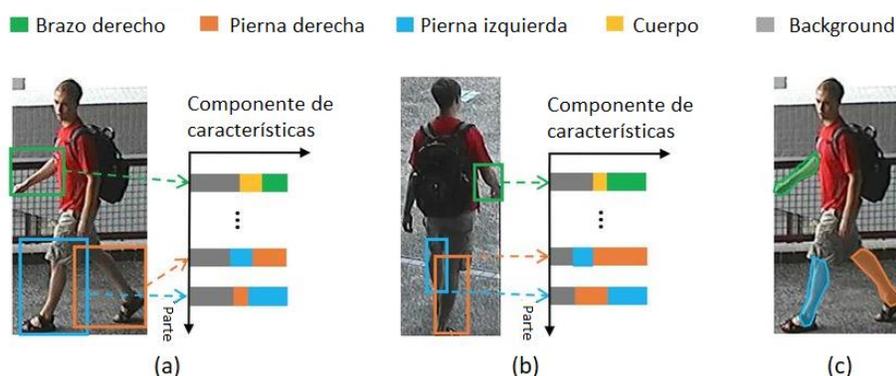
Nota. Recuperado de (He et al., 2005).

Técnicas no Holísticas

En los enfoques no holísticos se utilizan modelos no holísticos, que definen a la persona como un lugar o una forma. Sin embargo, los detectores basados en modelos menos holísticos, o a trozos Figura 4, que aceptan oclusiones parciales y variaciones de posición, caracterizan a la persona como una mezcla de muchas regiones o formas. Estas técnicas se clasifican como no holísticas (García-Martín & Martínez, 2015).

Figura 13

Representación del modelo no holístico.



Nota. Recuperado de (Huang et al., 2019).

Los enfoques no holísticos tienen la ventaja de poder captar información fina y precisa de la persona, como rasgos distintivos de la cara, patrones de ropa o accesorios, lo que aumenta la precisión de la identificación (Encalada & Carlos, 2015).

Técnicas de Visión por Computador

El campo de la visión por ordenador es amplio e incluye una gran variedad de métodos utilizados para alcanzar diversos objetivos y previsiones. Los distintos algoritmos de aprendizaje automático

utilizados en la actualidad hacen referencia a las diversas técnicas de visión por ordenador. La siguiente lista de técnicas de visión por ordenador incluye algunas de las más significativas (Paneru & Jeelani, 2021).

Clasificación de imágenes

La categorización de imágenes es el proceso de aplicar determinados criterios a agrupaciones de píxeles y vectores de una imagen Figura 5. Sin embargo, la aplicación de la categorización de imágenes se enfrenta a una serie de retos (Nalbant & Uyanik, 2021).

Figura 14

Clasificador de imágenes utilizado para clasificar equipos de construcción.



Nota. Recuperado de (Paneru & Jeelani, 2021).

Para distinguir entre las agrupaciones de píxeles en las que aparece una persona en una foto digital y aquellas en las que no, se emplea la clasificación de imágenes en el contexto de la re-identificación. El algoritmo en cascada de Haar, que crea una regla de categorización de imágenes, utiliza esta frase.

Detección de objetos

Una cámara, sistema de visión por ordenador u otro dispositivo pueden emplear el enfoque de detección de objetos en visión por ordenador para reconocer elementos en imágenes digitales (Zou et al., 2023).

La detección de objetos se esfuerza por ofrecer modelos y métodos para aplicaciones de visión por ordenador. La precisión (incluida la precisión de clasificación y la precisión de localización) y la velocidad son los dos parámetros más cruciales para la detección de objetos (Zou et al., 2023).

Figura 15

Detector de objetos usado para detectar una cámara.



Nota. Recuperado de (Cook, 2023).

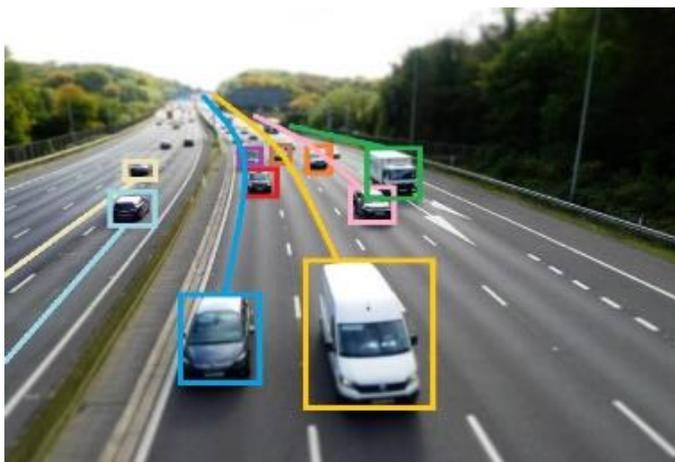
La detección de objetos es una habilidad crucial en el contexto de la re-identificación de personas para reconocer y localizar a seres humanos en diversos escenarios y entornos (Nalbant & Uyanik, 2021).

Seguimiento de objetos

El seguimiento de un objeto en movimiento en una o varias escenas se denomina seguimiento de objetos. Para el seguimiento de objetos, se emplean aplicaciones de aprendizaje profundo después de la detección del objeto, cuando se rastrea el movimiento del objeto. Los objetos que se rastrean tienen un símbolo junto a ellos (Soleimanitaleb et al., 2019).

Figura 16

Seguidor de objetos utilizado para el seguimiento de automóviles.



Nota. Recuperado de (Fleming, 2019).

El seguimiento de objetos es una técnica crucial en la reidentificación de individuos, ya que permite seguir a una persona a través de varias cámaras o escenas a lo largo del tiempo, lo que permite una identificación continua en diversos entornos.

Modelos y Algoritmos de Machine Learning

Un problema básico en el ámbito de las tecnologías de visión por ordenador es la re-identificación de individuos (Re-ID). Para re-identificar con éxito a una persona se utilizan diferentes métodos de aprendizaje automático con las funciones de clasificador, descriptor y detector que se describen a continuación.

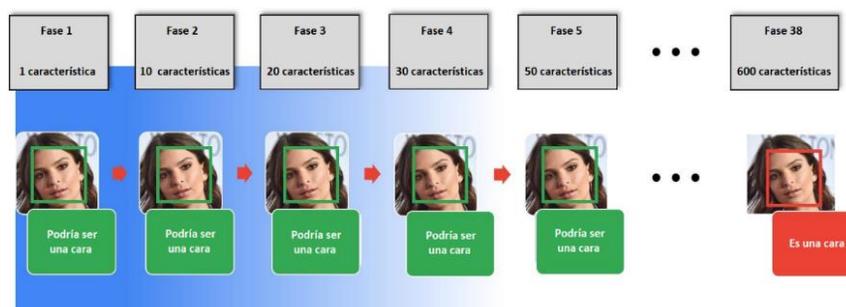
Algoritmo Haar Cascade

Para la detección de objetos se creó un algoritmo de aprendizaje automático llamado Haar Cascade que, una vez ajustado, también puede reconocer personas (Mohammed et al., 2021).

Para extraer información de las imágenes, el método Haar Cascade utiliza filtros que añaden efectos como localización de bordes, desenfoco y enfoque. Para integrar las fotos a través de las distintas etapas del algoritmo de clasificación, una porción a la vez, y reconocer el elemento en el contexto de la reidentificación, a una persona, se aplican a las imágenes filtros Haar Cascade, también conocidos como características Haar, Figura 8 (Mohammed et al., 2021), (Le et al., 2019).

Figura 17

Reconocimiento facial mediante el método Haar Cascade.



Nota. Recuperado de (Jaiswal, 2022).

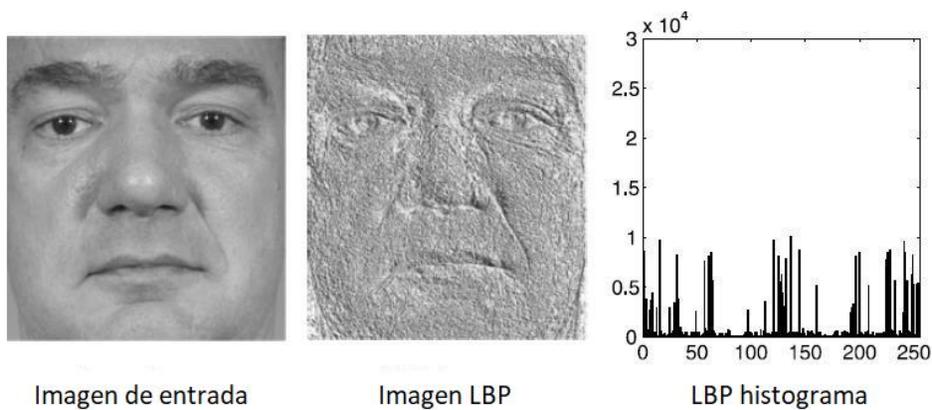
La capacidad del clasificador en cascada de Haar para identificar y localizar determinados objetos, como rostros humanos en imágenes y vídeos, es una de las ventajas de utilizarlo para la reidentificación de individuos.

Algoritmo Local Binary Pattern Histogram (LBPH)

Un enfoque de aprendizaje automático denominado Patrón Binario Local (LBP) etiqueta los píxeles binarios de una imagen en función del umbral y la posición de cada píxel. La característica de este algoritmo permite el análisis de imágenes en tiempo real. El LBPH, que se utiliza para representar imágenes faciales mediante un vector de datos básico, se elabora fusionando el LBP con histogramas (Verdeguer Valderrama & Campos Vasquez, 2021)

Figura 18

Transformación de imagen a LBP y su respectivo histograma.



Nota. Recuperado de(Pessanha Santos, 2014).

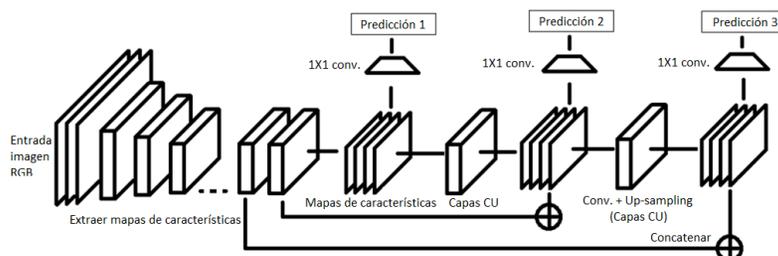
La capacidad del método del histograma de patrones binarios locales (LBPH) para describir de forma eficaz y eficiente los aspectos locales de los rostros es una de las ventajas de utilizarlo como descriptor en la re-identificación de individuos.

YOLO

El modelo "You Only Look Once" (YOLO) es un sistema de detección de objetos. Una red neuronal convolucional especial utilizada por YOLO predice la ubicación de los objetos basándose en la probabilidad de que aparezcan en la imagen original. La estructura de este sistema lo hace extremadamente útil para aplicaciones en tiempo real (Mihaescu et al., 2020).

Figura 19

Representación de la arquitectura de YOLO.



Nota. Recuperado de (Punn et al., 2021).

La alta velocidad y eficacia del algoritmo YOLO en la detección de objetos, incluidos rostros y cuerpos humanos, es una de las ventajas de utilizarlo como detector en la re-identificación de personas.

Fusión de características

La re-identificación de personas implica encontrar y luego reconocer a una persona desde varios ángulos de cámara (Fayyaz et al., 2020). Hay tres pasos únicos en el proceso de re-identificación facial o corporal de este trabajo que requieren vídeo o fotos de alta calidad tomados a la distancia más cercana.

Para volver a identificar la cara de un sujeto, primero se toma la figura 20 y luego se compara con la textura del sujeto, que se extrajo utilizando un descriptor de características biométricas blandas. La textura facial es una característica biométrica blanda que se extrae mediante el método LBPH y luego se transforma en histogramas.

Figura 20

Identificación facial y captura de imagen.



Para la reidentificación del cuerpo, a diferencia de la reidentificación facial, se emplean máscaras y se comparan con LBPH Figura 21, y la característica biométrica blanda -la textura del cuerpo- se recupera utilizando el mismo método descrito anteriormente.

Figura 21

Mascaras corporales para la comparación con LBPH.



Nota. Recuperado de (Lignia Tuapanta & Pichucho Tandalla, 2023).

En segundo lugar, tras utilizar el enfoque LBPH para obtener un gran número de histogramas, pasamos al entrenamiento en cascada mediante el algoritmo de aprendizaje automático Haar Cascade y pasamos por varias etapas. En este nivel, se crea una regla de clasificación en la que se combinan todas las características, se ejecuta una comparación óptima para mejorar aún más la precisión y se lleva a

cabo una operación de agrupación para dividir todos los histogramas. Aunque la cara esté oculta o el sujeto esté de espaldas, se puede identificar el torso (Fayyaz et al., 2020).

Si se dispone de un histograma completo de la cara o el cuerpo de la persona, será preferible utilizar estos datos para reducir el número de búsquedas en el conjunto de datos durante la fase de agrupación. De este modo, no es imprescindible recorrer toda la base de datos, ya que muchos individuos quedarán descalificados enseguida. Por lo tanto, para que esto sea realmente discriminatorio y se pueda reconocer a los candidatos, se debe alcanzar una precisión adecuada (Ammar et al., 2017).

A YOLO neural network that has been trained on data from a previous level predicts the identification at the end. This network can identify a person in real time from a distance of several meters (Figure 21) because of its multi-scale prediction function.

Figura 22

Reconocimiento de persona.



Metodología para el desarrollo de la interfaz gráfica para la visualización de sistemas de re-identificación de personas

Con el fin de identificar y especificar claramente los requisitos de la interfaz, una metodología ofrece un marco estructurado para la creación de una interfaz gráfica de usuario para la representación de un sistema de re-identificación de personas. Los métodos más eficaces para la re-identificación, como los algoritmos de detección de objetos, el seguimiento humano y la extracción de rasgos distintivos, pueden evaluarse y elegirse utilizando una metodología.

Se eligió Scrum como enfoque de diseño para la aplicación, lo que fue un movimiento inteligente y estratégico que tiene muchas ventajas para el desarrollo rápido y exitoso del proyecto.

Scrum

Scrum es una metodología de desarrollo ágil basada en sprints, que son ciclos breves para el proceso de desarrollo (Trigás Gallego, 2012).

Los cinco pasos del proceso de desarrollo ágil, tal y como los define Scrum, deben entenderse para comprender el ciclo de desarrollo de Scrum:

1. **Inicio:** Para garantizar el éxito, se toman decisiones importantes desde el principio y se definen los objetivos y el alcance del proyecto. Inicialmente, se completan tareas como la formación del equipo scrum, la selección del scrum master, el desarrollo épico, la creación de backlog priorizado y el desarrollo de objetivos del proyecto (Makadia, 2023).
2. **Planificación y estimación:** La generación, aprobación y evaluación de historias de usuario, así como el compromiso de las historias de usuario, la creación de tareas, la evaluación y la creación de un Sprint backlog son actividades incluidas en la fase de planificación (Makadia, 2023).

3. **Implementación:** Durante la fase de implementación, se llevan a cabo todos los procedimientos previstos para la construcción del producto de software. Esto implica la creación de varios componentes de software, así como la celebración de reuniones diarias (Makadia, 2023).
4. **Revisión y retrospectiva:** Esta etapa del ciclo de vida del proyecto se centra en evaluar lo que se ha conseguido hasta ahora, asegurarse de que el equipo se ha ceñido al plan predeterminado y hablar de futuras oportunidades de mejora (Makadia, 2023).
5. **Lanzamiento:** Los principales objetivos de esta fase son la obtención de resultados satisfactorios para el cliente y la identificación, documentación e incorporación de las lecciones aprendidas en relación con el proyecto (Makadia, 2023).

Uno de los principios fundamentales de esta metodología, Scrum, es que las reuniones diarias rigen estas iteraciones (Makadia, 2023).

Capítulo III

Implementación del sistema

El presente capítulo se centra en detallar exhaustivamente el procedimiento utilizado para el desarrollo de una interfaz de visualización de señales de video propuesto en este trabajo de investigación. Las señales de video serán generadas por un modelo de re-identificación de personas, utilizando características fáciles y soft-biométricas como la textura, a través de técnicas basadas de visión por computadora.

El proyecto se desarrolla en las instalaciones de la Universidad de las Fuerzas Armadas ESPE sede Latacunga, campus Guillermo Rodríguez Lara, en donde se probará la interfaz y la cual nos proporciona las secuencias de video, debido a que cuenta con un circuito cerrado de dos cámaras de video no superpuestas.

Para comprender como la interfaz muestra al operador las señales de video generadas por un modelo de re-identificación de personas, es necesario una explicación breve: i) La interfaz tiene tres secciones verticales, la primera sección contiene componentes para visualizar la entrada 1 y 2.

La primera entrada muestra la re-identificación de las personas a través de la característica facial. En el caso que no se detecte a la persona, en la segunda entrada se observa la re-identificación con características soft-biométricas como la textura, ii) el sistema de Re-ID al realizar la identificación por medio de la cámara 1, obtiene una secuencia de imágenes indexadas, que servirán para representar la segunda sección de la interfaz, en donde se encuentran las coincidencias y los descriptores con sus respectivos valores de precisión, iii) y finalmente en la última sección se muestra las imágenes del rostro y silueta corporal de la persona identificada por el sistema.

La metodología de desarrollo aplicada para la creación de la interfaz de visualización de señales de video, es una ágil, debido a que este tipo de metodologías se han utilizado especialmente en el desarrollo de proyectos software con equipos pequeños, calendarios de desarrollo cortos, cambios constantes de requisitos o muy emergentes (López-Martínez et al., 2016; Sachdeva, 2016).

Para este trabajo de investigación se pretende usar el marco de desarrollo ágil scrum, debido a su sencillez y versatilidad; además, tiene un enfoque incremental e iterativo para gestionar el desarrollo de productos software complejos, lo que permite construir un trabajo en iteraciones cortas donde cada iteración consta de un cuadro específicos de tiempo.

Comúnmente el ciclo de vida de desarrollo de un software empieza por el análisis de requerimientos, los mismo pueden ser obtenidos a través de las historias de usuario (HU), técnica propuesta por el marco de desarrollo ágil para agrupar las características que se requieren para cumplir con el objetivo de este proyecto.

Por lo general en las historias de usuario se destacan los roles, el requerimiento funcional y el resultado esperado bajo algunos criterios de aceptación. Esto ayuda a tener una visión completa de lo que se pretende desarrollar en la interfaz, por ejemplo, la estimación del tiempo necesario para cumplir una tarea, herramientas software y recursos hardware necesarios para el desarrollo y ejecución, y las pruebas del producto en cada incremento. Para ello es necesario conocer algunos principios que plantea el framework scrum para el desarrollo de productos software (Carvalho et al., 2011; Sachdeva, 2016).

- **Sprint:** Es un periodo de tiempo comprendido entre 1 a 4 semanas donde el equipo de desarrollo se compromete a completar un conjunto de historias de usuario seleccionadas del product backlog.

- **Product Backlog:** Es una lista priorizada y dinámica de características, requisitos, mejoras y arreglos de errores determinadas por el product owner en formato de historias de usuario.
- **Daily Scrum:** Tiene una duración entre una y dos horas, reúne a todo el equipo para establecer las tareas del día y debatir los resultados de las actividades anteriores.
- **Sprint Planning Meeting:** Es un encuentro crucial, donde el equipo decide que tareas se completarán durante el sprint y cómo se llevarán a cabo.
- **Sprint Backlog:** Esta lista de tareas que deben completarse durante cada sprint es un subconjunto del backlog del producto.
- **Sprint Review Meeting:** En esta reunión, que se celebra tras la conclusión de cada sprint, el equipo de desarrollo revisa lo que ha ido bien y mal, así como las lecciones aprendidas.

Análisis y diseño del sistema

Análisis del sistema

Los roles Product Owner, Scrum Master y Scrum Team o Team Development deben establecerse dentro del equipo de trabajo para poder implementar la metodología propuesta (López-Martínez et al., 2016). Estos roles son los encargados de trabajar conjuntamente para completar el sistema a través de iteraciones.

Roles Scrum

La Tabla 1 enumera los roles que se han asignado a este proyecto, junto con el nombre de cada miembro del proyecto y una descripción de sus funciones asociadas. En este escenario, hay dos personas en estos roles, además del profesor de la tesis, que se considera el propietario del producto, debido al reducido tamaño del equipo de trabajo.

Tabla 3*Roles Scrum.*

	Rol	Integrante	Descripción
1	Product Owner	Dr. José Luis Carrillo Medina	Responsable de la visión del producto, la recopilación y priorización de requisitos.
2	Scrum Master	Nancy del Pilar Jacho Guanoluisa	Es el responsable del proceso de scrum, cumplir los tiempos de entrega, guía al equipo.
3	Team Scrum	Erick Alejandro Fernández Bustos Jhonatan Javier Chiluisa Garcia	Equipo auto-organizado y multifuncional responsables del desarrollo e implementación del producto.

Nota. Designación de los roles principales comprendidos en la metodología Scrum.

Historias de Usuario

Las historias de usuario se utilizan para capturar de forma clara y sucinta los requisitos y necesidades del usuario final de acuerdo con la metodología scrum para la obtención de requisitos. Las historias de usuario facilitan la comunicación entre los distintos miembros del equipo scrum, el propietario del producto y las partes interesadas.

El equipo de desarrollo puede utilizar la técnica de estimación relativa para calibrar la dificultad y el tiempo necesarios para cumplir cada historia de usuario desglosando los requisitos en funcionalidades manejables que puedan crearse y entregarse gradualmente en cada sprint.

Las historias de usuario (HU) necesarias para desarrollar la interfaz de visualización se describen en la Tabla 2, donde figuran el código (un identificador único), el actor o usuario final del sistema, una breve descripción de la característica o funcionalidad y el resultado previsto.

Tabla 4

Historias de Usuario.

Código	Actor	Característica	Resultado
Hu_01	Como usuario	Quiero contar con una interfaz intuitiva, fácil de usar, con diseño responsivo y multiplataforma.	Visualizar y comprender rápidamente el proceso de re-identificación de personas.
Hu_02	Como usuario	Quiero visualizar las señales de video en tiempo real provenientes de las dos cámaras colocadas en dos ángulos y puntos estratégicos.	Observar las imágenes o frames de manera simultánea o en pantalla dividida.
Hu_03	Como usuario	Quiero visualizar pequeñas galerías de imágenes que se obtienen del sistema de re-identificación de personas.	Observar las imágenes que representan el conjunto de coincidencias y descriptores.

Código	Actor	Característica	Resultado
Hu_04	Como usuario	Quiero visualizar la re-identificación de la persona en tiempo real a través de las características faciales y soft-biométricas como la textura.	Observar el proceso de re-identificación de personas que realiza el sistema tanto en la cámara 1 y 2.
Hu_05	Como usuario	Quiero visualizar información adicional sobre las personas identificadas.	Acceder a datos como el nombre y porcentaje de presión del modelo de re-identificación.
Hu_06	Como usuario	Quiero visualizar notificaciones o alertas en tiempo real de la re-identificación de las personas.	Detectar eventos específicos en las señales de video.

Nota. Información sobre las características funcionales de la interfaz.

Product Backlog del proyecto

El backlog del producto (PBL), es un documento que incluye los requisitos prioritarios del producto de software (puede incluir requisitos funcionales y no funcionales), así como los requisitos técnicos. Este documento es necesario realizarlo, después de obtener las historias de usuario de las partes interesadas del proyecto (Kayes et al., 2016). Una técnica recomendada para el desarrollo ágil, un

PBL visible y accesible ayuda a reducir la documentación manual y sobrecarga administrativa (Shore, s. f.).

En la tabla 3: Product Backlog del proyecto se muestra las historias de usuario priorizadas que conforman el producto backlog, se detalla su número, el código o identificador, la estimación en días, fecha de inicio, fecha final y el sprint al cual pertenece.

Tabla 5

Product Backlog del Proyecto.

N° de HU	Código	Estimación (días)	Fecha de inicio	Fecha de finalización	N° de Sprint
1	Hu_01	7	24/04/2023	02/05/2023	01
2	Hu_02	15	03/05/2023	23/05/2023	02
3	Hu_03	10	24/05/2023	06/06/2023	02
4	Hu_04	17	7/06/2023	29/06/2023	03
5	Hu_05	12	30/06/2023	17/07/2023	03
6	Hu_06	4	18/07/2023	21/07/2023	03

Nota. Registro de las tareas priorizadas que se deben realizar con sus respectivas estimaciones y tiempos de entrega.

Herramientas software para el desarrollo de la interfaz

En la tabla 4 se muestran las herramientas de desarrollo con sus versiones que serán usadas para el diseño y construcción de la interfaz de visualización.

Tabla 6*Herramientas de desarrollo.*

Herramienta	Detalle
Entorno de desarrollo (IDE) para edición y construcción de código fuente.	El lenguaje de programación de alto nivel usado es Python en su versión 3.8.10, el IDE por defecto es Visual Studio Code con su versión 1.79.2
Diseño de mockups de la interfaz de visualización.	Para el diseño de la interfaz se utiliza un software gratuito que se encuentra en la web que ayuda en cuanto a UI/UX y se denomina Figma.
Construcción de interfaces de usuario (GUI) para aplicaciones de escritorio compatible con Python.	La construcción y diseño responsivo de la interfaz se realiza con la librería PyQt5 en su versión 5.15.9; un framework multiplataforma ampliamente utilizado para el desarrollo de aplicaciones de escritorio e interfaces graficas de usuario (GUI).

Herramienta	Detalle
Diseño de interfaces de usuario.	PyQt5 proporciona como parte del framework una herramienta gráfica y visual denominada Designer que permite diseñar y construir interfaces graficas de usuario de manera fácil y rápida, sin necesidad de escribir código manualmente.
Visualización simultanea de segmentos de video que proporciona un circuito cerrado de cámaras.	Para visualizar varias entradas de video que nos proporcionan las cámaras simultáneamente y en tiempo real se utiliza QThread, una clase que proporciona una forma conveniente de administrar hilos de procesamiento en una aplicación GUI.

Nota. Detalle de las tecnologías y herramientas software para el desarrollo de la interfaz de visualización.

Ambiente controlado para la implementación de la Interfaz de visualización para un sistema de Re-identificación de personas

La implementación de la interfaz de visualización se realiza en un ambiente controlado ubicado en las instalaciones de la universidad de las fuerzas armadas ESPE sede Latacunga, localización que cuenta con varias edificaciones divididas en bloques, en donde se ha seleccionado el Bloque A, pasillo del segundo piso.

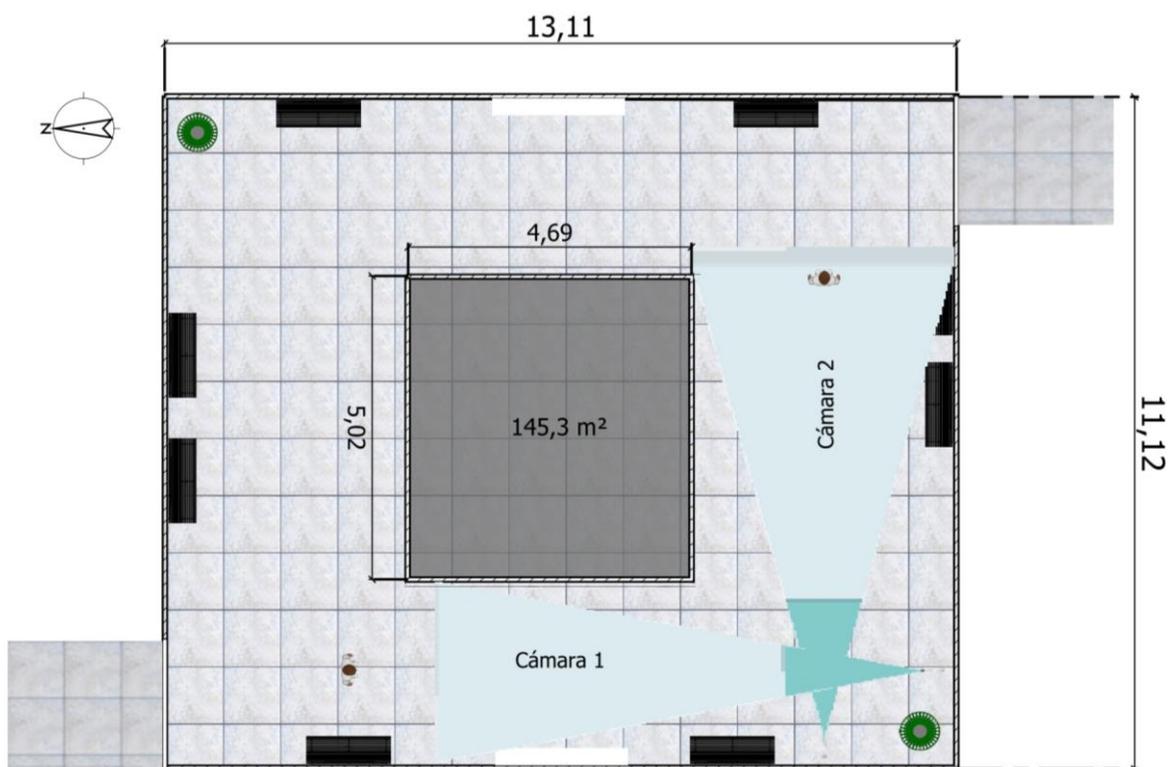
El segundo piso consta de un pasillo que pasa por el departamento de Ciencias de la computación. Las cámaras esta colocadas de forman que no se superponen, la primera cámara tiene la vista en dirección al Norte, la segunda cámara está colocada con su vista en dirección al Oeste.

La disposición de las cámaras ayuda a cubrir un área específica del pasillo; además, genera las señales de video necesarias para representar la re-identificación de la persona usando características faciales y soft-biométricas como la textura en la interfaz de visualización.

El entorno controlado se virtualizo como se muestra en la Fig. 1 y Fig. 2. La primera figura muestra el pasillo en dos dimensiones y la segunda figura muestra el pasillo en tres dimensiones. En las dos figuras se puede apreciar la disposición de las cámaras utilizadas en este proyecto, la primera genera señales de video que corresponde a la identificación facial de la persona, mientras que la cámara 2 genera las señales de video que representa la detección y re-identificación de la parte posterior de persona usando características soft-biométricas como la textura de la vestimenta.

Figura 23

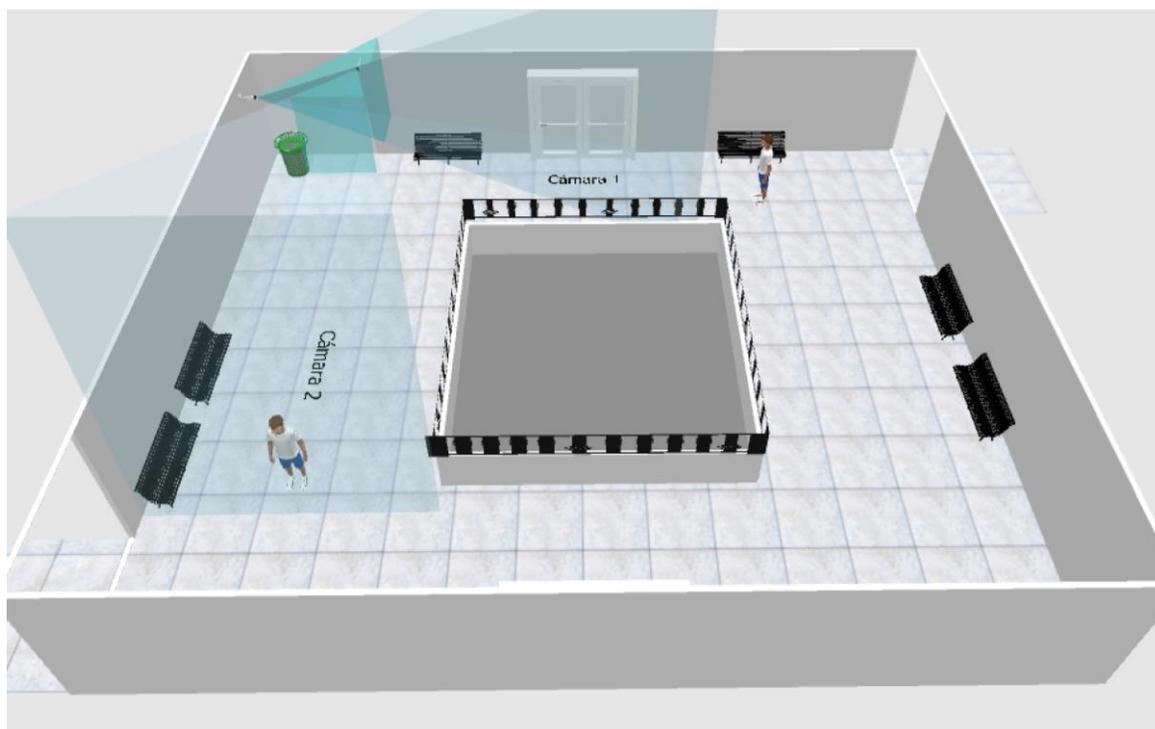
Plano 2D del Bloque A, segundo piso de la Universidad de las Fuerzas Armadas ESPE sede Latacunga.



Nota. Plano 2D del pasillo seleccionado para la captura de videos de prueba.

Figura 24

Representación 3D de la ubicación de las cámaras para capturar videos de prueba.



Definición e implementación de técnicas de visualización y optimización de recursos para captar modelos de re-identificación de personas

El sprint backlog es un artefacto clave de la metodología scrum que debe realizarse después de completar el product backlog. Este artefacto trata de la lista de tareas y elementos seleccionados del product backlog que el equipo de desarrollo se compromete a completar durante un Sprint específico. El sprint backlog es una previsión que pertenece al equipo de desarrollo, y solo el equipo de desarrollo puede modificarlo (*Scrum Guide | Scrum Guides*, s. f.).

Es importante destacar que se planificaron reuniones virtuales mediante las videoconferencias que proporciona Google Meet; además, se organizaron reuniones presenciales para la retroalimentación del proyecto, cumpliendo con las reuniones requeridas que contempla la metodología scrum.

A continuación, la tabla 5 muestra el hardware necesario para la implementación de la interfaz de visualización para un sistema de re-identificación de personas que usa características faciales y soft-biométricas como la textura de la ropa.

Tabla 7

Hardware necesario para implementar la interfaz.

Componente	Detalle
Procesador	Intel Core i7-13700k, 3.4 GHz de 16 Core
Memoria RAM	DDR4 de 32 GB
Sistema operativo	Windows 11 Pro
Tarjeta de video	NVIDIA GEFORCE RTX 3060 12GB

Nota. Detalle de los componentes hardware requeridos para implementar la interfaz de visualización.

Sprint 01: Diseño de una interfaz multiplataforma, fácil de usar e intuitiva

Se empieza con el Sprint 01 que comprende la historia de usuario Hu_01 que se encuentra detallada en la Tabla 2, cuyo propósito es la creación y diseño de una interfaz de visualización para un modelo de re-identificación de personas, con un diseño responsivo, intuitiva y de fácil de uso. A continuación, se detalla el proceso utilizado en la historia de usuario dentro del Sprint 01.

Historia de usuario 01: Diseño de una interfaz de visualización

La historia de usuario Hu_01 indica que se quiere diseñar una interfaz de visualización de señales de video generadas por un modelo de re-identificación de personas.

Historias de usuario detalladas

La tabla 6 muestra de forma detallada, la historia de usuario Hu_01 de la interfaz de visualización para un sistema de re-identificación de personas que usa características faciales y soft-biométricas como la textura, en donde se indica los desarrolladores a cargo de la interfaz, tiempos estimados y criterios de aceptación o validaciones.

Tabla 8

Historia de usuario para la creación de la interfaz de visualización.

Historia de usuario	
Código: Hu_01	Usuario: Usuario final
Nombre: Diseño de una interfaz de visualización.	Número de sprint: 1
Prioridad: Alta	Riesgo en desarrollo: Bajo
Estimación en días: 7	Iteración asignada: 1
Desarrolladores responsables: Jhonatan Chiluisa, Erick Fernández	
Descripción: Como usuario quiero contar con una interfaz intuitiva, fácil de usar, con diseño responsivo y multiplataforma.	
Validación:	
<ul style="list-style-type: none"> • La interfaz se adapta de manera adecuada y estética a diferentes tamaños de pantalla (Computadoras de escritorio). • La interfaz sigue los estándares y pautas de diseño recomendados para cada plataforma o sistema operativo. 	

Sprint Backlog

En la tabla 7 se observa el Sprint Backlog 01, donde se encuentra detallado las tareas y elementos de trabajo que el equipo de desarrollo ha identificado como necesarios para lograr los objetivos del sprint actual. Es importante destacar que Jhonatan Chiluisa y Erick Fernández son los responsables de llevar a cabo las tareas; además, el Sprint Backlog se encuentra finalizado.

Tabla 9

Sprint Backlog 01.

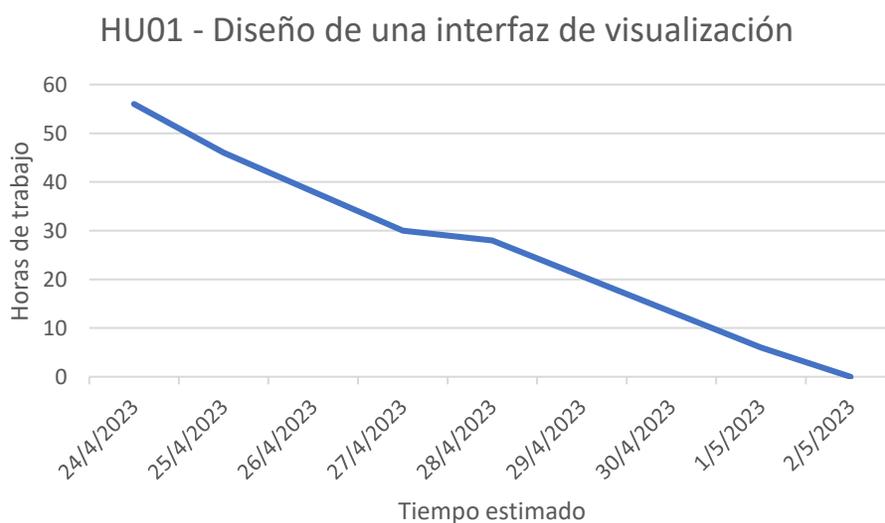
Sprint 1	Fecha Inicio 24/04/2023	Fecha Fin 02/05/2023		Jornada 8h
Hu_01	Tareas	Horas	Inicio	Fin
Tarea 01	Realizar una entrevista al usuario para obtener los requisitos de la interfaz.	2	24/04/2023	25/04/2023
Tarea 02	Revisar y analizar diseños de interfaces que se encuentran en la literatura científica.	8	24/04/2023	25/04/2023
Tarea 03	Diseñar el bosquejo de la interfaz en la herramienta Figma para obtener una maqueta visual altamente detallada y realista.	14	26/04/2023	27/04/2023
Tarea 04	Implementar el diseño de la interfaz con la librería PyQt5 para obtener una versión ejecutable en Python.	32	28/04/2023	02/05/2023

Burndown Chart

El Burndown chart o gráfico de tareas pendientes, es una herramienta visual utilizada en la metodología scrum para rastrear el progreso del equipo de desarrollo durante el Sprint. Según (Kayes et al., 2016), “es un indicador de “¿Cuánto?” trabajo se ha realizado y “¿Cuánto?” queda por hacer”.

Figura 25

Burndown Chart – Historia de Usuario 01.



Las horas estimadas, se obtienen al hacer una operación matemática al multiplicar el total de días estimados y la jornada horaria que se trabaja por día; por ende, este sprint se compone de 7 días y 8 horas diarias de trabajo tal como se observa en la Fig. 3. El resultado obtenido es un valor de 56 horas, que se toma como máximo para este periodo. Con el transcurso de los días, este valor se reducirá progresivamente hasta alcanzar cero, cumpliendo así el objetivo del sprint.

Resultados del Sprint 01

En este apartado, se detalla claramente el desarrollo del sprint y los resultados obtenidos en su ejecución. Para cumplir con el objetivo del sprint se utilizó la técnica de diseño centrado al usuario o por sus siglas (DCU), dicha técnica ayuda a satisfacer al usuario en cuanto a los requerimientos de la interfaz de visualización. Por lo tanto, al comprender los requisitos del software desde el principio, se minimizan los riesgos de crear un producto que no se ajuste a las necesidades del usuario final.

Un valor agregado del DCU, permite identificar y abordar problemas de usabilidad, lo que reduce la probabilidad de tener que realizar correcciones costosas en etapas posteriores de desarrollo.

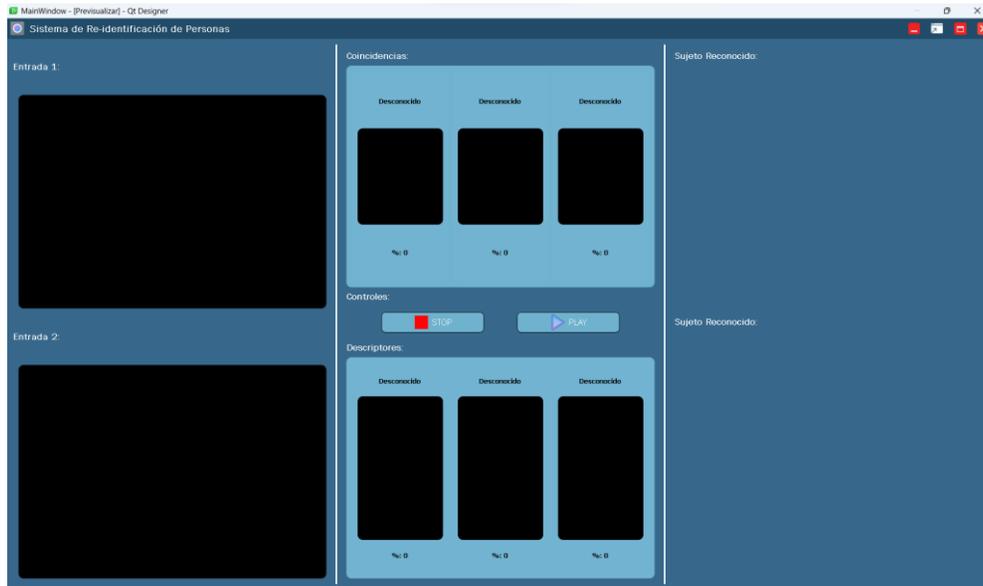
Luego de obtener el diseño de la interfaz a través de los requerimientos del usuario y usando DCU, se procede a implementar mediante la librería de Python PyQt5, cuya funcionalidad es la creación de interfaces de usuario (GUI) para aplicaciones de escritorio. El marco de desarrollo PyQt5; además, cuenta con un software que facilita el diseño de la interfaz de visualización con solo arrastrar los componentes a la ventana principal. El software llamado Designer, proporciona la interfaz en un formato de archivo .ui que puede anclarse con cualquier archivo de Python.

Para la selección de colores de la interfaz, se optó por una paleta derivada del azul. Esta elección se basa en la asociación positiva que el usuario suele hacer con este color, evocando sensaciones de tranquilidad, estabilidad, inteligencia, concentración y confianza. Asimismo, el color negro se utiliza para añadir un toque de elegancia al diseño.

En la Fig. 4 se aprecia la ejecución de la interfaz de visualización en la herramienta gráfica Designer, con cada una de sus secciones, tipografía y colores. Cabe recalcar que, al finalizar el diseño de la interfaz, el archivo se almacena en un formato XML especial (con extensión .ui) que contiene toda la información de la interfaz, incluidos los widgets, sus propiedades y sus relaciones.

Figura 26

Implementación del diseño de la interfaz de visualización.

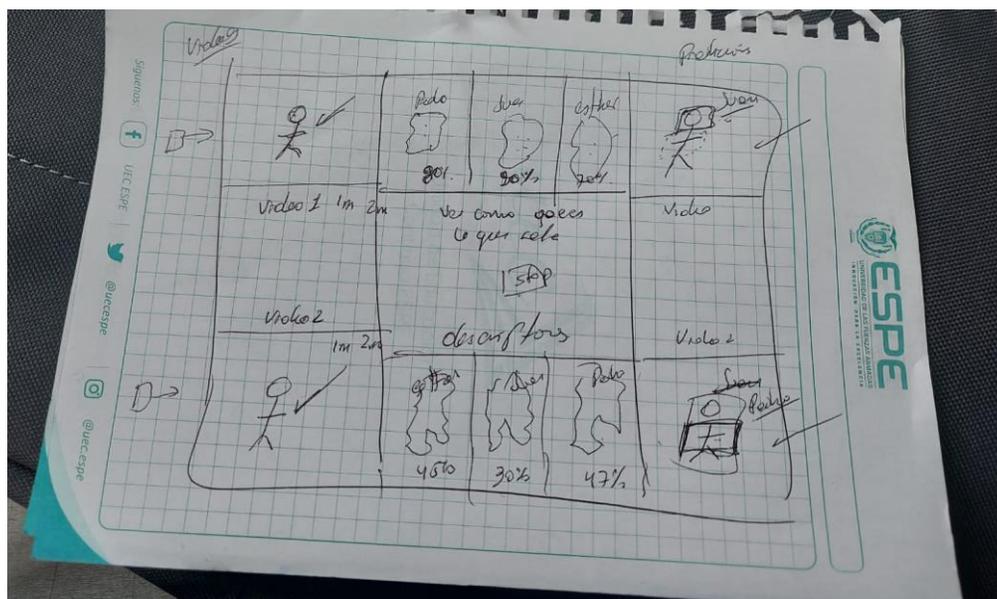


Resultados obtenidos en las tareas de la Historia de Usuario 01

Para la creación de los bosquejos de la interfaz se usó lo más simple, un lápiz y papel, esto ayudo a plasmar las ideas de diseño y requerimientos que tiene el usuario. Después de varios intentos, se obtuvo un diseño inicial como se observa en la Fig. 5.

Figura 27

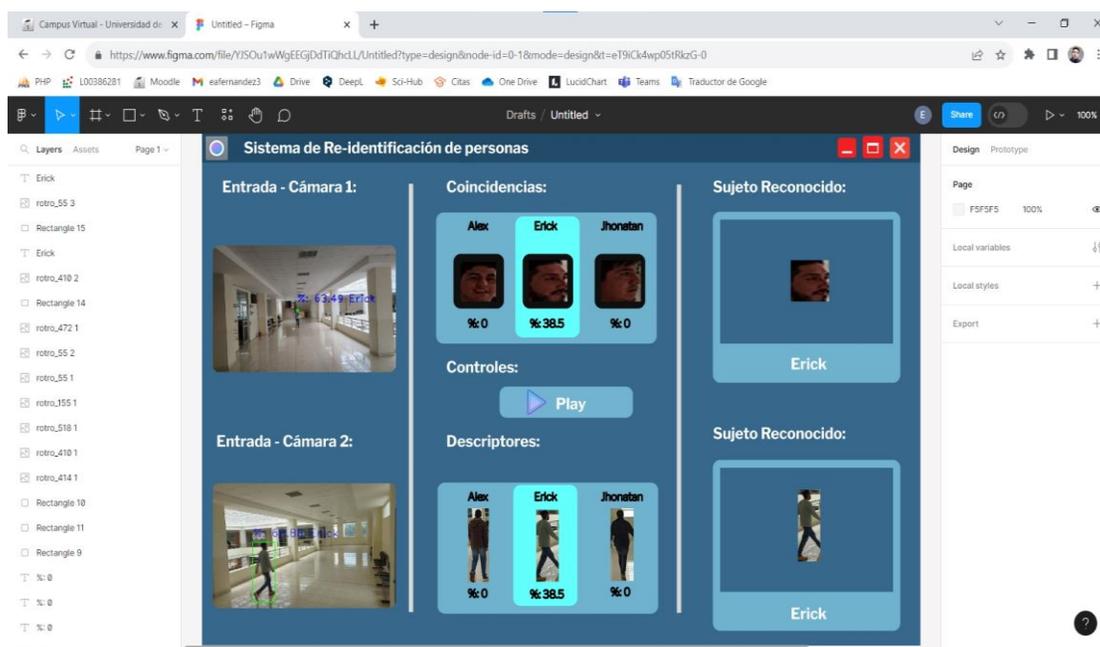
Bosquejo inicial de la interfaz de visualización.



La Fig. 6, refleja el trabajo realizado para plasmar un bosquejo de la interfaz de visualización de señales de video para un sistema de re-identificación de personas, en una maqueta visual de la herramienta Figma. La maqueta visual contiene los elementos de diseño visual (widgets, botones, iconos, etiqueta de texto), incluyendo su disposición y la jerarquía visual. Además, permite crear versiones de la interfaz al momento de realizar cambios o modificaciones de diseño, lo que facilita la iteración y el desarrollo colaborativo entre las partes interesadas del proyecto.

Figura 28

Diseño de la interfaz de visualización en Figma.



Es relevante destacar que la disposición de los elementos de la interfaz se ha diseñado teniendo en cuenta las preferencias del usuario al que se dirige y la cultura predominante occidental, en donde la lectura se realiza de arriba hacia abajo y de izquierda a derecha. Esto establece un orden preferencial de la ubicación de los elementos importantes en la interfaz de visualización.

Sprint 02: Visualización de señales de video generadas por un sistema de cámaras de video

En el desarrollo del Sprint 02, se consideraron las Historias de usuario Hu_02 y Hu_03 detalladas en la Tabla 2: Historias de Usuario, con el propósito de implementar en la interfaz una técnica de optimización de recursos computacionales para visualizar las señales de video generadas por un circuito cerrado de cámaras de video; además, visualizar pequeñas galerías de imágenes indexadas que proporciona el sistema re-identificación de personas.

A continuación, se describen en detalle los procesos utilizados en estas Historias de usuario durante el Sprint actual.

Historia de usuario 02: Visualización de señales de video generadas por un sistema de cámaras de video

En la Historia de usuario Hu_02 que se describe en la tabla 2, señala que en la interfaz se debe visualizar las señales de video que nos proporciona un sistema cerrado de dos cámaras de video ubicadas en localizaciones estratégicas para obtener datos necesarios para la re-identificación humana.

Por lo expuesto, se requiere implementar en la interfaz de visualización una técnica de optimización de recursos computacionales, con el fin de observar los flujos de video proporcionados por las cámaras de video de forma simultánea o en tiempo real.

Historias de usuario Detalladas

La Tabla 8, presenta de manera detallada la Historias de Usuario Hu_02 de la interfaz de visualización para un sistema de re-identificación de personas que utiliza características faciales y soft-biométricas como la textura. En esta tabla se proporciona los nombres de los desarrolladores responsables de desarrollo de la interfaz, puntos estimados en días y los criterios de aceptación.

Tabla 10

Historia de usuario para la visualización de señales de video generadas por cámaras de video.

Historia de usuario	
Código: Hu_02	Usuario: Usuario final
Nombre: Visualización de señales de video generadas por un sistema de cámaras de video	Número de sprint: 2
Prioridad: Alta	Riesgo en desarrollo: Bajo
Estimación en días: 15	Iteración asignada: 1
Desarrolladores responsables: Jhonatan Chiluisa, Erick Fernández	
Descripción: Como usuario quiero visualizar las señales de video en tiempo real provenientes de las dos cámaras colocadas en dos ángulos y puntos estratégicos.	
Validación:	
<ul style="list-style-type: none"> • La interfaz de visualización permite al usuario ver simultáneamente las señales de video de ambas cámaras en tiempo real. • La visualización de las señales de video debe ser clara y sin retrasos significativos para una experiencia fluida. 	

Sprint Backlog

En la Tabla 8 se presenta el Sprint Backlog 02, donde se detallan las tareas y elementos de trabajo identificados por el equipo de desarrollo como necesarios para alcanzar los objetivos del Sprint 02. Es relevante enmarcar que Jhonatan Chiluisa y Erick Fernández son los encargados de ejecutar las labores; además, se ha completado el Sprint Backlog en su totalidad.

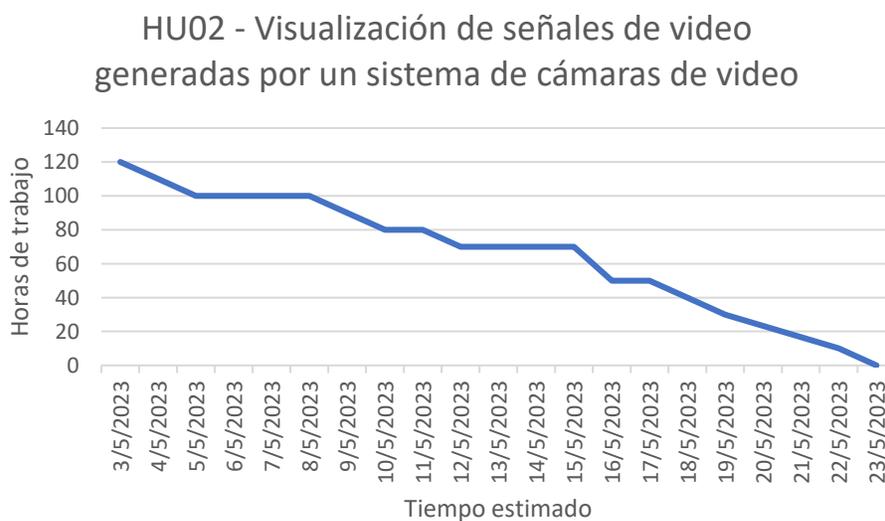
Tabla 11*Sprint Backlog 02.*

Sprint 2	Fecha Inicio 03/05/2023	Fecha Fin 23/05/2023	Jornada 8h	
Hu_02	Tareas	Horas	Inicio	Fin
Tarea 01	Captura de videos de múltiples personas caminando por el entorno controlado para realizar las debidas pruebas.	24	03/05/2023	05/05/2023
Tarea 02	Implementar en el diseño de la interfaz un widget para visualizar señales de video.	36	08/05/2023	12/05/2023
Tarea 03	Implementar QThread, una clase proporcionada por el framework PyQt5 para crear y administrar hilos de ejecución en la interfaz gráfica de usuario (GUI).	60	15/05/2023	23/05/2023

Burndown Chart

Figura 29

Burndown Chart – Historia de Usuario 02.



Para este Sprint, se estipula un periodo de 15 días y una jornada laboral de 8 horas por día, lo que resulta un valor máximo de 120 horas para el eje X, como se muestra en la Fig. 7. Con el transcurso de los días, la cantidad de horas se irá reduciendo progresivamente hasta alcanzar cero, lo que representa el cumplimiento

Historia de usuario 03: Visualización de pequeñas galerías de imágenes proporcionadas por el sistema de re-identificación de personas

La Historia de Usuario Hu_03 descrita en el contenido de la Tabla 2, indica la necesidad de visualizar pequeñas galerías de imágenes en la interfaz que representa al conjunto de coincidencias y descriptores que puede tener una persona identificada. Las imágenes se extraen del dataset que genera

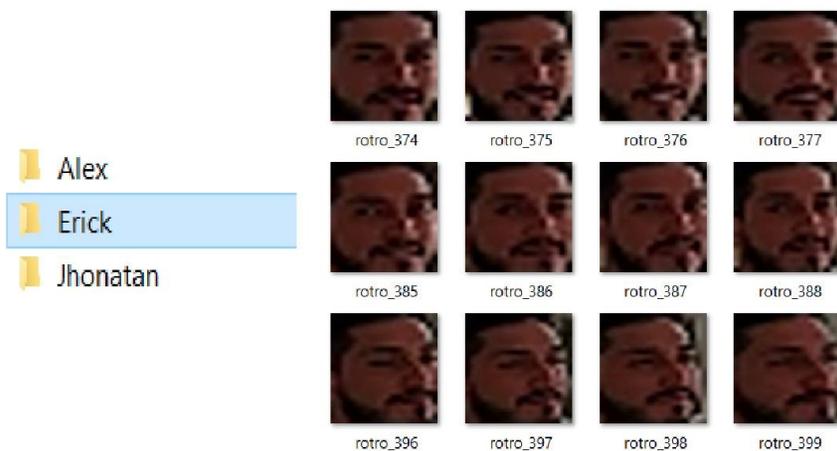
el sistema de re-identificación en la fase de identificación del individuo haciendo uso de las características faciales y soft-biométricas como la textura.

Es importante mencionar que, estas imágenes son indexadas a cada persona identificada por el modelo de re-identificación y deben actualizarse dinámicamente en la interfaz de visualización en un determinado intervalo de tiempo mientras la persona es captada por las cámaras de video.

En la Fig. 8 se aprecia el dataset con las imágenes que genera el sistema de re-identificación de personas. Estas imágenes están estandarizadas y tienen un tamaño regulado de 150x150 píxeles. Cabe mencionar que los conjuntos de imágenes se almacenan en carpetas específicas para cada clase, en este caso existen tres clases: Alex, Erick y Jhonatan.

Figura 30

Conjunto de carpetas e imágenes generadas por el sistema de re-identificación de personas.



a)

b)

Nota. El literal a) muestra las carpetas y el b) las imágenes indexadas a cada persona (clase).

Historias de usuario Detalladas

En la Tabla 9 se detalla la Historia de usuario Hu_03 para la visualización de pequeñas galerías de imágenes en la interfaz, y que serán extraídas del dataset que genera el sistema de re-identificación de personas. La tabla también incluye los nombres de los desarrolladores responsables en implementar dichos requerimientos, la estimación del tiempo en días y los criterios de validación que a historia de usuario debe cumplir.

Tabla 12

Historia de usuario para la visualización de pequeñas galerías de imágenes proporcionadas por el sistema de re-identificación de personas.

Historia de usuario	
Código: Hu_03	Usuario: Usuario final
Nombre: Visualización de pequeñas galerías de imágenes proporcionadas por el sistema de re-identificación de personas	Número de sprint: 2
Prioridad: Alta	Riesgo en desarrollo: Bajo
Estimación en días: 10	Iteración asignada: 1
Desarrolladores responsables: Jhonatan Chiluisa, Erick Fernández	
Descripción: Como usuario quiero visualizar pequeñas galerías de imágenes que se obtienen del sistema de re-identificación de personas.	

Validación:

- Cada galería de imágenes contiene imágenes relacionadas a la misma persona o individuo para facilitar su identificación.
 - Las galerías de imágenes se actualizan de forma automática para reflejar los cambios en el sistema de re-identificación de personas.
 - La velocidad de carga de las galerías de imágenes es óptima para garantizar una experiencia de usuario fluida.
-

Sprint Backlog

En la Tabla 10, se detalla el Sprint Backlog 03 donde se especifica las tareas que se desarrollan durante la ejecución del Sprint, los días en los que se planifico realizar el Sprint y el tiempo estimado en horas. Es importante destacar que Jhonatan Chiluisa y Erick Fernández son los responsables de llevar a cabo las tareas con compromiso; además, se ha logrado culminar en su totalidad el Sprint Backlog.

Tabla 13

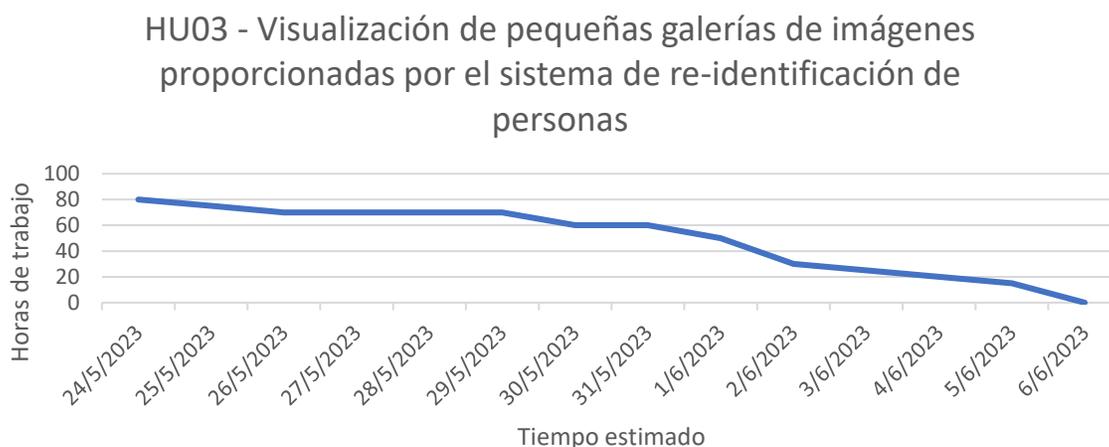
Sprint Backlog 03

Sprint 2	Fecha Inicio 24/05/2023	Fecha Fin 06/06/2023	Jornada 8h	
Hu_03	Tareas	Horas	Inicio	Fin
Tarea 01	Extraer las imágenes del dataset que genera el sistema de re-identificación mediante las rutas o directorios para mostrarlas en la sección de coincidencias y descriptores de la interfaz	20	24/05/2023	28/05/2023
Tarea 02	Implementar en el diseño de la interfaz unas etiquetas (labels) para mapear las imágenes y lograr visualizarlas de forma óptima y responsiva	10	29/05/2023	29/05/2023
Tarea 03	Implementar QTimer, una clase que proporciona un temporizador que puede utilizarse para actualizar las galerías de imágenes en un intervalo de tiempo específico.	50	30/05/2023	06/06/2023

Burndown Chart

Figura 31

Burndown Chart – Historia de Usuario 03.



Las horas estimadas, se obtienen al hacer una operación matemática de multiplicar el total de días estimados y la jornada horaria que se trabaja por día; por lo tanto, este sprint se compone de 10 días y 8 horas diarias de trabajo. El resultado obtenido es un valor de 80 horas, que se toma como máximo para el eje Y, como se observa en la Fig. 9. Con el transcurso de los días, este valor se reducirá progresivamente hasta alcanzar cero, cumpliendo así el objetivo del sprint.

Resultados de Sprint 02

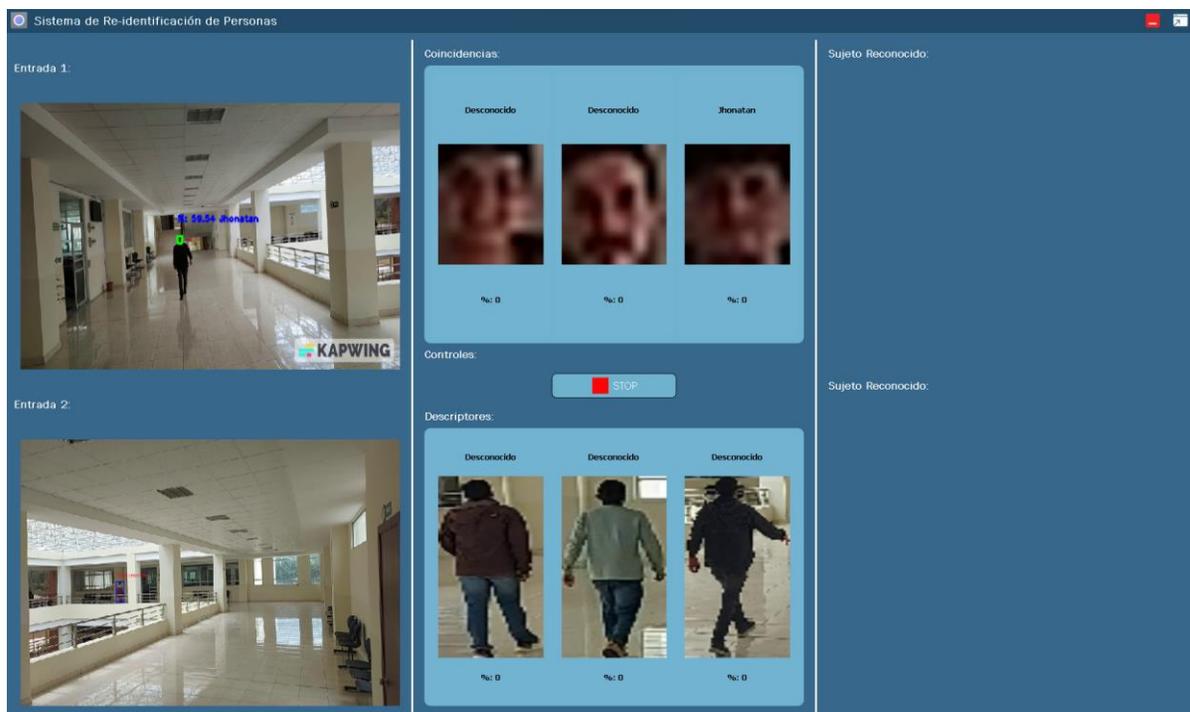
En este apartado, se muestra cómo se desarrolló el Sprint y los resultados obtenidos al implementar Qthread para visualizar en la interfaz las señales de video que generan las cámaras. También, se aprecia los resultados obtenidos al implementar las galerías de imágenes como los descriptores y coincidencias. Las imágenes se extraen del dataset generado por el sistema de re-identificación de personas que utiliza características fáciles y soft-biométricas como la textura.

Para que las galerías de imágenes se actualicen dinámicamente, se utiliza QTimer para controlar un temporizador y ejecutar las funciones necesarias para cumplir este requerimiento en un intervalo de tiempo determinado.

En la Fig. 10, se refleja la ejecución de la interfaz para visualizar en tiempo real las señales de video, las galerías de imágenes que representan las coincidencias y descriptores. Cabe mencionar que, para realizar la ejecución del código fuente se utilizó las grabaciones de prueba realizadas por los miembros del equipo de desarrollo.

Figura 32

Visualización de las señales de video, coincidencias y descriptores en tiempo real.

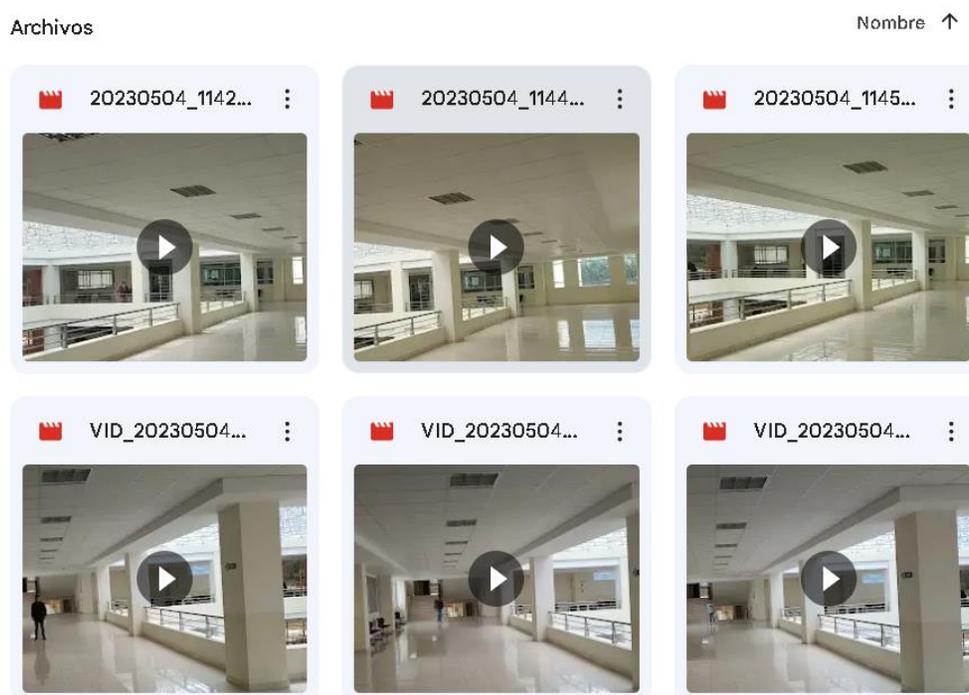


Resultados de las tareas de la Historia de Usuario 02

Para probar la visualización de las señales de video en la interfaz, previamente se grabaron 6 videos como se muestra en la Fig. 11. Estos videos son de distintas personas, habiendo uso tanto de la visión periférica que tendrá la cámara 1 y 2.

Figura 33

Conjunto de videos de prueba.



Nota. El conjunto de videos se los encuentra en el siguiente enlace:

https://drive.google.com/drive/folders/1I181rOgSV45ZITgTOeufRugu33wwlHK_?usp=drive_link

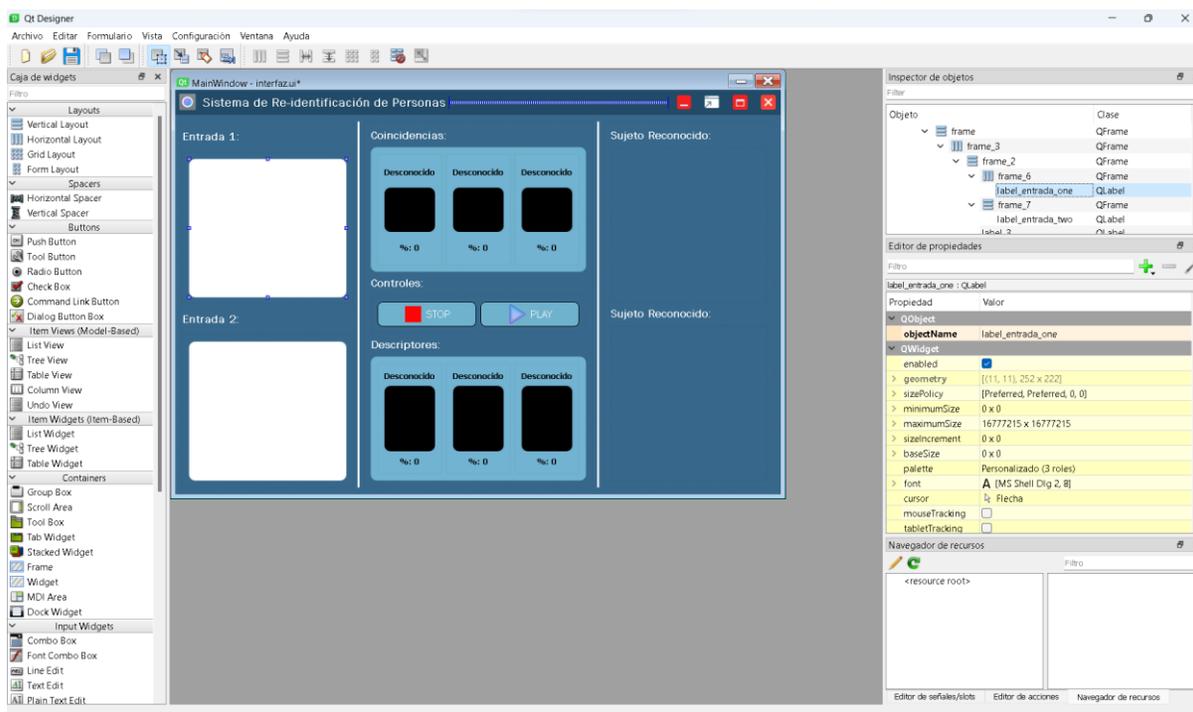
La interfaz esta compuestas por tres secciones, en la primera se visualiza las entradas de las señales de video, la segunda sección es destinada para las pequeñas galerías de imágenes que

representa a las coincidencias y los descriptores. La última sección contiene la imagen del rostro y de la parte posterior de la persona identificada.

En la Fig. 12 se aprecia que el frame_6 contiene una etiqueta label_entrada_one para mapear los frames de video de la cámara 1, de la misma forma para la entrada de la cámara 2.

Figura 34

Diseño para las entradas de video.



A continuación, en la Fig. 13 se muestra el código creado para visualizar ambas señales de video generadas por las cámaras de video en tiempo real. Para ello, se implementa la librería QThread que proporciona el marco de desarrollo PyQt5 para crear los hilos de ejecución, de esa forma se garantiza una experiencia fluida en la interfaz al ejecutar procesos en segundo plano.

Figura 35

Codificación para la visualización de las señales de video generadas por las dos cámaras de video.

```

#importar las librerías necesarias
from ultralytics import YOLO
import cv2
import os
import imutils
from ultralytics.yolo.utils.plotting import Annotator
from PyQt5.QtGui import *
from PyQt5.QtCore import *
#Definir una clase ReconocimientoFacial que hereda de QThread
class ReconocimientoFacial(QThread):
    def __init__(self, label_video):
        super().__init__()
        self.label_video=label_video
        dataPath = 'Data' #Nombre del dataset a listar
        self.imagePaths = os.listdir(dataPath)
        print('imagePaths=', self.imagePaths)
        self.face_recognizer = cv2.face.LBPHFaceRecognizer_create()
        self.face_recognizer.read('Face.xml')# Lectura del modelo entrenado
        self.model = YOLO('yolov8n-face.pt')
        #Captura de video desde una archivo de video
        self.cap = cv2.VideoCapture('pruebas/entrada_erick.mp4')
#Definición de señales personalizadas para comunicarse con la interfaz
Image_salida_upd=pyqtSignal(QImage)
Salida_Accuracy_Facial = pyqtSignal(str, str, int)
#método principal que se ejecuta en el hilo
    def run(self):
        self.hilo_corriendo=True
        self.cap.set(3, 640)#Ancho del video
        self.cap.set(4, 480)#Alto del video
        ...
        #Código necesario para la re-identificación de personas
        ...
        con=(200-result[1])/2#Valor de confianza
        if con < 70:#condición del valor de confianza
            cv2.putText(frame,"%: {:.2f}".format(con+20)+" "+'{}'.format(self.imagePaths[result[0]]),
(x1, y1-25),1,3,(255,0,0),2,cv2.LINE_AA)
            texto_formateado = "%: {:.2f}".format(con+20)
            persona = '{}'.format(self.imagePaths[result[0]])
            confianza = con
            #Cuando la condición es verdadera
            cv2.rectangle(frame, (x1, y1), (x2, y2),(0,255,0),3)
        else:
            cv2.putText(frame,'Desconocido',(x1,y1-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
            persona="Desconocido"
            texto_formateado=""
            cv2.rectangle(frame, (x1, y1), (x2, y2),(255, 0, 0),2)
            #Emitir la señal para actualizar la información en la interfaz
            self.Salida_Accuracy_Facial.emit(texto_formateado, persona, confianza)

```

```

#Redimensionar los frames del video
try:
    frame = imutils.resize(frame, width=800, height=800)
    Image=cv2.cvtColor(frame,cv2.COLOR_BGR2RGB)
    convertir_QT=QImage(Image.data,Image.shape[1],Image.shape[0],QImage.Format_RGB888)
    pic=convertir_QT.scaled(self.label_video.width(),self.label_video.height(),Qt.KeepAspectRatio)

    self.Image_salida_upd.emit(pic)
except:
    print("An exception occurred")
    #Comprobar si el hilo debe detenerse
    if self.hilo_corriendo==False:
        break
#Liberar el video y cerrar todas las ventanas de visualización
self.cap.release()
cv2.destroyAllWindows()
#Detiene el hilo de ejecución
def stop(self):
    self.hilo_corriendo=False
    self.quit()

```

Nota. El código que se presenta permite visualizar en la interfaz las señales de video en tiempo real

Resultados de las tareas de la Historia de Usuario 03

En la Fig. 14, se evidencia el código creado para extraer los conjuntos de imágenes que se encuentran en el dataset generado por el sistema de re-identificación de personas. Para obtener las imágenes, el algoritmo utiliza algunas funciones y métodos del módulo de Python “os” que permite realizar tareas relacionadas con el sistema operativo, como interactuar con archivos, directorios, procesos y variables de entorno, entre otras cosas.

Figura 36

Código para extraer las rutas de las imágenes que se usan en la interfaz.

```

#importar las librerías necesarias
import sys
import os
from PyQt5.QtGui import *
from PyQt5 import QtCore, QtWidgets
from PyQt5.uic import loadUi
from PyQt5.QtMultimedia import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import *
from PyQt5.QtMultimediaWidgets import *
from Facial.recFace_class import ReconocimientoFacial
from Textura.recBody_class import ReconocimientoBody
from PyQt5.QtCore import QTimer
#Definir una clase VentanaPrincipal que hereda de QMainWindow
class VentanaPrincipal(QMainWindow):
    def __init__(self):
        super(VentanaPrincipal, self).__init__()
        loadUi("interfaz.ui", self)# Carga la interfaz de usuario desde un archivo .ui
        ...
        # Código necesario para el funcionamiento de la interfaz
        ...
#Obtener las rutas de los directorios con las imágenes para las coincidencias
def getImage(self):
    input_dir_path = "C:/Users/ASUS/IA-RI/Data"
    lista = [os.path.join(input_dir_path, file_name) for file_name in os.listdir(input_dir_path)]
    return lista
#Obtener las imágenes para coincidencias
def mostrarImagenDir(self, index):
    lista = []
    listaDir = self.getImage()
    files_names = os.listdir(listaDir[index])
    for file_name in files_names:
        image_path = os.path.join(listaDir[index], file_name)
        lista.append(image_path)
    return lista
#Obtener las rutas de los directorios con las imágenes para los descriptores
def getImageDes(self):
    input_dir_path = "C:/Users/ASUS/IA-RI/Data_Textura"
    lista = [os.path.join(input_dir_path, file_name) for file_name in os.listdir(input_dir_path)]
    return lista
#Obtener los Descriptores
def mostrarImagenDirDes(self, index):
    lista = []
    listaDir = self.getImageDes()
    files_names = os.listdir(listaDir[index])
    for file_name in files_names:
        image_path = os.path.join(listaDir[index], file_name)
        lista.append(image_path)
    return lista

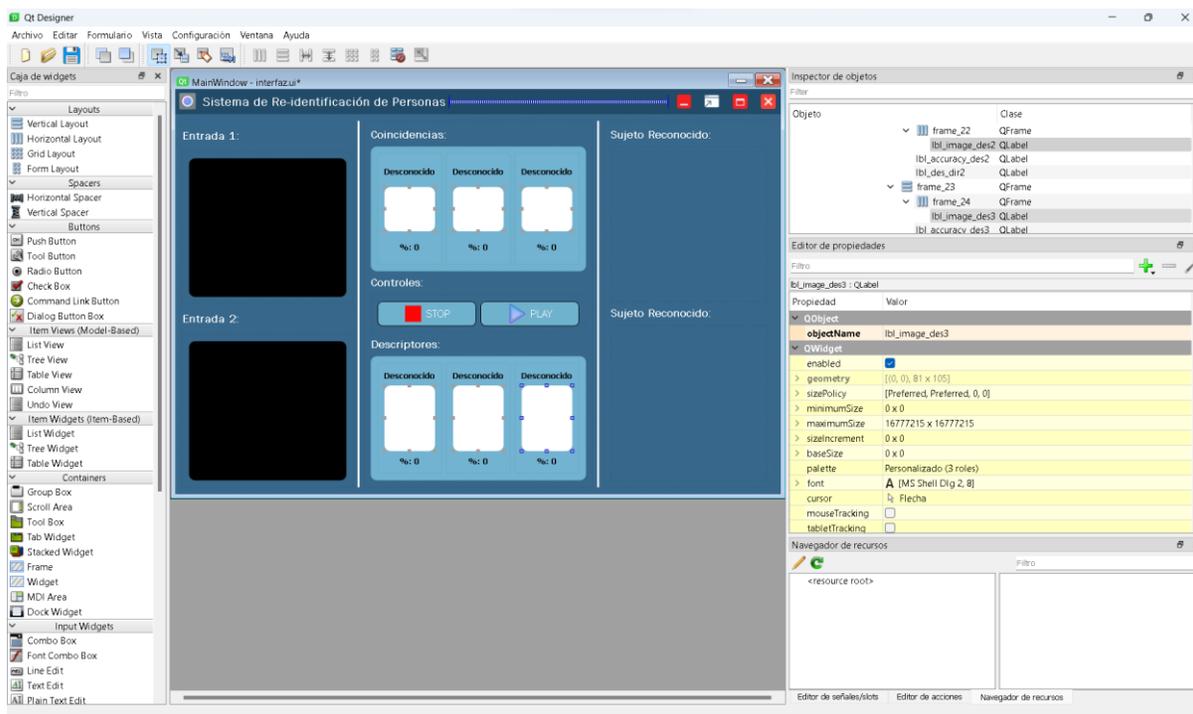
if __name__=="__main__":
    app=QApplication(sys.argv)
    mi_app=VentanaPrincipal()
    mi_app.start_carousel_Coincidencias_Descriptores()
    mi_app.play_video()
    mi_app.show()
    sys.exit(app.exec_())

```

Para el diseño de las pequeñas galerías de imágenes se utilizó el software designer. Las galerías están compuestas por tres frames que contienen labels o etiquetas de texto para mapear y visualizar las imágenes en tiempo real. Estos conjuntos de imágenes representan las coincidencias y descriptores en la interfaz de visualización, como se observa en la Fig.14.

Figura 37

Diseño para las coincidencias y descriptores.



Los conjuntos de imágenes no deben de ser estáticos, por el contrario, deben actualizarse dinámicamente. Para cumplir con este requerimiento se utiliza QTimer, una clase de framework PyQt5. La clase QTimer proporciona un temporizador que se manipula para para ejecutar funciones específicas en un intervalo de tiempo determinado.

A continuación, en la Fig. 15 se detalla el código creado para actualizar dinámicamente las listas de imágenes de las coincidencias y descriptores en la interfaz de visualización.

Figura 38

Código para actualizar las imágenes de las coincidencias y descriptores.

```

# Importar las bibliotecas necesarias
import sys
import os
from PyQt5.QtGui import *
from PyQt5 import QtCore, QtWidgets
from PyQt5.uic import loadUi
from PyQt5.QtMultimedia import *
from PyQt5.QtCore import *
from PyQt5.QtWidgets import *
from PyQt5.QtMultimediaWidgets import *
from PyQt5.QtCore import QTimer
# Definir una clase VentanaPrincipal que hereda de QMainWindow
class VentanaPrincipal(QMainWindow):
    def __init__(self):
        super(VentanaPrincipal, self).__init__()
        loadUi("interfaz.ui", self)
        #####Actualizar las imágenes#####
    def start_carousel_Coincidencias(self):
        self.current_image_index = 0
        self.timer = QTimer()
        self.timer.timeout.connect(self.update_image_Alex)
        self.timer.timeout.connect(self.update_image_Erick)
        self.timer.timeout.connect(self.update_image_Jhonatan)
        self.timer.start(2000) # Cambiar el intervalo de tiempo según las necesidades

    def update_image_Alex(self):
        imagenDir1 = self.mostrarImagenDir(0)
        if self.current_image_index >= len(imagenDir1):
            self.current_image_index = 0
        image_path = imagenDir1[self.current_image_index]
        pixmap = QPixmap(image_path)
        self.lbl_image_dir1.setPixmap(pixmap.scaled(self.lbl_image_dir1.size(), Qt.KeepAspectRatio))
        self.current_image_index += 1

    def update_image_Erick(self):
        imagenDir2 = self.mostrarImagenDir(1)
        if self.current_image_index >= len(imagenDir2):
            self.current_image_index = 0
        image_path = imagenDir2[self.current_image_index]
        pixmap = QPixmap(image_path)
        self.lbl_image_dir2.setPixmap(pixmap.scaled(self.lbl_image_dir2.size(), Qt.KeepAspectRatio))
        self.current_image_index += 1

    def update_image_Jhonatan(self):
        imagenDir3 = self.mostrarImagenDir(2)
        if self.current_image_index >= len(imagenDir3):
            self.current_image_index = 0
        image_path = imagenDir3[self.current_image_index]
        pixmap = QPixmap(image_path)
        self.lbl_image_dir3.setPixmap(pixmap.scaled(self.lbl_image_dir3.size(), Qt.KeepAspectRatio))
        self.current_image_index += 1

if __name__ == "__main__":
    app = QApplication

```

Nota. Al concluir este Sprint, no se encontraron inconvenientes para satisfacer plenamente los requerimientos establecidos.

Sprint 03: Visualización en tiempo real de la re-identificación de personas mediante el uso de características faciales y soft-biométricas, como la textura.

Durante la ejecución del Sprint 03, se dio especial importancia a la implementación de las Historias de usuario Hu_04, Hu_05 y Hu_06 en la interfaz de visualización, detalladas en la Tabla 2, las cuales establecen el objetivo de visualizar en tiempo real la re-identificación de una persona a través de las características faciales y soft-biométricas, como la textura; además, la visualización de información adicional como el nombre de la persona y el porcentaje de confianza que obtiene el sistema de re-identificación al realizar el reconocimiento del individuo.

Para que el operador tenga una respuesta rápida y efectiva ante eventos específicos en las señales de video, se requiere implementar alertas o notificaciones en tiempo real. Las alertas automáticas permiten reducir el tiempo y esfuerzo requerido para monitorear constantemente el flujo de video. En lugar de revisar todo el contenido, los operadores de seguridad o vigilancia pueden centrarse solo en situaciones relevantes, como la detección de personas no autorizadas o la identificación de individuos buscados por la Policía Nacional.

A continuación, se detalla minuciosamente los procesos empleados en estas Historias de Usuario durante el presente Sprint 03.

Historia de usuario 04: Visualización de la re-identificación de individuos en tiempo real utilizando datos biométricos blandos como la textura y los rasgos faciales

La Historia de Usuario Hu_04, delineada en la Tabla 2, establece la necesidad de mostrar en la interfaz la re-identificación en tiempo real de individuos, aprovechando las características faciales y soft-biométricas, como la textura.

Por lo antes mencionado, se requiere visualizar en tiempo real el reconocimiento facial en la señal de video generada por la cámara 1. Además, se visualizará el reconocimiento de la parte posterior de la persona mediante las características soft-biométricas, como la textura de la vestimenta, en la señal de video capturada por la cámara 2.

Historias de Usuario Detalladas

En la Tabla 11, se ofrece una descripción detallada de la Historia de Usuario Hu_04, que corresponde a la interfaz de visualización de señales de video generadas por un modelo de re-identificación de personas, que utiliza características fáciles y soft-biométricas como la textura. En esta tabla, se proporciona los nombres de los desarrolladores responsables en implementar la interfaz, la estimación en puntos de tiempo, y los criterios de aceptación establecidos para el desarrollo exitoso de la historia.

Tabla 14

Historia de usuario para la visualización en tiempo real de la re-identificación de personas, utilizando características faciales y soft-biométricas, como la textura.

Historia de usuario	
Código: Hu_04	Usuario: Usuario final
Nombre: Visualización de la re-identificación de individuos en tiempo real utilizando datos biométricos blandos como la textura y los rasgos faciales.	Número de sprint: 3
Prioridad: Alta	Riesgo en desarrollo: Bajo
Estimación en días: 17	Iteración asignada: 1

Desarrolladores responsables: Jhonatan Chiluisa, Erick Fernández

Descripción: Como usuario quiero visualizar en la interfaz, la re-identificación de la persona en tiempo real a través de las características faciales y soft-biométricas como la textura.

Validación:

- La interfaz presenta una alta presión en la re-identificación de personas basadas en las características faciales y soft-biométricas como la textura, al mostrar correctamente la identidad de cada individuo en tiempo real.
 - La visualización en tiempo real debe ser fluida y sin retrasos significativos, asegurando que la interfaz actualice rápidamente la información sobre las personas detectadas.
-

Sprint Backlog

La Tabla 12 proporciona un desglose detallado del Sprint Backlog 04, que enumera las tareas creadas durante el Sprint, junto con Jhonatan Chiluisa y Erick Fernández como gestores del proyecto, los días asignados para el Sprint, el tiempo previsto en horas y el estado actual de cada trabajo. Se puede observar que el Sprint Backlog está completamente terminado, lo que indica que todas las operaciones programadas se terminaron con éxito.

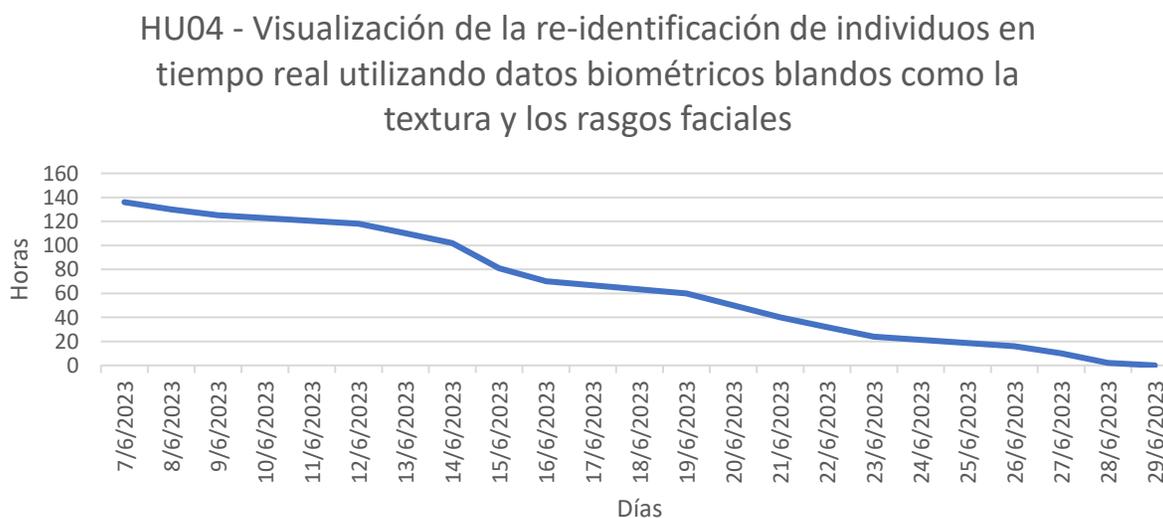
Tabla 15*Sprint Backlog 04.*

Sprint 3	Fecha Inicio 07/06/2023	Fecha Fin 29/06/2023	Jornada 8h	
Hu_04	Tareas	Horas	Inicio	Fin
Tarea 01	<p>Visualizar en la interfaz de forma clara y legible el reconocimiento facial de la persona identificada que genera el flujo de video capturado por la cámara 1 para que el personal de seguridad tome acciones rápidas y oportunas ante eventos específicos.</p> <p>Visualizar la re-identificación usando características soft-biométricas como la textura de la parte posterior de la persona que genera la señal de video de la cámara 2 para tener más precisión y confianza en el reconocimiento de la persona identificada por la primera cámara.</p>	68	07/06/2023	15/06/2023
Tarea 02	<p>Visualizar la re-identificación usando características soft-biométricas como la textura de la parte posterior de la persona que genera la señal de video de la cámara 2 para tener más precisión y confianza en el reconocimiento de la persona identificada por la primera cámara.</p>	68	16/06/2023	29/06/2023

Burndown Chart

Figura 39

Burndown Chart – Historia de Usuario 04.



Las horas estimadas, se calculan al multiplicar el total de días estimados por la jornada laboral diaria, que en este caso es de 8 horas. Para este Sprint en particular, se planifico un periodo de 17 días de trabajo, lo que resulta un total de 136 horas estimadas, como se muestra en la Fig. 16. Este valor máximo se presenta en el eje Y del Burndown Chart. A medida que transcurren los días y se avanza en la ejecución del sprint, este valor se reducirá progresivamente hasta llegar a cero.

Historia de usuario 05: Visualización de información adicional de la persona identificada como el nombre y el porcentaje de confianza que genera el modelo de re-identificación

La Historia de Usuario Hu_05, que se detalla en la Tabla 2, indica que se necesita visualizar información adicional como el nombre de las personas identificadas y el porcentaje de confianza que

proporciona el modelo de re-identificación usando características faciales y soft-biométricas como la textura.

Es importante mencionar que, esta información adicional se debe integrar en los conjuntos de imágenes que representan las coincidencias y descriptores en la interfaz de visualización; además la información debe actualizarse automáticamente mientras transcurre la re-identificación de la persona.

Historia de Usuario Detallada

La Tabla 13 que se presenta a continuación, muestra en detalle los elementos que componen la Historia de Usuario Hu_05, como los responsables a cargo del desarrollo de la interfaz, los puntos de la historia y los más importantes, las validaciones que satisfacen el desarrollo exitoso de la historia antes mencionada.

Tabla 16

Historia de usuario para la visualización en tiempo real de información adicional en la re-identificación de personas.

Historia de usuario	
Código: Hu_05	Usuario: Usuario final
Nombre: Visualización de información adicional de la persona identificada como el nombre y el porcentaje de confianza que genera el modelo de re-identificación.	Número de sprint: 3
Prioridad: Alta	Riesgo en desarrollo: Bajo
Estimación en días: 12	Iteración asignada: 1

Desarrolladores responsables: Jhonatan Chiluisa, Erick Fernández

Descripción: Como usuario quiero visualizar en la interfaz, información adicional sobre las personas identificadas.

Validación:

- La información adicional mostrada en la interfaz es precisa y correspondiente a cada persona identificada, evitando errores o inconsistencias.
 - Esta información adicional se actualiza en tiempo real, reflejando cualquier cambio o nueva identificación de personas mientras se visualiza la interfaz.
-

Sprint Backlog

La Tabla 14, proporciona una descripción de tallada del Sprint Backlog 05, que incluye las tareas planificadas en el Sprint, los responsables asignados para cada tarea como son Jhonatan Chiluisa y Erick Fernández, los días programados para su desarrollo, el tiempo estimado en horas y el estado actual de cada tarea.

Es importante destacar que el Sprint Backlog ha sido concluido satisfactoriamente, lo que permite confirmar el éxito en la finalización de todas las actividades previamente planificadas.

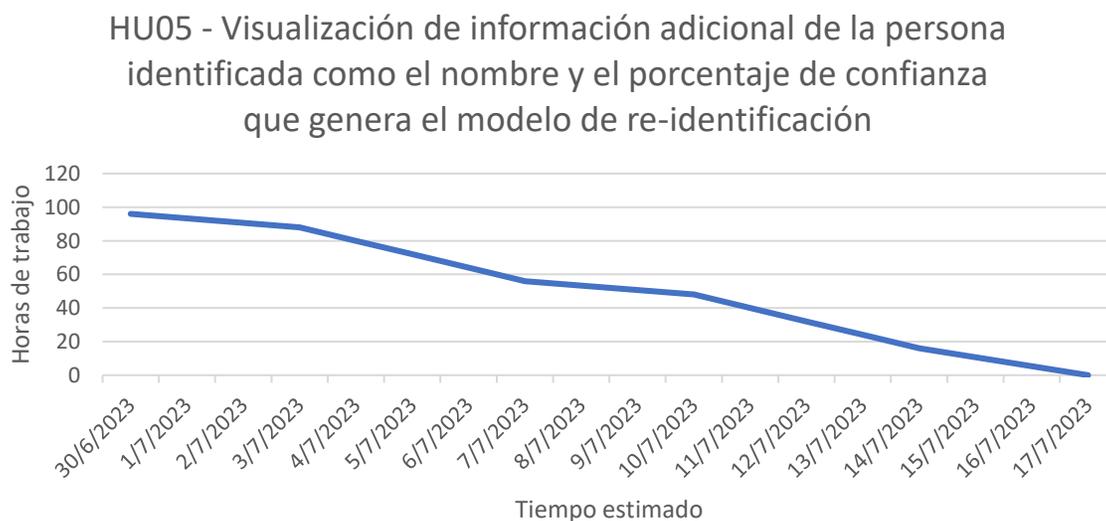
Tabla 17*Sprint Backlog 05.*

Sprint 3	Fecha Inicio 30/06/2023	Fecha Fin 17/07/2023	Jornada 8h	
Hu_05	Tareas	Horas	Inicio	Fin
Tarea 01	Implementar labels o etiquetas de texto en el diseño de la interfaz para visualizar la información adicional en el conjunto de coincidencias y descriptores.	48	30/06/2023	07/07/2023
Tarea 02	Implementar QSignal, una clase que representa una señal que puede ser emitida por un objeto en tiempo real para establecer interacciones entre la interfaz de visualización y la lógica del sistema.	48	10/07/2023	17/07/2023

Burndown Chart

Figura 40

Burndown Chart – Historia de Usuario 05.



Las horas estimadas, se determinan al multiplicar el número total de días estimados por la jornada laboral diaria estipulada como 8 horas. En el presente Sprint, se ha planificado un periodo de 12 días, lo que equivale a un total de 96 horas estimadas, tal como se observa en la Fig. 17. Este valor se presenta en el eje Y del Burndown Chart. A consecuencia que pasan los días y se desarrolla el sprint, este valor se reducirá progresivamente hasta alcanzar cero.

Historia de usuario 06: Visualización de alertas automáticas en la interfaz al momento de identificar una persona

La Historia de usuario Hu_06 que se especifica en la Tabla 2, indica que se requiere implementar algún tipo de alerta en tiempo real en la interfaz de visualización de señales de video generadas por un modelo de re-identificación de personas, lo cual ofrece múltiples beneficios, desde mejorar la eficiencia

y seguridad, hasta reducir costos operativos y facilitar la identificación de individuos en entornos de vigilancia.

Para alertar al operario de que el sistema identificó a una persona, se hará un cambio de color a los frames contenedores de la interfaz que componen la sección de coincidencias y descriptores. La modificación del color, se llevará a cabo únicamente si el porcentaje de confianza proporcionado por el modelo de re-identificación es igual o supera los umbrales previamente establecidos y validados por la comunidad científica, tanto para el reconocimiento facial como para el análisis soft-biométrico.

Historia de Usuario Detallada

A continuación, se presenta la Tabla 15 con detalles sobre los elementos que conforman la Historia de usuario Hu_06, incluyendo los responsables encargados del desarrollo de la interfaz, los puntos de la historia y, los más relevante, las validaciones que aseguran el éxito en la implementación de dicha historia.

Tabla 18

Historia de usuario para la visualización de alertas automáticas al momento de identificar una persona.

Historia de usuario	
Código: Hu_06	Usuario: Usuario final
Nombre: Visualización de alertas automáticas	Número de sprint: 3
en la interfaz al momento de identificar una persona.	
Prioridad: Alta	Riesgo en desarrollo: Bajo
Estimación en días: 4	Iteración asignada: 1

Desarrolladores responsables: Jhonatan Chiluisa, Erick Fernández

Descripción: Como usuario quiero visualizar notificaciones o alertas en tiempo real de la re-identificación de las personas en la interfaz.

Validación:

- Las alertas automáticas presentadas en la interfaz son precisas y corresponden de manera exacta a cada re-identificación de persona realizada.
 - Las alertas se muestran en tiempo real, sin retrasos significativos, para que el usuario pueda tomar acciones de manera oportuna.
-

Sprint Backlog

En la Tabla 15 se presenta el Sprint Backlog 06, que contiene el detalle de las tareas y elementos de trabajo identificados por el equipo de desarrollo para alcanzar los objetivos del sprint actual. Es esencial destacar que Jhonatan Chiluisa y Erick Fernández son los encargados de ejecutar las tareas eficientemente. Además, es gratificante informar que se ha completado en su totalidad el Sprint Backlog.

Tabla 19

Sprint Backlog 06.

Sprint 3	Fecha Inicio 18/07/2023	Fecha Fin 21/07/2023	Jornada 8h	
Hu_06	Tareas	Horas	Inicio	Fin
	Seleccionar un color adecuado para que la			
Tarea 01	alerta automática llame la atención del usuario.	12	18/07/2023	18/07/2023

Tarea 02	Utilizar el valor de confianza que nos proporciona el modelo de re-identificación de personas para determinar la activación de la alerta en tiempo real.	20	19/07/2023	21/07/2023
----------	--	----	------------	------------

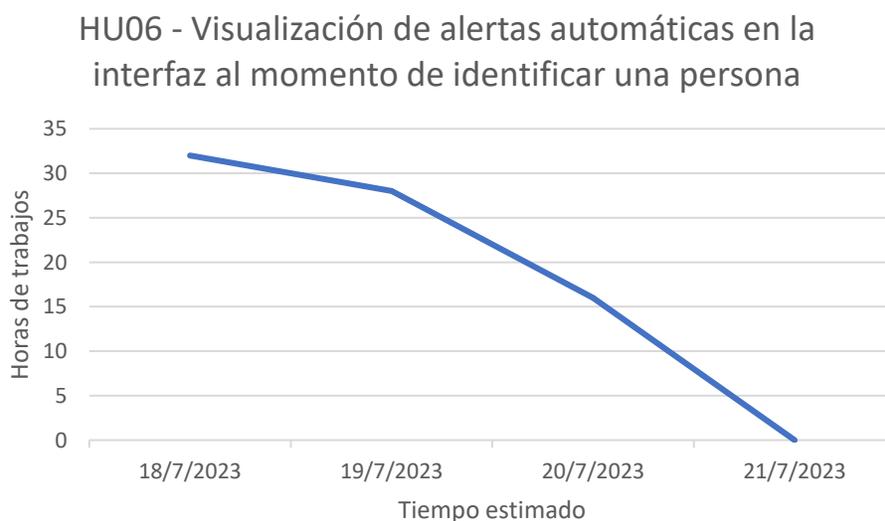
Burndown Chart

En la Fig. 18, denominada “Burndown Chart – Hu_06”, se presenta el progreso alcanzado durante los días estimados para el desarrollo del presente Sprint. En el eje X se presentan las fechas correspondientes al intervalo de tiempo desde el 18/07/2023 hasta el 21/07/2023, especificadas en la Tabla 15. Por otro lado, el eje Y muestra el número total de horas estimadas, calculado multiplicando el total de días estimados (4 días) por las horas trabajadas por días (8 horas), lo que resulta en un valor máximo de 32 horas.

A medida que avanzan los días, el valor de horas en el eje Y se reducirá progresivamente hasta llegar a cero, lo que indicara que se ha cumplido exitosamente el objetivo del Sprint.

Figura 41

Burndown Chart – Historia de Usuario 06.



Resultados del Sprint 03

Los logros alcanzados durante el Sprint son la visualización de la re-identificación de las personas a través de las características faciales y soft-biométricas como la textura de la vestimenta; además, la visualización de la información adicional como el nombre de la persona identificada y el porcentaje de confianza que genera la lógica del sistema.

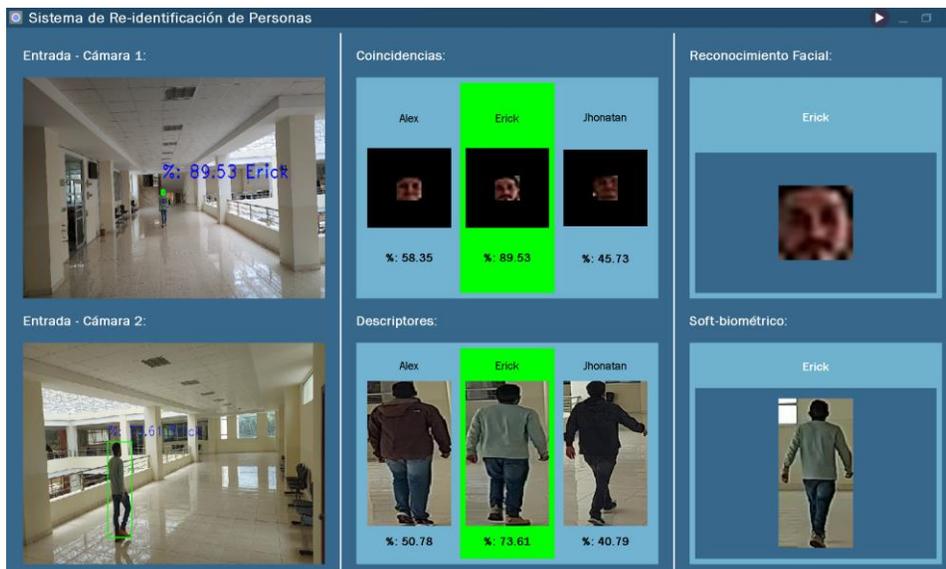
El valor de confianza requerido para el reconocimiento facial debe ser igual o mayor al 80%, mientras que para el análisis soft-biométrico de la textura, se debe alcanzar o superar el 60%. Estos valores activarán la alerta tanto en el conjunto de coincidencias como en los descriptores de la interfaz de visualización, respectivamente.

Cabe señalar que toda la información presentada en la interfaz de visualización al usuario es obtenida en tiempo real de la lógica del sistema de re-identificación humana que utiliza características faciales y soft-biométricas como la textura de la vestimenta.

Como se aprecia en la Fig. 19, en la primera sección de la interfaz, la entrada 1 muestra las señales de video capturadas por la primera cámara y de la misma forma con la entrada 2 que muestra el flujo de video que transmite la cámara 2. En la siguiente sección de la interfaz, se muestra el conjunto de imágenes de las coincidencias y descriptores; además, el nombre de las personas identificadas y el porcentaje de confianza que activara la alerta para que llame la atención del usuario y actúe de forma eficiente ante el reconocimiento de un sujeto.

Figura 42

Interfaz de visualización de señales de video generadas por un modelo de re-identificación a través de características faciales y soft-biométricas, como la textura.



Finalmente, en la última sección de la interfaz, se presenta imágenes tanto del rostro y la parte posterior de la persona reconocida como se evidencia en la Fig. 19. Para demostrar el funcionamiento de

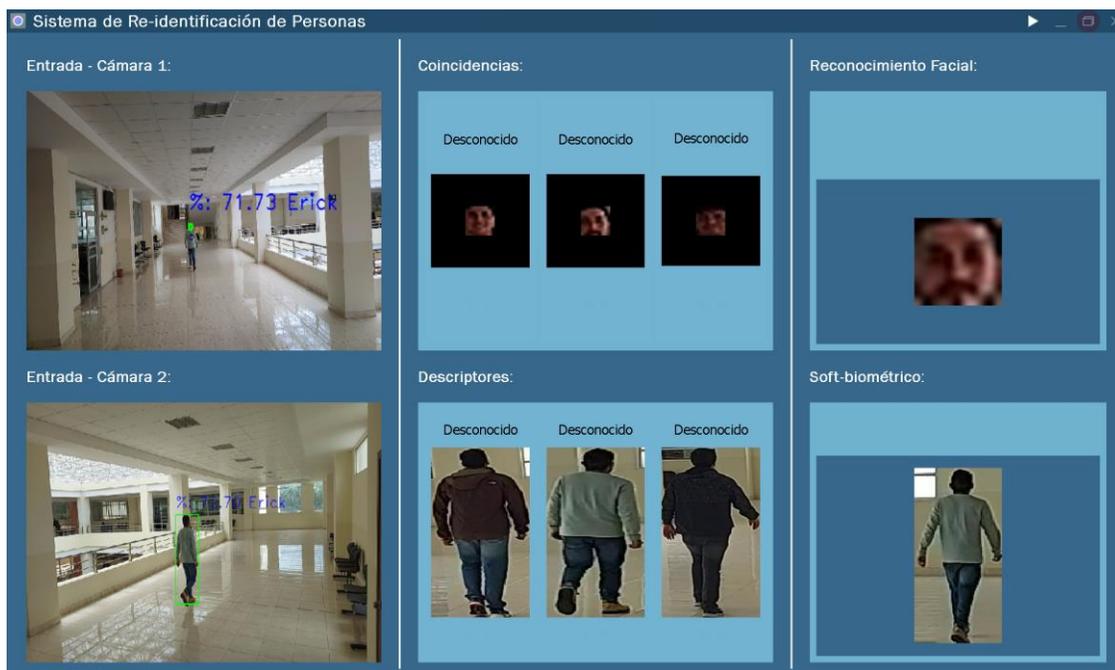
la interfaz se toma como ejemplo el reconocimiento de la persona Erick como un porcentaje de confianza en el reconocimiento facial del 89.53% y 73.61% en el reconocimiento de la parte posterior de la misma persona, evidenciando así la presión en la re-identificación y la legibilidad de la información presentada en la interfaz.

Resultados de las tareas de la historia de usuario 04

Gracias a las clases Qthread proporcionadas por PyQt5, logramos separar el proceso de reconocimiento facial, que opera en la entrada 1 y la re-identificación posterior de la persona basada en características soft-biométricas como la textura de la ropa, que se obtiene de la entrada 2. Como se aprecia en la Fig. 20, al iniciar la interfaz, los procesos se ejecutan en segundo plano, lo que permite visualizar más de dos flujos de video, conjuntos de imágenes de coincidencias, descriptores y los resultados de la identificación de una persona en particular de manera eficiente.

Figura 43

Visualización de la re-identificación facial y soft-biométrica.



Al emplear esta estrategia de optimización de recursos, centrada en el uso de hilos de procesamiento, logramos aprovechar eficazmente el hardware, garantizando que la respuesta de la interfaz no experimente retrasos notables. De esta manera, proporcionamos una experiencia fluida y transparente al usuario en todas las funcionalidades de la interfaz.

Resultados de las tareas de la historia de usuario 05

Para presentar información adicional sobre la re-identificación de una persona en la interfaz, se optó por utilizar una fuente tipográfica que no utilice remates (sans serif). En particular, hemos seleccionado “Franklin Gothic Book” para los títulos, subtítulos y texto como se ilustra en la fig. Es importante destacar que se empleó tamaños de fuentes específicos: 18pt para los títulos, 16pt para los subtítulos y 14pt para el texto, con el objetivo de asegurar una presentación legible y atractiva. Además,

se ha elegido colores como el blanco, que aporta armonía y serenidad, y el negro, que confiere elegancia a la fuente tipográfica empleada en la interfaz.

Figura 44

Visualización de información adicional para la identificación de una persona.



La informacional adicional, como el nombre de la persona identificada y el porcentaje de confianza, se extrae del sistema de re-identificación. Gracias a las clases Qsignal de PyQt5, logramos actualizar esta información dinámicamente en tiempo real, ya que se emiten directamente desde el backend hacia la interfaz de visualización.

Resultados de las tareas de la historia de usuario 06

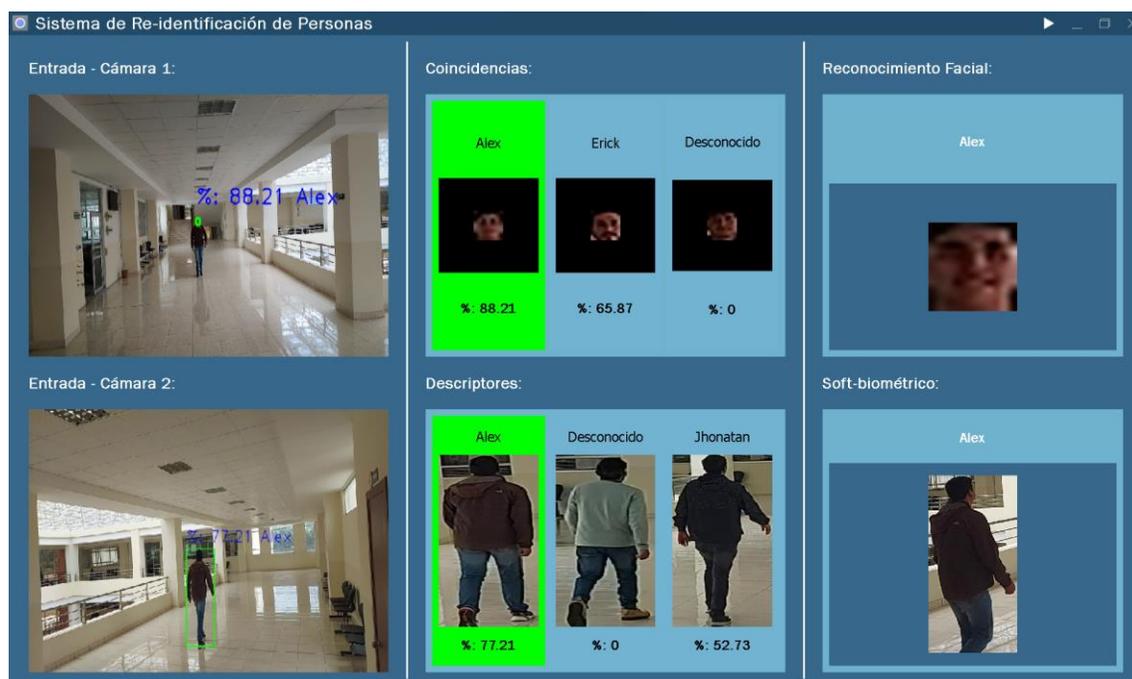
Para activar la alerta al identificar a una persona, se hace uso del valor de confianza proporcionado por el modelo de re-identificación. En el caso del reconocimiento facial, este valor debe

ser mayor o superior al 80% (Hong et al., 2021), para la alerta se active cuando se encuentra una coincidencia con la persona identificada a través de sus características faciales.

De manera similar, el valor de confianza necesario para la identificación de la parte posterior de la persona debe ser mayor o superior al 60% como se muestra en la Fig. Cabe señalar que estos rangos de valores de confianza tanto para el reconocimiento facial como para el soft-biométrico han sido evaluados y establecidos por la comunidad científica.

Figura 45

Activación de la alerta al identificar una persona.



Es esencial destacar que el color de la alerta se ha seleccionado como verde, ya que para el usuario simboliza una sensación positiva de seguridad.

Capítulo IV

Validación del sistema

Este capítulo se trata sobre la validación de la interfaz gráfica de usuario creada especialmente para visualizar sistemas de re-identificación. La aprobación de la interfaz gráfica de usuario es necesaria para comprobar que se ajusta a los objetivos y exigencias de los usuarios, así como para garantizar la usabilidad del sistema de re-identificación.

Marco de evaluación

El principal objetivo es evaluar la calidad de uso de la interfaz gráfica de usuario de un sistema de re-identificación, con métricas de usabilidad y experiencia del usuario. Estas métricas permiten medir factores como la eficiencia, eficacia y la satisfacción del usuario

Es esencial identificar las posibles áreas en las que hay que desarrollar y optimizar la interfaz gráfica de usuario mediante técnicas de evaluación. Además, se denotará cómo influyen las preferencias y necesidades de los usuarios en el diseño de la interfaz y en la UX.

Evaluar y cuantificar la usabilidad constituye un desafío constante en la búsqueda de mejoras en la experiencia del usuario. Después de interactuar con el sistema en tiempo real, el usuario proporcionó comentarios, medidas de tiempo, lo que permitió obtener una evaluación precisa de la aplicación, en este caso la interfaz gráfica. Este proceso involucró la implementación de un cuestionario, un método reconocido y económicamente viable para medir la usabilidad de manera habitual, (Finstad, 2010).

Aplicación de métricas de usabilidad y experiencia de usuario

La evaluación de este sistema contó con la participación de 10 usuarios que se dividieron en dos categorías: cinco usuarios con un alto nivel de conocimiento (HL) en Re-identificación, incluyendo

docentes universitarios, estudiantes de ingeniería de software y expertos en usabilidad, y cinco usuarios con conocimiento básico en habilidades informáticas (ML), como personal de vigilancia de una entidad financiera y docentes en educación primaria. Las métricas aplicadas se enfocaron en los aspectos de usabilidad previamente mencionados en el capítulo II, como la "eficacia, eficiencia y satisfacción", y se vincularon con factores de la experiencia del usuario (UX). Consecuentemente, se diseñó y administró un cuestionario que consta de 5 preguntas para cada uno de los tres factores de usabilidad, dirigido a todos los usuarios.

Eficacia

La eficacia es medida en función de si el usuario es capaz de resolver una tarea con la aplicación. Se aplica la escala de Likert para medir la eficacia de la interfaz gráfica.

Tabla 20

Cuestionario para medir la eficacia.

Preguntas	
Q1	La identificación del sujeto fue exitosa.
Q2	Observación clara de los resultados identificación.
Q3	La identificación del sujeto se realiza con el porcentaje de precisión.
Q4	La interfaz facilita la tarea de re-identificación.
Q4	La disposición de la visualización de las cámaras en la interfaz facilita el seguimiento del sujeto.

Cada usuario respondió de la siguiente manera:

Tabla 21

Resultados de la aplicación de cuestionario de eficacia a los diferentes usuarios.

Usuario	Q1	Q2	Q3	Q4	Q5
HL1	5	5	3	5	5
HL2	3	4	5	4	4
HL3	5	4	5	3	5
HL4	4	5	5	5	4
HL5	4	3	2	5	5
ML1	5	4	5	3	4
ML2	4	4	3	4	5
ML3	5	4	2	5	4
ML4	3	4	3	5	3
ML5	5	3	5	5	5
Promedio	4,3	4	3,8	4,4	4,4

Figura 46

Resultados representados en tasa de éxito al momento de realizar una tarea



Como promedio entre las diferentes puntuaciones de los usuarios para las preguntas del cuestionario de eficacia, la tasa de éxito es de 4,18 lo que quiere decir que el sistema es eficaz, tiene anticipación, es útil y alcanzable.

Eficiencia

La eficiencia se mide mediante el esfuerzo que realiza el usuario para cumplir sus objetivos. Para medir la eficiencia de la interfaz gráfica de usuario se tomará en cuenta el tiempo de cada lapso en que los usuarios logren finalizar una tarea. El tiempo es medido en segundos.

Tabla 22*Cuestionario para medir la eficiencia*

Preguntas	
Q1	Identificación del sujeto mediante la interfaz.
Q2	Interpretación de los resultados gráficos de reconocimiento e identificación.
Q3	Compresión de la relación entre el porcentaje de precisión con la alerta de sujeto identificado.
Q4	Asimilación del proceso de re-identificación.
Q5	Manejo de los controles de la interfaz.

Tabla 23*Resultados en segundos de la aplicación de cuestionario de eficiencia a los diferentes usuarios.*

Usuario	Q1	Q2	Q3	Q4	Q5
HL1	15	8	35	20	10
HL2	10	10	32	21	5
HL3	12	7	31	22	6
HL4	11	5	35	19	9
HL5	10	10	31	18	10
ML1	18	9	37	21	15
ML2	19	8	38	23	14

Usuario	Q1	Q2	Q3	Q4	Q5
ML3	20	10	36	22	13
ML4	11	8	39	23	12
ML5	10	6	40	25	11
Promedio	13,6	8,1	35,4	21,4	10,5

Figura 47

Resultados de la medición del tiempo en que los usuarios logran un objetivo con la aplicación.



Como resultado de la evaluación de la eficiencia del sistema se obtuvo un promedio de 17,8 segundos, tiempo en el que un usuario termina de realizar una tarea y comprende el sistema. Este tiempo es un buen resultado y quiere decir que el sistema es eficiente, autónomo, accesible, el usuario tiene una excelente curva de aprendizaje y mínima sorpresa.

Satisfacción

En la evaluación de la satisfacción, se usa de nuevo la escala de Likert, a diferencia que en esta sección se valorará las opiniones de los usuarios con respecto al uso de la interfaz gráfica.

Tabla 24

Cuestionario para medir la satisfacción.

Preguntas	
Q1	El sistema cumple las necesidades.
Q2	Esta aplicación tiene preferencia ante otras.
Q3	La interfaz es intuitiva y fácil de usar.
Q4	La disposición de los elementos de la interfaz gráfica es agradable.
Q5	La aplicación tiene una excelente experiencia de usuario.

Tabla 25

Resultados de la aplicación de cuestionario de satisfacción a los diferentes usuarios.

Usuario	Q1	Q2	Q3	Q4	Q5
HL1	4	5	4	4	5
HL2	4	5	5	5	5
HL3	3	4	4	5	4
HL4	4	5	4	4	5

Usuario	Q1	Q2	Q3	Q4	Q5
HL5	4	5	5	5	4
ML1	5	4	5	5	5
ML2	5	5	4	4	5
ML3	4	5	5	5	5
ML4	5	4	4	4	4
ML5	4	4	4	5	5
Promedio	4,2	4,6	4,4	4,6	4,7

Figura 48

Resultados de la puntuación del cuestionario de satisfacción.



Se obtuvo un promedio de total de 4,5 en lo que es satisfacción del usuario. La alta satisfacción del usuario al utilizar la interfaz gráfica descarta la necesidad de cambios futuros o nuevas implementaciones. La interfaz gráfica de usuario para la visualización de sistemas de re-identificación es satisfactoria, familiar, deseable, creíble y valiosa para quien la use.

Aplicación de test de escala de usabilidad del sistema (SUS)

Esta sección se centra en la aplicación del Test de Escala de Usabilidad del Sistema (SUS), este cuestionario proporciona un resultado que cuantifica la usabilidad de la interfaz gráfica de usuario para la visualización de señales de video generadas por un modelo de re-identificación de personas.

Para ejemplificar el cálculo de System Usability se usan las preguntas Q_n de la Tabla 1 y los valores de la escala de Likert de la Tabla 2, las fórmulas mencionadas en el capítulo anterior y se realizan con el usuario HL1.

El usuario HL1 respondió a $Q_{impares}$ y se obtuvieron $E_{impares}$. Al aplicar la formula obtenemos lo siguiente:

$$R_{impares} = \sum_{E_{impares}} x - 5$$

$$R_{impares} = (E_1 + E_3 + E_5 + E_7 + E_9) - 5$$

$$R_{impares} = (5 + 5 + 5 + 5 + 5) - 5$$

$$R_{impares} = 25 - 5$$

$$R_{impares} = 20$$

Ahora para Q_{pares} con sus respectivos E_{pares} y aplicando la formula se deduce lo siguiente:

$$R_{pares} = 25 - \sum x_{E_{pares}}$$

$$R_{pares} = 25 - (1 + 1 + 1 + 1 + 1)$$

$$R_{pares} = 25 - 5$$

$$R_{pares} = 20$$

Finalmente, para el cálculo de System Usability SU del usuario HL1, se presenta lo siguiente:

$$SU = (R_{impares} + R_{pares}) \cdot 2.5$$

$$SU = (20 + 20) \cdot 2.5$$

$$SU = 100$$

El cálculo de System Usability (SU) se aplicó a los resultados obtenidos en la aplicación del test de System Usability Scale a los 10 usuarios anteriormente mencionados y se presenta en la siguiente tabla.

Tabla 26

Resultados de la aplicación del test de SUS a 10 usuarios.

Usuario	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SU
HL1	5	1	5	1	5	1	5	1	5	1	100
HL2	4	5	4	2	5	2	5	1	5	1	80
HL3	4	2	4	1	4	1	4	2	4	2	80
HL4	4	1	4	1	5	2	4	1	4	1	87,5

Usuario	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SU
HL5	5	2	4	1	5	2	4	1	5	2	87,5
ML1	5	4	5	3	5	1	4	1	5	2	82,5
ML2	4	2	3	1	5	1	5	2	3	1	82,5
ML3	3	1	5	2	5	1	5	1	4	2	87,5
ML4	5	3	5	1	4	2	5	1	5	1	90
ML5	5	1	5	1	3	1	5	4	5	2	85
Promedio											86,25

La interpretación de los resultados obtenidos a partir del test del System Usability Scale Usabilidad (SUS) se basa en el promedio de la puntuación total de la encuesta en la que participaron todos los usuarios. La valoración de System Usability obtenida es de 86,25, puntaje que se sitúa en un rango alto de System Usability Scale (SUS), que generalmente esta entre 85 y 100. La valoración de SU indica que los usuarios experimentan interacciones fluidas, encuentran que el sistema es intuitivo y eficiente, y están altamente satisfechos con su experiencia de uso.

Capítulo V

Conclusiones

En la presente sección, se explican las conclusiones obtenidas en este trabajo.

Tras examinar múltiples artículos científicos procedentes de fuentes de datos fiables, se obtuvo un profundo conocimiento sobre el enfoque de Diseño Centrado en el Usuario (DCU), la optimización de recursos computacionales mediante hilos de procesamiento, y las interfaces de software empleadas en la representación de sistemas de re-identificación de personas. Estas técnicas se aplicaron con éxito en la creación de una interfaz gráfica de usuario destinada a la visualización de señales de video de dos cámaras ubicadas estratégicamente en un entorno controlado donde. Las señales de video se generan a partir de un modelo de re-identificación de personas que utiliza características faciales y soft-biométricas como la textura, color y silueta.

Por otro lado, ante la necesidad de cumplir con un plazo ajustado para el desarrollo de la interfaz y considerando la probabilidad de cambios en los requisitos, se optó por emplear la metodología Scrum; de esta forma, se acelera y gestiona el proceso de desarrollo del software, con la intención de aumentar la satisfacción del usuario, asegurar entregas incrementales y promover una colaboración activa y una mejora continua en el equipo de desarrollo.

En la ejecución de este proyecto, se empleó el framework PyQt5 para satisfacer los requerimientos del usuario en cuanto al diseño y la operatividad de la interfaz. PyQt5 se utilizó para optimizar la velocidad de respuesta de la interfaz, debido a que proporciona clases para la gestión de hilos de ejecución, permitiendo la visualización en tiempo real de las transmisiones de video de ambas cámaras. Esto amplió significativamente la capacidad de gestionar la interacción del usuario con el

sistema de re-identificación de personas, mejorando así la efectividad en el reconocimiento de individuos en situaciones específicas y permitiendo al personal de seguridad actuar de manera eficiente.

En relación a la validación de los resultados, se ha confirmado que la interfaz gráfica de usuario satisface los criterios de usabilidad enfocados a efectividad, eficiencia y satisfacción estipulados por el estándar ISO-9241-11. Los usuarios pueden alcanzar sus objetivos a través de la aplicación con un esfuerzo temporal mínimo, lo que se traduce en una experiencia positiva respaldada por opiniones favorables sobre el sistema. En cambio, para la evaluación de la usabilidad de la interfaz se aplicó el test de System Usability Scale que reveló resultados positivos con una valoración de 86,25, porcentaje que indica que el sistema tiene una gran usabilidad.

El proyecto que se presenta abre múltiples posibilidades para investigaciones futuras. Por ejemplo, se plantea la expansión de la interfaz de visualización para poder operar con un número mayor de cámaras de video. Esta extensión, posibilita la amalgama de diversos enfoques, incluyendo detalles como textura, silueta, color y rasgos faciales, con el propósito de elevar la precisión en la tarea de re-identificación de personas.

Bibliografía

- Albornoz, M. C. (2014). *Diseño de Interfaz Gráfica de Usuario*.
- Alejandra, M.-O. D. (2016). Re-identificación de personas a través de sus características soft-biométricas en un entorno multi-cámara de video-vigilancia. *Ingeniería, Investigación y Tecnología, 17*(2), 257-271. <https://doi.org/10.1016/j.riit.2016.06.010>
- Ammar, S., Zaghdien, N., & Neji, M. (2017). *A framework for people re-identification in multi-camera surveillance systems*.
- Bevan, N., Carter, J., & Harker, S. (2015). ISO 9241-11 Revised: What Have We Learnt About Usability Since 1998? En M. Kurosu (Ed.), *Human-Computer Interaction: Design and Evaluation* (pp. 143-151). Springer International Publishing. https://doi.org/10.1007/978-3-319-20901-2_13
- Bianchi, F. (2023). *Coolors—The super fast color palettes generator!* Coolors.Co. <https://coolors.co/>
- Billinghurst, M., Grasset, R., & Looser, J. (2005). Designing augmented reality interfaces. *ACM Siggraph Computer Graphics, 39*(1), 17-22.
- Brooke, J. (1995). SUS: A quick and dirty usability scale. *Usability Eval. Ind., 189*.
- Camps, O., Gou, M., Hebble, T., Karanam, S., Lehmann, O., Li, Y., Radke, R. J., Wu, Z., & Xiong, F. (2017). From the Lab to the Real World: Re-identification in an Airport Camera Network. *IEEE Transactions on Circuits and Systems for Video Technology, 27*(3), 540-553. <https://doi.org/10.1109/TCSVT.2016.2556538>
- Carvalho, B., Henrique, C., & Mello, C. (2011). Scrum agile product development method -literature review, analysis and classification. *Product: Management & Development, 9*, 39-49. <https://doi.org/10.4322/pmd.2011.005>
- Cook, K. (2023). *Fondos de pantalla de cámara—FondosMil*. <https://fondosmil.com/camara>

- Encalada, J., & Carlos, J. (2015). *Implementación de técnicas de identificación de objetos aplicadas al reconocimiento de rostros en vídeo vigilancia del SIS-ECU-911*.
- Fayyaz, M., Yasmin, M., Sharif, M., Shah, J. H., Raza, M., & Iqbal, T. (2020). Person re-identification with features-based clustering and deep features. *Neural Computing and Applications*, 32(14), 10519-10540. <https://doi.org/10.1007/s00521-019-04590-2>
- Finstad, K. (2010). The Usability Metric for User Experience. *Interacting with Computers*, 22(5), 323-327. <https://doi.org/10.1016/j.intcom.2010.04.004>
- Fleming, P. (2019, noviembre 16). *This photo was taken in Chorleywood, Hertfordshire, England, UK*. iStock. <https://www.istockphoto.com/es/foto/foto-de-cambio-de-inclinaci%C3%B3n-de-la-m25-london-orbital-motorway-cerca-de-junction-17-gm1181235774-331225411>
- Frøkjær, E., Hertzum, M., & Hornbæk, K. (2000). Measuring usability: Are effectiveness, efficiency, and satisfaction really correlated? *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 345-352. <https://doi.org/10.1145/332040.332455>
- García-Martín, Á., & Martínez, J. M. (2015). People detection in surveillance: Classification and evaluation. *IET Computer Vision*, 9(5), 779-788. <https://doi.org/10.1049/iet-cvi.2014.0148>
- Gómez, L. S. M. (2000). *Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación*.
- González Duque, R. (2011). *Python para todos*. Creative Commons Reconocimiento.
- Grier, R. A., Bangor, A., Kortum, P., & Peres, S. C. (2013). The System Usability Scale: Beyond Standard Usability Testing. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 57(1), 187-191. <https://doi.org/10.1177/1541931213571042>
- Haya, A. (2019, noviembre 3). *Importance of UI/UX for Virtual Reality*. - *Virtual Reality Augmented Reality Technology Latest News*. <https://www.affinityvr.com/importance-of-ui-ux-for-virtual-reality/>

- He, X., Yan, S., Hu, Y., & Niyogi, P. (2005). Face Recognition Using Laplacian Faces. *IEEE transactions on pattern analysis and machine intelligence*, 27, 328-340. <https://doi.org/10.1109/TPAMI.2005.55>
- Hong, P., Wu, T., Wu, A., Han, X., & Zheng, W.-S. (2021). Fine-Grained Shape-Appearance Mutual Learning for Cloth-Changing Person Re-Identification. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10508-10517. <https://doi.org/10.1109/CVPR46437.2021.01037>
- Huang, L., Huang, Y., Ouyang, W., & Wang, L. (2019). Part-Aligned Pose-Guided Recurrent Network for Action Recognition. *Pattern Recognition*, 92. <https://doi.org/10.1016/j.patcog.2019.03.010>
- Jaiswal, A. (2022, septiembre 27). *Object Detection Using Haar Cascade: OpenCV*. <https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/>
- Jansen, B. J. (1998). The graphical user interface. *ACM SIGCHI Bulletin*, 30(2), 22-26. <https://doi.org/10.1145/279044.279051>
- Kaur, P., Krishan, K., Sharma, S. K., & Kanchan, T. (2020). Facial-recognition algorithms: A literature review. *Medicine, Science and the Law*, 60(2), 131-139. <https://doi.org/10.1177/0025802419893168>
- Kayes, I., Sarker, M., & Chakareski, J. (2016). Product backlog rating: A case study on measuring test quality in scrum. *Innovations in Systems and Software Engineering*, 12(4), 303-317. <https://doi.org/10.1007/s11334-016-0271-0>
- Lata, P., Kiran, C., Tungathurthi, B., Ram, H., Hanmanth, R. M. R., Govardhan, D., & Reddy, D. (2009). Facial recognition using eigenfaces by PCA. *SHORT PAPER International Journal of Recent Trends in Engineering*, 1.
- Le, B. P., Jeon, H., Truong, N., & Hak, J. (2019). Applying the Haar-cascade Algorithm for Detecting Safety Equipment in Safety Management Systems for Multiple Working Environments. *Electronics*, 8, 1079. <https://doi.org/10.3390/electronics8101079>

- Liao, L., Yang, Z., Liao, Q., Li, K., Li, K., Liu, J., & Tian, Q. (2020). A half-precision compressive sensing framework for end-to-end person re-identification. *Neural Computing and Applications*, 32(4), 1141-1155. <https://doi.org/10.1007/s00521-019-04424-1>
- Lignia Tuapanta, L. D., & Pichucho Tandalla, R. V. (2023). *Sistema de Re-Identificación de personas a través de las características faciales y soft-biométricas la textura de la vestimenta, y modelos holísticos y no holísticos, utilizando algoritmos de Visión por Computadora y Machine Learning* [BachelorThesis, Universidad de las Fuerzas Armadas ESPE. ESPEL. Carrera de Ingeniería en Software.]. <http://repositorio.espe.edu.ec/jspui/handle/21000/35825>
- Ljupco. (2020, mayo 6). *Full length profile shot of a male student with a backpack walking...* iStock. <https://www.istockphoto.com/es/foto/estudiante-masculino-con-una-mochila-caminando-gm1218775476-356258723>
- Llamas, N. (2017). *Reconocimiento facial no supervisado en programas de televisión*. Universidad Politécnica de Cataluña.
- López-Martínez, J., Juárez-Ramírez, R., Huertas, C., Jiménez, S., & Guerra-García, C. (2016). Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review. *2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)*, 141-148. <https://doi.org/10.1109/CONISOFT.2016.30>
- Makadia, M. (2023). *Planning Your Scrum Sprint: A Step-by-Step Guide to Agile Success*. Maruti Techlabs. <https://marutitech.com/guide-to-scrum-sprint-planning/>
- Manosalvas Salazar, C. A. (2017). *Diseño de un sistema embebido para el control de ingreso y salida de vehículos a través de internet, en el acceso principal de la ESPOCH*. [BachelorThesis, Escuela Superior Politécnica de Chimborazo]. <http://dspace.esPOCH.edu.ec/handle/123456789/7621>

- Martinez, W. L. (2011). Graphical user interfaces: Graphical user interfaces. *Wiley Interdisciplinary Reviews: Computational Statistics*, 3(2), 119-133. <https://doi.org/10.1002/wics.150>
- Microsoft. (2023). *Why Visual Studio Code?* <https://code.visualstudio.com/docs/editor/whyvscode>
- Mihaescu, R.-E., Chindea, M., Paleologu, C., Carata, S., & Ghenescu, M. (2020). Person Re-Identification across Data Distributions Based on General Purpose DNN Object Detector. *Algorithms*, 13(12), 343. <https://doi.org/10.3390/a13120343>
- Mohammed, M., Zeebaree, D., Mohsin Abdulazeez, A., Zebari, D., Fadhil, Z., Ahmed, F., & Rashed, E. (2021). *Machine Learning Algorithm for Developing Classroom Attendance Management System Based on Haar Cascade Frontal Face*. 1-6. <https://doi.org/10.1109/ISIEA51897.2021.9509990>
- Muñoz Flórez, L. M. A. K. (2017). *Sistema de re-identificación de personas en escenas interiores mediante cámaras de video*. 7.
- Nalbant, K. G., & Uyanik, Ş. (2021). *Computer Vision in the Metaverse*.
- Paneru, S., & Jeelani, I. (2021). Computer vision applications in construction: Current state, opportunities & challenges. *Automation in Construction*, 132, 103940. <https://doi.org/10.1016/j.autcon.2021.103940>
- Pessanha Santos, N. (2014). *Sistema de Visão para Aterragem Automática de UAV*. <https://doi.org/10.13140/RG.2.2.33991.75685>
- Poupyrev, I., Tan, D. S., Billingham, M., Kato, H., Regenbrecht, H., & Tetsutani, N. (2002). Developing a generic augmented-reality interface. *Computer*, 35(3), 44-50. <https://doi.org/10.1109/2.989929>
- Punn, N. S., Sonbhadra, S. K., Agarwal, S., & Rai, G. (2021). *Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques* (arXiv:2005.01385). arXiv. <http://arxiv.org/abs/2005.01385>

- Reid, D. A., Samangooei, S., Chen, C., Nixon, M. S., & Ross, A. (2013). Soft Biometrics for Surveillance: An Overview. En *Handbook of Statistics* (Vol. 31, pp. 327-352). Elsevier.
<https://doi.org/10.1016/B978-0-444-53859-8.00013-8>
- Reina, A. (2021, diciembre 6). *Aplicaciones de realidad aumentada en educación: +6 ejemplos*.
<https://www.droiders.com/aplicaciones-realidad-aumentada-educacion/>
- Ren, H., Wang, Z., Wang, Z., Chen, L., Satoh, S., & Hu, D. (2020). An Interactive Design for Visualizable Person Re-Identification. *Proceedings of the 28th ACM International Conference on Multimedia*, 4536-4538. <https://doi.org/10.1145/3394171.3414450>
- Romero, G. E. (2012). *Diseño de interfaces gráficas orientadas a presentaciones digitales de materiales educativos e investigaciones*.
- Sachdeva, S. (2016). Scrum Methodology. *International Journal of Engineering and Computer Science*, 5(6), Article 6. <http://www.ijecs.in/index.php/ijecs/article/view/1989>
- Sampath, H., Merrick, A., & Macvean, A. (2021). Accessibility of Command Line Interfaces. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 1-10.
<https://doi.org/10.1145/3411764.3445544>
- Sastoque, S., Narváez, C., & Garnica, G. (2016). *Metodología para la construcción de Interfaces Gráficas Centradas en el Usuario*.
- Satta, R. (2013). *Appearance Descriptors for Person Re-identification: A Comprehensive Review* (arXiv:1307.5748). arXiv. <http://arxiv.org/abs/1307.5748>
- Scrum Guide | Scrum Guides*. (s. f.). Recuperado 25 de julio de 2023, de <https://scrumguides.org/scrum-guide.html#sprint-backlog>
- Shore, J. (s. f.). *The Art of Agile Development*.

- Soleimanitaleb, Z., Keyvanrad, M. A., & Jafari, A. (2019). Object Tracking Methods:A Review. *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, 282-288.
<https://doi.org/10.1109/ICCKE48569.2019.8964761>
- Trigás Gallego, M. (2012). *Metodologia scrum*.
- Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, 3(1), 71-86. <https://doi.org/10.1162/jocn.1991.3.1.71>
- Valenzuela, V. (2016, enero 27). *Serif vs sans serif: Diferencias y Semejanzas* • *Silo Creativo*. Silo Creativo.
<https://www.silocreativo.com/serif-vs-sans-serif-diferencias-y-semejanzas/>
- Verdeguer Valderrama, D., & Campos Vasquez, N. (2021). Diseño E Implementación De Un Sistema De Identificación De Personas Para La Seguridad De Los Accesos A Condominios, Basado En El Algoritmo De Reconocimiento Facial LBPH Faces. *Proceedings of the 19th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Prospective and trends in technology and skills for sustainable social development" "Leveraging emerging technologies to construct the future"*. The 19th LACCEI International Multi-Conference for Engineering, Education, and Technology: "Prospective and trends in technology and skills for sustainable social development" "Leveraging emerging technologies to construct the future".
<https://doi.org/10.18687/LACCEI2021.1.1.213>
- Willman, J. (2021). Overview of PyQt5. En J. Willman (Ed.), *Modern PyQt: Create GUI Applications for Project Management, Computer Vision, and Data Analysis* (pp. 1-42). Apress.
https://doi.org/10.1007/978-1-4842-6603-8_1
- Willman, J. M. (2022). Creating GUIs with Qt Designer. En J. M. Willman (Ed.), *Beginning PyQt: A Hands-on Approach to GUI Programming with PyQt6* (pp. 217-258). Apress.
https://doi.org/10.1007/978-1-4842-7999-1_8

Wonsick, M., & Padir, T. (2020). A Systematic Review of Virtual Reality Interfaces for Controlling and Interacting with Robots. *Applied Sciences*, 10(24), Article 24.

<https://doi.org/10.3390/app10249051>

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). *Object Detection in 20 Years: A Survey*

(arXiv:1905.05055). arXiv. <http://arxiv.org/abs/1905.05055>

Anexos